Testing EMV Payment Applications

CPT 3000v3 is designed for maximum flexibility and upgradability, to this end the card test sequences do not form part of the program but are controlled by separate Test Scripts. Because of this design approach the main manual and on line help documents are necessarily general in nature and do not cover specific features of individual card tests. This document/help file covers the test scripts and test scenario definitions for the EMV payment application tests.
The EMV terminal simulator script simulates a transaction in a terminal in order to interrogate the card and extract the personalised data from it. The extracted data can then be checked against the relevant specifications and the individual issuer requirements set up in the data analysis scenario.
The terminal simulation script can be set up to simulate a wide variety of transactions in a wide variety of terminal types, so that as well as verifying the card's personalisation data its functionality can be checked in a variety of transaction scenarios. It can also be programmed to do things a regular point of sale terminal would never do.
The data analysis script can call up verification scripts to check the card's data against EMV 4.3, or against any of the major Payment Scheme specifications. These scripts will change from time to time as Barnes issues updates to cover changes in the specifications and scheme operating regulations. In order to assimilate a new test script it is only necessary to copy the new file into the scripts folder, replacing the earlier file of the same name.
Note that all Tcl scripts are protected by secure checksums, this ensures that only scripts issued by Barnes or by a *bona fide* CAT3000v3 developer, and not damaged or tampered with, will run the card tests.

Understanding the EMV payment transaction.

This section is an attempt to describe the process involved in using a chip card to perform a financial transaction. It is not intended to provide an in depth knowledge of the technical details, for that the reader should consult the official specifications, nor does it attempt to give more than a cursory idea of the mathematics of the cryptography involved. It merely attempts to give the reader who is either non-technical or new to EMV cards an overall understanding of the process, to de-mystify some of the jargon and induce a basic understanding of the personalisation data in the card.
Given this basic overview, those who wish to delve deeper should be better prepared to understand the specifications, and more to the point, to understand the test results produced by the CPT 3000v3.

Overview

There are three computers involved in an EMV transaction, the card, the terminal and the host. Each one performs the function of representing its owner, namely the cardholder, the merchant and the issuer, respectively.
The card's computer (which we shall refer to simply as the card) is a single chip microcomputer embedded in the rectangle of plastic carried in the cardholder's wallet or purse, it is the smallest and least powerful of the three, and although the processing power available within the card is constantly increasing, the design of the transaction procedure was to a large extent conditioned by this lack of power in the earlier cards. The terms "smart card", "chip card" and "integrated circuit card (ICC)" are all alternative names for the same device, in this document we shall simply use the term "card".
The terminal will generally be a cash register or an ATM (but may be other types of machine, such as a mobile 'phone) and the host is a huge mainframe sitting in the issuer's data processing centre. Usually the card and the terminal are close coupled, i.e. directly connected to one another, and the host is always at the other end of a telephone line or other long distance communications link, such as the internet.
A terminal can be described as "online", or "offline", in the former case this means that the terminal and the host can communicate directly during the transaction, in the latter case they cannot. ATMs, for instance, are normally online, whereas the cash register in a small grocery shop is more likely to be offline. Offline terminals will send their transaction records to the host at some later time, the method of doing this will vary according to local circumstances.
Always keep in mind the underlying purpose, which is to transfer money from the cardholder's account to the merchant's account in an efficient and secure manner. There are a number of important considerations, they may seem obvious but are worth emphasising because they condition the design of the communications between the three computers and their decision making processes.

1. Authentication. All parties to the transaction must ensure that the other parties are who they say they are (e.g. The card is not forged, stolen or tampered with, the computer at the other end of the telephone really is the issuer's host).
2. Authorisation. All parties must ensure that the other parties are allowed to take part in this transaction. (e.g. The card is allowed to be used at this terminal and the cardholder's account has sufficient funds or credit for the transaction)
3. Efficiency. The whole process must take place within a very limited time frame (Long checkout queues are a waste of everyone's time and money).
NB. The EMV specifications do not cover contactless cards, so contactless payment applications are administered by the payment scheme operators. Although they are all based on EMV to a greater or lesser extent the process is modified to reduce the time the card and the terminal spend communicating. There is more variation from scheme to scheme in contactless than there is in contact cards.

