# A Companion Chip For Atmel CryptoRF & CryptoMemory Products

## Features

- Companion Chip to CryptoRF® and CryptoMemory®
  - Securely implements host algorithms
  - Securely stores host secrets
  - Verifies Host Firmware Digests
- High Security Features in Hardware
  - CryptoMemory and CryptoRF F2 Algorithm
  - SHA-1 Standard Cryptographic Algorithm
  - 64-bit Mutual Authentication Protocol (Under License of ELVA)
  - Permanently Coded Serial Numbers
  - High Quality Random Number Generator (RNG)
  - Metal Shield Over Memory
  - Data Scrambling in Nonvolatile Memory
  - Delay Penalties to prevent Systematic Attacks
  - Reset Locking to prevent Illegal Power Cycling
  - Voltage and Frequency Monitors
- Host-side Crypto Functions
  - Authentication Challenge Generation
  - Device Challenge Response
  - Message Authentication Codes (MAC) Generation
  - Data Encryption and Decryption
  - Secure Authentication Key Management
- Secure Storage and Key Management
  - Up to 16 sets of 64-bits Diversified Host Keys
  - Eight Sets of Two 24-bit Passwords
  - Secure and Custom Personalization
  - Up to 232-Byte Read/Write Configurable User Data Area
- Nonvolatile Up Counters
  - Four sets Unidirectional Counters
  - 6.4 Million Maximum Counts Per Counter
- Application Features
  - Low Voltage Supply: 2.7V – 3.6V
  - 2-Wire Serial Interface (TWI, 5V Compatible)
  - Standard 8-lead SOIC Plastic Package, Green compliant (exceeds RoHS)
- High Reliability
  - Endurance : 100,000 Cycles
  - Data Retention : 10 years
  - ESD Protection : 3,000 V min. HBM

# CryptoCompanion™ Chip for CryptoMemory and CryptoRF

# AT88SC018

5277C–CryptoCompanion–9/09

# 1. Product Overview

The AT88SC018 is designed as the mate to Atmel's CryptoRF® (CRF) and CryptoMemory® (CM) chips, collectively referred to in the remainder of this document as CRF. Within the operation descriptions, the AT88SC018 CryptoCompanion chip is sometimes referred to as CMC or CryptoMemory Companion chip.

The AT88SC018 makes extensive use of the SHA-1 hash algorithm as specified in http://www.itl.nist.gov/fipspubs/fip180-1.htm and elsewhere. In this document, the nomenclature SHA-1(a, b, c) means to concatenate a, b & c in that order and then pad them to a block size of 64 bytes before computing the digest. The AT88SC018 does not ever generate a SHA-1 digest of datasets larger than a single round

## 1.1. General Operation

The CRF chip contains secrets that must be known or derived by a host system in order to establish a trusted link between the two and permit communications to happen. The AT88SC018 stores these secrets in an obscured way in nonvolatile memory and contains all the circuitry necessary to perform the authentication, password and encryption/decryption functions specified in the CRF datasheet. In this manner, the secrets do not ever need to be revealed.

The general cryptographic strategy is as follows:

- Each CRF chip has a serial or identification number (ID) and authentication secret $G_i$ stored in EEPROM. ID is freely readable; $G_i$ can never be read and is unique for all tags.
- The AT88SC018 contains an EEPROM that contains a set of common secrets ($F_n$). The AT88SC018 combines $F_n$ with ID and $K_{ID}$ to compute a value of G that is expected to match that in the CRF chip. Specifically, G = SHA-1($F_n$, ID, $K_{ID}$)
- G is further diversified by the inclusion of a number ($K_{ID}$) generated by the host system in a manner of its choosing. Typically, it will be the result of a cryptographic operation on the CRF ID value calculated using other data, secrets and/or algorithms external to the AT88SC018. This permits scenarios that offer varying degrees of additional security.
- The AT88SC018 includes a general purpose cryptographic quality random number generator which is used to seed a mutual authentication process between the AT88SC018 and CRF. If the CRF confirms the CMC challenge, and the CMC confirms the CRF response, then the host system proceeds with CRF operations. In this way the host system may use the CRF without knowing the CRF's secrets directly.

## 1.2. CryptoCompanion Benefits

The following is a partial list of the benefits of using this chip versus storing the algorithms and secrets in standard FLASH system memory.

- Keep confidential those core secrets that are used to authenticate with and communicate to/from CRF. (Store them in EEPROM, use them on-chip)
- Flexible system implementation – multiple secrets and policies for different CRF locations within the system. Multiple manufacturer setup options.
- Hardware encryption engines, avoids algorithm disclosure from reverse-compilation of system operating code.
- Full hardware security implementation makes it harder for an attacker (even with lab equipment) to get secrets stored on the AT88SC018.
- Global secrets are protected using strong security, standard algorithm (SHA-1).
- Implements a crunching algorithm to prevent micro-controller based CRF replicas.
- Robust random number generation avoids accidental replay for all cryptographic operations using the system, not just with respect to CRF.

# 2 CryptoCompanion Chip

- Secure EEPROM storage for configuration information, etc. May permit reduction in the total BOM for the system.
- Easy to use – little programming required; no knowledge of security algorithms or protocols, fast time to market.

## 1.3. CryptoCompanion Security

The following is a partial list of the security features on this chip.

- Strong internal EEPROM encryption scheme
- Dynamically encrypted internal SRAM data.
- Programmable powerup penalty.
- Escalating Attack penalty.
- Authentication timeouts.
- Anti-tearing counters.
- Anti-tearing RNGSeed.
- Secure Personalization.
- Command usage limitations to prevent exhaustive attacks
- Uniquely encrypted F Secrets inside chip.
- High security Internal Clocking Scheme.
- Over and Under Voltage detection tampers.
- Internal Data integrity validation.
- Active shield over security sensitive blocks

## 1.4. Package, Pinout & IO

### 1.4.1. Pinout

All pins not otherwise specified are considered Test pins and should be grounded on the board.

#### *1.4.1.1. $V_{CC}$ , Gnd*

Power supply is 2.7 – 3.6V. Supply current less than 5 mA.

CryptoCompanion will be available to accept commands 60 ms after the later of $V_{CC}$ rising above 2.7V or Reset being driven high if CryptoCompanion is in a security delay then this interval is significantly longer.

During Power Up, $V_{CC}$ must exhibit a monotonic ramp at a minimum rate of 50 mV/mS until $V_{CC}$ has crossed the 2.7V level. During Power Down, $V_{CC}$ must exhibit a monotonic ramp at a minimum rate of 50 mV/mS once it has dropped below the 2.5V boundary. CryptoCompanion does not support hot swapping or hot plugging.

$V_{CC}$ must be bypassed with high quality surface mount capacitors that are properly located on the board. Atmel recommends two capacitors connected in parallel having a value of 1μF and 0.01μF. The capacitors should be manufactured using X5R or X7R dielectric material. These capacitors should be connected to the AT88SC018 using a total of no more than 1cm PC board traces. Atmel recommends the use of a ground plane and a trace length of less than 0.5cm between the capacitors and the $V_{CC}$ pin. Failure to follow these recommendations may result in improper operation.

#### *1.4.1.2. SDA*

Two wire interface data pin, 5 V compatible. Data setup time = 0.1 μs minimum data hold time = 0 μs min. The system board must include an external pull-up resistor.

#### 1.4.1.3. SCL

Two wire interface clock pin, 5 V compatible. Maximum SCL rate is 400KHz, min. $T_{LOW}$ = 1.2 µs, min. $T_{HIGH}$ = 0.6 µs. The system board must include an external pull-up resistor.

#### 1.4.1.4. Reset (RST)

This active low input will reset all states within the AT88SC018. It is honored regardless of the state of PowerDown.

#### 1.4.1.5. PowerDown(PDN)

When held low, the part operates normally. When held high the part will go to sleep and ignore all transitions on SDA and SCL, power consumption will drop to less than 10 µA. There is a 50 ms delay between this pin falling and the first transition on SDA or SCL that will be accepted by the chip.

### 1.4.2. Package

The AT88SC018 is packaged in an 8 lead SOIC package, pinout is as follows:

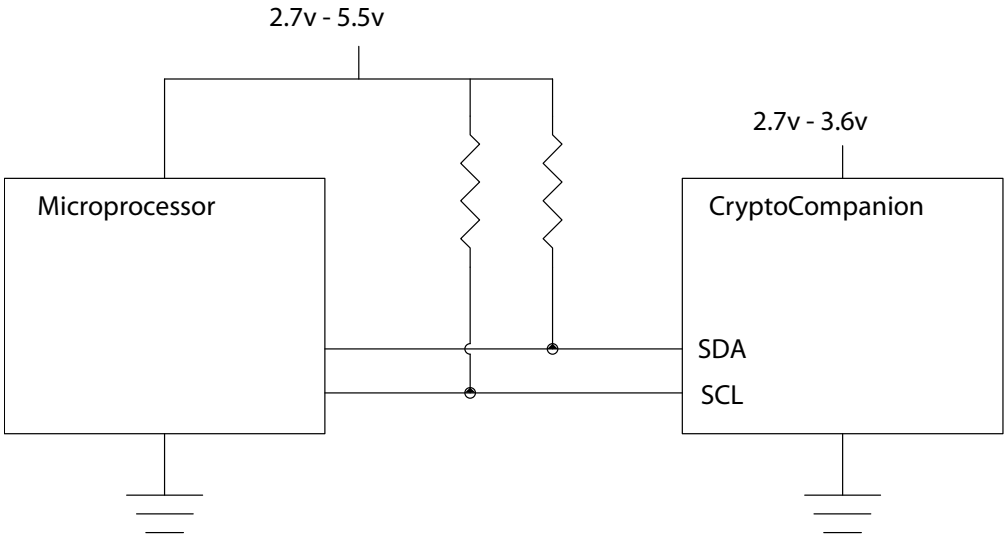
*Table 1. 8 lead SOIC package pinout*

| Pin Number | Pin Name |
|---|---|
| 1 | $V_{CC}$ |
| 5 | GND |
| 7 | SDA |
| 8 | SCL |
| 4 | RST |
| 3 | PDN |
| 2,6 | NC |

Pins 2 & 6 are not internally connected and should be connected to ground on the PC board.

