Content Protection for Recordable Media Specification

SD Memory Card Book
SD-SD (Separate Delivery) Part

Intel Corporation
International Business Machines Corporation
Panasonic Corporation
Toshiba Corporation

Revision 0.94
April 6, 2011

This page is intentionally left blank.

Preface

Notice

THIS DOCUMENT IS PROVIDED "AS IS" WITH NO WARRANTIES WHATSOEVER, INCLUDING ANY WARRANTY OF MERCHANTABILITY, NONINFRINGEMENT, FITNESS FOR ANY PARTICULAR PURPOSE, OR ANY WARRANTY OTHERWISE ARISING OUT OF ANY PROPOSAL, SPECIFICATION OR SAMPLE. IBM, Intel, Panasonic, and Toshiba disclaim all liability, including liability for infringement of any proprietary rights, relating to use of information in this specification. No license, express or implied, by estoppel or otherwise, to any intellectual property rights are granted herein.

This document is an intermediate draft and is subject to change without notice. Adopters and other users of this specification are cautioned that products based on it may not be interoperable with the final version or subsequent versions thereof.

Intellectual Property

Implementation of this specification requires license from the 4C Entity, LLC. Note that use of the AES-based technology also requires execution of an addendum to the applicable basic 4C license agreement.

Contact Information

Please address inquiries, feedback, and licensing requests to the 4C Entity, LLC:

- Licensing inquiries and requests should be addressed to 4C-Services@4Centity.com.
- Feedback on this specification should be addressed to 4C-Services@4Centity.com.

The URL for the 4C Entity, LLC web site is http://www.4CEntity.com.

This page is intentionally left blank.

Table of Contents

This page is intentionally left blank.

List of Figures

This page is intentionally left blank.

List of Tables

This page is intentionally left blank.

Chapter 1
Introduction

1. Introduction

1.1 Purpose and Scope

The *Content Protection for Recordable Media Specification* (CPRM) defines a robust and renewable method for protecting content stored on a number of physical media types. The specification is comprised of several "books." The *Introduction and Common Cryptographic Elements* book provides a brief overview of CPRM, and defines cryptographic procedures that are common among its different uses. The *SD Memory Card Book* specifies additional details for using CPRM technology to protect content stored on the SD Memory Card, and on other implementations of protected storage with an interface and security system equivalent to that of the SD Memory Card. Note that such other implementations must not provide any external interface to the memory other than one that adheres to the protocols described in this specification.

The *SD Memory Card Book* consists of the following parts, under the general title *CPRM Specification SD Memory Card Book:*

- *Common Part*
- *SD Application Specific Parts (e.g. SD-Audio, SD-Video, SD-Binding and SD-SD)*

This document is the *SD-SD (Separate Delivery) Part* of the *SD Memory Card Book,* and describes details of CPRM that are specific to the SD-SD format.

The use of this specification and access to the intellectual property and cryptographic materials required to implement it will be the subject of a license. A license authority referred to as the 4C Entity, LLC is responsible for establishing and administering the content protection system based in part on this specification.

SD-SD has the key management mechanism and additional profiles which define content types such as audio, video and electronic book. The key management mechanism treats content keys which encrypt contents. This document provides the protection of content keys and contents. Profile specific usage is provided in the following specifications:

- *SD Memory Card Book SD-SD (Separate Delivery) Audio Profile Part*
- *SD Memory Card Book SD-SD (Separate Delivery) Video Profile Part*
- *SD Memory Card Book SD-SD (Separate Delivery) eBook Profile Part*

1.2 Document Organization

This specification is organized as follows:

- Chapter 1 provides an introduction.
- Chapter 2 lists abbreviations and acronyms used in this document.
- Chapter 3 describes an overview of CPRM for SD-SD to protect SD-SD content
- Chapter 4 describes file configuration in User Data Area
- Chapter 5 describes file configuration in Protected Area
- Chapter 6 describes processes to handle keys and behaviors of devices

- Chapter 7 describes MKB Extension
- Chapter 8 describes MKB Update

1.3 References

This specification shall be used in conjunction with the following documents. When the documents are superceded by an approved revision, the latest revision shall apply.

