White Paper 8: 1-Wire SHA-1 Overview

While discussions of data security often focus on encryption, an arguably more valuable aspect is authentication. This paper presents a straightforward technique for digital authentication of hardware, data, and users. It goes on to provide an overview of the 1-Wire® Secure Hash Algorithm 1 (SHA-1) devices that use this technique. Finally, it provides references to other documents, kits, and examples for continued research and development.

Introduction

Security is a pervasive problem in all things digital. While discussions of data security often focus on encryption, an arguably more valuable aspect is authentication. Since information and communication all boil down to transient ones and zeros, their validity and authenticity can come into question. This paper presents a straightforward technique for authentication of hardware, data, and users. It goes on to provide an overview of the 1-Wire® Secure Hash Algorithm 1 (SHA-1) devices that use this technique. Finally it provides copious references to other documents, kits, and examples for continued research and development. (*Special terms, commands, or codes are shown in italics for clarity. A glossary of terms can be found in White Paper 4: Glossary of 1-Wire SHA-1 Terms*

Hash

A hash is a distillation of a message. This distillation is typically much smaller than the message and is a constant size. A *cryptographically strong* hash must be *nonreversible*, meaning that by looking at the hash result there is no way to derive any part of the original message. It must also change significantly with any small change, even a single bit, in the input message. This is called the *avalanche effect*. The hash should also be *collision-resistant*, meaning that it is impractical to find two messages with the same hash. A hash with these properties can be used to verify that a message has not been altered.

MAC

A MAC (message authentication code) is a hash where a portion of the input message is secret. Only participants that know the secret can re-compute and verify the authenticity of the MAC. See Figure 1 for MAC generation.

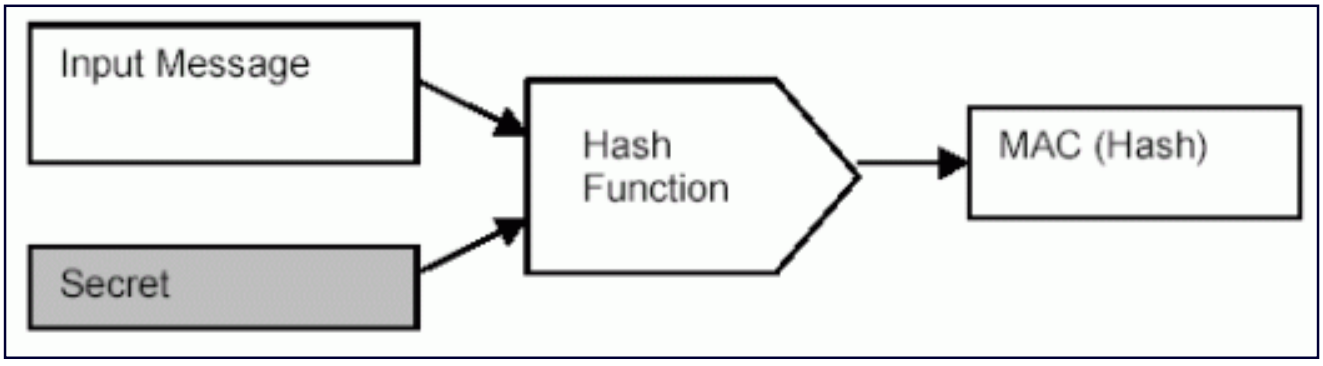

Figure 1. MAC generation.

Participants in a system (secret known) can therefore verify a message and MAC combination to be authentic. The transient ones and zeros mentioned earlier are now trusted without resorting to encryption. This is a powerful technique that has far-reaching applications (see *Applications* section below).

SHA-1

Selection of a cryptographically strong hash function is critical for the success of a hash-based authentication scheme. SHA-1 is nonreversible, collision-resistant, and has a good avalanche effect. SHA is a standard specified in Federal Information Publication 180-1 and 180-2 (FIPS 180-1, FIPS 180-2). The SHA series hashes are currently the only FIPS-approved method. SHA-1 is also specified in ISO/IEC 10118-3. According to Bruce Schneier in *Applied Cryptography, Second Edition*, John Wiley & Sons, 1996, "There are no known cryptographic attacks against SHA". For this reason, SHA-1 was selected. SHA-1 takes in one or more blocks of 512 bits (64 bytes) and creates a 160-bit (20-byte) hash.

Applications

Authentication is the process of proving to a host that a device, person, or message is valid and authentic. The authentication technique that takes advantage of a MAC is called challenge and response. A challenge is random data that is presented by a host to the device to be authenticated. The challenge is then included in the MAC calculation. This allows for every authentication session to be different and non-deterministic.

What can be authenticated? A message between two pieces of equipment could be authenticated if the secret is known by both pieces. A portable token could be authenticated to a door to allow access to a controlled room. See Table 1 below for a list of potential applications including both the host and authentication target.

