AES, DES, and RSA Support
(Intended for Domestic Use)

SASEBO-W Smart Card OS Specification

Version 0.4-5

April 1, 2011

National Institute of Advanced Industrial Science and Technology

(AIST)

Table of Contents

1. Overview

SASEBO-W Smart Card OS provides software implementations of cryptographic algorithms. This operating system is installed on an ISO/IEC 7816-3 contact card that includes an 8-bit Atmel AVR microcontroller, ATmega163.

The SASEBO-W Smart Card OS supports the following cryptographic algorithms.

- Advance Encryption Standard (AES-)128-bit encryption and decryption
- Data Encryption Standard (DES) encryption and decryption
- Rivest, Shamir and Adleman (RSA) 32-bit encryption and decryption

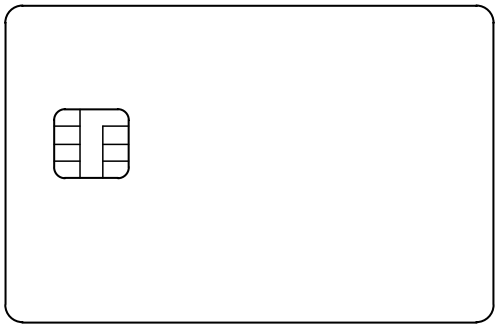

Figure 1： Card Schematic (Top View)

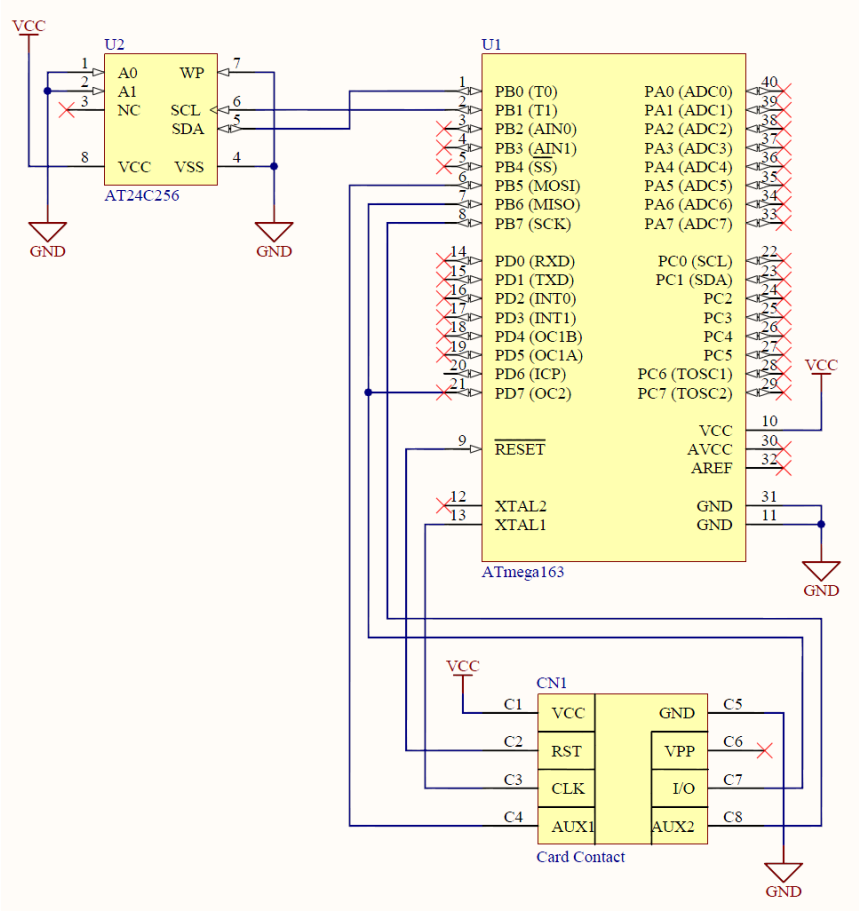

Figure 2： Internal Equivalent Circuit

2. Electrical Specifications

The hardware interface specifications of the ATmega microcontroller that contains SASEBO-W Smart Card OS are given below.

Table 1 : Card Interface Specifications

Parameter	Value
Transfer protocol	T = 0
Clock frequency	3.5712 MHz
Transfer bit rate	9600 bps

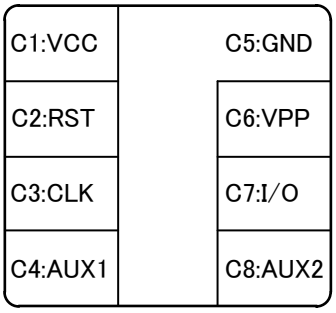

Figure 3 : Assignments of Card Contacts and Functions (Top View)

Table 2 : Pin Assignments

Contact	Function	Direction	Description
C1	VCC		Power
C2	RST	In	Reset (negative logic)
C3	CLK	In	Clock
C4	AUX1	Out	EXEC (pin driven high during processing)
C5	GND		Ground
C6	VPP		Unused
C7	I/O	In/Out	Serial data (open drain)
C8	AUX2	Out	Unused

Other electrical specifications conform to the Atmel ATmega163 specifications.

3. Answer To Reset (ATR)

SASEBO-W Smart Card OS returns the ATR as follows.

```
0x3B 0xAE 0x00 0x40 0x32 0x73 0x61 0x53 0x65 0x42 0x4F 0x32 0x30 0x31 0x30 0x31 0x32 0x30
0x31
```

Table 3： ATR Format

Offset	Data Element	Value	Description
0	TS	0x3B	Direct convention
1	T0	0xAE	TB1 & TD1 sent (0xA) Number of historical characters = 14 (0xE)
2	TB1	0x00	VPP contact is not used (II = 0, PI1 = 0)
3	TD1	0x40	TC2 sent (0x4) Transmission protocol T = 0 (0x0)
4	TC2	0x32	Work waiting time = 50×100 ms (WI = 50 (0x32))
5–18	Historical characters	“saSeBO20101201”	Identification & version number

4. Application Protocol Data Unit (APDU)

SASEBO-W Smart Card OS supports the following APDU commands (Terminal → Smart Card).

Table 4： APDU Command List

Command	CLA	INS	P1	P2	P3	Data	Response
MAKE_AES_KEY	0x80	0x12	0x00	0x00	0x10	Key	Status
ENCRYPT_AES	0x80	0x04	0x04	0x??	0x10	Plaintext	Status
DECRYPT_AES	0x80	0x08	0x04	0x??	0x10	Ciphertext	Status
MAKE_DES_KEY	0x80	0x0A	0x00	0x00	0x08	Key	Status
ENCRYPT_DES	0x80	0x04	0x00	0x00	0x08	Plaintext	Status
DECRYPT_DES	0x80	0x08	0x00	0x00	0x08	Ciphertext	Status
LOAD_RSA_EXP	0x80	0x00	0x00	0x00	0x04	Key	Status
LOAD_RSA_CRT_EXP1	0x80	0x00	0x01	0x00	0x02	Key	Status
LOAD_RSA_CRT_EXP2	0x80	0x00	0x02	0x00	0x02	Key	Status
LOAD_RSA_MOD	0x80	0x02	0x00	0x00	0x04	Key	Status
LOAD_RSA_CRT_PRM1	0x80	0x02	0x01	0x00	0x02	Key	Status
LOAD_RSA_CRT_PRM2	0x80	0x02	0x02	0x00	0x02	Key	Status
LOAD_RSA_CRT_COEF	0x80	0x02	0x03	0x00	0x02	Key	Status
MAKE_RSA_KEY	0x80	0x06	0x00	0x00	0x00	–	Status
ENCRYPT_RSA	0x80	0x04	0x01	0x??	0x04	Plaintext	Status
GET_RESPONSE	0x80	0xc0	0x00	0x00	length	–	Data + Status
NOP	0x80	0x80	0x00	0x00	0x00	–	Status
ECHO	0x80	0x82	0x00	0x00	length	Any data	Status
ATR	0x80	0x84	0x00	0x00	0x00	–	Status

The list of status words in the response APDU (Smart Card → Terminal) is given in the following table.

Table 5： Status List

Status	Description
0x9000	OK without response data
0x9F??	OK with response data (length = 0x??)
0x6D00	Bad INS
0x6E00	Bad CLS
0x6A86	Bad P1 or P2
0x6700	Bad P3

4.1. AES Commands

4.1.1. MAKE_AES_KEY

The MAKE_AES_KEY command sets the AES key.

This command must be re-issued after issuing commands to other cryptographic algorithms.

Command:

CLA	INS	P1	P2	P3	Data
0x80	0x12	0x00	0x00	0x10	Key (16-byte)

Response:

SW1	SW2	Description
0x90	0x00	Successfully executed
0x6A	0x86	Bad P1 or P2
0x67	0x00	Bad P3

4.1.2. ENCRYPT_AES

The ENCRYPT_AES command executes AES encryption.

A response is returned upon completion of AES encryption. The AES algorithm can be specified in the P2 value. Details of the algorithms are described in Section 7, Cryptographic Algorithms.

Command:

CLA	INS	P1	P2	P3	Data
0x80	0x04	0x04	0x??	0x10	Plaintext (16-byte)

P2:

P2	Algorithm
0x00	Byte-oriented implementation
0x01	Hardware-like implementation

Response:

SW1	SW2	Description
0x9F	0x10	Successfully executed (16-byte Ciphertext is available)
0x6A	0x86	Bad P1 or P2
0x67	0x00	Bad P3

4.1.3. DECRYPT_AES

The DECRYPT_AES command executes AES decryption.

A response is returned upon completion of AES decryption. The AES algorithm can be specified in the P2 value. Details of the algorithms are given in Section 7, Cryptographic Algorithms.

Command:

CLA	INS	P1	P2	P3	Data
0x80	0x08	0x04	0x??	0x10	Ciphertext (16-byte)

P2:

P2	Algorithm
0x00	Byte-oriented implementation
0x01	Hardware-like implementation

Response:

SW1	SW2	Description
0x9F	0x10	Successfully executed (16-byte Plaintext is available)
0x6A	0x86	Bad P1 or P2
0x67	0x00	Bad P3

4.2. DES Commands

4.2.1. MAKE_DES_KEY

The MAKE_DES_KEY command sets the DES key.

This command must be re-issued after issuing commands to other cryptographic algorithms.

Command:

CLA	INS	P1	P2	P3	Data
0x80	0x0A	0x00	0x00	0x08	Key (8-byte)

Response:

SW1	SW2	Description
0x90	0x00	Successfully executed
0x6A	0x86	Bad P1 or P2
0x67	0x00	Bad P3

4.2.2. ENCRYPT_DES

The ENCRYPT_DES command executes DES encryption.

A response is returned upon completion of DES encryption.

Command:

CLA	INS	P1	P2	P3	Data
0x80	0x04	0x00	0x00	0x08	Plaintext (8-byte)

Response:

SW1	SW2	Description
0x9F	0x08	Successfully executed (8-byte ciphertext is available)
0x6A	0x86	Bad P1 or P2
0x67	0x00	Bad P3

4.2.3. DECRYPT_DES

The DECRYPT_DES command executes DES decryption.

A response is returned upon completion of DES decryption.

Command:

CLA	INS	P1	P2	P3	Data
0x80	0x08	0x00	0x00	0x08	Ciphertext (8-byte)

Response:

SW1	SW2	Description
0x9F	0x08	Successfully executed (8-byte plaintext is available)
0x6A	0x86	Bad P1 or P2
0x67	0x00	Bad P3

4.3. RSA Commands

4.3.1. LOAD_RSA_EXP

The LOAD_RSA_EXP command sets the RSA exponent.

This command must be re-issued after issuing commands to other algorithms (include RSA-CRT).

