CryptoAuthentication™

AT88SA102S

Product Authentication Chip

Preliminary

Features

- **Secure authentication & key exchange**
- **Superior SHA-256 Hash Algorithm**
- **Best in class 256 bit key length**
- **Guaranteed Unique 48 bit Serial Number**
- **High speed single wire interface**
- **Supply Voltage: 2.5 – 5.5V**
- **1.8 – 5.5 V Communications**
- **<100nA Sleep Current**
- **4KV ESD protection**
- **Multi-level hardware security**
- **Secure personalization**
- **Green compliant (exceeds RoHS) 3 pin SOT-23 package**

Applications

- **Authentication of Replaceable Items**
- **Software anti-piracy**
- **Network & Computer Access control**
- **Portable Media Player & GPS System**
- **Key exchange for encrypted downloads**
- **Prevention of clones for demo and eval boards**
- **Authenticated communications for control networks**
- **Anti-clone authentication for daughter cards**
- **Physical access control (electronic lock & key)**

1. Introduction

The AT88SA102S is a member of the CryptoAuthentication family of cost-effective authentication chips designed to securely authenticate an item to which it is attached. It can also be used to exchange session keys with some remote entity so that the system microprocessor can securely encrypt/decrypt data. Each CryptoAuthentication chip contains a pre-programmed serial number which is guaranteed to be unique. In addition, it has been designed to permit secure personalization so that third parties can build devices containing an OEM secret without concern for the theft of that secret.

It is the first small standard product to implement the SHA-256 hash algorithm, which is part of the latest set of recommended algorithms by the US Government. The 256 bit key space renders any exhaustive attacks impossible.

The CryptoAuthentication family uses a standard challenge response protocol to simplify programming. The system generates a random number challenge and sends it to the CryptoAuthentication chip. The chip hashes that with a 256 bit key using the SHA-256 algorithm to generate a keyed 256 bit response which is sent back to the system.

The chip includes 128 single bit one time programmable fuses that can be used for personalization, status or consumption logging. Atmel programs 40 of these bits prior to the chip leaving the factory, leaving 88 for user purposes. Refer to Section 1.3 for more information.

Note: The chip implements a failsafe internal watchdog timer that forces it into a very low power mode after a certain time interval regardless of any command execution or IO transfers that may be happening at the time the timer expires. System programming must take this into consideration. Refer to Section 3.5 for more details.

1.1. Usage

There are many different ways in which CryptoAuthentication can add an authentication capability to a system. For more information, refer to the “CryptoAuthentication Usage Examples” Applications Note.

In general, however, all these security models usually employ one of two general key management strategies:

- Fixed challenge response number pair stored in the host. In this case, the host sends its particular challenge and only an authentic CryptoAuthentication can generate the correct response. Since no secret is stored on the host, there is no security cost on the host. Depending on the particulars of the system, each host may have a different challenge response pair and/or each client may have the same key.
- Host computes the response that should be provided for a particular client against a random challenge and/or include the client ID# in the calculation. In this case, the host needs to have the capability to securely store the secret from which diversified response will be computed. One way to do this is to use a CryptoAuthentication host chip. Since each client is unique, the host can maintain a dynamic black list of clients that have been found to be fraudulent.

1.2. Memory Resources

Fuse Block of 128 fuse bits that can be written through the 1 wire interface. Fuse[1] and Fuse[87] have special meanings, refer to Section 1.3 for more details. Fuse[88:95] are part of the manufacturing ID value fixed by Atmel. Fuse[96:127] are part of the serial number programmed by Atmel which is guaranteed to be unique. See Section 1.4 for more details on the Manufacturing ID and Serial Number.

ROM Metal mask programmed memory. Unrestricted reads are permitted on the first 64 bits of this array. The physical ROM will be larger and will contain other information that cannot be read.

ROM MfrID 2 bytes of ROM that specifies part of the manufacturing ID code. This value is assigned by Atmel and is always the same for all chips of a particular model number. For the AT88SA102S, this value is 0xFF FF. ROM MfrID can be read by accessing ROM bytes 0 & 1 of Address 0.

ROM SN 2 bytes of ROM that can be used to identify chips among others on the wafer. These bits reduce the number of fuses necessary to construct a unique serial number. The ROM SN is read by accessing ROM bytes 2 & 3 of Address 0. ROM SN can always be read by the system and is optionally included in the message digested by the MAC command.

RevNum 4 bytes of ROM that are used by Atmel to identify the model mask and/or design revision of the AT88SA102S chip. These bytes can be freely read as the four bytes returned ROM address 1, however system code should not depend on this value as it may change from time to time.

1.3. Fuse Map

The AT88SA102S incorporates 128 one-time fuses within the chip. Once burned, there is no way to reset the value of a fuse. Fuses, with the exception of the manufacturer ID and serial number bits initialized by Atmel, have a value of 1 when shipped from the Atmel factory and transition to a 0 when they are burned. Bits 0-63 can never be read, while bits 64-128 can always be read.

Table 1. The 128 fuses in the AT88SA102S chip are arranged in the following manner:

Fuse #	Name	Description
1	BurnFuse Enable	If this fuse is a 1, then the BurnFuse command is enabled. If it is burned to 0, then the BurnFuse command is disabled.
0 & 2 → 63	Secret Fuses	These fuses can be securely written by the BurnSecure command but can never be read directly with the Read command.
64 → 86	Status Fuses	These fuses can be written with the BurnSecure command and can always be read with the Read command.
87	Fuse Disable	The MAC command ignores the values of Fuse[0-86] while this fuse is a 1. Once it is burned to 0, the BurnSecure command is disabled.
88 → 95	Fuse MfrID	See Section 1.4. Set by Atmel, can't be modified in the field.
96 → 127	Fuse SN	See Section 1.4. Set by Atmel, can't be modified in the field.