Powering Up

When the card is inserted into the coupler the terminal applies power to the card and receives from it an ATR (Answer To Reset). The ATR describes to the terminal the card's communications capabilities, and allows the terminal to configure itself to talk to the card.
Communications between the terminal and the card are "half-duplex", that is, the same wire is used for messages in both directions. Obviously it is therefore important that the two units do not both transmit at the same time, so there is a strict protocol laid down to prevent this. In essence the terminal controls the wire, so all communication exchanges take the form of a command from the terminal followed by a response from the card. There are no exceptions to this.
In the CPT 3000v3, in common with most personalisation testers, this low level protocol is taken care of entirely within the card reader attached to the system. Barnes does not manufacture these readers, but sources them from commercial suppliers. (Barnes does manufacture a low level protocol test tool, the CET 3000, but this is a completely different type of machine).

Application Selection

Being a computer, the card can hold a number of applications. Just as a desktop computer can have a word processor, a spreadsheet, and an adventure game, all of which are software applications, so the card can hold a debit application, a credit application, a loyalty application, and so on. The first task for the card and the terminal is to agree on the correct card application for the transaction being conducted. Most early cards only had one application in them, but the trend in newer cards is to have two or more.
The first command the terminal sends to the card is a "Select" command. The Select command asks the card to activate an application, and of course must carry information as to what application the terminal wants to activate. The first Select command will request the activation of the PSE (Payment Systems Environment). The PSE is not a payment application itself, but a directory of the payment applications on the card, it performs the same function as the start menu on a PC, telling the user what software is available and giving him a quick means to access it.
The card will respond to any Select command either with details of the application that has just been activated, or with an error code that indicates the application is not available. The PSE is not mandatory for an EMV card, although most current cards have one, it does speed up the process of starting the payment application.
Following a successful PSE selection the terminal will issue a number of "Read Record" commands, to which the card will respond with the data held in each of the PSE's records. Each record will contain one or more entries describing an available payment application, as a minimum each of these entries will contain a "Label" which can be shown on the terminal's display, and an AID (Application IDentifier), which is a code the terminal sends to the card in another Select command to activate that particular application.
NB. Contactless cards have a slightly different protocol for their PPSE (Proximity Payment Systems Environment), to speed up the process all the information is returned in response to the Select command, no Read Record commands are needed.
When there are several applications present each one will normally also have a priority indicator which assists the terminal in choosing the best application for the job. The terminal may make the choice autonomously, depending on the applications it itself can support, or it may present the cardholder with a list and ask him or her to make the choice. Either way, once the decision is made the terminal knows the AID of the payment application to be used, and can issue a Select command to activate it.

If there is no PSE on the card the terminal has no choice but to try a number of AIDs until the card responds positively. The terminal will contain a list of AIDs for the applications it supports and will try each in turn to see which of them opens. Obviously if there is a long list this could take some time.
The AID is a hexadecimal number made up of two parts, the first five bytes are known as the RID (Registered application provider IDentifier), and identifies the payment scheme, the remainder is the PIX (Proprietary application Identifier eXtension) and identifies the account type. For instance the RID for Visa is A000000003. Generally schemes use the PIX 1010 for their standard or flagship account types, so A0000000031010 will be a standard VSDC application, whereas A0000000032010 indicates an Electron card. The RID values for the other major payment schemes are A000000004 for MasterCard, A000000065 for JCB and A000000025 for American Express.
A successful response to the Select command will contain information about the application, including a repeat of some of the data that was in the PSE, but also additional information which will enable the terminal to continue the process.

Get Processing Options.

The first command of the transaction proper is "Get Processing Options", this formally begins the transaction and allows both the card and the terminal to configure themselves for what is to follow.
Amongst the information returned by the card in the response to the Select command was the PDOL (Processing options Data Object List). Data object lists are the card's way of requesting data from the terminal, and there are several different ones that may or may not be used, all of them contain the letters "DOL" in their acronym.
The PDOL is not mandatory, it is only necessary if the card is capable of configuring itself to behave differently depending on the local conditions. Most current PDOLs contain a request for the Terminal Country Code, and could request other data such as local currency, terminal type etc. The terminal supplies this information with the Get Processing Options command and the card then returns two essential items of information, the Application Interchange Profile (AIP) and the Application File Locater (AFL).
NB. A major exception to this is Visa's qVSDC application ("q" for "quick"). qVSDC is used on contactless cards only and differs from an EMV application in that a great deal more information is requested by the PDOL, so that usually the transaction can be completed in the Get Processing Options command with no further processing.
The AIP is a two byte value, arranged as sixteen single bit "flags". Each bit represents a function that the card may or may not support, the bit is set to one if the function is supported. It mainly covers such things as authentication and authorisation methods. At the time of writing not all of the bits are allocated, and some of those that are allocated only apply to contactless cards (their meanings varying from scheme to scheme), the others are available for future expansion.
The AFL is a list of file and record references which the terminal then uses to issue a series of "Read Record" commands to the card in order to extract the data required to perform the transaction. The data contained in these records includes obvious things like the account number and the expiry date, but also more complex information which allows the terminal to verify that the card and the cardholder are both genuine and informs it how the issuer requires the terminal to react under different circumstances.

