Indian Institute of Technology Bhilai

Card Payment Systems

Group 9
Akash Diwaker - 11740080
Harsh Lathi - 11740380
Satyam Sachan - 11740890

November 15, 2019

Contents

I Technologies Used **3**

1 Magnetic Stripe Cards **4**
1.1 The Working Principle . . . 4
1.2 Relevant Standards . . . 4
1.2.1 Physical Properties . . . 4
1.2.2 Embosssing . . . 5
1.2.3 HiCo and LoCo . . . 5
1.2.4 Magnetic Tracks . . . 5
1.3 Data Encoding in Tracks . . . 6
1.3.1 Track 1 (IATA) . . . 6
1.3.2 Track 2 (ABA) . . . 7
1.3.3 Track 3 (THRIFT) . . . 8
1.4 Magnetic Stripe Cards in Financial Transactions . . . 9
1.5 Drawbacks . . . 9

2 Contact Based Cards **10**
2.1 Working principle . . . 10
2.2 Relevant Standards . . . 10
2.3 Dimensions and contacts . . . 10
2.3.1 Number and loaction of contacts . . . 10
2.3.2 Location relative to other technologies . . . 11
2.3.3 Electrical Characteristics . . . 11
2.3.4 Contacts . . . 12
2.4 Operating Procedure . . . 14
2.4.1 Activaiton, resets and class selection . . . 14
2.4.2 Clock stop and Deactivation . . . 16
2.4.3 Asynchronous character . . . 17
2.4.4 Error Signal . . . 17
2.4.5 Information exchange . . . 17
2.4.6 Characters and Coding convention . . . 18
2.4.7 Answer-to-Reset . . . 18
2.5 Protocol and Parameters Selection . . . 21
2.5.1 PPS request and Response . . . 21
2.6 Message Structure:APDU . . . 21
2.6.1 Command APDU Structure . . . 21
2.6.2 Response APDU Structure . . . 22
2.7 File Management in Cards . . . 23
2.7.1 Elementary files . . . 23
2.7.2 Data Objects . . . 23
2.8 Data Transmission Security . . . 24
2.8.1 Data objects for Plaintext . . . 24
2.8.2 Data objects for security mechanisms . . . 24

2.8.3 Data objects for security mechanisms . . . 24

3 Contactless Smart Cards **27**
3.1 Basic Working Principle . . . 27
3.2 Relevant Standards . . . 27
3.2.1 Physical Properties . . . 27
3.2.2 Dimensions . . . 27
3.2.3 Power Transfer . . . 27
3.2.4 Operating field . . . 28
3.2.5 Signal Interface . . . 28
3.3 Communication Protocols . . . 28
3.3.1 Initial dialogue for proximity cards . . . 28
3.3.2 Communication signal interface Type A . . . 28
3.3.3 Communication signal interface Type B . . . 30
3.3.4 Polling . . . 32
3.3.5 Initialization and Anti-collision . . . 33
3.4 Contactless Smart Cards in Financial Transactions . . . 33
3.5 Protection by Cryptographic Measures . . . 33
3.6 Drawbacks . . . 35

II Transaction Protocols **36**

4 Financial Transactions - `ISO/IEC 8583` **37**
4.1 Message Structure . . . 37
4.1.1 Message Type Identifier (MTI) . . . 37
4.1.2 Bitmap . . . 38
4.1.3 Data Elements . . . 38
4.2 Issues . . . 39

III Security Protocols **40**

5 SSL/TLS **41**
5.1 Transport Layer Security . . . 41
5.1.1 Architecture of SSL/TLS . . . 41
5.1.2 Authentication . . . 42
5.1.3 Confidentiality . . . 44
5.1.4 Integrity . . . 44

IV References **45**

Part I

Technologies Used

Chapter 1

Magnetic Stripe Cards

Magnetic stripe cards are used in various industries due to their ease of use, low manufacturing costs and the simple data formatting. Their extensive use has led to the establishment of various standards (and their subsequent revisions).
As time progressed, newer and safer technologies like contact-based cards have been introduced but have not been able to entirely phase out the magnetic stripe.

1.1 The Working Principle

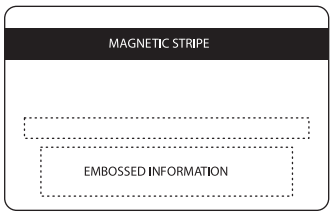
The magnetic stripes contain a 'recording medium' to retain the information. The high retentivity of ferromagnetic material (typically containing iron or nickel) makes such materials ideal for storage. Particles of these materials are analogous to bits in that their two orientations are used to store data in a binary manner.
In the initial state (demagnetized/non-active) all the particles point in the same direction. The 'writing head' of a device may write data onto the card by applying a strong magnetic field to change the direction in which the particles are polarized in. The 'reading head' of a device simply captures the magnetic field of each particle and converts it to the corresponding (pre-agreed) bit.

1.2 Relevant Standards

- The standard `ISO/IEC 7811` specifies the recording techniques used in Identification Cards.
- The standard `ISO/IEC 7813` specifies the properties of financial transaction cards.

1.2.1 Physical Properties

A regular magnetic stripe card looks like this:



The dimensions of the card are as per the standard `ISO/IEC 7811-1` in accordance with the standard `ISO/IEC 7810`, which defines the physical characteristics for identification cards. Some common properties are:

- Length and breadth: 85.72 to 85.47 mm and 54.03 to 53.92 mm respectively.

- Radius of edge rounding: 3.48 to 2.88 mm
- Thickness: 0.84 to 0.68 mm

1.2.2 Embosssing

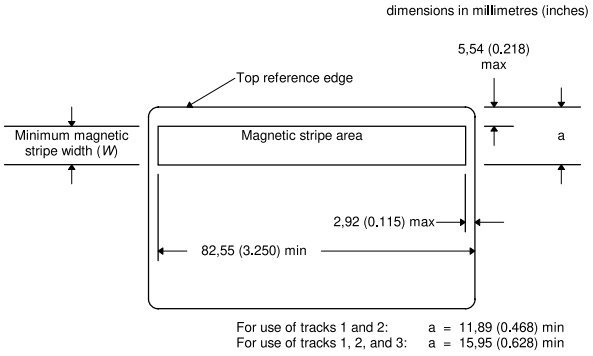
The embossed information present is usually to be read visually . There are machines that can read this information, but this way of storing information is now obsolete (due to physical constraints).

1.2.3 HiCo and LoCo

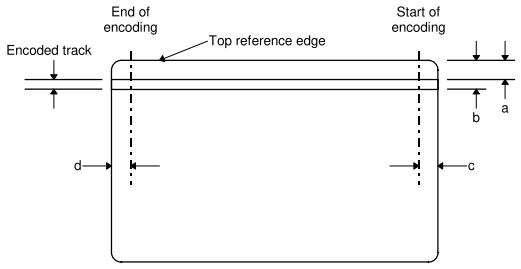
The standards `ISO/IEC 7811-2` and `ISO/IEC 7811-6` describe what are known as LoCo (low coercivity) and HiCo (high coercivity) cards. The basic difference between the two is the strength of the magnetic field use to write onto the tracks of the magnetic card. The HiCo cards are require a higher magnetic field to encode, and thus can be used for credit cards and other cards that are routinely used. On the other hand, cards that contain data that needs to be changed frequently (like hotel room cards) are mainly LoCo cards, as the magnetic energy used to encode these cards is less than that used to encode HiCo cards. Note that reading the two types of cards is not different, the same read head that can read a LoCo card can read a HiCo card. However, a LoCo write head cannot write onto a HiCo card.

1.2.4 Magnetic Tracks

Each card has a minimum of two tracks (track 1 and 2). Each track has an encoding format and configuration. There are cards with three tracks, however the third track is rarely seen in the ususal cards we are familiar with. Thus, the placement of the stripe differs. As per the standard `ISO/IEC 7811`:



The tracks are located in the following manner:



Term	Track 1	Track 2	Track 3
a	5.79 mm (max)	8.33 mm to 9.09 mm	11.63 mm to 12.65 mm
b	8.33 mm to 9.09 mm	11.63 mm to 12.65 mm	15.19 mm to 15.82 mm
c	7.44 ± 1.00 mm	7.44 ± 0.50 mm	7.44 ± 1.00 mm
d	6.93 mm	6.93 mm	-

Note that all the tracks have the same minimum track width of 2.54 mm.

1.3 Data Encoding in Tracks

1.3.1 Track 1 (IATA)

The track is also called the IATA track due to historical reasons. Track properties:

- Average Recording Density (bits per inch) = 210 bpi ± 8% (measured in a longitudinal direction parallel to the top reference edge)
- 7 bits per character
- 79 alphanumeric characters (including control characters)

The coded character set is as follows:

	Char.	Binary								Char	Binary						
		P	2^5	2^4	2^3	2^2	2^1	2^0			P	2^5	2^4	2^3	2^2	2^1	2^0
	space	1	0	0	0	0	0	0		@	0	1	0	0	0	0	0
	!	0	0	0	0	0	0	1		A	1	1	0	0	0	0	1
	“	0	0	0	0	0	1	0		B	1	1	0	0	0	1	0
	#	1	0	0	0	0	1	1		C	0	1	0	0	0	1	1
	$	0	0	0	0	1	0	0		D	1	1	0	0	1	0	0
	%	1	0	0	0	1	0	1		E	0	1	0	0	1	0	1
	&	1	0	0	0	1	1	0		F	0	1	0	0	1	1	0
	‘	0	0	0	0	1	1	1		G	1	1	0	0	1	1	1
	(	0	0	0	1	0	0	0		H	1	1	0	1	0	0	0
	)	1	0	0	1	0	0	1		I	0	1	0	1	0	0	1
	*	1	0	0	1	0	1	0		J	0	1	0	1	0	1	0
	+	0	0	0	1	0	1	1		K	1	1	0	1	0	1	1
	,	1	0	0	1	1	0	0		L	0	1	0	1	1	0	0
	-	0	0	0	1	1	0	1		M	1	1	0	1	1	0	1
	.	0	0	0	1	1	1	0		N	1	1	0	1	1	1	0
	/	1	0	0	1	1	1	1		O	0	1	0	1	1	1	1
	0	0	0	1	0	0	0	0		P	1	1	1	0	0	0	0
	1	1	0	1	0	0	0	1		Q	0	1	1	0	0	0	1
	2	1	0	1	0	0	1	0		R	0	1	1	0	0	1	0
	3	0	0	1	0	0	1	1		S	1	1	1	0	0	1	1
	4	1	0	1	0	1	0	0		T	0	1	1	0	1	0	0
	5	0	0	1	0	1	0	1		U	1	1	1	0	1	0	1
	6	0	0	1	0	1	1	0		V	1	1	1	0	1	1	0
	7	1	0	1	0	1	1	1		W	0	1	1	0	1	1	1
	8	1	0	1	1	0	0	0		X	0	1	1	1	0	0	0
	9	0	0	1	1	0	0	1		Y	1	1	1	1	0	0	1
	:	0	0	1	1	0	1	0		Z	1	1	1	1	0	1	0
	;	1	0	1	1	0	1	1		[	0	1	1	1	0	1	1
	<	0	0	1	1	1	0	0		\	1	1	1	1	1	0	0
	=	1	0	1	1	1	0	1		]	0	1	1	1	1	0	1
	>	1	0	1	1	1	1	0		^	0	1	1	1	1	1	0
	?	0	0	1	1	1	1	1		_	1	1	1	1	1	1	1

NOTE This coded character set is identical to the coded character set in ISO/IEC 7811-6 (derived from ASCII.)

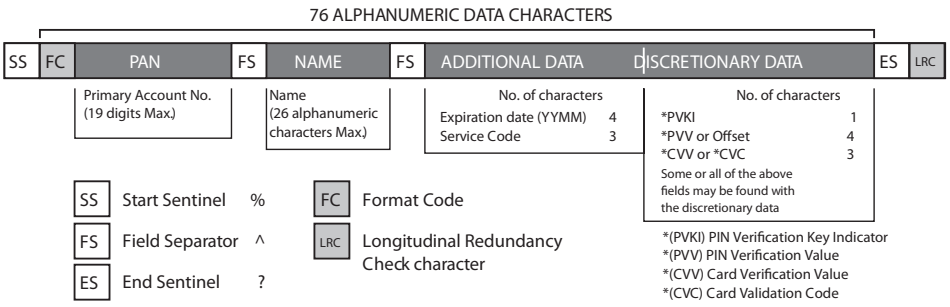
The control characters (**! & * + , : ; < = > @ _**) may not be used for any information data. The characters (**% ?**) are the start sentinel, the field separator and the end sentinel respectively.
Further, this track contains:

- **Format Code**. This is single character field, and equals 'B' for use in financial cards.
- **PAN** (Primary Account Number)
- **Name**
- **Additional data** (the expiry date and service code)

- **Discretionary data** (the PIN Verification Key Indicator, PIN Verification Value, Card Verification Value and the Card Validation Code).

