



FeliCa Lite-S

FeliCa Lite-S

Starter's Manual

Version 1.11
No. M740-E01-11

- FeliCa is a contactless IC card technology developed by Sony Corporation.
- FeliCa is a trademark of Sony Corporation.
- All names of companies and products contained herein are trademarks or registered trademarks of the respective companies.
- No part of this document may be copied, or reproduced in any form, without the prior consent of Sony.
- Information in this document is subject to change without notice.
- Sony assumes no liability for damages arising from, or in connection with, the use of this document.

Introduction

This document provides an overview for customers who are using (or intend to use) FeliCa Lite-S products for the first time.

Descriptions in this document apply to the FeliCa Lite-S-based contactless IC cards and IC chips sold by Sony.

For the differences between the FeliCa Lite-S documents and FeliCa Lite documents, see "Documentation Differences Between FeliCa Lite and FeliCa Lite-S".

If you have any questions about the development of application software that is compatible with mobile FeliCa cards, please contact FeliCa Networks, Inc. (info-fn@FeliCaNetworks.co.jp).

The content of this document does not guarantee the correct operation of the system with all existing or future FeliCa Lite-S products.

The following notational conventions apply in this document, unless otherwise specified:

- Numerical values are expressed in decimal notation.
- Hexadecimal values are identified by "h" (meaning hexadecimal) appended to the value.
- Binary values are identified by "b" (meaning binary) appended to the value.
- The Byte order is Big Endian, unless otherwise specified.

In this document, a FeliCa Lite-S series product is expressed as "FeliCa Lite-S", and a FeliCa Lite-S card is expressed as "card". This document does not apply to FeliCa Standard series, FeliCa Lite series, and FeliCa Plug series products.

FeliCa technology refers to the following standards:

- JIS X 6319-4: Specification of implementation for integrated circuit(s) cards – Part 4: High speed proximity cards
- ISO/IEC 18092: Information technology – Telecommunications and information exchange between systems – Near Field Communication – Interface and Protocol-1 (NFCIP-1)
- NFC Forum: <http://www.nfc-forum.org/>

Contents

1	What is FeliCa Lite-S?	5
1.1	Product concept	5
1.2	Main specifications	7
2	Technical information about FeliCa Lite-S	9
3	Data Format Code (DFC)	10
3.1	What is DFC?	10
3.2	Using DFC	11
3.3	Data format distinction sequence with DFC	11
4	Issuance	14
4.1	Issuance of FeliCa Lite-S card	14
4.2	1 st issuance and 2 nd issuance	15
5	Functions and usage examples	17
5.1	How to use functions	17
5.2	Usage examples	18
5.2.1	Number control tag	18
5.2.2	NFC-compliant smart poster	19
5.2.3	Coupon card	20
5.2.4	Employee ID and student ID cards	21
5.2.5	Points card	23
6	Cautions for correct use	29
6.1	Distinction of FeliCa Lite-S	29
6.2	Distinction of data format	31
6.3	Maximum response time	32
6.4	Command Packet Data	33
7	Frequently asked questions about FeliCa Lite-S	35
7.1	Uniqueness of IDm	35
7.2	Setting of card key	36
7.3	Authentication	37
7.3.1	Internal Authentication (unilateral authentication)	37
7.3.2	External Authentication	39
7.3.3	Mutual Authentication	41
7.4	Differences from FeliCa Standard	43
7.5	Reader/Writer used for FeliCa Standard	44
7.6	Using FeliCa Standard, FeliCa Lite, and FeliCa Lite-S cards together	45
7.6.1	Using FeliCa Standard and FeliCa Lite-S cards together	45
7.6.2	Using FeliCa Lite and FeliCa Lite-S together	46
7.7	Anti-tearing	47

1 What is FeliCa Lite-S?

1.1 Product concept

FeliCa Lite-S has the following features:

Lower price

With simplified security functionality, an optimized file system and less memory, FeliCa Lite-S products are offered at lower prices than conventional FeliCa chips. FeliCa Lite-S products are suited for embedding in such low-cost products as sales promotion articles, membership cards, points cards, game cards, gift cards, and multi-use tickets.

Use of existing FeliCa infrastructure

FeliCa Lite-S products use the same wireless communication method as conventional FeliCa-compatible products and a command system compatible with that for conventional FeliCa chips. Therefore, you can use your existing infrastructure, including FeliCa ports, and development environment without change.

You can issue and operate FeliCa Lite-S products with only three commands: Polling, Read Without Encryption, and Write Without Encryption.

Smaller size and lower power consumption

FeliCa Lite-S products are the custom chips of a power-saving design, so they can operate even with a very low-strength electromagnetic field. You can use FeliCa Lite-S products for stickers and other non-card media that have miniaturized antennas.

Mutual Authentication function

FeliCa Lite-S products have a function that differs from that in conventional FeliCa chips (FeliCa Standard), for Mutual Authentication with the Reader/Writer. You can use FeliCa Lite-S products for ID cards, points cards, and so on.

Modification detection

FeliCa Lite-S products enable modification detection by attaching MAC (Message Authentication Code) to data to be read or written.

Compliance with NFC Forum Type 3 Tag

FeliCa Lite-S products comply with the Type 3 Tag defined by the NFC Forum. Therefore, they can communicate with any Reader/Writer that is compliant with the NFC Forum standard. You can use FeliCa Lite-S products for handover and smart posters.

FeliCa Lite-S Starter's Manual

Compared with the FeliCa Standard products used for electronic money and transportation tickets, FeliCa Lite-S products are designed to have less complex security functionality, less memory and, therefore, lower prices. Figure 1-1 shows how FeliCa Lite-S is applied to the markets, compared to FeliCa Standard.

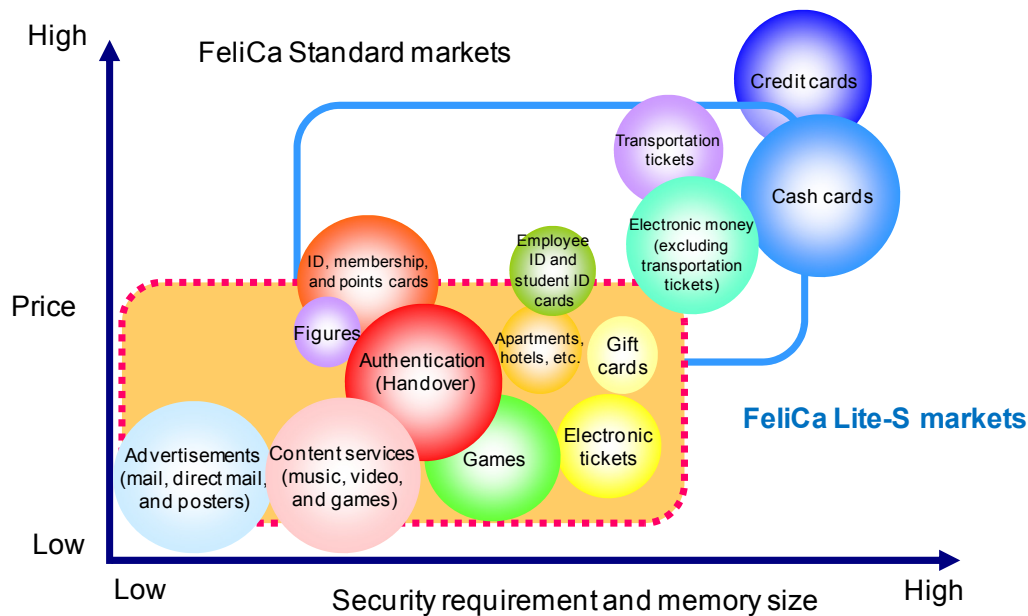


Figure 1-1: FeliCa Lite-S markets

1.2 Main specifications

This section describes the main specifications of FeliCa Lite-S. Table 1-1 describes each Blocks of the FeliCa Lite-S file system.

- The Manufacture ID (IDm) of FeliCa Lite-S is a unique 8-Byte value.
- The Polling command, Read Without Encryption command, and Write Without Encryption command are supported.
- Anti-broken transaction (anti-tearing mechanism) is supported.
- A fixed file format is used.
- Sony assigns each service provider a unique 2-Byte value as the Data Format Code (DFC).
- MAC_A can be attached to Block Data to be read or written, to detect data modification. (Block Number modification can be also detected.)
- Mutual Authentication is enabled by reading ID Block with MAC_A Block and writing STATE Block with MAC_A Block.
- Each User Block can be specified separately with the following attributes:
 - Read-Only or Read-Write
 - Authentication required or not required for reading (Read After Authentication)
 - Authentication required or not required for writing (Write After Authentication)
 - MAC required or not required for writing (Write With MAC)
- Communication channel encryption is not supported.

Table 1-1: Overview of each Block in the FeliCa Lite-S file system

Name	Block description (1 Block = 16 Bytes)	Remarks
S_PAD0 – S_PAD13	Block in which values can be rewritten arbitrarily (which can be used freely by the user)	User Block
REG	Block for subtraction (in which only a value the same as or less than the current value can be written for use in multi-use tickets and so on)	
RC	Block to write a random number (to be used by authentication functions)	Authentication Function Block
MAC	Block to read the result of the MAC calculation (to be used for authentication and modification detection) ^{*1}	
ID	Block to store the ID of the card issuer (a DFC and arbitrary 6-Byte data are set)	System Block
D_ID	Block to store the chip ID (unique value, which is set at shipment and cannot be changed)	
SER_C	Block to store the base code of Service Code ("0000h" is set and Service Code "0009h" or "000Bh" shall be used)	
SYS_C	Block to store System Code ("88B4h" is set at shipment and the value cannot be rewritten)	
CKV	Block to store the card Key version (to be set during card issuance when the authentication function is used)	
CK	Block to store the card key (to be set during issuance when the authentication function is used; the Read value is always all-zero, regardless of the key value)	
MC	Block to be used for 1st issuance, 2nd issuance, and settings to store NDEF data-storage settings	
WCNT	Block for the write counter to prevent replay attack for Write With MAC	Authentication Function Block
MAC_A	Block to read or write the result of MAC calculation (to be used for authentication and modification detection) ^{*2}	
STATE	Block to set the state of authentication and Polling Disable	Inspection Block
CRC_CHECK	Block to check CRC collectively	

^{*1, *2} Recommends to use MAC_A block whose security functions are enhanced.

2 Technical information about FeliCa Lite-S

You can find technical information about FeliCa Lite-S at the following website:

<http://www.sony.net/Products/felica/business/tech-support/index.html>

When you use FeliCa Lite-S, see to the following documents (to be issued in series):

Documents common to FeliCa

- "FeliCa Technology Code Descriptions"
- "Format Distinction Sequence Design Guidelines"

Documents specific to FeliCa Lite-S

- "FeliCa Lite-S Starter's Manual" (this document)
- "FeliCa Lite-S User's Manual"
- "FeliCa Lite-S Technical Note for Software Development"
- "FeliCa Lite-S Security Application Note"
- "FeliCa Lite-S Diversified Card Key Standard Generation Algorithm"

The FeliCa Lite-S Starter's Manual (this document) is intended for first-time users of FeliCa Lite-S products; it includes an overview of FeliCa Lite-S, the main points of the other reference documents, and frequently-asked questions about FeliCa Lite-S.