TLV Encoding and Data Object Lists

It is worth turning aside at this point to describe the way data is encoded for storage within the card and for transfer between the card and the terminal. The technique is known as TLV encoding, where TLV stands for Tag, Length and Value. This method is a good compromise between simplicity of processing and efficient use of limited memory, and so is ideal for use in small low powered computers such as the one in the card.
The tag is the identifier of the item of data, and is always the first one or two bytes of the data stream (the encoding of the first byte defines whether or not another byte follows). EMV assigns tags to the data objects used in the transaction, most are stored in the card and transmitted to the terminal, some are stored in the terminal and supplied to the card, others are generated dynamically during the transaction, and a few are stored in the card and never transmitted to anyone. EMV also assigns a range of tags for proprietary use by individual payment schemes.
People refer to data objects in different ways depending on their area of interest and level of expertise, all objects have a tag and a descriptive name, many also have an abbreviated acronym. e.g. Tag 5A identifies the Application Primary Account Number, or PAN.
Immediately following the tag is the length, again this can occupy one or more bytes depending on the encoding of the first byte, and defines the number of bytes taken up by the value.

The remaining bytes, the value, are the actual data. The interpretation of the data depends on the data object itself, i.e. on the tag, and in some instances includes a number of other, embedded TLV encoded objects.
There are a number of variations on the TLV theme, the one used in EMV transactions (BER-TLV, or Basic Encoding Rules TLV) is fully described in Annex B of EMV 4.2 Book 3.
When the card passes a DOL (Data Object List, e.g. the PDOL) to the terminal, the DOL has it's own tag, e.g. the PDOL's tag is 9F38, and its value field consists of a list of tags and lengths without their values. The terminal then supplies the values associated with these tags (using the lengths and order indicated by the DOL) as the data field of a subsequent command.
The example below is taken from a CPT 3000v3 engineer's log file, it shows a successful selection of a VSDC application, with a PDOL requesting the Terminal Country Code (tag 9F1A, length 2). The terminal then passes the value (0826 representing the UK) back to the card in the Get Processing Options Command.

```
Selecting Application A0000000031010
First Selection
<snd> (013) 00 A4 04 00 07 A0 00 00 00 03 10 10 00
<rcv> (059) 6F 37 84 07 A0 00 00 00 03 10 10 A5 2C 50 0B 56
49 53 41 20 43 52 45 44 49 54 9F 38 03 9F 1A 02
9F 12 0F 43 52 45 44 49 54 4F 20 44 45 20 56 49
53 41 87 01 01 9F 11 01 01 90 00
*Tag 84 - DF Name - A0 00 00 00 03 10 10
*Tag A5 - FCI Proprietary Template - Length 44
*Tag 50 - Application Label - VISA CREDIT
*Tag 9F38 - PDOL - 9F 1A 02
*Tag 9F12 - Application Preferred Name - CREDITO DE VISA
*Tag 87 - Application Priority Indicator - 01
*Tag 9F11 - Issuer Code Table Index - 01
Building PDOL 0826
Get Processing Options
<snd> (010) 80 A8 00 00 04 83 02 08 26 00
<rcv> (018) 80 0E 5C 00 08 01 01 00 10 01 03 00 18 01 02 01
90 00
*Tag 82 - Application Interchange Profile - 5C 00
*Tag 94 - Application File Locater - 08 01 01 00 10 01 03 00 18 01 02 01
*File Locater 1 - 1 1 1 0
*File Locater 2 - 2 1 3 0
*File Locater 3 - 3 1 2 1
```