### 1.4.3. Connection Diagram

*Figure 1. Connection Diagram*



### 1.4.4. Environmental

The AT88SC018 is guaranteed to operate over the industrial temperature range of -40°to 85°C. ESD is rated at 2KV, Human Body Model.

### 1.4.5. TWI Input/Output Operation

The AT88SC018 communicates to the system using a two wire interface (TWI), which is similar to SMBus™. The chip operates as a slave and does not support clock stretching. This two wire protocol is identical to that supported by the Atmel AT24C16B serial EEPROM chips. Refer to the datasheet on the Atmel web site for detailed timing and protocol information.

The system processor is expected to properly format commands for the AT88SC018 (which may include information from the CRF chip), and then process the outputs of the AT88SC018 (which may include sending some of the outputs to the CRF chip).

The AT88SC018 cannot directly communicate with CRF or CM chips. Both CRF/CM and the AT88SC018 are slave devices. The bus master may use one or two busses to communicate with them. Separate TWI addresses must be used if both chips are on the same bus.

*Table 2. AT88SC018 communications packets naming conventions.*

| AT88SC018 Name | TWI Name | Description |
|---|---|---|
| Device Address | Device Name | This byte selects a particular chip on the two wire bus. Bit 1 of this byte on the AT88SC018 selects between accesses to command/data (if 0) or the status register (if 1). Bit 0 of this byte is the standard two wire R/W pin, if 1 then the bytes following the device address travel from the slave to the master (Read) if 0 these bytes flow to the slave (Write). |
| Cmd | Word Address | If the device address specified a command input (TWI write), then this byte specifies the command to be executed by the AT88SC018. This byte doesn't exist on read operations. |
| Size | $Data_N$ | The total number of bytes to follow this byte may be 0 in the case that there are no operand bytes. This byte doesn't exist on status read operations. |
| Data | $Data_{N+1}$ , … | Operand input or output bytes as specified in the command descriptions in Command Descriptions. |

If the upper 6 bits of the device address byte sent over the TWI match the upper 6 bits of the Dev field in the EEPROM, then the AT88SC018 may respond to this transmission, otherwise it will NACK this byte. Dev is set to a value of 0xC0 on shipment from Atmel.

In general, the AT88SC018 will fail to ACK (NACK) the device address byte if bit 1 of the device address is 0 (command/data transfer) and the AT88SC018 is busy.

The AT88SC018 is designed in such a way that the TWI Size field should be consistent with the count values specified in the command parameter descriptions from Command Descriptions. If the TWI size field is inconsistent with the command parameter count value, the AT88SC018 will respond in different ways depending on the specific command. Some of these responses may include security penalties, other error indications or some input bytes may be silently ignored.

### 1.4.5.1. Command Input

*Table 3. Command Input Byte Sequence*

| Byte # | Direction | Name | Description |
|---|---|---|---|
| 0 | To Slave | Device Address | This byte selects a particular chip on the two wire bus. Bit 1 of this byte should be 0 to indicate a command transfer to the AT88SC018. Bit 0 of this byte should be 0 to indicate that the data bytes travel from the master to the slave (TWI write). |
| 1 | To Slave | Cmd | The ordinal of the command to be executed by the AT88SC018, from the table below. |
| 2 | To Slave | Size | The total number of bytes to follow this byte may be 0 in the case that there are no operand bytes. |
| 3, … | To Slave | Data | Operand bytes as specified in Command Descriptions. |

If the command ordinal is legal, the AT88SC018 will ACK the command input and start processing. It takes a variable amount of time to process the command, up to 20ms depending on the number of EEPROM pages to be written. If an illegal command ordinal (≥0x15) is sent to the chip it will lock up for a “security delay”, then resume normal operation. Refer to Section 1.6.4.

Values in the Cmd byte are chosen from the table below:

*Table 4. Cmd Byte Values*

| Command | | Value |
|---|---|---|
| VerifyFlash | | 0x01 |
| Startup | | 0x02 |
| ChallengeResponse | | 0x03 |
| Auth_1 | | 0x04 |
| Auth_2 | | 0x05 |
| EncryptPassword | | 0x06 |
| Encryption_1 | | 0x07 |
| Encryption_2 | | 0x08 |
| GrindBytes | | 0x09 |
| GetRandom | | 0x0A |
| IncrementCounter | | 0x0B |
| ReadCounter | | 0x0C |
| WriteMemory | | 0x0D |
| WriteMemoryEncrypted | | 0x0E |
| WriteMemoryAuthorized | | 0x0F |
| ReadMemory | | 0x10 |
| ReadMemoryDigest | | 0x11 |
| ReadManufacturingID | | 0x12 |
| Lock | | 0x13 |
| Clear | | 0x14 |
| Crunch | | 0x15 |

### 1.4.5.2. Command Output

The command output can be extracted from the AT88SC018 using the following byte sequence.

*Table 5. Command Output Byte Sequence*

| Byte # | Direction | Name | Description |
|---|---|---|---|
| 0 | To Slave | Device Address | This byte selects a particular chip on the two wire bus. Bit 1 of this byte should be 0 to indicate that this is a command output. Bit 0 of this byte should be 1 to indicate that the data will travel from the slave to the master. |
| 1 | To Master | Size | The total number of bytes to follow this byte may be 0 in the case that there are no output bytes. |
| 2, … | To Master | Data | Output bytes as specified in Command Descriptions. |

Command output bytes can be repeatedly read from the AT88SC018 as they remain valid until a new command is sent to the AT88SC018. Until <size> bytes of the new command have been sent, DataAvailable will remain set and that number of bytes can be read from the SRAM output buffer, though the new input bytes will overwrite the old output bytes.

Some commands do not have any data output, for instance ‘Clear’. On completion of these commands, the DataAvailable bit will be cleared and the system can read just the size byte, which will have a value of 0.

### 1.4.5.3. Status

This register can be read to determine the current status or the error information using the following byte stream. This sequence can be run at any time, regardless of whether or not the AT88SC018 is busy or locked.

*Table 6. Byte Stream Sequence*

| Byte # | Direction | Name | Description |
|---|---|---|---|
| 0 | To Slave | Device Address | This byte selects a particular chip on the two wire bus. Bit 1 of this byte should be 1 to select the status register. Bit 0 of this byte is the standard two wire R/W pin and should be 1 (data bytes travel from the slave to the master). |
| 1 | To Master | Status | Returns the current value of the status register. |

The status register value is described in the following table:

*Table 7. Status Register Value*

| Byte # | Name | Description |
|---|---|---|
| 0 | Data Available | The AT88SC018 has completed processing of the command and data is available in the output buffer. A successfully completed command that does not have any output will NOT set this bit. |
| 1 | Busy | The AT88SC018 is processing a command and is unable to accept more input or provide output, or it is in some sort of security penalty period. |
| 2 | StartupDone | The ChallengeResponse command has successfully run this power cycle. Once set, this bit will remain set until the next reset or power cycle. |
| 3 – 4 | Reserved | Will always be 0. |
| 5 – 7 | Error | An error occurred during prior input or command processing. The value of these three bits denotes the particular condition that occurred. |

The 8 error codes are used as follows:

*Table 8. Error Codes*

| Name | Value | Description |
|---|---|---|
| OK | 0 | Enabled, no error. |
| RstLocked | 1 | The AT88SC018 is disabled until the next power cycle or reset assertion. Whenever the error bits are in this state, the Busy bit in the status register will also be asserted. |
| BadCmd | 2 | The formatting of the command was invalid, or one of the operands had an unacceptable value. |
| TimeDelay | 3 | The AT88SC018 is disabled up for a certain period of time and will respond to commands after this delay has elapsed. This delay may be a Power Delay (Section 1.6.2) or Security Delay (Section 1.6.3). Whenever the error bits are in this state, the Busy bit in the status register will also be asserted. |
| AuthFail | 4 | Either authentication must be completed prior to the execution of this command or there was a problem during the execution of the auth commands themselves. |
| — | 5 | |
| — | 6 | |
| — | 7 | |

The system must poll this register (using TWI reads) after sending a command to the chip before attempting to read the result.

This register cannot be written, attempts to do so will result in a NACK.

### 1.4.6. Byte Order

The AT88SC018 uses a big-endian byte order for all large integers (addresses, counters) which means that the most significant byte appears first on the bus. Within this document, that byte is shown on the left side of the page. Arrays (F values, cryptograms, passwords, digests) appear in index order, byte 0 first (or on the left of the page).

The two wire protocol specifies that the most significant bit within a byte appears first on the bus, and it appears on the left side of the page.

## 1.5. Memory Architecture

The 4K bit (512 byte) EEPROM within the AT88SC018 is organized into a number of sections, each of which have different access restrictions.

### 1.5.1. Memory Locking

On shipment from Atmel, certain locations are preloaded by Atmel, per Section 1.5.13. All other data locations are unknown. The system manufacturer should load all areas important for proper system operation with the desired initial values.

