Initializing the CryptoMemory® Device for Smart Card Applications

CryptoMemory provides a cost-effective tool to bring the security of data encryption to smart card applications. It is competitively priced and offers a simpler and quicker process of loading the device for use. Since the functions and security of CryptoMemory are built into the device, there is no operating system to load and no program to develop and load.

To prepare CryptoMemory for use, several registers are programmed to indicate the selected security features to be used, and the appropriate passwords and keys are loaded into the device. Very little programming is needed to initialize the device. Depending on the options selected, a maximum of 2 Kbits of the configuration zone might have to be programmed.

This application note describes the process of organizing data and determining security settings for the device. The proper sequence for writing to CryptoMemory is also described. In some cases, the application developer may have already determined the data and security settings to be used. The commands necessary to load this information are given and examples are included.

CryptoMemory®

Application Note

Gemplus Patents

Rev. 5025B–SMEM–02/04

Default Device Configuration

CryptoMemory devices are fully tested at Atmel whether delivered in module or wafer form. All functions are verified, and all memory locations are tested and then set to default values. These values are:

- User Zones – All user zones (4, 8, or 16) are programmed to all ones ($FF) throughout the entire zone.
- Configuration Zone: Answer to Reset – This field of the configuration zone is programmed to a preset value compliant with ISO 7816-3, PC/SC, and EMV standards. The value programmed includes two or three historical bytes that indicate the memory density.

Table 1. Low Density CryptoMemory ATR Values

Device	TS	T0	TA(1)	TB(1)	TD(1)	TA(2)	T1	T2
AT88SC0104C	$3B	$B2	$11	$00	$10	$80	$00	$01
AT88SC0204C	$3B	$B2	$11	$00	$10	$80	$00	$02
AT88SC0404C	$3B	$B2	$11	$00	$10	$80	$00	$04
AT88SC0808C	$3B	$B2	$11	$00	$10	$80	$00	$08
AT88SC1616C	$3B	$B2	$11	$00	$10	$80	$00	$16

Table 2. High Density CryptoMemory ATR Values

Device	TS	T0	TA(1)	TB(1)	TD(1)	T1	T2	T3
AT88SC3216C	$3B	$B3	$11	$00	$00	$00	$00	$32
AT88SC6416C	$3B	$B3	$11	$00	$00	$00	$00	$64
AT88SC12616C	$3B	$B3	$11	$00	$00	$00	$01	$28
AT88SC25616C	$3B	$B3	$11	$00	$00	$00	$02	$56

- Configuration Zone: Fab Code – This field is programmed to a preset value.
- Configuration Zone: Lot History Code – This field is programmed to a preset value. This field is locked and cannot be changed after leaving Atmel.
- Configuration Zone: Secure Code – This field, also known as the Write 7 Password, is programmed to a preset value.
- Remaining Configuration Zone – All remaining fields in the configuration zone are programmed to all ones ($FF). This includes all access and password key registers, all encryption keys, and all passwords except the secure code. In this configuration, access to all user zones is open and free.

Determine Initial Data and Security Settings

The first step in initializing CryptoMemory is to determine what data will be stored in the device and the structure for that data. Data may be thought of as different files, and these files will need to be organized within the various user zones of CryptoMemory. The security requirements for each file should be determined. Files with identical security requirements may be placed in the same user zone. If a file or group of files requires more than one user zone, multiple user zones may be set with the same security requirements to accommodate the large data files. In most cases, the application developer will determine the data structure and security settings for CryptoMemory.

Initial Data

With the data structure established, determine what data should be written to the device during initialization. This is typically data that needs to be in the device before the card is issued to the end user. In addition to the user zones, there is also a 4-byte card manufacturer code and a 16-byte issuer code that may be programmed in the configuration zone as part of initialization.

Security Settings

The next step is to determine the security requirements of the application and how each zone of the CyptoMemory needs to be protected. Each user zone has one access register and one password key register, allowing the security requirements for each zone to be set independently. If multiple zones are required to store large data files, they may be set to the same security requirements. CryptoMemory offers a variety of security options; these options can be explored using the CryptoMemory Evaluation Kit (AT88SC25616C-EK). Security options are also documented in the CryptoMemory Detailed Specification, available under NDA. Briefly, the available security options are:

- Open or free access – no restrictions for read or write of a user zone
- One-time programmable – the data programmed during initialization is locked and may not be changed.
- Program only – the data programmed during initialization may only be changed from a logic value “1” to logic “0” on a bit-by-bit basis. This can be used for a countdown function.
- Write protect – the data programmed during initialization may be protected byte by byte within a user zone.
- Password protection – separate passwords may be required for read and/or write privileges to the user zone; eight separate password sets are available.
- Authentication protection – a successful authentication sequence is required for read and/or write privileges to the user zone; four separate key sets are available.
- Encryption required – after a successful authentication, data is required to be encrypted for both read and write operations to the user zone.