Table 1. Applications using MAC

Application	Host	Authentication Target
Device authentication	Equipment	Peripheral device with embedded token
Physical access control	Electronic door lock Building access Filing cabinet lock Safe	Portable token
User access control	Workstation	Portable token and user
Software authorization	PC Software Equipment Firmware	Portable, attached, or embedded token
Electronic cash (eCash)	Vending machine Parking meter Toll booth Pay phone Gaming	Portable token

Creating a Trusted Token

The key component to most of the above applications is some kind of portable electronic token. To perform authentication with an SHA-1 MAC, a trusted client must be created that can securely keep the secret and compute the MAC. The client must be physically secure, rugged, and be able to perform the SHA-1 operation quickly. Dallas Semiconductor/Maxim has been making the iButton® portable tokens for over 10 years and now provides models with SHA-1 capability. Where the iButton form factor is not required a chip version is also available (see Figure 2).

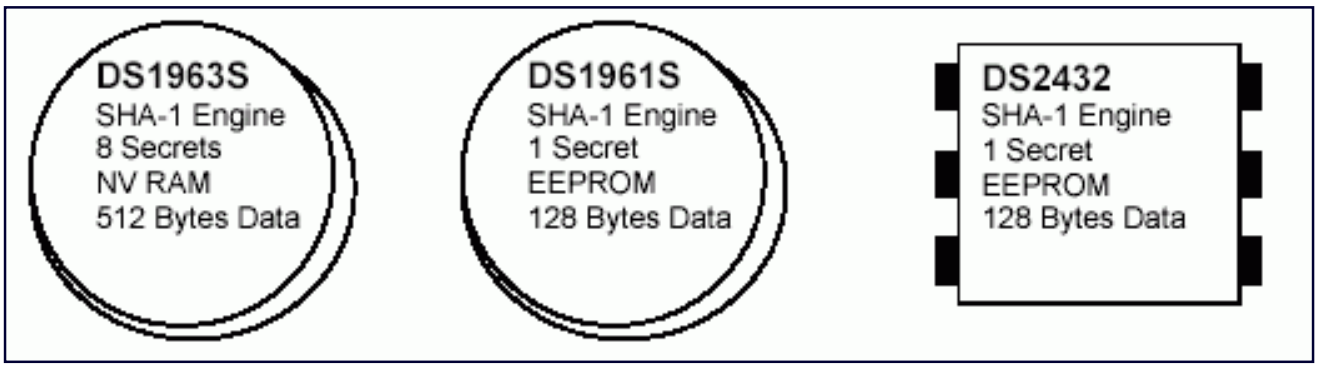

Figure 2. 1-Wire SHA-1 devices.

Discussions about applications and use of these tokens are simplified when provided in a context. Figure 3 illustrates a complete service. A service provider (e.g., vending company) would deploy tokens with valid data to their user base. The user would take the token to a *service control unit* (SCU) (e.g., candy machine) to have it authenticated as part of the service and the data validated (electronic cash, eCash) and optionally updated. The SCU must also be able to compute the SHA-1 MACs. This can be performed in an internal coprocessor. Specific fields and features of the token and the SCU will be explained in the following sections.