3 Data Format Code (DFC)

3.1 What is DFC?

[Reference technical document]

“FeliCa Technology Code Descriptions”
Chapter 4 “Data Format Code”

What is DFC?

DFC is the value of the 2 Bytes that identify the format of the data to be stored on the card.
For the FeliCa Lite-S series, DFC is the 9th and 10th Bytes (Byte 8 and Byte 9) of ID Block. (When its value is not used, DFC is 0000h.)

What is distinction of the data format?

FeliCa Lite-S cards are used for various purposes. For example, given that the S_PADO value of a card is 100, the value is assumed to indicate either 100 points if the card is used as a points card, or a membership number of 100 if the card is used as a membership card.

When the data format of a card is distinguished, the meaning of the data written on the card is specified. DFC is a 2-Byte value that distinguishes the data format of a card. When multiple cards with different data formats are presented to a Reader/Writer, the Reader/Writer first distinguishes their data formats by checking their DFCs.

3.2 Using DFC

[Reference technical document]

"FeliCa Lite-S User's Manual"

Chapter 3 "File system"

Chapter 7 "Issuance"

Section A.1 "Data Format Code (DFC)"

When using DFC

To acquire DFC, contact Sony.

Write the acquired DFC to Byte 8 and Byte 9 of ID Block (ID[8] [9]).

<How to apply the DFC>

DFC is reserved and granted by Sony.

For details of the application procedure, please contact your FeliCa Lite-S sales agent.

When not using DFC

You do not need to contact Sony.

Write 0000h to the Byte 8 and Byte 9 of ID Block (ID[8] [9]).

ID (Block Number: 82h)	<div> <div>0 1 2 3 4 5 6 7 8 9 10 11 12 13 14 15</div> <div> <div>[0][1][2][3][4][5][6][7][8][9][10][11][12][13][14][15]</div> <div> <div>IDd</div> <div>DFC</div> <div>Arbitrary value</div> </div> </div> </div>
D_ID (Block Number: 83h)	<div> <div>0 1 2 3 4 5 6 7 8 9 10 11 12 13 14 15</div> <div> <div>[0][1][2][3][4][5][6][7][8][9][10][11][12][13][14][15]</div> <div> <div>IDd</div> <div>PMm</div> </div> </div> </div>
Response to the Polling command	<div> <div>0 1 2 3 4 5 6 7 8 9 10 11 12 13 14 15</div> <div> <div>[0][1][2][3][4][5][6][7][8][9][10][11][12][13][14][15]</div> <div> <div>IDd</div> <div>PMm</div> </div> </div> </div>

Figure 3-1: DFC operation rules

3.3 Data format distinction sequence with DFC

Figure 3-2 shows an example of the processing sequence for data format distinction using DFC. Figure 3-3 shows the process flow between the Reader/Writer and the card for the data format distinction.

Firstly, the Reader/Writer issues the Polling command, specifying System Code (88B4h) of the FeliCa Lite-S card to capture the card. Upon receiving a response, the Reader/Writer confirms that the FeliCa Lite-S card exists. Then, the Reader/Writer reads ID Block of the FeliCa Lite-S card. When the value in the

Byte 9 and Byte 10 of ID Block (ID[8][9]) matches the desired DFC, the Reader/Writer processes the target FeliCa Lite-S card.

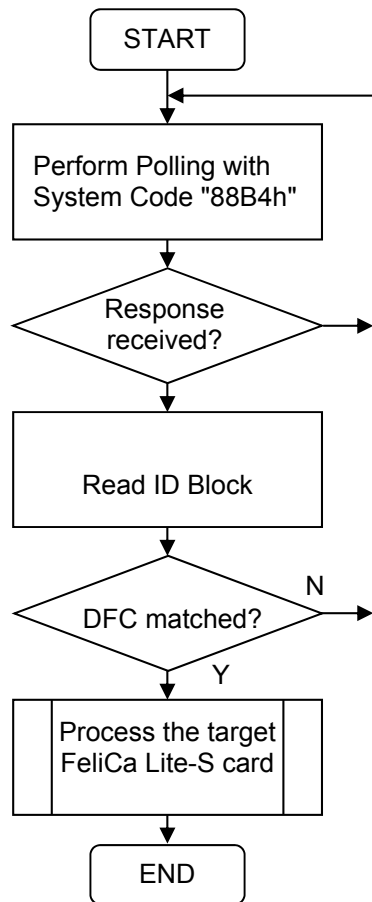


Figure 3-2: Data format distinction sequence with DFC

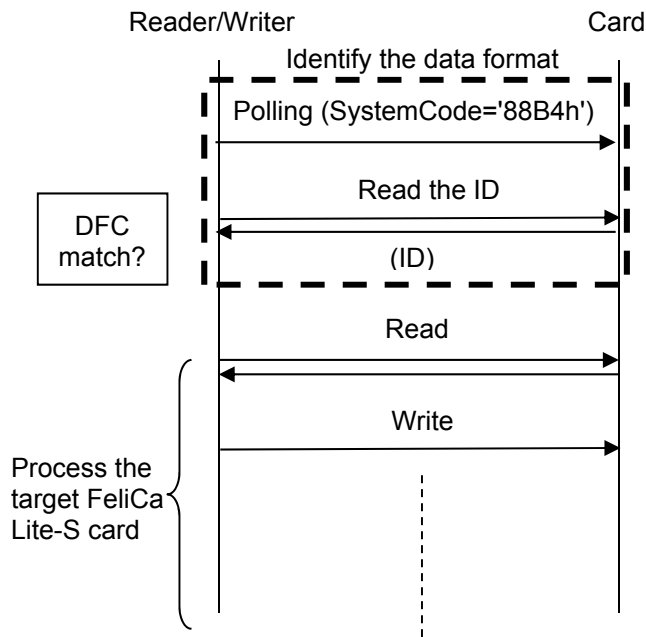


Figure 3-3: Example of the processing sequence for data format distinction with DFC

4 Issuance

4.1 Issuance of FeliCa Lite-S card

[Reference technical document]

“FeliCa Lite-S User’s Manual”

Chapter 3 “File system”

Chapter 7 “Issuance”

What is issuance of FeliCa Lite-S card?

To issue a FeliCa Lite-S card, set both System Block and User Block.

You can perform all issuance steps with only three commands: Polling, Read Without Encryption, and Write Without Encryption. There is no special command for issuance. You do not need to prepare a special Reader/Writer for issuance.

You do not need to design a file format because FeliCa Lite-S uses a fixed file format. You shall, however, decide how to use User Block (i.e., decide which data is to be located in which User Block).

The issuing of a FeliCa Lite-S card is the writing (setting) of values in System Block and User Block, using the Write Without Encryption command.

How does this differ from the issuance of FeliCa Standard card?

For FeliCa Lite-S card:

- There is no special command for issuance.
- You do not need to prepare a special Reader/Writer for issuance.
- You do not need to design the file format.
- You do not need to set IDm, PMm, or System Code.
- You need to set DFC.

4.2 1st issuance and 2nd issuance

1st issuance and 2nd issuance procedures for FeliCa Lite-S Card

Each FeliCa Lite-S chip is shipped from the factory after IDd (IDm), PMm, and System Code have been set in the chip.

The 1st issuance procedure is to set the data in System Block. To disable the settings (i.e., to prevent them from being changed), write 00h to MC [2], and then shut off the power to the chip. Subsequently, the value of MC [2] cannot be changed back to FFh.

The 2nd issuance procedure is to set the data in User Block. To make User Block in which data is set Read Only, write 0 to the bits 0 to 7 of MC [0] and the bits 0 to 6 of MC [1], which correspond to the access permissions of S_PAD0 to S_PAD13 and REG, respectively, and then shut off the power to the chip. After any User Block is made Read Only, it cannot be reverted to read/write Block.

Similarly, you can set the bits of MC [6] and MC [7] to specify whether authentication is necessary for reading, the bits of MC [8] and MC [9] to specify whether authentication is necessary for writing, and the bits of MC [10] and [11] to specify whether MAC is necessary for writing.

To prevent the access permissions set for each User Block from being changed, write 0 to the bit 7 of MC [1], and then shut off the power to the chip. Subsequently, the value of the bit 7 of MC [1] cannot be reverted to 1.

Items to be set in the 1st issuance and 2nd issuance procedures

Table 4-1 lists the steps in the process to set each Block.

Before completion of the 2nd issuance procedure, disable the rewriting of all System Block and read only User Block.

Table 4-1: Block to be set in 1st issuance and 2nd issuance procedures

Block	Block description	Category	Before chip shipment	1 st issuance	2 nd issuance
S_PAD0 – S_PAD13	Block that can be used freely by the user (Values can be rewritten arbitrarily.)	User Block	–	–	Setting of user data
REG	Block with a limitation on writing		–	–	Setting of user data
RC	Block to write the random number for authentication	Authentication Function Block	–	–	–
MAC	Block to read the result of the MAC calculation		–	–	–
ID	Block to store the ID of the issuer	System Block	–	Setting	–
D_ID	Block to store chip-specific values		Setting of IDd and PMm	–	–
SER_C	Block to store Service Code		–	Setting of fixed value	–

Block		Block description	Category	Before chip shipment	1 st issuance	2 nd issuance
SYS_C		Block to store System Code		Setting	–	–
CKV		Block to store Card Key Version		–	Setting	–
CK		Block to store Card Key		–	Setting	–
MC	[0][1]	Bytes to set the access permission for User Block		–	–	Setting
	[2]	Byte to set the access permission for System Block other than SYS_C and MC [0][1] [6] - [12] ^{*1} Reset of WCNT		–	Setting	–
	[3]	Byte for the setting to store NDEF data		–	Setting	–
	[4]	RF parameter		–	Setting	–
	[5]	Byte for the setting to write CK and CKV with MAC		–	Setting	–
	[6][7]	Bytes for the setting to read User Block after authentication		–	–	Setting
	[8][9]	Bytes for the setting to write User Block after authentication		–	–	Setting
	[10][11]	Bytes for the setting to write User Block with MAC		–	–	Setting
	[12]	Byte for the setting to write STATE with MAC		–	Setting	–
WCNT		Block for nonce of the card for authentication	Authentication Function Block	–	–	–
MAC_A		Block to read and write MAC		–	–	–
STATE		Block for authentication and disabling polling		–	–	–
CRC_CHECK		Block to check CRC collectively	Inspection Block	–	–	–

^{*1} Data can be set in MC [0][1][6] to [12] even after MC [2] is locked.

5 Functions and usage examples

5.1 How to use functions ---

FeliCa Lite-S provides the functions to set User Block access permission and to generate MACs, as follows:

- Function for setting the User Block access permission
Use this function to set the access permission for User Block to Read-Only (RO) or Read-Write (RW).
- Function for MAC-generation function
Use this function to perform Read With MAC or Write With MAC, to detect data modification.