Command:

CLA	INS	P1	P2	P3	Data
0x80	0x00	0x00	0x00	0x04	Exponent (4-byte)

Response:

SW1	SW2	Description
0x90	0x00	Successfully executed
0x6A	0x86	Bad P1 or P2
0x67	0x00	Bad P3

4.3.2. LOAD_RSA_CRT_EXP1

The LOAD_RSA_CRT_EXP1 command sets the first exponent of RSA-CRT.

This command must be re-issued after issuing commands to other cryptographic algorithms (including RSA).

Command:

CLA	INS	P1	P2	P3	Data
0x80	0x00	0x01	0x00	0x02	Exponent (2-byte)

Response:

SW1	SW2	Description
0x90	0x00	Successfully executed
0x6A	0x86	Bad P1 or P2
0x67	0x00	Bad P3

4.3.3. LOAD_RSA_CRT_EXP2

The LOAD_RSA_CRT_EXP2 command sets the second exponent of RSA-CRT.

This command must be re-issued after issuing commands to other cryptographic algorithms (including RSA).

Command:

CLA	INS	P1	P2	P3	Data
0x80	0x00	0x02	0x00	0x02	Exponent (2-byte)

Response:

SW1	SW2	Description
0x90	0x00	Successfully executed
0x6A	0x86	Bad P1 or P2
0x67	0x00	Bad P3

4.3.4. LOAD_RSA_MOD

The LOAD_RSA_EXP command sets the RSA modulus.

This command must be re-issued after issuing commands to other algorithms (including RSA-CRT).

Command:

CLA	INS	P1	P2	P3	Data
0x80	0x02	0x00	0x00	0x04	Modulus (4-byte)

Response:

SW1	SW2	Description
0x90	0x00	Successfully executed
0x6A	0x86	Bad P1 or P2
0x67	0x00	Bad P3

4.3.5. LOAD_RSA_CRT_PRM1

The LOAD_RSA_CRT_PRM1 command sets the first prime of RSA-CRT.

This command must be re-issued after issuing commands to other cryptographic algorithms (include RSA).

Command:

CLA	INS	P1	P2	P3	Data
0x80	0x02	0x01	0x00	0x02	Prime (2-byte)

Response:

SW1	SW2	Description
0x90	0x00	Successfully executed
0x6A	0x86	Bad P1 or P2
0x67	0x00	Bad P3

4.3.6. LOAD_RSA_CRT_PRM2

The LOAD_RSA_CRT_PRM1 command sets the second prime of RSA-CRT.

This command must be re-issued after issuing commands to other cryptographic algorithms (include RSA).

Command:

CLA	INS	P1	P2	P3	Data
0x80	0x02	0x02	0x00	0x02	Prime (2-byte)

Response:

SW1	SW2	Description
0x90	0x00	Successfully executed
0x6A	0x86	Bad P1 or P2
0x67	0x00	Bad P3

4.3.7. LOAD_RSA_CRT_COEF

The LOAD_RSA_CRT_COEF command sets the coefficient of RSA-CRT.

This command must be re-issued after issuing commands to other cryptographic algorithms (including RSA).

Command:

CLA	INS	P1	P2	P3	Data
0x80	0x02	0x03	0x00	0x02	Coefficient (2-byte)

Response:

SW1	SW2	Description
0x90	0x00	Successfully executed
0x6A	0x86	Bad P1 or P2
0x67	0x00	Bad P3

4.3.8. MAKE_RSA_KEY

The MAKE_RSA_KEY command initializes the RSA key.

This command must be re-issued after the settings of RSA or RSA-CRT are changed.

Command:

CLA	INS	P1	P2	P3
0x80	0x06	0x00	0x00	0x00

Response:

SW1	SW2	Description
0x90	0x00	Successfully executed
0x6A	0x86	Bad P1 or P2
0x67	0x00	Bad P3

4.3.9. ENCRYPT_RSA

The ENCRYPT_RSA command executes RSA or RSA-CRT encryption and decryption.
A response is returned upon completion of encryption or decryption. Selection of the algorithm used for the encryption or decryption is made with P2.

Command:

CLA	INS	P1	P2	P3	Data
0x80	0x04	0x01	0x??	0x04	Plaintext (4-byte)

P2:

P2	Algorithm
0x00	RSA
0x01	RSA (equal timing)
0x02	RSA-CRT
0x03	RSA-CRT (equal timing)

Response:

SW1	SW2	Description
0x9F	0x04	Successfully executed (4-byte ciphertext is available)
0x6A	0x86	Bad P1 or P2
0x67	0x00	Bad P3

4.4. Other Common Commands

4.4.1. GET_RESPONSE

The GET_RESPONSE command returns the response of the last executed command.
This command always returns the P3 byte whether or not an error occurred. When an error did occur, the values of the response data are undefined.

Command:

CLA	INS	P1	P2	P3
0x80	0xC0	0x00	0x00	0x??

Response:

Data	SW1	SW2	Description
Data (0x?? byte)	0x90	0x00	Successfully executed
Undefined Data (0x?? byte)	0x6A	0x86	Bad P1 or P2
Undefined Data (0x?? byte)	0x67	0x00	Bad P3

4.4.2. NOP

The NOP command does nothing.

Command:

CLA	INS	P1	P2	P3
0x80	0x80	0x00	0x00	0x00

Response:

SW1	SW2	Description
0x90	0x00	Successfully executed
0x6A	0x86	Bad P1 or P2
0x67	0x00	Bad P3

4.4.3. ECHO

The ECHO command sets the size of the data made available in GET_RESPONSE.

Command:

CLA	INS	P1	P2	P3	Data
0x80	0x82	0x00	0x00	0x??	Data (0x?? byte)

Response:

SW1	SW2	Description
0x9F	0x??	Successfully executed (0x?? byte data is available)
0x6A	0x86	Bad P1 or P2
0x67	0x00	Bad P3

4.4.4. ATR

The ATR command enables GET_RESPONSE to receive the ATR.

Command:

CLA	INS	P1	P2	P3
0x80	0x84	0x00	0x00	0x00

Response:

SW1	SW2	Description
0x9F	0x??	Successfully executed (0x?? byte ATR is available)
0x6A	0x86	Bad P1 or P2
0x67	0x00	Bad P3

5. Command Sequences

The command sequences for each encryption and decryption algorithm are given below.

