Keywords: SHA-1, secure hash algorithm, secure memory, 1-Wire, authentication,

Oct 24, 2005

APPLICATION NOTE 3675

Protecting the R&D Investment—Two-Way Authentication and Secure Soft-Feature Settings

*Abstract: In the age of identity theft and faked IDs, achieving positive identification is of paramount importance. This is not only true for individuals, but also applies to electronic products. System vendors need to protect their products against hacker attacks from the "outside" and ensure that the security is not compromised on the "inside" through cloned hardware. The key to realizing these diverging requirements is **authentication**.*

This article explains the concept of authentication and specifically a Maxim solution in the form of secure memories to provide secure control and protection from application requirements including intellectual property protection, embedded HW/SW license management, secure soft-feature and status setting, and tamper-proof data storage.

What is Authentication?

Authentication is a process with the objective to establish proof of identity between two or more entities. In the case of one-way authentication, just one party is involved proving its identity to another. With two-way authentication, both parties prove their identity to each other. The most commonly used method of authentication is the password. The main problem with passwords is that they are exposed when used, making them vulnerable to spying.

After reviewing the historical use of cryptography, in 1883 the Flemish linguist Auguste Kerckhoffs published his findings in a groundbreaking article on military cryptography. Kerckhoffs argued that instead of relying on obscurity, security should depend on the strength of keys, because in the event of a breach, only the keys would need to be replaced, not the whole system.

Key-based authentication works as shown in **Figure 1**: a (secret) key and the to-be-authenticated data ("message") are taken as input to compute a message authentication code or MAC. The MAC is then attached to the message. The recipient of the message performs the same computation and compares its version of the MAC to the one received with the message. If both MACs match, the message is authentic. A weakness with this basic model, however, is that an intercepted message can later or subsequently be replayed by a nonauthentic sender and be mistaken as authentic.

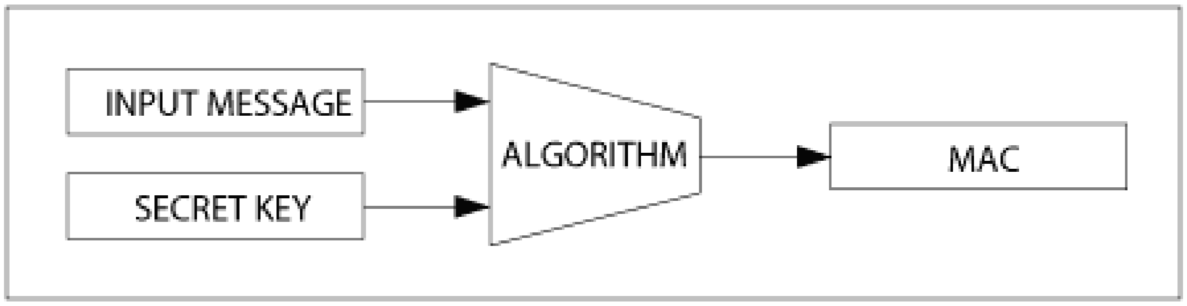


Figure 1. MAC computation model.

To prove the authenticity of the MAC originator, the recipient generates a random number and sends it as a challenge to the originator. The MAC originator must then compute a new MAC based on the secret, message, and challenge and send it back to the recipient. If the originator proves capable of generating a valid MAC for any challenge, it is very certain that it knows the secret and therefore can be considered authentic, (see **Figure 2**). The technical term for this process is challenge-and-response authentication.

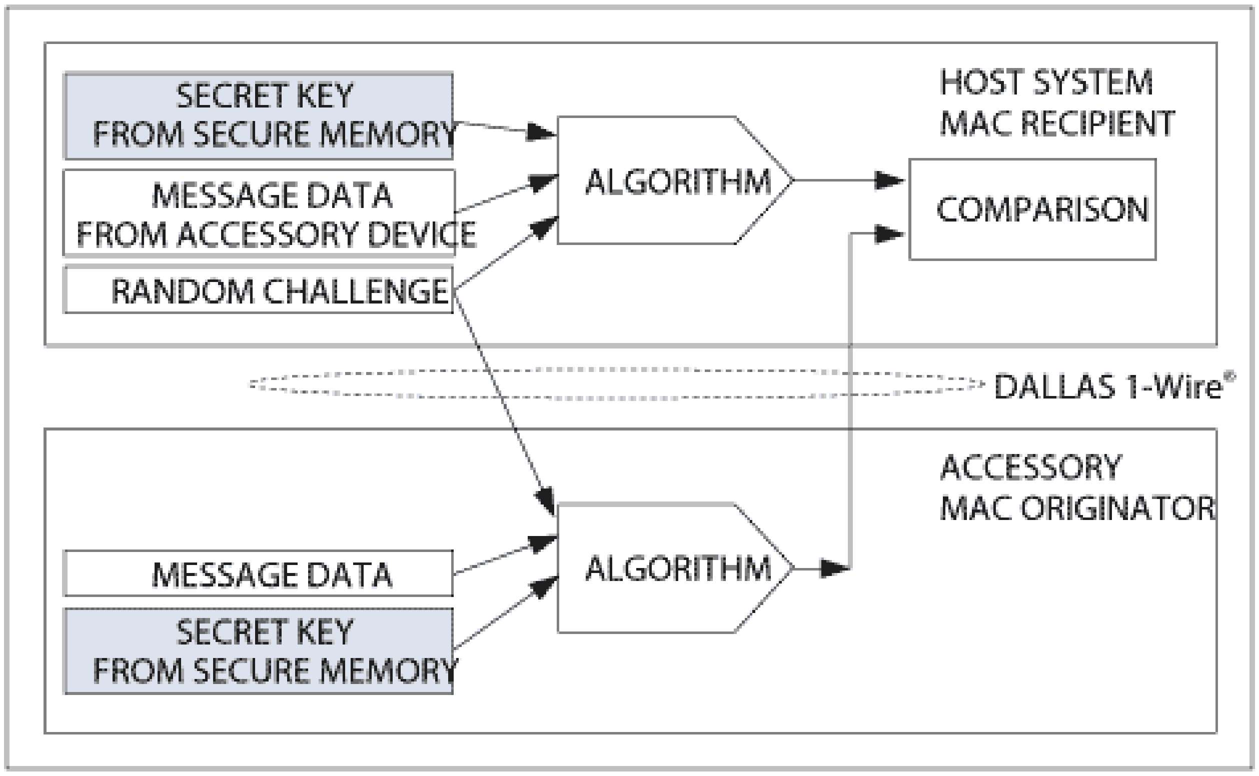

SECRET KEY FROM SECURE MEMORY
MESSAGE DATA FROM ACCESSORY DEVICE
RANDOM CHALLENGE
ALGORITHM
HOST SYSTEM MAC RECIPIENT
COMPARISON
DALLAS 1-Wire®
ACCESSORY MAC ORIGINATOR
MESSAGE DATA
ALGORITHM
SECRET KEY FROM SECURE MEMORY

Figure 2. Challenge and response authentication data flow.