Table 5-1 shows how to use these two functions and includes examples of FeliCa Lite-S usage. The usage examples marked with an asterisk (*) in Table 5-1 are described in section 5.2 "Usage examples". Refer to them while designing the flows for issuance and processing.

Table 5-1: How to use functions and usage examples

	Not using MAC_A generation	Using MAC_A generation
All User Block is RO.	Simplified ID card Number control tag* NFC-compliant smart poster* Handover	ID card Employee ID or student ID card* Membership card
Some of User Block is RW.	Coupon card*	Points card* Gift card

5.2 Usage examples

Use the examples described in this section as reference while designing the flows of issuance and processing.

5.2.1 Number control tag

Summary

- Set the serial number in S_PAD0.
- Set each unused User Block to RO without changing their default values (ALL_00h).

Example of issuance

- Write the serial number to S_PAD0.
- Write ALL_00h to S_PAD1 to S_PAD13, and ALL_FFh to REG.
- Set each Block to RO.

S_PAD0	RO	Serial number		
S_PAD1	RO	00_00_00_00_..._00		
S_PAD13	RO	00_00_00_00_..._00		
REG	RO	FF_FF_FF_FF_..._FF		
ID	RO	IDd (8)	DFC (2)	Arbitrary value (6)
CK	RO	00_00_00_00_..._00		

Figure 5-1: Example data format of a number control tag

Example flow of processing

Figure 5-2 shows an example flow of processing. For the distinction of data format, see Figure 3-3.

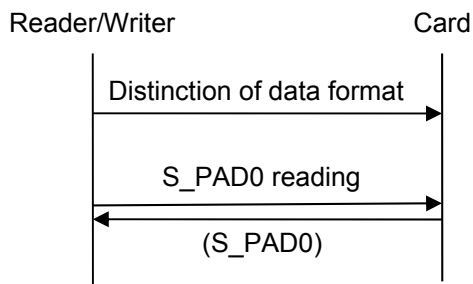


Figure 5-2: Example flow of processing for a number control tag

5.2.2 NFC-compliant smart poster

Summary

- Set Attribute Information Block and Smart Poster RTD in S_PAD0 to S_PADn (the value of n depends on the length of URL).
- Set each unused User Block to RO without changing their default values (ALL_00h).
- Set 01h in MC [3] so the chip responds to the Polling command with System Code 12FCh.

Example of issuance

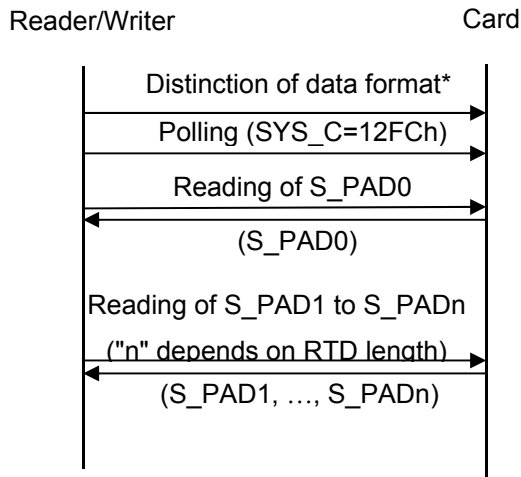
- Write Attribute Information Block code to S_PAD0.
- Write Smart Poster RTD (URL) to S_PAD1 to S_PADn (the value of n depends on the length of URL).
- Write ALL_00h to the remaining S_PAD Block, and ALL_FFh to REG.
- Write IDd (8 Bytes), DFC (2 Bytes), and an arbitrary value (6 Bytes) specified by the issuer to ID.
- Set each Block to RO.
- Set 01h in MC [3].

S_PAD0	RO	Attribute Information Block		
S_PAD1	RO	Smart Poster RTD		
S_PADn	RO	Smart Poster RTD		
S_PAD13	RO	00_00_00_00_..._00		
REG	RO	FF_FF_FF_FF_..._FF		
ID	RO	IDd (8)	DFC (2)	Arbitrary value (6)
CK	RO	00_00_00_00_..._00		

Figure 5-3: Example data format of an NFC-compliant smart poster

Example flow of processing

Figure 5-4 shows an example flow of processing. For the distinction of data format, see Figure 3-3.



* When the card responds to the Polling command (SYS_C = 12FCh), the card uses the NDEF format. Therefore, the DFC check can be omitted.

Figure 5-4: Example flow of processing for a NFC-compliant smart poster

5.2.3 Coupon card

Summary

- Write a coupon code to S_PAD0.
- Set each unused User Block to RO without changing their default values (ALL_00h).

Example of issuance

- Write a coupon code to S_PAD0.
- Write ALL_00h to S_PAD1 to S_PAD13, and ALL_FFh to REG.
- Write IDd (8 Bytes), DFC (2 Bytes), and an arbitrary value (6 Bytes) specified by the issuer to ID.
- Set S_PAD0 Block to RW, and each other User Block to RO.

S_PAD0	RW	Coupon code		
S_PAD1	RO	00_00_00_00_..._00		
S_PAD13	RO	00_00_00_00_..._00		
REG	RO	FF_FF_FF_FF_..._FF		
ID	RO	IDd (8)	DFC (2)	Arbitrary value (6)
CK	RO	00_00_00_00_..._00		

Figure 5-5: Example data format of a coupon card

Example flow of processing

Figure 5-6 shows an example flow of processing. For the distinction of data format, see Figure 3-3.

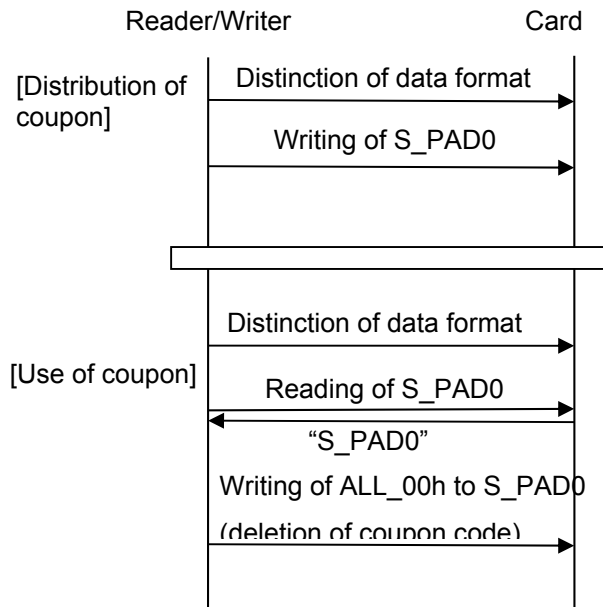


Figure 5-6: Example flow of processing for a coupon card

5.2.4 Employee ID and student ID cards

[Reference technical document]

"FeliCa Lite-S Diversified Card Key Standard Generation Algorithm"

"FeliCa Lite-S Security Application Note"

Section 5.4.2 "Integrity"

Here, an employee number is taken as an example for explanation.

Summary

- Set an employee number in S_PAD0.
- Set each unused User Block to RO without changing their default values (ALL_00h).
- Set the value of ID and Diversified Card Key.

Example of issuance

- Write an employee number to S_PAD0.
- Write ALL_00h to S_PAD1 to S_PAD13, and ALL_FFh to REG.
- Write IDd (8 Bytes), DFC (2 Bytes), and an arbitrary value (6 Bytes) specified by the issuer to ID.
- Write a card key, which is diversified by the ID value and unique to each card, to CK. (For details of how to generate the key, see the "FeliCa Lite-S Diversified Card Key Standard Generation Algorithm".)

- Set each Block to RO.

S_PAD0	RO	Employee number		
S_PAD1	RO	00_00_00_00_..._00		
S_PAD13	RO	00_00_00_00_..._00		
REG	RO	FF_FF_FF_FF_..._FF		
ID	RO	IDd (8)	DFC (2)	Arbitrary value (6)
CK	RO	Key unique to each card		

Figure 5-7: Example data format of an employee ID card

Example flow of processing

Figure 5-8 shows an example flow of processing. For the distinction of data format, see Figure 3-3.

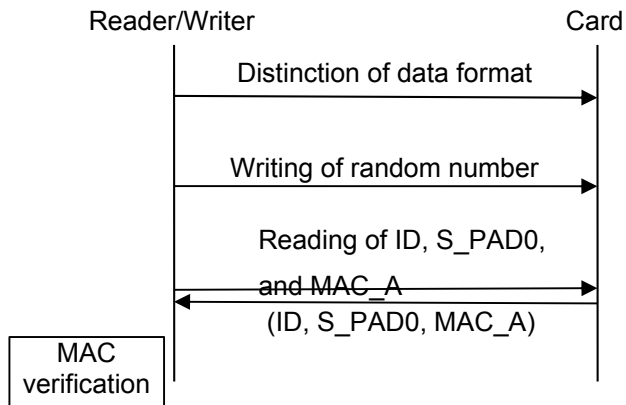


Figure 5-8: Example flow of processing for an employee ID card

5.2.5 Points card

Block to store the number of points shall be set to RW so it can be rewritten in operation. Note, however, that only the Reader/Writer that has access permission shall be allowed to rewrite Block.

FeliCa Lite-S implements points cards using Mutual Authentication. Using Write With MAC, FeliCa Lite-S can also detect modification on the communication channel.

Points cards can also be supported, using signature in the same way that FeliCa Lite uses.

5.2.5.1 Using Mutual Authentication

FeliCa Lite-S uses the Mutual Authentication function to allow only the Reader/Writer that has permission to rewrite the data.

[Reference technical document]

“FeliCa Lite-S Diversified Card Key Standard Generation Algorithm”

“FeliCa Lite-S Security Application Note”

Section 5.4.2 “Integrity”

Summary

- Write the card number to S_PAD0, the log to S_PAD1 and S_PAD2, and the number of points to S_PAD3.
- Set each unused User Block to RO without changing their default values (ALL_00h).
- Set the log to the Read After Authentication attribute and the Write After Authentication attribute.
- Set the number of points to the Write After Authentication attribute (authentication not required for reading).
- Set the value of ID and Diversified Card Key.
- MAC_A is computed by the Reader/Writer when the number of points is written.

Example of issuance

- Write the card number to S_PAD0.
- Write the log to S_PAD1 and S_PAD2.
- Write the number of points to S_PAD3.
- Write ALL_00h to S_PAD4 to S_PAD13, and ALL_FFh to REG.
- Write IDd (8 Bytes), DFC (2 Bytes), and an arbitrary value (6 Bytes) specified by the issuer to ID.
- Write a card key, which is diversified by the ID value and unique to each card, to CK. (For details of how to generate the key, see the “FeliCa Lite-S Diversified Card Key Standard Generation Algorithm”.)
- Set the S_PAD1 to S_PAD3 to RW, and each other User Block to RO.
- Set the S_PAD1 to S_PAD2 to the Read After Authentication attribute and the Write After Authentication attribute, and S_PAD3 to the Write After Authentication attribute.