4C Entity, LLC, *CPRM License Agreement*

4C Entity, LLC, *CPRM Specification: Introduction and Common Cryptographic Elements, Revision 1.1*

4C Entity, LLC, *CPRM Specification: SD Memory Card Book Common Part, Revision 0.97*

4C Entity, LLC, *Content Protection System Architecture White Paper, Revision 0.81*

SD Association, *SD Memory Card Specifications, Part 3: Security Specification, Version 3.00*

SD Association, *SD Memory Card Specifications, Part 15: SD-SD (Separate Delivery) Specification, Version 1.20*

SD Association, *SD Memory Card Specifications, Part 15: Audio Profile Specifications, Addendum to Separate Delivery Specification, Version 1.10*

SD Association, *SD Memory Card Specifications, Part 15: Video Profile Specifications, Addendum to Separate Delivery Specification, Version 1.10*

SD Association, *SD Memory Card Specifications, Part 15: eBook Profile Specifications, Addendum to Separate Delivery Specification, Version 1.0*

1.4 Notation

Except where specifically noted otherwise, this document uses the same notations and conventions for numerical values, operations, and bit/byte ordering as described in the *Introduction and Common Cryptographic Elements* book of this specification.

In addition, this specification uses two other representations for numerical values. Binary numbers are represented as a string of binary (0, 1) digits followed by a suffix ‘b’ (e.g., 1010b). Hexadecimal numbers are represented as a string of hexadecimal (0..9, A..F) digits followed by a suffix ‘h’ (e.g., 3C2h).

Chapter 2
Abbreviations and Acronyms

2. Abbreviations and Acronyms

The following abbreviations and acronyms are used in this document:

AES	Advanced Encryption Standard
AKE	Authentication and Key Exchange
C-CBC	Converted Cipher Block Chaining
C2	Cryptomeria Cipher
CCI	Copy Control Information
CKI	Content Key Information
CKMG	Content Key Manager
CMAC	Cipher-based Message Authentication Code
CMTB	Copy/Move Trigger Bit
CPRM	Content Protection for Recordable Media
K_{c128}	128-bit Content Key
K_{emu}	Expanded Media Unique Key
K_{u128}	128-bit User Key
ID	Identifier
LLC	Limited Liability Company
MKB	Media Key Block
PTB	Playback Trigger Bit
RBP	Relative Byte Position
TBUR	Time Based Usage Rules
TS	Time Stamp
UKA	User Key Area
UKURE	User Key & Usage Rule Entry
UKURE_SRN	UKURE Search Number
UKURMG	User Key & Usage Rule Manager
UKURMGI	User Key & Usage Rule Manager Information
UKURMMG	User Key & Usage Rule Master Manager
UR_C	Usage Rules for Content
UR_U	Usage Rules for User Key

This page is intentionally left blank.

Chapter 3
CPRM for SD-SD

3. CPRM for SD-SD (Separate Delivery)

3.1 Introduction

This chapter specifies details for using CPRM to protect the SD-SD content keys and content. This chapter describes details on using CPRM to realize such as "Move," "Copy," and "Playback" operations.

SD-SD and SD Memory Card formats can be licensed from the SD Association, which also publishes specifications describing them in detail (see the corresponding references in Section 1.3). This chapter assumes that readers are familiar with these formats, as defined in their corresponding specifications.

3.2 Overview

In this section, an overview of SD-SD (Separate Delivery) is described. The CPRM architecture of this application is different from the one of other applications. This application handles content data and content key separately. SD Memory Card is used for storage of keys and encrypted content data itself may be stored in a media other than the same SD Memory Card such as HDD. It is expected that a user would utilize an SD Memory Card which has some content keys with other storage media which has encrypted content data.

