



CRYPTOMATHIC

# White Paper

## EMV Key Management – Explained





## Introduction

This white paper strides to provide an overview of key management related to migration from magnetic stripe to chip in the payment card industry. The paper is written as an educational note that enables the reader to gain a good understanding of the certificates, keys, and security processes involved with EMV key management.

## What is EMV?

The technology move from magnetic stripe based payment cards to chip cards has now been underway for more than a decade. It was originally initiated by Europay (now part of MasterCard), MasterCard and Visa, and abbreviated as EMV<sup>1</sup>. The move has largely been regionally driven by security due to fraud, liability shift, and technology – e.g. contactless, and more recently mobile.

## EMV Chip

During the past decade chips have evolved with regards to capabilities, while the price has gone down, so acquiring cards with cryptographic co-processors and dual interfaces for contact and contactless payment is quite inexpensive. The combination of security and functionality, particularly through the contactless interface, has opened the door to the mobile market space, which is here to stay.

The main tasks related to the process of issuing EMV cards are to extract customer information from a bank's database, feed it into a data preparation system (which adds additional data including digital certificates and cryptographic keys) and finally write that data onto the chip. The last step is termed personalization.

## EMV Keys and EMV Certificates

EMV introduces a well-structured security design. The proven security mechanisms of RSA and 3DES are deployed across a six-entity issuing model as indicated in Figure 1 over page.

## Secure Usage – Cryptography in EMV

All EMV cards have a mandated minimum requirement for using one card unique 3DES key and have a choice between three increasingly secure usages of RSA signatures and keys, termed SDA (Static Data Authentication), DDA (Dynamic Data Authentication), and CDA (Combined Data Authentication).

## SDA – Static Data Authentication

The initial and most basic layer of crypto is RSA signatures authenticating the payment card itself when it is used at the ATM and POS terminal.

For SDA, the smart card contains application data which is signed by the private key of the issuer's RSA key pair. When a card with an SDA application is inserted into a terminal, the card sends this signed static application data, the CA index, and the issuer certificate to the terminal. The terminal verifies the issuer certificate and the digital signature by comparing these to the actual application data present on the card. In short, an RSA signature gives the assurance that the data is in fact original and created by the authorised issuer.

## DDA – Dynamic Data Authentication

SDA does not prevent replay attacks as it is the same static data that is presented in every transaction. This is improved with DDA where the smart card has its own card-unique RSA key that signs dynamic data, i.e. unpredictable and transaction-dependent data, and sends this to the terminal. When a card with a DDA application is inserted into a terminal, the card sends the signed dynamic application data, the CA index, the issuer certificate and the card certificate to the terminal. The terminal then verifies the issuer certificate, the smart card certificate and the signed dynamic application data.

## CDA – Combined Dynamic Data Authentication-Application Cryptogram Generation

The SDA and DDA schemes both suffer from protocol weaknesses that may be exploited for criminal purposes. The security mechanism in SDA is there to compare what is on the actual card (PAN, expiry date etc.) with signed data generated at the time of personalization. The digital certificate is a static certificate, i.e. independent of the actual transaction, and hence could be subject to replay attacks.

DDA is stronger and makes use of a card resident unique RSA key to dynamically sign unpredictable and transaction unique data. This, however, is only for the purpose of authenticating the card. The EMV protocol for transaction approval or denial does contain more logical processing, and there is a potential weakness between the steps of verifying the card (using SDA or DDA) and the step comprising of approving the actual transaction. Once the card has been approved a subsequent step is for the card to validate whether the actual transaction shall be denied, approved, or sent online for issuer decision. The card makes that decision based on other card parameters, and it is possible to first go through the SDA/DDA process and then change the message from the card with the verdict on the transaction, although the latter does use card-generated cryptograms. A scheme has been devised that combines both the card authentication and the transaction approval decision in one step. The scheme is termed 'Combined Dynamic Data Authentication-Application Cryptogram Generation' and is abbreviated to CDA. Essentially, it consists of including the card decision among the data being signed by the card's RSA key.