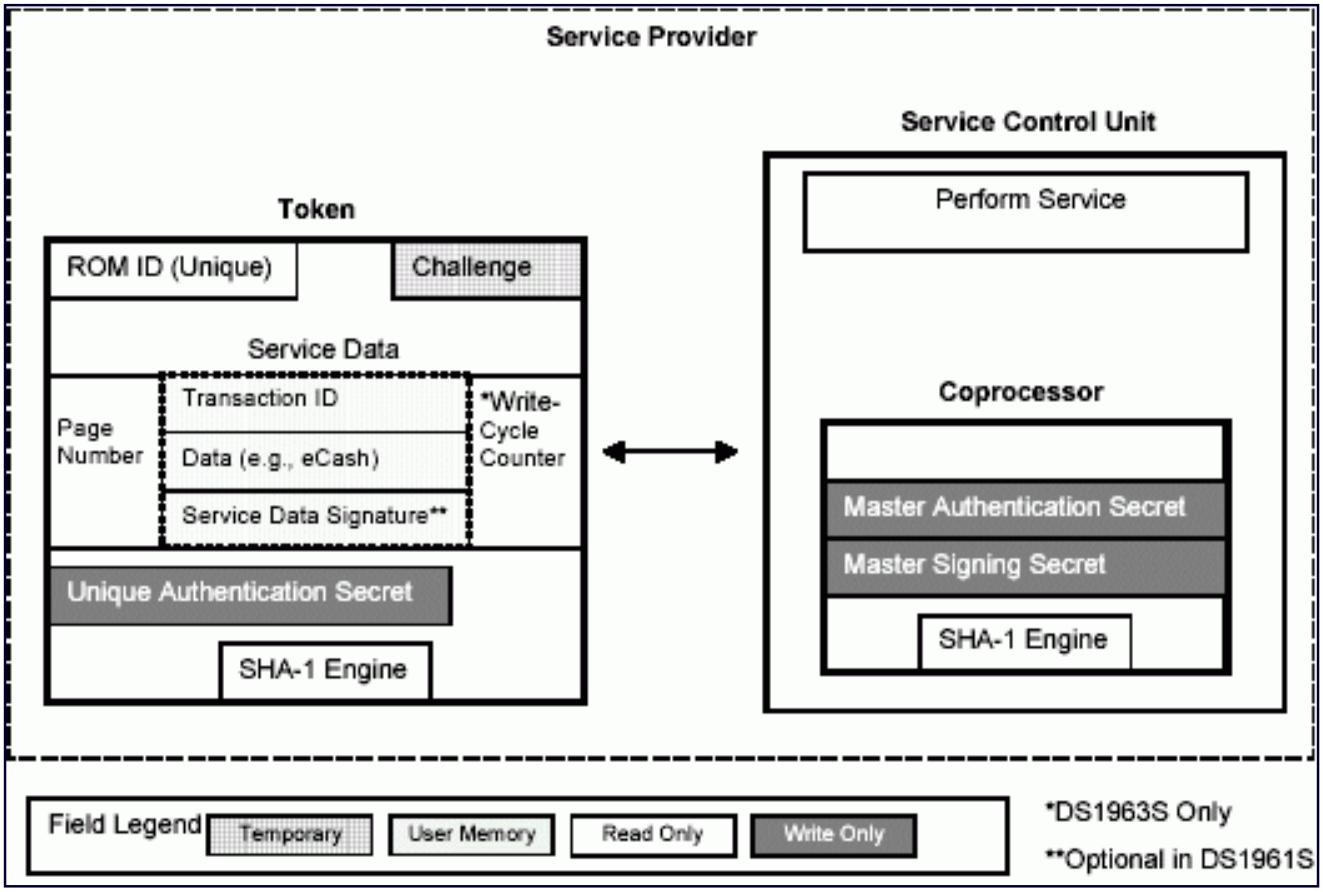

Figure 3. Service.

There are two different 1-Wire SHA-1 tokens. The DS1963S is the premium part with eight different SHA-1 secrets and 512 bytes of nonvolatile (NV) RAM user data space. The DS1963S can also be used as a coprocessor in the SCU to compute all of the necessary host-side SHA-1 MACs. This has the added security of removing the secrets from the SCU processor entirely. The secrets for both verifying the service data (*master signing secret*) and authenticating the token (*master authentication secret*) are securely kept in the DS1963S coprocessor. An internal lithium cell backs up the NV RAM inside the DS1963S. Removing this power source even for a short time (during a physical attack for instance) will clear the secrets and contents.

The DS1963S can support seven different dynamic service records, each with its own secret. The eighth secret is reserved for coprocessor-type functions. This device also contains eight more general-purpose 32-byte data pages that can be used for static service records. These extra pages have a secret associated with them but do not have a write-cycle counter. A write-cycle counter is a read-only, non-rolling-over page write counter that is used for detection of writes. The write-cycle is used to avert certain attacks to the system.

The DS1963S uses two different MACs for each service record. The first one is generated by the 1-Wire token and authenticates it to the system (*unique authentication secret*). The second MAC is created by the SCU host and included in the service record and is used to validate it. This MAC is called the *service data signature*. The secret used to create this signature (*master signing secret*) does not reside on the user device: it only exists within the SCU or its coprocessor.

The other two devices, DS1961S and DS2432, are essentially the same devices in different packages. The DS1961S comes in the standard iButton package and the DS2432 is in a small TSOC package or

in the chip-scale package (2.5mm by 1.5mm). These two devices use EEPROM technology instead of NV RAM, thus allowing a package without a lithium cell. They have only one secret associated with all four pages. There is no write restriction on the DS1963S. However, writing to the DS1961S/DS2432 requires presenting an authentication MAC to the device. This effectively write protects the device. Since write access is restricted to SCUs that know the authentication secret, the devices do not have a write-cycle counter and do not need a MAC signature embedded in the data. The DS1961S/DS2432 can only support one service provider since there is only one authentication secret. This service provider could install up to four different services since the device has four pages.

Note that the SHA-1 result created by the 1-Wire devices conforms to the FIPS 180-1 specification, except the additional final constants were removed. In the FIPS-180 specification, these constants are added to the result after each block is computed. Since the 1-Wire devices only do one block, adding these constants don't add any security and was removed to speed up the MAC computation in hardware. If FIPS 180-1 compliance is required, the SCU can add the constants back on.