NOTE For details of how to assure the concurrent writing of the log and number of points, see section 7.7 “Anti-tearing”.

S_PAD0	RO	Card number		
S_PAD1	RW	Side A of log		
S_PAD2	RW	Side B of log		
S_PAD3	RW	Number of points		
S_PAD4	RO	00_00_00_00_..._00		
S_PAD13	RO	00_00_00_00_..._00		
REG	RO	FF_FF_FF_FF_..._FF		
ID	RO	IDd (8)	DFC(2)	Arbitrary value (6)
CK	RO	Key unique to each card		

Figure 5-9: Example data format of a points card

Example flow of processing

For the distinction of data format, see Figure 3-3.

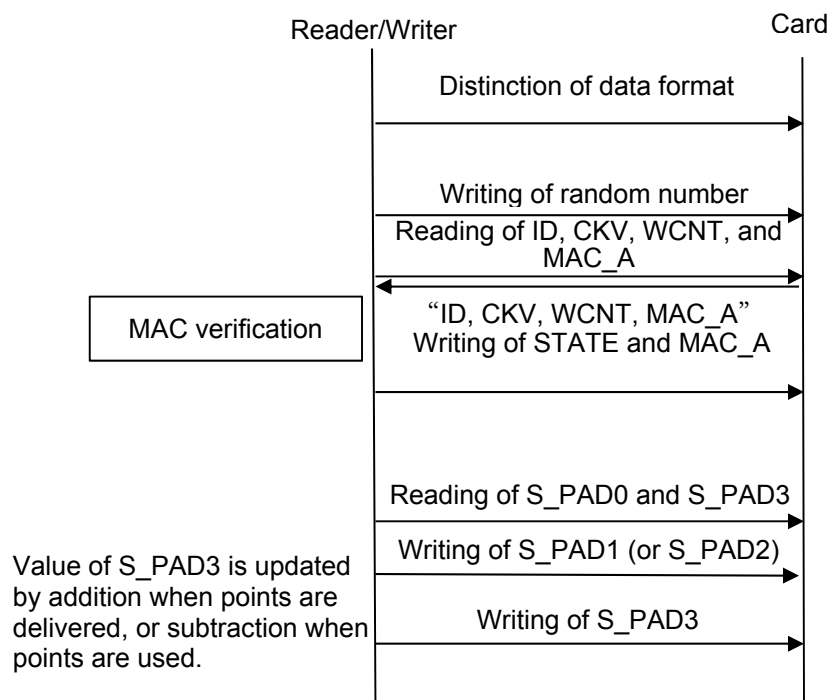


Figure 5-10: Example data format of a points card

5.2.5.2 Using Mutual Authentication and Write With MAC

Section 5.2.5.1 "Using Mutual Authentication" shows an example of implementing points cards using Mutual Authentication. Based on that example, if there is a risk that modification of data may occur on the communication channel (such as in online usage), access to the number of points shall use Read with MAC and Write With MAC. To prevent writing without the correct MAC, User Block to store the number of points shall be set with the Write With MAC attribute.

[Reference technical document]

"FeliCa Lite-S Diversified Card Key Standard Generation Algorithm"

"FeliCa Lite-S Security Application Note"

Section 5.4.2 "Integrity"

Summary

- Write the card number to S_PAD0, the log to S_PAD1 and S_PAD2, and the number of points to S_PAD3.
- Set each unused User Block to RO without changing their default values (ALL_00h).
- Set the logs to the Read After Authentication attribute and the Write After Authentication attribute.
- Set the number of points to the Write After Authentication With MAC attribute (authentication not required for reading).
- Set the value of ID and Diversified Card Key.
- MAC_A is computed by the Reader/Writer when the number of points is written.

Example of issuance

- Write the card number to S_PAD0.
- Write the log in two parts, referred to as Side A and Side B. Write Side A to S_PAD1. Write Side B to S_PAD2
- Write the number of points to S_PAD3.
- Write ALL_00h to S_PAD4 to S_PAD13, and ALL_FFh to REG.
- Write IDd (8 Bytes), DFC (2 Bytes), and an arbitrary value (6 Bytes) specified by the issuer to ID.
- Write a card key, which is diversified by the ID value and unique to each card, to CK. (For details of how to generate the key, see the "FeliCa Lite-S Diversified Card Key Standard Generation Algorithm".)
- Set the S_PAD1 to S_PAD3 to RW, and each other User Block to RO.
- Set the S_PAD1 to S_PAD2 to be the Read After Authentication attribute and the Write After Authentication attribute, and S_PAD3 to the Write After Authentication With MAC attribute.

NOTE For details of how to assure the concurrent writing of the log and number of points, see section 7.7 "Anti-tearing".

S_PAD0	RO	Card number		
S_PAD1	RW	Side A of log		
S_PAD2	RW	Side B of log		
S_PAD3	RW	Number of points		
S_PAD4	RO	00_00_00_00_..._00		
S_PAD13	RO	00_00_00_00_..._00		
REG	RO	FF_FF_FF_FF_..._FF		
ID	RO	IDd (8)	DFC (2)	Arbitrary value (6)
CK	RO	Key unique to each card		

Figure 5-11: Example data format of a points card

Example flow of processing

For the distinction of data format, see Figure 3-3.

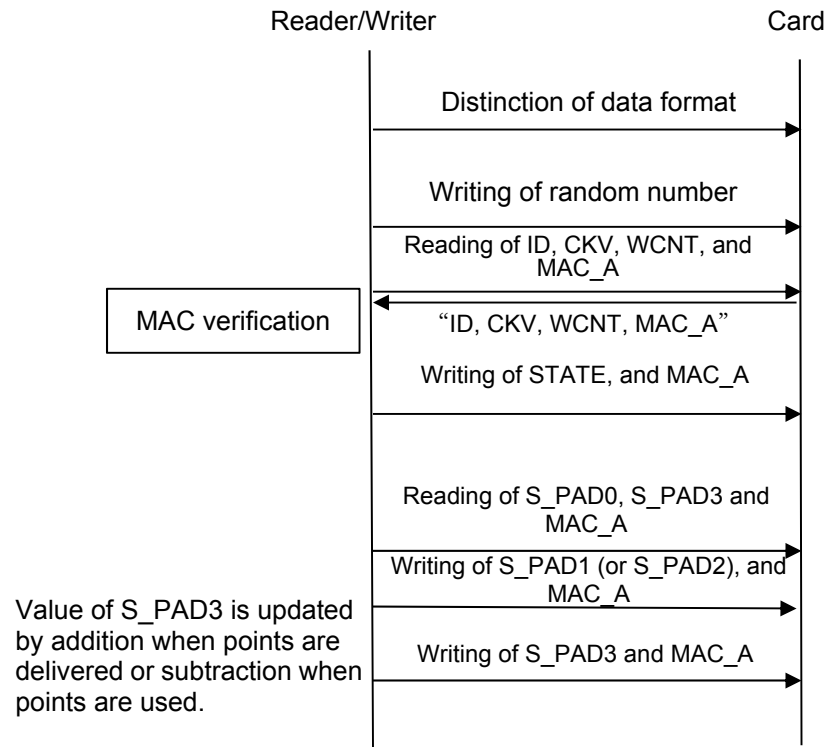


Figure 5-12: Example data format of a points card

5.2.5.3 Using signature

FeliCa Lite has no function that allows only the valid Reader/Writer to write. Therefore, it is recommended for the Reader/Writer to attach a signature by the Reader/Writer. This usage can also be applied to FeliCa Lite-S.

[Reference technical document]

“FeliCa Lite-S Diversified Card Key Standard Generation Algorithm”

“FeliCa Lite-S Security Application Note”

Section 5.4.2 “Integrity”

Summary

- Write the number of points to S_PAD0, and the signature to S_PAD1.
- Set each unused User Block to RO without changing their default values (ALL_00h).
- Set the value of ID and Diversified Card Key.
- The Reader/Write computes the points signature.

Example of issuance

- Set the Block Number to the first byte of all user blocks.
- Write the number of points to S_PAD0.
- Write the signature (to be diversified with a serial number to enable detection of card abuse by copying) to the upper 8 Bytes of S_PAD1, and ALL_00h to the last 8 Bytes of S_PAD1.
- Write ALL_00h to S_PAD2 to S_PAD13, and ALL_FFh to REG.
- Write IDd (8 Bytes), DFC (2 Bytes), and an arbitrary value (6 Bytes) specified by the issuer to ID.
- Write a card key, which is diversified by the ID value and unique to each card, to CK. (For details of how to generate the key, see the “FeliCa Lite-S Diversified Card Key Standard Generation Algorithm”.)
- Set the S_PAD0 and S_PAD1 to RW, and each other User Block to RO.

NOTE To prevent only S_PAD0 from being written, a mechanism to assure the concurrent rewriting of S_PAD0 and S_PAD1 is required.

S_PAD0	RW	00_Number of points (15 Bytes)		
S_PAD1	RW	01_Signature (8 Bytes) _00..._00 (7 Bytes)		
S_PAD2	RO	02_00_00_00..._00		
S_PAD13	RO	0D_00_00_00..._00		
REG	RO	0E_FF_FF_FF..._FF		
ID	RO	IDd (8)	DFC (2)	Arbitrary value (6)
CK	RO	Key unique to each card		

Figure 5-13: Example data format of a points card

Example flow of processing

For the distinction of data format, see Figure 3-3.

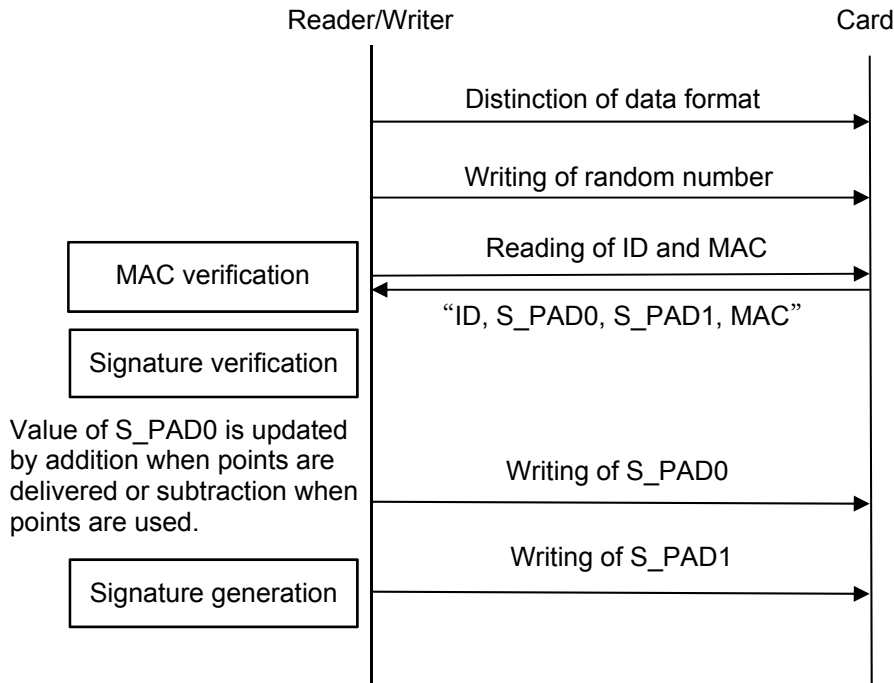


Figure 5-14: Example flow of processing for a points card

6 Cautions for correct use

6.1 Distinction of FeliCa Lite-S

System Code of FeliCa Lite-S is 88B4h. Use System Code to distinguish FeliCa Lite-S cards from other cards.

