Keywords: secure hash, 1-wire, i2c, iButton, parasitic power, security, sha1, sha2, sha3, sha256, sha3-256, authentication, intellectual property protection, puf, physically unclonable function, chipdna, eeprom, cloning, counterfeit, medical disposables

APPLICATION NOTE 7015

BACK TO BASICS: SECURE HASH ALGORITHMS

Abstract: This application note goes over the basics of Secure Hash Algorithms (SHA) and discusses the variants of the algorithm. It then briefly touches on how the algorithm is used for authentication, including the concept of a Hashed Message Authentication Code (HMAC). It concludes by looking at some of the Maxim secure authenticators that can be used to very easily deploy SHA algorithms for security applications.

Introduction

In this application note, we will discuss the Secure Hash Algorithms (SHA) that are widely used in symmetric key cryptography. The basic idea behind a SHA function is to take data of variable size and condense it into a fixed-size bit string output. This concept is called hashing. The SHA functions are a family of hashing algorithms that have been developed over time through oversight by the National Institute of Standards and Technology (NIST). The latest of these is the SHA-3 function. Maxim has a family of secure authenticator products that provide both SHA-2 and SHA-3 functions.

Figure 1 shows the basic concept of secure hash generation.

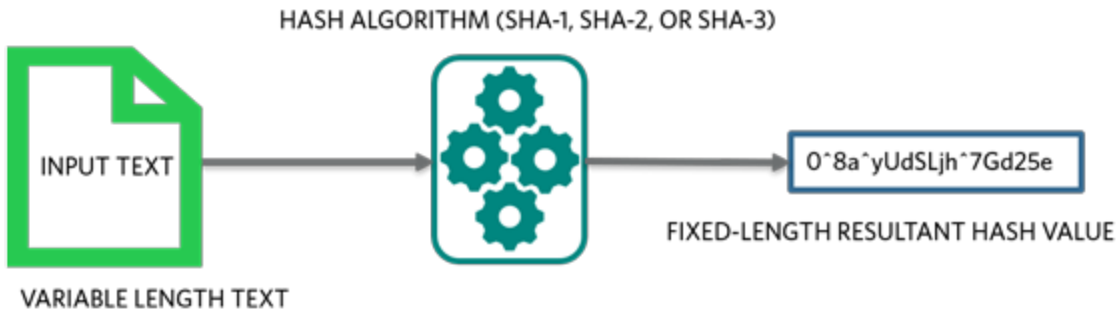

Figure 1. Secure hash generation, basic concept.

SHA Characteristics

The SHA functions have the following characteristics:

- They have variable input length and fixed output length.
- They are one-way functions. Figure 1 shows that it is infeasible to use the resultant hash value to regenerate the input text other than trying each possible input text. This becomes computationally impossible for sufficiently large inputs.

- If the same input message is fed to the SHA function, it will always generate the same resultant hash.
- It is not possible to generate the same hash value using two different input values. This is called “collision resistance.”
- A small change in the input value, even a single bit, completely changes the resultant hash value. This is called the “avalanche effect.”

If a hash function satisfies all of the above, it is considered a strong hash function.

Types of SHA Functions

Some of the SHA functions currently in use are:

- SHA-1
- SHA-2
- SHA-3

Because SHA-1 is being phased out and is not recommended for any new designs, this application note only discusses SHA-2 and SHA-3.

SHA-2

The SHA-2 function has four main types based on output bit lengths as follows:

- SHA-224 – hash is 224 bits long.
- SHA-256 – hash is 256 bits long.
- SHA-384 – hash is 384 bits long.
- SHA-512 – hash is 512 bits long.

As an example, **Figure 2** shows a block diagram of a SHA-256 engine.