Figure 3 also shows the pertinent memory and special registers of the 1-Wire tokens that will be used in verification of one service record. Each token contains a 64-bit unique device ID that is lasered into ROM. Since each device has this unique *ROM ID*, it can be used as one component in the calculation of the secret to make the authentication secret on each device unique (*unique authentication secret*). A host such as the SCU must re-compute the *unique authentication secret* from the *master authentication secret* and the *ROM ID* when an authentication is done. Each device also contains a temporary memory location to hold the authentication challenge sent by the SCU. A challenge is random data created by the SCU to make each authentication sequence unique. The *page number* is the physical location in memory that the data message resides in.

The 1-Wire devices take each of the fields and internally run them through the built-in SHA-1 engine, and then return the MAC for authentication. The SCU takes the same information using the calculated *unique authentication secret* and verifies that the MAC is correct. Figure 4 shows how the authentication MAC is constructed inside the device.

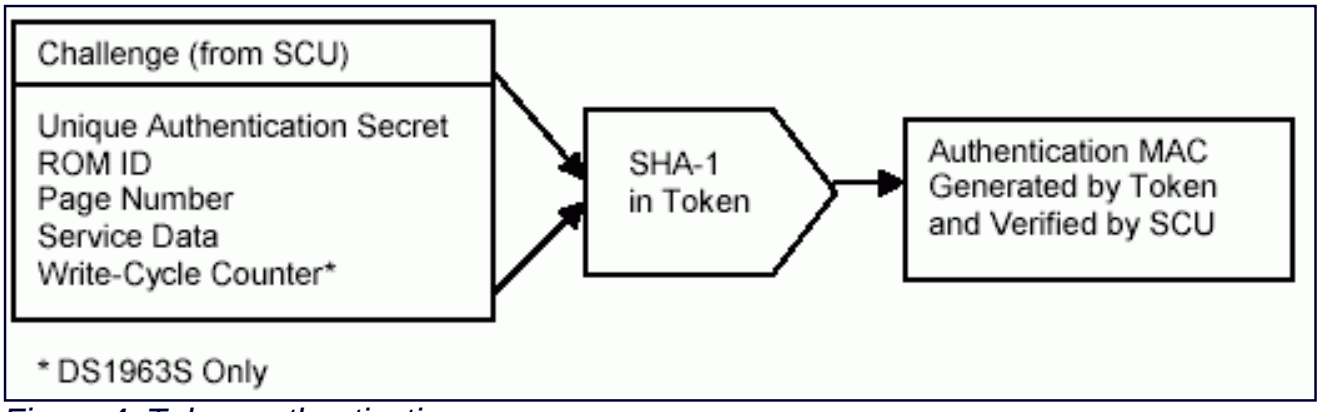

Figure 4. Token authentication.

Since the DS1963S has unrestricted write access, the service record must include a *service data signature* for validation of the data. This MAC is saved along with the service data and used to verify that the account data is valid. Figure 5 shows what information is put in to produce and verify the *service data signature*. Figure 6 shows a complete transaction with the SCU and a DS1963S token and Figure 7 shows the transaction for the DS1961S token.

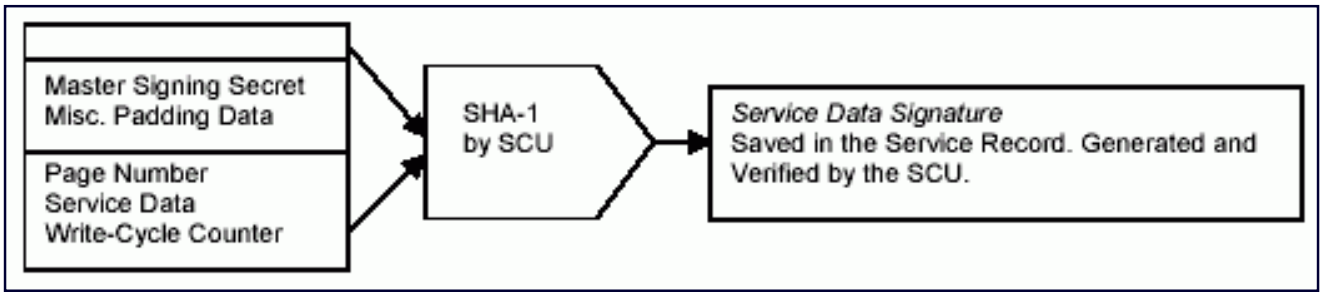

Figure 5. DS1963S service data validation.