The above list is not exhaustive, and various combinations of these security options may be implemented.

Writing to CryptoMemory

Once the data structure, initial data, and security settings have been established, this information must be written into the CryptoMemory device to complete initialization. This is accomplished by following the sequence below. Once this process is completed and the security fuses are set, the device is ready to issue to the end user. Details of the commands used and a simple example are included in the *Asynchronous Communications* section on page 5.

Write Data to User Zones

In the default configuration from Atmel, all user zones have free access rights. Writing initial data into the user zones should be done before setting security configurations. Use the *Set User Zone* command and *Write User Zone* command to write initial data into the user zones. The *Read User Zone* command may be used to verify the data written.

Unlock Configuration Zone

Before any data can be written to the configuration zone, it must be unlocked by presenting the correct security code (Write 7 Password). Use the *Verify Secure Code* command with the proper password supplied by Atmel to unlock the configuration zone. A status word indicating “command successfully executed” is confirmation that the proper security code was sent and that access is now granted to the configuration zone.

Write Data to Configuration Zone

Data contained in the configuration zone is divided into four types of information:

CODES: These include the Answer to Reset value, fabrication zone, identification number, card manufacturer zone, and issuer zone. The Answer to Reset value is used by the CryptoMemory device when responding to a valid cold or warm reset. The other codes may be used to store information regarding fabrication, card manufacturer, or issuing of the cards. This information may be used by the application as determined by the application developer. The Answer to Reset and Fabrication Zone codes are programmed to default values by Atmel but may be modified. These codes are all write-protected once all security fuses have been set.

REGISTERS: For each user zone, there is one 8-bit access register and one 8-bit password/key register. These registers are used to define the security requirements for each user zone. In addition, there is one device configuration register that determines several options affecting the security of all user zones globally.

KEYS: There are four sets of keys (Secret Seeds) used for authentication and four sets of keys (Session Keys) used for encryption; each key is 64 bits long. There are also four initial cryptogram values that may be defined and one identification number used with all four sets of keys.

PASSWORDS: There are eight sets of passwords; each set consists of a read access password and a write access password.

Writing this data is accomplished by performing the *Write Configuration Zone* command at the appropriate location. The *Read Configuration Zone* command may be used to verify the data written. Please consult the CryptoMemory Detailed Specification for proper memory address within the configuration zone for each data field. As soon as values are written to the registers, keys, and passwords, they become effective in determining the security of the user zones.

Set Security Fuses

Three security fuses protect the information contained in the configuration zone.

FAB FUSE: This fuse protects the Answer to Reset value and the fabrication zone. These areas are write-protected and cannot be changed once the fab fuse is set.

CMA FUSE: This fuse protects the card manufacturer zone. This area is write-protected and cannot be changed once the CMA fuse is set.

PER FUSE: This fuse protects the rest of the configuration zone. The issuer code is write-protected and cannot be changed once the PER fuse is set. All registers, keys, and passwords are write-protected once the PER fuse is set. Modifications to these areas are only permitted by the internal logic; further detail may be found in the CryptoMemory Detailed Specification. One exception is the eight password sets. Once a write password is properly presented, only the read and write password in that set can be modified by using the *Write Configuration Zone* command. Once the PER fuse is set, the secure code (Write 7 Password) will no longer give access to the configuration zone.

These three fuses must be set in the order listed above (FAB, then CMA, then PER). The *Write Fuse* command is used to set each of the three fuses individually. The P2 value for setting each fuse is shown in Table 3 below.

Table 3. Write Fuse – Fuse ID

Fuse	P2
FAB	$06
CMA	$04
PER	$00

The *Read Fuse Byte* command may be used at any time to check the status of all three fuses.

Table 4. Read Fuse Byte – Returned Data

Fuse Condition	Fuse Byte	
No Fuses Set	$07	0111
FAB Set	$06	0110
FAB, CMA Set	$04	0100
FAB, CMA, PER Set	$00	0000

Asynchronous Communications

CryptoMemory communicates using an asynchronous protocol compliant to ISO 7816-3. A variety of equipment from PC/SC card readers to high-throughput testers may be used to program CryptoMemory using this standard protocol. Commands transmitted by CryptoMemory must be in the ISO 7816-3 standard Transmission Protocol Data Unit (TPDU) format (see Annex 1 of ISO 7816-3). The command set required to perform all functions of programming CryptoMemory is listed in Table 5 below.