This is an example of flow to distribute the SD-SD content from a content provider to a user. The SD-SD content distributed by the content provider is encrypted with a content key. The content provider should give a Content ID to both the SD-SD content and the content key. The encrypted SD-SD content would pass through various distribution channels. Network or broadcast may be used and storage media such as CD-ROM or DVD-ROM may be also used. A user would get the encrypted SD-SD content. When the user desires to playback it, the user would take a transaction to buy the right of accessing the SD-SD content. Likely the transaction would include a payment. Then the user would download the content key to access the SD-SD content and store it to an SD Memory Card. When playing it back, the player will search both the encrypted content data and corresponding content key by matching content IDs, then the player can decrypt the SD-SD content and playback it.

In an SD Memory Card, two kinds of key are stored. One is called a content key which is used to encrypt corresponding content. The other is called a User Key which is used to protect the Content Key values and its usage rules. These keys can be also delivered separately. An example of separate distribution is drawn in Figure 3-1.

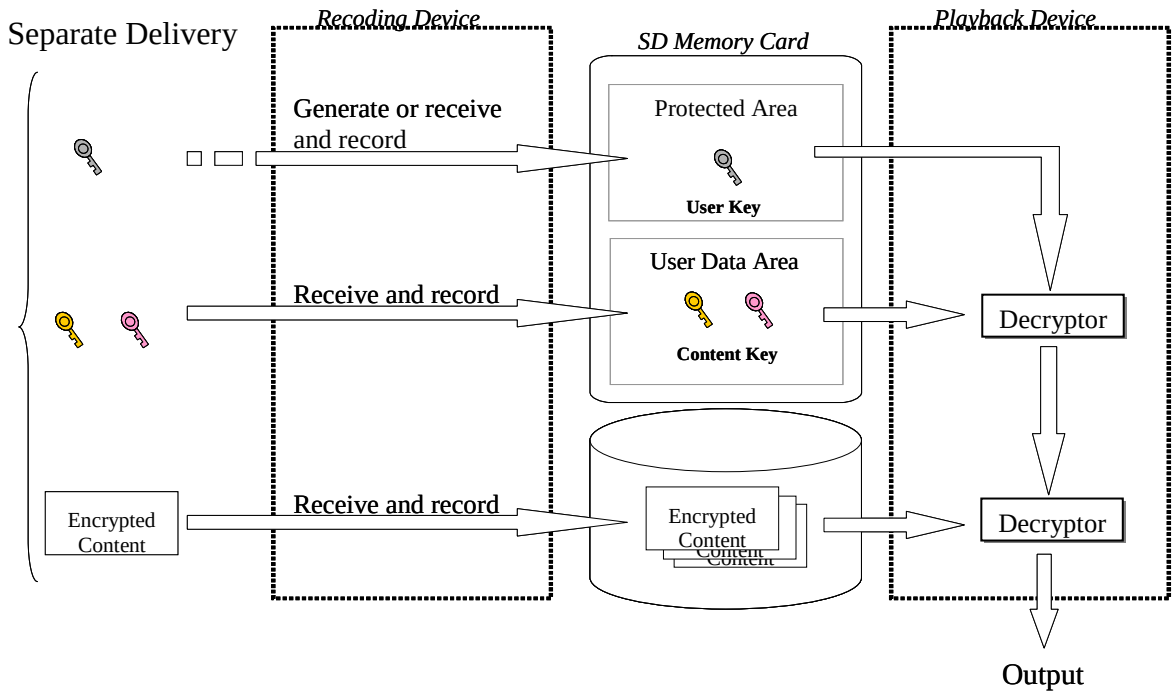


Figure 3-1 An Example of Separate Distribution

When the user is given a copy of SD-SD content, if allowed by the content provider, a Content Key may be copied to a device or another storage. Note that transactions of storing, copying, moving and deleting Content Key do not require storing, copying, moving and deleting the corresponding SD-SD content. Only the Content Key can be treated independently.