Asymmetric Key Cryptography and Data Authentication

Once all the data has been extracted the terminal performs the offline data authentication process, this process ensures that the card is a genuine card issued by the financial institution it claims to represent, and that it has not been tampered with.
Offline data authentication uses Asymmetric Key Cryptography, sometimes known as public/private key cryptography, or PKI. Asymmetric key means that there are two keys, one of which is used for encryption and the other for decryption, the keys are generated in such a way that it is impossible to calculate one from the other. Then one key is kept secret and the other is made public.
For data authentication in an EMV transaction, the encryption key is kept secret and the decryption key is made public. The mathematical algorithm used is RSA (after Rivest, Shamir and Adelman, the three mathematicians who designed it). The major disadvantage of RSA is that it requires significant computing power to perform the calculations, although as processors increase in power this is becoming a much less important issue. Elliptic Curve Cryptography is another asymmetric key algorithm, which is gaining in popularity because of its lower demands on the computer, but it is not currently used by EMV.
Asymmetric key cryptography is used for offline authentication because the terminal is owned or managed by a different organisation from the card and the host, so the use of a public key means there are no security issues to worry about. online authentication takes places between the card and the host, which are both managed by the same organisation, and so it is possible to use symmetric key cryptography, where the same, secret, key is used for encryption and decryption.
online authentication is described in more detail below.

Static Data Authentication

SDA is the simplest form of offline authentication, it requires no RSA processing within the card.

One of the data objects recovered from the card during the "Read Record" phase is the Issuer Public Key Certificate (tag 90), which is a large block of data stored pre-encrypted on the card. Associated with this is another item called the Certification Authority Public Key Index (tag 8F), the "Certification Authority" is generally the Payment Scheme, i.e. Visa, MasterCard etc. The data in the certificate is supplied by the issuer and encrypted on their behalf by the payment scheme authority, the issuer does not have access to the private key used. The corresponding public keys are held in the terminal, each one being uniquely identified by the application's RID and the CA Public Key Index. Successful decryption of the Issuer Public Key Certificate using the payment scheme's published public key verifies that the data was truly encrypted by that authority using their private key.
NB. An RSA key is in two parts, known as the exponent and the modulus, in fact there are two moduli, one public and the other private, the same modulus is used with both. Making the calculation with public exponent exactly reverses the effect of the private exponent, and vice versa, so either can be used for encryption. The public exponent is a one or three byte value, the modulus and the private exponent both have a much greater length. The RSA algorithm always operates on a data block which is the same length as the modulus and the private exponent. The term "public key" is used to describe the modulus and the public exponent, the term "private key" refers to the private exponent.
Successful decryption of the Issuer Public Key Certificate yields the Issuer Public Key, no surprises there then. The private key associated with the Issuer Public Key is the property of the card issuer. Usually the Issuer Public Key is too large to be entirely accommodated within the certificate, along with the other data that is held there, so only the first part of the key is in the certificate and the rest of it is held in clear under tag 92, the Issuer Public Key Remainder.
Also recovered from the Issuer Public Key Certificate is a twenty byte hash (a hash is a sophisticated form of checksum) of the data within the certificate and the remainder, so the terminal can verify that no data has been corrupted or tampered with.
Armed with the Issuer Public Key the terminal can now decrypt the Signed Static Application Data (tag 93). This yields a hash of the card's sensitive data, i.e. data that is recovered from the card in clear but needs to be protected against corruption and tampering. The data to be hashed is identified by coding in the AFL. If this hash verifies correctly then SDA has succeeded, i.e. all the data is correct and genuine.
NB. All the encryption processes involved in SDA are carried out before the card is personalised and the card simply stores the encypted data. This means that is it possible for a fraudster to "skim" the card data. So, although online transactions are protected by the symmetric key cryptography described later, offline transactions are vulnerable.