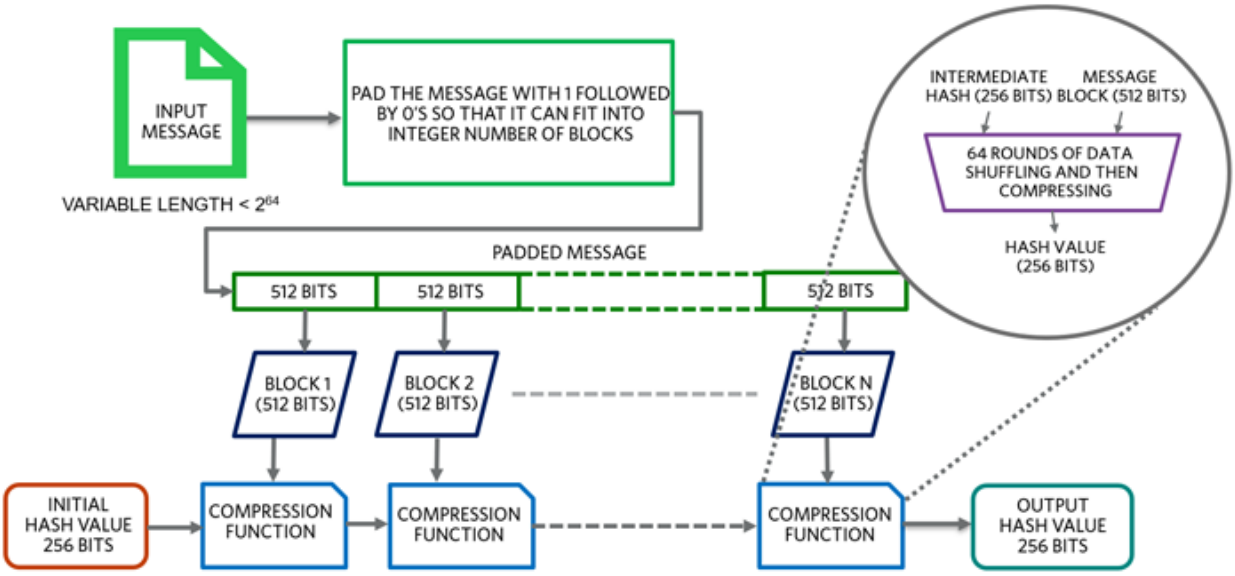

Figure 2. SHA-256 – hash generation flow.

The input message is first padded to make sure that it will completely fit in “n” number of 512-bit blocks.

The input message is first padded to make sure that it will completely fit in “n” number of 512-bit blocks.

SHA-3

The SHA-3 function has no predefined output length. The input and output lengths have no maximums either. For comparison purposes with SHA-2, we can define four main types based on output bit lengths:

- SHA3-224 – hash is 224 bits long.
- SHA3-256 – hash is 256 bits long.
- SHA3-384 – hash is 384 bits long.
- SHA3-512 – hash is 512 bits long.

All SHA-3 types use a Keccak sponge function. Just like a sponge, the first step is to soak in or absorb the input message. In the next phase, the output hash is squeezed out. **Figure 3** illustrates these phases using the block diagram of a SHA3-256 function.