BurnFuse Enable This fuse is used to prevent operation of the BurnFuse command in the application. This fuse may only be burned to 0 using the BurnSecure command.

Secret Fuses These 63 fuses are used to augment the keys stored elsewhere in the chip. Knowledge of both the internally stored keys and the values of the Secret Fuses is required to generate the correct response to the Cryptographic command of the AT88SA102S. An arbitrary selection of these fuses is burned during personalization via the BurnSecure command.

Within this document, "Secret Fuses" is used to refer to the entire array of 64 bits: Fuse[0-63], even though the value of Fuse[1] is fixed for most applications and its value can be derived from the operation of the chip.

Status Fuses These 23 fuses can be used to store information which is not secret, as their value can always be determined using the Read command. They can be written at the same time as the secret fuses using the BurnSecure command, or they can be individually burned at a later time with the BurnFuse command. Two common usage models for these fuses are:

1. Calibration or model number information. In this situation, the 23 bits are written at the factory. This method can also be used for feature enabling. In this case, the BurnFuse command should not be run in the field, and the BurnFuse Enable bit should be a 0.
2. Consumption logging, i.e. burn one bit after every n uses, the host system keeps track of the number of uses so far for this serial number. In this case, the BurnFuse command is necessary to individually burn one of these 23 bits, and the BurnFuse Enable bit should be a 1.

Within this document, "Status Fuses" is used to refer to the entire array of 24 bits: Fuse[64-87], even though the value of Fuse[87] is fixed after personalization and cannot be modified in the field.

Fuse Disable This fuse is used to disable/enable the ability of the MAC command to read the fuse values until the BurnSecure command has completed properly. When it has a value of 1 (unburned), the bit values in the message that would normally have been filled in with Fuse values are all set to a 1. When FuseDisable is burned, the MAC command fills in the message with the requested fuse values. Additionally, this bit, when burned, disables the BurnSecure command to prevent modification of the secret fuses and BurnFuse enable bit in the end customer application.

1.4. Chip Identification

The chip includes a total of 72 bits of information that can be used to distinguish between individual chips in a reliable manner. The information is distributed between the ROM and fuse blocks in the following manner.

Serial Number This 48 bit value is composed of ROM SN (16 bits) and Fuse SN (32 bits). Together they form a serial number that is guaranteed to be unique for all devices ever manufactured within the CryptoAuthentication family. This value is optionally included in the MAC calculation.

Manufacturing ID This 24 bit value is composed of ROM MfrID (16 bits) and Fuse MfrID (8 bits). Typically this value is the same for all chips of a given type. It is always included in the cryptographic computations.

1.5. Key Values

The values stored in the AT88SA102S internal key array are hardwired into the masking layers of the chip during wafer manufacture. All chips have the same keys stored internally, though the value of a particular key cannot be determined externally from the chip. For this reason, customers should ensure that they program a unique (and secret) number into the 64 secret fuses and they should store the Atmel provided key values securely.

Individual key values are made available to qualified customers upon request to Atmel and are always transmitted in a secure manner.

When the serial number is included in the MAC calculation then the response is considered to be diversified and the host needs to know the base secret in order to be able to verify the authenticity of the client. A diversified response can also be obtained by including the serial number in the computation of the value written to the secret fuses. An Atmel CryptoAuthentication host chip provides a secure hardware mechanism to validate responses to determine if they are authentic.

1.6. SHA-256 Computation

CryptoAuthentication performs only one cryptographic calculation – a keyed digest of an input challenge. It optionally includes various other information stored on the chip within the digested message.

CryptoAuthentication computes the SHA-256 digest based on the algorithm documented here:

http://csrc.nist.gov/publications/fips/fips180-2/fips180-2.pdf

Throughout this document, the complete message processed by the AT88SA102S chip is documented. According to the above specification, this always includes a single bit of ‘1’ pad after the message, followed by a 64 bit value representing the total number of bits being hashed (less pad and length). If the length is less than 447 (512-64-1) then the necessary number of ‘0’ bits are included between the ‘1’ pad and ‘length’ to stretch the last message block out to 512 bits.

When using standard libraries to calculate the SHA-256 digest, these pad and length bits should probably not be passed to the library as most standard software implementations of the algorithm add them in automatically.

1.6.1. SHA Computation Example

In order to ensure that there is no ambiguity, the following example vector is provided in addition to the sample vectors in the NIST document. In this example, all values are listed in hex format. For all but the key, bytes are listed in the order that they appear on the bus – first on the bus is listed on the left side of the page. The key value below is listed in the same order as the challenge, so the 01 at the left of the key string corresponds to the first byte in the SHA-256 document.

Key	01030507090B0D0F11131517191B1D1F21232527292B2D2F31333537393B3D3F
Challenge	020406080A0C0E10121416181A1C1E20222426282A2C2E30323436383A3C3E40

Opcode	08	
Mode	50	(all optional information included in message)
KeyID	FFFF	

Secret Fuses	0000111122223333
Status Fuses	445566
Fuse MfrID	77
Fuse SN	8899AABB

ROMMfrID	CCDD
ROM SN	EEFF

The 88 bytes over which the digest is calculated are:

0103…3D3F0204…3E400850FFFF00001111…EEFF

And the resulting digest is:

6CA7129C8DA9CE80EA6357DDCFB1DDCBBBD89ED373419A5A332D728B42642C62

1.7. Security Features

CryptoAuthentication incorporates a number of physical security features designed to protect the keys from release. These include an active shield over the entire surface of the part, internal memory encryption, internal clock generation, glitch protection, voltage tamper detection and other physical design features.

Pre-programmed keys stored on the AT88SA102S are encrypted in such a way as to make retrieval of their values via outside analysis very difficult.

Both the clock and logic supply voltage are internally generated, preventing any direct attack via the pins on these two signals.