In cryptography, an algorithm that generates a (fixed-length) message authentication code from a message is called one-way hash function. "One-way" indicates that it is extremely difficult to conclude from the fixed length MAC output the usually larger message. With encryption, in contrast, the size of the encrypted message is proportional to the original message.

A thoroughly scrutinized and internationally certified one-way hash algorithm is SHA-1, which was developed by the National Institute of Standards and Technology (NIST). SHA-1 has evolved into the international standard ISO/IEC 10118-3:2004. The math behind the algorithm is publicly available through the NIST website. Distinctive characteristics of the SHA-1 algorithm are: 1) irreversibility—it is computationally infeasible to determine the input corresponding to a MAC; 2) collision-resistance—it is impractical to find more than one input message that produces a given MAC; and 3) high avalanche effect—any change in input produces a significant change in the MAC result. For these reasons, as well as the international scrutiny of the algorithm, Maxim selected SHA-1 for challenge-and-response authentication of its secure memories.

Low-Cost Secure Authentication—A Functional Implementation

Thanks to its 1-Wire interface, the DS2432 can easily be added to any circuit containing some digital processing capabilities, such as a microcontroller (µC). In the simplest case, all that is needed is one free I/O pin and a pullup resistor for the 1-Wire line, as shown in **Figure 3**. If the computing capabilities on the board or the remaining program storage space are insufficient to compute a MAC, one can use a DS2460 SHA-1 coprocessor or leave this task to the nearest host in the system or network. The coprocessor has the additional advantage of securely storing the system secret in secure memory rather than in the host process program code.

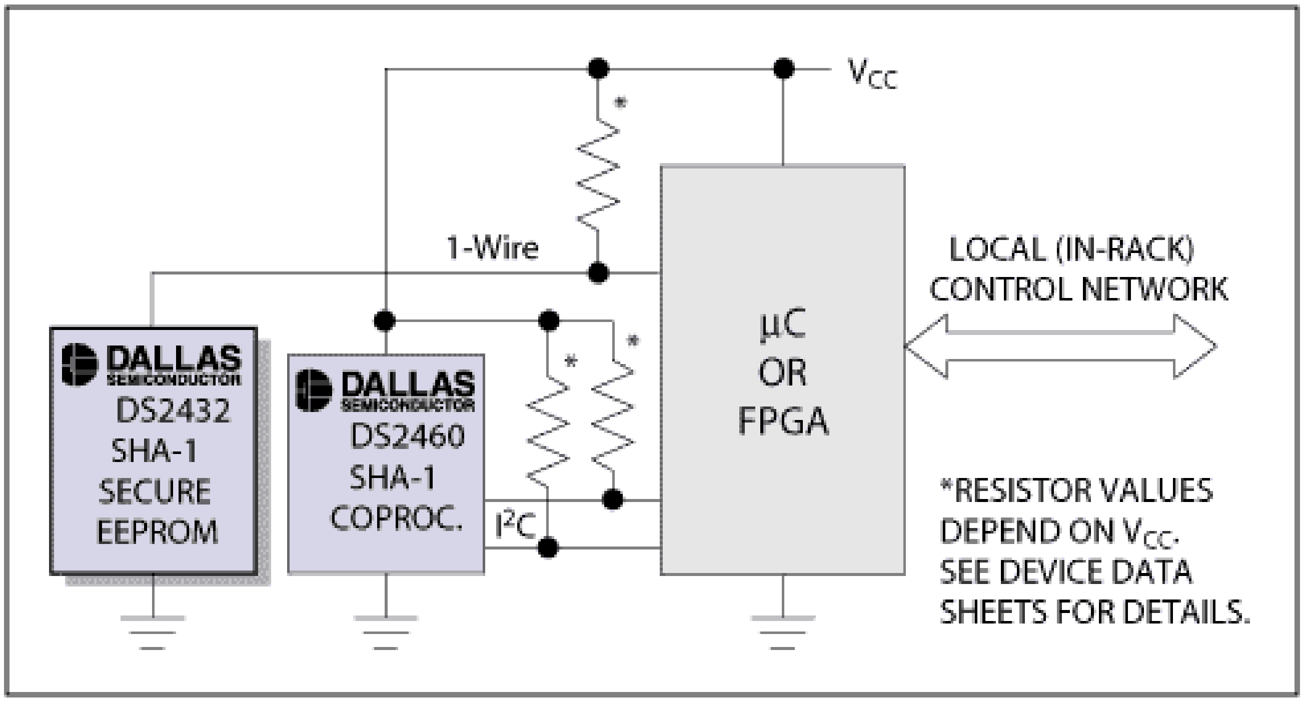

V$_{CC}$
1-Wire
LOCAL (IN-RACK) CONTROL NETWORK
DALLAS SEMICONDUCTOR
DS2432 SHA-1 SECURE EEPROM
DALLAS SEMICONDUCTOR
DS2460 SHA-1 COPROC.
I^{2}C
µC OR FPGA
*RESISTOR VALUES DEPEND ON V$_{CC}$. SEE DEVICE DATA SHEETS FOR DETAILS.

Figure 3. Typical system environment.