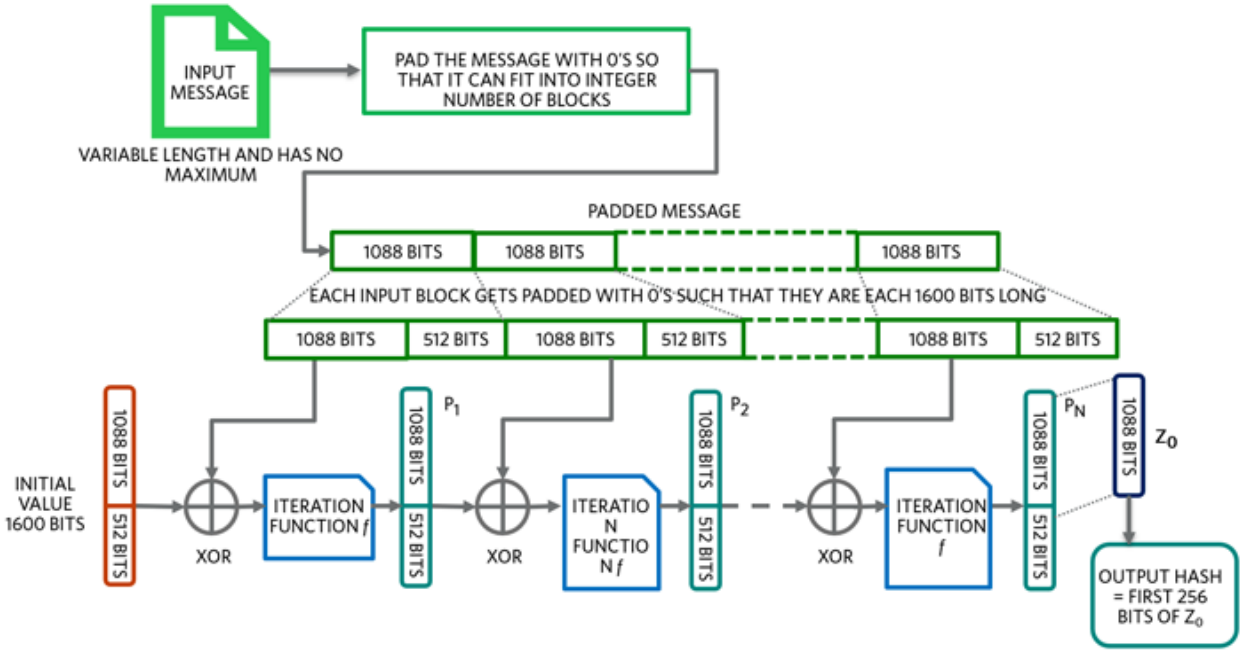

Figure 3. SHA3-256 – Keccak sponge hash generation flow.

The iteration function in the **Figure 3** diagram takes in the 1600 bits of data, puts it through 24 rounds of permutation using a specific algorithm, and then passes it to the next stage as a 1600-bit block. This continues until the absorbing phase is complete.

Once the absorbing phase is complete, the last 1600-bit block is passed to the squeezing phase. In this case, as the SHA3-256 output hash length is less than 1088 bits, the squeezing phase does not need any iteration functions. The first 256 bits from the last stage is the output hash.

If the required hash length was 2500 bits, for example, we would have needed three more instances of the iteration function to get the desired length hash.

Differences between a Secure Hash and an HMAC

Before exploring message authentication, it is important to understand the differences between a secure hash and a hashed message authentication code (HMAC), which are illustrated in **Figure 4**. Essentially, the secure hash uses a hashing algorithm such as SHA-3 to produce a fixed-length hash of the message regardless of the message length. HMAC is similar but uses a key as an additional input to the hashing engine. It also produces a fixed-length hash regardless of the input message length.

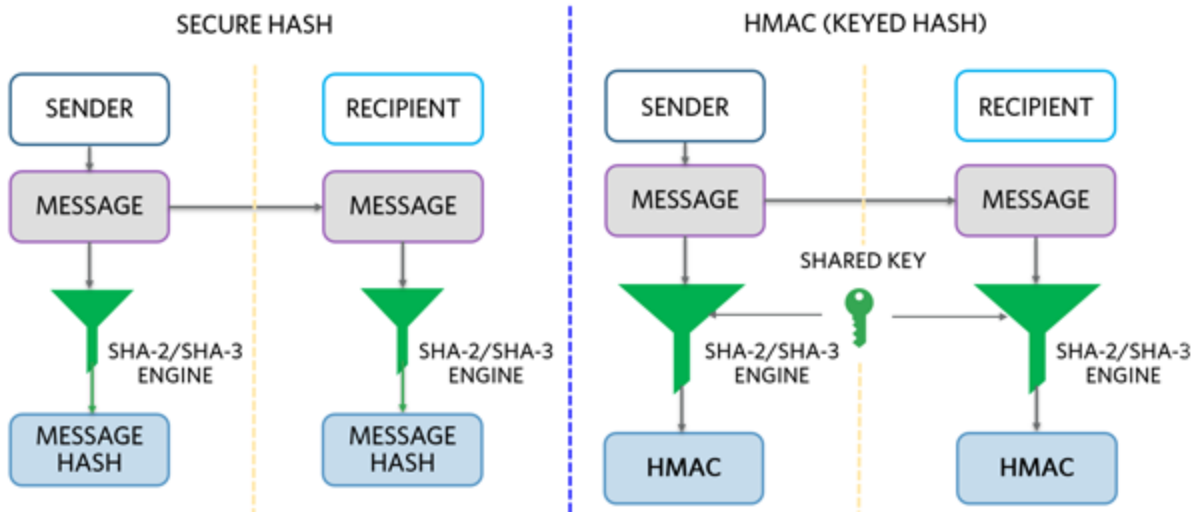

Figure 4. Secure hash and HMAC.