<sup>1</sup> EMV is a trademark of EMVCo

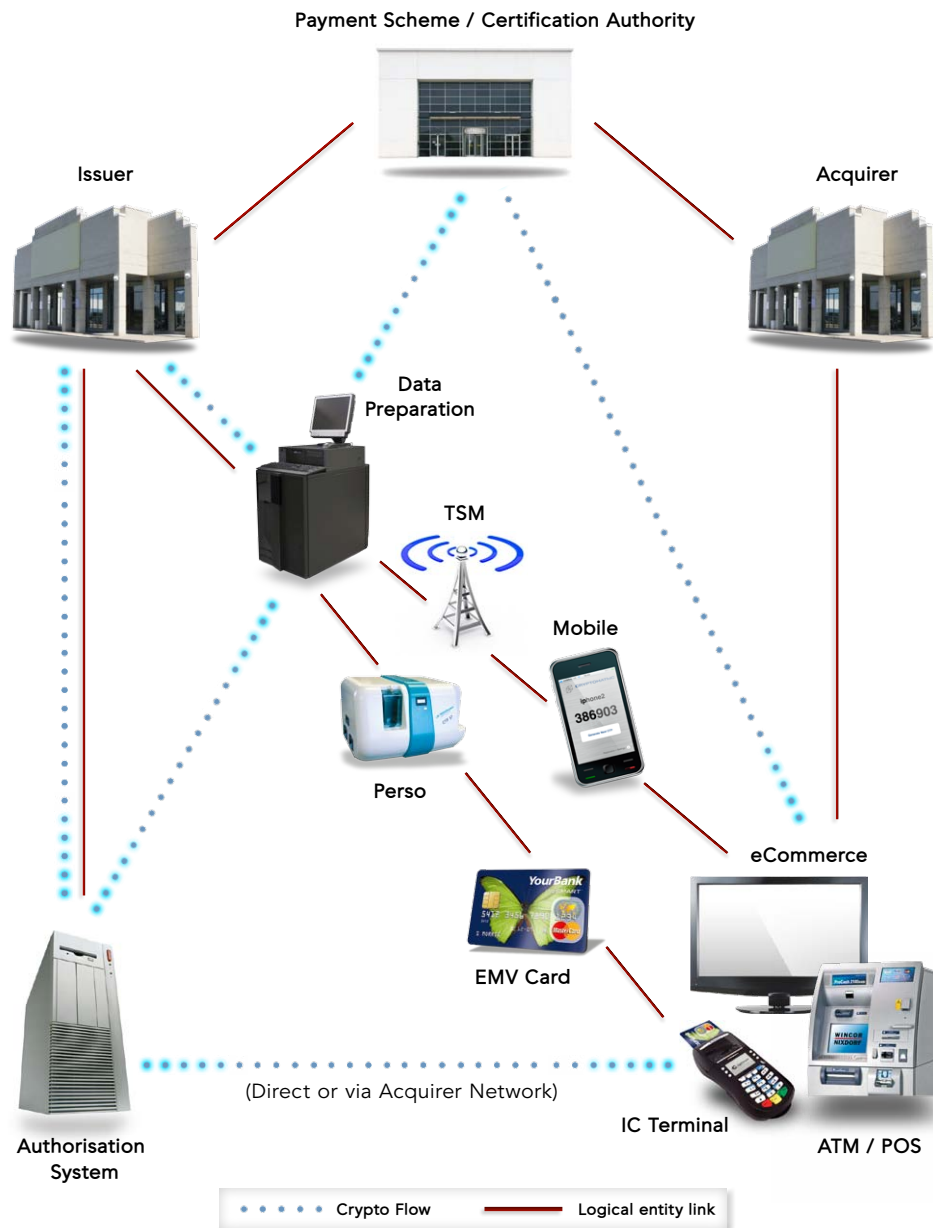


Figure 1: EMV Entities (from a Key Management Viewpoint)

## 3DES Crypto

Whereas SDA, DDA and CDA are based on the RSA public key infrastructure, 3DES is used for evaluating the actual transaction request. The evaluation is done jointly by parameter logic in the card, ATM/POS, and if need be, the issuer through online communication. A 3DES key is used for encrypting the card's part of the decision and to verify a response from the issuer. Further, the issuer may choose to use the opportunity to send additional commands to the card, such as parameter updates,

which necessitates two more 3DES keys to be present in the card for secure command verification. The latter process is termed scripting.

In the issuing scenario the mathematical security perspective of the RSA scheme is the most complex, as it is a multi-layer RSA public key certificate structure. The 3DES scheme is simply for the issuer's data preparation system and authorisation system to share a set of master keys. At



issuing run-time, the data preparation system uses the master key to encrypt each cardholder's account number to create a new key which is put on the individual cards. At processing run-time, the authorisation system encrypts the same account number with the same master key so secure communication is possible.

Contactless cards involve a little more. There will be an extra RSA certificate, for handling/signing contactless data, and an extra master key.

Several entities are involved and are each responsible for their own part. Focussing on security components there are choices as to who generates, manages and distributes cryptographic keys (please see 'Who manages your keys'), but operationally speaking, certain keys and certificates must, at processing run-time, be available at specific entities and operate as specified for that entity.

### Issuer

The issuer supplies card holder data from its card management or host system to a data preparation system. There is a data set for each individual cardholder including the brand of card to produce, account number, card parameters, such as spending limits, plus 'profile' information. A profile defines which cryptographic keys are to be used, settings for PINs, and fixed card type specifics, e.g. risk parameters. In essence an issuer must have the complete overview of all security aspects and cryptographic keys.