The PAN format is defined in the standard `ISO/IEC 7812`. It consists of two major parts: the **Issuer Identification Numbers (IINs)** and **individual account identification number**. The IINs are predefined by the issuer, and are 8 digits in length. The first digit is the major industry, and the following digits are the issuer identifier digits. The next 10 digits of the PAN are used to identify the card holder. Finally, the last digit is the checksum (calculated with the Luhn Algorithm).
The format in which the data is stored is the following:

- The first character is the **Start Sentinel (%)**.
- The next character is the **Format Code**. It is set to 'B' for credit/debit cards.
- The next 19 characters constitute the **PAN**. The character set is limited to digits for representing the PAN.
- The Field Separator ^.
- The card holder's **name**. Consists of 26 alphanumeric characters.
- The Field Separator ^.
- The **Expiry Date** in YYMM format. If unused, a ^ will be used.
- The **Service Code**. If unused, a ^ will be used.
- The PIN Verification Key Indicator, PIN Verification Value, Card Verification Value and the Card Validation Code fields. Each is optional.
- The End Sentinel **?**.
- The LRC check character.



1.3.2 Track 2 (ABA)

Track properties:

- Average Recording Density (bits per inch) = 75 bpi ± 5% (measured in a longitudinal direction parallel to the top reference edge)
- 5 bits per character
- 40 numeric characters (including control characters)

Note that track 2 cannot have characters from the alphabet, as is also the case for the third track (wherever present). Thus, the first track is the only one that has the name of the card holder.
The coded character set is as follows:

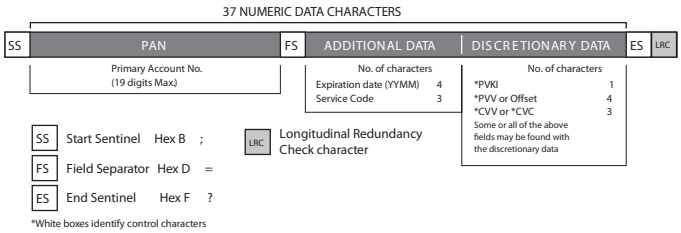
	Char.	Binary						Char.	Binary				
		P	2^3	2^2	2^1	2^0			P	2^3	2^2	2^1	2^0
	0	1	0	0	0	0		**8**	0	1	0	0	0
	1	0	0	0	0	1		**9**	1	1	0	0	1
	2	0	0	0	1	0		**:**	1	1	0	1	0
	3	1	0	0	1	1		**;**	0	1	0	1	1
	4	0	0	1	0	0		**<**	1	1	1	0	0
	5	1	0	1	0	1		**=**	0	1	1	0	1
	6	1	0	1	1	0		**>**	0	1	1	1	0
	7	0	0	1	1	1		**?**	1	1	1	1	1
NOTE This coded character set is identical to the coded character set in ISO/IEC 7811-6 (derived from ASCII.)													

The control characters (**:** $<$ $>$) may not be used for any information data. The characters (**;** $=$ **?**) are the start sentinel, the field separator and the end sentinel respectively.
Further, this track contains:

- **PAN** (Primary Account Number)
- **Additional data** (the expiry date and service code)
- **Discretionary data** (the PIN Verification Key Indicator, PIN Verification Value, Card Verification Value and the Card Validation Code).

The PAN format is as in track 1.
The format in which the data is stored is the following:

- The first character is the **Start Sentinel (;)**.
- The next 19 characters constitute the **PAN**. The character set is limited to digits for representing the PAN.
- The Field Separator =.
- The **Expiry Date** in YYMM format. If unused, a = will be used.
- The **Service Code**. If unused, a = will be used.
- The PIN Verification Key Indicator, PIN Verification Value, Card Verification Value and the Card Validation Code fields. Each is optional.
- The End Sentinel **?**.
- The LRC check character.



1.3.3 Track 3 (THRIFT)

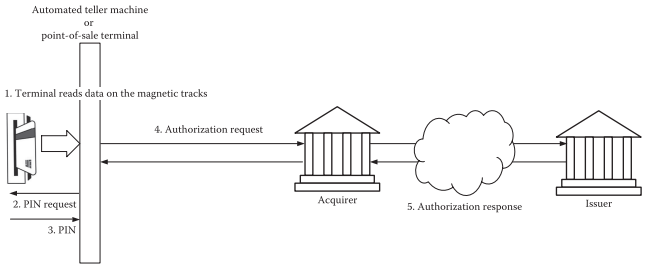
Track properties:

- Average Recording Density (bits per inch) = 210 bpi ± 8% (measured in a longitudinal direction parallel to the top reference edge)
- 5 bits per character
- 107 numeric characters (including control characters)

As mentioned above, while this track's physical properties are well defined in the standard `ISO/IEC 7811-2`, this track is rarely ever present in the magnetic strip cards we see. The encoding standard for this track is as in `ISO/IEC 4909`.

1.4 Magnetic Stripe Cards in Financial Transactions

The widespread use of magnetic stripe cards is evident in their applications in credit cards and debit cards. The general process of a transaction is as follows:



1.5 Drawbacks

Due to the inherent simplicity of the manner in which data is stored, magnetic stripe cards are often considered as easy to manipulate. A few drawbacks are as listed:

- The card holder's details are not encrypted. Thus, anyone with a magnetic card reader who has access to the card can read the data.
- In the same vein as the point above, the exchanges between the card and card reader are not encrypted. Anyone who can pick up the transmitted message will be able to read the data.
- The card is a passive participant in the transaction. This means that as the card has no computational abilities and therefore the stored data is a lot less secure than when it is stored in the IC cards due to their ability to perform cryptographic computations for confidentiality, integrity, and authentication.
- Data recorded on the magnetic stripe can easily be altered using a standard read/write device, and it is difficult to prove such changes afterward.

Chapter 2

Contact Based Cards

A chip-based smart card is a plastic card with a computer chip embedded in it. This chip can either be a memory-only or a microprocessor chip. Their ability to do several complex computations make them most suitable for banking. There are several other applications of chip-based smart cards such as healthcare, transportation, SIM etc. They provide better security of data than magnetic stripe cards that is why most of the banking services now use chip-based cards.

2.1 Working principle

Chip-based cards store data in chip embedded in the card. These chip have a operating system (depends upon the purpose of card) which provides access control. When the card is connected to a interface device, the card authenticates the device via challenge-response mechanism and gives access to relevant data. This data is used by the interface device to perform the transaction.

2.2 Relevant Standards

The standard ISO/IEC 7816 specifies the protocols used in Identification Cards-Integrated circuit cards.

2.3 Dimensions and contacts

Each contact of the chip have a minimum dimension of $2mm \times 1.7mm$ and each contact is isolated from other contacts.

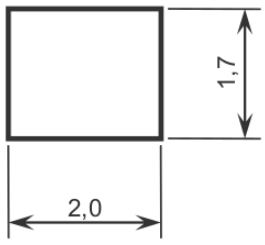


Figure 2.1: Dimension of contact(in mm)

2.3.1 Number and loaction of contacts

There are a total of 8 contacts from C1 to C8. The useage of each contact will be discussed later.
The contact are in front of the card. The dimensions are referenced to the left and upper edges of the front

surface of the card as defined in ISO/IEC 7810 as shown in Figure 2.2.
Unused contact areas are non conductive or electricly isolated.

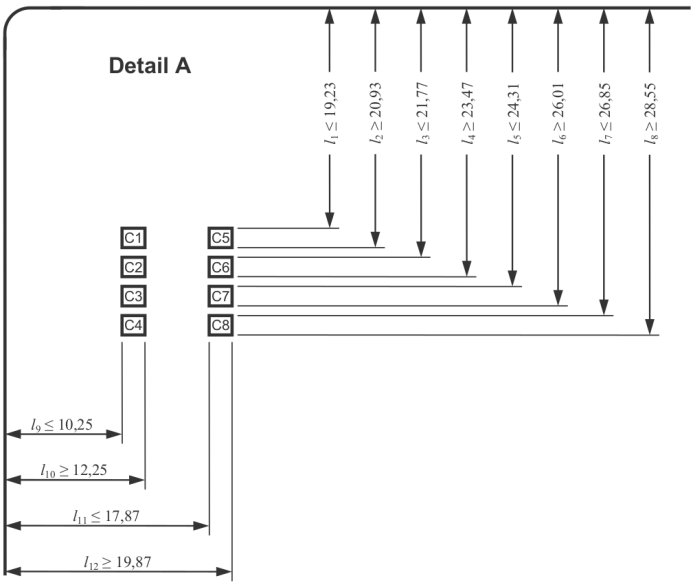


Figure 2.2: Relative Loaction of contacts

2.3.2 Location relative to other technologies

Embossing on the card is generally present on the same side as that of contact while Magnetic stripes are always present on the opposite side to the contacts.

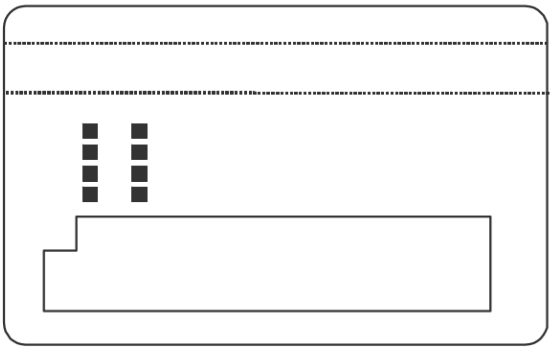


Figure 2.3: Front side of card

2.3.3 Electrical Characteristics

General

The contacts of chip has following usage:

C1: Supply power Input(V_{CC})

C2: Reset signal Input(RST)

C3: Clock signal Input(CLK)

C5: Ground (GND)

C6: Standard or Proprietary use(SPU)

C7: Input/Output for serial data(I/O)

We don't use C4 and C8. These are reserved auxiliary functions.

classes of operating conditions

There are three classes based on V_{CC} supply by interface to card.

1. 5V for class A
2. 3V for class B
3. 1.8V for class C

The card supports atleast one of the above classes. If the interface applies one of the above class the card should operate as specified.

2.3.4 Contacts

C1(V_{CC})

This contact is used to supply the card with power.

The specified Maximum and Minimum values of voltage and current for different classes is shown in table 2.1.

Symbol	Conditions	Minimum	Maximum	Unit
Voltage at $V_{CC}(U_{CC})$	Class A	4.5	5.5	V
	Class B	2.7	3.3	
	Class C	1.62	1.98	
Current at $V_{CC}(I_{CC})$	Class A,at maximum allowed frequency		60	mA
	Class B,at maximum allowed frequency		50	
	Class C,at maximum allowed frequency		30	
	When the clock is stopped		0.5	

Table 2.1: Electrical Characteristics of V_{CC} under normal operating conditions

The maximum charge is half the product of the maximum duration and the maximum variation.
The maximum variation is the difference in supply current with respect to the average value.

Class	Maximum charge	Maximum duration	Maximum variation of I_{CC}
A	20 nA.s	400 ns	100 mA
B	10 nA.s	400 ns	50 mA
C	6 nA.s	400 ns	30 mA

Table 2.2: Spikes on I_{CC}

C2(RST)

This contact is used to provide the card with reset signal.
There are two types of reset, 'cold reset' and 'warm reset'. We will discuss about these in further details later.

Symbol	Conditions	Minimum	Maximum	Unit
U_{IH}		$0.8U_{CC}$	U_{CC}	V
I_{IH}	U_{IH}	-20	+150	μA
U_{IL}		0	$0.12U_{CC}$	V
I_{IL}	U_{IL}	-200	+20	μA
t_R t_F	$C_{IN} = 30pF$		1	μs

Table 2.3: Electrical characteristics of RST under normal operating conditions

The voltage shall remain between -0.3V to U_{CC}+0.3V.