When this initialization is complete the Lock command should be executed which limits access to the memory per the restrictions listed later in this section. The system can determine the current lock value by using the ReadManufacturingID command to read out the ManufacturingID value (MfrID) and the lock byte.

The table below describes the encoding of the least significant two bits of the Lock byte. On shipment from Atmel, Lock[1:0] will have a value of either 10 or 00, depending on the part number ordered. An AT88SC018 in either of these two states is considered ‘unlocked’. It is not possible to change from one of these unlocked states to the other.

After the Lock command has been executed, the Lock byte will have the value 0xFF. Subsequent changes to the Lock byte are impossible.

*Table 9. Memory Locking*

| LockBit 1 | Lock Bit 0 (LSB) | Meaning |
|---|---|---|
| 1 | 1 | Locked. ReadMemory and WriteMemory enabled, subject to the restrictions in this section. WriteMemoryEncrypted and ReadMemoryDigest disabled. |
| 1 | 0 | Unlocked/Confidential. ReadMemoryDigest, WriteMemory and WriteMemoryEncrypted enabled. ReadMemory disabled. |
| 0 | 0 | Unlocked. ReadMemory and WriteMemory enabled. WriteMemoryEncrypted and ReadMemoryDigest disabled. |

### 1.5.2. Secure Personalization

Customers desiring to write secrets into the AT88SC018 during personalization without exposing these secrets to attackers should purchase the version of the chip in which Lock[1:0] is 10.

In these parts, Atmel will write a transport key into the EncKey location within EEPROM during wafer probe. Once the AT88SC018 leaves the Atmel factory, the EncKey location cannot be written under any circumstances.

When the part is unlocked and therefore in the personalization phase, the WriteMemoryEncrypted command permits the incoming data to be encrypted using EncKey as the encryption key. Data can also be written unencrypted if desired. Verification of the EEPROM contents must use the ReadMemoryDigest command as ReadMemory is prohibited in these parts as shipped. Once locked, the WriteMemoryEncrypted and ReadMemoryDigest commands are prohibited – WriteMemory and ReadMemory are then enabled over a restricted address space.

The value written into EncKey will be the first 16 bytes of the SHA-1 digest of the concatenation of the 15 byte ManufacturingID with a 16 byte secret provided to Atmel by the system manufacturer. The upper 6 bits of the Lock byte will contain a secret tag assigned by Atmel to differentiate between various secrets that may have been used to generate EncKey. This tag will be erased when the AT88SC018 is locked, leaving the Lock byte with the value 0xFF.

### 1.5.3. ManufacturingID (MfrID)

These 15 bytes contain unique wafer manufacturing information. This data can be used as the AT88SC018 serial number if desired and can also be used by Atmel to track production of the part. It is written by Atmel at wafer test and cannot be modified by the customer, regardless of whether or not the part has been locked.

The ManufacturingID value can only be obtained by executing the ReadManufacturingID command. Note, however, that if Lock[1:0] is ‘10’, then the contents of the second 32 byte block which includes this value can be accessed with ReadMemoryDigest. ReadMemory can never be used to access the first 48 bytes of memory (SHA Constant, EncKey, MfrID & Lock).

### 1.5.4. Passwords

P0-P15. These are the passwords used to enable reading and/or writing of various zones in CRF. For example, CP0 is the configuration byte for P0, and determines the particular attributes which govern the use of P0. The password configuration bytes are organized as below:

*Table 10. Password Configuration Bits.*

| Bit # | Name | Description |
|---|---|---|
| 0 | Encrypt | If 1, EncryptPassword will return this password value in the clear. In this situation, the password offers little security value but may be useful for mapping. |
| 1 | Connect | If 1, then obey the “F number” restrictions below. If 0, ignore “F Number”. |
| 2 – 3 | Reserved | Must be 0. |
| 4 – 7 | F Number | The secret to which this password is connected. Unless the current authentication session has been computed using this secret this password cannot be read in either clear or encrypted mode. |

Once the AT88SC018 is locked, these elements (P0-P15 & CP0-CP15) can never be read directly, nor can they be written.

### 1.5.5. Nonvolatile Counters

The AT88SC018 implements 4 counters that can each increment to a maximum value of 6.4 million. They cannot be reset, nor can they be decremented. Their current state can be read using the ReadCounter command and they are incremented with the IncrementCounter command. It is recommended that the IncrementCounter command not be issued after the counter has reached a value of 6.4 million. Access to these two commands does not require authorization to have completed.

The above constraints only apply to a locked CMC. In an unlocked AT88SC018, the contents of the EEPROM locations that hold the current state of the various counters can be freely read and/or written using ReadMemory (ReadMemoryDigest) or WriteMemory (WriteMemoryEncrypted).

They should be initialized to a count of 0 before the AT88SC018 is locked, by writing the following values into all four of the 16 byte counter areas: “0xFF 0xFF 0xFF 0xFF 0xFF 0xFF 0xFF 0xFF 0x00 0x00 0x00 0x00 0xFF 0x00 0x00 0x00” at addresses “x0, x1, …”. Atmel recommends that all counters be properly initialized even if the application does not utilize all of them.

### 1.5.6. RNGSeed

This location within the EEPROM is initialized during Atmel manufacturing with a 16 byte random number obtained from an external high quality hardware random number generator. It is used as part of the input to the random number generation capability within the AT88SC018. It may be read and/or written when the part is unlocked. Atmel does not recommend that it be written to a fixed value.

### 1.5.7. Read Only Memory

When the part is locked, the memory in this area can be read but never written except as described in the next paragraph. After the system has properly responded to the startup challenge, there are no restrictions on the reading of this memory. This memory section starts at address 0x110 and extends to 0x100 | RW-Bound – 1.

RW-Bound must be at least 0x10 and less than 0xF8 or F-Bound, whichever is smaller.

### 1.5.8. Read / Write Memory

The memory in this area has general read/write permissions, similar to a standard serial EEPROM. After the system has properly responded to the startup challenge, there are no restrictions on the access to this memory.

The first byte in this section is at address 0x100 | RW-Bound. If RW-Bound is less than 0x10 the results will be unpredictable.

### 1.5.9. Secrets

F0-F15. These secrets are used to generate the $G_C$ value for the particular CM/CRF chip based on the F1 algorithm, SHA-1. Up to 16 F values that can be supported by the AT88SC018.

The low byte of the memory address of the first should be written into F-Bound. The 3 least significant bits of F-bound are ignored. The first F value is always F0, independent of F-Bound. If F-bound is < RW-Bound or if F-Bound is < 0x80, the results will be unpredictable.

***Example***
If F-Bound is 0xD0, the first F value is F0 at memory address 0x1D0. The last F value is F5 at address 0x1F8.

***Example***
If 0xFF is written into F-Bound, CMC will use only a single secret, named F0, which will be located at address 0x1F8 (since the low three bits of F-bound are ignored).

These elements can never be read directly, nor can they be written after the part has been locked.

### 1.5.10. CF0 – CF15

This location within the EEPROM is initialized during Atmel manufacturing with a 16 byte random number obtained from an external high quality hardware random number generator. It is used internally within the AT88SC018. It may be read and/or written when the part is unlocked. Atmel does not recommend that it be written to a fixed value.

### 1.5.11. Restricted Bytes

These locations within the EEPROM are initialized during Atmel manufacturing with a 4 byte random number obtained from an external high quality hardware random number generator. It is used internally within the AT88SC018. It cannot be read and/or written when the part is unlocked or locked. When reading from these locations, the result will be 0xFF for these 4 bytes.

### 1.5.12. Memory Map

*Figure 2. Memory Map*

| *Most Significant Address Bits* | | *Least Significant Address Bits* | | | | | | | |
|---|---|---|---|---|---|---|---|---|---|
| | | *0* | *1* | *2* | *3* | *4* | *5* | *6* | *7* |
| | *0x000* | SHA Constant | | | | | | | |
| | *0x008* | | | | | | | | |
| | *0x010* | EncKey | | | | | | | |
| | *0x018* | | | | | | | | |
| | *0x020* | ManufacturingID | | | | | | | |
| | *0x028* | | | | | | | | Lock |
| | *0x030* | P0 | | | CP0 | P1 | | | CP1 |
| | *0x038* | P2 | | | CP2 | P3 | | | CP3 |
| | *0x040* | P4 | | | CP4 | P5 | | | CP5 |
| | *0x048* | P6 | | | CP6 | P7 | | | CP7 |
| | *0x050* | P8 | | | CP8 | P9 | | | CP9 |
| | *0x058* | P10 | | | CP10 | P11 | | | CP11 |
| | *0x060* | P12 | | | CP12 | P13 | | | CP13 |
| | *0x068* | P14 | | | CP14 | P15 | | | CP15 |
| | *0x070* | Counter0 | | | | | | | |
| | *0x078* | | | | | | | | |
| | *0x080* | Counter1 | | | | | | | |
| | *0x088* | | | | | | | | |
| | *0x090* | Counter2 | | | | | | | |
| | *0x098* | | | | | | | | |
| | *0x0A0* | Counter3 | | | | | | | |
| | *0x0A8* | | | | | | | | |
| | *0x0B0* | SystemSecret | | | | | | | |
| | *0x0B8* | | | | | | | | |
| | *0x0C0* | CmcSecret | | | | | | | |
| | *0x0C8* | | | | | | | | |
| | *0x0D0* | RNGSeed | | | | | | | |
| | *0x0D8* | | | | | | | | |
| | *0x0E0* | FlashDigest | | | | | | | |
| | *0x0E8* | | | | | | | | |
| | *0x0F0* | | | | | RstProt | RW-Bound | F-Bound | Dev |
| | *0x0F8* | CF0 | CF1 | CF2 | CF3 | CF4 | CF5 | CF6 | CF7 |
| | *0x100* | CF8 | CF9 | CF10 | CF11 | CF12 | CF13 | CF14 | CF15 |
| | *0x108* | Mode | PwrDelay | *spare* | *spare* | *Restricted* | *Restricted* | *Restricted* | *Restricted* |
| | *0x110* | Read Only Memory | | | | | | | |
| | ... | | | | | | | | |
| | *0x178* | Read / Write Memory | | | | | | | |
| | *0x180* | F0 | | | | | | | |
| | *0x188* | F1 | | | | | | | |
| | *0x190* | F2 | | | | | | | |
| | *0x198* | F3 | | | | | | | |
| | *0x1A0* | F4 | | | | | | | |
| | *0x1A8* | F5 | | | | | | | |
| | *0x1B0* | F6 | | | | | | | |
| | *0x1B8* | F7 | | | | | | | |
| | *0x1C0* | F8 | | | | | | | |
| | *0x1C8* | F9 | | | | | | | |
| | *0x1D0* | F10 | | | | | | | |
| | *0x1D8* | F11 | | | | | | | |
| | *0x1E0* | F12 | | | | | | | |
| | *0x1E8* | F13 | | | | | | | |
| | *0x1F0* | F14 | | | | | | | |
| | *0x1F8* | F15 | | | | | | | |