5.1. AES Encryption

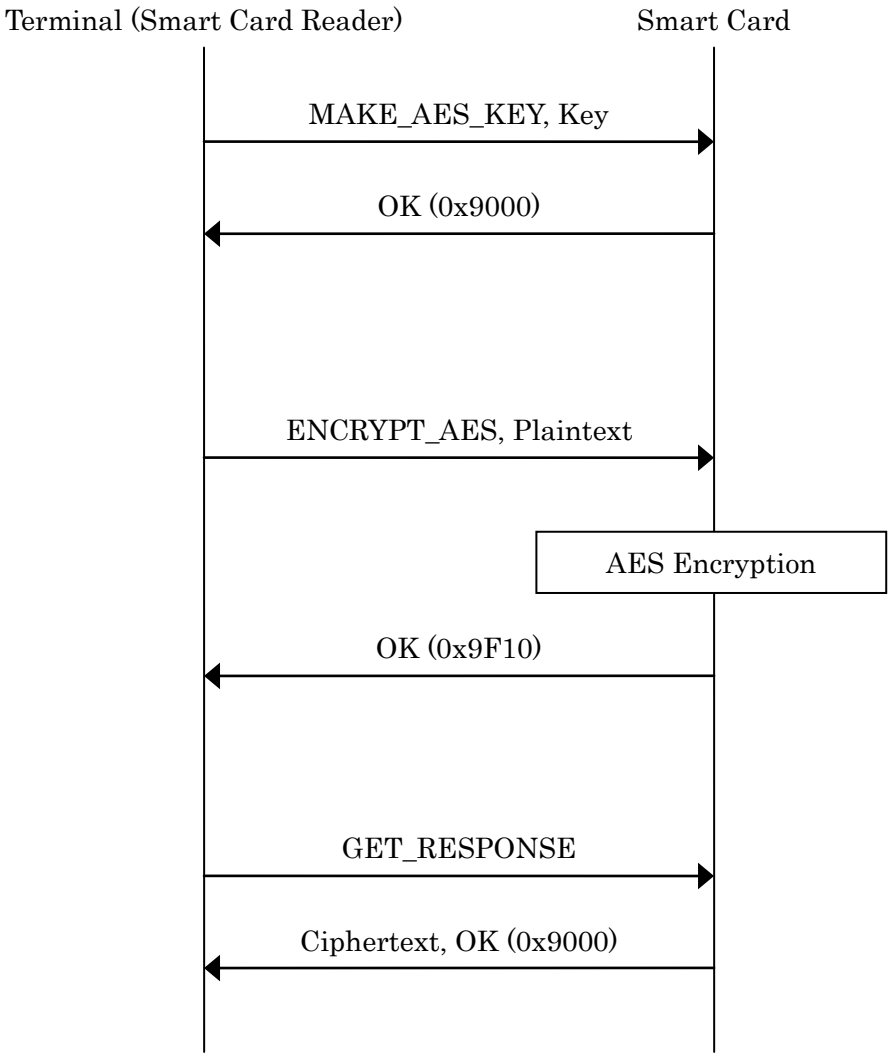

Figure 4： AES Encryption Sequence

5.2. AES Decryption

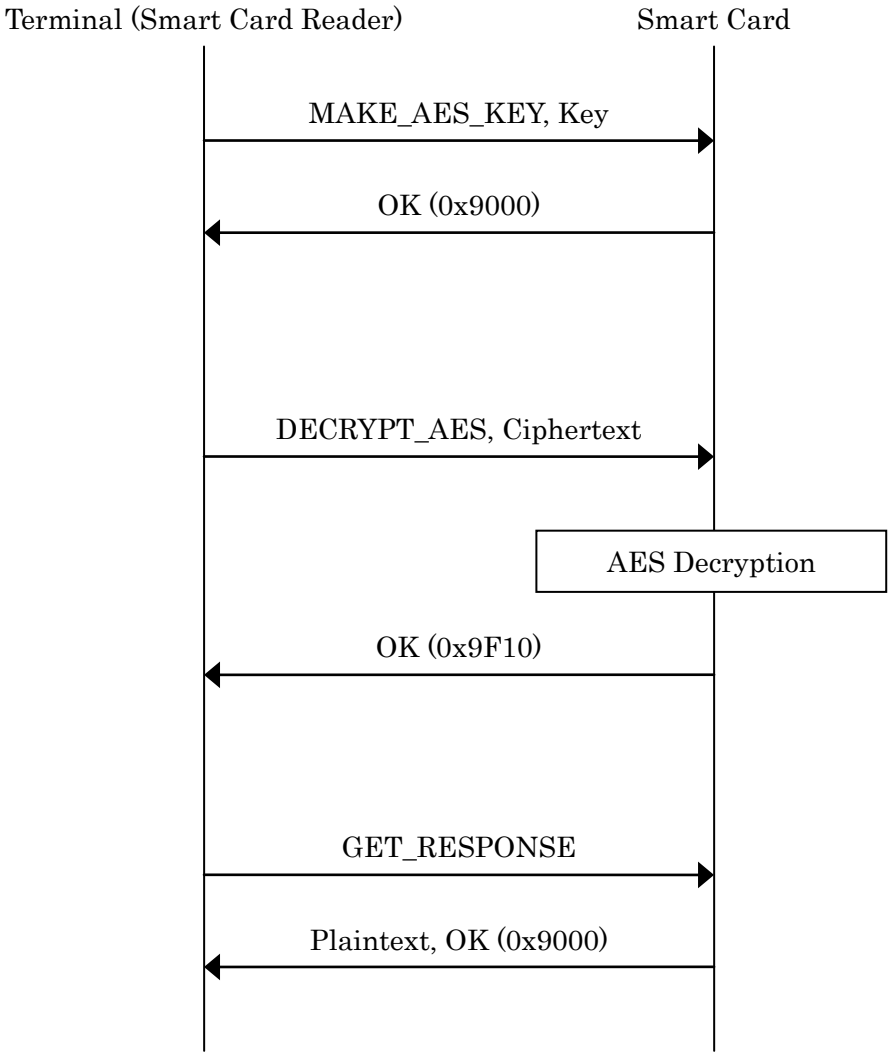

Figure 5： AES Decryption Sequence

5.3. DES Encryption

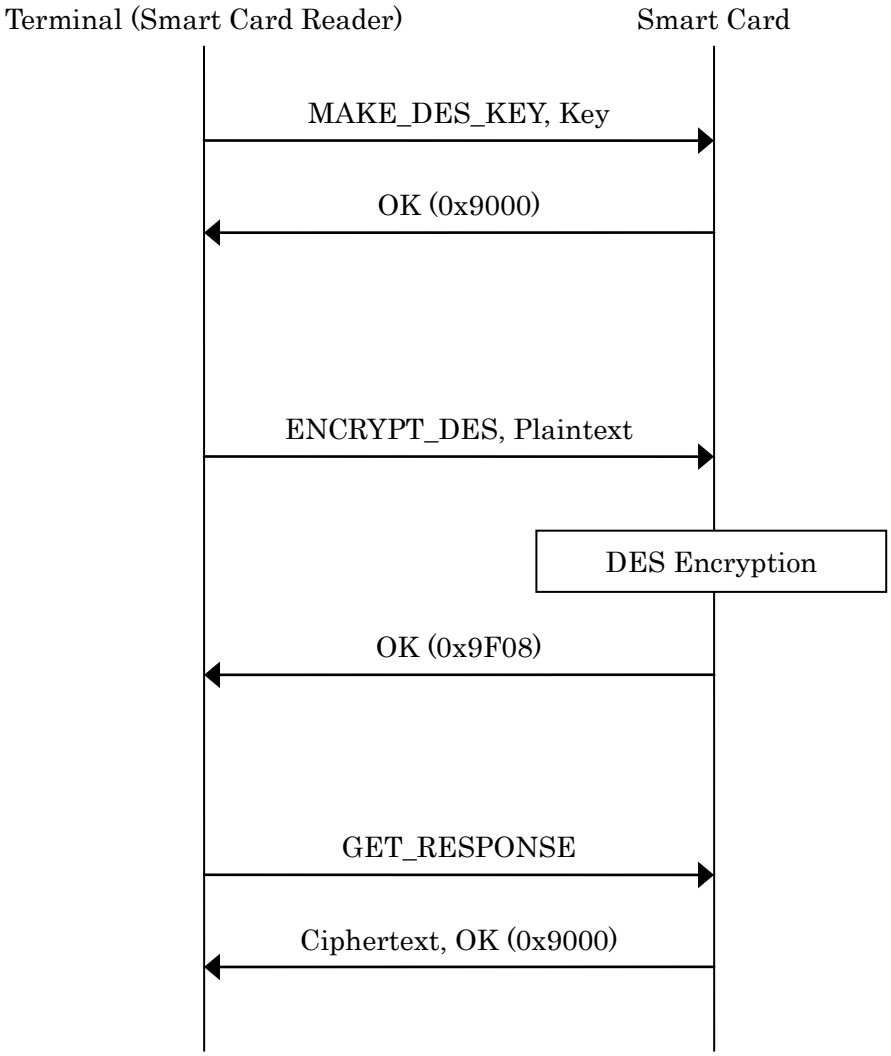

Figure 6： DES Encryption Sequence

5.4. DES Decryption

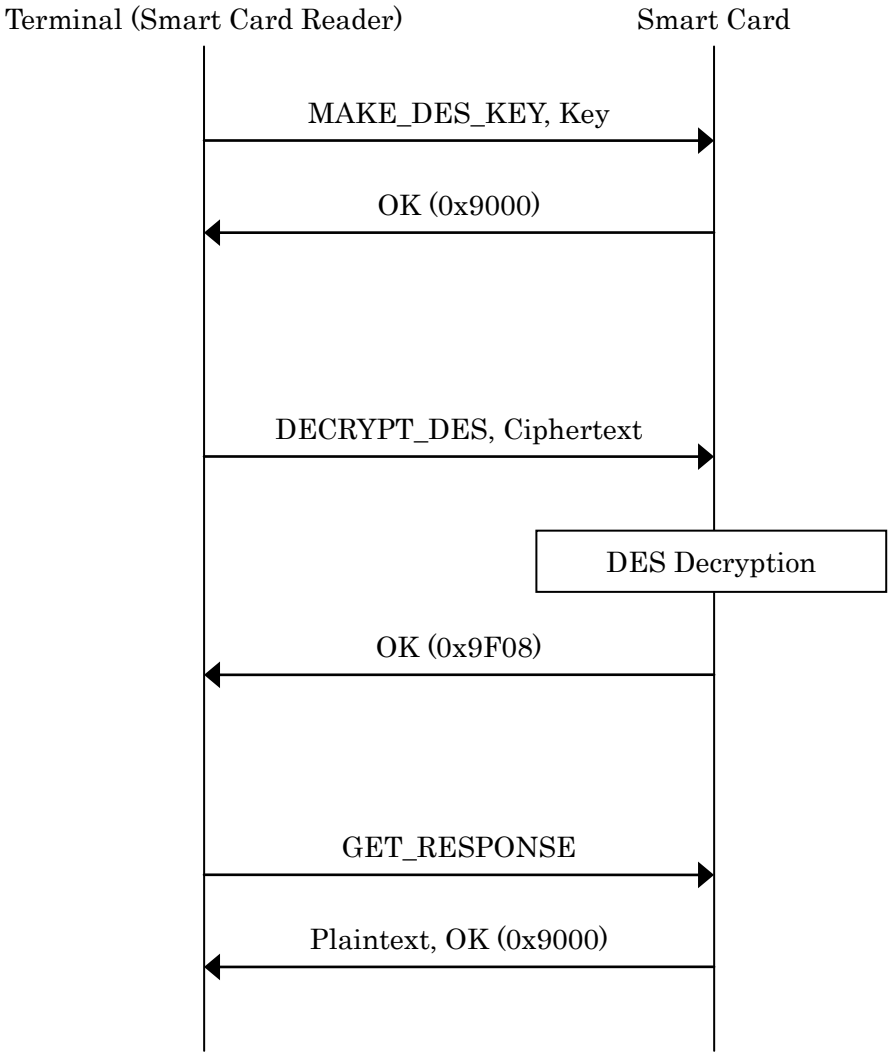

Figure 7： DES Decryption Sequence

5.5. RSA Encryption

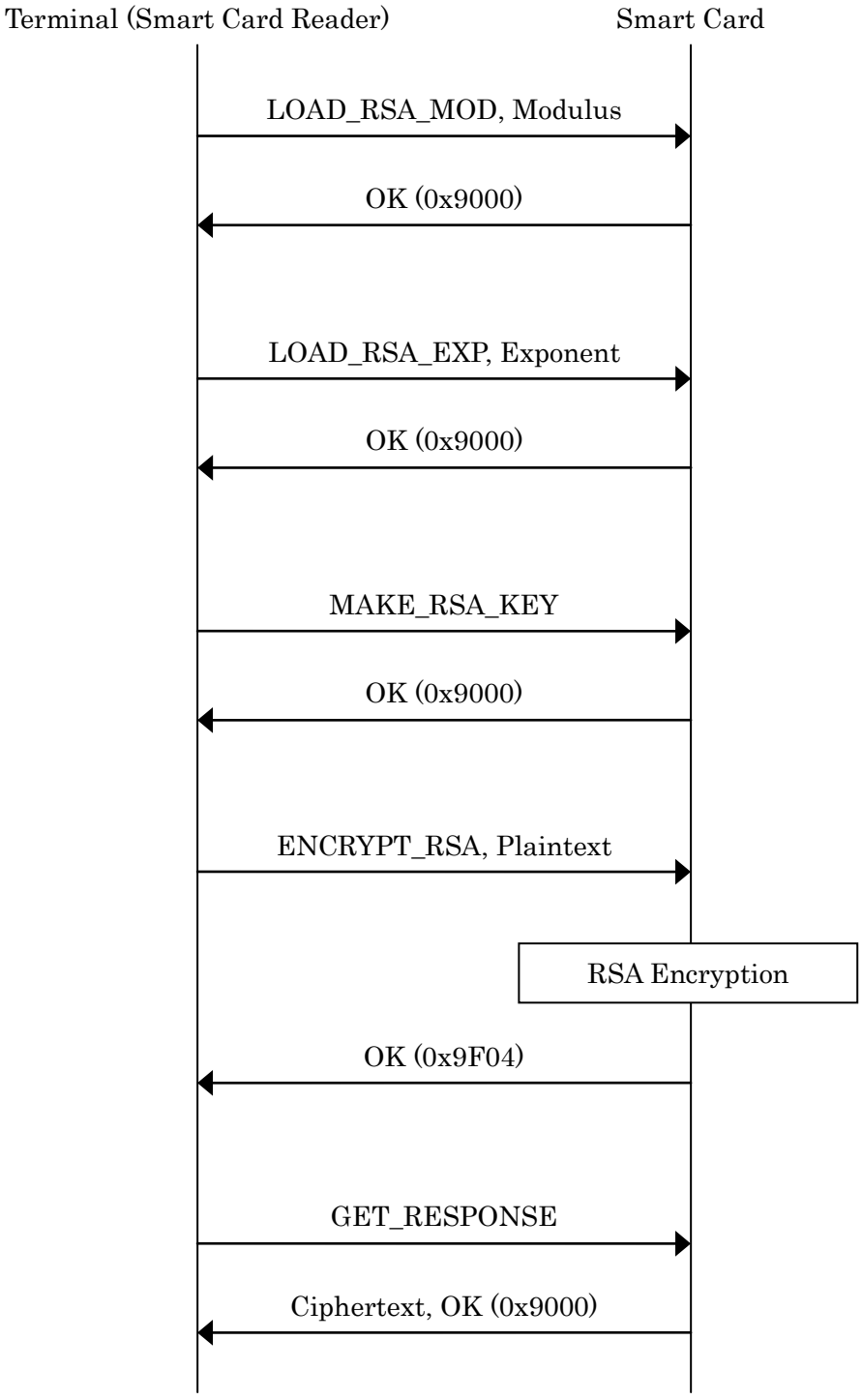

Figure 8： RSA Encryption Sequence

5.6. RSA-CRT Decryption

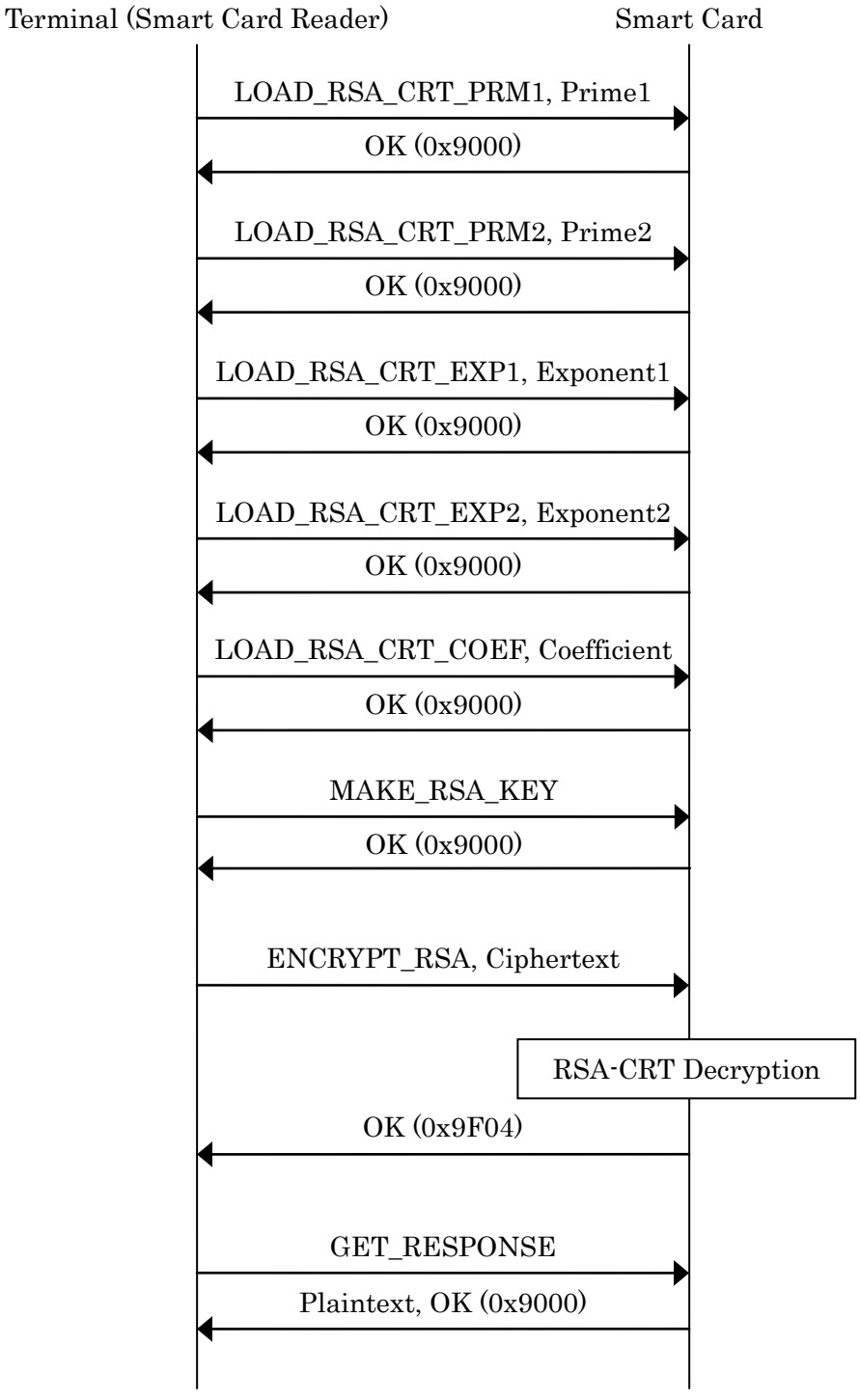

Figure 9 : RSA-CRT Decryption Sequence

6. Test Sequences

This command does nothing.

6.1. NOP

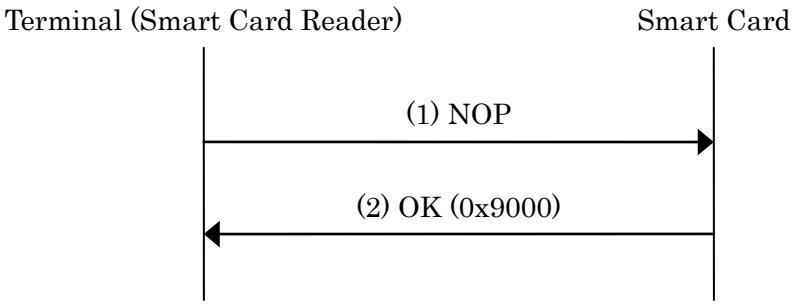

Figure 10： Test Sequence for NOP Command

Table 6： Sequence Data for NOP Command Test

Index	Data (hexadecimal)
(1)	80 80 00 00 00
(2)	90 00

6.2. ECHO

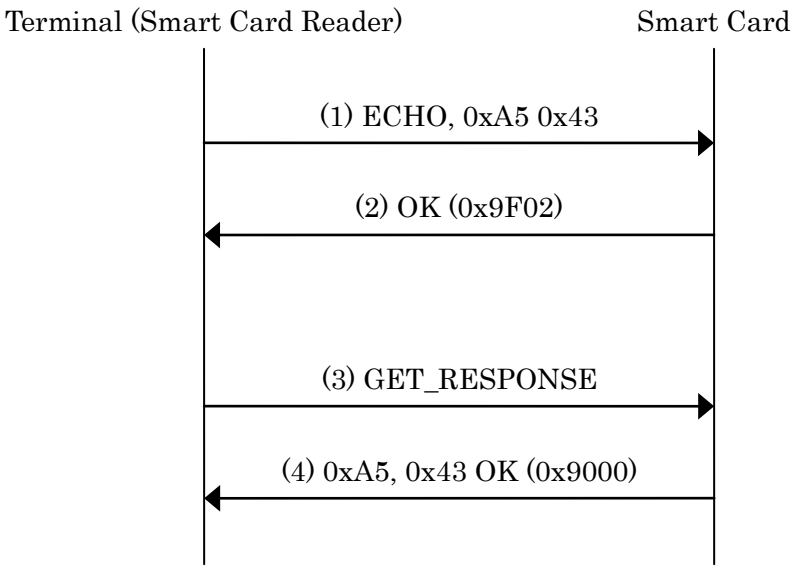

Figure 11： Test Sequence for ECHO Command