### The CA – Certificate Authority

In preparation for issuing EMV cards an issuer must establish a relationship with a payment scheme and exchange cryptographic keys and digital certificates. This occurs via a secure exchange between the bank's

data preparation system (be it in-house or outsourced) and the payment scheme's certificate authority – the CA.

### Issuer and CA Interaction

It is an initial task for the issuer to interact with the CA. Files with digital certificates and files with corresponding hash-values are exchanged; the exchange method varies slightly for each scheme, but in all cases it is a well defined sequence of easy steps. Aside from file extension differences the logical content exchanged is identical.

First the issuer sends a 'certificate-request'; the data preparation system generates an RSA key pair, embeds the public key in a self-signed certificate, and sends it to the scheme CA. The CA evaluates the key and returns a certificate on the key. This is called an issuer certificate. The issuer certificate is a data block containing the issuer's original public key and related data, all signed by a private key belonging to a CA RSA key pair. The CA also sends its own self-signed certificate on its corresponding CA public key, so that the data preparation system can verify the issuer certificate. For security purposes, a separate file is finally sent with a hash of the certificate. It is a simple procedure that is exactly the same for SDA, DDA, CDA and contactless cards (See Figure 2).

With the issuer RSA key pair certified, the data prep system can start generating data for the chip, magnetic stripe and embossing.

### Authorisation System

In preparation for EMV issuing, a bank must establish a technological infrastructure with an authorisation system and exchange cryptographic keys. The keys are all 3DES keys which are used for encrypting transaction data.



Figure 2: Interacting with a CA

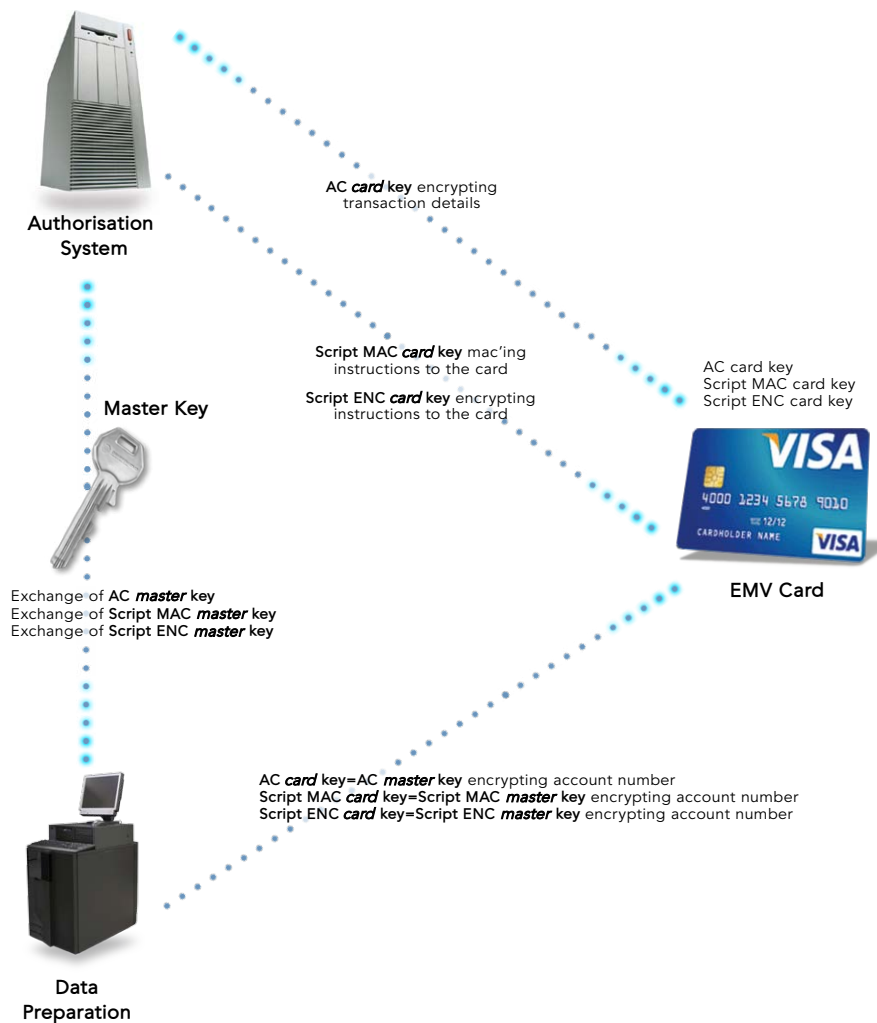


Figure 3: Sharing Keys

At set-up time, a minimum of one key is exchanged between the data preparation system and the authorisation system – the AC key (where AC stands for Application Cryptogram). When a transaction goes online, the card itself encrypts transaction details with the AC key and that cryptogram is sent to the authorisation system.