2. IO Protocol

Communications to and from CryptoAuthentication take place over a single asynchronously timed wire using a pulse count scheme. The overall communications structure is a hierarchy:

Table 2. IO Hierarchy

Tokens	Implement a single data bit transmitted on the bus, or the wake-up event.
Flags	Comprised of eight tokens (bits) which convey the direction and meaning of the next group of bits (if any) which may be transmitted.
Blocks	Of data follow the command and transmit flags. They incorporate both a byte count and a checksum to ensure proper data transmission
Packets	Of bytes form the core of the block without the count and CRC. They are either the input or output parameters of a CryptoAuthentication command or status information from CryptoAuthentication

Refer to Applications Notes on Atmel's website for more details on how to use any microprocessor to easily generate the signaling necessary to send these values to the chip.

2.1. IO Tokens

There are a number of IO **tokens** that may be transmitted along the bus:

Input: (To CryptoAuthentication)

Wake Wake CryptoAuthentication up from sleep (low power) state

Zero Send a single bit from system to CryptoAuthentication with a value of 0

One Send a single bit from system to CryptoAuthentication with a value of 1

Output: (From CryptoAuthentication)

ZeroOut Send a single bit from CryptoAuthentication to the system with a value of 0

OneOut Send a single bit from CryptoAuthentication to the system with a value of 1

The waveforms are the same in either direction, however there are some differences in timing based on the expectation that the host has a very accurate and consistent clock while CryptoAuthentication has significant part to part variability in its internal clock generator due to normal manufacturing and environmental fluctuations.

The bit timings are designed to permit a standard UART running at 230.4K baud to transmit and receive the tokens efficiently. Each byte transmitted or received by the UART corresponds to a single bit received or transmitted by CryptoAuthentication. Refer to Applications Notes on Atmel's website for more details.