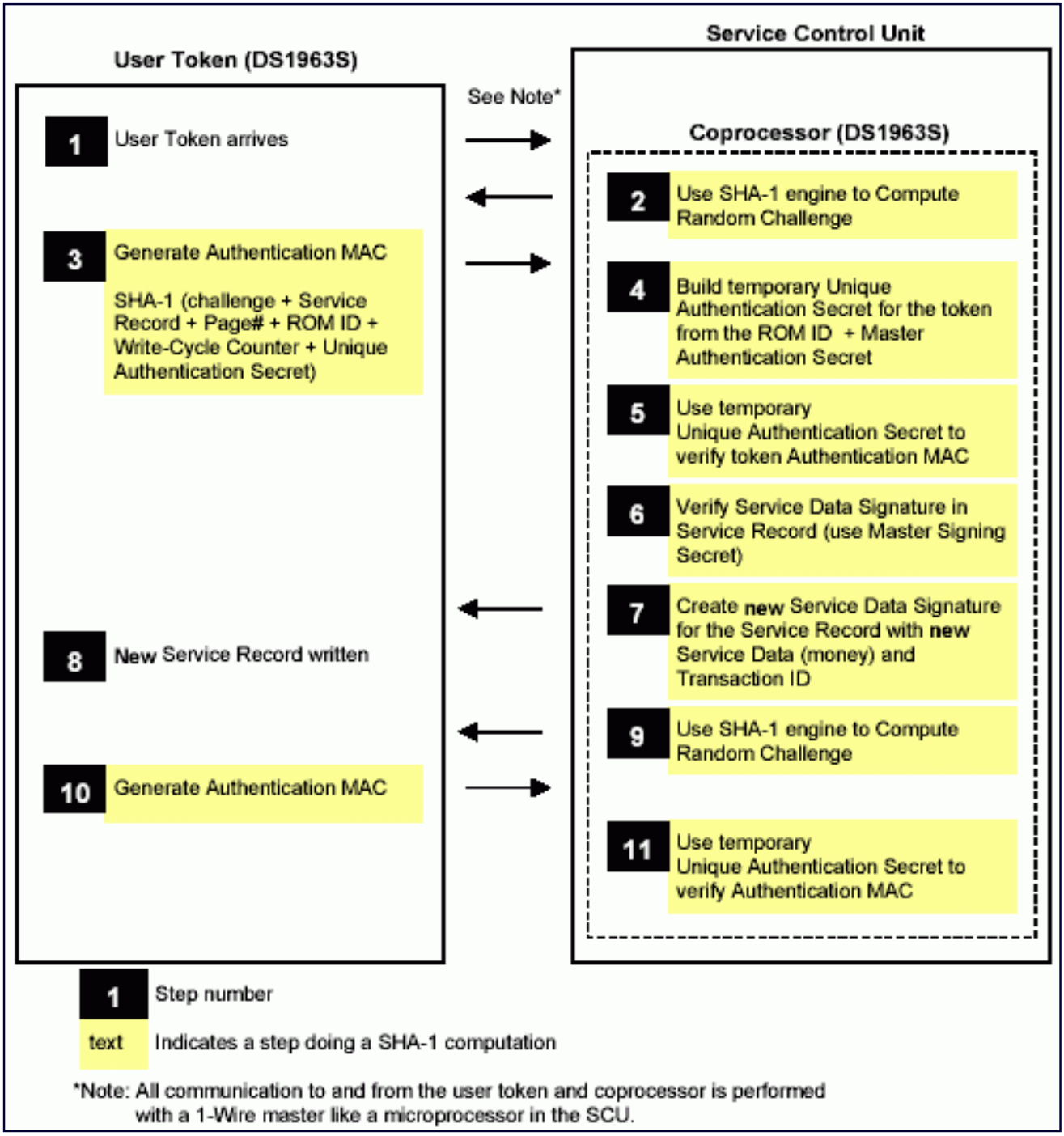

Figure 6. DS1963S typical transaction flow.