### 1.5.13. Memory Initialization Values

Upon shipment from the Atmel factory, the following locations will have predefined values. The contents of all other locations are not guaranteed by Atmel.

*Table 11. Predefined Initial Memory Values*

| Name | Initial Value |
|---|---|
| SHA Constant | Defined by FIPS PUB 180-1.This is written at the Atmel factory and cannot subsequently be changed. |
| EncKey | Customer Specific, Contact Atmel. See Section 1.5.2 for more details. |
| ManufacturingID | A unique value for all AT88SC018 chips, see Section 1.5.3 |
| Lock | xxxx_xx10 or xxxx_xx00, per Sections 1.5.1 & 1.5.2. Consult Atmel for ordering information. |
| RNGSeed | Random values for each AT88SC018. See Section 1.5.6 |
| Dev | TWI bus address, shipped as 0xC0. See Section 1.5.4 |
| CF0 – CF15 | Random values for each AT88SC018. See Section 1.5.10 |

Certain values within the AT88SC018 memory array MUST be properly programmed prior to locking of the memory. Failure to properly initialize these locations will result in unpredictable and/or unsecure operation of the part.

*Table 12. Customer Defined Memory Values*

| Name | Initial Value |
|---|---|
| SystemSecret, CmcSecret | These values are used to perform a mutual authentication between the AT88SC018 and the system processor. See the Startup (Section 3.2) and ChallengeResponse (Section 3.3) for more details. |
| RW_Bound | The boundary between ReadOnly and ReadWrite memory. See Sections 1.5.7 & 1.5.8 for more information. |
| F_Bound | Controls the number of F secrets in the array, see Section 1.5.9 for value limitations. |
| Mode | The lower 2 bits control the way in which VerifyFlash is run, see Section 3.1 for more details. The upper 5 bits MUST be ‘0’ for proper operation; other values may result in security or functional issues. |
| FlashDigest | If Mode.Bit[1:0] is set to 0, then this must be set to the proper value per the descriptions in the VerifyFlash command, see Section 3.1. |

## 1.6. Security Features

### 1.6.1. Environmental Detectors

The AT88SC018 contains an over and under voltage detector for $V_{CC}$ and includes a POR detector to prevent any unknown startup states. If this detector is triggered, the AT88SC018 will be held in reset until the condition is cleared.

The operating clock is internally generated independent of SDA & SCL, and glitches on those pins are filtered out. The AT88SC018 includes a metal obfuscation pattern over the memory block.

### 1.6.2. Reset Protection & Power Delay

There is a reset protection register in EEPROM (RstProt) that normally has a value of 1 before power is applied. On reset, the AT88SC018 writes this register in the EEPROM to a value of 0, and starts a counter. That counter counts 1 MHz clocks up to a total delay interval of approximately 67 seconds, and at that time the AT88SC018 writes the protection register to a value of 1. If a command is in progress when this time interval is reached, the register will be updated at the completion of the command. After this write, the reset protection circuit goes idle until the next reset.

If at the time of reset or power-up the protection register already has a value of 0, then the AT88SC018 goes into a “Power Delay” state for the same amount of time during which it will neither accept nor acknowledge any command. At the end of the time interval, it will reset the register to a value of 1 and resume normal operation. A power-up or pin reset during the “Power Delay” interval will restart the delay counter and start a new interval during which commands will be ignored.

The AT88SC018 is designed to permit the system to execute the reset operation (and operate for at least 67 seconds) a minimum of 1 million times. If the part is continuously reset every 67 seconds, this limit will be reached in about 2 years.

The Power Delay of 67 seconds is the maximum delay that the AT88SC018 can support. The actual delay is derived from the contents PwrDelay byte within the EEPROM, according to the following table. The measured delay will vary by up to +/- 25% over manufacturing and operating conditions.

*Table 13. Reset Protection & Power Delay*

| PwrDelay | Nominal Delay Interval | | PwrDelay | Nominal Delay Interval |
|---|---|---|---|---|
| 0x00 | 262ms | | 0x10 | 4.5s |
| 0x01 | 524ms | | 0x20 | 8.7s |
| 0x02 | 785ms | | 0x40 | 17s |
| 0x04 | 1.3s | | 0x80 | 34s |
| 0x08 | 2.4s | | 0xFF | 67s |
| Other | Unpredictable | | | |

**Note**: Short power delay times may decrease the overall security of the system.

The reset protection circuit and associated power delay operates regardless of whether the AT88SC018 is locked or unlocked.

Failure to meet Power up and Power down conditions listed in Section 1.4.1.1 may result in invoking a reset protection state, causing a “Power Delay” interval.

### 1.6.3. Reset Locking

Certain conditions cause the AT88SC018 to lock up until the reset pin is asserted or the power is cycled. Depending on the time interval from the last power-up, this action may or may not cause a delay to be enforced. During this time, the status register will show the RstLocked error state and the busy pin will be asserted.

a) Some command other than VerifyFlash is attempted before Startup/ChallengeResponse has been run or some command other than ChallengeResponse follows Startup.

b) ChallengeResponse is run but the preceding command is not Startup.

c) VerifyFlash fails for any reason other than that it has been disabled.

d) ChallengeResponse fails for any reason.

e) Second attempt to run VerifyFlash in a single power cycle.

### 1.6.4. Security Delay

When certain operations do not complete successfully, the AT88SC018 will enter a temporary security delay for a period of time during which no commands will be honored by the AT88SC018. During this time, the system may read the status register which will contain the TimeDelay error code & busy bit set.

The following conditions cause the AT88SC018 to enter a security delay when it is locked. Unlocked AT88SC018 chips never enter the security delay sequence.

a) A second attempt to run Startup after the first has completed within the same power or reset cycle.

b) Some command other than Auth_2 follows Auth_1.

c) The values sent to the AT88SC018 for Auth_2 do not match those computed internally (authentication failed).

d) The values sent to the AT88SC018 for Encryption_2 do not match those computed internally (encryption key verification failed).

e) An illegal command ordinal is sent to the AT88SC018.

The first time one of these conditions is detected after a power cycle or reset event, the AT88SC018 will delay ~260ms. After each subsequent failure condition is detected, the AT88SC018 will delay for an interval twice the length of the previous delay.
Once this doubling reaches a delay equal to or greater than PwrDelay, all subsequent failure conditions will trigger a lockout interval equal to PwrDelay. The maximum Security Delay is 32s, regardless of the value of PwrDelay.

### 1.6.5. Command Sequencing

Depending on whether the AT88SC018 is locked or not, some commands must be executed in a certain order, this section outlines those restrictions.

#### *1.6.5.1. When the AT88SC018 is Unlocked*

When the AT88SC018 is unlocked, there is no security delay and there is no requirement that Startup/Challenge be executed prior to any other command. This strategy may facilitate quicker initialization.

**Note**: The Power Delay continues to be active when unlocked and authentication must still be run for those commands that require it (EncryptPassword, Encryption_1&2, GrindBytes).

When the AT88SC018 is unlocked, the following commands are enabled:

- Read Memory can be run only if the least significant two bits of the lock byte in EEPROM are both 0. All locations from 0x30 onwards can be read.
- ReadMemoryDigest can be run on all locations within the EEPROM if Lock[1:0] has a value of 0x10.
- WriteMemory can be run over all locations from 0x30 onwards.
- WriteMemoryEncrypted can only be run if Lock[1:0] has a value of 0x10.
- The Lock command can be run to exit the unlocked state.

#### 1.6.5.2. When the AT88SC018 is Locked

When the AT88SC018 is locked, the security delays from Section 1.6.3 apply.

The first command run after powerup or a reset must be either VerifyFlash or Startup. If the first command is Startup, then VerifyFlash cannot be run until the next power cycle. If the first command is VerifyFlash, then the next command must be Startup. After Startup, the next command must always be Challenge Response.

No other command can be run until ChallengeResponse has successfully completed. Any attempt to run another command prior to ChallengeResponse or a failure of the ChallengeResponse command will cause the AT88SC018 to lock up until the next power cycle or reset assertion.

A complete and successful authentication sequence (Auth_1 & Auth_2) must be run prior to those commands that require it: EncryptPassword, Encryption_1, Encryption_2 and GrindBytes. Failure to run the authentication sequence will result in an error code in the status register but no delay.