2.2. AC Parameters

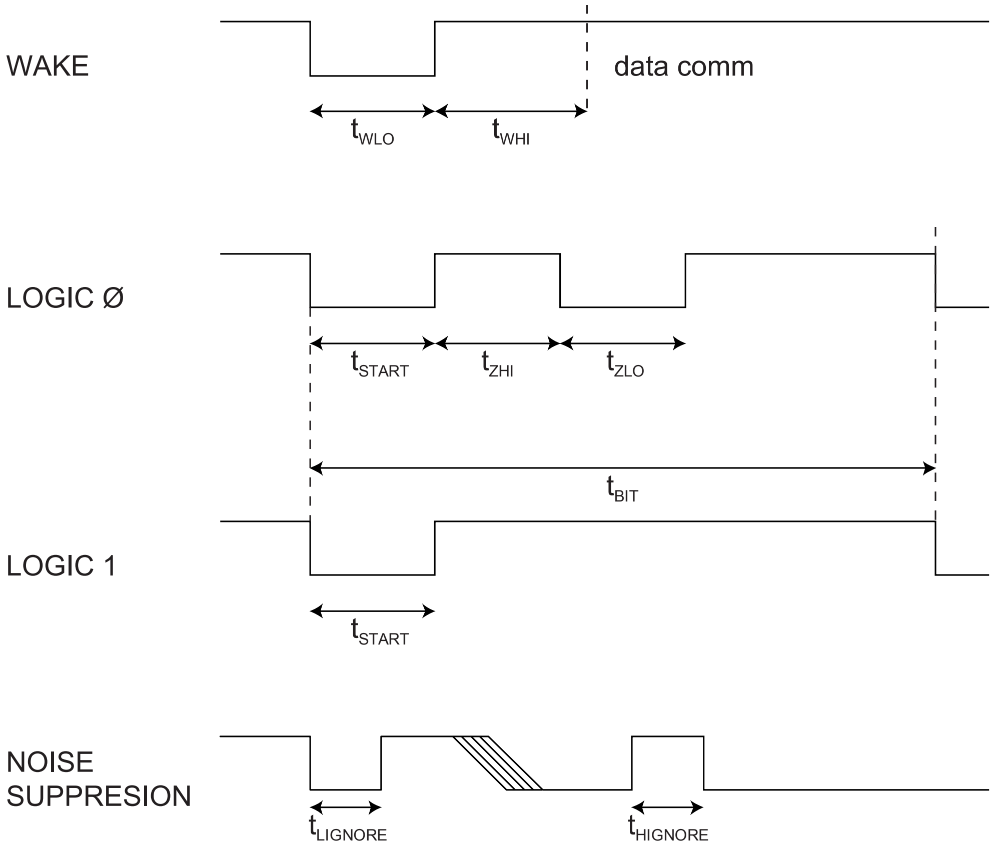


Table 3. AC Parameters

Parameter	Symbol	Direction	Min	Typ	Max	Unit	Notes
Wake Low Duration	t_{WLO}	To CryptoAuthentication	60		-	µs	Signal can be stable in either high or low levels during extended sleep intervals.
Wake Delay to Data Comm.	t_{WHI}	To CryptoAuthentication	1		-	ms	Signal should be stable high for this entire duration.
Start pulse duration	t_{START}	To CryptoAuthentication	4.1	4.34	4.56	µs	
		From CryptoAuthentication	4.62	6.0	8.6	µs	
Zero transmission high pulse	t_{ZHI}	To CryptoAuthentication	4.1	4.34	4.56	µs	
		From CryptoAuthentication	4.62	6.0	8.6	µs	
Zero transmission low pulse	t_{ZLO}	To CryptoAuthentication	4.1	4.34	4.56	µs	
		From CryptoAuthentication	4.62	6.0	8.6	µs	
Bit time‡	t_{BIT}	To CryptoAuthentication	37.1	39	-	µs	If the bit time exceeds $t_{TIMEOUT}$ then CryptoAuthentication will enter sleep mode and the wake token must be resent.
		From CryptoAuthentication	46.2	60	86	µs	
Turn around delay	$t_{TURNAROUND}$	From CryptoAuthentication	46.2	60	86	µs	CryptoAuthentication will initiate the first low going transition after this time interval following the end of the Transmit flag
		To CryptoAuthentication	46.2	60	86	µs	After CryptoAuthentication transmits the last bit of a block, system must wait this interval before sending the first bit of a flag
High side glitch filter @ active	$t_{HIGNORE\_A}$	To CryptoAuthentication	45			ns	Pulses shorter than this in width will be ignored by the chip, regardless of its state when active
Low side glitch filter @ active	$t_{LIGNORE\_A}$	To CryptoAuthentication	45			ns	Pulses shorter than this in width will be ignored by the chip, regardless of its state when active
High side glitch filter @ sleep	$t_{HIGNORE\_S}$	To CryptoAuthentication	2			µs	Pulses shorter than this in width will be ignored by the chip when in sleep mode
Low side glitch filter @ sleep	$t_{LIGNORE\_S}$	To CryptoAuthentication	2			µs	Pulses shorter than this in width will be ignored by the chip when in sleep mode
IO Timeout	$t_{TIMEOUT}$	To CryptoAuthentication	7	10	13	ms	Starting as soon as 7ms up to 13ms after the initial signal transition of a token the chip will enter sleep if no complete & valid token is received.
Watchdog reset	$t_{WATCHDOG}$	To CryptoAuthentication	3	4	5.2	s	Max. time from wake until chip is forced into sleep mode. Refer to Watchdog Failsafe Section 3.5

‡ START, ZLO, ZHI & BIT are designed to be compatible with a standard UART running at 230.4K baud for both transmit and receive.

3. DC Parameters

Table 4. DC Parameters

Parameter	Symbol	Min	Typ	Max	Unit	Notes
Operating temperature	T_A	-40		85	°C	
Power Supply Voltage	V_{cc}	2.5		5.5	V	
Fuse Burning Voltage	V_{BURN}	3.0		5.5	V	Voltage applied to V_{cc} pin during BurnSecure and/or BurnFuse
Active Power Supply Current	I_{CC}		-	10	mA	
Sleep Power Supply Current	I_{SLEEP}			100	nA	When chip is in sleep mode, V_{cc} = 3.7V, Vsig = 0.0 to 0.5V or Vsig = V_{cc}-0.5V to V_{cc}.
Input Low Voltage @ V_{cc} = 5.5V	V_{IL}	-0.5		.15 * V_{cc}	V	Voltage levels for wake token when chip is in sleep mode
Input Low Voltage @ V_{cc} = 2.5V	V_{IL}	-0.5		0.5	V	Voltage levels for wake token when chip is in sleep mode
Input High Voltage @ V_{cc} = 5.5V	V_{IH}	.25 * V_{cc}		6.0	V	Voltage levels for wake token when chip is in sleep mode
Input High Voltage @ V_{cc} = 2.5V	V_{IH}	1.0		3.0	V	Voltage levels for wake token when chip is in sleep mode
Input Low Voltage when Active	V_{IL}	-0.5		0.5	V	When chip is in active mode, V_{cc} = 2.5 – 5.5V
Input High Voltage when Active	V_{IH}	1.2		6.0	V	When chip is in active mode, V_{cc} = 2.5 – 5.5V
Output Low voltage	V_{OL}			0.4	V	When chip is in active mode, V_{cc} = 2.5 – 5.5V
Output Low current	I_{OL}			4	mA	When chip is in active mode, V_{cc} = 2.5 – 5.5V, V_{OL} = 0.4V
Maximum Input Voltage	V_{MAX}			V_{cc} + 0.5	V	
ESD	V_{ESD}		4		KV	Human Body Model, Sig & V_{cc} pins.

3.1. IO Flags

The system is always the bus master, so before any IO transaction, the system must send an 8 bit **flag** to the chip to indicate the IO operation that is to be performed, as follows:

Value	***Name***	***Meaning***
0x77	Command	After this flag, the system starts sending a command block to the chip. The first bit of the block can follow immediately after the last bit of the flag.
0x88	Transmit	After a turn-around delay, the chip will start transmitting the response block for a previously transmitted command block.
0xCC	Sleep	Upon receipt of a sleep flag, the chip will enter a low power mode until the next wake token is received.

All other values are reserved and will be ignored.

As the single signal wire may be shared with a CryptoAuthentication host chip, the AT88SA102S chip includes a PauseLong command which causes it to ignore all activity on the signal pin until the expiration of the watchdog timer.

3.1.1. Command Timing

After a command flag is transmitted, a command block should be sent to the chip. During parsing of the parameters and subsequent execution of a properly received command, the chip will be busy and not respond to transitions on the signal pin. The delays for these operations are listed in the table below:

Table 5. Command Timing

Parameter	Symbol	Min	Max	Unit	Notes
Parsing Delay	t_{PARSE}	0	50	µs	Delay to check CRC and parse opcode and parameters before an error indication will be available
MacDelay	t_{EXEC_MAC}	15	30	ms	Delay to execute MAC command
MemoryDelay	t_{EXEC_READ}	50	100	µs	Delay to execute Read command
Fuse Delay	t_{EXEC_FUSE}	190	400	µs	Delay to execute BurnFuse command at Vcc > 4.5V. See Section 4.3 for more details.
SecureDelay	t_{EXEC_SECURE}	13	29	ms	Max delay to execute BurnSecure command at Vcc > 4.5V. See Section 4.5 for more details.
PersonalizeDelay	t_{PERSON}	7	15	ms	Delay to execute GenPersonalizationKey

In this document, t_{EXEC} is used as shorthand for the delay corresponding to whatever command has been sent to the chip.