Embedded HW/SW License Management

Reference designs, which are subsequently licensed and possibly manufactured by third parties, require barriers to prevent illegal use of the intellectual property. For revenue reasons, it is also necessary to track and confirm the number of reference uses. A preprogrammed DS2432 (secret and memory settings installed prior to delivery to the third party manufacturer) easily solves these requirements and more. As a power-up self-check, the reference (**Figure 4**) performs an authentication sequence with the DS2432. Only a DS2432 with valid secret, known only by the licensing company and reference electronics, will be successful at reply with a valid MAC. The reference processor would take appropriate, application-specific action when an invalid MAC is detected. The additional benefit of this approach is the ability to selectively license and enable reference features through settings in secure DS2432 memory (for more on this concept, see section **Soft-Feature Management**).

The DS2432 with a 64-bit valid secret is supplied to the licensee or third-party manufacturer through one of two secure methods: a) preprogrammed by the company licensing the reference, or b) preprogrammed by Maxim per the licensing company's input and then delivered to the third-party manufacturer. In either case, the number of devices sent to the licensee or manufacturer is known and can be used to validate license fees.

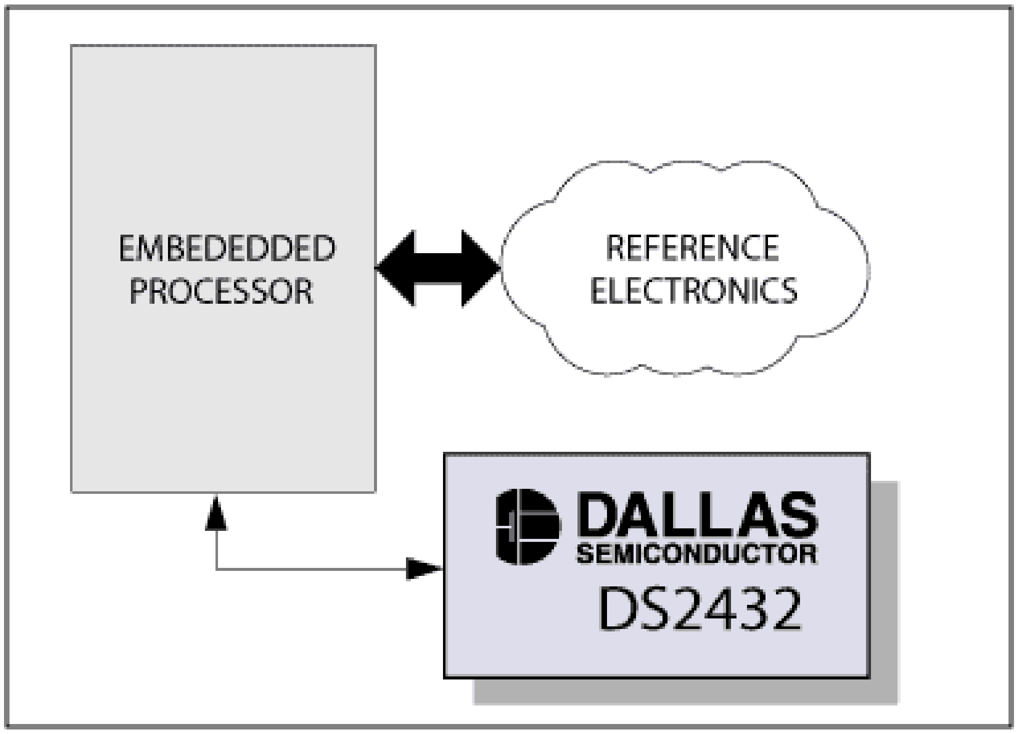


Figure 4. Authenticating the reference design.

Verification of Hardware Authenticity
When verifying hardware authenticity, there are two cases to be considered (**Figure 5**): 1) a cloned circuit board with an exact copy of the firmware/FPGA configuration, and 2) a cloned system host.

