Western Digital.

WHITE PAPER FEBRUARY 2021

e.MMC Security Methods

A Detailed Overview of the Different Security Methods One Can Use in an e.MMC Storage Device

Einav Zilberstein | Adi Klein

Western Digital Corporation

Contents

Introduction

The term "information security" can cover a number of very different design features. In general, information security is the practice of preventing unauthorized access, use, disclosure, disruption, modification, inspection, recording, or destruction of information.

The three fundamental goals for information security are confidentiality, integrity, and availability:

- **Confidentiality** means that information that should stay secret can be read and understood only by authorized entities. Others without access authorization can't read or understand the confidential information.
- **Integrity** means the ability to ascertain that the information is protected from unauthorized alteration, modification, or deletion. Integrity of information covers its origin, completeness, and correctness using methods such as identification and authentication.
- **Availability** means that the information is always available to the authorized users.

Every system design will support these security goals in different ways, depending on the type and value of the assets it is trying to protect. Every security solution should be able to protect against subsets of possible attacks, but a combination of several solutions is more likely to achieve a design that is completely secure. For example, e.MMC write protect is designed to ensure data availability. The replay protected memory block (RPMB) solution is designed to ensure data integrity. Compare that to Android FDE (full device encryption), a different type of security solution that is designed to protect data privacy and to ensure confidentiality.

The Evolution of e.MMC Security Features

e.MMC devices contain multiple data protection and security features including: password lock/unlock, write protect, and RPMB. These features have evolved over the years and continue to improve with each version of the eMMC specification.

Password Lock

Password lock was the first security feature integrated into the eMMC spec; previously it had been implemented in legacy SD™ cards. The password lock feature is designed to protect the contents of the user area from any type of access (read, write, or erase).

The password lock/unlock feature is set using CMD42. After password lock is enabled, a host can perform certain actions—including reset, initialize, select, query for status, etc.—but may not access any data on the user area of the device. The host can still access the boot partitions, RPMB, and general partition area.

This kind of protection can be useful against data theft, but it also limits what anyone (including the data owner) can do with the device because no access (of any type) is allowed to the protected data.

Write Protect

Write protect was designed to protect against data corruption or erasure (whether malicious or unintentional). In eMMC4.3 and earlier specs, write protect provided protection only for the user area. With the introduction of partitions in eMMC4.4, write protect was updated to support both different partitions and areas within partitions. With the release of eMMC5.1, write protect enables the use of authentication to protect partitions and areas within them.

Once write protect is set, a host can't erase or write to the specified protected area. However, unlike password lock, data can still be read from this area.

In e.MMC version 5.1, there are four types of write protection:

- **Permanent:** once write protect is enabled, it cannot be disabled.
- **Power-on:** once write protect is enabled, it can only be reversed by a power cycle or by toggling the device reset pin, which causes the device to reboot itself.
- **Temporary:** write protect can be enabled and disabled.
- **Secured:** write protect can be enabled and disabled only for those who are authorized to use the RPMB.

Today, Write protect feature can be enabled on the overall device (including boot and RPMB partition) as well as on small areas within a partition called write protect groups:

- **The entire Device** (including the Boot Area Partitions, General Purpose Area Partition, RPMB, and User/Enhanced User Data Area Partition) may be write-protected by setting the permanent or temporary write protect bits in the CSD.
- **Boot partitions** can be permanent, secured or power-on write protected.
- **User Data Area (UDA) and General Purpose Partition (GPP)** write protection can be applied to specific segments (or as they called in spec, "write protect groups"), which may be set to permanent, secured, power-on, or temporary write protect.

RPMB – Replay Protect Memory Block

The RPMB (Replay Protected Memory Block) feature was first introduced in eMMC4.4. This feature enables a device to store data in a small, specific area that is authenticated and protected against replay attack.

RPMB is a self-contained security protocol with its own command opcodes and data structures. The mechanism for this protocol involves a shared key and a HMAC (Hash Massage Authentication Code), which is used to sign all the read/write operations accessing the secured area.

> With the release of eMMC5.1, write protect enables the use of authentication to protect partitions and areas within them.

RPMB – Authentication and Integrity for Replay Attack Protection

RPMB enables an e.MMC device to store data in a specific area (typically 4MB in size), where it is authenticated and protected against replay attack.

What Is a Replay Attack?

A replay (or playback) attack occurs when a program copies data from a legitimate interaction involving two entities and then retransmits the same data in a later stage. As the original information contains the right sender and destination identifiers, as well as proof of data authenticity, the retransmitted data will (if no measures are taken against the replay attack) be accepted, just as it was the first time it was transmitted.

Such an attack may be perpetrated by the originator who initiated the first / legitimate interaction, or by a "man-in-the-middle" (MITM) who had sniffed the original data and retransmitted it later on.