Table 7： Sequence Data of ECHO Command Test

Index	Data (hexadecimal)
(1)	80 82 00 00 02 A5 43
(2)	9F 02
(3)	80 C0 00 00 02
(4)	A5 43 90 00

6.3. AES Encryption

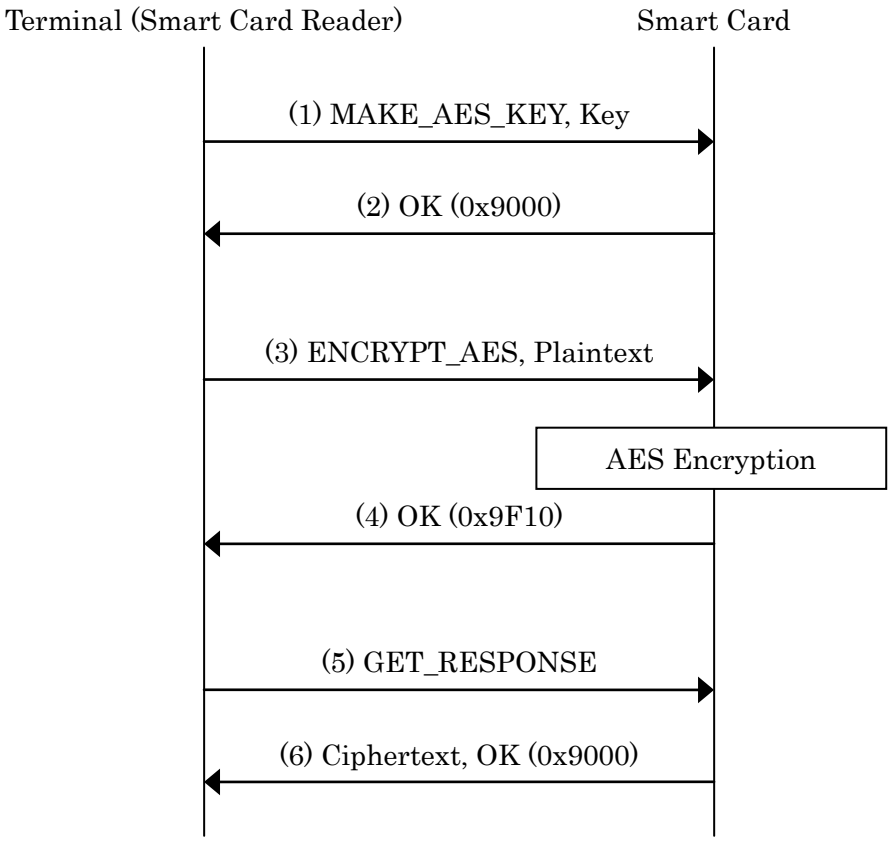

Figure 12： Test Sequence for AES Encryption

Table 8： Sequence Data for AES Encryption Test

Index	Data (hexadecimal)
(1)	80 12 00 00 10 00 01 02 03 04 05 06 07 08 09 0A 0B 0C 0D 0E 0F
(2)	90 00
(3)	80 04 04 00 10 76 2A 5A B5 09 29 18 9C EF DB 99 43 47 90 AA D8
(4)	9F 10
(5)	80 C0 00 00 10
(6)	00 11 22 33 44 55 66 77 88 99 AA BB CC DD EE FF 90 00

6.4. AES Decryption

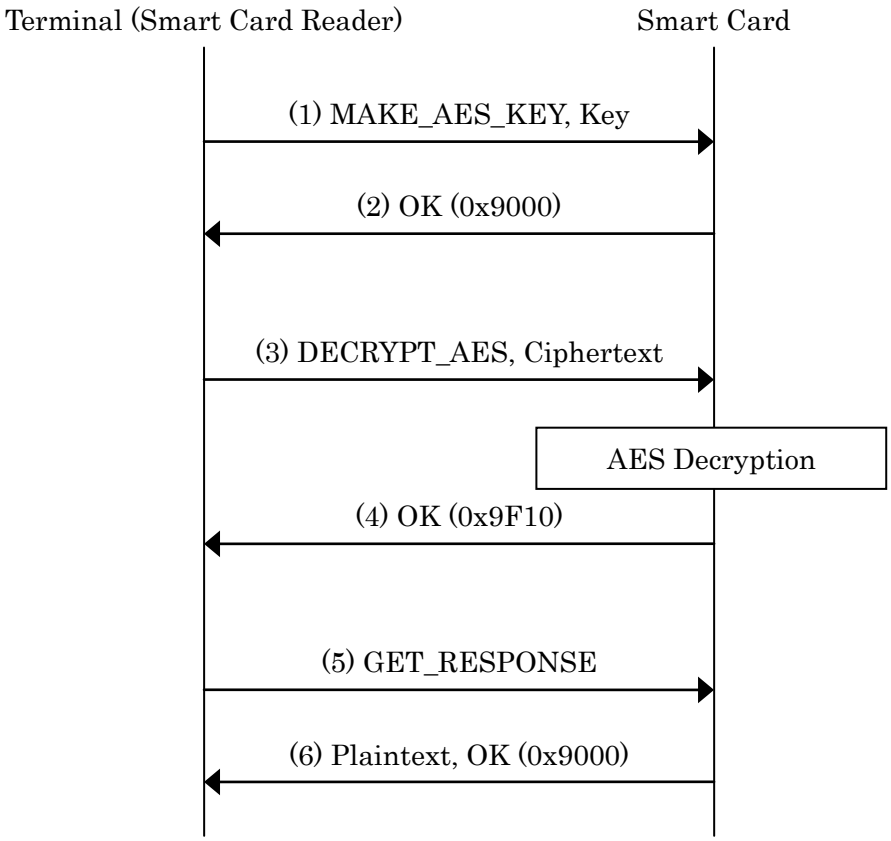

Figure 13： Test Sequence for AES Decryption

Table 9： Sequence Data for AES Decryption Test

Index	Data (hexadecimal)
(1)	80 12 00 00 10 00 01 02 03 04 05 06 07 08 09 0A 0B 0C 0D 0E 0F
(2)	90 00
(3)	80 08 04 00 10 1B 87 23 78 79 5F 4F FD 77 28 55 FC 87 CA 96 4D
(4)	9F 10
(5)	80 C0 00 00 10
(6)	FF EE DD CC BB AA 99 88 77 66 55 44 33 22 11 00 90 00