Table 5. Initialization TPDU Command Set

		CLA	INS	P1	P2	P3	Data (N)
Write User Zone	(Devices 0104C–1616C)	$00	$B0	addr	addr	N ≤ $10	N bytes
	(Devices 3216C–25616C)	$00	$B0	addr	addr	N ≤ $80	N bytes
Read User Zone		$00	$B2	$00	addr	N	
System Write	Write Config Zone (Devices 0104C–1616C)	$00	$B4	$00	addr	N ≤ $10	N bytes
	Write Config Zone (Devices 3216C–25616C)	$00	$B4	$00	addr	N ≤ $80	N bytes
	Write Fuse	$00	$B4	$01	fuse ID	$00	
	Set User Zone	$00	$B4	$03	zone	$00	
System Read	Read Config Zone	$00	$B6	$00	addr	N	
	Read Fuse Byte	$00	$B6	$01	$00	$01	
Verify Secure Code		$00	$BA	$07	$00	$03	3-byte password

Following each command, the device will return two status words. Values supported by CryptoMemory and used in the initialization process are listed in Table 6.

Table 6. Status Word Values

SW1SW2	Meaning
$67 $00	The length in incorrect.
$69 $00	The command is unauthorized.
$6D $00	The instruction code is invalid.
$90 $00	The command was successfully executed.

Initialization Example Using TPDU Commands

The AT88SC0104C is used for this example. A small pattern is written into each of the four user zones. Security for each of the four user zones and the associated register values are shown in Table 7. Simple values for codes, keys, and passwords are used.

Table 7. Security Settings – AT88SC0104C Example

User Zone	Data	Security Requirements	Access Register	Password/Key Register
0	Zone 0	None	$FF	$FF
1	Zone 1	Read/Write Password (Set 1)	$7F	$F9
2	Zone 2	Read/Write Authentication (Set 2)	$DF	$BF
3	Zone 3	Read/Write Password (Set 1) Read/Write Authentication (Set 1) with Encryption Required	$57	$B9

The following example shows the TPDU commands sent to the CryptoMemory device for the purpose of initializing the device for use in the field. The flow is consistent with the steps described in this application note, and comments have been added.

```
*AT88SC0104C Initialization Example
power_on
*WRITE DATA TO USER ZONES
*Set User Zone 0
00 B4 03 00 00
*Write data = Zone 0 Data
00 B0 00 00 0B 5A 6F 6E 65 20 30 20 44 61 74 61
*Set User Zone 1
00 B4 03 01 00
*Write data = Zone 1 Data
00 B0 00 00 0B 5A 6F 6E 65 20 31 20 44 61 74 61

*Set User Zone 2
00 B4 03 02 00
*Write data = Zone 2 Data
00 B0 00 00 0B 5A 6F 6E 65 20 32 20 44 61 74 61

*Set User Zone 3
00 B4 03 03 00
*Write data = Zone 3 Data
00 B0 00 00 0B 5A 6F 6E 65 20 33 20 44 61 74 61

*UNLOCK CONFIGURATION ZONE
00 BA 07 00 03 FF FF FF

*WRITE CODES IN CONFIGURATION ZONE

*Write Card Mfg Code = P001
00 B4 00 0B 04 50 30 30 31
*Write Identification Number = 00000000012345
00 B4 00 19 07 00 00 00 00 01 23 45
*Write Issuer Code = STATION 035
```

```
00 B4 00 40 10 53 54 41 54 49 4F 4E 20 30 33 35 00 00 00 00 00

*WRITE REGISTERS IN CONFIGURATION ZONE

*Write Registers AR1/PR1 = 7F F9, AR2/PR2 = DF BF, AR3/PR3 = 57 B9
00 B4 00 22 06 7F F9 DF BF 57 B9

*WRITE KEYS IN CONFIGURATION ZONE

*Write Ci for set 2 = 22222222222222
00 B4 00 71 07 22 22 22 22 22 22 22

*Write Gc for set 2 = 5B4F9AE4B5098BE7
00 B4 00 A0 08 5B 4F 9A E4 B5 09 8B E7

*WRITE PASSWORDS IN CONFIGURATION ZONE

*Write Passwords, read 7 = 10 00 01, write 7 = 11 00 11
00 B4 00 B9 07 11 00 11 FF 10 00 01

*READ ENTIRE CONFIGURATION ZONE TO VERIFY
00 B6 00 00 F0

Response:
3B B2 11 00 10 80 00 01 10 10 FF 50 30 30 31 FF
8C AD A8 10 0A AB FF FF FB 00 00 00 00 01 23 45
FF FF 7F F9 DF BF 57 B9 FF FF FF FF FF FF FF FF
FF FF FF FF FF FF FF FF FF FF FF FF FF FF FF FF
53 54 41 54 49 4F 4E 20 30 33 35 00 00 00 00 00
FF FF FF FF FF FF FF FF FF FF FF FF FF FF FF FF
FF FF FF FF FF FF FF FF FF FF FF FF FF FF FF FF
FF 22 22 22 22 22 22 22 FF FF FF FF FF FF FF FF
FF FF FF FF FF FF FF FF FF FF FF FF FF FF FF FF
FF FF FF FF FF FF FF FF FF FF FF FF FF FF FF FF
5B 4F 9A E4 B5 09 8B E7 D8 FF FF FF FF FF FF FF
FF FF FF FF FF FF FF FF FF 11 00 11 FF 10 00 01
FF FF FF FF FF FF FF FF FF FF FF FF FF FF FF FF
FF FF FF FF FF FF FF FF FF FF FF FF FF FF FF FF
FF FF FF FF FF FF FF FF FF FF FF FF FF FF FF FF
90 00

*SET SECURITY FUSES

*Set FAB Fuse
00 B4 01 06 00

*Set CMA Fuse
00 B4 01 04 00

*Set PER Fuse
```

```
00 B4 01 00 00
*Read Fuse Byte = X0
00 B6 01 00 01

Response:
00
90 00

power_off
```