3.1.2. Transmit Flag

The transmit flag is used to turn around the signal so that CryptoAuthentication can send data back to the system, depending on its current state. The bytes that CryptoAuthentication returns to the system depend on its current state as follows:

Table 6. Return Codes

State Description	Error/Status	Description
After wake, but prior to first command	0x11	Indication that a proper wake token has been received by CryptoAuthentication.
After successful command execution	–	Return bytes per "Output Parameters" in Command section of this document. In some cases this is a single byte with a value of 0x00 indicating success. The transmit flag can be resent to CryptoAuthentication repeatedly if a re-read of the output is necessary.
Execution error	0x0F	Command was properly received but could not be executed by CryptoAuthentication. Changes in CryptoAuthentication state or the value of the command bits must happen before it is re-attempted.
After CRC or other communications error	0xFF	Command was NOT properly received by CryptoAuthentication and should be re-issued by the system. No attempt was made to execute the command.

CryptoAuthentication always transmits complete blocks to the system, so in the above table the status/error bytes result in 4 bytes going to the system – count, status/error, CRC x 2.

After receipt of a command block, CryptoAuthentication will parse the command for errors, a process which takes t_{PARSE} (Refer to Section 3.1.1). After this interval the system can send a transmit token to CryptoAuthentication – if there was an error then CryptoAuthentication will respond with an error code. If there is no error then CryptoAuthentication internally transitions automatically from t_{PARSE} to t_{EXEC} and will not respond to any transmit tokens until both delays are complete.

3.1.3. Sleep Flag

The sleep flag is used to transition CryptoAuthentication to the low power state, which causes a complete reset of CryptoAuthentication's internal command engine and input/output buffer. It can be sent to CryptoAuthentication at any time when CryptoAuthentication will accept a flag.

To achieve the specified I_{SLEEP}, Atmel recommends that the input signal be brought below V_{IL} when the chip is asleep. To achieve I_{SLEEP} if the sleep state of the input pin is high, the voltage on the input signal should be within 0.5V of V_{CC} to avoid additional leakage on the input circuit of the chip.

The system must calculate the total time required for all commands to be sent to CryptoAuthentication during a single session, including any inter-bit/byte delays. If this total time exceeds $t_{WATCHDOG}$ then the system must issue a partial set of commands, then a Sleep flag, then a Wake token, and finally after the wake delay the remaining commands.

3.1.4. Pause State

The pause state is entered via the PauseLong command and can be exited only when the watchdog timer has expired and the chip transitions to a sleep state. When in the pause state, the chip ignores all transitions on the signal pin but does not enter a low power consumption mode.

The pause state provides a mechanism for multiple AT88SA102S chips on the same wire to be selected and to exchange data with the host microprocessor. The PauseLong command includes an optional address field which is compared to the values in Fuses 84-87. If the two match, then the chip enters the pause state, otherwise it continues to monitor the bus for subsequent commands. The host would selectively put all but one AT88SA102S's in the pause state before executing the MAC command on the active chip. After the end of the watchdog interval all the chips will have entered the sleep state and the selection process can be started with a wake token (which will then be honored by all chips) and selection of a subsequent chip.

3.2. IO Blocks

Commands are sent to the chip, and responses received from the chip, within a **block** that is constructed in the following way:

Byte Number	*Name*	*Meaning*
0	Count	Number of bytes to be transferred to the chip in the block, including count, packet and checksum, so this byte should always have a value of (N+1). The maximum size block is 39 and the minimum size block is 4. Values outside this range will cause unpredictable operation.
1 to (N-2)	Packet	Command, parameters and data, or response. Refer to Section 3.1.2 & Section 4 for more details.
N-1, N	Checksum	CRC-16 verification of the count and packet bytes. The CRC polynomial is 0x8005, the initial register value should be 0 and after the last bit of the count and packet have been transmitted the internal CRC register should have a value that matches that in the block. The first byte transmitted (N-1) is the least significant byte of the CRC value so the last byte of the block is the most significant byte of the CRC.

3.3. IO Flow

The general IO flow for the MAC command is as follows:

1. System sends Wake token.
2. System sends Transmit flag.
3. Receive 0x11 value from the AT88SA102S to verify proper wakeup synchronization.
4. System sends Command flag.
5. System sends complete command block.
6. System waits t_{PARSE} for the AT88SA102S to check for command formation errors.
7. System sends Transmit flag. If command format is OK, the AT88SA102S ignores this flag because the computation engine is busy. If there was an error, the AT88SA102S responds with an error code.
8. System waits t_{EXEC}, refer to Section 3.1.1.
9. System sends Transmit flag.
10. Receive output block from the AT88SA102S, system checks CRC.
11. If CRC from the AT88SA102S is incorrect, indication transmission error, system resends Transmit flag.
12. System sends sleep flag to the AT88SA102S.

All commands other than MAC have a short execution delay. In these cases, the system should omit steps 6, 7 & 8 and replace this with a wait of duration $t_{PARSE} + t_{EXEC}$.

3.4. Synchronization

Because the communications protocol is half duplex, there is the possibility that the system and CryptoAuthentication will fall out of synchronization with each other. In order to speed recovery, CryptoAuthentication implements a timeout that forces the chip to sleep.

3.4.1. IO Timeout

After a leading transition for any data token has been received, CryptoAuthentication will expect another token to be transmitted within a $t_{TIMEOUT}$ interval. If the leading edge of the next token is not received within this period of time, CryptoAuthentication assumes that the synchronization with the host is lost and transitions to a sleep state.

After CryptoAuthentication receives the last bit of a command block, this timeout circuitry is disabled. If the command is properly formatted, then it is re-enabled with the first transmit token that occurs after t_{PARSE} + t_{EXEC}. If there is an error in the command, then it is re-enabled with the first transmit token that occurs after t_{PARSE}.

In order to limit the active current if CryptoAuthentication is inadvertently awakened, the IO timeout is also enabled when CryptoAuthentication wakes up. If the first token does not come within the $t_{TIMEOUT}$ interval, then CryptoAuthentication will go back to sleep without performing any operations.