When the AT88SC018 is locked, the following commands are disabled: WriteMemoryEncrypted, ReadMemoryDigest and Lock. WriteMemory is available only for Read/Write memory (the region between RWBound and F-Bound). ReadMemory is only available for ReadOnly + ReadWrite memory (the region between address 0x110 and F-Bound). Any attempt to violate these restrictions will result in a BadCmd error message but no penalty.

## 2. CMC ⇔CRF Authentication

The AT88SC018 supports the mutual authentication sequence of the CRF chip in a manner such that the shared secrets are not ever exposed on the AT88SC018 or CRF busses. This section describes that mutual authentication sequence. To be consistent with the parameter names in the command descriptions, the AT88SC018 is referred to by its alternate name of CMC.

### 2.1. Nomenclature

$X_i$ The subscript ‘i’ indicates a key index in the CRF memory array. CRF contains 4 sets of key values, only those from a single set can be used in a successful authentication sequence.

$Y^A$, $Y^E$ The superscripts ‘A’ and ‘E’ indicate the two possible phases of the crypto setup for CRF. ‘A’ indicates the authentication phase which prefaces all cryptographic communication with CRF. The ‘E’ indicates the optional encryption phase.

C The initial cryptogram state from CRF to CMC. It is the state generated as a result of a previous authentication or encryption sequence and is unique.

CH, $C_i$ These values are the challenge and response during the mutual authentication & encryption sequences. $CH^A$ is the authentication challenge to CRF from CMC. $C_i^A$ is the authentication response from CRF to CMC, $C^A$ is the copy of this computed within CMC. $CH^E$ is the encryption challenge to CRF from CMC. $C_i^E$ is the encryption response from CRF to CMC, $C^E$ is the copy of this computed within CMC.

F2 This is the Atmel proprietary algorithm implemented within CMC and CRF. [A, B, C] = F2(X, Y, Z) indicates that X, Y & Z are inputs to the F2 algorithm and that execution of the algorithm on these inputs yields the set of outputs A, B & C.

G, $G_i$ The secret stored in CRF or computed on CMC from ID and Fn.

ID This is the unique serial or identification number for CRF which is obtained from the Nc register within the CRF EEPROM.

$K_{ID}$ This is a constant generated by the external system in a manner of its choosing. It should typically be a function of the ID number and an external secret, but may also include other information about the item to which CRF is attached, the system configuration or other values held external to CMC. CMC treats $K_{ID}$ as a constant and does not interpret its value.

Q These are random values created in the RNG of CMC which are used as part of the authentication and encryption sequences.

$S^A$, $S^{iA}$ These are the encryption keys generated as part of the authentication sequence – $S^A$ is generated by CMC and $S_i^A$ is independently generated by CRF. Their value should be identical. The S keys generated by the encryption sequence are ignored.

## 2.2. Authentication & Encryption Sequence

*Table 14. Authentication & Encryption Sequence*

| | CMC Command | CMC Computation | Dir. | CRF Computation | CRF Command |
|---|---|---|---|---|---|
| A. | | | ⇐ | ID, C | Read Config |
| B. | Auth_1 | G = F1($F_n$, $K_{ID}$, ID)<br>$Q^A$ = RNG<br>[$CH^A$, $C^A$, $S^A$] = F2 (G, C, $Q^A$)<br>$CH^A$, $Q^A$ | ⇒ | | |
| C. | | | ⇐ | [CH, $C_i^A$, $S_i^A$] = F2($G_i$, $C_i$, $Q^A$)<br>$CH^A$ =? CH<br>$C_i^A$ | Verify Crypto |
| D. | Auth_2 | $C_i^A$ =? $C^A$ | | | |
| E. | Encrypt_1 | $Q^E$ = RNG<br>[$CH^E$, $C^E$, $S^E$] = F2 ($S^A$, $C^A$, $Q^E$)<br>$CH^E$, $Q^E$ | ⇒ | | |
| F. | | | ⇐ | [CH, $C_i^E$, $S_i^E$] = F2($S_i^A$, $C_i^A$, $Q^E$)<br>$CH^E$ =? CH<br>$C_i^E$ | Verify Crypto |
| G. | Encrypt_2 | $C_i^E$ =? $C^E$ | | | |

# 3. Command Descriptions

## 3.1. VerifyFlash

System sends information to the AT88SC018 which would typically be based on the state of an external nonvolatile (e.g. FLASH) program store. If the input digest indicates a problem, the AT88SC018 will set up the status register to indicate a RstLocked error code but will accept no commands until the next reset or power cycle. This command can be run once only per reset.

If Mode.Bit [1:0] == 00, this command simply verifies that the incoming digest matches that stored in memory. This is useful if the external ASIC has hardware that can verify the boot code, in which case that hardware would respond to the return code of this command.

If Mode.Bit [1:0] == 01, this command implements a simple signature mechanism for an externally loaded module. In this case the FlashDigest stored in EEPROM is a secret also known by the entity that generates legal download images. The system sends both the download digest and the signature to the AT88SC018; the AT88SC018 generates a comparison signature using its stored value and verifies that they are the same. This mode is useful if the external system has some confidence in the boot code, but does not have sufficient space to implement a full public key signature verification module.

If Mode.Bit [1:0] == 11, this command is disabled. If Mode.Bit [1:0] == 00 or 01, then VerifyFlash MUST run before startup. Mode.Bit [1:0] == 10 should not be used, if it is the VerifyFlash command will return OK without any computation or comparison being performed.

*Table 15. Inputs*

| Name | Size | Description |
|---|---|---|
| Digest | 20 | Digest of external memory. |
| Signature | 20 | SHA-1(Digest, FlashDigest), ignored if Mode.Bit [1:0] = 00. |

*Table 16. Outputs*

| Name | Size | Description |
|---|---|---|

## 3.2. Startup

The AT88SC018 resets all internal state, generates a 20 byte random number, and sends to system as challenge start. To permit the system processor to mutually authenticate the AT88SC018, it will also compute a response to a challenge from the system. CmcResponse = SHA-1(CmcChallenge, CmcSecret).

This command can be run only once per reset or power cycle.

*Table 17. Inputs*

| Name | Size | Description |
|---|---|---|
| CmcChallenge | 20 | Authentication challenge to the AT88SC018 from system processor. |

*Table 18. Outputs*

| Name | Size | Description |
|---|---|---|
| SysChallenge | 20 | Authentication challenge to system processor from RNG |
| CmcResponse | 20 | Challenge response to CmcChallenge |

## 3.3. ChallengeResponse

System sends 20 byte challenge response to the AT88SC018. The AT88SC018 computes SHA1 (SysChallenge, SystemSecret) and compares with response. If incorrect, the AT88SC018 locks up until next time the reset pin is asserted or power is removed.

The prior command must have been Startup, or the AT88SC018 will enter the RstLocked state.

*Table 19. Inputs*

| Name | Size | Description |
|---|---|---|
| SysResponse | 20 | Calculated response from system |

*Table 20. Outputs*

| Name | Size | Description |
|---|---|---|

## 3.4. Auth_1

Loads into the AT88SC018 the accessible information about the CRF for which authentication is to be computed and builds the values needed for the CRF chip to perform its authentication sequence. This step computes the values of $C^A$ and $S^A$. These values are retained in volatile registers within the AT88SC018 (named C & S) for use during Auth_2 and Encrypt_1. See Section 2.2 for more details on the authentication algorithm.

Execution of this command automatically resets any previous state including C & S registers, and causes a reset of the crypto engine state.

After execution of Auth_1, the next command must be Auth_2. If it is not, the AT88SC018 locks up for some time. See Section 1.6.3.

*Table 21. Inputs*

| Name | Size | Description |
|---|---|---|
| C | 8 | Initial cryptogram seed from CRF |
| $K_{ID}$ | 16 | Constant value to be included in G calculation. |
| ID | 8 | Serial number from which G is calculated. Referred to as Nc in CRF documentation. |
| Selector | 1 | Selects one of the F values from the EEPROM to be used for authentication. |

*Table 22. Outputs*

| Name | Size | Description |
|---|---|---|
| $Q^A$ | 8 | Random number input to authentication sequence |
| $CH^A$ | 8 | Authentication challenge from Cmc to CRF. |

## 3.5. Auth_2

Receives the output of the CRF authentication command and verifies that the CRF chip has knowledge of G. See Section 2.2 for more details on the authentication algorithm.

If the incoming $Ci^A$ value is incorrect, the AT88SC018 locks up for some time, see Section 1.6.3.

The authentication times out when a delay of 1 second expire, at this point one must re-authenticate.

*Table 23. Inputs*

| Name | Size | Description |
|---|---|---|
| $Ci^A$ | 8 | Authentication response from CRF to the AT88SC018, second half of mutual authentication |

*Table 24. Outputs*

| Name | Size | Description |
|---|---|---|

## 3.6. EncryptPassword

Compute an encrypted password to be sent to the CRF, using the current state of the crypto engine. This can be run at any time after the authentication sequence has completed. This command is optional.

*Table 25. Inputs*

| Name | Size | Description |
|---|---|---|
| Selector | 1 | Which password to use |

*Table 26. Outputs*

| Name | Size | Description |
|---|---|---|
| EncPwd | 3 | Encrypted password to be sent to CRF. |

## 3.7. Encryption_1

Similar to Auth_1, this sequence generates an intermediate value used for subsequent encryption of data to/from CRF. This pass through the crypto engine is similar to the computation done during authentication with the exceptions that G is replaced by S, the input C is replaced with the AT88SC018 register C, and $Q^E$ is newly generated by the RNG on the AT88SC018. See Section 2.2 for more details on the encryption algorithm.