To prevent compatibility problems, do not use any parameter (such as Manufacturer Code or IC Code) whose value varies depending on the product model for this purpose.

System Code, Manufacturer Code, and IC Code are described as follows:

System Code

[Reference technical document]

“FeliCa Technology Code Descriptions”

Chapter 3 “System Code”

“FeliCa Lite-S User’s Manual”

Chapter 4 “Commands”

System Code is the value of the two Bytes used to identify System on the card.

System Code is assigned per service provider and for specific usage.

The Reader/Writer specifies System Code as the parameter of the Polling command, and any card whose System Code matches the parameter of the Polling command returns a Polling response.

System Code of FeliCa Lite-S is 88B4h. Use System Code to distinguish FeliCa Lite-S cards from other cards.

Manufacturer Code

[Reference technical document]

“FeliCa Technology Code Descriptions”

Chapter 1 “Manufacture ID (IDm)”

Manufacturer Code is a value set in the upper 2 Bytes of Manufacture ID (IDm). The value of Manufacturer Code of FeliCa Lite-S can vary, depending on the IC (chip) product.

Do not use Manufacturer Code to distinguish FeliCa Lite-S cards from other cards.

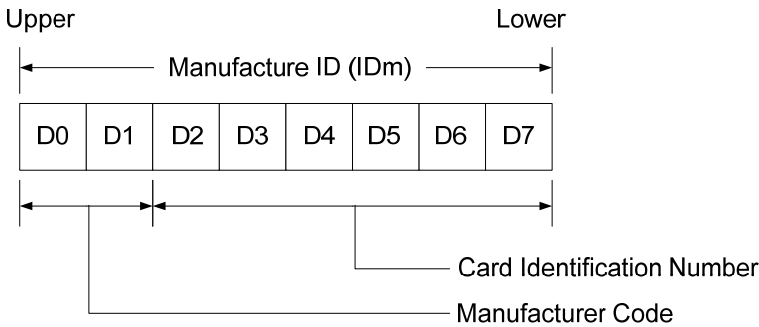


Figure 6-1: Manufacturer Code

IC Code

[Reference technical document]

“FeliCa Technology Code Descriptions”
Chapter 2 “Manufacture Parameter (PMm)”

IC Code is a value set in the first 2 Bytes of Manufacture Parameter (IDm). The Reader/Writer uses IC Code to identify the type of IC (i.e., chip) installed on a card. IC Code consists of “ROM Type” (1 Byte) and “IC Type” (1 Byte). The value of IC Code varies depending on the IC (chip) product.

Use System Code to distinguish FeliCa Lite series (FeliCa Lite and FeliCa Lite-S) cards from other cards.

Use IC Type to distinguish FeliCa Lite-S from FeliCa Lite among FeliCa Lite series, as follows:

- products including FeliCa Lite chip: IC Type = F0h
- products including FeliCa Lite-S chip: IC Type = F1h

For example, regard a card as FeliCa Lite when System Code is 88B4h and IC Type is F0h, and regard it as FeliCa Lite-S when System Code is 88B4h and IC Type is F1h to F7h.

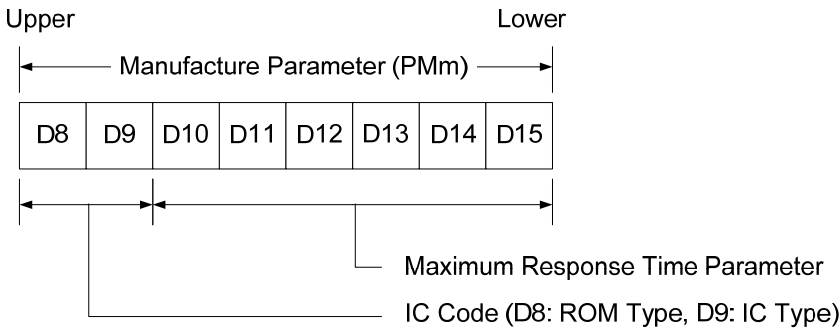


Figure 6-2: Manufacturer Code

6.2 Distinction of data format

System Code of FeliCa Lite-S is fixed to 88B4h. Unlike FeliCa Standard, FeliCa Lite-S does not allow you to use System Code to distinguish the data format. Use Data Format Code (DFC) to distinguish the data format of FeliCa Lite-S cards.

DFC

[Reference technical document]

“FeliCa Technology Code Descriptions”
Chapter 4 “Data Format Code”

DFC is the value of the two Bytes that identify the format of the data to be stored on the card. When a FeliCa Lite-S card is issued, DFC is written to ID Block. Use DFC to distinguish the data format of FeliCa Lite-S cards.

How to set DFC

[Reference technical document]

“FeliCa Lite-S User’s Manual”
Chapter 3 “File system”
Chapter 7 “Issuance”
Section A-1 “Data Format Code (DFC)”

To set DFC, write the value of DFC to ID Block.

Write the value of DFC to ID[8] and ID[9], and a value specified by the customer to ID[10] to ID[15]. IDd has been written to ID[0] to ID[7] before shipment and it cannot be changed.

ID[10] to ID[15] are assumed to be used to identify the application version. This part of ID Block, however, can also be used for other purposes.

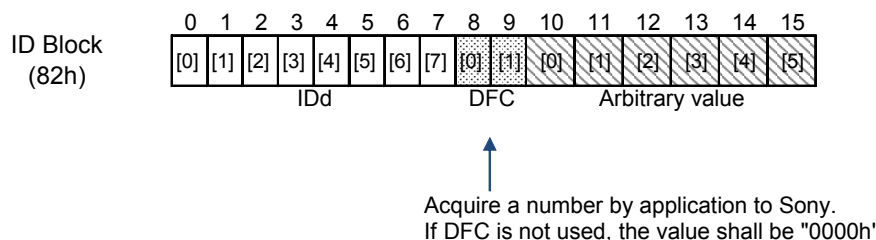


Figure 6-3: Setting of ID Block

6.3 Maximum response time

[Reference technical document]

"FeliCa Technology Code Descriptions"

Chapter 2 "Manufacture Parameter (PMm)"

The maximum response time parameter is a value set in the lower 6 Bytes of Manufacture Parameter (PMm). On a FeliCa Lite-S card, D13 indicates the maximum response time for the Read Without Encryption command, and D14 indicates that for the Write Without Encryption command.

The maximum response time varies depending on the IC (chip). Calculate the maximum response time from the PMm value in the Polling response, and set an appropriate timeout period on the Reader/Writer. If a fixed value is set as the timeout period on the Reader/Writer, some types of cards can reach the timeout limit before being able to respond to commands.

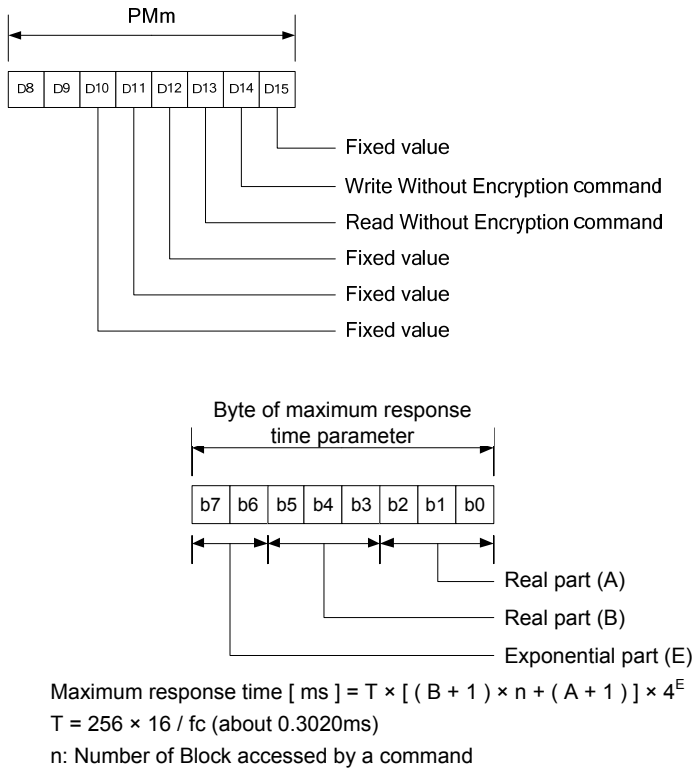


Figure 6-4: Maximum response time parameter

6.4 Command Packet Data

[Reference technical document]

"FeliCa Lite-S User's Manual"

Section 4.2 "Access to Block"

If a card receives an incorrect command packet, in most cases the card returns No Response or returns an error response. Specific types of erroneous command packet, however, can cause the card not to return No Response or to return an error response and to perform an unintended operation. Note that unintended operations can cause problems.

The following examples show the command packet to read 3 Blocks as S_PAD0 to S_PAD2 on a card. Each example assumes that Block List is followed by a CRC calculated from parameters such as LEN, Command Code, IDm, Number of Service, Service Code List, Number of Block, and Block List.

Example of a correct command packet

Item	LEN	Command Code	IDm	Number of Service	Service Code List	Number of Block	Block List		
No. of Bytes	1	1	8	1	2	1	2	2	2
Value (hex)	14h	06h	IDm	01h	0Bh 00h	03h	80h 00h	80h 01h	80h 02h

Figure 6-5: Example of a correct command packet

Example of an incorrect command packet (where the length of Packet Data does not match the value of LEN)

If 13h or 15h is specified as the value of LEN, the card shall not respond. The CRC position in the packet is distinguished by the value of LEN, so the card detects a CRC error and returns No Response. Note, however, that, if LEN = 13h, CRC can accidentally match due to a specific value of the 3rd Block List Element, in which case the card can behave unexpectedly.

Item	LEN	Command Code	IDm	Number of Service	Service Code List	Number of Block	Block List		
No. of Bytes	1	1	8	1	2	1	2	2	2
Value (hex)	13h	06h	IDm	01h	0Bh 00h	03h	80h 00h	80h 01h	80h 02h
Value (hex)	15h	06h	IDm	01h	0Bh 00h	03h	80h 00h	80h 01h	80h 02h

Figure 6-6: Examples of a command packet with an incorrect LEN value

Example of an incorrect command packet (where Number of Block does not match number of Block List Element)

If 02h is specified as Number of Block, the card returns the values of S_PAD0 and S_PAD1 (i.e., the card performs an unintended operation).