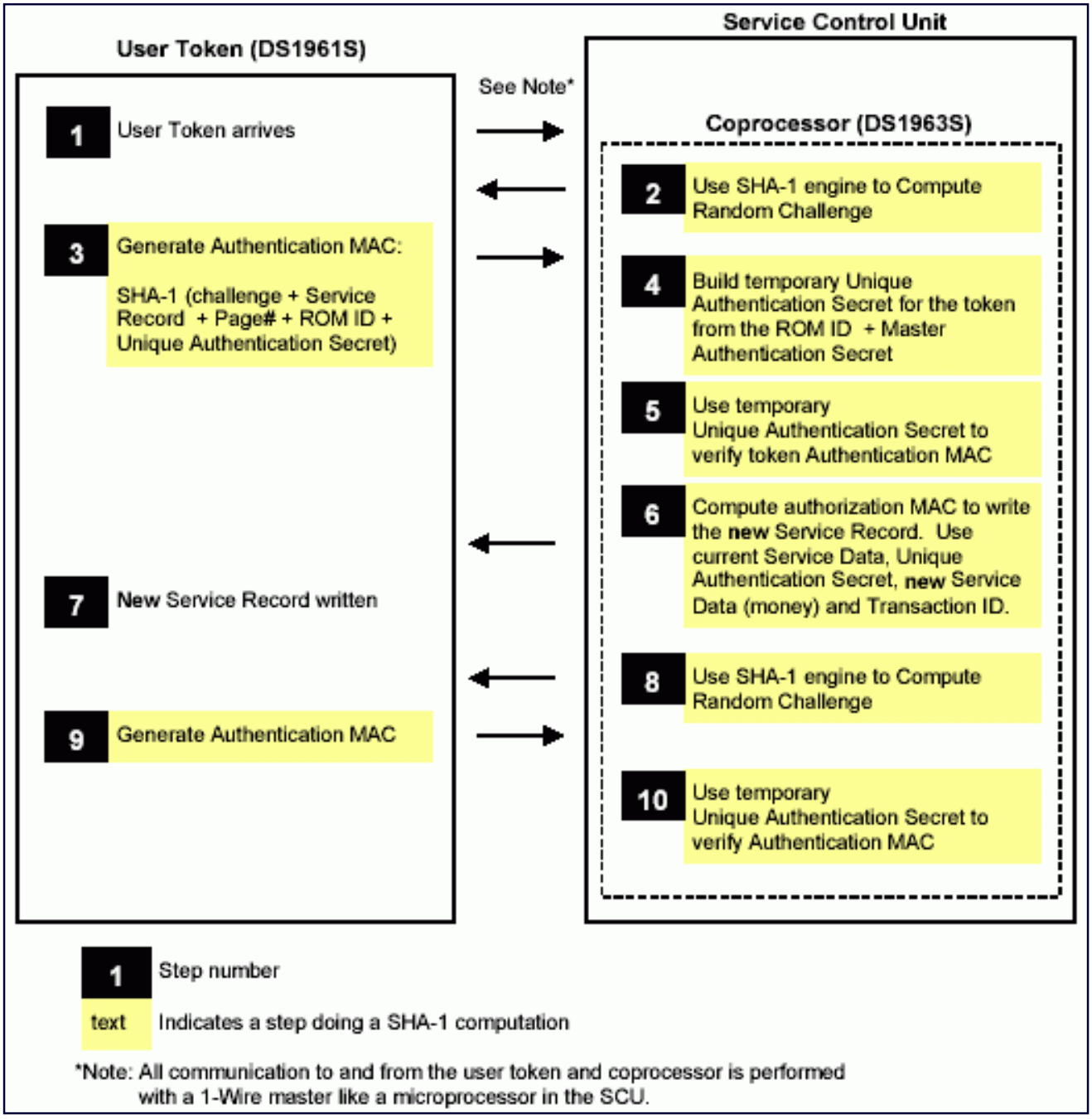

Figure 7. DS1961S typical transaction flow.

More Information

There is a large number of documents available to aid in understanding and implementing a system based on 1-Wire SHA-1 devices. The information in these documents can be divided into three categories: *Theory*, *Implementation Details*, and *APIs* (see Figure 8). The *Theory* resources discuss how electronic cash can be made secure with the algorithms available. The *Implementation Details* describe every step and bit of communication to accomplish the authentication and electronic cash operations with 1-Wire devices. The *API's* resources are useful to jump right in and start using the SHA-1 devices. The source code to the APIs can also be a useful development tool for custom implementations.