The data preparation system uses an AC *master* key to create a unique 3DES key for each card; the AC *card* key is derived using the account number. The card uses the AC *card* key to encrypt transaction data, and when the authorisation system receives that encrypted data it can then, at run-time, use the AC *master* key to derive the AC *card* key and so decrypt the data. Messages going back to the card follow the same model.

To have the AC *master* key at both data preparation and the authorisation system means that it must be created in one place and distributed to both systems. It can be made by the data preparation system, authorisation system, issuer's own separate key management system

(if one exists), or by the payment scheme. It is the issuer's decision (See Figure 3).

### PINs

PINs require special management: for online PIN verification the authorisation system may need it; for offline PIN verification the card, and hence the data preparation system may need it; for cards supporting both options, all systems need it. Many non-EMV issuers may already be generating PINs and can re-use them also for EMV. Care should be taken that the PIN formats used at the various points are synchronised; ISO-0, ISO-1, and ISO-2 formats are all in play.

### Data Preparation System

With the issuing model decided, and keys and certificates exchanged, data preparation can begin.

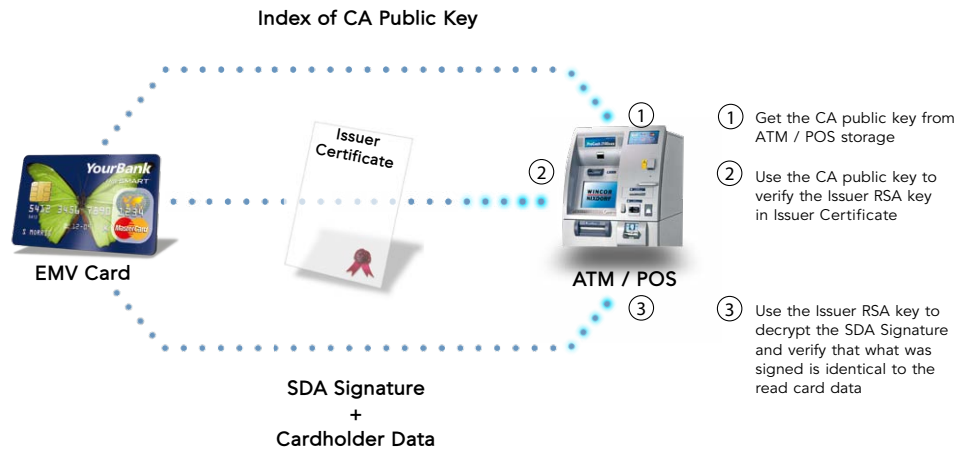


Figure 4: SDA

The task is to generate cryptographic keys, to calculate (derive) cryptographic keys, and to format keys and data to match (1) the format specs of the target chip and (2) the generic machine format used for making the card.

3DES keys are made from other 3DES keys – the master keys. By encrypting an account number with a master key, a card key is created. For simple cards only one transaction key is made. Advanced models require two more for encrypting and MAC'ing online updates to the card. And for contactless 'magstripe' cards, yet another transaction key is made to allow the card itself to later generate CVVs/CVCs/CSCs on the fly.

For SDA cards, the data preparation system uses an issuer's RSA key to generate a digital signature on the card data. It then puts this digital signature and the imported CA-signed issuer certificate onto the card. Each ATM/POS has the actual scheme CA RSA public key

available, and hence can verify that the issuer certificate was properly CA signed, and that the signature was correct. It compares the signed data with the actual data stored on the card to see that it is identical (See Figure 4).

For DDA and CDA cards, the data preparation system also makes an RSA key for each card. The system puts the public part of this RSA key into a card certificate and signs the certificate with the issuer RSA private key. The card certificate is put onto the card together with the issuer certificate that was previously imported from the scheme CA.

Thus, the card's RSA key can be trusted by verifying that the certificate on the card's key is signed by a trusted issuer key, and by verifying that a real scheme CA signed the certificate with the issuer key. Because ATM/POS terminals have the actual scheme CA RSA public key available the verification structure is established.