Item	LEN	Command Code	IDm	Number of Service	Service Code List	Number of Block	Block List		
No. of Bytes	1	1	8	1	2	1	2	2	2
Value (hex)	14h	06h	IDm	01h	0Bh 00h	02h	80h 00h	80h 01h	80h 02h

Figure 6-7: Command packet with incorrect Number of Block (1)

If 04h is specified as Number of Block, the card returns an error response. Block List for the 4th Block is not transmitted, so the card references undefined values. As a result, the card returns the error response indicating incorrect Access Mode, incorrect Service Code List Order, or incorrect Block Number. Note, however, that the card can perform unintentional operations if specific data is referenced.

Item	LEN	Command Code	IDm	Number of Service	Service Code List	Number of Block	Block List		
No. of Bytes	1	1	8	1	2	1	2	2	2
Value (hex)	14h	06h	IDm	01h	0Bh 00h	04h	80h 00h	80h 01h	80h 02h

Figure 6-8: Command packet with incorrect Number of Block (2)

Example of an incorrect command packet (where Block List Element is used incorrectly)

If the packet includes 2-Byte Block List Element that do not begin with 80h, the card returns an error response.

Item	LEN	Command Code	IDm	Number of Service	Service Code List	Number of Block	Block List		
No. of Bytes	1	1	8	1	2	1	2	2	2
Value (hex)	14h	06h	IDm	01h	0Bh 00h	03h	00h 00h	00h 01h	00h 02h

Figure 6-9: Command packet with incorrect Block List Element

7 Frequently asked questions about FeliCa Lite-S

7.1 Uniqueness of IDm ---

[Reference technical document]

"FeliCa Technology Code Descriptions"
Chapter 1 "Manufacture ID (IDm)"

What is IDm?

IDm is an 8-Byte ID that a card returns in a response to the Polling command. IDm is intended primarily to identify a card during communication between a Reader/Writer and the card. IDm is also used to trace a manufactured card.

Is the value of IDm unique?

The IDm of a FeliCa Lite-S IC (chip) is given a unique value when the IC chip is manufactured.
The IDm of a FeliCa Lite-S also differs from the IDm of any other FeliCa ICs (such as FeliCa Standard, FeliCa Lite, and FeliCa Plug ICs).

7.2 Setting of card key

[Reference technical document]

"FeliCa Lite-S User's Manual"

Chapter 5 "Security"

"FeliCa Lite-S Security Application Note"

Chapter 5 "Security"

"FeliCa Lite-S Diversified Card Key Standard Generation Algorithm"

Is it necessary to set a card key?

When authentication is performed, the card generates a session key from the card key, and then generates MAC from the session key and Block Data.

If the authentication functionality is used, the card key shall be set.

If the authentication functionality is not used, the card key does not need to be set.

Is it necessary to diversify the card key?

Internal Authentication is the function that checks whether the card has a valid card key.

Even when the card key is not diversified, authentication can be performed functionally if ID Block has been diversified. Sony, however, strongly recommends that the card key should be diversified to minimize the adverse influence that can result if the card key is compromised.

How can I diversify the card key?

Sony provides an algorithm to diversify the card keys for FeliCa Lite-S cards. For details of the algorithm, see the "FeliCa Lite-S Diversified Card Key Standard Generation Algorithm". Sony also provides a sample program. For the sample program, contact your Sony sales team.

The algorithm enables you to generate Diversified Card Key from Master Key for diversification and the value of ID Block. The Reader/Writer that has Master Key for diversification can generate Diversified Card Key for each card by reading ID Block of the card. You do not need to manage Diversified Card Keys of all cards.

7.3 Authentication

FeliCa Lite-S provides the following three authentication functions:

- Internal Authentication (unilateral authentication)
Enables the Reader/Writer to check that the card has a valid card key.
- External Authentication
Enables the card to check that the Reader/Writer has a valid card key.
- Mutual Authentication
Enables the Reader/Writer to check that the card has a valid card key, and enables the card to check that the Reader/Writer has a valid card key.

7.3.1 Internal Authentication (unilateral authentication)

FeliCa Lite has only the authentication function that enables the Reader/Writer to check that the card has a valid card key, and therefore the function is referred to as unilateral authentication. Because FeliCa Lite-S has the mutual (i.e., bilateral) authentication function, the function to enable the (external) Reader/Writer to authenticate the (internal) card is referred to as the Internal Authentication.

[Reference technical document]

“FeliCa Lite-S User’s Manual”

Chapter 5 “Security”

“FeliCa Lite-S Security Application Note”

Chapter 5 “Security”

“FeliCa Lite-S Diversified Card Key Standard Generation Algorithm”

What is Internal Authentication?

Internal Authentication is the function to enable the Reader/Writer to check that the card has a valid card key.

What are the requirements for Internal Authentication?

The card shall satisfy the following condition:

- The card has Diversified Card Key set in CK Block.

The Reader/Writer shall satisfy the following conditions:

- The Reader/Writer can generate random numbers.
- The Reader/Writer has Master Key for diversification.
- The Reader/Writer can generate Diversified Card Key from Master Key for diversification and the data read from ID Block.
- The Reader/Writer can generate a session key from Diversified Card Key and a random number.
- The Reader/Writer can generate MAC_A from the session key and read Block Data.
- The Reader/Writer can verify the MAC_A read from a card with the MAC_A the Reader/Writer generated.

How does Internal Authentication processing flow?

Figure 7-1 shows an example flow of Internal Authentication processing. In this example, N Master Keys for diversification are prepared and separately used, to minimize the adverse influence that can result if any of them leaks out.

The flow of processing is described as follows (each operation described in parentheses is a card operation):

- 1) The Reader/Writer acquires a card.
- 2) The Reader/Writer generates a random number, and then writes it to the card. ((2)' The card generates a session key.)
- 3) The Reader/Writer reads ID Block, CKV Block, and MAC_A Block at the same time. ((3)' The card generates a MAC_A.)
- 4) The Reader/Writer selects Master Key for diversification from Key Version.
- 5) The Reader/Writer generates Diversified Card Key.
- 6) The Reader/Writer generates a session key.
- 7) The Reader/Writer generates a MAC_A.
- 8) The Reader/Writer verifies the MAC_A generated by the card.

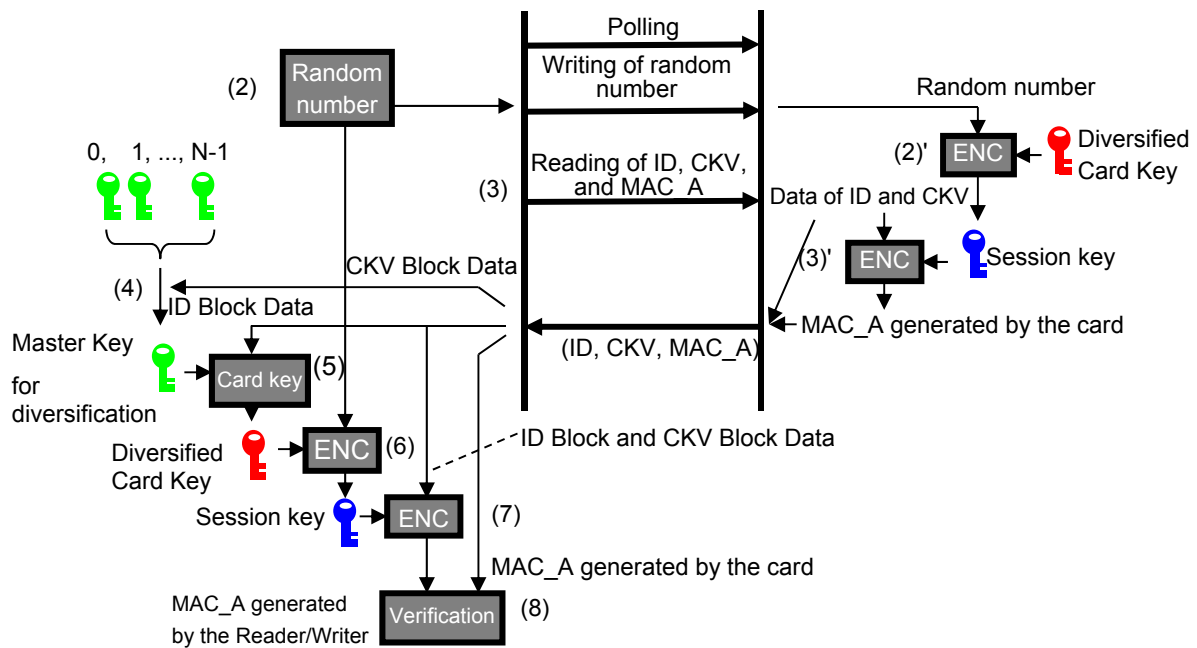


Figure 7-1: Flow of Internal Authentication processing

7.3.2 External Authentication

[Reference technical document]

"FeliCa Lite-S User's Manual"

Chapter 5 "Security"

"FeliCa Lite-S Security Application Note"

Chapter 5 "Security"

"FeliCa Lite-S Diversified Card Key Standard Generation Algorithm"

What is External Authentication?

External Authentication is the function to enable the card to check that the Reader/Writer has a valid card key.

What are the requirements for External Authentication?

The card shall satisfy the following conditions:

- The card has Diversified Card Key set in CK Block.
- STATE Block is set to Write With MAC_A.

The Reader/Writer shall satisfy the following conditions:

- The Reader/Writer can generate random numbers.
- The Reader/Writer has Master Key for diversification.
- The Reader/Writer can generate Diversified Card Key from Master Key for diversification and the data read from ID Block.
- The Reader/Writer can generate a session key from Diversified Card Key and a random number.
- The Reader/Writer can generate MAC_A from the session key and read Block Data.
- The Reader/Writer can generate MAC_A from the session key and Block Data to be written.

How does External Authentication processing flow?

Figure 7-2 shows an example flow of External Authentication processing. In this example, N Master Keys for diversification are prepared and separately used, to minimize the adverse influence that can result if any of them leaks out.

The flow of processing is described as follows (each operation described in parentheses is a card operation):

- 1) The Reader/Writer acquires a card.
- 2) The Reader/Writer generates a random number, and then writes it to the card. ((2)' The card generates a session key.)
- 3) The Reader/Writer reads ID Block, CKV Block, and MAC_A Block at the same time.
- 4) The Reader/Writer selects Master Key for diversification from Key Version.
- 5) The Reader/Writer generates Diversified Card Key.
- 6) The Reader/Writer generates a session key.

- 7) The Reader/Writer generates MAC_A from the data of STATE Block and WCNT, and writes it to the card. ((7)' The card generates MAC_A from the written STATE Block value and WCNT. (7)'' The card verifies MAC_A.)