Initialization Example Using Atmel API

The AT88SC0104C is used for this example. A small pattern is written into each of the four user zones. Security for each of the four user zones and the associated register values are shown below. Simple values for codes, keys, and passwords are used.

Table 8. Security Settings – AT88SC0104C Example

User Zone	Data	Security Requirements	Access Register	Password/Key Register
0	Zone 0	None	$FF	$FF
1	Zone 1	Read/Write Password (Set 1)	$7F	$F9
2	Zone 2	Read/Write Authentication (Set 2)	$DF	$BF
3	Zone 3	Read/Write Password (Set 1) Read/Write Authentication (Set 1) with Encryption Required	$57	$B9

The following example uses the API from the CryptoMemory Development Kit (AT88SC25616C-DK). The flow is consistent with the steps described in this application note, and comments have been added.

```
//AT88SC0104C Initialization Example

PowerOn()

//WRITE DATA TO USER ZONES
data0=0x5A 6F 6E 65 20 30 20 44 61 74 61
data1=0x5A 6F 6E 65 20 31 20 44 61 74 61
data2=0x5A 6F 6E 65 20 32 20 44 61 74 61
data3=0x5A 6F 6E 65 20 33 20 44 61 74 61

//Set User Zone 0
Set_Zone(0,0)
//Write data = Zone 0 Data
Write_User_Zone(0x00,11,data0)

//Set User Zone 1
Set_Zone(1,0)
//Write data = Zone 1 Data
Write_User_Zone(0x00,11,data1)

//Set User Zone 2
Set_Zone(2,0)
```

```
//Write data = Zone 2 Data
Write_User_Zone(0x00,11,data2)

//Set User Zone 3
Set_Zone(3,0)
//Write data = Zone 3 Data
Write_User_Zone(0x00,11,data3)

//UNLOCK CONFIGURATION ZONE
Verify_Password(0,7,0xFFFFFF,false)

//WRITE CODES IN CONFIGURATION ZONE
cmc=0x50 30 30 31
idnum=0x00 00 00 00 01 23 45
issue=0x53 54 41 54 49 4F 4E 20 30 33 35 00 00 00 00 00

//Write Card Mfg Code = P001
Write_Config(0x0B,4,cmc)
//Write Identification Number = 00000000012345
Write_Config(0x19,7,idnum)
//Write Issuer Code = STATION 035
Write_Config(0x40,16,issue)

//WRITE REGISTERS IN CONFIGURATION ZONE
Register=0x7F F9 DF BF 57 B9

//Write Registers AR1/PR1 = 7F F9, AR2/PR2 = DF BF, AR3/PR3 = 57 B9
Write_Config(0x22,6,Register)

//WRITE KEYS IN CONFIGURATION ZONE
Crypti=0x22 22 22 22 22 22 22
Seed=0x5B 4F 9A E4 B5 09 8B E7

//Write Ci for set 2 = 22222222222222
Write_Config(0x71,7,Crypti)

//Write Gc for set 2 = 5B4F9AE4B5098BE7
Write_Config(0xA0,8,Seed)

//WRITE PASSWORDS IN CONFIGURATION ZONE
Pass=0x11 00 11 FF 10 00 01

//Write Passwords, read 7 = 10 00 01, write 7 = 11 00 11
Write_Config(0xB9,7,Pass)

//READ ENTIRE CONFIGURATION ZONE TO VERIFY
Read_Config(0x00,F0,configdata)

Response:
Configdata=0x
```

```
3B B2 11 00 10 80 00 01 10 10 FF 50 30 30 31 FF
8C AD A8 10 0A AB FF FF FB 00 00 00 00 01 23 45
FF FF 7F F9 DF BF 57 B9 FF FF FF FF FF FF FF FF