C3(CLK)

This contact is used to provide card with clock signal. The value of frequency of the clock is denoted by f.

Symbol	Conditions	Minimum	Maximum	Unit
U_{IH}		$0.7U_{CC}$	U_{CC}	V
I_{IH}	U_{IH}	-20	+100	μA
U_{IL}	class A and class B	0	0.5	V
U_{IL}	class C	0	0.2 U_{CC}	V
I_{IL}	U_{IL}	-100	+20	μA
t_R t_F	$C_{IN} = 30pF$		9% of cycle	

Table 2.4: Electrical characteristics of CLK under normal operating conditions

The voltage shall remain between -0.3V to U_{CC}+0.3V.

If not specified, the duty cycle of the clock is 40% to 60% of the cycle. Switching of frequency should happen only

- after completion of an answer to reset, while the card is waiting for a character.
- after completion of a successful PPS exchange, while the card is waiting for a character.

No information shall be exchanged while switching the value of frequency.

C6(SPU)

This contact is used for standard and proprietary use.

Depending upon whether the card uses SPU or not, the first TB for T=15 shall be present or absent in the Answer-to-Reset: this global interface byte indicates whether the use is standard or proprietary.

C7(I/O)

This contact is used as input(reception mode) and output(transmission mode).

The electrical circuit has the following two states:

- H if the card and the interface device are in reception mode or if the transmitter imposes this state;
- L if the transmitter imposes this state.

Both the card and interface should be in either one of above states to successfully exchange information.

The following are the electrical characteristics of I/O under normal operating conditions.

Symbol	Conditions	Minimum	Maximum	Unit
U_{IH}		$0.7U_{CC}$	U_{CC}	V
I_{IH}	U_{IH}	-300	+20	μA
U_{IL}		0	$0.5U_{CC}$	V
I_{IL}	U_{IL}	-1000	+20	μA
U_{OH}	External pull-up resistor: 20 kÎ to U_{CC}	$0.7U_{CC}$	U_{CC}	V
I_{OH}	U_{OH} and external pull-up resistor: 20 kÎ to U_{CC}		+20	μA
U_{OL}	I_{OL} = 1 mA for class A and class B, $I_{OL} = 500\mu A$ for class C	0	$0.15U_{CC}$	V
t_R t_F	$C_{IN} = 30pF$ $C_{OUT} = 30pF$		1	

Table 2.5: Electrical characteristics of I/O under normal operating conditions

2.4 Operating Procedure

After the interface device is mechanically connectd to the card the device applies certain classes of operating conditions.

It first applies activation and then a cold reset. It may or may not apply warm reset. If the card supports the class it answers the reset with Answer-to-reset.

The interface determines the class of operating conditions and start PPS(Parameters and Protocol selection) exchange and agree upon a transmission protocol for exchanging information.

2.4.1 Activaiton, resets and class selection

Activation

In order to initiate an interaction with the mechanically connected card the interface device activates the electrical circuits according to a class of operating conditions: A, B or C, in the following order.

- puts RST to L.
- supplies power to V_{CC}.
- puts I/O in the interface device in reception mode.
- provides CLK with a clock signal.

The delays between powering V_{CC}, setting I/O in reception mode and providing the clock signal on CLK are not defined.

Cold reset

By the end of activation the card is ready for a cold reset. The internal state of the card before a cold reset does not matter.

According to Figure 2.4, the clock signal is applied to CLK at time T_a. The card should set I/O to state H within 200 clock cycles (delay t_a) after the clock signal is applied to CLK (at time $T_a + t_a$). The cold reset results from maintaining RST at state L for at least 400 clock cycles (delay t_b) after the clock signal is applied to CLK (at time $T_a + t_b$). The interface device ignores the state on I/O while RST is at state L.

At time T_b, RST is put to state H. The answer on I/O should begin between 400 and 40,000 clock cycles (delay t_c) after the rising edge of the signal on RST (at time $T_b + t_c$). If the answer does not begin within 40,000 clock cycles with RST at state H, the interface device performs a deactivation.

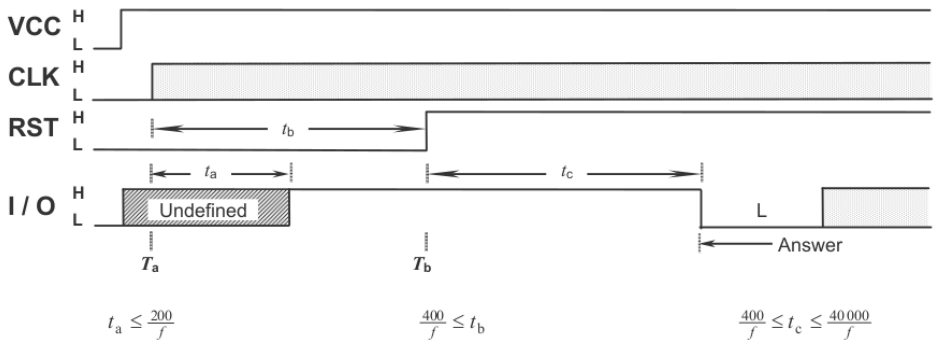


Figure 2.4: Activation and cold reset

Warm reset

The interface device may warm reset the card at any time, even during the answer to reset, but not before reception of the mandatory characters TS and T0. The warm reset should not be initiated in less than 4464 ($= 12 \times 372$) clock cycles after the leading edge of character T0.

The interface device initiates a warm reset(at time T_c) by putting RST to state L for at least 400 clock cycles(delay t_e) while V_{CC} remains powered and CLK provided with a suitable and stable clock signal. The card should set I/O to state H within 200 clock cycles (delay t_d) after state L is applied to RST(at time $T_c + t_d$). The interface device ignores the state on I/O while RST is at state L.

At time T_d , RST is put to state H. The answer on I/O shall begin between 400 and 40,000 clock cycles(delay t_f) after the rising edge of the signal on RST(at time $T_d + t_f$). If the answer does not begin within 40,000 clock cycles with RST at state H, the interface device performs a deactivation.

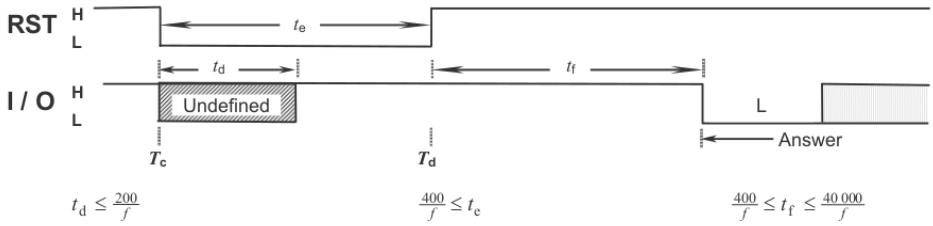


Figure 2.5: Warm reset

Class selection

- If the Answer-to-Reset carries a class indicator indicating the class being applied, then normal operation continues. Otherwise, the interface device performs a deactivation and after a delay of at least 10ms, applies another class supported by the card.
- If the Answer-to-Reset carries no class indicator, then the interface device maintains the current class. If after completion of the answer-to-reset the card does not operate, then the interface device performs a deactivation and after a delay of at least 10 ms, applies another class.
- If the card does not answer-to-reset, then the interface device performs a deactivation and either after a delay of at least 10 ms, applies another class, if any, or abort the selection process. After abortion of a selection process, the interface device may initiate another selection process.

Figure 2.6 gives the basic idea of class selection process.

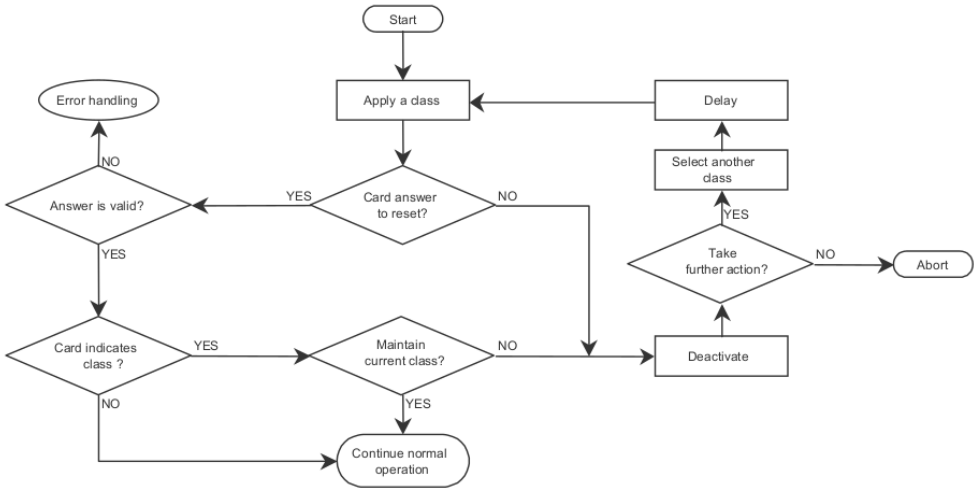


Figure 2.6: Class selection by interface

2.4.2 Clock stop and Deactivation

For card supporting clock stop, when the interface expects no transmission from card and the I/O has remained on H for at least 1860 clock cycles, then according to Figure 2.7, the interface device may stop the clock on CLK.

When the clock is stopped (from time T_e to time T_f), CLK shall be maintained either at state H or at state L according to the clock stop indicator X.

At time T_f, the interface device restarts the clock and the information exchange on I/O may continue after at least 700 clock cycles (at time $T_f + t_h$).

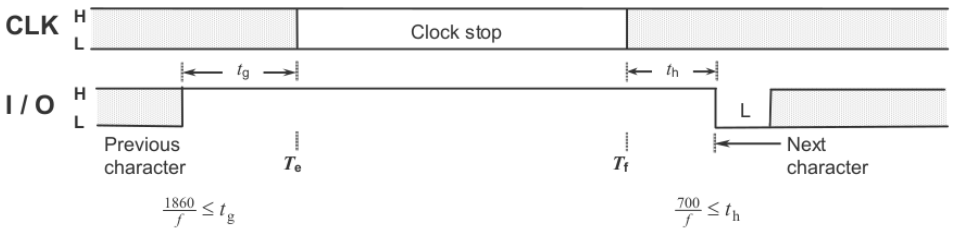


Figure 2.7: Clock Stop

When information exchange is completed or aborted, the interface device deactivates the circuit in the following order:

- RST shall be put to state L.
- CLK shall be put to state L.
- I/O shall be put to state L.
- VCC shall be deactivated.

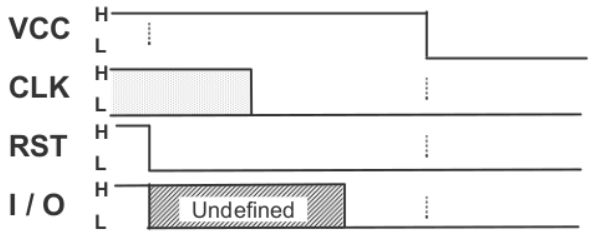


Figure 2.8: Deactivation

2.4.3 Asynchronous character

The information is exchanged in terms of characters where each character consists of moments numbered 1 to 10. Each moment has either state H or L.

- Before moment 1, the electrical circuit I/O should be at state H.
- Moment 1 should be at state L. It is the character start.
- Moments 2 to 9 shall encode a byte according to a coding convention.
- Moment 10 shall encode the character parity.
- After moment 10, both the card and the interface device shall remain in reception mode (in error-free operation) for a certain time of pause, so that the electrical circuit I/O remains at state H.

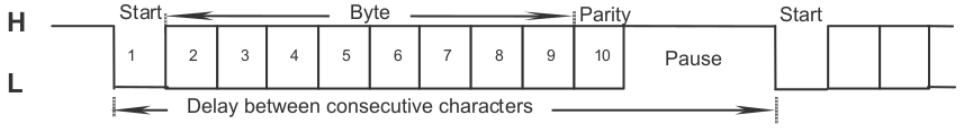


Figure 2.9: Character frame

2.4.4 Error Signal

When character parity is incorrect, the receiver transmits an error signal on the electrical circuit I/O and expects a repetition of the character.

To signal an error, the receiver puts I/O to state L in receiver for 1 etu minimum to 2 etu maximum.

To detect an error signal, the transmitter reads I/O.

- The correct reception is assumed if the state is H.
- The incorrect reception is assumed if the state is L. After a delay of at least 2 etu from the detection of the error signal, the transmitter repeats the character.