For an example of a replay attack, imagine a digital wallet service provider sending an authenticated message that sets a user account balance to $2,000. When the user pays a $1,600 bill with the digital wallet, the available balance drops to $400. If a piece of malware running a replay attack intercepts the initial message (the one setting the account balance to $2,000), by resending that same message after the $1,600 purchase it can reset the account balance to $2,000.

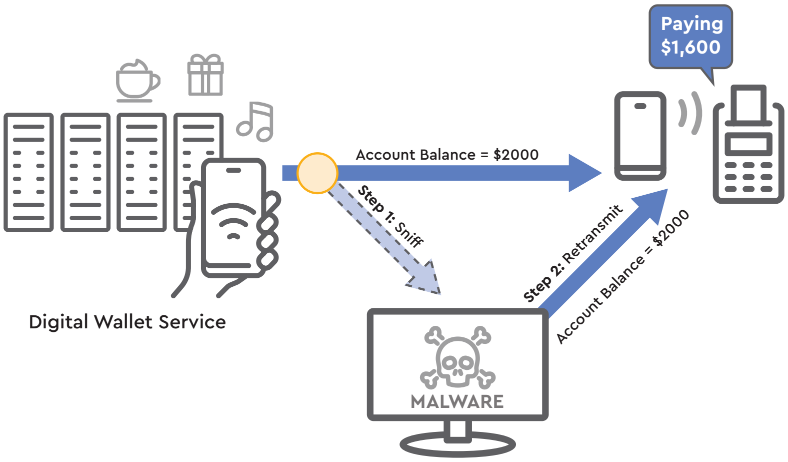

Figure 1: Replay Attack General Example

RPMB Authentication

RPMB uses symmetric key authentication, in which that same authentication key is used by both the host and the device (this key is also known as the "shared secret"). It works in the following manner:

- The RPMB authentication key information is first programmed by the host to the eMMC device (this must take place in a secure environment, usually on the production line).
- The authentication key is then used by both host and device to sign and authenticate read and write messages involving the RPMB area.
- Signing the message involves a message authentication code (MAC), which is calculated using the HMAC SHA-256 algorithm.

RPMB Protection Against Replay Attack

The basic idea of a replay protection is to ensure that every message is unique. In RPMB, the device manages a read-only counter that is incremented after every write message and whose new value will be included in the calculation of the next authentication code to be sent.

RPMB commands are authenticated by the HMAC SHA-256 calculation which takes as input:

- The shared/secret key.
- The message, containing the write command or the read results.
- The write counter, which counts the total number of writes to the RPMB.
- A nonce, which is a randomly generated number for each read command.

The resulting MAC is a 256 bit (32Byte) cypher that is embedded in the RPMB data frame and sent with the message data itself.

Writing to RPMB

When an eMMC device receives a write command message to the RPMB, it verifies the validity of the command by checking that (1) the counter was increased and (2) the MAC that was sent by the host is identical to the MAC that the device generated using its saved key.

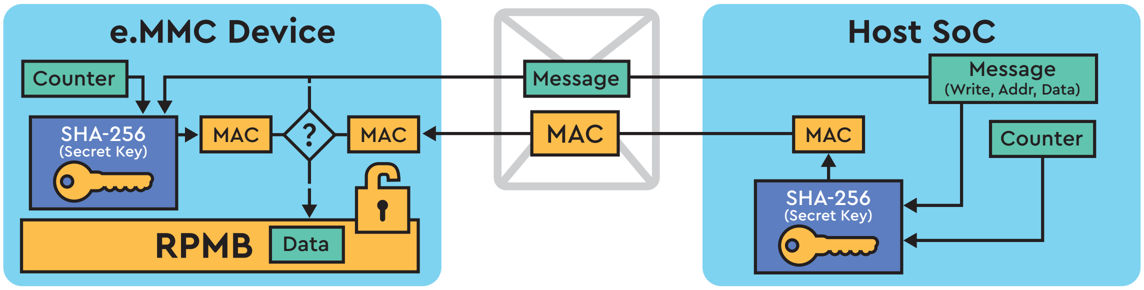

Figure 2: RPMB Write Sequence Block Diagram

Reading from RPMB

The eMMC device will send the read data to the host together with a MAC signature. The host receives the message and uses the shared key to generate a MAC as well. Only if the two MACs are identical will the host trust the data that is being read from the RPMB.

The use of random number generation and a counter register are key to providing protection against replay attacks:

- In case of a write to RPMB, the value of the MAC is influenced by the RPMB write counter, which is being increased (by both host and device) upon each successful write operation to RPMB.
- In case of a read from RPMB, the value of the MAC is influenced by a random number generated by the host and sent as part of the read request.

RPMB Use Cases

Different vendors use RPMB for different purposes, but certain use cases are good candidates for RPMB. Some well-known use cases include software version authentication, fingerprint verification, secure key storage, network vendor information, digital rights management (DRM) and secure payments.

