Keywords: access control, secure, challenge & response, authentication, RFID, iButton, message authentication code, MAC

APPLICATION NOTE 4784

Secure Access Control Through Challenge and Response Authentication

By: Bernhard Linke, Principal Member Technical Staff
Apr 01, 2011

Abstract: This application note examines keys for physical access control from a new perspective: information technology (IT). It compares various types of key technology (mechanical, magnetic, contact, RFID), and evaluates these keys for their strengths and weaknesses. Keys with challenge and response authentication overcome the limitations of the traditional static data keys. The challenge and response concept is discussed; suitable keys are presented and compared. The document concludes with an explanation of why challenge and response authentication is more secure than concepts that rely on static data.

A version of this app note was published by Embedded.com on March 15, 2011.

Click here for an overview of the wireless components used in a typical radio transceiver.

Introduction

For millennia people have used locks and keys to control access to their dwellings and treasures. As technology changed, so did the locks. Today mechanical locks still dominate the world. However, as a close look at your car key or employee badge most likely reveals, electronics has already entered the access control territory.

This application note reviews keys for access control: mechanical, magnetic, contact, RFID. It describes challenge and response authentication (the challenge, secret, and message authentication code or MAC) and the important role of the SHA-1 algorithm. Finally, the article explains why challenge and response authentication is more secure.

The Key as an Information Technology (IT) Device

From a strictly logical perspective, any key stores information like a ROM (read only memory). The lock "reads" they key's data and, if it matches the lock's criteria, gives access. The physical size and the smallest dimension detail (i.e., an increment) of a **mechanical key** limit the available code space. For a given key style, hundreds or thousands of keys can be manufactured without duplication; the exact number depends on the style. **Magnetic stripe** key cards store information with tiny iron-based magnetic particles. The magnetic stripe can be written in multiple parallel tracks of more than 500 bits each.

Contact-based **electronic token keys** (e.g., iButton® devices, chip cards) store information in silicon chips. The number of bits available can be as low as 64 (DS1990A) or virtually unlimited. Contactless keys start with models featuring only 26 bits (see Wiegand Public Format, PDF) and have practically no upper limit. Magnetic stripe key cards are popular for room access in hotels. Electronic token keys, with and without contact, are popular for employee badges.

Status Quo and Its Issues

The actual opening of the lock, be it mechanical or electronic, is solely based on the presence of static data that satisfies the lock's built-in criteria. With electronic locks, this data could be a simple identification number, hundreds or thousands of memory bits (e.g., a magnetic stripe or memory chip card), or a combination of both. The less information that a key carries, the more keys a given lock can memorize.

Mechanical keys are available in many styles and sizes.[1] The "owner" of the lock has no protection against unauthorized key duplications. In addition, inexpensive tools are available to open the lock without the right key.[2] Due to the limited code space, moreover, the uniqueness of a key is not guaranteed. Over time the fine structures of a key wear off, making it increasingly more difficult to open the lock.

Although code space is not an issue with magnetic stripe key cards, they can easily be duplicated[2] or erased. They deteriorate from wear and tear.

ROM-based electronic keys are subject to emulation (replay) and copying. This is true for contact keys[2] and RFID keys.[2] Except for applications based on the Wiegand format and derivatives (26 bits or 36 bits), electronic keys have enough code space to guarantee a unique code for every key.

The Next Level of Security: Challenge and Response Authentication

Traditional electronic locks rely on static data that the key needs to produce to gain access. This unchanging criterion makes it easy to succeed with cloned keys. A much higher level of security is achieved if the key can receive an unpredictable data inquiry from the lock and respond with a data pattern that depends on the data received. The process involves openly readable data and hidden data that is known only to the key and the lock.

The technical term for the unpredictable data that the lock sends to the key is a *random challenge*. The hidden data is called the *secret*, and the response is commonly referred to as the *message authentication code*, or MAC. The message consists of the challenge, openly readable data, the secret, and constants (padding). To verify the authenticity of a key, the lock computes a MAC using the same challenge, data read from the key, the secret, and constants. If the MAC computed by the lock matches the MAC from the key's response, the lock knows that the key is authentic. The technical term for this process, illustrated in **Figure 1**, is challenge and response authentication. If, in addition to the authenticity, the openly readable data in the key also matches the lock's criteria, the lock gives access.