If either the card or the interface device provides no character repetition, it ignores and does not suffer damage from the incoming error signal.

2.4.5 Information exchange

After completion of answer to reset the card waits for characters from interface device.

If TA_2 is present in the Answer-to-reset (card in specific mode), then the interface device shall start the specific transmission protocol using the specific values of the transmission parameters.

Otherwise (card in negotiable mode), for the transmission parameters, the values used during the answer to reset shall continue to apply.

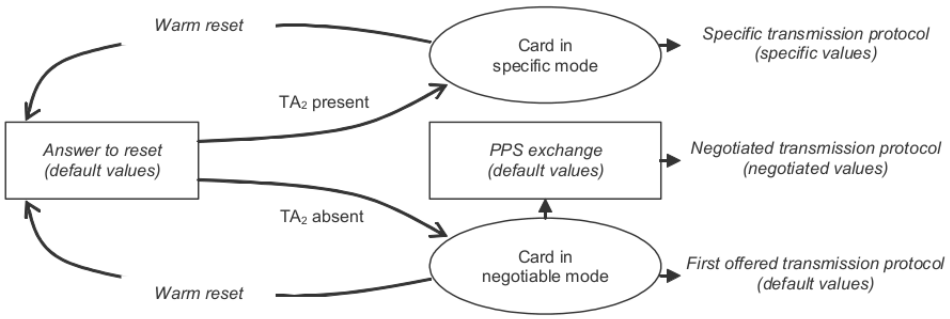


Figure 2.10: Selection of transmission parameters and protocol

2.4.6 Characters and Coding convention

The first character of the data is called "initial character" denoted by TS which is followed by second character named "format character" which is denoted by T0.

The initial character TS has two possible patterns.

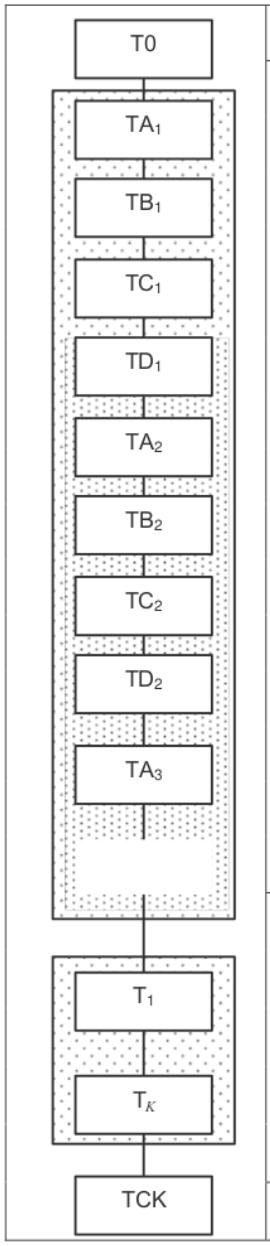
- (H)LHHL LLL LLH sets up the inverse convention: state L encodes value 1 and moment 2 conveys the most significant bit (msb first). When decoded by inverse convention, the conveyed byte is equal to '3F'.
- (H)LHHL HHH LLH sets up the direct convention: state H encodes value 1 and moment 2 conveys the least significant bit (lsb first). When decoded by direct convention, the conveyed byte is equal to '3B'.

The initial character is followed by a sequence of at most 32 characters.

- Format character T0 is mandatory. Others are optional.
- The interface charaters TA, TB, TC, TD.
- The historical characters $T_1, T_2, T_3 \ldots T_K$
- The check character TCK.

2.4.7 Answer-to-Reset

The following image illustrates the Answer-to-Reset.



Format byte T0 (mandatory)		Encodes Y_1 and K
Interface bytes (optional)		
	TA1	Global, encodes Fi and Di
	TB1	Global, deprecated
	TC1	Global, encodes N
	TD1	Structural, encodes Y_2 and T
	TA2	Global, specific mode byte
	TB2	Global, deprecated
	TC2	Specific to T=0, see 10.2
	TD2	Structural, encodes Y_3 and T
	For $i > 2$,	
	TD$_{i-1}$	Structural, encodes Y_i and T
	TA$_i$	— Specific to T after T from 0 to 14 in TD$_{i-1}$
	TB$_i$	— Global after T=15 in TD$_{i-1}$
	TC$_i$	
	TD$_i$	Structural, encodes Y_{i+1} and T
Historical bytes (optional)		
	T1	See ISO/IEC 7816-4
	T2	
	…	
	TK	
Check byte TCK (conditional)		

Figure 2.11: Answer-to-reset

Format byte T0

- Bits 8 to 5 form an indicator Y_1.
- Bits 4 to 1 encode a number K.

Figure 2.12 shows the coding of T0.

Figure 2.12: Coding of T0

The Bits of Y_1(from msb) state whether TD_1, TC_1, TB_1, TA_1 are present or not.

Interface Bytes

The coding structure of TD is as following:

- Bits 8 to 5 form an indicator Y_{i+1}.
- Bits 4 to 1 encode a number T.

Figure 2.13: Coding of TD_i

Here, Bit 5 corresponds to TA_i, Bit 6 to TB_i, Bit 7 to TC_i and Bit 8 to TD_i.These bits of Y_i state whether corresponding interface bytes are present or not.

The type T refers to transmission protocol or qualifies interface Bytes.

- T=0 refers to half-duplex transmission of characters.
- T=1 refers to half-duplex transmission of blocks.
- T=2 and T=3 are reserved for future full-duplex operations.
- T=4 is reserved for an enhanced half-duplex transmission of characters.
- T=5 to T=13 are reserved for future use by ISO/IEC JTC 1/SC 17.
- T=14 refers to transmission protocols not standardized by ISO/IEC JTC 1/SC 17.
- T=15 does not refer to a transmission protocol, but only qualifies global interface bytes.

Each interface byte TA, TB and TC is either global or specific. The interpretation of TA_i, TB_i, TC_i for $i > 2$ depends on the type T encoded in TD_{i-1} .

1. For T=0 to T=14, TA_i, TB_i, TC_i are specific to the transmission protocol T.
2. After T=15, TA_i, TB_i, TC_i are global.

Bits 8 to 5 of TA_1 encodes the maximum frequency(f_{max}) supported by the card and clock rate conversion integer(F_i) and Bits 4 to 1 indicate the value of Baud rate adjustment integer(D_i).
TB_1, TB_2 are deprecated and are no longer used i.e. the interface should ignore them.
TC_1 encodes the extra Gaurd Time integer ranging from 0 to 255.

Historical Bytes and Check Byte

The historical byte describes the operating charcteristics of card. If $K = 0$ then Historical Byte is not present.

For check byte(if present), the XOR of all the Bytes from T0 to TCK shall give '00'.Any other value is invalid. If T=0 is indicated by default only then TCK is absent.

2.5 Protocol and Parameters Selection

The interface device transmits a PPS request to the card. The card recieves the request and sends back a response to interface device.

2.5.1 PPS request and Response

A PPS request and response each consists of initial bytes PPSS and PPS0 followed by three optional bytes PPS_1, PPS_2, PPS_3 and a check byte PCK.

- PPSS identifies PPS request and response and is set to 'FF'.
- In PPS0, bit 5,6,and 7 indicate the presence of PPS_1, PPS_2, and PPS_3 while bit 4 to 1 indicate the value of T(Transmission Protocol).Bit 8 is reserved for future use.
- PPS_1 allows the interface device to propose values of F and D to the card. If it is not present default values are used.
- PPS_2 allows the interface device to propose a use of SPU.
- PPS_3 is reserved for future use.
- XOR of all bytes from PPSS to PCK should give '00'.

2.6 Message Structure:APDU

APDU(Application protocol data unit)s are used to exchange data between the card and the interface device. The design of the APDU that comply with ISO/IEC 7816-4 does not depend on the transmission protocol.

2.6.1 Command APDU Structure

The APDU command structure consists of a header and a body as shown in Figure 2.14.

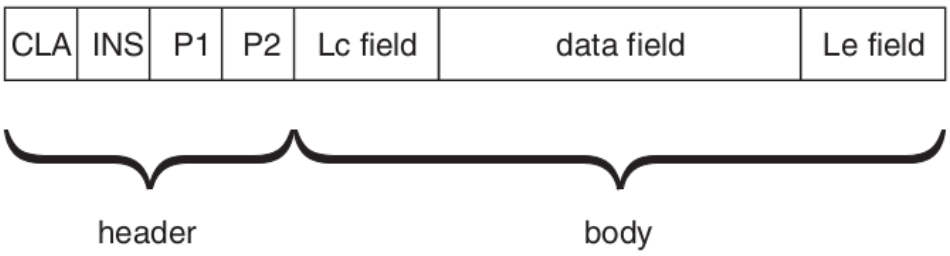


Figure 2.14: Command APDU Structure

The Header has four elements: the class byte (CLA), the instruction byte (INS), and two parameter bytes (P1 and P2).

The class byte is used to identify applications and their specific command sets, for example the class byte for credit cards is '8X' while 'A0' is used for GSM.

The next byte in the header is the instruction byte, which codes the actual command.

The two parameter bytes are primarily used to provide more information for the command selected by the instruction byte.

The body of the APDU command consistes of three fields L_c(Length command) field, data field and L_e(Length expected) field. The body of command can be omitted. It serves two purposes. First, it specifies the length of the data field sent to the card and the length of the data returned by the card. Second, it contains the command data sent to the card.

The L_c and L_e fields usually have a size of 1 byte, but they can be converted into three bytes.

2.6.2 Response APDU Structure

The response APDU sent by the card in reply to a command APDU consists of an optional body and a mandatory trailer, as shown in Figure 2.15.

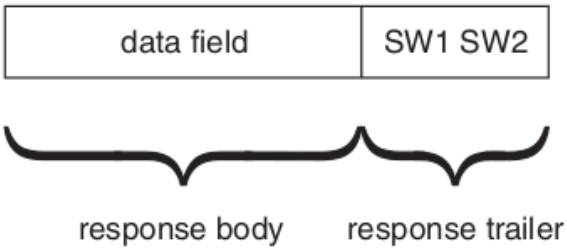


Figure 2.15: Response APDU Structure

The response body consists of a single field called data field the length of which is specified by L_e byte of preceding APDU command. If due to some error or incorrect parameters the card terminates the command processing, the length of data field can be 0 irrespective of specified L_e byte. This is incated by the trailer of the response.

The trailer bytes(SW1 SW2) contains response of the command that is why the card must always send the trailer. Figure 2.16 shows the classification scheme of return code (SW1 SW2).

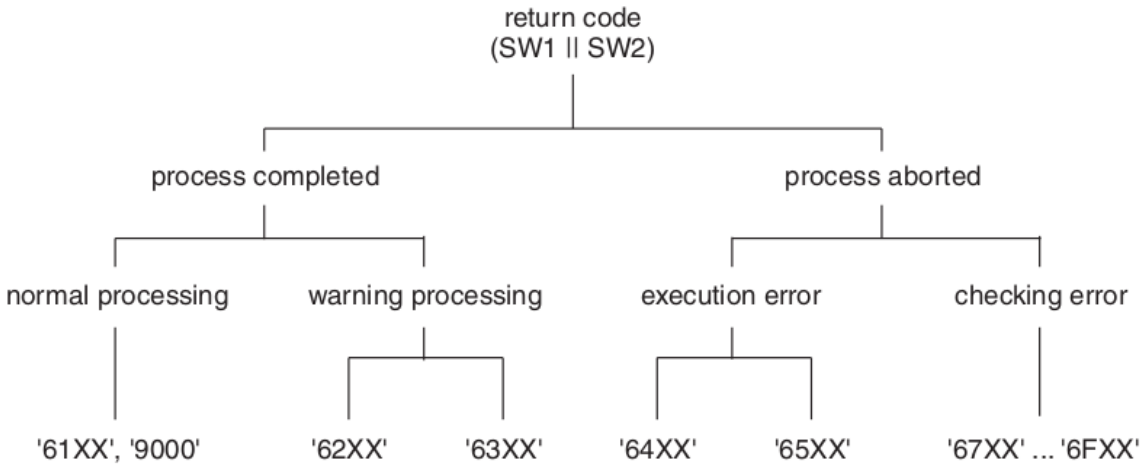


Figure 2.16: Return code classification scheme defined by ISO/IEC 7816-4

2.7 File Management in Cards

The applications and data in the cards are stuctured in the following way. At root there is a Master file in which there can be several, Data objects, Elementary files and Dedicated files. The Dedicated files can further have Data objects, Dedicated files and Elementary files. The Elementary files can have Data objects, Records and Data strings. Figure 2.17 shows the Structure of applications and data in cards.