6.5. DES Encryption

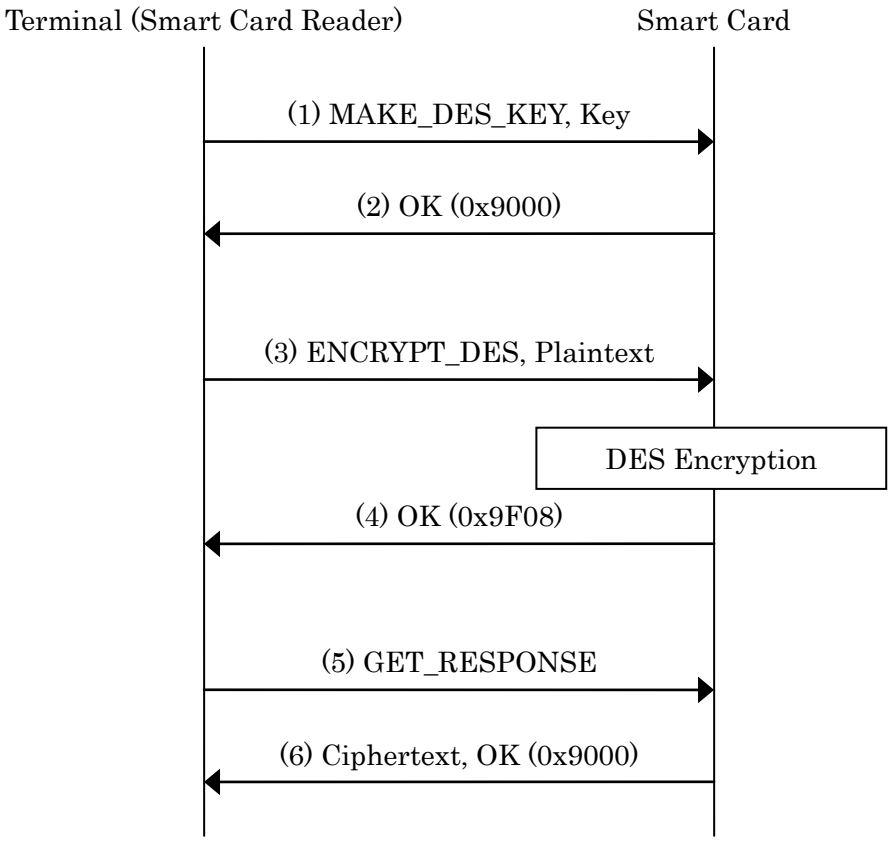

Figure 14： Test Sequence for DES Encryption

Table 10： Sequence Data for DES Encryption Test

Index	Data (hexadecimal)
(1)	80 0A 00 00 08 08 09 0A 0B 0C 0D 0E 0F
(2)	90 00
(3)	80 04 00 00 08 6B 11 84 37 ED 22 B9 FE
(4)	9F 08
(5)	80 C0 00 00 08
(6)	77 66 55 44 33 22 11 00 90 00

6.6. DES Decryption

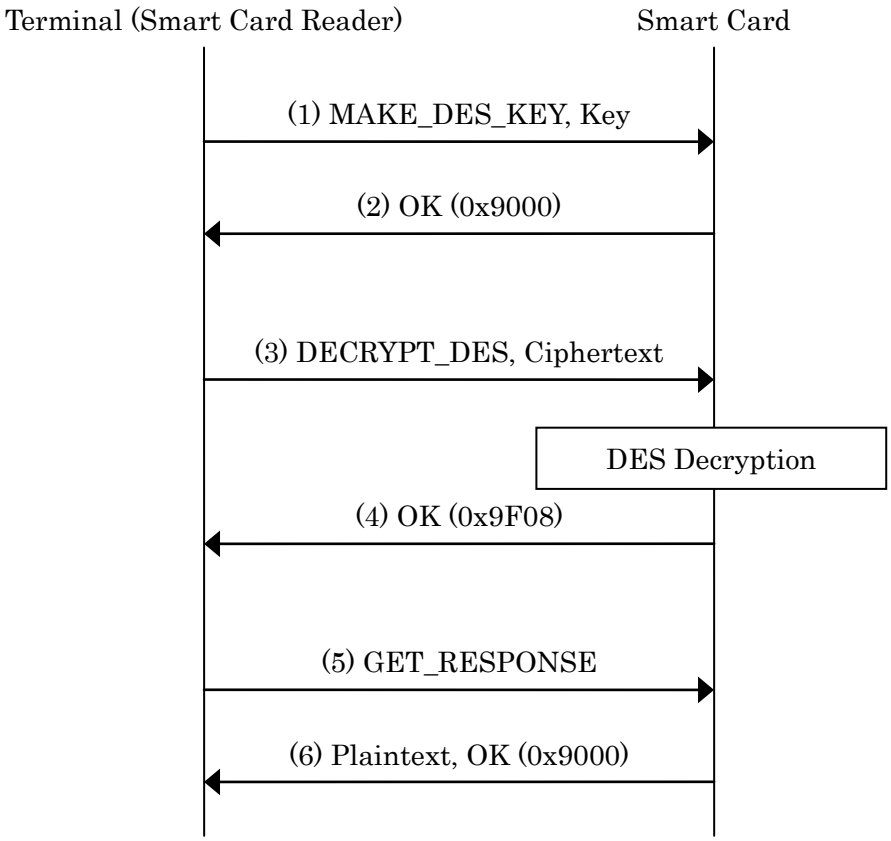

Figure 15： Test Sequence for DES Decryption

Table 11： Sequence Data for DES Decryption Test

Index	Data (hexadecimal)
(1)	80 0A 00 00 08 08 09 0A 0B 0C 0D 0E 0F
(2)	90 00
(3)	80 08 00 00 08 F7 C1 27 61 C9 AF E5 CB
(4)	9F 08
(5)	80 C0 00 00 08
(6)	88 99 AA BB CC DD EE FF 90 00

6.7. RSA Encryption

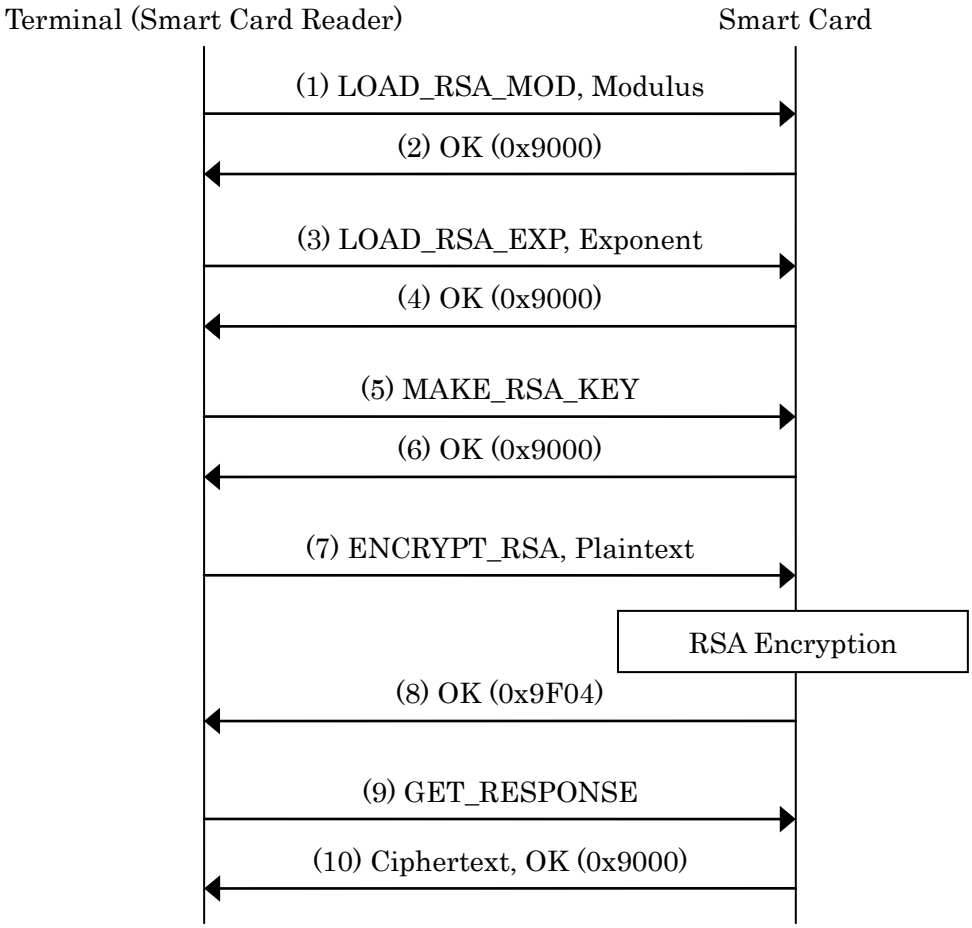

Figure 16： Test Sequence for RSA Encryption

Table 12： Sequence Data for RSA Encryption Test

Index	Data (hexadecimal)
(1)	80 02 00 00 04 C3 05 42 E9
(2)	90 00
(3)	80 00 00 00 04 00 01 00 01
(4)	90 00
(5)	80 06 00 00 00
(6)	90 00
(7)	80 04 01 00 04 21 19 2B 21
(8)	9F 04
(9)	80 C0 00 00 04
(10)	89 AB CD EF 90 00

6.8. RSA Decryption

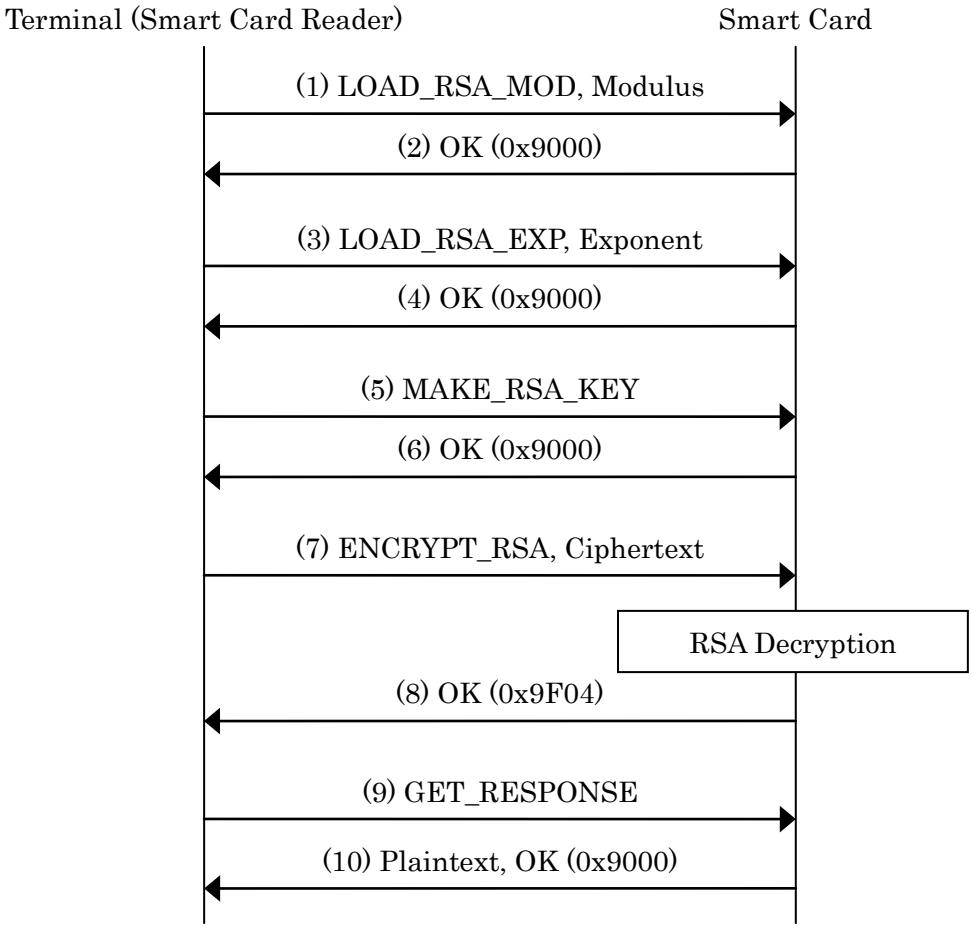

Figure 17： Test Sequence for RSA Decryption

Table 13： Sequence Data for RSA Decryption Test

Index	Data (hexadecimal)
(1)	80 02 00 00 04 C3 05 42 E9
(2)	90 00
(3)	80 00 00 00 04 B9 B1 AE 25
(4)	90 00
(5)	80 06 00 00 00
(6)	90 00
(7)	80 04 01 00 04 33 F1 64 F2
(8)	9F 04
(9)	80 C0 00 00 04
(10)	01 23 45 67 90 00