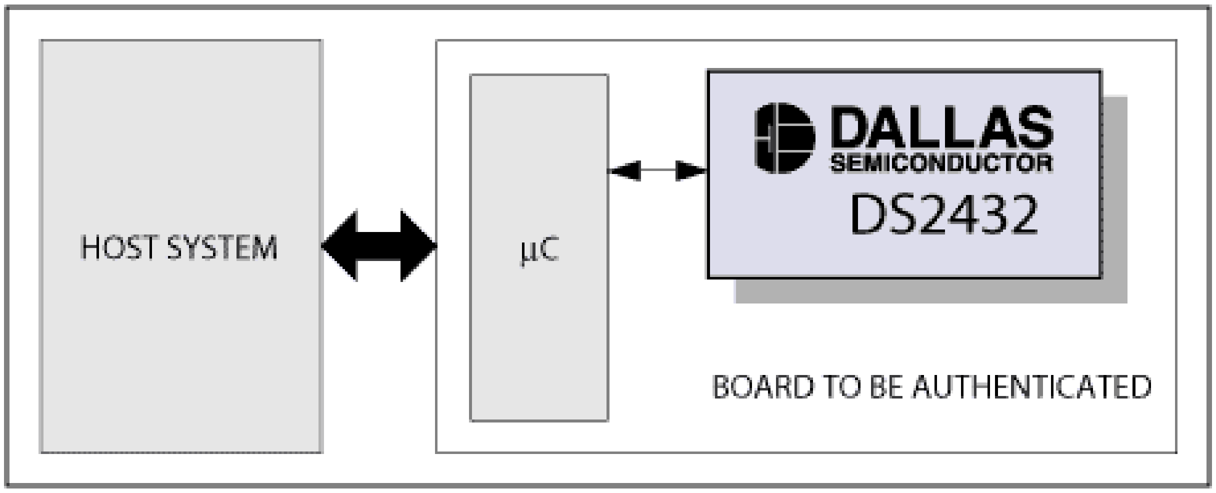


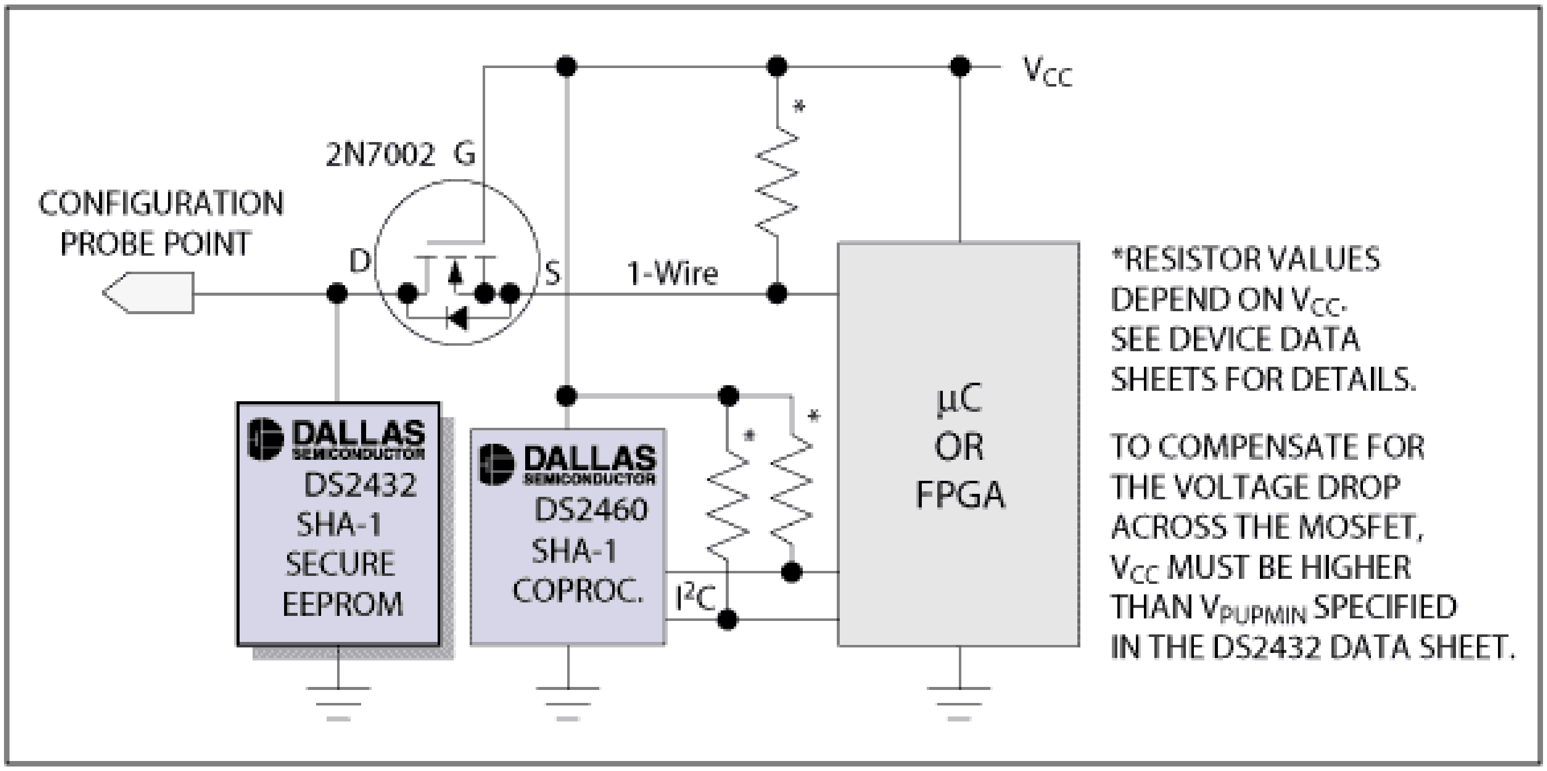
Figure 5. HW authentication example.

In the first case, the firmware/FPGA attempts to authenticate the cloned circuit board. The clone manufacturer must load a secret into the DS2432 in order to write data into the user EEPROM. While this may make the data look correct, the secret is not valid in the system. Due to the complexity in changing the firmware/FPG and to remain compatible with the host, the firmware/configuration has to be an exact copy of the original. If the board performs a challenge-and-response authentication of the DS2432 during the power-up phase, the MAC generated by the DS2432 will be different from the MAC computed by the microcontroller/FPGA. This MAC mismatch is strong evidence that the board is not authentic. This would be detected by the system performing a challenge/response sequence with the board, and application-specific action would then be taken.

In the second case, the circuit board attempts to authenticate the host system. The board can verify the authenticity of the host using the following procedure: 1) generate a challenge and let the DS2432 compute a challenge-and-response authentication MAC; and 2) send the same MAC computation input data (except for the secret, of course) to the network host, which then computes and returns a challenge-and-response authentication MAC from that data and its own secret. If both MACs match, the board can assume that the host is authentic. A match, however, can also be achieved if both the DS2432 and the network host are compromised.

Soft-Feature Management
Electronic systems range from handheld products to units that fill several racks. The larger the unit's size, the more costly it is to develop. To keep the cost under control, there is a desire to construct a large system from a limited selection of smaller subsystems (boards). Often, not all features of a subsystem are needed in the application. Instead of removing these features, it is more cost-effective to leave the board as is and to simply disable them in the control software. This, however, creates a new problem: a smart customer who needs several fully featured systems could just buy one fully featured unit and several units with reduced features. Then, copying the software, the simpler units behave like the fully featured unit but for a lower price, shortchanging the system vendor.

A DS2432 on the board of each subsystem can protect the system vendor from this type of fraud. In addition to serving for challenge-and-response authentication, the same DS2432 can store the individual configuration settings in its user EEPROM. As explained later in the section **Data Security**, this data is protected from unauthorized changes, giving full control to the system vendor. The configuration settings can be stored in the form of a bitmap or code words as deemed appropriate by the system designer. For practical reasons, the configuration should be as easy to set as possible. Again, thanks to the 1-Wire interface, all that is necessary is to add a single transistor and a probe point, as shown in **Figure 6**. Through the probe point, the configuration can be written to the DS2432 without powering the rest of the board. The MOSFET isolates the DS2432 from the other circuitry without impeding normal access to the DS2432 when the subsystem is operated in its normal environment.