3.4.2. Synchronization Procedures

When the system and CryptoAuthentication fall out of synchronization, the system will ultimately end up sending a transmit flag which will not generate a response from CryptoAuthentication. The system should implement its own timeout which waits for $t_{TIMEOUT}$ during which time CryptoAuthentication should go to sleep automatically. At this point, the system should send a Wake token and after t_{WLO} + t_{WHI}, a Transmit token. The 0x11 status indicates that the resynchronization was successful.

It may be possible that the system does not get the 0x11 code from CryptoAuthentication for one of the following reasons:

1. The system did not wait a full $t_{TIMEOUT}$ delay with the IO signal idle in which case CryptoAuthentication may have interpreted the Wake token and Transmit flag as data bits. Recommended resolution is to wait twice the $t_{TIMEOUT}$ delay and re-issue the Wake token.
2. CryptoAuthentication went into the sleep mode for some reason while the system was transmitting data. In this case, CryptoAuthentication will interpret the next data bit as a wake token, but ignore some of the subsequently transmitted bits during its wake-up delay. If any bytes are transmitted after the wake-up delay, they may be interpreted as a legal flag, though the following bytes would not be interpreted as a legal command due to an incorrect count or the lack of a correct CRC. Recommended resolution is to wait the $t_{TIMEOUT}$ delay and re-issue the Wake token.
3. There is some internal error condition within CryptoAuthentication which will be automatically reset after a $t_{WATCHDOG}$ interval, see below. There is no way to externally reset CryptoAuthentication – the system should leave the IO pin idle for this interval and issue the Wake token.

3.5. Watchdog Failsafe

After the Wake token has been received by CryptoAuthentication, a watchdog counter is started within the chip. After $t_{WATCHDOG}$, the chip will enter sleep mode, regardless of whether it is in the middle of execution of a command and/or whether some IO transmission is in progress. There is no way to reset the counter other than to put the chip to sleep and wake it up again.

This is implemented as a fail-safe so that no matter what happens on either the system side or inside the various state machines of CryptoAuthentication including any IO synchronization issue, power consumption will fall to the low sleep level automatically.

3.6. Byte & Bit Ordering

CryptoAuthentication is a little-endian chip:

- All multi-byte aggregate elements within this spec are treated as arrays of bytes and are processed in the order received.
- Data is transferred to/from CryptoAuthentication least significant bit first on the bus.
- In this document, the most significant bit and/or byte appears towards the left hand side of the page.

4. Commands

The command packet is broken down in the following way:

Byte	***Name***	***Meaning***
0	Opcode	The Command code
1	Param1	The first parameter – always present
2-3	Param2	The second parameter – always present
4 +	Data	Optional remaining input data

If a command fails because the CRC within the block is incorrect or there is some other communications error then immediately after t_{PARSE} the system will be able to retrieve an error response block containing a single byte packet. The value of that byte will be all 1's. In this situation, the system should re-transmit the command block including the proceeding Transmit flag – providing there is sufficient time before the expiration of the watchdog timeout.

If the opcode is invalid, one of the parameters is illegal, or CryptoAuthentication is in an illegal state for the execution of this command then immediately after t_{PARSE} the system will be able to retrieve an error response block containing a single byte packet. The value of that byte will be 0x0F. In this situation, the condition must be corrected before the (modified) command is sent back to CryptoAuthentication.

If a command is received successfully then after the appropriate execution delay the system will be able to retrieve the output block as described in the individual command descriptions below.

In the individual command description tables below, the Size column describes the number of bytes in the parameter documented in each particular row. The total size of the block for each of the commands is fixed, though that value is different for each command. If the block size for a particular command is incorrect, the chip will not attempt the command execution and return an error.

4.1. MAC

Computes a SHA-256 digest of a key stored inside the chip, an input challenge and other information on the chip. The output of this command is the digest of this message.

If the message includes the serial number of the chip, then the response is said to be diversified. Protocols that utilize diversified responses may be more secure because two CryptoAuthentication chips with same key will return different responses to an identical challenge based on their unique serial number.

Table 7. Input Parameters

	Name	Size	Notes
Opcode	MAC	1	0x08
Param1	Mode	1	Controls which fields within the chip are used in the message.
Param2	KeyID	2	Which internal key is to be used in the message.
Data	Challenge	32	Input portion of message to be digested.

Table 8. Output Parameters

Name	Size	Notes
Response	32	SHA-256 digest.

Regardless of the value of <mode> the first 512 bit block of the message that will be hashed with the SHA-256 algorithm will consist of:

256 bits	key[KeyID]
256 bits	challenge

The second block consists of the following information:

8 bits	Opcode (always 0x08)
8 bits	Mode
16 bits	KeyID
64 bits	Secret Fuses including BurnFuse and BurnSecure enable (or 0’s, see below)
24 bits	Status Fuses including FuseDisable (or 0’s, see below)
8 bits	Fuse MfrID fuses, (Fuse[88:95]) (never zero’d out)
32 bits	Fuse SN, (Fuse[96:127]) (or 0’s, see below)
16 bits	ROM MfrID (never zero’d out)
16 bits	ROM SN (or 0’s, see below)

1 bit	‘1’ pad
255 bit	‘0’ pad
64 bit	total length of message in bits (512+192=704), excluding pad and length

Mode is encoded as follows:

Table 9. Mode Encoding

Bits	***Meaning***
7	Should be 0
6	If set and Fuse[87] is burned, include the 48 bit serial number (combination of fuses and ROM values) in the message. Otherwise, the corresponding message bits are set to 0.
5	If set and Fuse[87] is burned, include the 64 secret fuses (Fuse[0] through Fuse[63]) in the message. Otherwise, the corresponding message bits are set to 0. If Mode[4] is set, then the value of this mode bit is ignored.
4	If set and Fuse[87] is burned, include the 64 secret fuses and 24 status fuses (Fuse[0] through Fuse[87]) in the message. Otherwise, the corresponding message bits are set to 0.
3-0	Should be 0

If Fuse[87] is unburned, then the secret and status fuses are NOT included in the message and they are replaced with 0's.