6.9. RSA-CRT Decryption

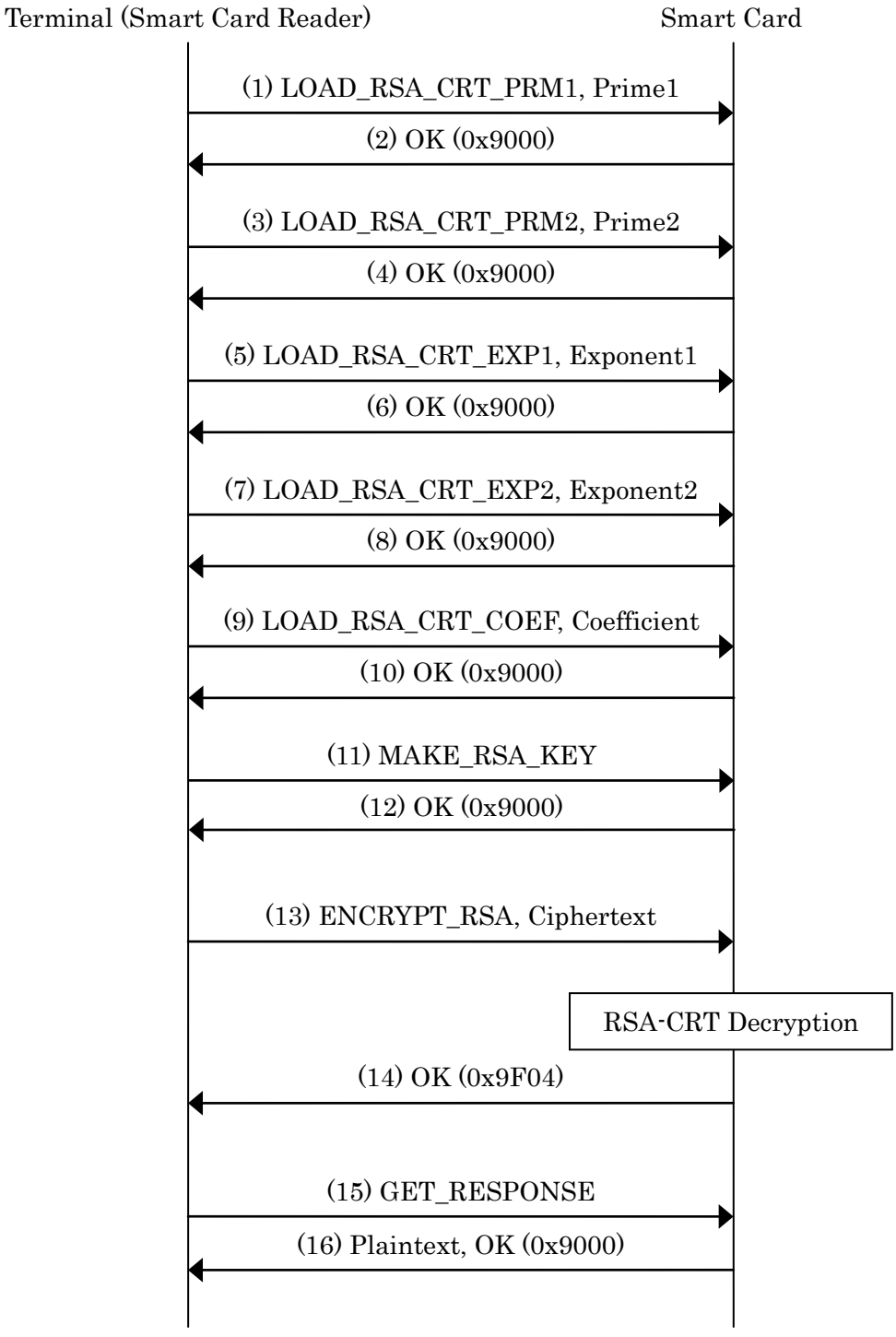

Figure 18： Test Sequence for RSA-CRT Decryption

Table 14： Sequence Data for RSA-CRT Decryption Test

Index	Data (hexadecimal)
(1)	80 02 01 00 02 E6 57
(2)	90 00
(3)	80 02 02 00 02 D8 BF
(4)	90 00
(5)	80 00 01 00 02 4D 39
(6)	90 00
(7)	80 00 02 00 02 05 CD
(8)	90 00
(9)	80 02 03 00 02 C0 26
(10)	90 00
(11)	80 06 00 00 00
(12)	90 00
(13)	80 04 01 02 04 33 F1 64 F2
(14)	9F 04
(15)	80 C0 00 00 04
(16)	01 23 45 67 90 00

7. Cryptographic Algorithms

Flowcharts for the cryptographic algorithms are shown below.

7.1. AES Encryption (Byte-oriented)

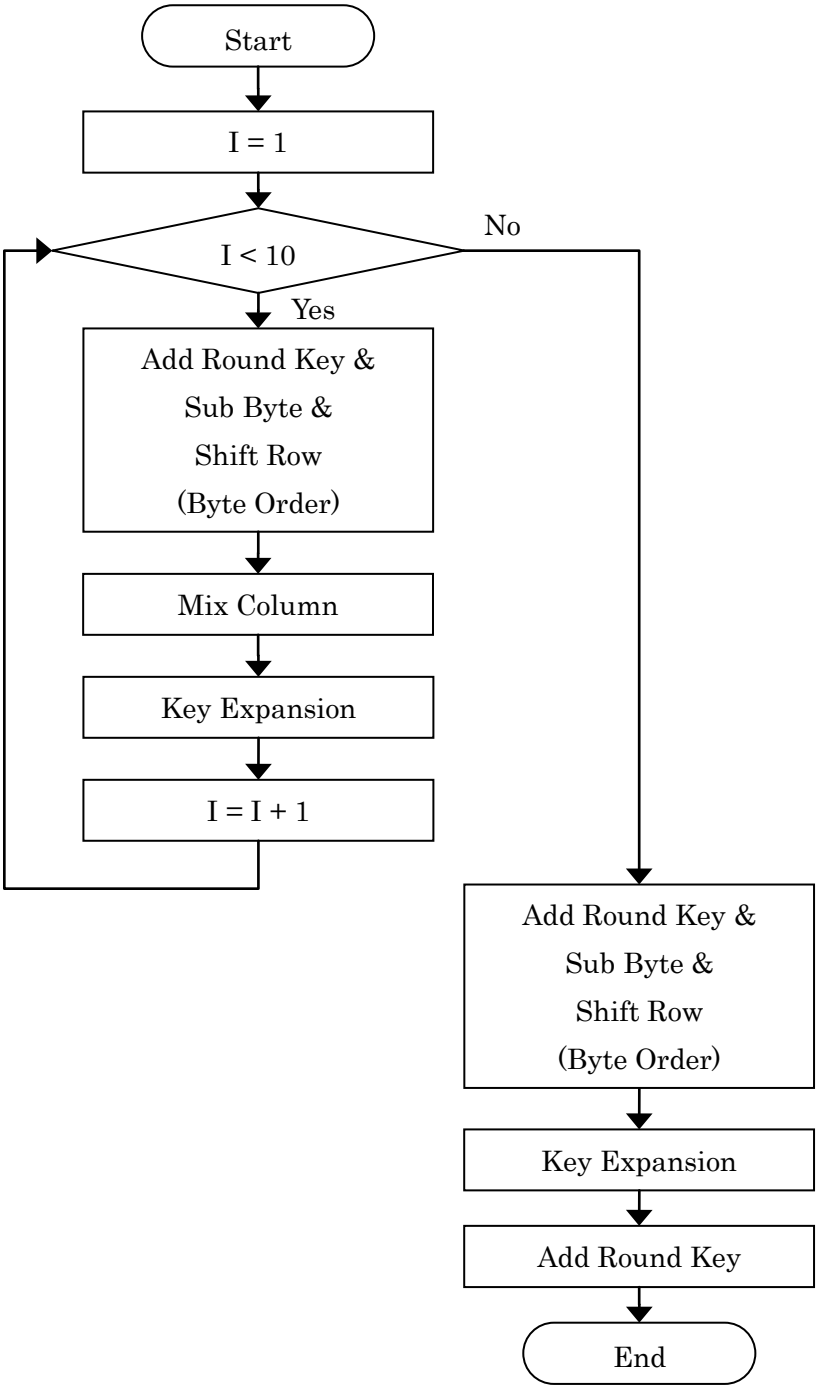

Figure 19 : AES Encryption Flowchart (Byte-oriented)

7.2. AES Decryption (Byte-oriented)

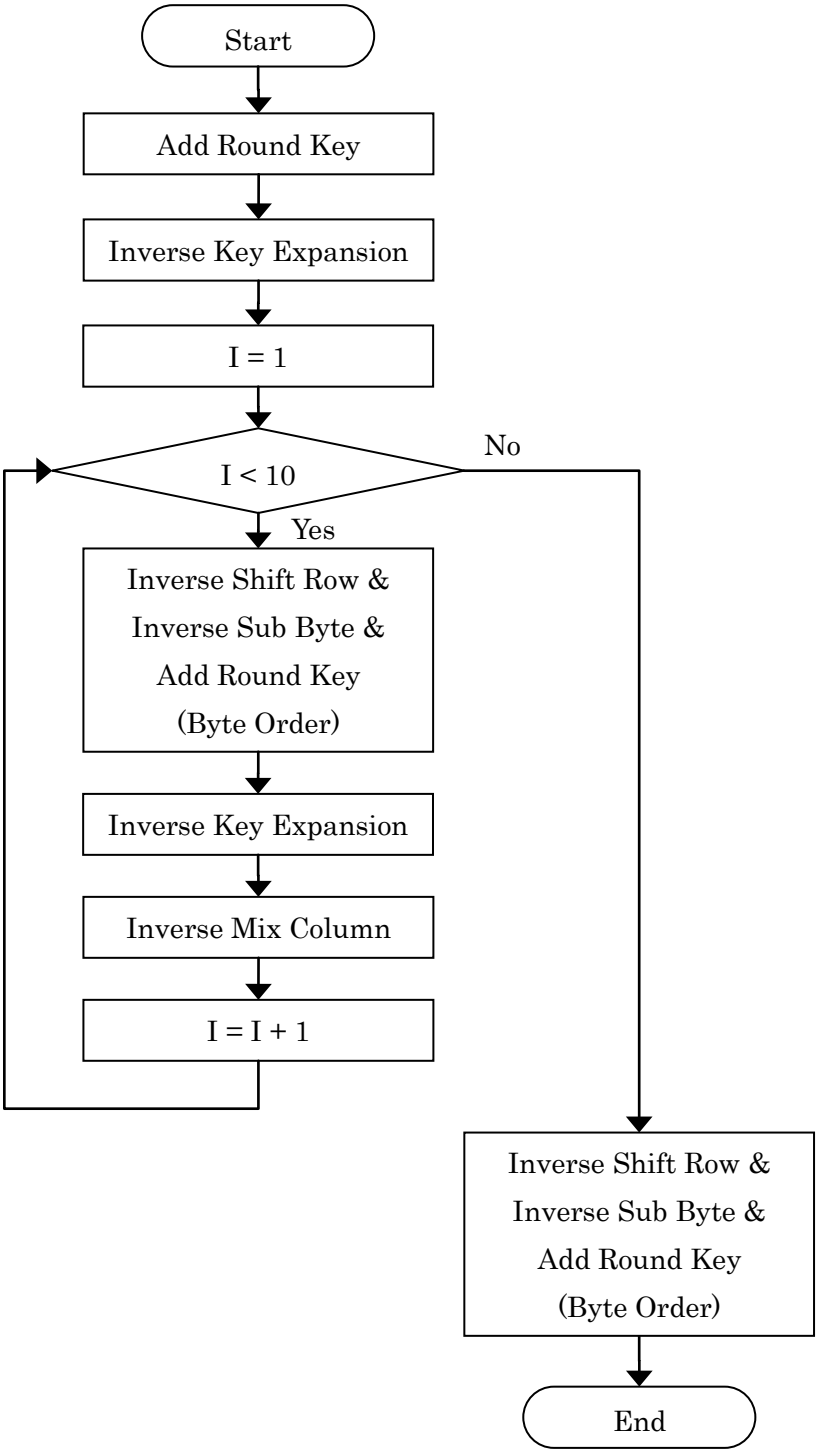

Figure 20 : AES Decryption Flowchart (Byte-oriented)