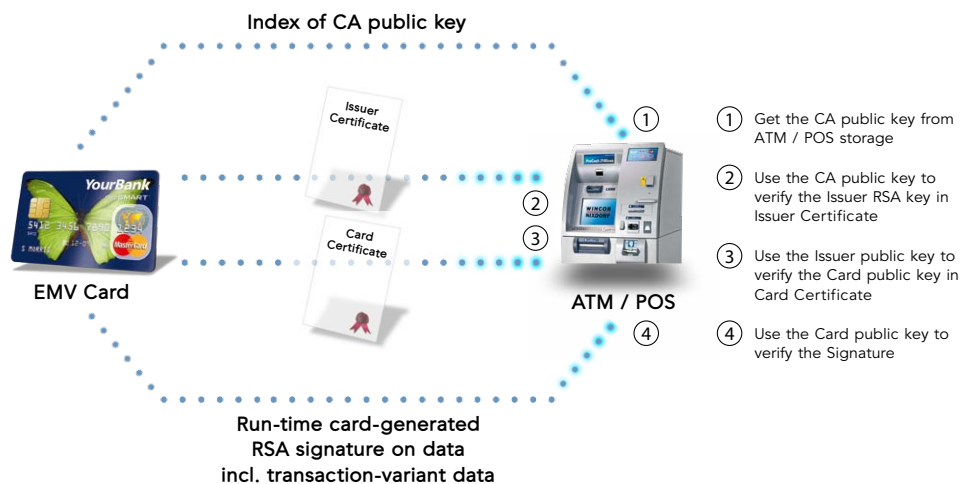


Figure 5: DDA/CDA





With a now trusted card RSA key, the card can use the RSA key for signing transaction details and the ATM/POS can trust the signature. It is near impossible to clone a card with an RSA key and thus security of DDA/CDA transactions is very high. The difference between DDA and CDA consists of the way the card RSA key is used, and relates to which data is signed by the card (CDA includes more data in the signature). For more information please see 'Cryptomathic Cardlnk Technical White Paper' or contact: [technical\\_enquiry@cryptomathic.com](mailto:technical_enquiry@cryptomathic.com) (See Figure 5 on previous page).

## Personalization Equipment

Advanced machinery is used for credit card production. White plastic cards, in the case of centralised issuance<sup>2</sup>, will physically have a chip embedded into it, graphics printed on it, letters embossed, data coded on the magnetic stripe, and finally have data loaded onto the chip. Machines have modules for each part, and various production models exist to enable graphics printing and/or chip initialisation to be carried out at separate locations (See Figure 6).

Machines are run by a software program. The non-chip modules are managed by fixed software to e.g. emboss the account number at a specific location and print a photo. An operator simply needs to set and load bank specific settings for these 'mechanical' parts.

The chip module is more complex. Chips come from different vendors and different vendors do things differently. Next are the applications

from the different payment schemes. Perhaps the card will have one application or maybe multiple. The vendors each have different ways of storing applications, and added to that, the applications are individual per vendor. Much effort is put into standardising applications, but currently each vendor product differs. In practise issuing a card means using completely individual software for the chip part.

## Interacting with Machines

At issuing run-time, interaction with a machine, from the perspective of a data preparation system, consists of one single message. Upon request the data preparation system produces data and secret keys for one card, or a batch of cards, and the result is delivered as a structured data block. The data block can be delivered as a physical file on a disk or as a piece of data kept in computer memory. This data is retrieved by the software that controls the machine and is subsequently used to produce the final card.

Output from the data preparation system is of a confidential nature. There are regular data such as account number and spending limits, but also secret keys and PINs. Payment scheme laws require that these three parts are each encrypted using different cryptographic keys. So the data preparation system encrypts cardholder data under a data transport key [DTK], encrypts keys under a key transport key [KTK] and encrypts PINs under a PIN transport key [PTK] – all 3DES keys. During set-up time these keys are manually shared with the machine software.



Figure 6: A Personalization Machine

<sup>2</sup> The paper largely focuses on the process of key management related to card issuing and does therefore not go into detail around the personalisation process, which could be for centralised issuance, branch issuance or OTA mobile and NFC issuance. For the sake of simplicity and focus on key management the paper gives mention to the process associated with central and instant issuance.



At run-time, the data preparation system encrypts its output with these keys, hands over the data to the machine, which then decrypts the data using the same keys. Immediately after, the machine re-encrypts the data under a card specific storing key. It then stores the data on the chip which finally decrypts the data using the same card specific

card storing key. Variations occur regarding the data encryption key as certain machines cannot handle a dedicated key for the data part, and instead uses whole-file encryption. Keys and PINs are, however, always encrypted using dedicated keys and a Hardware Security Module (HSM) must be deployed for this (See Figure 7).