A valid authentication sequence must be run before these commands, which will have set up the C & S registers. This command (and its mate, Encryption_2) can be run multiple times per authentication sequence, but running it more than once will cause the AT88SC018 to be out of synchronization with CRF until the next Auth_1/Auth_2 sequence is run.

After execution of Encryption_1, the next command must be Encryption_2. If not, the AT88SC018 will lock up for a security delay.

*Table 27. Inputs*

| Name | Size | Description |
|---|---|---|

*Table 28. Outputs*

| Name | Size | Description |
|---|---|---|
| $Q^E$ | 8 | Random number for encryption sequence |
| $CH^E$ | 8 | Encryption challenge from AT88SC018 to CRF |

## 3.8. Encryption_2

Similar to Auth_2, this sequence takes the encryption response from CRF and compares it the value computed at the end of Encryption_1.

This command can only be run after the execution of Encryption_1. If the incoming $Ci^E$ value is incorrect, the AT88SC018 locks up for a security delay (refer to Section 1.6.3) and sets the error code in the status register to AuthFail.

*Table 29. Inputs*

| Name | Size | Description |
|---|---|---|
| $Ci^E$ | 8 | Authentication response from CRF to the AT88SC018 |

*Table 30. Outputs*

| Name | Size | Description |
|---|---|---|

## 3.9. GrindBytes

Passes a variable number of bytes through the crypto engine on the AT88SC018 and sends the output of the crypto engine back to the system. This command is used to keep the AT88SC018 in sync with the crypto engine on the CRF chip, to decrypt encrypted data read from CRF, to encrypt data to be written to CRF and to generate or verify a checksum.

The AT88SC018 does not interpret these bytes, merely passes them through the crypto engine.

GrindBytes cannot be run prior to the successful execution of the Auth_2 nor after the execution of the Clear command.

There is a limit of 4096 for maximum number of GrindBytes that can be run per Authentication.

*Table 31. Inputs*

| Name | Size | Description |
|---|---|---|
| Size | 1 | One less than the number of bytes to be sent through crypto engine. If this byte is 0 grind 1 byte, if 0x13 grind 20 bytes. If ≥ 0x14, return BadCmd. |
| Data | — | Crypto engine input bytes, maximum 20. |

*Table 32. Outputs*

| Name | Size | Description |
|---|---|---|
| Data | — | Crypto engine output bytes, maximum 20. |

## 3.10. GetRandom

The AT88SC018 generates a 20 byte random number using its internal high quality random number generator and outputs this value. There is no restriction on the system as to where these random numbers may be used – their cryptographic quality makes them suitable for any operation on the system in addition to the CRF operations.

When the AT88SC018 is unlocked, the random numbers generated will follow a predictable pattern based on the state of the RNGSeed EEPROM value and the number of power cycles since this seed has been written. This mechanism facilitates testing.

*Table 33. Inputs*

| Name | Size | Description |
|---|---|---|

*Table 34. Outputs*

| Name | Size | Description |
|---|---|---|
| Data | 20 | Random bytes from the RNG |

## 3.11. IncrementCounter

Increment the value of the specified counter by 1.

*Table 35. Inputs*

| Name | Size | Description |
|---|---|---|
| Counter | 1 | Counter index to be incremented, must be from 0-3. The upper 4 bits of this parameter are ignored. |

*Table 36. Outputs*

| Name | Size | Description |
|---|---|---|

## 3.12. ReadCounter

Returns the 32 bit current state of the specified counter. There are no read restrictions on the counters.

*Table 37. Inputs*

| Name | Size | Description |
|---|---|---|
| Counter | 1 | Counter index to be read, must be from 0-3. The upper 4 bits of this parameter are ignored. |

*Table 38. Outputs*

| Name | Size | Description |
|---|---|---|
| Value | 4 | Current value of counter. |

## 3.13. WriteMemory

Writes the contents of the specified address and those following it up to the end of the read/write memory space. Prior to locking, any byte after the lock byte can be written with this command. After the AT88SC018 has been locked, only the read/write space can be written with this command.

The input data must always be 16 bytes long, though fewer bytes may be written into the EEPROM. While the AT88SC018 ignores these pad bytes, Atmel recommends that they always be 0xFF.

*Table 39. Inputs*

| Name | Size | Description |
|---|---|---|
| Address | 2 | Address in EEPROM of the first byte of data to be written. The most significant 7 bits are ignored. |
| Count | 1 | If 0, write 1 byte… if 0x0F, write 16 bytes. The upper 4 bits are ignored. |
| Data | 16 | Clear text bytes, padded to 16 bytes total. |

*Table 40. Outputs*

| Name | Size | Description |
|---|---|---|

## 3.14. WriteMemoryEncrypted

Writes a 16 byte page of the EEPROM, using the encryption algorithm described below. Smaller blocks of memory cannot be written using this command.

This command cannot be run after the AT88SC018 has been locked.

*Table 41. Inputs*

| Name | Size | Description |
|---|---|---|
| Address | 2 | Address of the 16 byte page within EEPROM to which data is to be written. The least significant 4 and most significant 7 bits are ignored. |
| Data | 16 | Encrypted data |
| Nonce | 16 | Random value used to seed encryption |

*Table 42. Outputs*

| Name | Size | Description |
|---|---|---|

The AT88SC018 will compute the SHA-1 hash of (Address, EncKey, Nonce). The first 16 bytes of the resulting digest will be used as an XOR key to decrypt the incoming data, which will then be written to the specified page in EEPROM.

## 3.15. ReadMemory

Reads the contents of the EEPROM from the specified address and those following it up to the end of R/W EEPROM. Once locked, only the read-only and read/write spaces can be read. Addresses 0 through 0x2F may never be read.

Up to 16 bytes may be accessed within a single read operation.

This command can be run prior to locking of the memory only if the least two significant bits of the lock byte have a value of 0.

*Table 43. Inputs*

| Name | Size | Description |
|---|---|---|
| Address | 2 | Address in EEPROM of the first byte of data to be read. The most significant 7 bits are ignored. |
| Count | 1 | If 0, read 1 byte… if 0x0F, read 16 bytes. The upper 4 bits are ignored. |

*Table 44. Outputs*

| Name | Size | Description |
|---|---|---|
| Data | — | Clear text bytes, maximum of 16 |

## 3.16. ReadMemoryDigest

Reads the specified 32 byte block from the EEPROM, computes the SHA-1 digest of that block and returns that digest to the user. This command provides a mechanism of verifying that the personalization of the chip completed correctly before the one-time lock has been run.

**Note**: Specifying an address of 0 requires that the verifier know the value of EncKey.

This command cannot be run after the AT88SC018 has been locked or if the unlocked state is Lock[1:0] == 00. When it can be run it can access all locations within the EEPROM.

*Table 45. Inputs*

| Name | Size | Description |
|---|---|---|
| Address | 2 | Address of the 32 byte block within EEPROM which should be read. The least significant 5 and most significant 7 bits are ignored. |

*Table 46. Outputs*

| Name | Size | Description |
|---|---|---|
| Data | 20 | Digest of the selected 32 byte block of the EEPROM |

## 3.17. ReadManufacturingID

Reads the contents of the ManufacturingID and Lock Byte from the EEPROM. This command can always be executed, regardless of whether or not the AT88SC018 has been locked.

*Table 47. Inputs*

| Name | Size | Description |
|---|---|---|

*Table 48. Outputs*

| Name | Size | Description |
|---|---|---|
| MfrID | 16 | ManufacturingID & Lock Byte |

## 3.18. Lock

Locks the current memory values into the AT88SC018, per the description in Section 1.5.1. Once Locked, the AT88SC018 cannot be unlocked. After the execution of this command, the Lock Byte will have a value of 0xFF. This command has no effect on locked parts.

There are no inputs or outputs to this command.

## 3.19. Clear

Clears the current authentication state, empties the C & S registers and prepares the chip for a new authentication. A new startup challenge/response is NOT required. There are no input or output arguments to this command.

After execution of this command, the Auth_1 / Auth_2 sequence must be successfully completed before subsequent execution of EncryptPassword, Encryption_1&2 and/or GrindBytes.

## 3.20. Crunch

Passes a random number of 8 bytes through the crunch engine on the AT88SC018 and sends the output of the crunch engine back to the system. This command is used to ensure the AT88SC018 is talking with an actual CRF chip, which should respond with the same answer in the given timeframe.

The AT88SC018 does not interpret these bytes, merely passes them through the crunch engine.

*Table 49. Inputs*

| Name | Size | Description |
|---|---|---|
| Iterations | 1 | A maximum of 255 iterations can be run through the crunch engine. A 1 in this filed will compute one Iteration through the crunch engine. |
| Data | 8 | Crunch engine input bytes. |

*Table 50. Outputs*

| Name | Size | Description |
|---|---|---|
| Data | 8 | Crunch engine output bytes. |

## 4. Command Execution Times

The following table lists the nominal execution times for the various commands above, subject to the assumptions following the table.

Some of the commands take a variable amount of time based on the input parameters and/or the current state of the AT88SC018. In general, the table below shows the worst case operational flow, subject to the list of assumptions following the table. Actual execution time will vary from the nominal by ±25% due to variations of the internal oscillator.

This preliminary data is advisory in nature. Designs should not depend on the specific execution times below, but rather use the standard handshake mechanisms described above. The values below are characterized on the part but are not tested in production.