Example 1 – Software Version Authentication to Prevent a Downgrade Attack

Consider a scenario in which a manufacturer must push several software updates to a device (such as a phone, a car, etc.). During the initial update, the new software image is written into the eMMC's main area while the information about the software release version is stored in the RPMB.

Then, imagine that the manufacturer discovers that this version has a security breach or a safety bug.

The manufacturer issues another software update to fix the problem. As before, the new software image is written into the eMMC main area and updated version information is stored in the RPMB.

But a hacker might try using this same mechanism to downgrade the software on a user's device to take advantage of the security breach or bugs in the previous release. They might try to mimic the procedure that the manufacturer used when pushing out the upgrade—but the hacker would actually push out the earlier, compromised version of the application or operating system.

Software using RPMB to protect itself from a downgrade attack would check for a new, updated version number during the upgrade procedure. If the "new" version number is lower than the one already present in RPMB, the installer would reject the "update."

Due to the nature of RPMB, there is no way for an attacker to change the software version information stored in the RPMB, as that would require access to the secret key.

Example 2 – Preventing Unauthorized Unlock

In this example, RPMB can be used to ensure that only an authorized individual can unlock a device (such as a phone, car, or computer).

First, a user records a PIN code / fingerprint / swipe sequence that will unlock the device. Subsequently, whenever someone attempts to unlock the device, the time of each attempt is recorded in the RPMB. If too many attempts are made within a specified period of time, software controlling the lock will prevent further attempts to unlock the device for a set period of time. If a hacker or thief tries to unlock the device—even using automated tools that would try to inject every possible PIN code until the correct one is found—they will be stopped as soon as the specified number of incorrect unlock trials within the specified period is reached. The hacker can't override the tracking of unlock attempts because the information about the attempts is stored in the RPMB. Unless the hacker stumbles upon the right PIN in the first few attempts, it becomes almost impossible for someone who doesn't know the PIN to unlock the device.

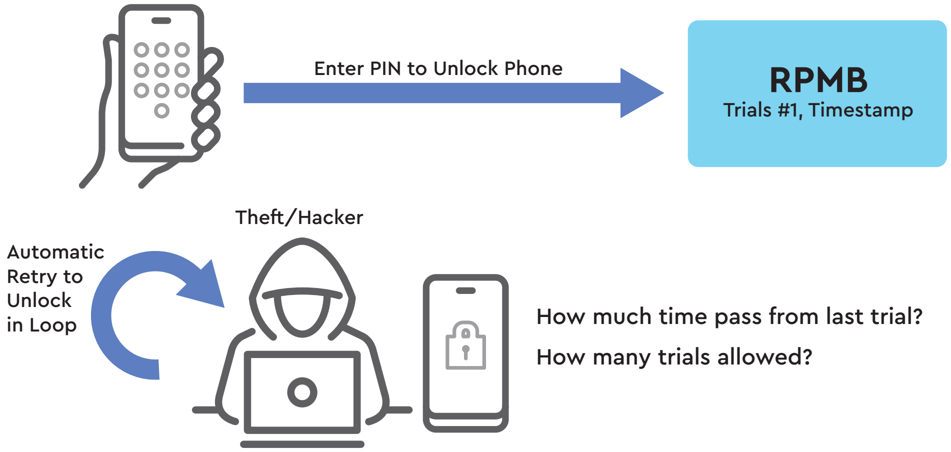
Enter PIN to Unlock Phone
RPMB
Trials #1, Timestamp
Theft/Hacker
Automatic Retry to Unlock in Loop
How much time pass from last trial?
How many trials allowed?

Figure 3: Phone Unlock Protection Scheme

Example 3 – Secured Boot and Secured Write Protect

Prevention of undesired (hacked) code from running on a device starts with an assurance that the very first piece of code that the processor reads from the storage device and executes is legitimate. This initial code—the bootloader —is located on the e.MMC device boot partition, which must be write protected from malware modification.

Prior to eMMC5.1, the permanent write protect mechanism offered the only sure-fire way to protect the boot area, but in addition to protecting the boot area from hackers it also precluded manufacturers from updating those areas if they ever need to. That proved problematic for manufacturers, which led to the development of the secured write protect feature in eMMC5.1.

The secured write protect feature ensures that every change to the write protection configuration itself must be authenticated using the RPMB secret key. The secured write protect mechanism is primarily used to protect the boot code or other sensitive data on the eMMC device from changes or deletion by unauthorized applications.

Conclusion

An e.MMC storage device is more than "just" a code and data storage area. It has built-in features and functionalities that can solve several security concerns which are often raised by device manufacturers.

Understanding the capabilities of the e.MMC device and using them in the overall system design can make products more secure in terms confidentiality, integrity, and availability.

Western Digital.

5601 Great Oaks Parkway
San Jose, CA 95119, USA
www.westerndigital.com

Contact OEMProducts@wdc.com
for all inquiries.