Message Authentication Using SHA-3

Figure 5 illustrates an example of how a message can be authenticated using SHA-3 showing all the concepts already discussed.

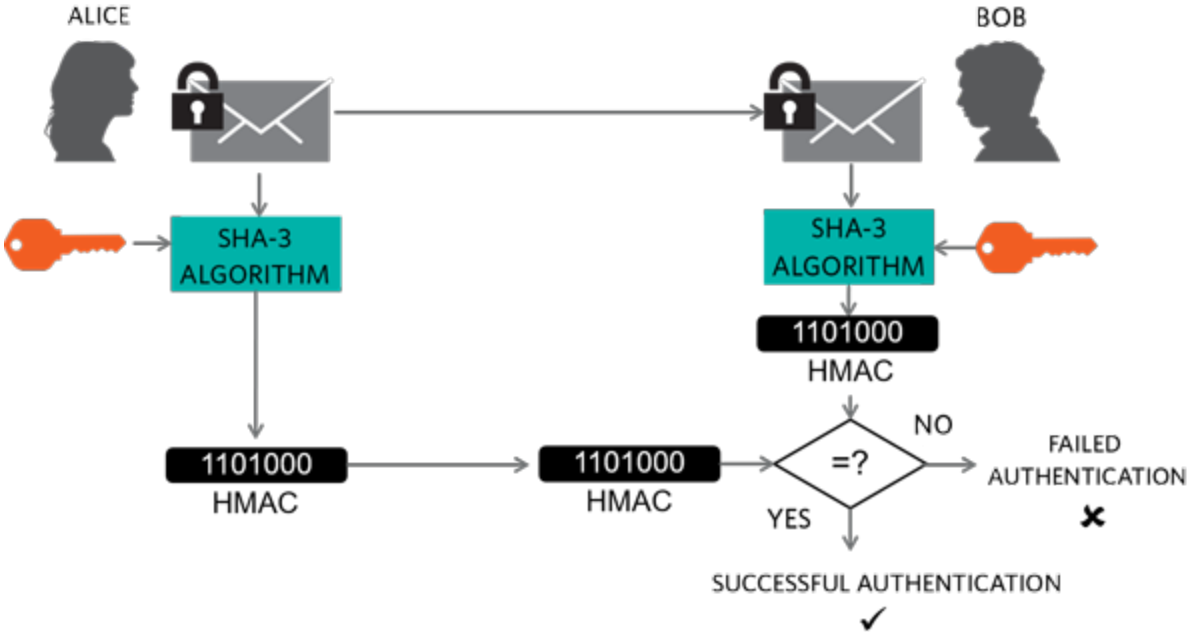

Figure 5. Message authentication using SHA-3 HMAC.

In **Figure 5**, Alice calculates the HMAC of a message by feeding it to a SHA-3 engine along with a specific key. Alice has securely shared this key previously with Bob.

Alice sends the resultant HMAC along with the message to Bob. Bob then generates his own HMAC of the message using the same key Alice shared with him earlier. Bob compares the HMAC he generated with the one he received from Alice. If they match, the message has not been tampered with and is authentic. In this scenario, someone could intercept the HMAC and the message and then alter the message and generate a new HMAC and send it to Bob. However, that will not work, because the interceptor will not have the secret

key and the received HMAC will not match the computed HMAC. Thus, Bob will realize that the message was not authentic.

The most important aspect of this sequence is for Alice and Bob to keep their shared key a secret from everyone else.

Maxim's secure authenticator products, such as the DS28E50, has built-in SHA engines and a multitude of secure features like ChipDNA™ that help secure any key, per the user's requirements. For more information, see Maxim Application Note 6767, How ChipDNA Physically Unclonable Function Technology Protects Embedded Systems.

Secure Authenticators with Built-in SHA Engines

Table 1 outlines the various secure authenticators that are available from Maxim with the other main features and target applications.