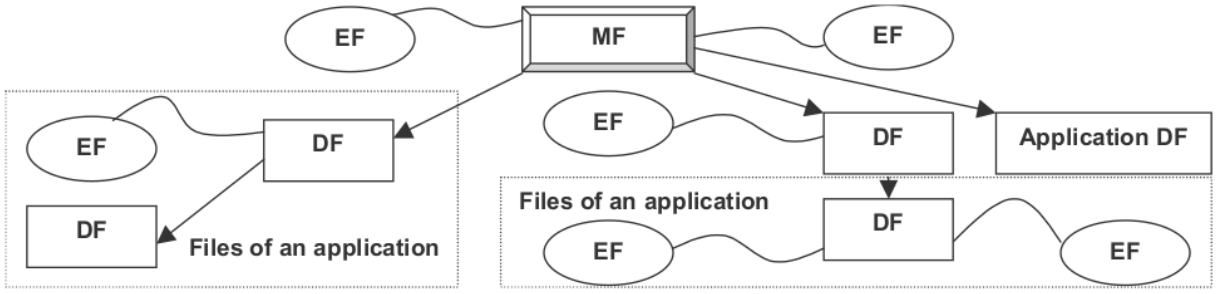


Figure 2.17: Heirarchy of Files in cards

2.7.1 Elementary files

Elementary files are typically of two types in cards:

- An internal Elementary file that stores data used by the card for management and control puposes.
- An external elemenatary file that stores data used by outside world.

The structure of elemenatry files can be of five types:

- Transparent Structure: The data is stored in a stream of bytes.
- Linear structure with records of fixed size: The data is stored in form of records of fixed length.
- Linear structure with records of variable size.
- Cyclic structure with records of fixed size (the arrow references the most recently written record).
- SIMPLE-TLV or BER-TLV structure.

2.7.2 Data Objects

There are two types of Data Objects used in chip based cards:

SIMPLE-TLV

Each SIMPLE-TLV data object consits of two or three consecutive fields Tag, Length and Value.

- Tag field T contains a single byte encoding a number from 1 to 254. The values 00 and FF are invalid.
- Length field L consists of one or three consecutive bytes.
  1. If the first byte is not set to FF then the length field consists a number from 0 to 254.
  2. If the first byte is set to FF then the length field consists a number from 0 to 65535.
- Value field V contains the actual data and the number of bytes V is the value of Length field L. If L is null then the data object is empty.

BER-TLV

In BER-TLV also there are two or more consecutive fields Tag, Length and Value.

- The tag field T consists of one or more consecutive bytes. It encodes a class, a type and a number.
- The length field consists of one or more consecutive bytes. It encodes an integer L.
- If L is not null, then the value field consists of L consecutive bytes. If L is null, then the data object is empty.

2.8 Data Transmission Security

To prevent evesdropping and other types of attacks during data transmission various mechanisms and techniques are used. The objective of secure messaging is to ensure the authenticity, and if necessary the confidentiality, of part or all of the transmitted data.

ISO/IEC 7816-4 specifies secure messaging based on embedding of data in TLV coded data objects. There are three types of data objects defined:

- Data Objects for Plaintext.
- Data Objects for security mechanisms.
- Data Objects for auxiliary functions.

The class Byte of APDU message indicates whether secure messaging is used or not.

2.8.1 Data objects for Plaintext

All the data that is not BER-TLV coded must be encoded in data objects. The tags of the data objects indicate whether data is BER-TLV coded. Bit 1 of these tags indicates whether the data object is included in the computation of the cryptographic checksum.

2.8.2 Data objects for security mechanisms

Data objects into security mechanisms are divided into objects used for authentication and objects used for confidentiality.

For authentication cryptographic Checksum and Digital signatures are used. For confidentiality data encryption and tags necessary for the same are used.

2.8.3 Data objects for security mechanisms

For authentic transmission of APDU i.e. to protect the data against manipulation during transmission first the command APDU in its initial format is generated. The data field of the APDU is converted into TLV coded data with the appropriate padding. The cryptographic checksum(CCS) of the datagram is calculated and a data object containing the CCS is appended to the APDU. The new command APDU as shown in Figure 2.18 is now sent to the card. The response APDU by the card is also generated in a similar manner.

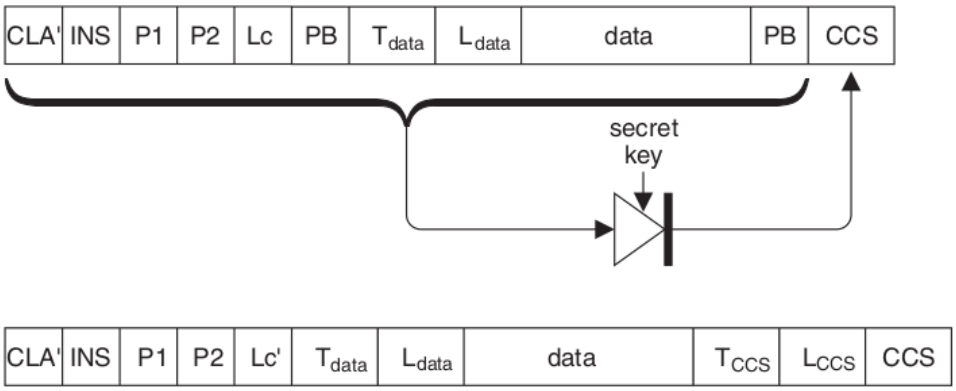


Figure 2.18: Command APDU for Authentic transmission

For confidential transmission of data First the command APDU is generated and its Cryptographic checksum is calculated in the same way as in authentic transmission. After calculation of CCS the whole data field is then encrypted using a secret key and the new data object with the encrypted value is then attached to the APDU. This new command APDU as shown in Figure 2.19 is now sent to the card.

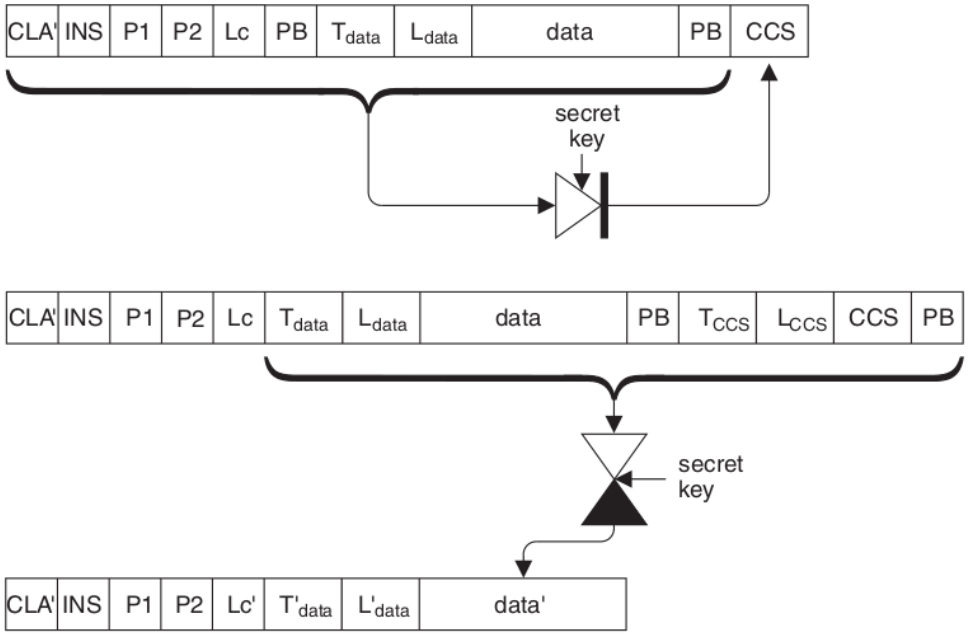


Figure 2.19: Command APDU for Confidential transmission

To detect the deletion or insertion of an APDU immediately, Send sequence count is used. The interface device generates a random number x and sends it to the card. The card now with the Response APDU send x+1. The next APDU interface sends will contain x+2. Both interface and card increment the value recieved by 1 and send it.

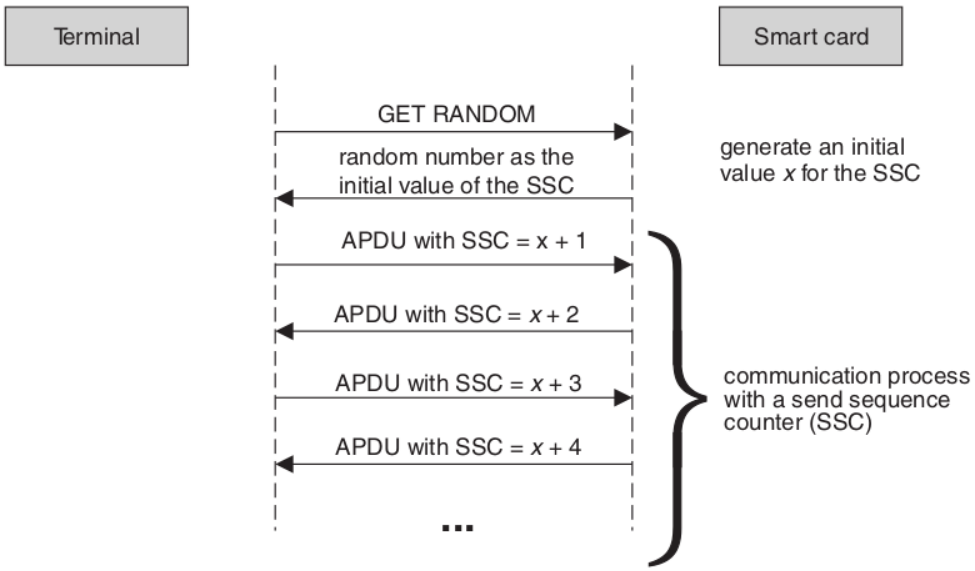


Figure 2.20: Command APDU for Confidential transmission

Use of Send sequence count combined with APDU message also provides security against replay attacks as with a good encryption algorithm changing a single bit will affect around 50% of the message.

Chapter 3

Contactless Smart Cards

3.1 Basic Working Principle

Contactless smart card, in which the chip communicates with the card reader through RFID induction technology. These cards require only close proximity to an antenna to complete transaction. They are often used when transactions must be processed quickly or hands-free, such as on mass transit systems, where smart cards can be used without even removing them from a wallet.

3.2 Relevant Standards

- `ISO/IEC 14443` - Defines proximity cards used for identification, and the transmission protocols for communicating with it.
- `ISO/IEC 15693` - ISO standard for vicinity cards, i.e. cards which can be read from a greater distance as compared with proximity cards.

ISO/IEC 14443 uses following terms for components:
PCD:Proximity coupling device (PCD), The reader/writer device that uses inductive coupling to provide power to the PICC and also to control the data exchange with the PICC.
PICC: Proximity Integrated Circuit Card, An ID-1 card type into which integrated circuit(s) and coupling means have been placed and in which communication to such integrated circuit(s) is done by inductive coupling in proximity of a coupling device.

3.2.1 Physical Properties

The PICC shall have physical characteristics according to the requirements specified for ID-1 cards in ISO/IEC 7810.

3.2.2 Dimensions

The nominal dimensions of the PICC shall be as specified in ISO/IEC 7810 for the ID-1 type cards.