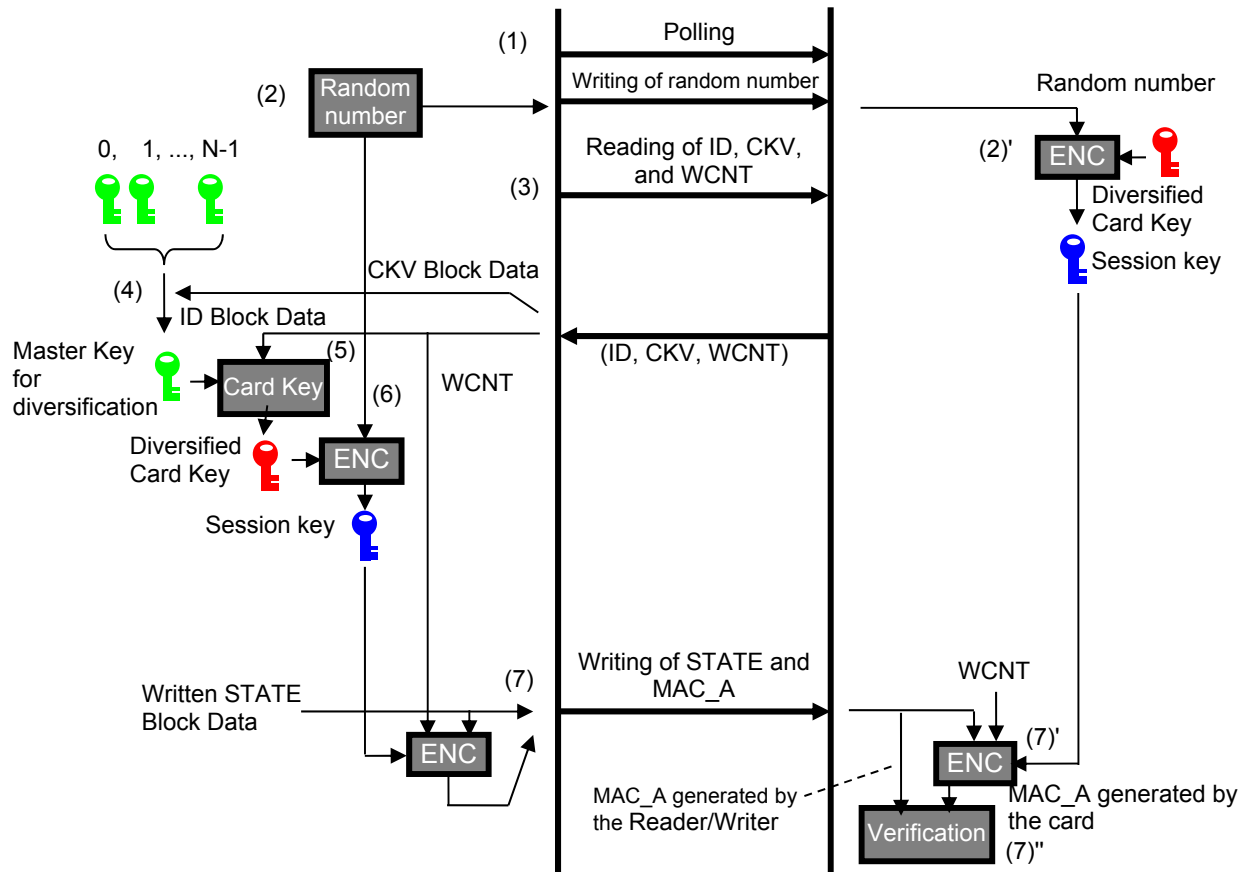


Figure 7-2: Flow of External Authentication processing

7.3.3 Mutual Authentication

[Reference technical document]

"FeliCa Lite-S User's Manual"

Chapter 5 "Security"

"FeliCa Lite-S Security Application Note"

Chapter 5 "Security"

"FeliCa Lite-S Diversified Card Key Standard Generation Algorithm"

What is Mutual Authentication?

Mutual Authentication is the function to enable the Reader/Writer to check that the card has a valid card key, and enable the card to check that the Reader/Writer has a valid card key.

What are the requirements for Mutual Authentication?

The card shall satisfy the following conditions:

- The card has Diversified Card Key set in CK Block.
- STATE Block is set to Write With MAC_A.

The Reader/Writer shall satisfy the following conditions:

- The Reader/Writer can generate random numbers.
- The Reader/Writer has Master Key for diversification.
- The Reader/Writer can generate Diversified Card Key from Master Key for diversification and the data read from ID Block.
- The Reader/Writer can generate a session key from Diversified Card Key and a random number.
- The Reader/Writer can generate MAC_A from the session key and read Block Data.
- The Reader/Writer can verify the MAC_A read from a card with the MAC the Reader/Writer generated.
- The Reader/Writer can generate MAC_A from the session key and Block Data to be written.

How does Mutual Authentication processing flow?

Figure 7-3 shows an example flow of Mutual Authentication processing. In this example, N Master Keys for diversification are prepared and separately used, to minimize the adverse influence that can result if any of them leaks out.

The flow of processing is described as follows (each operation described in parentheses is a card operation):

- 1) The Reader/Writer acquires a card.
- 2) The Reader/Writer generates a random number, and then writes it to the card. ((2)' The card generates a session key.)
- 3) The Reader/Writer reads ID Block, CKV Block, and MAC_A Block at the same time. ((3)' The card generates MAC_A.)
- 4) The Reader/Writer selects Master Key for diversification from Key Version.
- 5) The Reader/Writer generates Diversified Card Key.
- 6) The Reader/Writer generates a session key.
- 7) The Reader/Writer generates MAC_A.
- 8) The Reader/Writer verifies MAC_A.
- 9) The Reader/writer generates MAC_A from the data of STATE Block and WCNT, and then writes it to the card. ((9)' The card generates MAC_A from the written STATE Block value and WCNT. (9)'' The card verifies MAC_A.)

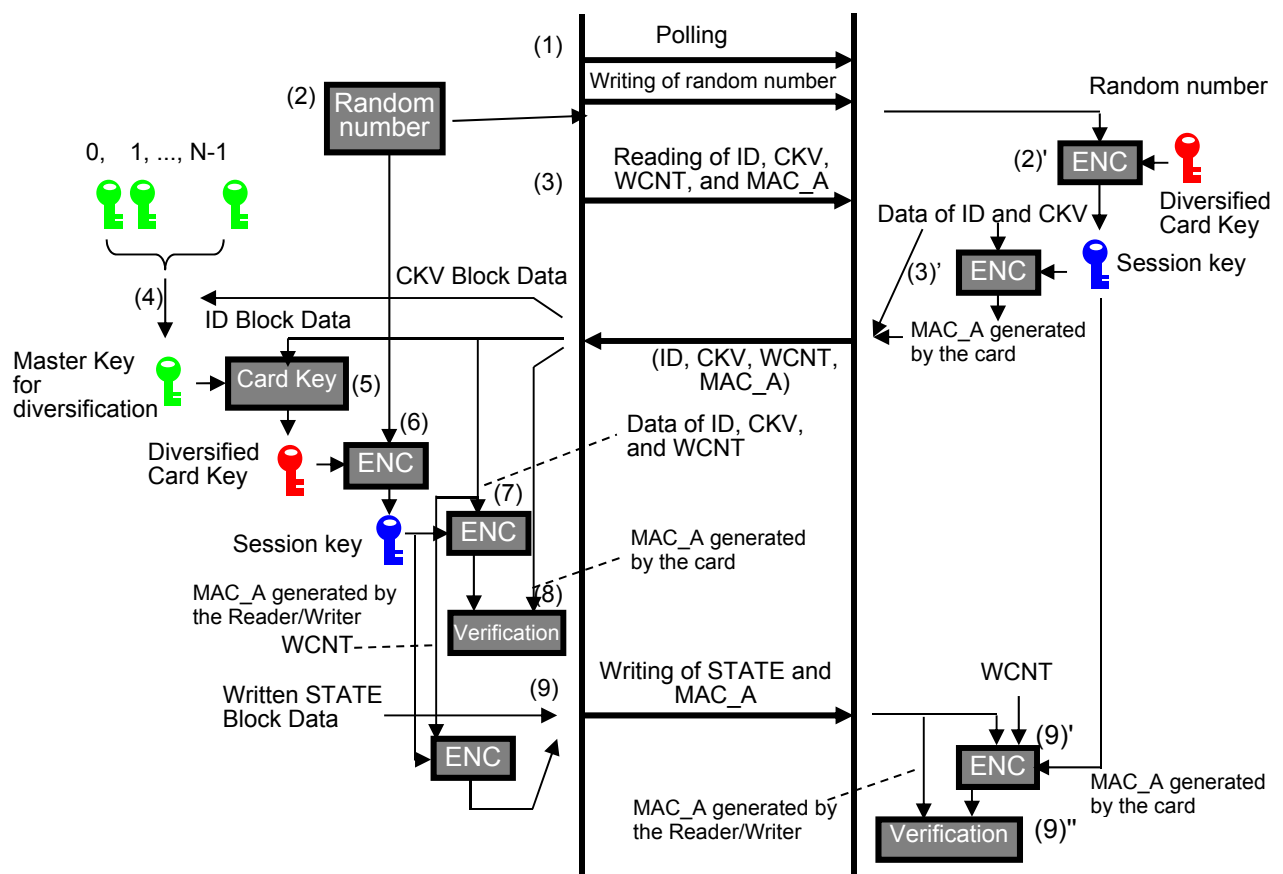


Figure 7-3: Flow of Mutual Authentication processing

7.4 Differences from FeliCa Standard

[Reference technical document]

“FeliCa Lite-S Security Application Note”

Chapter 5 “Security”

Table 7-1 lists the main differences between FeliCa Lite, FeliCa Lite-S, and FeliCa Standard.

Table 7-1: Main differences among FeliCa Lite, FeliCa Lite-S, and FeliCa Standard

Category	Item	FeliCa Lite	FeliCa Lite-S	FeliCa Standard
File structure	File format	Fixed		Freely specifiable
	Area (hierarchical structure)	Not supported		Supported
	Number of User Block	114 (224 Bytes)		150 (2400 Bytes)
Security	Channel encryption	Not supported		Supported
	Authentication	Unilateral authentication	Mutual Authentication	Mutual Authentication
	Access Key	One per card		Specifiable for each Service
	Access permission (RW and RO)	Supported		Supported
	Access type	Random only		Random, Cyclic, Purse
	Anti-tearing	Supported (for User Block only)	Supported (for all Block)	Supported (for all Block)
	Authentication code generation (MAC)	Supported		Not supported
	Subtraction register	Supported		Not supported
	Anti-tampering function	Supported		Supported (security certification acquired)
Operation	Identifier	IDm, ID		IDm, IDi
	System Code	Fixed to '88B4h'		Varied by card usage
	Timing of IDm writing	At chip manufacturing (by Sony)		At card manufacturing

7.5 Reader/Writer used for FeliCa Standard

The following items are common to FeliCa Lite-S and FeliCa Standard:

- Physical layer
Carrier frequency (13.56 MHz), ASK modulation, and Manchester code
- Data link layer
Definitions of packet fields
- Application layer
Definitions of command and response packets
- Supported commands
Polling, Read Without Encryption, Write Without Encryption
(FeliCa Standard also supports other commands.)

The Reader/Writer that is used for FeliCa Standard cards can send and receive the commands supported by FeliCa Lite-S cards.