7.3. AES Encryption (Hardware-like)

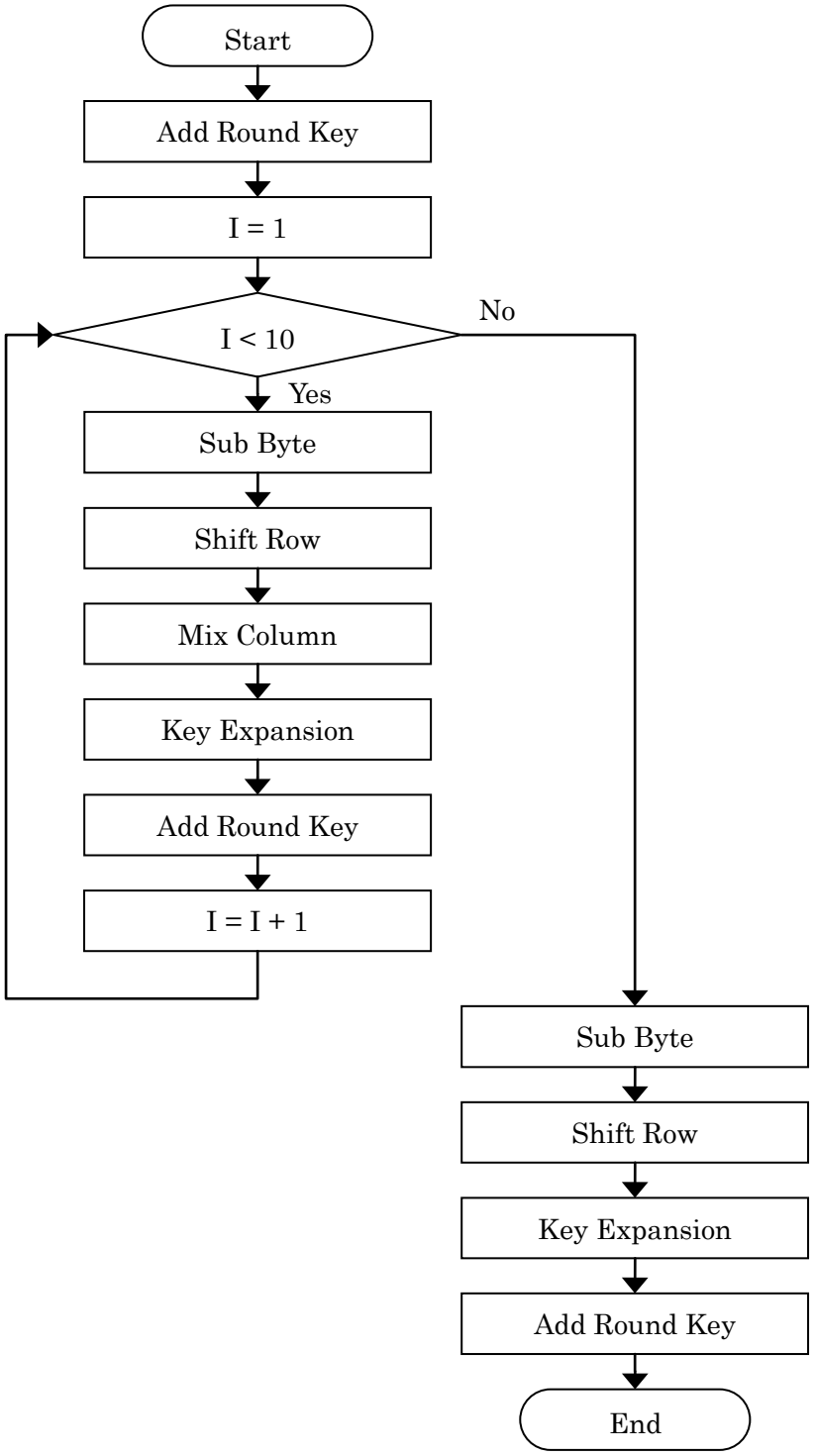

Figure 21： AES Encryption Flowchart (Hardware-like)

7.4. AES Decryption (Hardware-like)

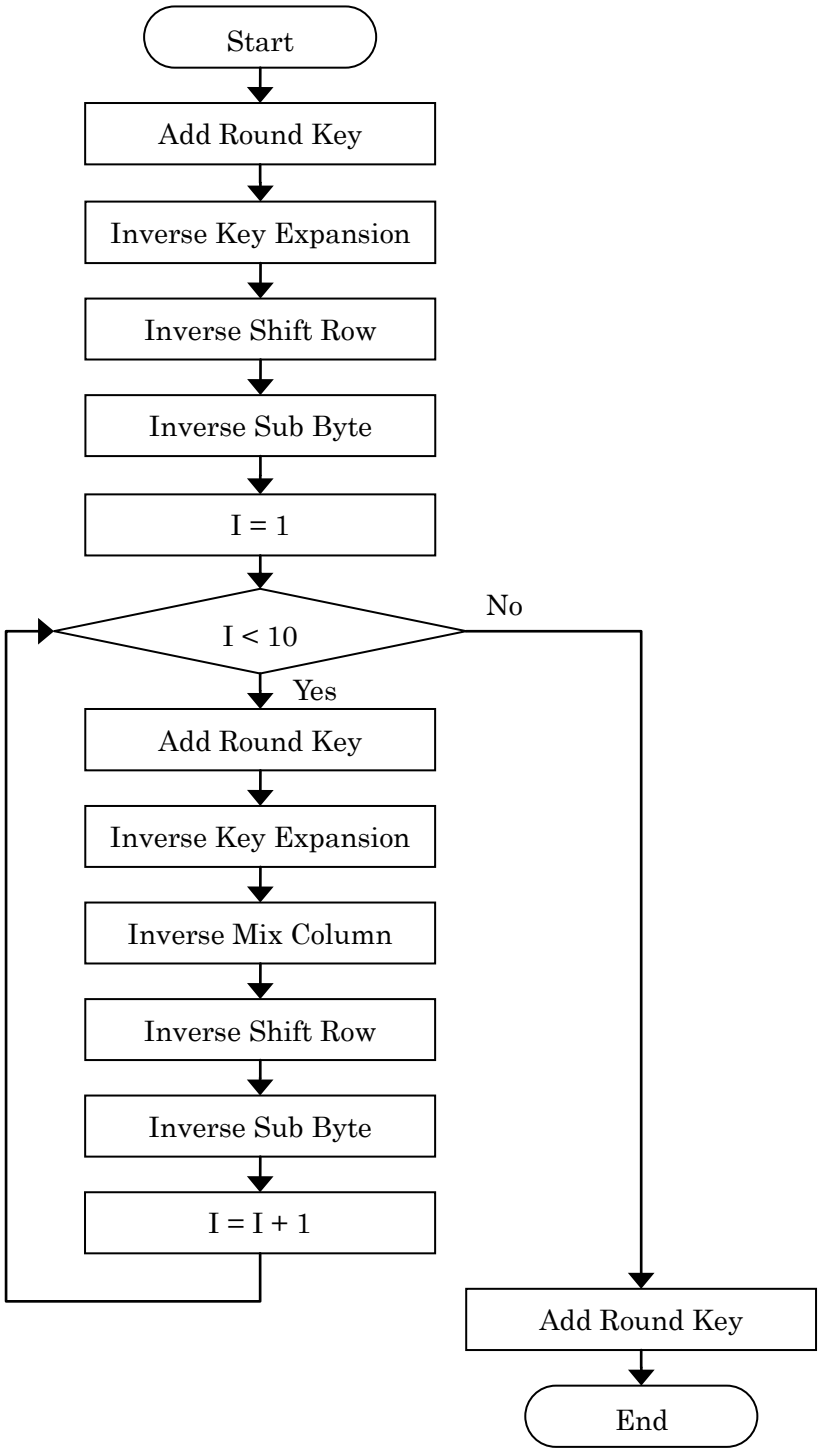

Figure 22： AES Decryption Flowchart (Hardware-like)