Dynamic Data Authentication

DDA is a little more complex, but provides better security. DDA requires the card to perform some RSA calculations.
DDA begins in the same way as SDA, with the decryption of the Issuer Public Key Certificate to obtain the Issuer Public Key. In this case, however, the Issuer Public Key is used to decrypt the ICC Public Key Certificate, which is similar in format to the Issuer Public Key Certificate, but is encrypted using the issuer's private key, and yields the ICC Public Key. ICC is "Integrated Circuit Card", and the ICC public key is unique to the card, with the corresponding private key held within the card. The issuer has the option to include other card data within the hash used to verify the ICC public key, so a *de facto* SDA check is carried out during key retrieval.
The terminal then transmits to the card an "Internal Authenticate" command, which carries with it data defined by the DDOL (DDA DOL) supplied by the card during the "Read Record" phase. If the card does not supply a DDOL then EMV defines default DDOL which must be used. Among the data is always an "Unpredictable Number", generated by the terminal at transaction time. The unpredictable number may be random or sequential, so long as it is different for every transaction (i.e. the card must not be able to predict it, even if the terminal can). The card then generates a twenty byte hash of the data supplied by the terminal concatenated with another unpredictable number generated by the card. The hash and the card's unpredictable number are then incorporated into the Signed Dynamic Application Data, which is encrypted using the card's private key and returned to the terminal in the response to the Internal Authenticate command.
The terminal decrypts the Signed Dynamic Application Data using the ICC Public Key, recovers the ICC Dynamic Data and verifies the hash against the data that was transmitted. If the hash checks out then DDA has succeeded, i.e. the card not only contains correct and genuine data, but also the private key corresponding to the ICC public key encrypted by the issuer. The card's private key will never be transmitted to any terminal, it will only be used internally by the card, and its corresponding public key is encrypted using the issuer's private key which has never been anywhere near the card, this arrangement makes life very difficult for the potential fraudster.

Combined Data Authentication

There is a third method of data authentication, CDA, or to give it its full title "Combined Dynamic data Authentication and generate application cryptogram". The authentication process is the same as DDA but the data is exchanged during the "Generate Application Cryptogram" command, covered below. CDA provides further security by including transaction data within the signed data.

Cardholder Verification

Having established that the card is genuine, the terminal must now ensure that the person presenting the card is the genuine cardholder.

The terminal examines the contents of tag 8E, the CVM list (Cardholder Verification Methods). The CVM list contains two monetary amounts, called X and Y, and a prioritised list of verification methods. The two amounts may optionally be used to condition the choice of verification method, but this is rare. Each verification method is encoded in two parts, the method itself, and the conditions of use. The terminal works its way down the list applying the method if the conditions permit and taking the indicated action if the verification fails. As soon as one method succeeds then the cardholder is verified.

As an example, the list might define "Offline Encrypted PIN" and "If Terminal Supports" as its first CVM, so a terminal which is not fitted with a PIN pad would skip this one and adopt the next, which might be "Paper Signature", and so on.

Offline PINs can be submitted to the card either in clear (plain text) or encrypted. Plain text PIN can only be used when the connection between the PIN pad and the card is secure, i.e. entirely within a tamper-proof enclosure, if there is any danger of the message from the PIN pad to the card being intercepted then encrypted PIN must be used.

With contactless cards communicating with the terminal by radio (albeit very short range radio) plaintext offline PIN is obviously out of the question, so if a contactless card supports offline PIN at all, it must be encrypted.

When an encrypted offline PIN is submitted to the card the terminal encrypts the PIN using a public key supplied by the card. If the card supports DDA or CDA then the ICC public key obtained there is often used to encrypt the PIN, alternatively a specific PIN encipherment public key may be used, this is obtained using the Issuer Public Key in exactly the same way as the ICC Public Key for DDA or CDA was obtained.

Decision Time

With all of the above completed it is now time to make the decision whether or not the transaction will be allowed to go ahead. The terminal and the card "vote" on what to do, if either of them votes "no", then the transaction will not proceed, they must both vote "yes" before any money changes hands. If the terminal is online then the issuer's host may also be involved in the decision making process.

The terminal fills in a set of binary flags called the TVR, or Terminal Verification Results, the TVR is a five byte value with each bit representing a different status. Not all of the bits are currently allocated. TVR bits represent such things as "New card", "Expired Application", "PIN Not Entered" and so on. These bits are then compared against various sets of "action codes" to determine the next step.

Action codes are five byte data objects with exactly the same bit allocations as the TVR, each one specifies an action that must be taken if a "1" bit in the action code matches a "1" bit in the TVR.

The terminal reads from the card a set of Issuer Action Codes, named Default, Denial and Online, these represent the card issuer's preference as to what action should be taken under each circumstance. Optionally the terminal may carry a matching set of Terminal Action Codes, which represent the merchant's preferences.

When there is a matching "1" bit in the TVR and either of the Action Codes then the action implied by the name of that code must be taken. E.g. if the bit allocated to "PIN Not Entered" is set to "1" in both the TVR and the IAC - Online, then the terminal must refer the transaction to the issuer by going online to the host.

The action codes are processed in the order, denial, online, then default, such that conditions that generate a denial take the highest priority, followed by online. The default codes are used to generate a denial if the terminal is unable to go online, or if for any reason the communications with the host fail.