*Table 51. Nominal Execution Times*

| Command | Nominal Time | Notes |
|---|---|---|
| VerifyFlash | 4000 µs | |
| Startup | 8000 µs | |
| ChallengeResponse | 4000 µs | |
| Auth_1 | 8000 µs | |
| Auth_2 | 60 µs | |
| EncryptPassword | 100 µs | |
| Encryption_1 | 4100 µs | |
| Encryption_2 | 60 µs | |
| GrindBytes | 50 µs | |
| GetRandom | 4000 µs | |
| IncrementCounter | 50 µs + 10 ms | (5 EE writes worst case) |
| ReadCounter | 50 µs | |
| WriteMemory | 200 µs + 4 ms | (2 EE writes, if not within a page) |
| WriteMemoryEncrypted | 4100 µs + 2 ms | (1 EE write) |
| ReadMemory | 200 µs | |
| ReadMemoryDigest | 4000 µs | |
| ReadManufacturingID | 200 µs | |
| Lock | 8000 µs + 36 ms | (18 EE write worst case) |
| Clear | 5 µs | |

Assumptions:

1. TWI clock assumed to be at 400 KHz.
2. TWI command times – 0 bytes of data ~ 75 µs. Additional byte ~ 25 µs.
3. VerifyFlash command is run with "Mode.Bit [1:0] = 01" case.
4. GrindBytes command assumes 20 bytes of data.
5. WriteMemory and ReadMemory commands assume 16 bytes of data.
6. These processing times do not include data transfer on the TWI.

## 5. AC & DC Characteristics

*Table 52. DC Characteristics* [1]

Applicable over recommended operating range from $V_{CC}$ = +2.7 to 3.6 V,
$T_{AC}$ = -40° C to 85° C (unless otherwise noted)

| Symbol | Parameter | Test Condition | Min | Typ | Max | Units |
|---|---|---|---|---|---|---|
| $V_{CC}$ | Supply Voltage | | 2.7 | | 3.6 | V |
| $I_{CC}$ | Supply Current | 400kHz | | | 5 | mA |
| $I_{SB}$ | Standby Current | $V_{IN} = V_{CC}$ or GND | | | 15 | µA |
| $V_{IL}$ | SDA Input Low Voltage | | -0.3 | | $V_{CC}$ x 0.3 | V |
| $V_{IL}$ | CLK Input Low Voltage | | -0.3 | | $V_{CC}$ x 0.3 | V |
| $V_{IL}$ | RST Input Low Voltage | | -0.3 | | $V_{CC}$ x 0.3 | V |
| $V_{IL}$ | PDN Input Low Voltage | | -0.3 | | $V_{CC}$ x 0.3 | V |
| $V_{IH}$ | SDA Input High Voltage | | $V_{CC}$ x 0.7 | | 5.25 | V |
| $V_{IH}$ | SCL Input High Voltage | | $V_{CC}$ x 0.7 | | 5.25 | V |
| $V_{IH}$ | RST Input High Voltage | | $V_{CC}$ x 0.7 | | 5.25 | V |
| $V_{IH}$ | PDN Input High Voltage | | $V_{CC}$ x 0.7 | | 5.25 | V |
| $I_{IL}$ | SDA Input Low Current | $0 < V_{IL} < V_{CC}$ x 0.15 | -10 | | 10 | µA |
| $I_{IL}$ | SCL Input Low Current | $0 < V_{IL} < V_{CC}$ x 0.15 | -10 | | 10 | µA |
| $I_{IL}$ | RST Input Low Current | $0 < V_{IL} < V_{CC}$ x 0.15 | -10 | | 10 | µA |
| $I_{IL}$ | PDN Input Low Current | $0 < V_{IL} < V_{CC}$ x 0.15 | -10 | | 10 | µA |
| $I_{IH}$ | SDA Input High Current | $V_{CC}$ x 0.7 < $V_{IH}$ < $V_{CC}$ | -10 | | 10 | µA |
| $I_{IH}$ | SCL Input High Current | $V_{CC}$ x 0.7 < $V_{IH}$ < $V_{CC}$ | -10 | | 10 | µA |
| $I_{IH}$ | RST Input High Current | $V_{CC}$ x 0.7 < $V_{IH}$ < $V_{CC}$ | -10 | | 10 | µA |
| $I_{IH}$ | PDN Input High Current | $V_{CC}$ x 0.7 < $V_{IH}$ < $V_{CC}$ | -10 | | 10 | µA |
| $V_{OH}$ | SDA Output High Voltage | 20k Ohm External Pull-up | | | $V_{CC}$ x 0.8 | V |
| $V_{OL}$ | SDA Output Low Voltage | $I_{OL}$ = 1mA, Vcc=2.7V | | | 0.4 | V |

***Note:*** *1. Typical values at 25° C. Maximum values are characterized values and not test limits in production.*

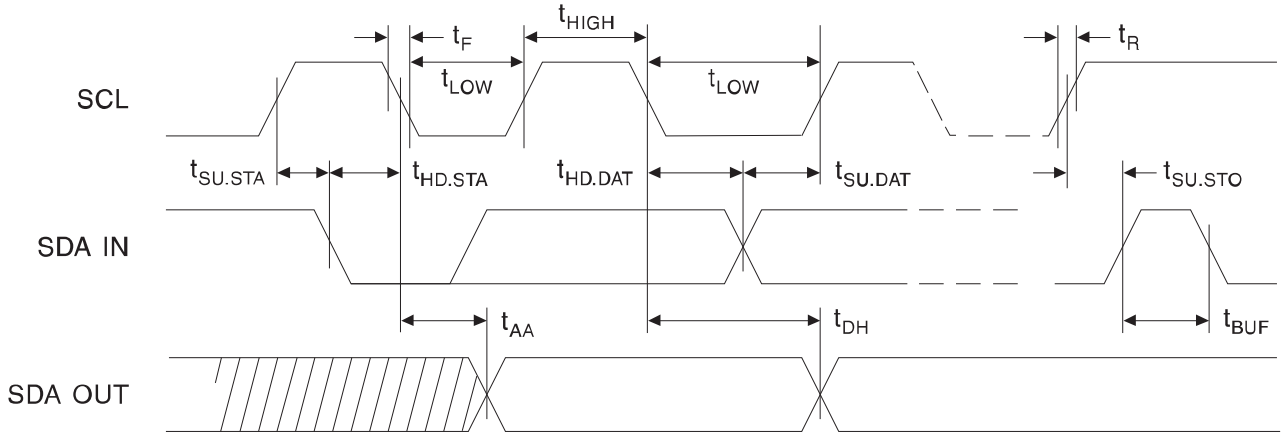
*Table 53. AC Characteristics* (1)

Applicable over recommended operating range from $V_{CC}$ = +2.7 to 3.6 V,
$T_{AC}$ = -40° C to 85° C, CL = 30pF (unless otherwise noted)

| Symbol | Parameter | Min | Max | Units |
|---|---|---|---|---|
| $f_{CLK}$ | Clock Frequency | 0 | 400 | kHz |
| | Clock Duty cycle (2) | 40 | 60 | % |
| $t_R$ | Rise Time - SDA, RST, PDN (2) | | 300 | nS |
| $t_F$ | Fall Time - SDA, RST, PDN (2) | | 300 | nS |
| $t_R$ | Rise Time - SCL (2) | | 300 | nS |
| $t_F$ | Fall Time - SCL (2) | | 300 | nS |
| $t_{AA}$ | Clock Low to Data Out Valid | | 900 | nS |
| $t_{HD.STA}$ | Start Hold Time | 600 | | nS |
| $t_{SU.STA}$ | Start Set-up Time | 600 | | nS |
| $t_{HD.DAT}$ | Data In Hold Time | 100 | | nS |
| $t_{SU.DAT}$ | Data In Set-up Time | 100 | | nS |
| $t_{SU.STO}$ | Stop Set-up Time | 600 | | nS |
| $t_{DH}$ | Data Out Hold Time | 50 | 900 | nS |

***Note:*** *1. Typical values at 25° C. Maximum values are characterized values and not test limits in production.*
*2. This parameter is not tested. Values are based on characterization and/or simulation data.*

*Figure 3. SCL: Serial Clock, SDA: Serial Data I/O®*



## 6. Transport Key

Certain operational modes of CryptoCompanion chip require knowledge of a key for proper custom configuration. When applicable, Atmel shall program customer provided key values at the factory for production orders. For generic and sample orders, this key, available as a transport key, shall be:

0x17 0x44 0x1A 0x48 0xDA 0xDB 0x23 0xFB 0x70 0xCC 0xB8 0x43 0x09 0x20 0x59 0xEB

## 7. Ordering Codes

*Table 54. Ordering Codes*

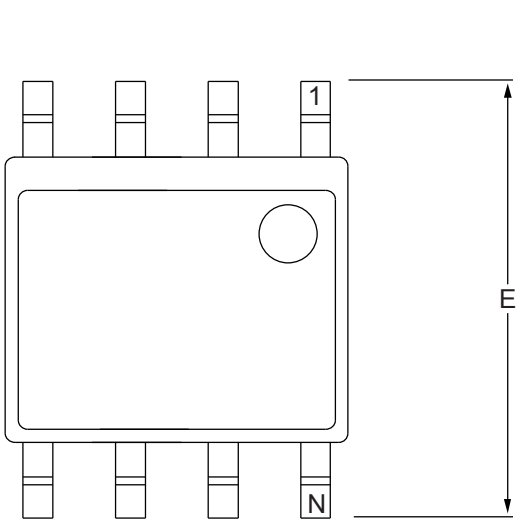
| Ordering Code | Package | Voltage Range | Memory Locking *(see Section 1.5.1 for Lock Definitions)* | Temperature Range |
|---|---|---|---|---|
| AT88SC018-SU-CM | 8S1 | 2.7V – 3.6V | 00 (Unlocked) | Green compliant (exceeds RoHS), Industrial (-40° C to 85° C), Bulk |
| AT88SC018-SU-CM-T | 8S1 | 2.7V – 3.6V | 00 (Unlocked) | Green compliant (exeeds RoHS), Industrial (-40° C to 85° C), Tape and Reel |
| AT88SC018-SU-CN | 8S1 | 2.7V – 3.6V | 10 (Unlocked/Confidential) | Green compliant (exceeds RoHS), Industrial (-40° C to 85° C), Bulk |
| AT88SC018-SU-CN-T | 8S1 | 2.7V – 3.6V | 10 (Unlocked/Confidential) | Green compliant (exceeds RoHS), Industrial (-40° C to 85° C), Tape and Reel |