4.2. Read

Reads 4 bytes from Fuse or ROM. Returns an error if an attempt is made to read any fuse address that is illegal.

Table 10. Input Parameters

	Name	Size	Notes
Opcode	READ	1	0x02
Param1	Mode	1	Fuse or ROM
Param2	Address	2	Which 4 bytes within array. Bits 2-15 should be 0.
Data	Ignored	0	

Table 11. Output Parameters

Name	Size	Notes
Contents	4	The contents of the specified memory location.

Table 12. Mode Encoding

Name	Value	Notes
ROM	0x00	Reads four bytes from the ROM. Bit 1 of the address parameter must be 0.
Fuse	0x01	Reads the value of 32 fuses. Bit 1 of the address parameter must be 1. The input address parameter << 5 provides the fuse number corresponding to the LSB of the first returned byte.

4.3. BurnFuse

Burns a single one of the 24 status fuse bits (Fuse[64] – Fuse[87]). No other fuses can be burned with this command – use BurnSecure at personalization time to burn any of the first 88 fuses.

If the BurnFuse Enable bit (Fuse 1) has been burned to a 0, then attempts to run this command will return an error.

The power supply pin must meet the V_{BURN} specification during the entire BurnFuse command in order to burn fuses reliably. If Vcc is greater than 4.5V, then the BurnTime parameter should be set to 0x00 and the internal burn time will be up to 250µs. If Vcc is less than 4.5V but greater than V_{BURN} then the BurnTime parameter should be set to 0x8000 and the internal burn time will be up to 190ms. The chip does NOT internally check the supply voltage level.

There is a very small interval during t_{EXEC_BURN} when the fuse element is actually being burned. The power supply must not be removed during this interval and the watchdog timer must not be allowed to expire during this interval, or the fuse may end up in a state where it reads as un-burned but cannot be burned.

Table 13. Input Parameters

	Name	Size	Notes
Opcode	BURNFUSE	1	0x04
Param1	FuseNum	1	Which bit within fuse array, minimum value is 64, and maximum value is 86.
Param2	BurnTime	2	Must be 0x00 00 if Vcc > 4.5V, must be 0x80 00 otherwise.
Data	Ignored	0	

Table 14. Output Parameters

Name	Size	Notes
Success	1	Upon successful execution, a value of 0 will be returned by the AT88SA102S.

4.4. GenPersonalizationKey

Loads a personalization key into internal memory and then uses that key along with an input seed to generate a decryption digest using SHA-256. Neither the key nor the decryption digest can be read from the chip. Upon completion, an internal bit is set indicating that a secure personalization digest has been loaded and is ready for use by BurnSecure. This bit is cleared (and the digest lost) when the watchdog timer expires or the power is cycled.

This command will fail if Fuse[87] has been burned.

Table 15. Input Parameters

	Name	Size	Notes
Opcode	GenPers	1	0x20
Param1	Zero	1	Must be 0x00
Param2	KeyID	2	Identification number of the personalization key to be loaded.
Data	Seed	16	Seed for digest generation. The least significant bit of the last byte is ignored by the AT88SA102S.

Table 16. Output Parameters

Name	Size	Notes
Success	1	Upon successful execution of HOST0, a value of 0 will be returned by the AT88SA102S.

The SHA-256 message body used to create the resulting digest internally stored in the chip consists of the following 512 bits:

256 bits	PersonalizeKey[KeyID]
64 bits	Fixed value of all 1's
127 bits	Seed from input stream
1 bits	'1' pad
64 bits	Length of message in bits, fixed at 512

4.5. BurnSecure

Burns any combination of the first 88 fuse bits. Verification that the proper secret fuse bits have been burned must occur using the MAC command – there is no way to read the values in the first 64 fuses to verify their state. The 24 status fuses can be verified with the Read command.

The fuses to be burned are specified by the 88 bit input map parameter. If a bit in the map is set to a ‘1’, then the corresponding fuse is burned. If a bit in the map parameter is 0, then the corresponding fuse is left in its current state. The first bit sent to the AT88SA102S corresponds to Fuse[0] and so on up to Fuse[87]. Note that since a ‘1’ bit in the Map parameter results in a ‘0’ data value in the actual fuse array, the value in the Map parameter should generally be the inverse of the desired secret or status value. See Section 1.3 for more details.

To facilitate secure personalization of the AT88SA102S, this map may be encrypted before being sent to the chip. If this mode is desired, then the Decrypt parameter should be set to 1 in the input parameter list. The decryption (transport) key is computed by the GenPersonalizationKey command, which must have been run immediately prior to the execution of BurnSecure. In this case, prior to burning any fuses, the input Map parameter is XOR’d with the first 88 bits of that digest from the GenPersonalizationKey command. The GenPersonalizationKey and BurnSecure commands must be run within a single wake cycle prior to the expiration of the watchdog timer.

The power supply pin must meet the V_{BURN} specification during the entire BurnSecure command in order to burn fuses reliably. If Vcc is greater than 4.5V, then the BurnTime parameter should be set to 0x00 and the internal burn time will be 250µs. If Vcc is less than 4.5V but greater than V_{BURN} then the BurnTime parameter should be set to 0x8000 and the internal burn time will be 190ms per fuse bit burned. The chip does NOT internally check the supply voltage level.

The total BurnSecure execution delay is directly proportional to the total number of fuses being burned. If Vcc is less than 4.5V, then the total BurnSecure execution time may exceed the interval remaining before the expiration of the watchdog timer. In this case, the BurnSecure command should be run repeatedly, with each repetition burning only as many fuses as there is time available. The system software is responsible for counting the number of ‘1’ bits in the clear-text version of the map parameter sent to the chip – no error is returned if the fuse burn count is too high. Other than Fuse[87] (see below), the fuses may be burned in any order.