If the transaction passes all these tests then the terminal may apply its own action codes to see if the merchant's requirements dictate a denial or an online referral.

Based on these codes the terminal chooses one of three courses of action, deny the transaction outright, refer the transaction online to the issuer, or accept the transaction offline. This decision is passed to the card as part of the first "Generate Application Cryptogram" command.

Application Cryptogram Generation.

Among the data extracted from the card during the Read Record phase will be two DOLs known as CDOL1 and CDOL2, (Card risk management Data Object List), which define the data the card

requires from the terminal when the "Generate Application Cryptogram" commands are issued. A maximum of two Generate AC commands may be issued to the card, the data accompanying the first is defined by CDOL1, and the data for the second, where required, by CDOL2. Data defined by the CDOLs always includes information about the transaction itself, amount spent, time and date etc. and the Generate AC command itself also carries the terminal's completion decision.
The three decision codes have acronyms that are not obvious. A denial is indicated by AAC (Application Authentication Cryptogram), an online request by ARQC (Authorisation ReQuest Cryptogram) and an offline acceptance by TC (Transaction Certificate).
On receipt of the Generate AC command, the card performs its own internal risk assessment processing, compares the result of this with the request from the terminal, and responds with a decision. The card's decision will never be more optimistic than the terminal's, i.e. if the terminal requests a denial then the card must always agree, but if the terminal requests offline acceptance the card may choose to go online or even to deny the transaction.
As well as this decision the card will transmit to the terminal a cryptogram. The cryptogram is a secure hash of the transaction data, and is generated using DES encryption so that only the host can verify it, the validation of the cryptogram assures the host that the card is genuine. The terminal sends the cryptogram to the host along with all the other transaction information either immediately (in the case of an online referral) or later as part of the day's transaction data.
In the case of an online referral, the host will reply with another cryptogram which the terminal passes to the card using an "External Authenticate" command. The card then verifies that this cryptogram really did come from its issuer and responds positively or negatively.
If the External Authenticate was successful then the terminal issues a second Generate AC command, the data for which includes the host's decision, which is final. The second cryptogram from the card is later sent to the host as confirmation of what actually occurred following the referral.
Recent trends are to abandon the use of the External Authenticate command and instead include the host's cryptogram in the data defined by CDOL2, so the authentication is performed during the second Generate AC. This achieves the same objective with less communications traffic.
If the card uses CDA, then the data required for DDA will be passed to the card with the first Generate AC command and the response will include the Signed Dynamic Application Data, which will need to be verified before the card's decision is accepted and an online cryptogram is accepted for onward transmission to the host.

Symmetric Key Cryptography

The cryptogram is generated using the DES (Data Encryption Standard) algorithm. DES was originally developed for the US military for use as a high speed machine cipher. For some years now it has been in the public domain and is now widely used in non-military applications. DES is a symmetric key algorithm, that is, the same key is used for encryption and decryption, and so must never be made public. It is also economical in its use of processing power and so is convenient for use in small, low powered, computers such as smart cards. Moves are afoot to replace it with the AES algorithm, which uses longer keys and is more secure.
The basic DES algorithm takes an eight byte clear data block and an eight byte key and combines them to form an eight byte encrypted data block. Decryption is the exact reverse and uses the same key. Longer messages are encrypted by breaking the data up into eight byte blocks and adding padding bytes if necessary to make the last block up to eight bytes, then encrypting each block independently. Two methods are defined for this, ECB (Electronic Code Book) and CBC (Cipher Block Chaining). In ECB each eight byte block is encrypted in isolation, while in CBC each block is combined with the encrypted data from the preceding block before itself being encrypted. Using CBC makes the code a little harder to crack but also has the characteristic that the value of the last encrypted block is dependant upon the values of all the blocks preceding it, which means it can be used as a secure checksum. The cryptogram generated by the card, and that returned by the host, are in fact secure checksums generated in this way, they do not carry any data themselves, but verify the authenticity of the data that accompanies them in clear.
Triple DES is a technique for increasing the security if the DES algorithm by using a longer key. In triple DES a sixteen byte key is provided, but is actually used as two eight byte keys. The technique, which is applied to each eight byte data block in turn, is to encrypt the block with the first key, decrypt it with the second key, then re-encrypt it with the first.
In order to maintain security of keys it is desirable that a DES key is used as little as possible, once only is the ideal. The practicality of doing this when both the card and the host need to use the same key on any given transaction is ensured by using derived keys.
The host maintains a master key (the MDK or Master Derivation Key), which is kept under strict security. Each card is given a unique master key of its own, the UDK (Unique Derivation Key) which is