*Table 55. Package Type*

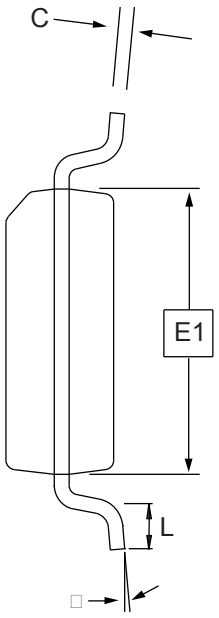
| Package Type | Description |
|---|---|
| 8S1 | 8-lead, 0.150” Wide, Plastic Gull Wing Small Outline Package (JEDEC SOIC) |

## 8. Package Drawing

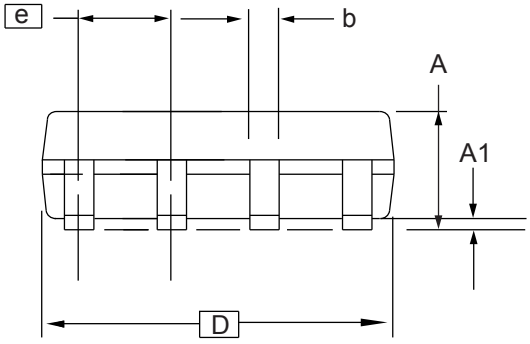
*Figure 4. 8S1 – JEDEC SOIC*




Top View




End View




Side View

**COMMON DIMENSIONS**
(Unit of Measure = mm)

| SYMBOL | MIN | NOM | MAX | NOTE |
|---|---|---|---|---|
| A | 1.35 | – | 1.75 | |
| A1 | 0.10 | – | 0.25 | |
| b | 0.31 | – | 0.51 | |
| C | 0.17 | – | 0.25 | |
| D | 4.80 | – | 5.05 | |
| E1 | 3.81 | – | 3.99 | |
| E | 5.79 | – | 6.20 | |
| e | 1.27 BSC | | | |
| L | 0.40 | – | 1.27 | |
| □ | 0° | – | 8° | |

Note: These drawings are for general information only. Refer to JEDEC Drawing MS-012, Variation AA for proper dimensions, tolerances, datums, etc.

10/07/03

| | | | |
|---|---|---|---|
| 1150 E. Cheyenne Mtn. Blvd.<br>Colorado Springs, CO 80906 | **TITLE**<br>**8S1,** 8-lead (0.150" Wide Body), Plastic Gull Wing Small Outline (JEDEC SOIC) | **DRAWING NO.**<br>8S1 | **REV.**<br>B |

## 9. Command Flow Diagrams

*Figure 5. Command Input*

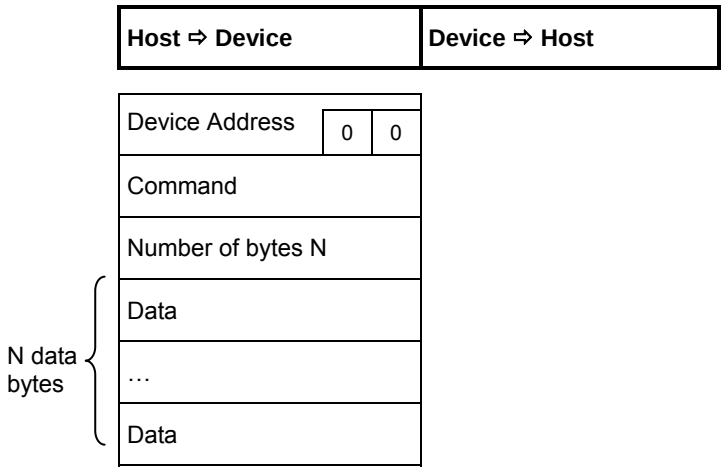



*Figure 6. Command Output*

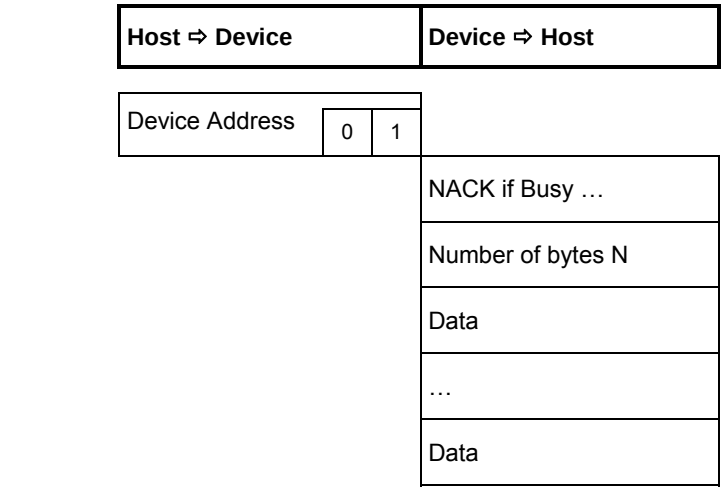
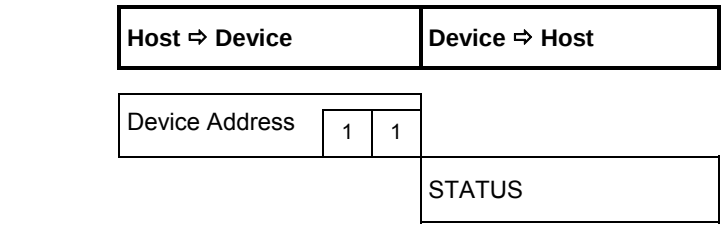



*Figure 7. Command Status*



## Appendix A. Revision History

| Doc. Rev. | Date | Comments |
|---|---|---|
| 5277C | 9/2009 | Finalized AC & DC Characteristics. Updated Counter information. |
| 5277B | 2/2009 | Document updated. Changed to AT88SC018 part number. |
| 5277A | 2/2008 | Initial document release. |

## Headquarters


***Atmel Corporation***
2325 Orchard Parkway
San Jose, CA 95131
USA
Tel: 1(408) 441-0311
Fax: 1(408) 487-2600


## International


***Atmel Asia***
Room 1219
Chinachem Golden Plaza
77 Mody Road Tsimshatsui
East Kowloon
Hong Kong
Tel: (852) 2721-9778
Fax: (852) 2722-1369

***Atmel Europe***
Le Krebs
8, Rue Jean-Pierre Timbaud
BP 309
78054 Saint-Quentin-en-
Yvelines Cedex
France
Tel: (33) 1-30-60-70-00
Fax: (33) 1-30-60-71-11

***Atmel Japan***
9F, Tonetsu Shinkawa Bldg.
1-24-8 Shinkawa
Chuo-ku, Tokyo 104-0033
Japan
Tel: (81) 3-3523-3551
Fax: (81) 3-3523-7581


## Product Contact

***Web Site***
www.atmel.com

***Technical Support***
cryptomemory@atmel.com

***Sales Contact***
www.atmel.com/contacts

***Literature Requests***
www.atmel.com/literature


**Disclaimer:** The information in this document is provided in connection with Atmel products. No license, express or implied, by estoppel or otherwise, to any intellectual property right is granted by this document or in connection with the sale of Atmel products. **EXCEPT AS SET FORTH IN ATMEL'S TERMS AND CONDITIONS OF SALE LOCATED ON ATMEL'S WEB SITE, ATMEL ASSUMES NO LIABILITY WHATSOEVER AND DISCLAIMS ANY EXPRESS, IMPLIED OR STATUTORY WARRANTY RELATING TO ITS PRODUCTS INCLUDING, BUT NOT LIMITED TO, THE IMPLIED WARRANTY OF MERCHANTABILITY, FITNESS FOR A PARTICULAR PURPOSE, OR NON-INFRINGEMENT. IN NO EVENT SHALL ATMEL BE LIABLE FOR ANY DIRECT, INDIRECT, CONSEQUENTIAL, PUNITIVE, SPECIAL OR INCIDEN-TAL DAMAGES (INCLUDING, WITHOUT LIMITATION, DAMAGES FOR LOSS OF PROFITS, BUSINESS INTERRUPTION, OR LOSS OF INFORMATION) ARISING OUT OF THE USE OR INABILITY TO USE THIS DOCUMENT, EVEN IF ATMEL HAS BEEN ADVISED OF THE POSSIBILITY OF SUCH DAMAGES.** Atmel makes no representations or warranties with respect to the accuracy or completeness of the contents of this document and reserves the right to make changes to specifications and product descriptions at any time without notice. Atmel does not make any commitment to update the information contained herein. Unless specifically provided otherwise, Atmel products are not suitable for, and shall not be used in, automotive applications. Atmel's products are not intended, authorized, or warranted for use as components in applications intended to support or sustain life.

© 2009 Atmel Corporation. All rights reserved. Atmel®, Atmel logo and combinations thereof, CryptoMemory®, CryptoRF®, and others are registered trademarks, CryptoCompanion™, and others are trademarks of Atmel Corporation or its subsidiaries. Other terms and product names may be trademarks of others.