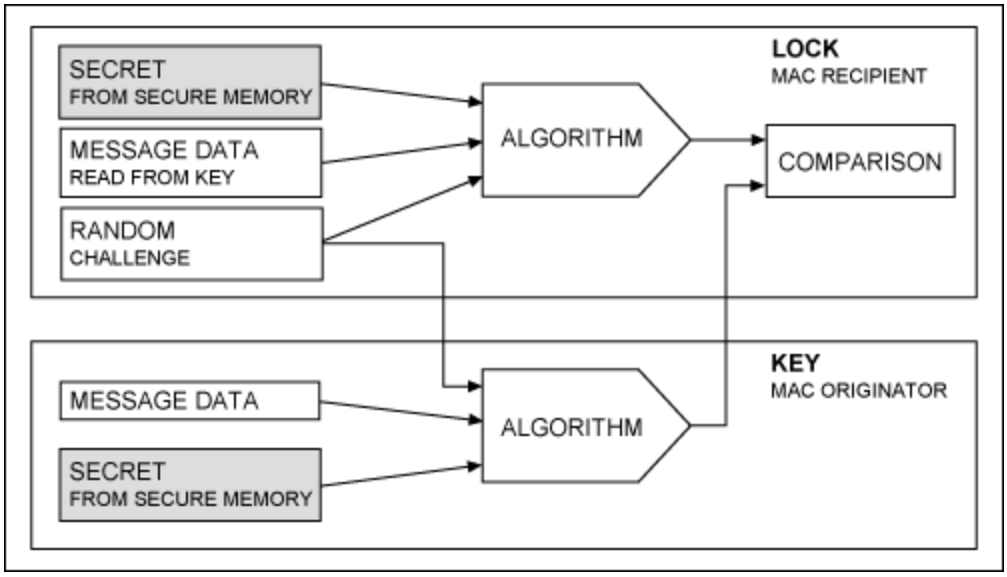

Figure 1. Challenge and response authentication data flow.

In cryptography, an algorithm that generates a fixed-length MAC from a message is called a 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.
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.

Keys for Challenge and Response Authentication

Locks for electronic keys, with and without contact, have been developed by several companies worldwide and are deployed in large numbers. Their main component is a microcontroller with built-in firmware, (i.e., the software program) and memory that stores the criteria for the keys to be accepted by the lock, (e.g., identification numbers or text strings). By design, the lock has all the resources to work with challenge and response keys. All the lock needs is a firmware upgrade.

The DS1961S challenge and response iButton with SHA-1 engine was introduced in the year 2002. A contactless device, the MAX66140 ISO 15693-Compliant Secure Memory Fob, followed in 2010. Although their communication interface and form factor are very different, both devices have a lot in common. **Table 1** shows the details. They both support SHA-1 authentication using a 64-bit secret and have 1024 bits of user-programmable EEPROM. The secret can be loaded or computed—no authentication is needed for this step—and write-protected. Writing to the memory requires authentication, i.e., one can only write if the secret stored in the device is known. As a newer

development, the MAX66140 uses a 5-byte challenge compared to the 3-byte challenge used with the DS1961S. The MAX66140 also has memory write-cycle counters, which make tamper detection easy and extend the device's application beyond access control to closed-loop monetary systems.

Table 1. Comparison Between the DS1961S and MAX66140

Feature	DS1961S	MAX66140*
Form factor	iButton, F3 and F5 size	Plastic key fob
Communication interface	Contact-based, 1-Wire® protocol	Wireless, 13.56MHz ISO15693 and ISO18000-3 Mode 1
Data rate	Standard speed: up to 15.3kbps; overdrive speed: up to 125kbps	Slow speed: 1.6kbps down, 6.6kbps up; fast speed: 26kbps down and up
ID#	64-bit 1-Wire ROM ID	64-bit UID, ISO compliant
Authentication method	160-bit SHA-1 MAC	160-bit SHA-1 MAC
Secret size	64-bit (read protected)	64-bit (read protected)
Secret generation	Load, compute; optional write protection through separate register write access	Load, compute, optional **automatic** write protection
User memory	1024 bits organized as four pages of 32 bytes; write access in 8-byte blocks; user-programmable write protection for page 0 only or for all four pages together; user-programmable EPROM emulation mode for page 1 only	1024 bits organized as 16 blocks of 8 bytes; four blocks form a 32-byte page; write access in 8-byte blocks; **individual block write cycle counter**; user-programmable write protection for each individual block; user-programmable EPROM emulation mode for each individual page; user-programmable read protection for page 3
Write authentication MAC	Involves ID#, page #, page data, new data, secret, constants	Involves ID#, page #, page data, new data, secret, **write cycle counter**, constants
Read authentication MAC	Involves ID#, page #, page data, 3-byte challenge, secret, constants	Involves ID#, page #, page data, **5-byte** challenge, secret, constants

*An equivalent key with ISO/IEC 14443 Type B interface, the MAX66040, is in preparation.