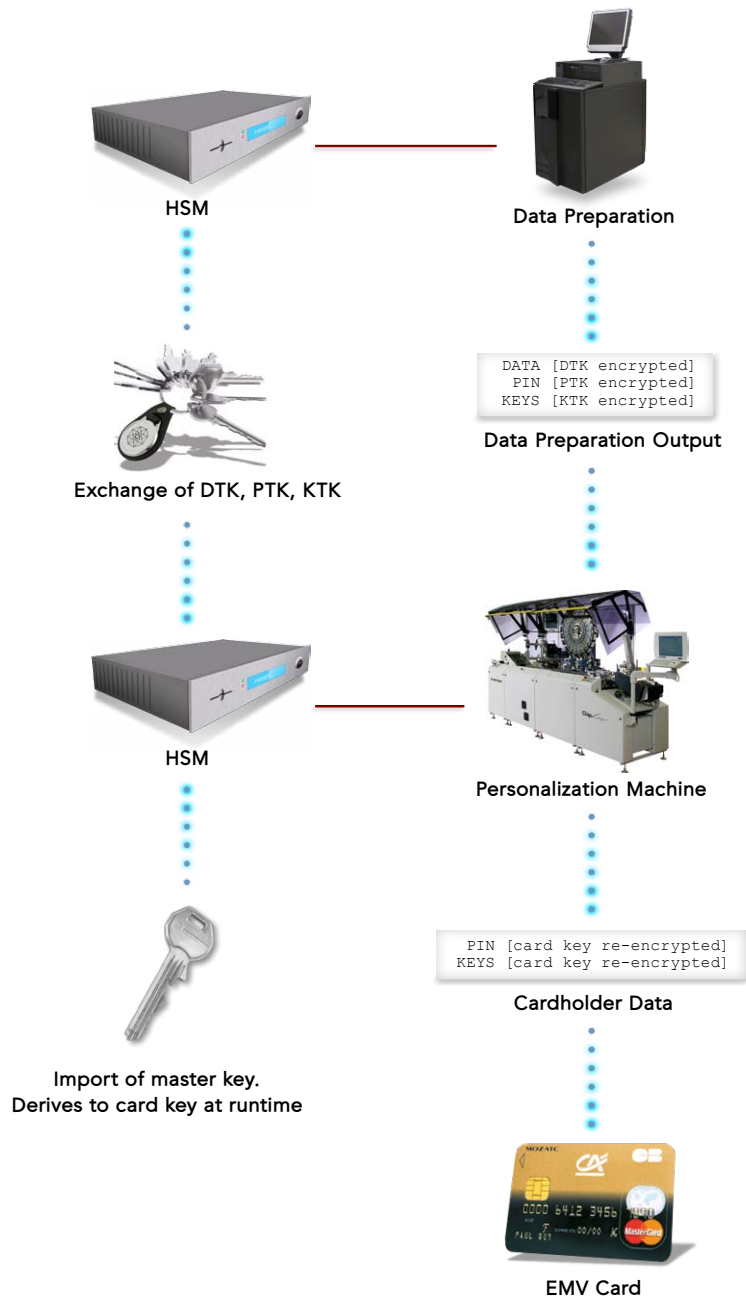


Figure 7: Personalization Cryptography





Figure 8: Hardware Security Modules (HSMs)

## Secure Key Management

As shown, EMV issuing involves key management. Key management involves creating, deleting, storing and distributing keys. For EMV a number of requirements must be met when managing keys, some for physical security and others for procedural aspects, such as the renowned 'dual control'.

The primary security device for key management is a dedicated Hardware Security Module (HSM). An HSM is a small computer encapsulated within a tamper-evident coating. It can either be a stand-alone box or an electronics board inserted into a computer. The rule is that a key must only be in 'clear' form inside an HSM. Outside, it must either be in encrypted form, with the encryption taking place inside the HSM, or be split into several independent components. When a key is to be used for encryption it must be imported into the HSM and used exclusively within the protected environment (See Figure 8).

HSMs can be accessed and managed in two ways, either via a simple command interface or via dedicated software. Some systems are stand-alone key management systems which only manage keys, others are

systems targeted to specific tasks, such as a data preparation key management module designed to handle keys for a data preparation system.

Paradoxically, the simpler a key management system is, the more paperwork is required. The reason for this comes from the payment schemes' strict requirement for usage control. In small black box systems where keys are identified and usage determined only by key naming, this necessitates extra paper management. Dedicated systems will provide GUI controlled usage management and have secured electronic logs.

Irrespective of the software chosen, procedures are an important part of daily key management. The activities surrounding any key activity is called a key ceremony, and signed manual paper forms with all details must be completed. Any aid to this work is welcome, such as a secure paper-printout of key components which can be attached to ceremony forms (See Figure 9 next page).

Please see 'Cryptomathic KMS Technical White Paper' for more information or contact us on: [technical\\_enquiry@cryptomathic.com](mailto:technical_enquiry@cryptomathic.com)



Key Activity: Key Import	Reference ID: 298756
Key Owner: Iowa Agricultural Bank	Date: 23/08/2011 Time: 16:17:05
Key Type: AMEX Issuer AC Master Key	Location: IAB Key Management Room
Encrypted Key <input type="checkbox"/>	System: IAB Data Prep 658465
Zone Encryption Key ID: 23FE99A4	HSM ID: IAB243
Zone Encryption Key KCV: B24D88	System inspected for tampering <input type="checkbox"/>
Key ID: 23FE99A5	KCV verified under dual control <input type="checkbox"/>
Key KCV: A5EE89	
Encrypted Key Value:	11B5 2345 49C3 C4DD 931A 27BD 8CA3 CD82
Key Custodian 1: John Andersson, Empl. ID 198237	
Key Custodian 2: Michael Green, Empl. ID 198289	
Key Manager: Antoine Bree, Empl. ID 198242	