generated by encrypting the card's account number using the MDK. The UDK is stored in the card at personalisation time. For each transaction the card derives a session key from its UDK using the Application Transaction Counter, which is incremented every time the card receives a "Get Processing Options" command, and so is unique to a transaction, or session. When the host receives the cryptogram, it also receives the card's account number and current ATC, and so can calculate the session key from the MDK.
In practise the host will normally maintain a number of MDKs, and the card will return an index number, the Derivation Key Index (DKI) which will inform the host which one was used to derive this card's UDK. It is also normal practise for the card to hold three UDKs, each generated from a different MDK, so that a different session key is used for cryptogram generation, for secure data transfer, and for message authentication (see Issuer Script Processing, below).

Proprietary Data

As mentioned above, EMV defines a range of tags for proprietary data, that is personalisation data that is not defined or controlled by EMV, but by the payment scheme operator. The interpretation and use of these data items is different for each payment scheme, but generally they are used to extend or qualify the card's decision making capability. Because of interoperability issues it is extremely rare for the terminal to have any interaction with proprietary data.

Issuer Script Processing.

If the transaction is referred online to the host there is an opportunity for the host to send commands to the card. This is normally performed after second Generate AC. In theory almost anything can be done here, from installing a new PIN to a complete software update, but in practise only a limited number of commands are in general use.
Among the possibilities are ordering a card to block itself and refuse to operate further, (e.g. when it has been reported stolen or the cardholder has not paid his account), to allow the cardholder to alter the PIN, or to change account dependant settings when the cardholder's status changes (e.g. credit limit or foreign travel restrictions etc.).
The host makes up the command and sends it to the terminal in a packet labelled "Please send this to the card". The terminal extracts the command and its accompanying data and transmits it verbatim to the card. In this situation the terminal is merely part of the communications system, it takes no active part in the operation itself.
Sensitive data, such as a new PIN value, is encrypted using a session key derived from the card's secure data transfer UDK, the command itself and any non sensitive data are transmitted in clear. All commands and data are authenticated using a MAC (Message Authentication Code). The MAC is generated from the command and data by the same method as the generation of the cryptogram, but using a session key based on the card's MAC UDK, only if the card validates the MAC will it act upon the command.

Glossary of Acronyms

The terms below are mostly explained in the section on understanding EMV transactions, a few are used without explanation and some are related terms that are not mentioned there.

AC Application Cryptogram, encoded data sent from the card to the issuer's host to authenticate the transaction

AFL Application File Locaters, data which enables the terminal to issue the correct Read Record commands to the card in order to obtain the necessary information to conduct the transaction

AID Application Identifier, a series of bytes which identify an application, payment or otherwise, within the card.

AIP Application Interchange Profile, a set of indicators which inform the terminal what features the card supports

Application - The software within the card that services a specific type of transaction, payment or otherwise.

Asymmetric key cryptography - A cryptographic process using two keys, one for encryption, the other for decryption. Normally one of the keys is public.

ATC Application Transaction Counter, a number which is incremented each time the card is used, used for cryptographic key diversification and in Velocity Checking.

ATM Automatic Teller Machine, also known as a cash dispenser or a hole in the wall.

ATR Answer To Reset. A message sent by a smart card to the terminal when it is first powered up, carries information which enables communications to be established.