3.2.3 Power Transfer

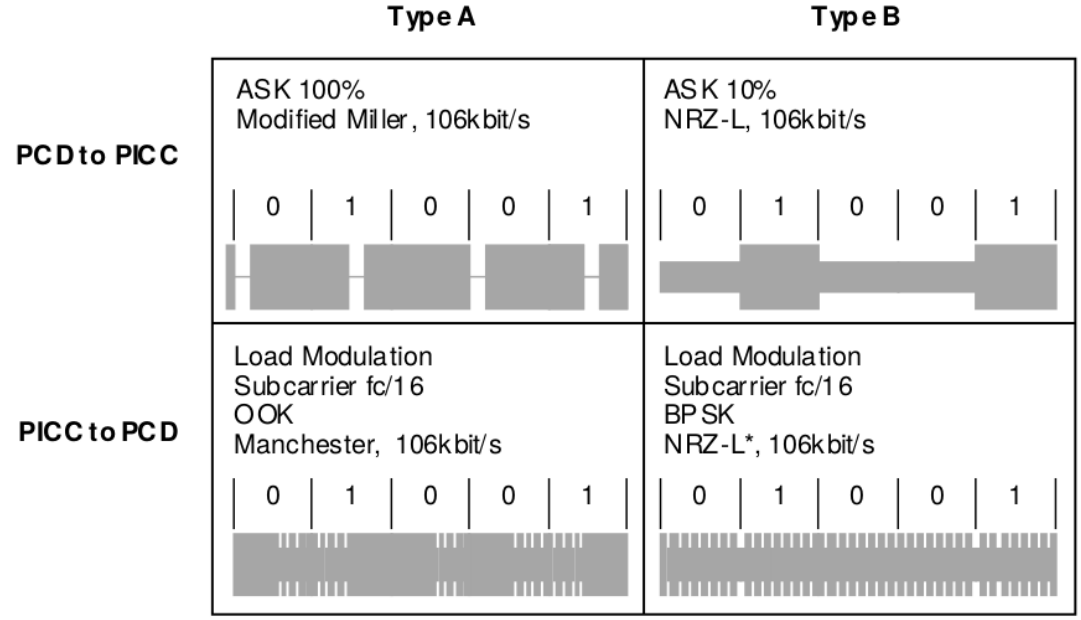
The PCD shall produce an energizing RF field which couples to the PICC to transfer power and which shall be modulated for communication.
The frequency (fc) of the RF operating field shall be 13.56 MHz ± 7 kHz.

3.2.4 Operating field

The minimum unmodulated operating field shall be Hmin and has a value of 15 A/m (rms).
The maximum unmodulated operating field shall be Hmax and has a value of 75 A/m (rms).
A PICC shall operate as intended continuously between Hmin and Hmax.

3.2.5 Signal Interface

Two communication signal interfaces, Type A and Type B, are described in the following clauses. The PCD shall alternate between modulation methods when idling before detecting the presence of a PICC of Type A or Type B. Only one communication signal interface may be active during a communication session until deactivation by the PCD or removal of the PICC. Subsequent session(s) may then proceed using either modulation method.



* Inversion of data is also possible

3.3 Communication Protocols

3.3.1 Initial dialogue for proximity cards

The initial dialogue between the PCD and the PICC shall be conducted through the following consecutive operations:

- activation of the PICC by the RF operating field of the PCD
- PICC waits silently for a command from PCD
- transmission of a command by PCD
- transmission of a response by PICC

3.3.2 Communication signal interface Type A

Data Format:
Data is stored the form of frames. We would define the frame format used during initialization and anticollision. We wil also define the bit representation and coding.
Now, Frames shall be transferred in pairs. PCD to PICC followed by PICC to PCD in the following sequence.

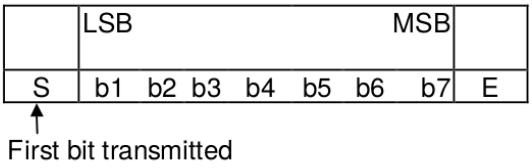
- PCD Frame:
  - PCD start of communication
  - Information and, where required, error detection bits sent by the PCD
  - PCD end of communication
- Frame delay time PCD to PICC
- PICC Frame:
  - PICC start of communication
  - Information and, where required, error detection bits sent by the PICC
  - PICC end of communication
- Frame delay time PICC to PCD

The frame delay time FDT is defined as the time between two frames transmitted in opposite directions. The two basic frame formats are defined as following:
1. Short Frame
2. Standard Frame

Short Frame:
A short frame is used to initiate communication and consists of the following, in the same order :

- Start of communication
- 7 data bits transmitted. LSB first.
- End of communication

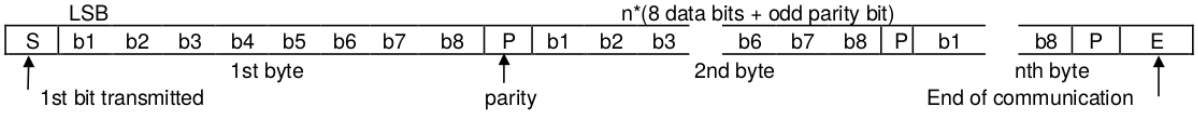
No parity bit is added. The following figure shows the structure of a short frame.



Standard Frame:
Standard frames are used for data exchange and consist of :

- Start of communication
- n*(8 data bits + odd parity bit), with n≥1. The LSB of each byte is transmitted first. Each byte is followed by an odd parity bit. The parity bit P is set such that the number of 1s is odd in (b1 to b8, P) ;
- End of communication

The following figure shows the structure of a short frame.



Communication PCD to PICC:

- **Data Rate** - The data bit rate for the transmission during initialization and anticollision shall be fc/128 (106 kbit/s).
- **Bit representation and coding** - The following sequences are defined:

sequence X	after a time of 64/fc a "pause" shall occur
sequence Y	for the full bit duration (128/fc) no modulation shall occur
sequence Z	at the beginning of the bit duration a "pause" shall occur

They are used to code the following information.

logic "1"	sequence X
logic "0"	sequence Y with the following two exceptions: (i) If there are two or more contiguous "0"s, sequence Z shall be used from the second "0" on. (ii) If the first bit after a "start of frame" is "0" , Sequence Z shall be used to represent this and any "0"s which follow directly thereafter.
Start of communication	sequence Z
End of communication	logic "0" followed by sequence Y
No information	at least two sequences Y

Communication PICC to PCD:

- **Data Rate** - The data bit rate for the transmission during initialization and anticollision shall be fc/128 (106 kbit/s).
- **Bit representation and coding** - It will be as following:

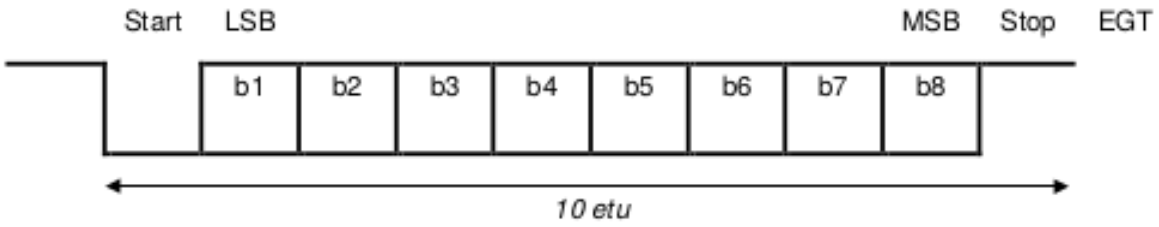
sequence D	the carrier shall be modulated with the subcarrier for the first half (50%) of the bit duration
sequence E	the carrier shall be modulated with the subcarrier for the second half (50%) of the bit duration
sequence F	the carrier is not modulated with the subcarrier for one bit duration
logical "1"	sequence D
logical "0"	sequence E
Start of communication	sequence D
End of communication	sequence F
No information	no subcarrier

3.3.3 Communication signal interface Type B

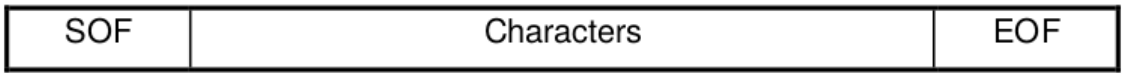
Data Format:
Data is stored the form of frames. We would define the character, frame format used during initialization and anticollision for PICCs of type B. We wil also define the bit representation and coding.
Character Transmission format:
Bytes are transmitted and received between PICCs and a PCD by characters, the format of which during the Anti- collision sequence is as follows :

- 1 start bit at logic "0" ;
- 8 data bits transmitted, LSB first ;
- 1 stop bit at logic "1".

The transmission of one byte is performed with a character requiring 10 etu(Elementary time unit) as illustrated below:

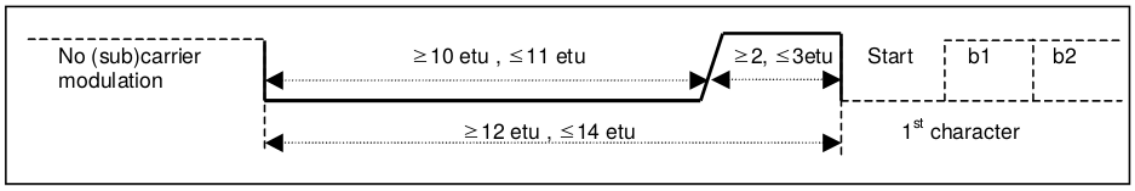


Frame Format
PCDs and PICCs shall send characters as frames. The frame is normally delimited by SOF and by EOF as shown in following figure.



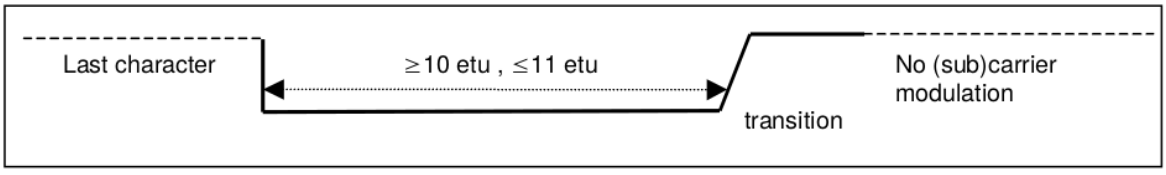
SOF
SOF is composed of:

- one falling edge ;
- followed by 10 etu with a logic "0" ;
- followed by one single rising edge located anywhere within the following etu ;
- followed by at least 2 etu (but no more than 3 etu) with a logic "1".



EOF
EOF is composed of:

- one falling edge ;
- followed by 10 etu with a logic "0" ;
- followed by one single rising edge located anywhere within the following etu.



Communication PCD to PICC:

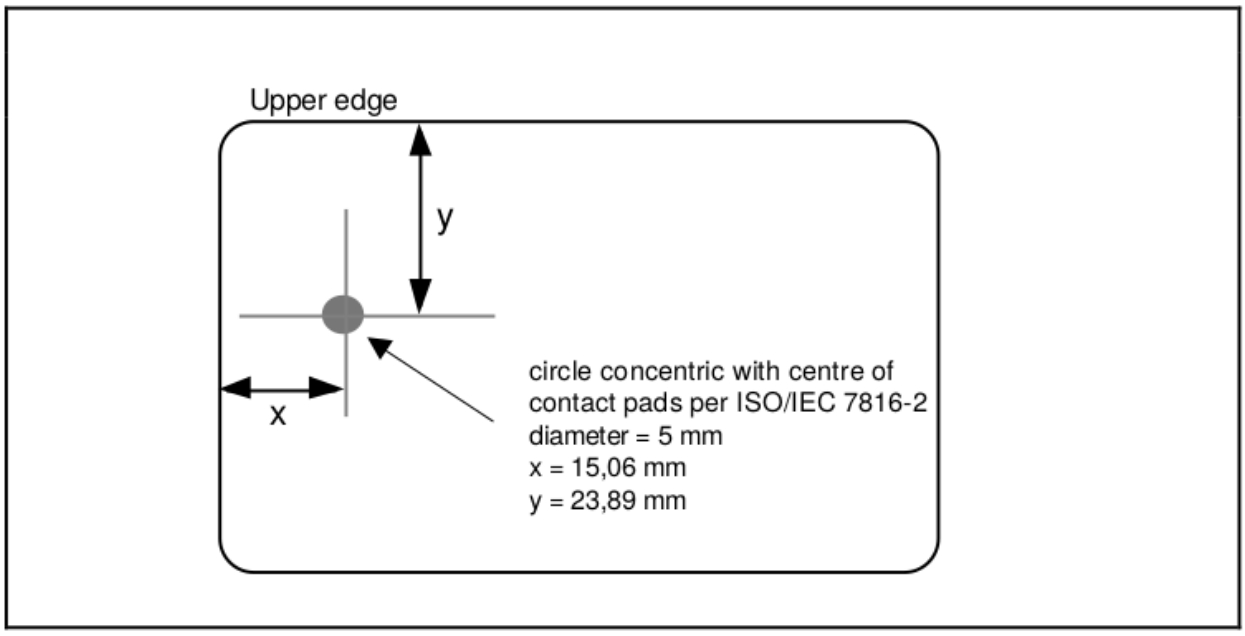
- **Data Rate** - The data bit rate for the transmission during initialization and anticollision shall be nominally fc/128 (106 kbit/s). Tolerance and bit boundaries are defined in ISO/IEC 14443-3.
- **Bit representation and coding** - Bit coding format shall be NRZ-L with logic levels defined as follows:
  logic "1" carrier high field amplitude (no modulation applied).
  logic "0" carrier low field amplitude.