Prior to execution of BurnSecure, the AT88SA102S verifies that Fuse[87] is un-burned. If it has been burned, then the BurnSecure command will return an error. Fuse[87] can either be burned during the last repetition of BurnSecure or it can be individually burned with BurnFuse.

There are a series of very small intervals during t_{EXEC_SECURE} when the fuse element is actually being burned. The power supply must not be removed during this interval and the watchdog timer must not be allowed to expire during this interval, or the fuse may end up in a state where it reads as un-burned but cannot be burned.

Table 17. Input Parameters

	Name	Size	Notes
Opcode	BURNSECURE	1	0x10
Param1	Decrypt	1	If 1, decrypt Map data before usage. If 0, the map is transmitted in plain text.
Param2	BurnTime	2	Must be 0x00 00 if Vcc > 4.5V, must be 0x80 00 otherwise.
Data	Map	11	Which fuses to burn, may be encrypted.

Table 18. Output Parameters

Name	Size	Notes
Success	1	Upon successful execution, a value of 0 will be returned by the AT88SA102S.

4.6. PauseLong

Forces the chip into a busy mode until the watchdog timer expires, after which it will automatically enter into the pause state. During execution of this command and while in the pause state the chip will ignore all activity on the IO signal. This command is used to prevent bus conflicts in a system that also includes other AT88SA102S chips or a CryptoAuthentication host chip sharing the same signal wire.

Table 19. Input Parameters

	Name	Size	Notes
Opcode	PAUSELONG	1	0x01
Param1	Selector	1	Which chip to put into the pause state, 0x00 for all AT88SA102S chips
Param2	Zero	2	Must be 0x00 00
Data	Ignored	0	

Table 20. Output Parameters

Name	Size	Notes
Success	1	If the command indicates that some other chip should go into the pause state, a value of 0x0F will be returned by the AT88SA102S. If this chip goes into the pause state no value will be returned.

The Selector parameter provides a mechanism to select which device will pause if there are multiple devices on the bus:

If the Selector parameter is 0x00, then every AT88SA102S chip receiving this command will go into the pause state and no chip will return a success code.

If any of the bits of the Selector parameter are set, then the chip will read the values of Fuse[84-87] and go into the pause state only if those fuse values match the least significant 4 bits of the Selector parameter. If the chip does NOT go into the pause state, it returns an error code of 0x0F. Otherwise it goes into the pause state and never returns any code.

5. Pinout

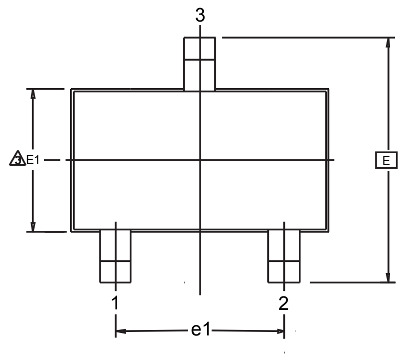
There are three pins on the chip.

Table 21. Chip Pins

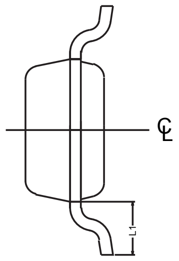
Pin #	Name	Description
1	Signal	IO channel to the system, open drain output. It is expected that an external pull-up resistor will be provided to pull this signal up to V_{CC} for proper communications. When the chip is not in use this pin can be pulled to either V_{CC} or V_{SS}.
2	V_{CC}	Power supply, 2.5 – 5.5V. This pin should be bypassed with a high quality 0.1µF capacitor close to this pin with a short trace to V_{SS}. Additional applications information at www.atmel.com.
3	V_{SS}	Connect to system ground.

6. Package Drawing

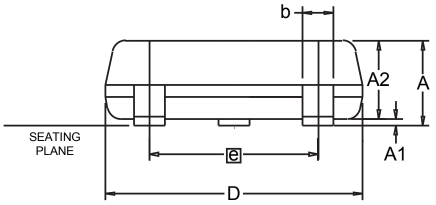
3TS1 - Shrink SOT



Top View



End View



Side View

Notes:
1. Dimension D does not include mold flash, protrusions or gate burrs. Mold flash, protrusions or gate burrs shall not exceed 0.25 mm per end. Dimension E1 does not include interlead flash or protrusion. Interlead flash or protrusion shall not exceed 0.25 mm per side.
2. The package top may be smaller than the package bottom. Dimensions D and E1 are determined at the outermost extremes of the plastic body exclusive of mold flash, tie bar burrs, gate burrs and interlead flash, but including any mismatch between the top and bottom of the plastic body.
3. These dimensions apply to the flat section of the lead between 0.08 mm and 0.15 mm from the lead tip.

This drawing is for general information only. Refer to JEDEC Drawing TO-236, Variation AB for additional information.

COMMON DIMENSIONS
(Unit of Measure = mm)

SYMBOL	MIN	NOM	MAX	NOTE
A	0.89	-	1.12	
A1	0.01	-	0.10	
A2	0.88	-	1.02	
D	2.80	2.90	3.04	1,2
E	2.10	-	2.64	
E1	1.20	1.30	1.40	1,2
L1	0.54 REF			
e1	1.90 BSC			
b	0.30	-	0.50	3

11/5/08

Package Drawing Contact: packagedrawings@atmel.com	TITLE **3TS1, 3-lead, 1.30 mm Body, Plastic Thin** Shrink Small Outline Package (Shrink SOT)	GPC TBG	DRAWING NO. 3TS1	REV. A

7. Revision History

Table 22. Revision History

Doc. Rev.	Date	Comments
8584A	03/2009	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
Unit 1-5 & 16, 19/F
BEA Tower, Millennium City 5
418 Kwun Tong Road
Kwun Tong, Kowloon
Hong Kong
Tel: (852) 2245-6100
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
securemem@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.