V_{CC}
2N7002 G
CONFIGURATION PROBE POINT
D
S
1-Wire
*
DALLAS SEMICONDUCTOR
DS2432 SHA-1 SECURE EEPROM
DALLAS SEMICONDUCTOR
DS2460 SHA-1 COPROC.
I^2C
µC OR FPGA
*RESISTOR VALUES DEPEND ON V_{CC}. SEE DEVICE DATA SHEETS FOR DETAILS.
TO COMPENSATE FOR THE VOLTAGE DROP ACROSS THE MOSFET, V_{CC} MUST BE HIGHER THAN V_{PUPMIN} SPECIFIED IN THE DS2432 DATA SHEET.

Figure 6. Adding a configuration probe point.

As an added bonus, this way of setting configurations allows for remote feature upgrade/change after the system is installed at the customer's site. Any user EEPROM that is not used for configuration/feature management is available for board identification purposes in the form of an electronic nameplate. This feature is explained in detail in Application Note 178: Printed Circuit Board Identification Using 1-Wire Products on the Maxim website.

DS2432 Authentication Feature Details

General Device Architecture
The major data elements and the data-flow paths of the DS2432 1kb SHA-1 secure memory with 1-Wire interface are shown in **Figure 7**. Easily recognized are the 8-byte secret key and the buffer memory (scratchpad), which temporarily stores the challenge. Data elements not mentioned previously are the unique device ID number (a standard 1-Wire feature), four pages of user EEPROM, control registers, and system constants.

The device ID serves as a node address in 1-Wire networks, but also contributes to authentication. The user memory holds the major part of the to-be-authenticated "message". Seed constants are needed to meet formatting requirements and as padding to compose the 64-byte input data block for the SHA-1 computation. The control registers perform device-specific functions, such as optional write protection of the secret or EEPROM emulation mode; they do not contribute to the authentication process in general.

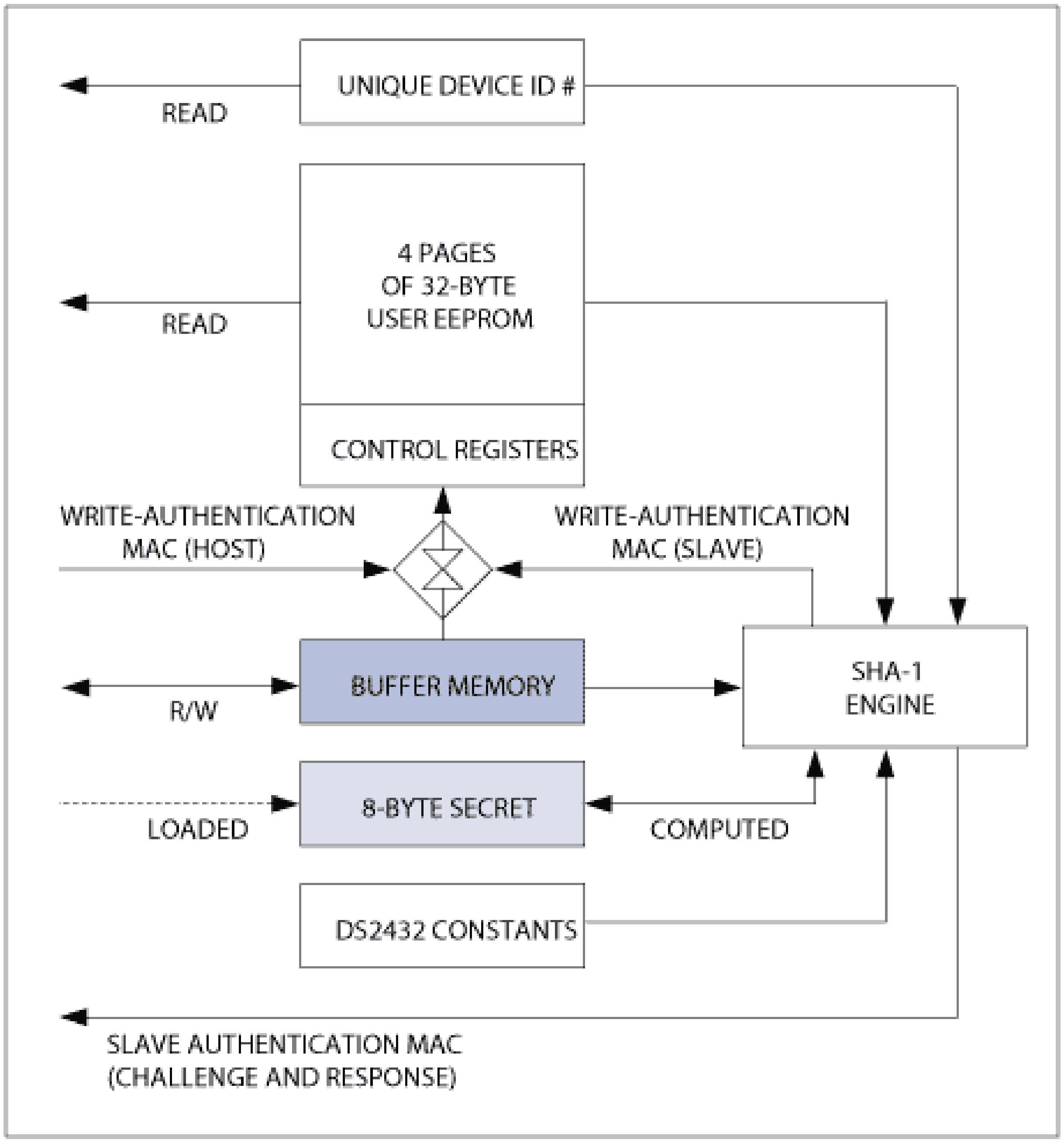

UNIQUE DEVICE ID #
READ
4 PAGES OF 32-BYTE USER EEPROM
READ
CONTROL REGISTERS
WRITE-AUTHENTICATION MAC (HOST)
WRITE-AUTHENTICATION MAC (SLAVE)
BUFFER MEMORY
R/W
SHA-1 ENGINE
8-BYTE SECRET
LOADED
COMPUTED
DS2432 CONSTANTS
SLAVE AUTHENTICATION MAC (CHALLENGE AND RESPONSE)

Figure 7. DS2432 SHA-1 Secure memory data flow model.