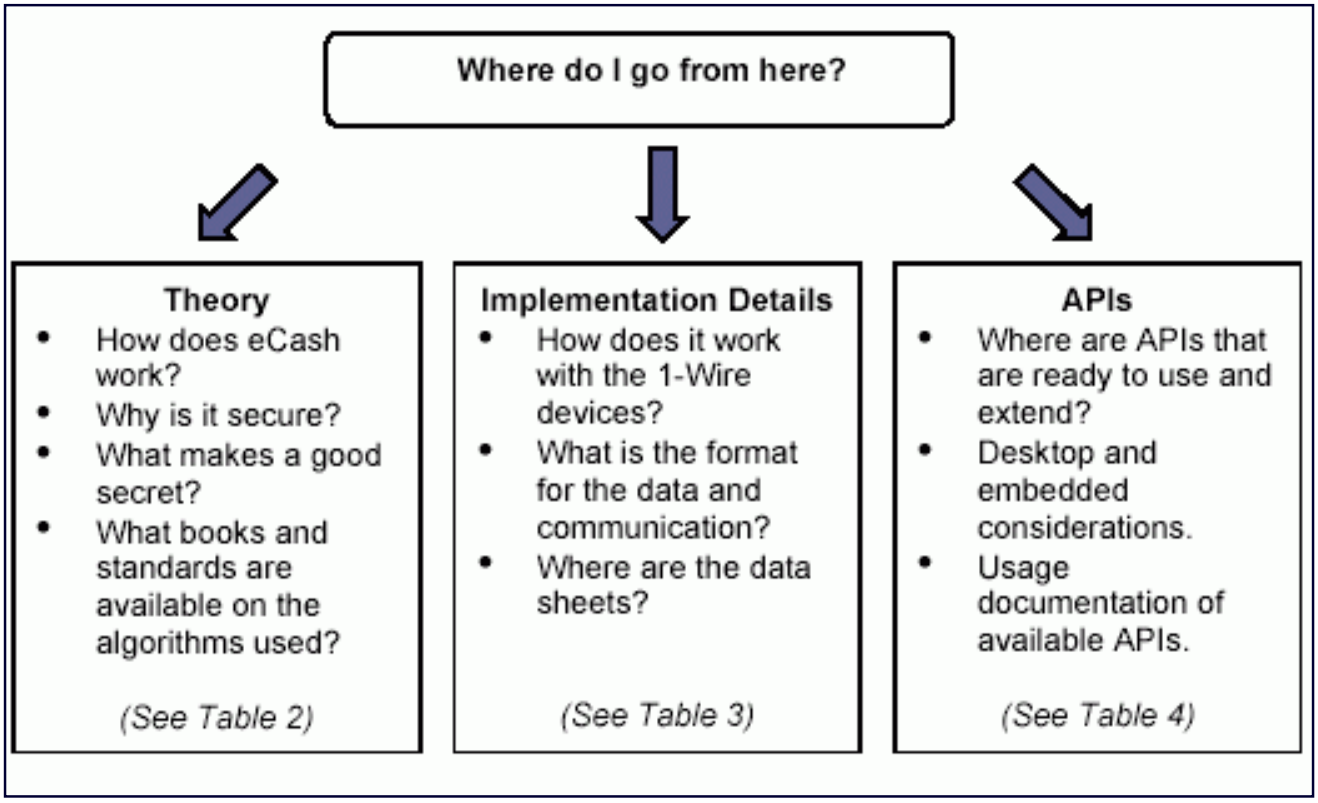

Figure 8. Resource overview.

Table 2. Theory

Electronic Cash and User Authentication using Dallas Semiconductor / Maxim DS1963S
Tutorial presentation containing a detailed introduction to electronic cash and the features of the SHA-1 1-Wire devices. It starts at the very beginning and builds up the necessary cryptographic concepts and required facilities of a portable token in an electronic cash (eCash) system.
Document Tutorial (Power Point)
URL ftp://ftp.dalsemi.com/pub/auto_id/public/ecash_sha_tutorial.ppt (4MB)
Glossary of 1-Wire SHA-1 Terms
This document contains a list of terms pertaining to the use of 1-Wire SHA-1 devices such as the DS1963S, DS1961S, and DS2432.
Document White Paper 4
URL http://www.maxim-ic.com/appnotes.cfm/appnote_number/1099
Challenge and Response with 1-Wire SHA devices
This application note describes the basic mechanisms of challenge and response. It provides an example implementation using the 1-Wire SHA devices. The example code using the 1-Wire API for Java™is available for download.
Document Application Note 190
URL http://www.maxim-ic.com/appnotes.cfm/appnote_number/190
SHA iButton Secrets and Challenges

Overview of the importance of selecting good secrets and truly random challenges. This paper provides several rules of thumb to avoid weaknesses in a system.

Document Application Note 152

URL http://www.maxim-ic.com/appnotes.cfm/appnote_number/152

Why are 1-Wire SHA-1 Devices Secure?

Presents a usage scenario of 1-Wire SHA-1 devices in a *Service*. Outlines possible attacks on the *Service* and how the features of the 1-Wire devices or a recommended usage procedure thwart them.

Document White Paper 3

URL http://www.maxim-ic.com/appnotes.cfm/appnote_number/1098

Passwords in SHA Authentication

This document describes a technique where a password or PIN can be used with 1-Wire SHA-1 devices to authenticate a user with the device.

Document Application Note 154

URL http://www.maxim-ic.com/appnotes.cfm/appnote_number/154

Information technology - Security techniques - Hash-Functions - Part 3

ISO standard presenting various hash standards including SHA-1. This standard is then referenced in other ISO standards relating to MAC creation.

Document ISO/IEC 10118-3 Standard (PDF for purchase)

URL http://webstore.ansi.org/ansidocstore/product.asp?sku=ISO%2FIEC+10118%2D3%3A1998

Secure Hash Standard

Federal Information Publication Standard (FIPS) specifying the SHA-1 standard.

Document FIPS 180-1

URL http://www.itl.nist.gov/fipspubs/fip180-1.htm

Applied Cryptography, Second Edition

Good general introduction text on cryptography techniques and theory. One way hash functions including SHA-1 are described in Chapter 18.