Figure 9: A Key Ceremony Form

On a detailed level, keys are just sequences of bytes. But where do these bytes go when keys are imported or exported? To get into or out of an HSM keys travel through cables and other electronic circuits. The traditional way of complying with the requirements for protecting sensitive material is to disconnect key loading/exporting computers from networks while a key ceremony takes place. When exporting or

importing key components they are allowed to travel from the HSM, via temporary computer memory, to a computer screen, or be typed in via a keyboard through temporary computer memory to the HSM. Certain systems, however, connect the HSM physically to a dedicated hardware terminal for direct communication to the HSM, and hence there is no key material in the computer memory.



Figure 10: Direct Key Import/Export

## Key Distribution

For EMV issuing most keys are shared keys. Key distribution is a manual procedure whereby a key being exported in one of the allowed forms, and under procedural control, is transferred to a remote recipient. The payment scheme regulations for key distribution are very formal and detailed procedures extend all the way down to typing in individual key digits for import into an HSM.

A key concept is key zones. Transport keys must be unique per 'zone' between two entities, so going back to Figure 1, each zone between any two entities sharing key material is a distinct key zone. In practise this means that if a key is generated in one place and is distributed in encrypted form to two recipient locations then two different key transport keys must be used. Each of these transport keys must likewise be exchanged, either as split components or encrypted under a zone master key.

A start-up issuing project normally uses as few keys as possible, but the amount grows and key distribution becomes resource demanding – an example being transfer of all keys to a disaster recovery site.

## Who Manages The Keys?

It is clear that there are several processes related to key management. In some cases all processes are managed by third party service providers, e.g. card payment processors. In other cases some of the systems are managed in-house while others are outsourced, e.g. personalization. Finally, there are cases where everything is managed in-house, which is most common for large financial institutions. How an issuer chooses to manage the process, whether through internal or external processes, or through a combination of both is really down to preference and cost – however, it is important to note that the issuer is responsible for ownership and management of the keys, and that a suitable key management strategy is constructed. There might be security policies in place to ensure that sensitive keys are only managed

in-house, or there might be a long history of successful outsourcing due to limited internal knowledge about the card issuing and acquiring procedure, or perhaps the procedure is centralised through a banking organisation, and so on.

The big dilemma is to find the most productive and cost-efficient way to manage systems.

Experience shows that banks, which manage in-house data preparation systems, have the ability to shop around for the best personalization service, producing savings of up to several million Dollars/Euros/Pounds per year for large issuers. So it may make sense to do, at least, the data preparation in-house, but not necessarily.

## Contact Cryptomathic

If you have found this document useful but have questions then feel free to contact us on: [technical\\_enquiry@cryptomathic.com](mailto:technical_enquiry@cryptomathic.com)

## EMV Key Management Dictionary

### 3DES

Triple DES. Symmetric cipher algorithm using a 128-bit Data Encryption Standard key.

### AC – Application Cryptogram

Cryptogram calculated by the card as part of transaction evaluation.

### CA

Payment scheme certificate authority. Owns root certificates which authenticates issuers' RSA keys in the payment infrastructure.

**CA Certificate**

Payment scheme root certificate which authenticates issuers' RSA keys in the payment infrastructure.

**CDA**

Combined Data Authentication. Verification of card-created RSA signature on changing transaction data combined with the card's transaction acceptance verdict.

**Common Personalization**

Industry standard for EMV card personalization. Includes standard format for data preparation.

**CSC / CVC / CVV**

Card Security Code / Card Verification Code / Card Verification Value.

**Data Preparation**

The process of preparing the data to be loaded (personalized) onto a payment chip or magnetic stripe. Includes generating cryptographic keys / certificates and formatting the data to the intended target i.e. personalization machine or over-the-air personalization software for mobile phones.

**DDA**

Dynamic Data Authentication. Verification of card-created RSA signature on changing transaction data.

**EMV**

Trademark owned by American Express, JCB, MasterCard and Visa as EMVCo. EMVCo defines the global chip-based payment infrastructure.

**HSM**

Hardware Security Module. Tamper resistant security electronics for performing cryptographic calculations. Must be FIPS 104-2 Level 3 or 4.

**ICC**

Integrated Circuit Card. A white plastic card with an electronic chip.

**ICC Certificate**

Digital certificate on a card's RSA public key. Signed during data preparation by an issuer's RSA private key.

**Issuer Certificate**

Digital certificate on an issuer's RSA public key (IPK). Signed by a payment scheme CA.

**Issuer Master Key**

3DES key which, during data preparation, is derived with the cardholder's account number to a card-unique 3DES key.