FF FF FF FF FF FF FF FF FF FF FF FF FF FF FF FF
53 54 41 54 49 4F 4E 20 30 33 35 00 00 00 00 00
FF FF FF FF FF FF FF FF FF FF FF FF FF FF FF FF
FF FF FF FF FF FF FF FF FF FF FF FF FF FF FF FF
FF 22 22 22 22 22 22 22 FF FF FF FF FF FF FF FF
FF FF FF FF FF FF FF FF FF FF FF FF FF FF FF FF
FF FF FF FF FF FF FF FF FF FF FF FF FF FF FF FF
5B 4F 9A E4 B5 09 8B E7 D8 FF FF FF FF FF FF FF
FF FF FF FF FF FF FF FF FF 11 00 11 FF 10 00 01
FF FF FF FF FF FF FF FF FF FF FF FF FF FF FF FF
FF FF FF FF FF FF FF FF FF FF FF FF FF FF FF FF
FF FF FF FF FF FF FF FF FF FF FF FF FF FF FF FF

//SET SECURITY FUSES

//Set FAB Fuse
Write_Fuses(06)

//Set CMA Fuse
Write_Fuses(04)

//Set PER Fuse
Write_Fuses(00)

//Read Fuse Byte = 0x0
Read_Fuses(fusebyte)

Response:
Fusebyte=0x0

PowerOff
```

Synchronous Communications

CryptoMemory can communicate by using either the ISO 7816-3 protocol or a synchronous two-wire protocol. This synchronous two-wire interface is identical to that used on Atmel's AT24Cxxx family of serial EEPROMs but with a unique command set. This communications mode provides faster communication with the device for loading initial data and programming the configuration zone. Synchronous communications may be performed with a clock speed of up to 1.5 MHz. The command set used for initializing CryptoMemory in synchronous mode is shown in Table 9.

Table 9. Initialization Synchronous Command Set

		Command	Addr 1	Addr 2	N	Data (N)
Write User Zone	(Devices 0104C–1616C)	$B0	addr	addr	N ≤ $10	N bytes
	(Devices 3216C–25616C)	$B0	addr	addr	N ≤ $80	N bytes
Read User Zone		$B2	$00	addr	N	
System Write	Write Config Zone (Devices 0104C–1616C)	$B4	$00	addr	N ≤ $10	N bytes
	Write Config Zone (Devices 3216C–25616C)	$B4	$00	addr	N ≤ $80	N bytes
	Write Fuse	$B4	$01	fuse ID	$00	
	Set User Zone	$B4	$03	zone	$00	
System Read	Read Config Zone	$B6	$00	addr	N	
	Read Fuse Byte	$B6	$01	$00	$01	
Verify Secure Code		$BA	$07	$00	$03	3-byte password

In the AT88SC0104C initialization example shown on pages 6–10, the total time for all operations listed is approximately 2.9 seconds. This example uses the ISO 7816-3 asynchronous protocol with its default communications speed of 9600 baud. The time will be the same whether using TPDU commands or the Atmel API as they both use this asynchronous protocol. By comparison, if the same example is executed using the synchronous two-wire protocol, the total time for all operations is approximately 162 ms at a clock speed of 1.5 MHz. The dual protocol of CryptoMemory allows faster initialization of cards by using the synchronous protocol while maintaining compatibility with standard ISO 7816-3 readers in the field.