Why Challenge and Response Authentication Is More Secure

To set up and maintain a challenge and response authentication system one needs a key programmer (i.e., an electronic device), and, depending on the system concept, a master key. The key programmer must know vendor-specific data conventions and the algorithm to generate secrets. If the system supports this feature, the master key would be used to update the known key ID# list stored in a lock. As with any security system, the physical access to these tools must be strictly controlled to prevent unauthorized use.

Creating a New Key or Key Duplication

Using the key programmer, the authorized locksmith installs a valid secret in a blank key and then programs the memory with valid data. (In case of a duplication: data is copied from the other key.) The new key is now ready for use. Depending on the lock firmware, it may be necessary to instruct the lock(s) using a master key to add the new key's ID# to the list of known keys. A hacker can load an arbitrary secret into a blank key and then program the openly readable memory with valid data. This key,

however, fails the challenge and response authentication because its secret is not valid in the system.

Changing the Access Rights of a Valid Key

Using the key programmer, the authorized locksmith updates the data in the key's memory for the changes. Without knowing the key's secret or without access to suitable equipment, the hacker cannot generate the write authentication MAC needed to write to the key's memory.

Taking a Key Out of Service

Using the key programmer, the authorized locksmith changes the key's memory contents to "factory default" or any other pattern that is easily recognized as invalid. The key's secret can stay as is. If the locks maintain a list of known keys, it is advisable to also delete the ID# of the invalidated key from the locks. The key can be reprogrammed for later use. The hacker's option is to delete the key using brute force.

Defeating Key Emulation

Consider this scenario. With eavesdropping and recording equipment in place, the hacker repeatedly presents a valid key to the lock. Next the hacker analyses the recorded data to see the challenges sent by the lock and the read authentication MACs generated by the key. If the firmware is properly designed, the challenges are random, making it impossible to record all combinations of challenge and response. This forces the hacker to give up.

Poorly designed lock firmware uses a constant challenge or randomly picks the challenge from a small list of patterns—exactly the weakness that a hacker is looking for. In that case the hacker can program a key emulator with the valid key's ID# and memory data, the challenges sent by the lock, and the corresponding read authentication MACs. If the lock maintains a list of known keys, the easiest countermeasure is deleting that key from the list in the lock. Systems that do not use such a list are not defenseless, though. To detect a key emulator, one could have the lock write random data to an otherwise unused memory section in the key. The emulator would accept the write access, since it cannot check the validity of the write access MAC. Next, the lock reads back the just-written data together with the read authentication MAC of that page. Since it is not prepared for this activity, the emulator is unveiled because it is unable to produce a valid MAC.

Defeating a Leaked Secret

The 64-bit data that serves as a secret for challenge and response authentication can be loaded or computed. The worst action that one can do is to put the same secret into all keys of the system. Once this secret is leaked or discovered by trial and error, the system security is broken. The keys for challenge and response authentication, therefore, can compute a secret from the initial (i.e., the current or loaded) secret, a partial secret, data from one of the memory pages, and device-specific (i.e., known) constants. This way, the secret is never exposed. The secret can also be made device specific by using the key's 64-bit ID# as partial secret. Should a single key's secret ever be disclosed, it would compromise that particular key but not the entire system.

Conclusion

In places where electronic locks or electronic access control are already installed, security can be significantly improved through challenge and response authentication. Challenge and response keys are available with a contact interface or as wireless key fobs. Data in challenge and response keys is protected against unauthorized changes. A memory write-cycle counter can expose tampering. Cloned challenge and response keys fail the authentication test, even if the openly readable memory data is valid. Upgrading an existing installation to challenge and response keys can be as simple as issuing new

keys and installing new firmware in the locks or readers.

References

[1]Key (lock): http://en.wikipedia.org/wiki/Key_(lock).
[2]An Internet search will identify hundreds or thousands of possible sources for these items. This is, by itself, an indication of the potential problems of electronic locks without challenge and response security.

Related Parts		
DS1961S	1Kb Protected EEPROM iButton with SHA-1 Engine	
MAX66040	ISO/IEC 14443 Type B-Compliant Secure Memory	Free Samples
MAX66140	ISO 15693-Compliant Secure Memory	Free Samples

More Information

For Technical Support: http://www.maximintegrated.com/support
For Samples: http://www.maximintegrated.com/samples
Other Questions and Comments: http://www.maximintegrated.com/contact

Application Note 4784: http://www.maximintegrated.com/an4784
APPLICATION NOTE 4784, AN4784, AN 4784, APP4784, Appnote4784, Appnote 4784