Device ID number and user EEPROM can be read without restriction. There is full read/write access to the buffer memory. The secret can be loaded directly, but never read. Changing the content of the user memory or the registers requires that both host and slave (i.e., the DS2432) compute matching write-authentication MACs to open the path from the buffer memory to the EEPROM.

The SHA-1 engine of the DS2432 can be operated in three different ways, depending on the purpose of the MAC result. In any case, the SHA-1 engine gets 64 bytes of input data and computes from it a 20-byte MAC result. The differences are in the input data. As a fundamental requirement of secure systems, the host must either know or be able to compute the secret of a slave device that is valid/authentic in the application.

Challenge-and-Response Authentication MAC
The primary purpose of the DS2432 is challenge-and-response authentication. The host sends a random challenge and instructs the DS2432 to compute a response MAC from the challenge, the secret, data from one of the

memory pages selected by the host, and additional data that together constitute the "message" (**Figure 8**).

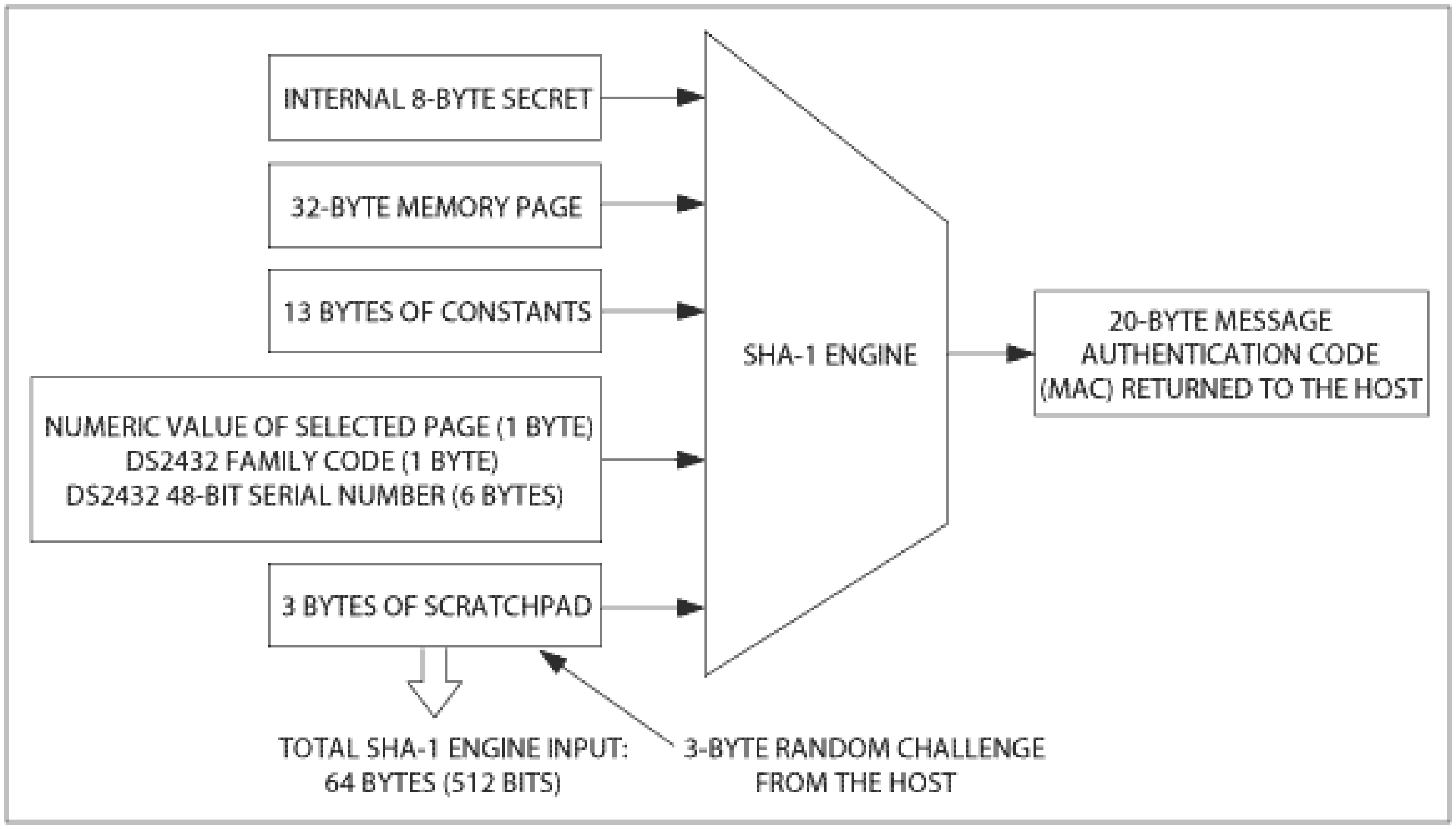

INTERNAL 8-BYTE SECRET
32-BYTE MEMORY PAGE
13 BYTES OF CONSTANTS
NUMERIC VALUE OF SELECTED PAGE (1 BYTE)
DS2432 FAMILY CODE (1 BYTE)
DS2432 48-BIT SERIAL NUMBER (6 BYTES)
3 BYTES OF SCRATCHPAD
SHA-1 ENGINE
20-BYTE MESSAGE AUTHENTICATION CODE (MAC) RETURNED TO THE HOST
TOTAL SHA-1 ENGINE INPUT: 64 BYTES (512 BITS)
3-BYTE RANDOM CHALLENGE FROM THE HOST

Figure 8. Input data for challenge-and response authentication MAC.

After it has finished computing, the DS2432 sends its MAC to the host for verification. The host then duplicates the MAC computation using a valid secret and the same message data that was used by the DS2432. A match of the MAC received from the DS2432 provides authentication of the device, as only an authentic DS2432 will respond to the challenge-and-response sequence correctly. It is crucial that the challenge is based on random data. A never-changing challenge opens the door to replay attacks using a valid static MAC that is recorded and replayed instead of a MAC that is instantly computed by an authentic DS2432.