Communication PICC to PCD:

- **Data Rate** - The data bit rate for the transmission during initialization and anticollision shall be fc/128 (106 kbit/s).
- **Bit representation and coding** - Bit coding shall be NRZ-L where a change of logic state shall be denoted by a phase shift (180^o) of the subcarrier.
  After any command from the PCD a guard time TR0 shall apply in which the PICC shall not generate a subcarrier. TR0 shall be greater than 64/fs. The PICC shall then generate a subcarrier with no phase transition before a delay TR1 establishing a subcarrier phase reference θ_0. TR1 shall be greater than 80/fs.Subsequently the logic state shall be defined according to the subcarrier phase reference:

θ_0	logic state 1
$\theta_0 + 180^o$	logic state 0

PICC minimal coupling zone
The PICC coupling antenna may have any form and location but shall encircle the zone shown in the following figure.



3.3.4 Polling

To detect PICC's which are present in the operating field, a PCD shall send repeated request commands. The PCD sends REQA(Request command Type-A), and REQB(request command Type-B) in any sequence and may send other additional commands.
When a PICC is exposed to an unmodulated operating field, it shall be able to accept a request within 5 ms.

3.3.5 Initialization and Anti-collision

Whenever at least two PICCs simultaneously transmit bit patterns with one or more bit positions having complementary values, PCD shaal be designed to detect the collision. Here, bit patterns merge and the carrier is modulated with the subcarrier for the whole bit duration(100%).

3.4 Contactless Smart Cards in Financial Transactions

Contactless payment systems are credit cards and debit cards, key fobs, smart cards, or other devices, including smartphones and other mobile devices, that use radio-frequency identification (RFID) or near field communication (NFC, e.g. Samsung Pay, Apple Pay, Google Pay) for making secure payments. The embedded chip and antenna enable consumers to wave their card, fob, or handheld device over a reader at the point of sale terminal. Contactless payments are made in close physical proximity, unlike mobile payments which use broad-area cellular or WiFi networks and do not involve close physical proximity.

3.5 Protection by Cryptographic Measures

RFID systems are increasingly being used in high-security applications, such as access systems and systems for making payments or issuing tickets. However, the use of RFID systems in these applications necessitates the use of security measures to protect against attempted attacks, in which people try to trick the RFID system in order to gain unauthorised access to buildings or avail themselves of services (tickets) without paying.High-security RFID systems must have a defence against the following individual attacks:

- skimming of a data carrier in order to clone and/or modify data;
- placing a foreign data carrier within the interrogation zone of a reader with the intention of gaining unauthorised access to a building or receiving services without payment;
- eavesdropping on radio communications.

Mutual Symmetrical Authentication
Mutual authentication between reader and transponder is based upon the principle of three-pass mutual authentication in accordance with ISO/IEC 9798-2 , in which both participants in the communication check the other party's knowledge of a secret cryptological key.

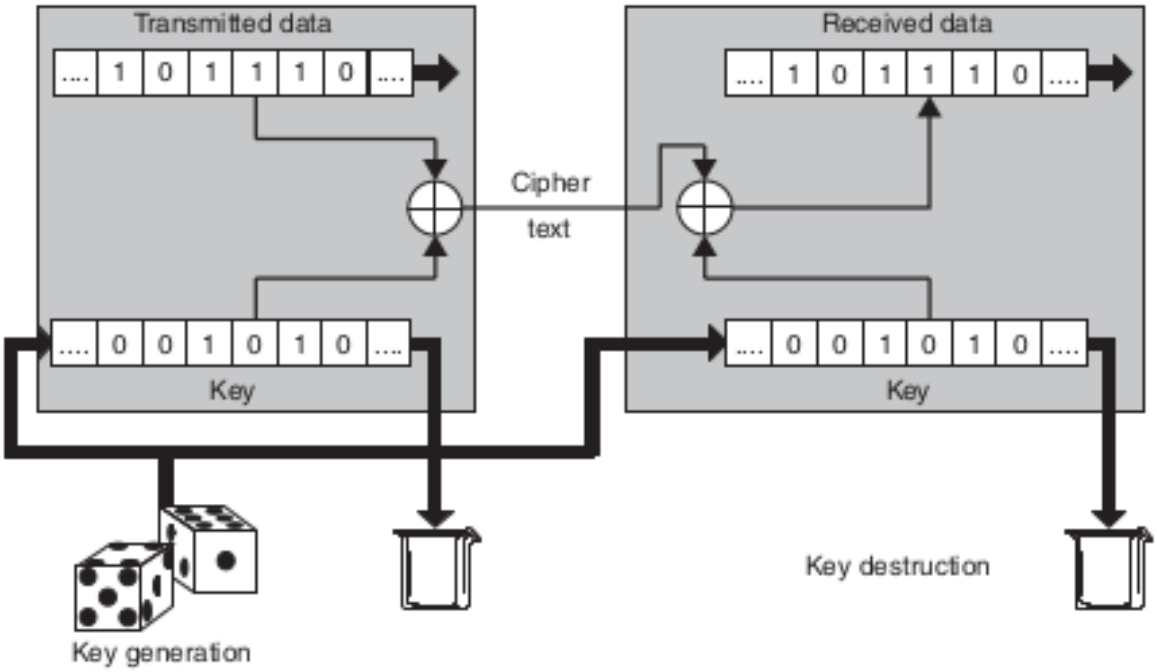
The mutual authentication procedure has the following advantages:

- The secret keys are never transmitted over the airwaves, only encrypted random numbers are transmitted.

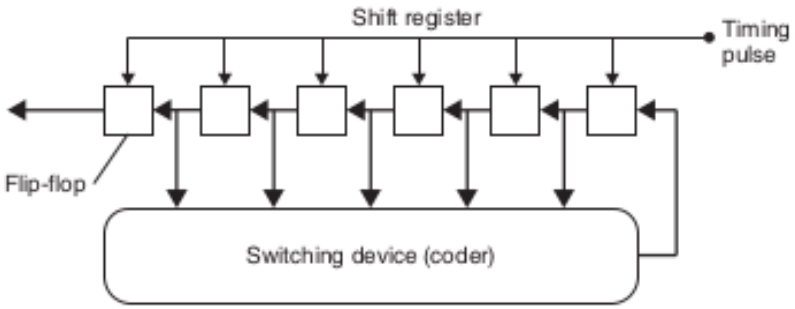
- The strict use of random numbers from two independent sources means that recording an authentication sequence for playback at a later date (replay attack) would fail.
- A random key (session key) can be calculated from the random numbers generated, in order to cryptologically secure the subsequent data transmission.

Stream Cipher

Sequential ciphers or stream ciphers are encryption algorithms in which the sequence of plain text characters is encrypted sequentially using a different function for every step.
Here, a random key is generated before transmission of encrypted data and it is made available to both parties. The random sequence must be as long as the message to be encrypted because periodic repeatitions of a short key would favour cryptanalysis and it creates an attack on transmission. The following stream cipher is impractical for RFID systems because of transmission of the key between transmitter and recipient.



To overcome the problem of key generation and distribution, systems have been created based upon the principle of the one-time pad stream cipher, that use a so-called pseudorandom sequence instead of an actual random sequence. Pseudorandom sequences are generated using so-called pseudorandom generators. An example of pseudorandom generator is the Linear Feedback Shift Register(LFSR).



The input of a LFSR is a linear function of its previous state. The most commonly used linear function of single bits is XOR. Thus, an LFSR is most often a shift register whose input bit is driven by the XOR of

some bits of the overall shift register value. The initial value of the LFSR is called the seed, and because the operation of the register is deterministic, the stream of values produced by the register is completely determined by its current (or previous) state. Likewise, because the register has a finite number of possible states, it must eventually enter a repeating cycle. However, an LFSR with a well-chosen feedback function can produce a sequence of bits that appears random and has a very long cycle.

3.6 Drawbacks

- Theft and Fraud - Contactless technology does not necessarily prevent use of a PIN for authentication of the user, but it is common for low value transactions (bank credit or debit card purchase, or public transport fare payment) not to require a PIN. This may make such cards more likely to be stolen, or used fraudulently by the finder of someone else's lost card.
- Multiple cards detection - When two or more contactless cards are in close proximity the system may have difficulty determining which card is intended to be used. The card-reader may charge the incorrect card or reject both.
- Privacy - Using a smart card for mass transit presents a risk for privacy, because such a system enables the mass transit operator, the banks, and the authorities, to track the movement of individuals.

Part II

Transaction Protocols

Chapter 4

Financial Transactions - ISO/IEC 8583

The standard ISO/IEC 8583 specifies how data is exchanged between systems during electronic transactions initiated by cardholders using payment cards. This includes all sorts of cards (magnetic stripe, contact chip and contactless) along with the PoS devices and ATM machines.
Major financial services companies like Mastercard and VISA base their communications on this standard. It is important to note that although the standard defines the fields used in communication, it is also adaptible to individual networks and custom usage by the service providers. For example, VISA uses BASE1 and BASE2, two protocols that define the messsage structure.
There have been two revisions to the original standard ISO/IEC 8583:1987, the ISO/IEC 8583:1993 and ISO/IEC 8583:2003. ISO/IEC 8583:1987 remains the most popular, with Mastercard and VISA using it even after all the current revisions.

4.1 Message Structure

The message structure defined in ISO/IEC 8583 is as follows.

This structure is often prepended by a network-specific header. It may be different for each network as per their requirements and is therefore not standardized.

4.1.1 Message Type Identifier (MTI)

The Message Type Identifier is 4 digits long, and describes the overall message class and function of the message. The four digits identify the following fields:

- The first digit indicates the version of ISO/IEC 8583 used in the message. The most common three configurations of the first digit are 0,1 and 2.
  - If 0, the encoding of the message is as in ISO/IEC 8583:1987.
  - If 1, the encoding of the message is as in ISO/IEC 8583:1993.
  - If 2, the encoding of the message is as in ISO/IEC 8583:2003.
- The second digit specifies the purpose of the message.
- The third digit specifies the message function, which later defines the flow of information within the communication channels.
- The fourth digit defines the source of the message.

Field definitions for the second digit are as follows:

Digit Value	Function	Explanation
x0xx	Use reserved by ISO	
x1xx	Authorization Message	Determine if funds are available, get an approval but do not post to account for reconciliation.
x2xx	Financial Messages	Determine if funds are available, get an approval and post directly to the account.
x3xx	File-action	Hot-card (one time cards) and TMS exchanges
x4xx	Reversal and Chargeback	Reverses the actions of a previous authorization
x5xx	Reconciliation	Transmits settlement information message
x6xx	Administrative	Administrative advice, often used for failure messages
x7xx	Fee Collection	
x8xx	Network Management	Used to manage network functions
x9xx	Use reserved by ISO	

Field definitions for the third digit are as follows:

Digit Value	Function	Explanation
xx0x	Request	A simple request to the receiver from the sender. Sender may accept/reject.
xx1x	Request Response	Response to the sender from the receiver of a request.
xx2x	Advice	A message that informs the receiver that an action has taken place.
xx3x	Advice Response	Response to the sender from the receiver of an advice message.
xx4x	Notification	A message that notifies the sender that an event has taken place.
xx5x	Notification Acknowledgement	Response to the sender from the receiver of a notification message.
xx6x	Instruction	Unique to `ISO/IEC 8583:2003`.
xx7x	Instruction Acknowledgement	Unique to `ISO/IEC 8583:2003`.
xx8x	Reserved for ISO use	
xx9x	Reserved for ISO use	

The exact definitions of each field is unique to the version of `ISO/IEC 8583` we are using.
Field definitions for the fourth digit are as follows:

Digit Value	Function
xxx0	Acquirer
xxx1	Acquirer Repeat
xxx2	Issuer
xxx3	Issuer Repeat
xxx4	Other
xxx5	Other Repeat
xxx6	Reserved for ISO use
xxx7	Reserved for ISO use
xxx8	Reserved for ISO use
xxx9	Reserved for ISO use

In case no response was received, it's usually possible to resend the identical message with the originator information set to 'Repeat', and thus we require the fields xxx1, xxx3 and xxx5.