**EMV Chip Card**

A chip card which is either based on a proprietary (native) OS or executed via the Java language (Java Card), and associated with Common Personalization or Multos (high secure payment chip OS).

**Key Zone**

Logical zone between two entities who exchange cryptographic keys.

**MAC**

Message Authentication Code.

**Master Key**

Ultimate key in key hierarchy from which other keys are derived or protected.

**Personalization**

The process of storing cardholder 'personal' data on a blank chip or magnetic stripe.

**PIN**

Personal Identification Number. A code verifying the cardholder's authenticity.

**PKI**

Public Key Infrastructure.

**RSA**

Rivest, Shamir, Adleman. Asymmetric encipherment algorithm using a key pair with a private key and a public key.

**Script**

Here a command sent to a chip to update a parameter in the chip. E.g. a PIN-unblock command.

**SDA**

Static Data Authentication. Verification of RSA signature on non-changing card data.

**Transport Key**

Also known as a Key Encryption Key – used for encrypting keys.

**XOR**

Binary addition without carry.  $0+0=0$  |  $0+1=1$  |  $1+1=0$ .

**Useful Links**

[www.emvco.com](http://www.emvco.com)

[www.globalplatform.org](http://www.globalplatform.org)

[www.multos.com](http://www.multos.com)

[www.smartcardalliance.org/pages/activities-emv-migration-forum](http://www.smartcardalliance.org/pages/activities-emv-migration-forum)

[www.cryptomathic.com/products/emv](http://www.cryptomathic.com/products/emv)



## About Cardlnk

Cardlnk is an EMV data preparation system, which offers best-of-breed centralised data formatting and key management while maintaining flexibility to meet any card issuing environment (e.g. mag stripe, chip, single- and multi-applications, instant issuing, NFC). Cardlnk is implemented by bureaux, data processors and card issuers alike. Cardlnk is very scalable, making it perfect for high volume production but also suitable for small scale production.

Cardlnk is used by more than 100 customers across the globe to issue hundreds of millions of EMV cards annually.

Cardlnk integrates with various issuing systems from major vendors such as ACI Worldwide, Atlantic Zeiser, Datacard and Mühlbauer, and it supports in excess of 10 international and national payment schemes, including AmEx, Discover, MasterCard and Visa.

Cardlnk features a format converter, allowing simple integration for both input files (raw files that Cardlnk processes) and output files (completed files from Cardlnk sent to personalization and/or card management).

Cardlnk is the only major system that is both HSM vendor and card platform independent – this ensures that customers are not tied in to one particular technology, and hence ensures a high return on investment.

<http://www.cryptomathic.com/products/emv/cardlnk>

## About EMV Certification Authority

The Cryptomathic EMV CA is an essential service component for EMV card authentication. The main purpose of the EMV CA is to allow a central authority to issue and manage the certificates of Card Issuers within a given region.

EMV CA supports several CAs, which may each have a number of self-signed CA certificates. A CA can be set up to be compliant with either MasterCard or VISA. The compliance determines which formats are used for data exchange with other systems.

EMV CA manages all certificate related tasks including: issuing, exporting and revoking keys, lifecycle management of EMV Issuer CA certificates, and certification authority CRL (Certificate Revocation List).

EMV CA is designed in a flexible client-server structure enabling the payment scheme provider to tailor the system to the specific needs of its organisation in-line with e.g. regional and national requirements.

<http://www.cryptomathic.com/products/emv/emv-ca>

## Contact us:

[technical\\_enquiry@cryptomathic.com](mailto:technical_enquiry@cryptomathic.com)

[enquiry@cryptomathic.com](mailto:enquiry@cryptomathic.com)

## Disclaimer

© 2013, Cryptomathic A/S. All rights reserved

Jægergårdsgade 118, DK-8000 Aarhus C, Denmark

This document is protected by copyright. No part of the document may be reproduced in any form by any means without prior written authorisation of Cryptomathic.

Information described in this document may be protected by a pending patent application.

This document is provided "as is" without warranty of any kind.

Cryptomathic may make improvements and/or changes in the product described in this document at any time. The document is not part of the documentation for a specific version or release of the product, but will be updated periodically.

[www.cryptomathic.com](http://www.cryptomathic.com)

## ABOUT CRYPTOMATHIC

Cryptomathic is one of the world's leading providers of security solutions to businesses across a wide range of industry sectors, including finance, smart card, digital rights management and government. With more than 25 years' experience, Cryptomathic provides customers with systems for e-banking, PKI initiatives, card personalization, ePassport, card issuing and advanced key management utilizing best-of-breed

security software and services. Cryptomathic prides itself on its strong technical expertise and unique market knowledge. Together with an established network of partners, Cryptomathic assists companies around the world with building security from requirement specification to implementation and delivery.