Note, however, that the command packets the Reader/Writer sends shall be modified, because the file format of FeliCa Lite-S cards is not always the same as that of the FeliCa Standard card you are using.

In most cases, the hardware of the Reader/Writer used for FeliCa Standard cards can be used for FeliCa Lite-S cards without modification, but the software to control the Reader/Writer shall be modified.

7.6 Using FeliCa Standard, FeliCa Lite, and FeliCa Lite-S cards together

7.6.1 Using FeliCa Standard and FeliCa Lite-S cards together

[Reference technical document]

“Format Distinction Sequence Design Guidelines”

Can FeliCa Standard and FeliCa Lite-S cards be used together?

FeliCa Lite-S and FeliCa Standard cards use the same commands and file system, but they have different file formats.

FeliCa Lite-S and FeliCa Standard cards can safely be used together with the same Reader/Writer without problems, as follows:

The Reader/Writer distinguishes the type of the card, and branches the subsequent processing accordingly.

How should processing be done if FeliCa Lite-S and FeliCa Standard cards are used together?

Firstly, the Reader/Writer uses System Code to distinguish whether the card is a FeliCa Standard card, and processing is performed according to the result of that distinction. Next, the Reader/Writer uses DFC to distinguish whether the card is a FeliCa Lite-S card, and processing is performed according to the result of that distinction.

Figure 7-4 shows an example flow of processing.

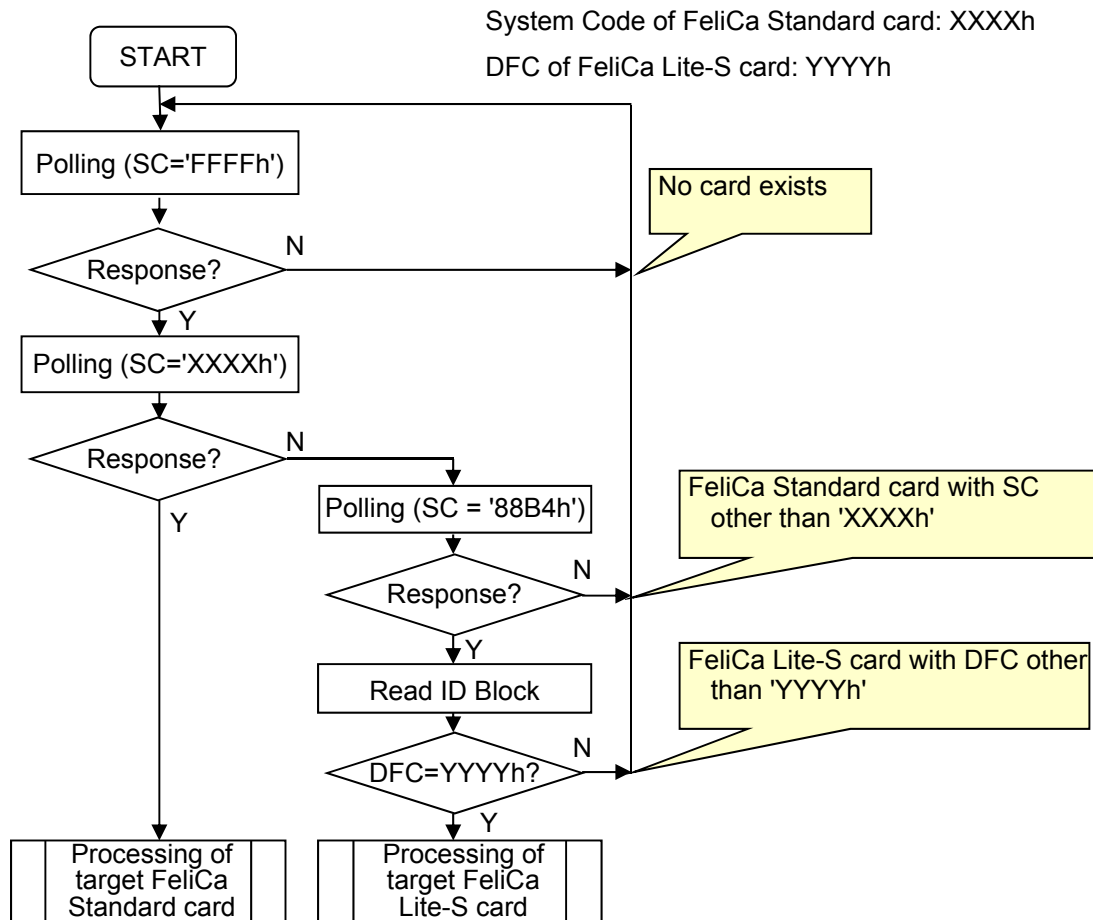


Figure 7-4: Example flow of processing for combined use of FeliCa Lite-S and FeliCa Standard cards

7.6.2 Using FeliCa Lite and FeliCa Lite-S together

[Reference technical document]

“Format Distinction Sequence Design Guidelines”

Can FeliCa Lite and FeliCa Lite-S cards be used together?

FeliCa Lite-S is upward compatible with FeliCa Lite. All the functions of FeliCa Lite can be used with FeliCa Lite-S without distinguishing between FeliCa Lite-S and FeliCa Lite.

If FeliCa Lite-S is not used properly, compatibility problem can occur. For details, see section 6 “Cautions for correct use”.

7.7 Anti-tearing

How can concurrent writing be assured in case of a power failure?

FeliCa Lite-S allows writing of up to 1 Block of data per command (or 2 Blocks of data only when the 2nd Block is MAC_A). Concurrent writing of 1 Block (16 Bytes) of data is assured. I.e., if the power is lost during writing, either all of the 16 Bytes of data are written or none is written. This functionality prevents incomplete data from being written.

Therefore, the Reader/Writer does not need to assure concurrent writing when only 1 Block of data is rewritten in a series of processing. The Reader/Writer shall assure concurrent writing if 2 or more Blocks are rewritten.

Figure 7-5 shows an example of the process that assures the concurrent writing of 3 Blocks. In this example, 5 Blocks of S_PAD0 to S_PAD4 are used. Two sets are used for the 2 Blocks to be written firstly. S_PAD0, S_PAD1, and S_PAD4 are treated as a set, and S_PAD2, S_PAD3, and S_PAD4 are treated as another set. S_PAD4 includes a Valid flag indicating which set of Block is valid.

Valid=0 indicates that the set of S_PAD0, S_PAD1, and S_PAD4 is valid.

Valid=1 indicates that the set of S_PAD2, S_PAD3, and S_PAD4 is valid.

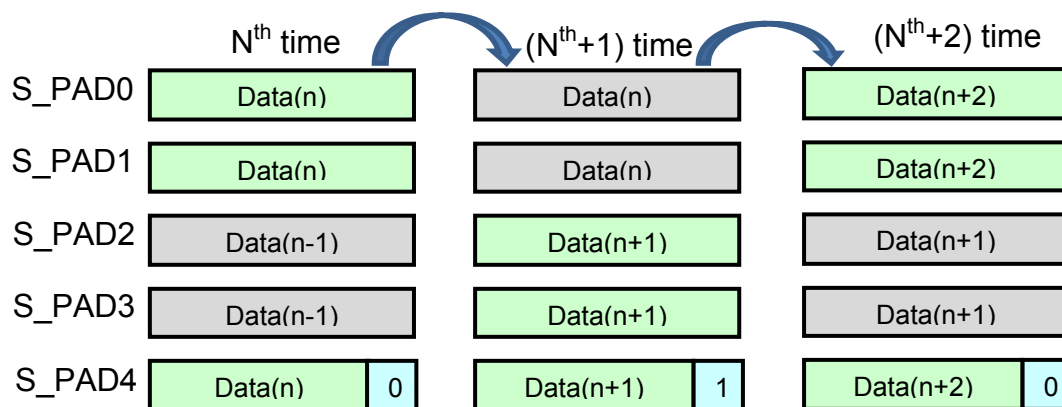


Figure 7-5: Example of process that assures for the concurrent writing of 3 Blocks

Upon completion of writing to S_PAD4, the Valid flag is updated and the valid set is switched. If the power is lost before completion of writing of S_PAD4, no data is written as a result. Figure 7-6 shows the writing procedure in assuring process for concurrent writing of 3 Blocks.

The following describes the process where Data(n) written in S_PAD0, S_PAD1, and S_PAD4 is updated to Data(n+1):

- 1) Data(n+1) is written in S_PAD2.
- 2) Data(n+1) is written in S_PAD3.
- 3) Data(n+1) and Valid=1 are written in S_PAD4.

The figure shows that the stored data, Data(n), is unchanged and remains as it is if the power is lost while Data(n) is being updated to Data(n+1).

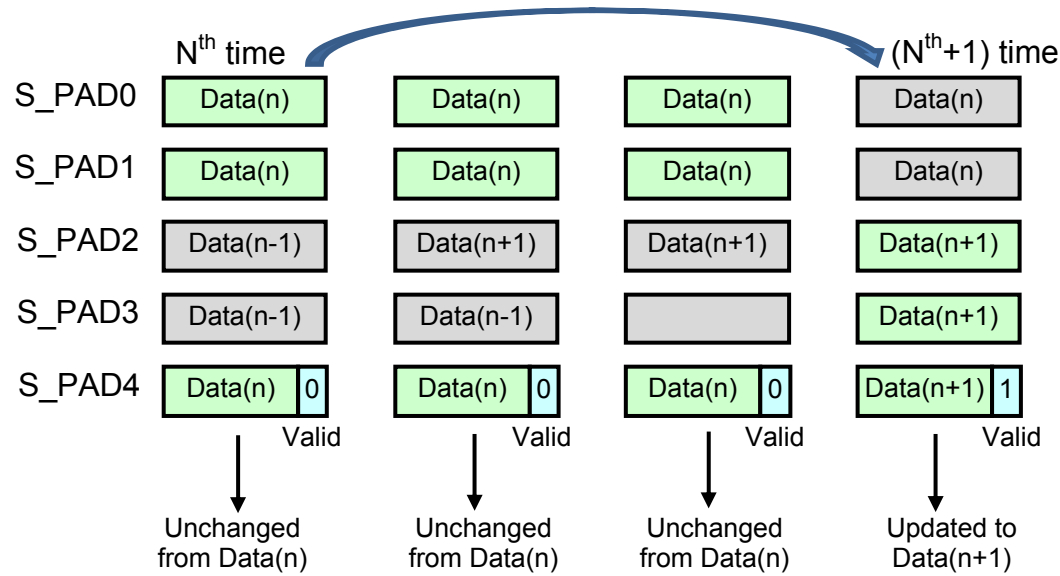


Figure 7-6: The writing procedure in the process that assures the concurrent writing of 3 Blocks

FeliCa Lite-S

FeliCa Lite-S Starter's Manual

Version 1.11

March 2012

First Edition

FeliCa Business Division

April 2017

Revision

Sony Imaging Products & Solutions Inc.

No. M740-E01-11

© 2012, 2017 Sony Imaging Products & Solutions Inc.

Printed in Japan