4.1.2 Bitmap

The bitmap is a binary sequence that tells us if a particular data element is present in the message body or not. The data element is said to be present if the bit corresponding to that element (a pre-defined bit as per the contents of the data element) is set to 1.
A bitmap can indicate the presence of upto 192 field elements. The entire bitmap is divided into the sections depending upon the data element it can represent. The primary, secondary and tertiary bitmaps can represent 0-64, 65-128 and 129-192 elements (the numbers are the pre-defined index of the data elements). It is important to note that the secondary and tertiary bitmaps are optional, and their presence is indicated in the previous bitmap.
The bitmap can be represented in two forms, by a byte representation or hex-value representation.

4.1.3 Data Elements

Data elements are the actual data carrying fields, and contain data in a specified format. As the data is linearly written, it is important to consult the bitmap regarding which field element is written, which is then delimited by some value. Also, as not every field element may be of a fixed size, the length indicator is used. Along with the 'index' (more of a relative position with respect to the other fields) of the data element in

the message body, the size of the element is also defined. For example, if a field is defined as `n 4`, the field is a fixed 4 digits in length. Similarly, a definition `b 4` would mean 4 bits of data.
An `LLVAR` field element is of variable length, and can take values upto 100 digits. Similarly, `LLLVAR` is used to denote field value lengths of upto 1000 digits. For example, the Primary account number (PAN) is defined as a `19VAR` variable. This means that the PAN is of length upto 19 digits.
As mentioned earlier, there are upto 192 field elements. A few of the relevant ones are as follows.

Data Field	Type	Definition
1	`b 64`	Second Bitmap
2	`n ..19`	PAN
3	`n 6`	Processing Code
4	`n 12`	Amount, transaction
7	`n 10`	Transmission date and time
65	`b 1`	Extended bitmap indicator
64	`b 64`	Message Authentication Code (MAC)

There are a few fields that have been reserved for private use.

4.2 Issues

`ISO/IEC 8583` is a standard that defines the message format. However, multiple implementation details have not been standardized (to allow for the individual preferences of a provider to be integrated into the system). This leads to multiple interoperability issues.

- The encoding scheme for data (hex, ASCII, etc.) has not been mentioned. This may lead to misinterpretation of data.
- The usage of different versions of the same standard by different vendors could possibly lead to interoperability issues, especially when the message format is only based upon `ISO/IEC 8583` and is not a direct implementation of the standard. The changes in the ordering and other details could lead to miscommunication.

To prevent this from becoming an issue, we use interworking functions that act to convert the data from some form of data into a globally understood format. The conversion of the above mentioned format into the XML and then HTTP and other formats is often undertaken to allow interoperability. The IBM Integration Bus is an example of one such interworking functions.

Part III

Security Protocols

Chapter 5

SSL/TLS

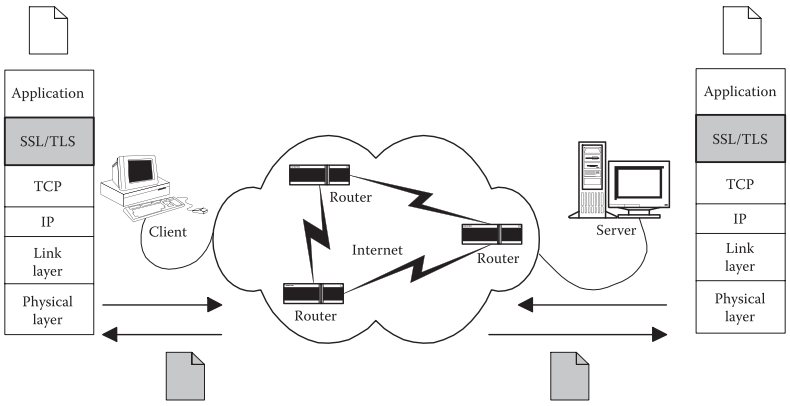
Multiple protocols were designed to secure bank card transactions initiated on open networks. TLS (Transport Layer Security) is the main focus of this report. SSL (Secure Socket Layer) is a predecessor of TLS and will be considered alongside TLS.

5.1 Transport Layer Security

Transport Layer Security, or TLS is a protocol used to secure any exchange between a client and server. It is a general protocol that is readily integrated into browsers and is not used solely for financial transactions.

5.1.1 Architecture of SSL/TLS

SSL/TLS sits between the Application and Transport layers in the OSI (Open Systems Interconnection) networking model of the TCP/IP protocol stack.



It is important to note that only those transport protocols that can offer reliable transmission of data (such as TCP) can make use of the advantages of SSL/TLS. This is because an unreliable transport protocol (like UDP) may cause a flow interruption, which may be misinterpreted as a security break by the TLS protocol, thus causing a termination of the open session.
The three goals of authentication, integrity and confidentiality are all served by TLS by combining all the functions that are needed to implement the three into a package called the cipher suite. The above-mentioned functions are decided upon during the 'handshake' phase of the transaction.
The handshake is the very first event to occur during a transaction. It happens after the TCP handshake, another procedure that creates the reliable channel over which further communication will take place. The purpose of the handshake is to:

- Specify the version of TLS. The most recent version of TLS is `TLS 1.3`.
- Decide on which cipher suite they will use.
- Authenticate the identity of the server via the server's public key and the certificate authority's digital signature.
- Generate session keys in order to use symmetric encryption after the handshake is complete.

The actual implementation of this handshake depends upon various factors. The most important factor to consider is the version of TLS, as the handshakes across the different versions of TLS (such as `TLS 1.3` and `TLS 1.2`) differ in many ways. The handshake procedure in `TLS 1.3` is faster as a result of a change in what is sent across in every message. The increase in speed is thus attributed to a decrease in the number of round trips. Also, in `TLS 1.3` the 0-RTT (zero round trip) mechanism allows for data to be sent on the first message to the server on sites the client has previously visited.

5.1.2 Authentication

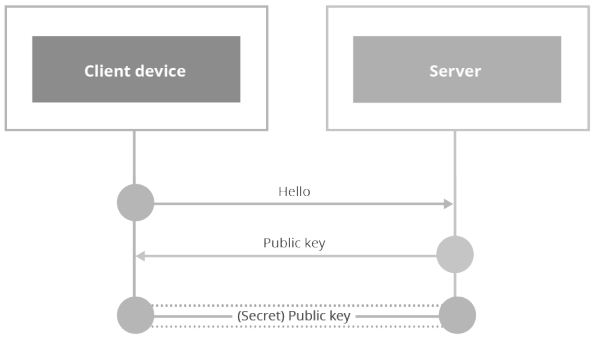
Authentication takes place only at the session establishment and before the first set of data has been transmitted, which makes it a part of the handshake process. Authentication uses a X.509 certificate to authenticate a client to the server, which may then accept or reject the session establishment.

Key Exchange

TLS is a hybrid cryptosystem. It uses both symmetric key cryptography (encryption and decryption keys are the same) to exchange actual data and public key cryptography (encryption and decryption keys are different) to exchange the key to be used to encrypt data.
Public key cryptography in TLS is usually done through RSA (`TLS 1.2`) or the Ephemeral Diffie-Hellman Algorithm (`TLS 1.3`).
In an RSA handshake (a handshake in which the key exchange mechanism is RSA) the following steps take place:

- The 'client hello' message: The client initiates the handshake by sending a "hello" message to the server. The message will include which TLS version the client supports, the cipher suites supported, and a string of random bytes known as the "client random."
- The 'server hello' message: In reply to the client hello message, the server sends a message containing the server's SSL certificate, the server's chosen cipher suite, and the "server random," another random string of bytes that's generated by the server.
- Authentication: The client verifies the server's SSL certificate with the certificate authority that issued it. This confirms that the server is who it says it is, and that the client is interacting with the actual owner of the domain.
- The premaster secret: The client sends one more random string of bytes, the "premaster secret." The premaster secret is encrypted with the public key and can only be decrypted with the private key by the server. (The client gets the public key from the server's SSL certificate.)
- Private key used: The server decrypts the premaster secret.
- Session keys created: Both client and server generate session keys from the client random, the server random, and the premaster secret. They should arrive at the same results.
- Client is ready: The client sends a "finished" message that is encrypted with a session key.
- Server is ready: The server sends a "finished" message encrypted with a session key.
- Secure symmetric encryption achieved: The handshake is completed, and communication continues using the session keys.

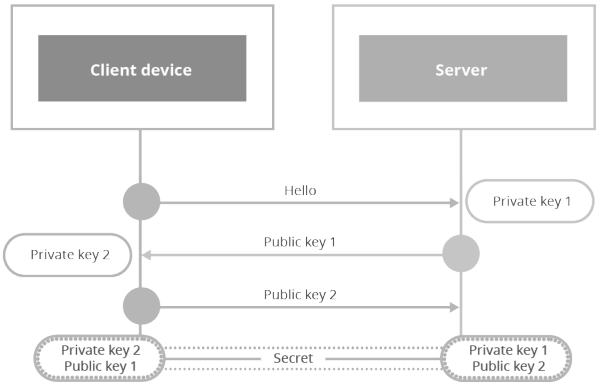
A summary of the above:



An ephemeral Diffie-Hellman handshake is slightly different from the above procedure. The Diffie-Hellman Parameter (DH parameter) is used to establish the secret symmetric key separately by both entities instead of the client sending the symmetric key encrypted with the server's public key. The procedure is as follows:

- Client hello: The client sends a client hello message with the protocol version, the client random, and a list of cipher suites.
- Server hello: The server replies with its SSL certificate, its selected cipher suite, and the server random.
- Server's digital signature: The server uses its private key to encrypt the client random, the server random, and its DH parameter. This encrypted data functions as the server's digital signature, establishing that the server has the private key that matches with the public key from the SSL certificate.
- Digital signature confirmed: The client decrypts the server's digital signature with the public key, verifying that the server controls the private key and is who it says it is. The client sends its DH parameter to the server.
- Client and server calculate the premaster secret: Instead of the client generating the premaster secret and sending it to the server, as in an RSA handshake, the client and server use the DH parameters they exchanged to calculate a matching premaster secret separately.
- Session keys created: Now, the client and server calculate session keys from the premaster secret, client random, and server random, just like in an RSA handshake.
- Client is ready.
- Server is ready.
- Secure symmetric encryption achieved.

A summary of the above:



This is a better way of sending across information, as the key pairs used on either side are destroyed right after the handshake. This is called *perfect forward secrecy*, and the reason that RSA is not used in `TLS 1.3` is that it cannot offer perfect forward secrecy.

5.1.3 Confidentiality

Message confidentiality is based on the utilization of the symmetric encryption algorithms, whether stream encryption or block encryption. The same algorithm is used on both sides, but each side uses its own key, sharing it with the other party.
Note that `TLS 1.2` allows stream ciphers, although `TLS 1.3` doesn't.

5.1.4 Integrity

The integrity of the data is assured with hash functions that employ the HMAC procedure.

Part IV

References

References

1. ISO/IEC
   7811-2: Magnetic stripe - Low coercivity
   7816-2: Cards with contacts - Dimensions and location of the contacts
   7816-3: Electrical interface and transmission protocols
   7816-4: Organization security and commands for interchange
   14443-1: Contactless cards - Physical characteristics
   14443-2: Contactless cards - Radio frequency power and signal interface
   14443-3: Contactless cards - Initialization and anticollision
   14443-4: Contactless cards - Transmission protocol

2. Mostafa Hashem Sherif
   Protocols for Secure Electronic Commerce

3. ISO8583 - A layman's guide to understanding the ISO8583 Financial Transaction Message

4. *Magtek*
   MAGNETIC STRIPE CARD STANDARDS

5. *Medium*
   ISO 8583. An Introduction. Plain and Simple.

6. *Wikipedia*
   ISO 8583

7. *CloudFlare*
   A detailed look at RFC 8446 (a.k.a TLS 1.3)

8. *Kinsta*
   An overview of TLS 1.3

9. Wolfgang Rankl and Wolfgang Effing
   The Smart Card Handbook

10. Klaus Finkenzeller
    RFID Handbook - Fundamentals and Applications in Contactless Smart Cards, Radio Frequency Identification and Near-Field Communication