Document Bruce Schneier, John Wiley & Sons, 1996 (Print)

URL http://www.wiley.com/cda/product/0,,0471128457|desc|2941,00.html

Handbook of Applied Cryptography

Extremely detailed text covering various cryptographic techniques including hashes. Chapter 9 specifically covers hashes and data security.

Document A. Menezes, P. van Oorschot, and S. Vanstone, CRC Press, 1996 (Print, PS, PDF)

URL http://www.cacr.math.uwaterloo.ca/hac/

Table 3. Implementation Details

SHA iButton 1-Wire API Overview

A detailed overview of the steps necessary to create an API for a complete SHA-1 eCash system. This is the guide that was used to create the APIs put forth in the developer's kits by Dallas Semiconductor.

Document Application Note 157

URL http://www.maxim-ic.com/appnotes.cfm/appnote_number/157

Dallas Digital Monetary Certificates

Explains the standard certificate data format used in the Dallas Semiconductor APIs to represent a monetary account. The format includes the location of the data and the MAC associated with it.

Document Application Note 151

URL http://www.maxim-ic.com/appnotes.cfm/appnote_number/151

SHA-1 Devices used in Small Cash Systems

Very detailed step by step operation descriptions to do authentication and eCash transactions on 1-Wire devices. This provides an excellent reference for implementation and verification of an eCash system.

Document White Paper 1

URL http://www.maxim-ic.com/appnotes.cfm/appnote_number/1820

eCash Firmware Guide

This guide explains the flow and states of the firmware used in the eCash evaluation board. The eCash board contains a processor and two DS1963S to serve as coprocessors for both DS1963S and DS1961S user tokens. The firmware is written mostly in portable C code with small sections in 8051 assembly.

Document Application Note

URL (not published yet, please monitor http://www.maxim-ic.com/appnotes10.cfm/ac_pk/1)

DS1963S SHA iButton

The data sheet for the DS1963S contains all of the commands and operating conditions for the device.

Document Data Sheet

URL http://www.maxim-ic.com/quick_view2.cfm/qv_pk/2822

DS1961S 1kb Protected EEPROM iButton With SHA-1 Engine

The data sheet for the DS1961S contains all of the commands and operating conditions for the device.

Document Data sheet

URL http://www.maxim-ic.com/quick_view2.cfm/qv_pk/3557

1-Wire File Structure

The eCash certificate that is used in the APIs and documentation provided by Dallas Semiconductor are contained in a file on the device that conforms to a standard 1-Wire file structure.

Document Application Note 114

URL http://www.maxim-ic.com/appnotes.cfm/appnote_number/114

Table 4. Application Programmer Interfaces (APIs)

1-Wire Public Domain Kit

The 1-Wire Public Domain kit is a portable 'C' library with extensive examples covering almost all 1-Wire devices including iButtons. There are examples to implement simple authentication and a complete eCash system. The complete source is provided under a very liberal license.

Document kit (zip)

URL http://www.maxim-ic.com/products/ibutton/software/1wire/wirekit.cfm

1-Wire API for Java Software Development Kit

The 1-Wire API for Java was designed from the ground up to be a very robust, highly object-oriented foundation for building 1-Wire applications in Java. It has examples to implement a complete eCash system. The complete source is provided under a very liberal license and supports J2ME (Java 2 Micro Edition).

Document kit (tar-aip, tgz)

URL http://www.maxim-ic.com/products/ibutton/software/1wire/1wire_api.cfm

Using the eCash Processor

The *eCash Processor* is a microprocessor with one or more DS1963S SHA-1 coprocessors running the embedded eCash reference implementation. An example of this can be found in the eCash Evaluation Kit. The *eCash Processor* has a convenient serial interface to control and track eCash debiting. This document explains this interface so the *eCash Processor* can be used as an element in a larger system.

Document White Paper

URL (not published yet, please monitor http://www.maxim-ic.com/appnotes10.cfm/ac_pk/1)

DS1963S SHA 1-Wire API Users Guide

This guide explains how to use the DS1963S eCash implementations in the 1-Wire Public Domain 'C' and 1-Wire API for Java provided by Dallas Semiconductor.

Document Application Note 156

URL http://www.maxim-ic.com/appnotes.cfm/appnote_number/156

Guide to the Embedded eCash Processor Reference Implementation

This document explains the design considerations, flow, and coding details of the eCash Processor reference implementation. The source code for the reference implementation is almost entirely in portable 'C' optimized for microprocessors and other small memory devices. An example of this can be found in the eCash Evaluation Kit.

Document Application Note

URL (not published yet, please monitor http://www.maxim-ic.com/appnotes10.cfm/ac_pk/1)

More Information

DS1961S: QuickView -- Full (PDF) Data Sheet -- Free Samples

DS1963S: QuickView -- Full (PDF) Data Sheet -- Free Samples

DS2432: QuickView -- Full (PDF) Data Sheet -- Free Samples