Authentication - The process of establishing that a party to the transaction is authentic.
Authorisation - The process of establishing that the transaction is allowable
Bit - Binary digit, the smallest unit of data storage within a computer, represents 0 or 1, often interpreted as true of false .
Byte - The basic unit of number and text storage in a computer, eight bits representing a number from 0 to 255 or a single printable character
CIAC - Card Issuer action codes, instructions from the issuer to the card on how to react to various situations.
CBC - Cipher block chaining, a sequential method of using DES encryption
CDA Combined dynamic data authentication and generate application cryptogram
CDOL Card risk management data object list
Certification authority - The organisation overseeing the authentication process, usually the payment scheme operator.
Checksum - The result (usually one or two bytes in length) of a sequential calculation on a series of bytes. If any of the bytes should change then the checksum will also change, used to detect data corruption. See also hash, below.
Contact card - A card which is powered and communicated with using electrical connections between the coupler and the card
Contactless Card - A card which is powered and communicated with using high frequency radio signals between the card and the coupler.
Coupler - The component in the terminal which communicates directly with the card
Cryptogram - A secure checksum used validate a message from the card to the host or vice versa.
CVM Cardholder verification method, e.g. signature, PIN etc.
CVR Card verification results, generated by the card to inform the host of its decision process.
DDA Dynamic data authentication - uses RSA cryptography within the card to verify the card's authenticity
DDOL - DDA data object list, identifies the data to be authenticated using DDA
DEA - Data encryption algorithm, an alternative acronym for DES.
DES - Data encryption standard - the symmetric key algorithm used in smart cards.
ECB - Electronic code book, the simplest method of implementing DES encryption.
EMV - The organisation originally formed and owned by Europay, MasterCard & Visa to co-ordinate and oversee the establishment of standards and specifications for financial transactions using smart cards. Since its establishment Europay has merged with MasterCard and JCB and American Express have become members.
EmvCo - The executive arm of EMV, responsible for publishing specifications, approving products etc.
Generate AC - The command which causes the card to calculate an application cryptogram.
GPO - Get processing options - the first command issued to a payment application, establishes the conditions for the transaction.
Hash - The result (usually sixteen or twenty bytes in length) of a sequential calculation on a series of bytes. If any of the bytes should change then the hash will also change, used to detect data corruption or tampering. See also checksum,above.
Host - The issuer's mainframe computer. NB Sometimes host functions are run by payment schemes as a service their members, or even "short circuited" by large merchants using the "Acquirer Truncation" procedure.
IAC - Issuer action codes, instructions from the issuer to the merchant, provided to the terminal by the card, on how to react to various situations.
ICC - Integrated circuit card, smart card or chip card.
Interoperability - The ability of systems built and managed by a number of diverse organisations to communicate and work together.
Issuer - The organisation issuing the card, usually a bank or credit card company
Issuer script processing - A method by which the issuer can make changes to a card after it has been issued.
MAC - Message authentication code, a secure checksum used to authenticate issuer script commands delivered to a card.
MDK - Master derivation key, the key from which unique DES keys for the cards are derived
Offline - Unable to communicate directly with the host during the transaction.
Online - Able to communicate directly with the host during the transaction.
Diversification - The process of changing a DES key so that the same one is never used twice.
PAN - Primary account number, usually the number embossed on the front of the card.
Payment scheme - an umbrella organisation coordinating debit and/or credit card payment systems

PDOL - Processing options data object list, information the card requires at the start of the transaction, the data is deliverd to the card in Get Processing Options.
Personalisation - The process of writing unique data (ie. the cardholder's personal data) to the card.
PIN - Personal identification number
PIX - The proprietary section of the AID, distinguishes between account types within a single payment scheme
PPSE - Proximity payment systems environment, a directory of the payment applications on a contactless card
Profile - A particular configuration of the payment application, some card applications can select a profile based onPDOL data.
Proprietary - defined and managed by a payment scheme or issuer, rather than by EMV
PSE - Payment systems environment, a directory of the payment applications on a contact card
Read Record - The command sent by the terminal to request data from the card's filing system
RID - The first part of the AID, identifies the payment scheme.
RSA - The asymmetric key cryptography algorithm used by EMV
Select - The command sent by the terminal to the card to activate an application.
SDA - Static data authentication, verification of the data without requiring the card to perform RSA calculations.
Signed Static Application Data - The encrypted data block containing the data hash for SDA
Signed Dynamic Application Data - The encrypted data block containing the data hash for DDA
SHA-1 - The algorithm used for hash generation by EMV
Symmetric key cryptography - encryption method that uses the same key for encryption and decryption.
Tag - A one or two byte identifier for a data object
TAC - Terminal action codes, the merchant's preferences for action during a transaction
Terminal - The merchant's component in the transaction, usually a cash register.
TLV - Tag, length and value, an economical way of encoding and storing data
Triple DES - a modification of the DES algorithm which effectively doubles the length of the key
TVR - Terminal verification results, the results of the terminal's risk assessment tests.
UDK - Unique derivation key, a DES key, unique to a card, derived from a master key held by the issuer.
Velocity Checking - A method of managing financial risk by limiting the number of transactions or total spend that can be allowed offline before an online transaction must be performed.

Barnes International Limited

July 2014