Data Security

Beyond proving the authenticity of a slave device, it is highly desirable to know that the data stored in the device can be trusted. For this reason, write access to the DS2432 EEPROM is securely restricted. Before copying data from its scratchpad buffer memory to the EEPROM or control registers, the DS2432 requires the requesting host to supply a write-access authentication MAC to prove its authenticity. The DS2432 computes this MAC from the new data in its scratchpad buffer memory, its secret, data from the memory page to be updated, and additional data, as shown in **Figure 9**.

An authentic host knows the secret and is able to compute a valid write-access MAC. When receiving the MAC from the host during the copy command, the DS2432 compares it to its own result. Data is transferred from the buffer memory to the destination in EEPROM only if both MACs match. Of course, memory pages that are write protected cannot be modified, even if the MAC is correct.

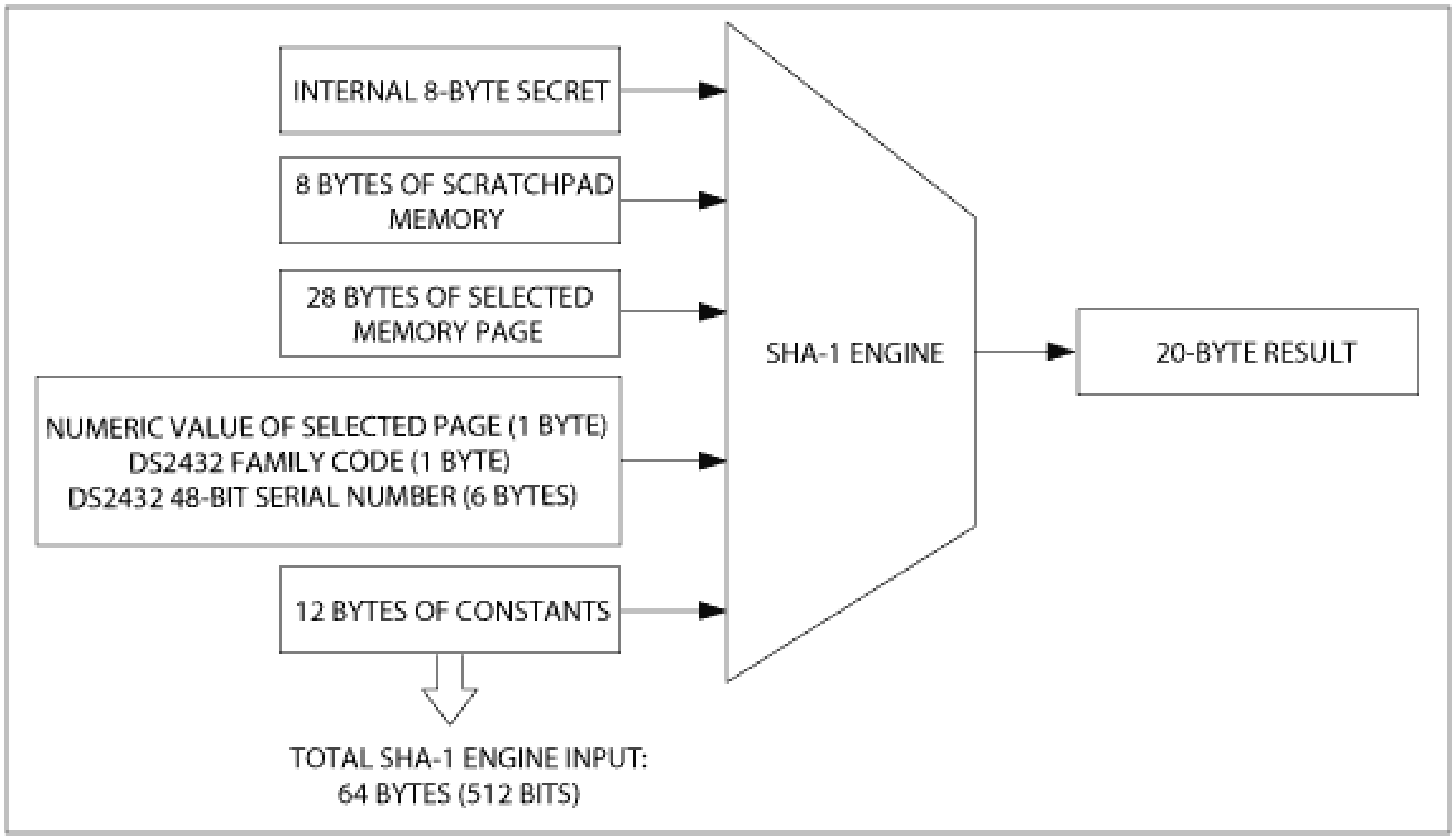

INTERNAL 8-BYTE SECRET
8 BYTES OF SCRATCHPAD MEMORY
28 BYTES OF SELECTED MEMORY PAGE
NUMERIC VALUE OF SELECTED PAGE (1 BYTE)
DS2432 FAMILY CODE (1 BYTE)
DS2432 48-BIT SERIAL NUMBER (6 BYTES)
12 BYTES OF CONSTANTS
TOTAL SHA-1 ENGINE INPUT: 64 BYTES (512 BITS)
SHA-1 ENGINE
20-BYTE RESULT

Figure 9. Input data for a write-access authentication MAC.

Secret Protection

The architecture of the DS2432 allows direct load of a secret into the device. Secret protection is provided by both read-protection and, if desired, write-protection, which prevents the secret from ever being changed. This level of protection is effective so long as access to the secret is secure and controlled at the equipment production site.

The quality of the secret can be increased in various ways: a) let the DS2432 compute its secret, b) let the DS2432 compute its secret in multiple stages performed at different sites, c) create device-specific secrets by including the unique device ID number in the computation of the secret, or d) a combination of b and c.

If each DS2432 computes its secret, only the ingredients of the secret are known but the secret itself is never exposed. If the secret is computed in multiple stages using different sites, only the local ingredients of the secret are known, which provides a method to control the knowledge of "final" secret. If the secrets are device-specific, an additional computing step is required for the host, but the potential damage is minimal if a device secret is accidentally discovered. If the secret is computed in multiple stages and made device specific, the highest possible secrecy is achieved. However, the hosts, like the slaves, need to be set-up at different sites to prevent compromise of system secrecy.