7.5. DES Encryption and Decryption

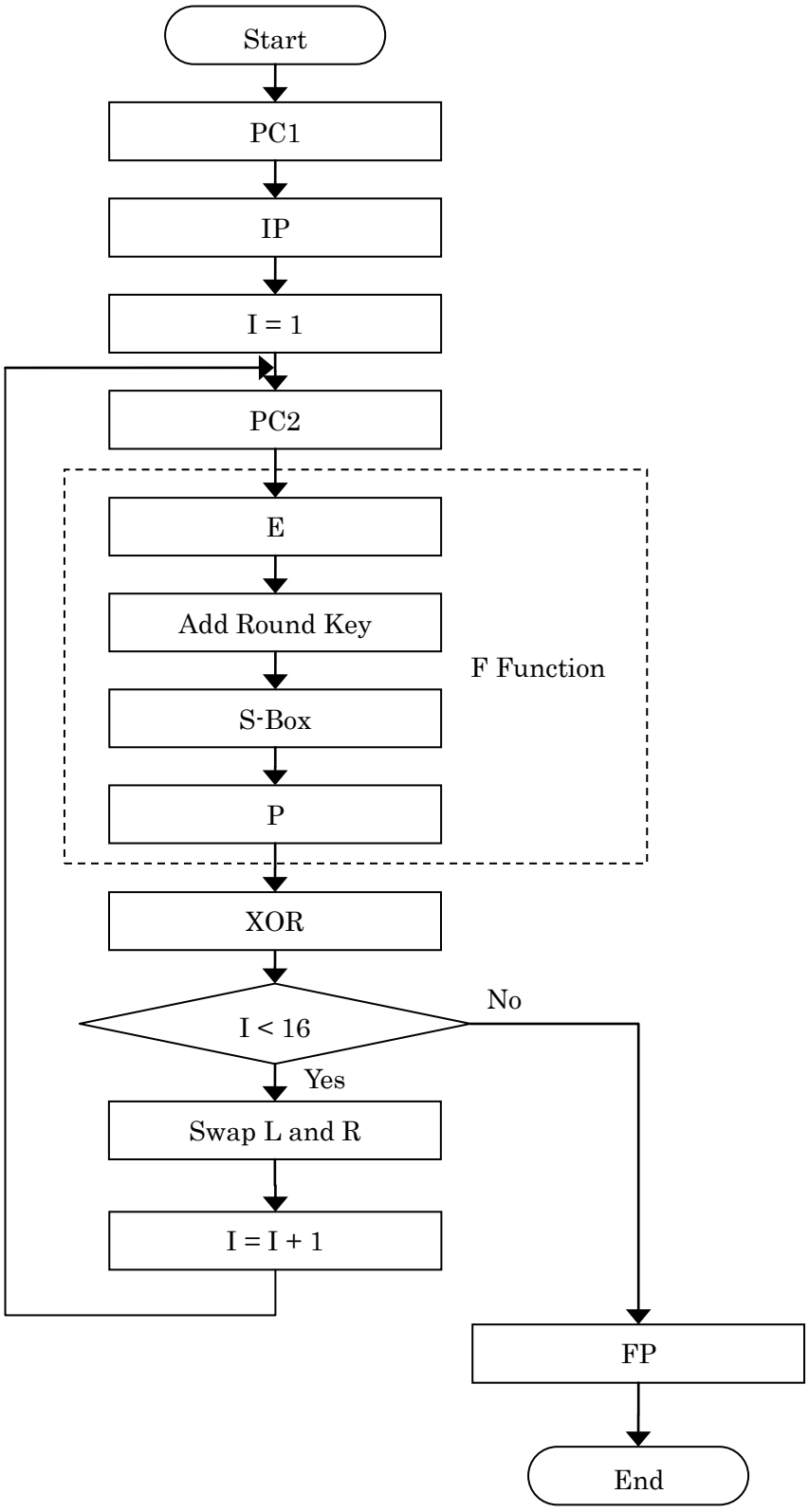

Figure 23： DES Encryption and Decryption Flowchart

7.6. RSA Encryption and Decryption

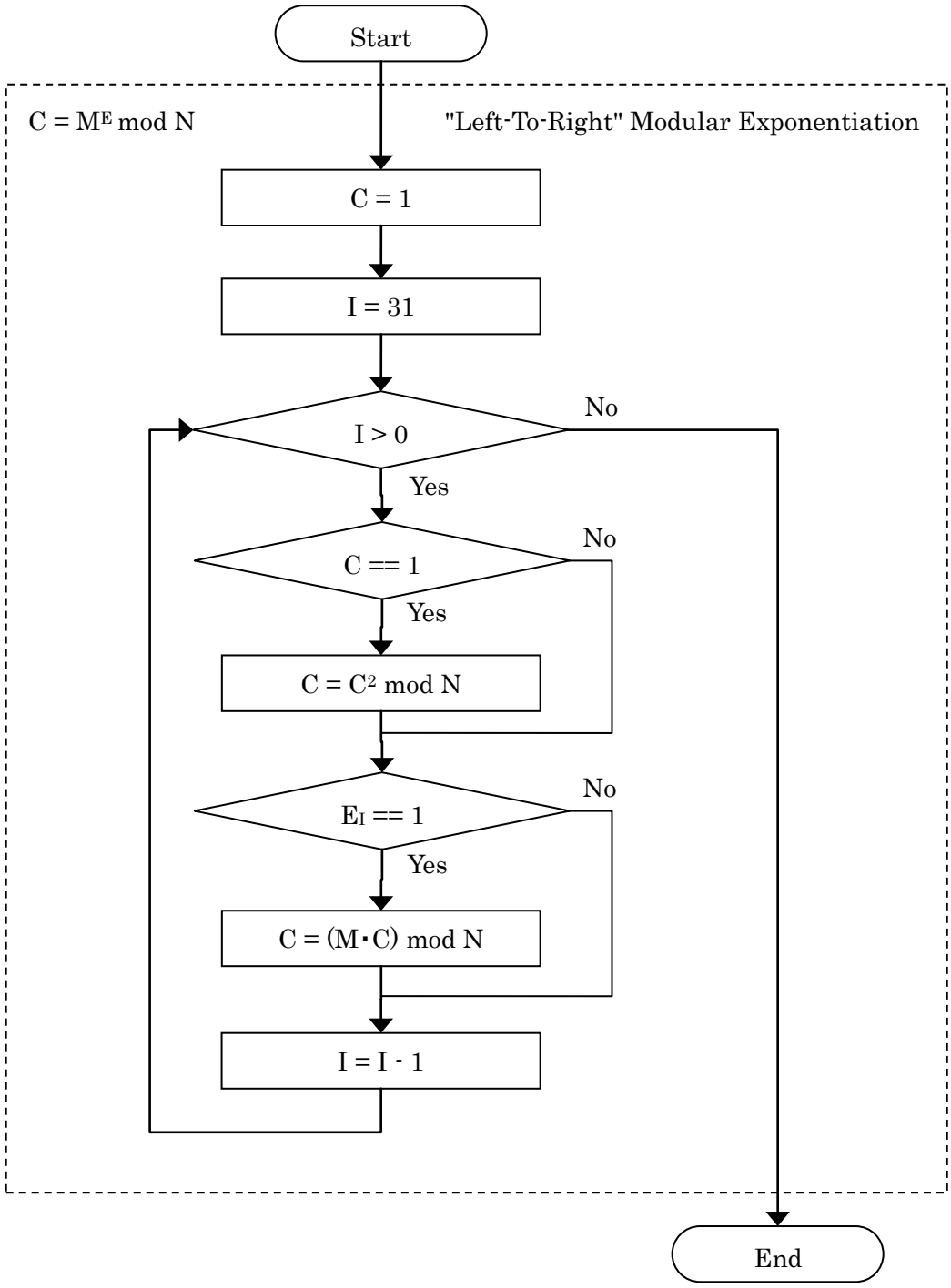

Figure 24： RSA Encryption and Decryption Flowchart

7.7. RSA Encryption and Decryption (Equal Timing)

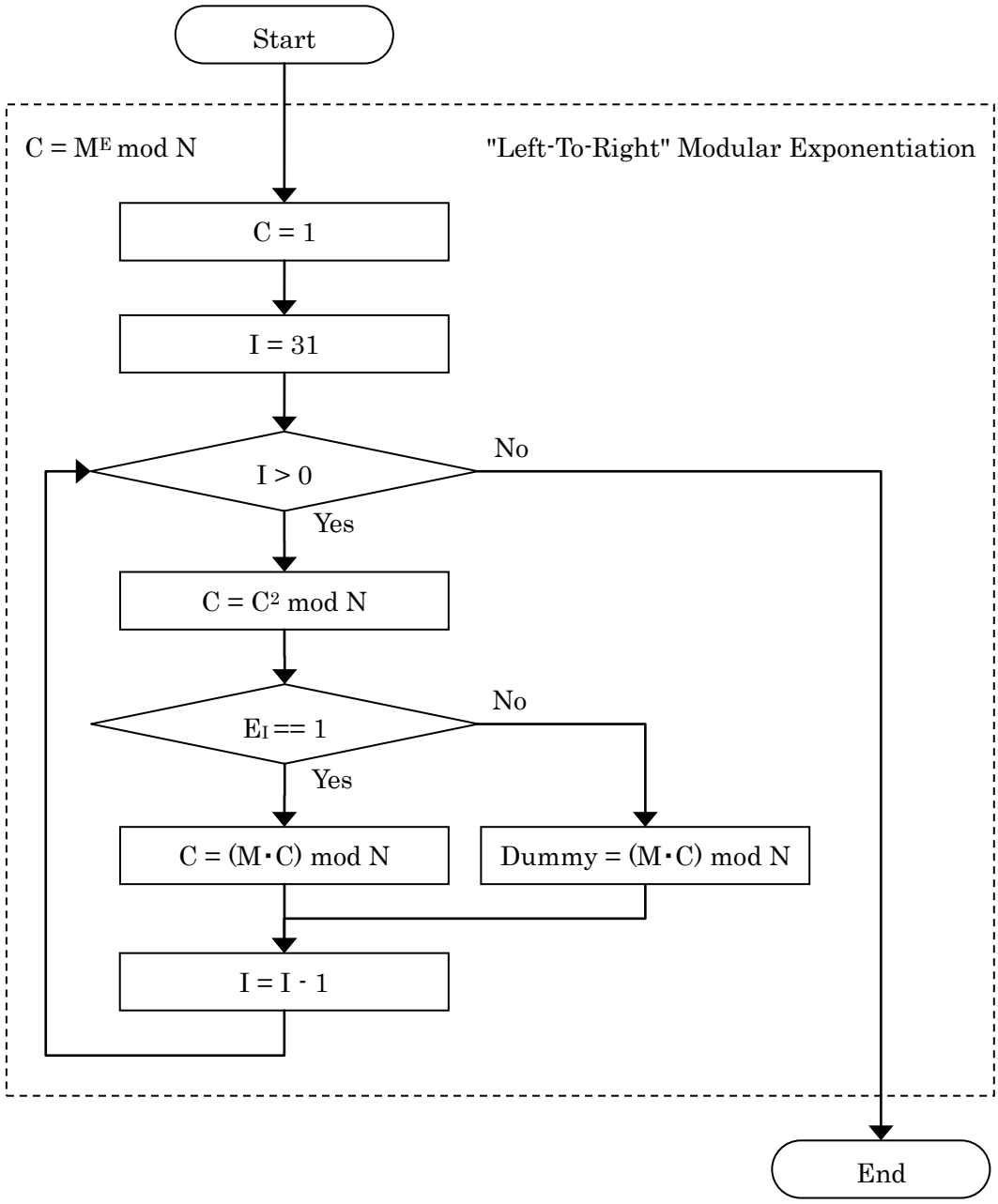

Figure 25： RSA Encryption and Decryption Flowchart (Equal Timing)

7.8. RSA-CRT Decryption

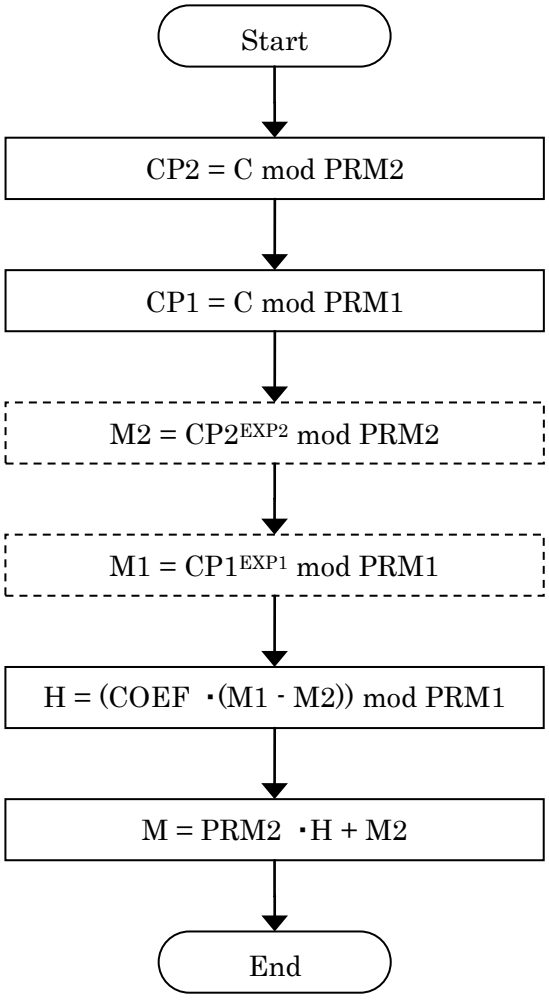

Figure 26： RSA-CRT Decryption Flowchart

The SASEBO-W Smart Card OS was developed by AIST while undertaking projects sponsored by the Japan Science and Technology Agency (JST)

Contact Us

Please send technical inquiries to**:**

National Institute of Advanced Industrial Science and Technology (AIST)
AIST Tsukuba Central 2 Room 2309
1-1-1 Umezono, Tsukuba-shi, Ibaraki 305-8568, Japan
TEL: +81-29-861-2979
FAX: +81-29-861-5285