Table 1. Maxim Devices with Built-in SHA Engines for Cryptographic Applications (E: 1-Wire, C: I^2C)

Part	SHA Engine	Other Features	Typical Applications
DS28E50	SHA-3	2Kb memory, ChipDNA	• Authentication of Medical Sensors and Tools • IoT Node Authentication • Peripheral Authentication • Printer Cartridge Identification and Authentication
DS28E16	SHA-3	256 bits of secure memory, low cost	• Printer Cartridge and Battery Identification and Authentication
DS28E15 DS28EL15	SHA-256	512-bit memory, the EL device is a low-voltage (1.671V to 1.89V) device	• Medical Consumable Identification and Authentication • Secure Feature Control • Printer Cartridge Identification and Authentication
DS28E22 DS28EL22	SHA-256	2048-bit memory, the EL device is a low-voltage (1.671V to 1.89V) device	• Authentication of Network-Attached Appliances • Printer Cartridge ID/Authentication • Sensor/Accessory Authentication and Calibration • System Intellectual Property Protection

Part	SHA Engine	Other Features	Typical Applications
DS28E25 DS28EL25	SHA-256	4096-bit memory, bidirectional authentication, the EL device is a low-voltage (1.671V to 1.89V) device	• Identification and Authentication of Consumables • Sensor/Accessory Authentication and Calibration • System Intellectual Property Protection
DS1964S	SHA-256	512-bit memory, iButton	• Access Control • Authentication of Consumables
DS28E36 DS28C36	SHA-256	8Kb memory, bidirectional ECDSA or SHA-256 authentication, two GPIOs	• Accessory and Peripheral Secure Authentication • Secure Storage of Cryptographic Keys for a Host Controller
DS28E83	SHA-256	10Kb of OTP memory, high-radiation resistance	• Accessory and Peripheral Secure Authentication • Medical Consumables Secure Authentication
DS28E84	SHA-256	15Kb of FRAM, 10Kb of OTP memory, high-radiation resistance	

Summary

Using SHA to secure/authenticate a physical item or a piece of intellectual property is a very straightforward process, provided the correct supporting tools are used. Maxim's secure authenticators are ideal for these purposes. They have built in SHA engines with many features that make implementing security for any application a relatively simple process. Each device has comprehensive support systems like evaluation kits and free software, including C-based demonstration codes, to assist a developer to quickly deploy their solution.

Related Parts		
DS1964S	DeepCover Secure Authenticator iButton with SHA-256	
DS28C36	DeepCover Secure Authenticator	Samples
DS28E15	DeepCover Secure Authenticator with 1-Wire SHA-256 and 512-Bit User EEPROM	Samples
DS28E16	1-Wire SHA-3 Secure Authenticator	Samples
DS28E22	DeepCover Secure Authenticator with 1-Wire SHA-256 and 2Kb User EEPROM	Samples
DS28E25	DeepCover Secure Authenticator with 1-Wire SHA-256 and 4Kb User EEPROM	Samples

DS28E36	DeepCover Secure Authenticator	Samples
DS28E50	DeepCover Secure SHA-3 Authenticator with ChipDNA PUF Protection	Samples
DS28E83	DeepCover Radiation Resistant 1-Wire Secure Authenticator	Samples
DS28E84	DeepCover Radiation Resistant, High-Capacity 1-Wire Secure Authenticator	Samples
DS28EL15	DeepCover Secure Authenticator with 1-Wire SHA-256 and 512-Bit User EEPROM	Samples
DS28EL22	DeepCover Secure Authenticator with 1-Wire SHA-256 and 2Kb User EEPROM	Samples
DS28EL25	DeepCover Secure Authenticator with 1-Wire SHA-256 and 4Kb User EEPROM	Samples

More Information
For Technical Support: https://www.maximintegrated.com/en/support
For Samples: https://www.maximintegrated.com/en/samples
Other Questions and Comments: https://www.maximintegrated.com/en/contact

Application Note 7015: https://www.maximintegrated.com/en/an7015
APPLICATION NOTE 7015, AN7015, AN 7015, APP7015, Appnote7015, Appnote 7015