Before computing a secret, it is necessary to first load a known value as secret. With the help of this known secret, 32 bytes of the data to be included in the computation of the new secret need to be written to one of the four memory pages. Next a partial secret should be written to the DS2432 scratchpad buffer memory. The partial secret could, for example, be the number of the memory page used for the computation and the unique device ID number, omitting the CRC byte, or any other application-specific, 8-byte value.

If instructed to compute a secret, the DS2432 starts its SHA-1 engine and computes a MAC using these input data items as shown in **Figure 10**. The lower 8 bytes of the 20-byte MAC are automatically copied to the memory location of the secret and become effective immediately.

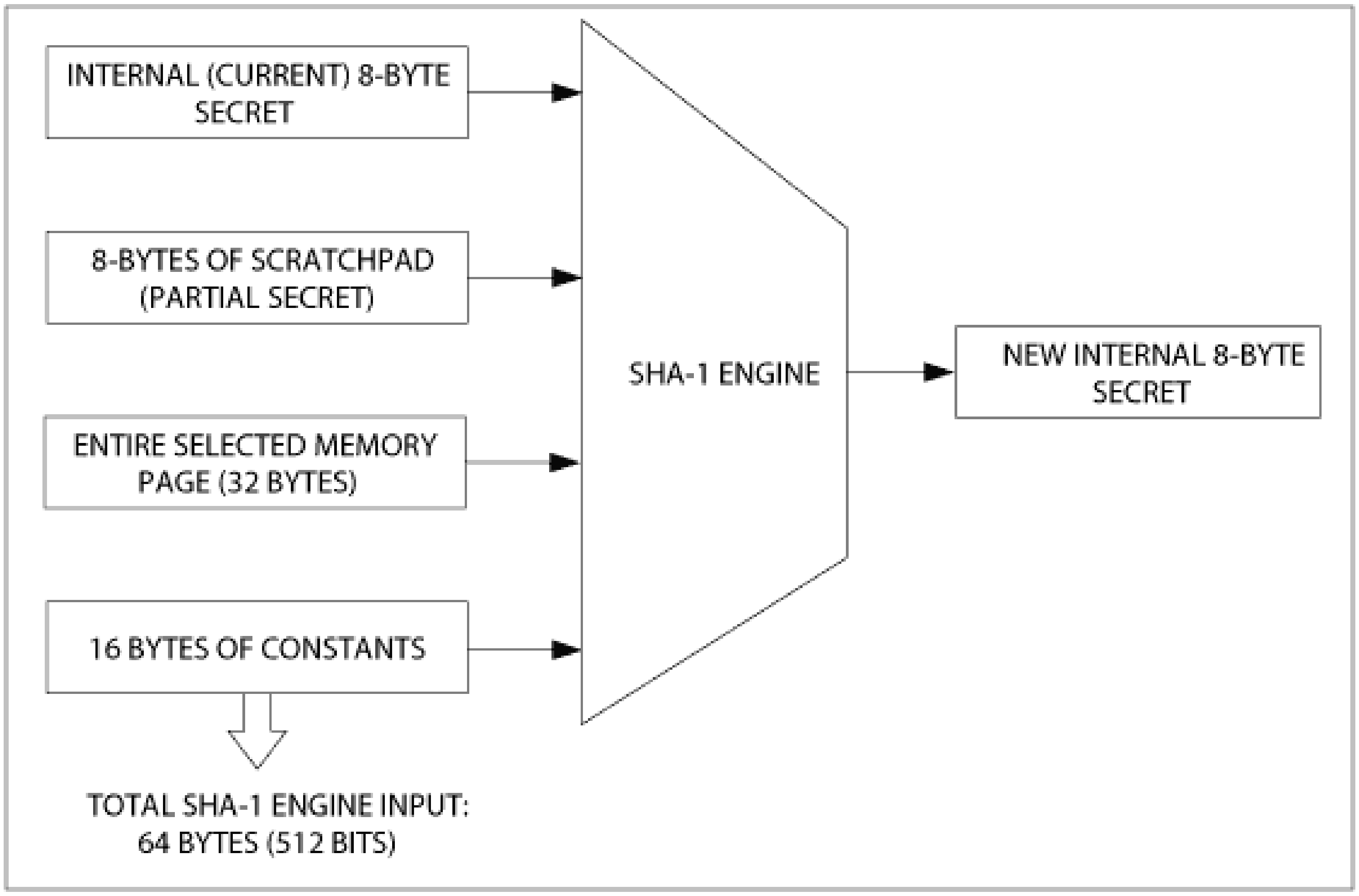

INTERNAL (CURRENT) 8-BYTE SECRET
8-BYTES OF SCRATCHPAD (PARTIAL SECRET)
ENTIRE SELECTED MEMORY PAGE (32 BYTES)
16 BYTES OF CONSTANTS
TOTAL SHA-1 ENGINE INPUT: 64 BYTES (512 BITS)
SHA-1 ENGINE
NEW INTERNAL 8-BYTE SECRET

Figure 10. Input data for secret computation.

Conclusion

Knowing secure authentication functions and implementing them wisely gives a competitive advantage. It not only protects intellectual property, but also helps reduce production cost through common HW platforms with secure, soft-feature settings. The data security of the DS2432 even allows for remote configuration changes, saving the technician valuable time. A small silicon chip can make a big difference to the bottom line.

Application Note 3675: http://www.maxim-ic.com/an3675

More Information
For technical questions and support: http://www.maxim-ic.com/support
For samples: http://www.maxim-ic.com/samples
Other questions and comments: http://www.maxim-ic.com/contact

Related Parts

DS1961S:	QuickView
DS1963S:	QuickView
DS2432:	QuickView -- Abridged Data Sheet
DS2460:	QuickView -- Abridged Data Sheet
DS28E01-100:	QuickView -- Abridged Data Sheet

AN3675, AN 3675, APP3675, Appnote3675, Appnote 3675