Content Keys are used to decrypt various kind of content such as audio or video. The format of content is defined as a profile defined in SD Association specifications and the CPRM protection for content in the profile is defined in each applicable CPRM specification.

3.3 Device Requirements

Regarding the Device Requirements, refer to Section 3.2 of *SD Memory Card Book Common Part*.

3.3.1 Clock for Time-Based Usage Rules

Each CPRM compliant device that supports the "Time-Based Usage Rules" operation described shall have at least one of the following clock functions:

- Clock A: Ordinary clock function, which may be changeable by a user and implemented without tamper resistant manner.

- Clock B: Tamper resistant clock function, which shall be implemented with tamper resistant manner and unchangeable by a user. Clock B shall be accurate to within 5 minutes per month and not be used without synchronization to an accurate time source more than 60 days.

Moreover, Clock B is classified as follows;

- Clock B1: Tamper resistant clock function (Clock B) without adjustment capability to trusted time source. Clock B1 shall be set by the Time Stamp in the Time Stamp file, when Clock B1 is in an "unset state" or "inaccurate state." The details are described in Section 6.13. Here, "unset state" means the state to which a Clock stops (e.g. device is new or device has lost power). A Clock in unset state shall not make the Clock work, until being possible to adjust at appropriate time even if the battery recovers. "Inaccurate state" means the state to which a Clock adjusts at inaccurate time.

- Clock B2: Tamper resistant clock function (Clock B) with adjustment capability to trusted time source. Clock B2 shall be set by the capability, e.g. accessing Web-based, secure time service, or a network clock, when Clock B2 is in an "unset state."

In this specification, it is called Mode "X" when a device has Clock "X" and it operates based on the Clock "X." It is noticed that the accuracy of Clock A is adjusted by the Time Stamp file and it could happen that a user is intentionally able to set forward the time of Clock A and set back the Clock A by the correctly adjusted Time Stamp file. Therefore it is not recommended that the device in Mode A be used for the *time capsule content* which can be played only after a date and time in the future. On the other hand, Mode B requires some robustness for Clock B of a device. The device is designed to prevent the intentional change of time.

A device may have multiple clock functions described above and these clock functions may be switched as long as they are used appropriately. For example, a device may have two clock functions, Clock A and Clock B1, and use the Clock A instead of Clock B1 when the Clock B1 does not work. For another example, a device that works in Mode B2 may switch to Mode B1 when the adjustment capability of the Clock B2 does not work. However the device shall not change the Mode during a process described in Section 6. To the extent, if the device starts to work in Mode B1 at playback, the device works in Mode B1 during the playback and is prohibited to change any other Modes.

3.4 CPRM Components

Regarding the CPRM Components, refer to Section 3.3 of *SD Memory Card Book Common Part*.

3.4.1 System Area

Regarding the System Area, refer to Section 3.3.1 of *SD Memory Card Book Common Part*.

3.4.1.1 Media Key Block (MKB)

In order to protect the User Key and Usage Rules of the SD-SD content, the "MKB for SD-SD" is used. The MKB Number for SD-SD is described in the Supplementary Note of *SD Memory Card Specifications Part 3: Security Specification*.

3.4.2 Hidden Area

Regarding the Hidden Area, refer to Section 3.3.2 of *SD Memory Card Book Common Part*.

3.4.3 Protected Area

Regarding the Protected Area, refer to Section 3.3.3 of *SD Memory Card Book Common Part*.

In the case of the SD-SD specifications, User Keys and Usage Rules for User Key are used instead of Title Keys and Usage Rules described in Section 3.3.3 of *SD Memory Card Book Common Part*, respectively. Thus the Protected Area contains Encrypted User Keys and Encrypted Usage Rules for User Key. The User Key and Usage Rules for User Key are concatenated and encrypted together by a Media Unique Key, which is unique for each SD Memory Card. The encryption process of the User Key and Usage Rules for User Key is described in Section 3.5.1. The Encrypted User Key and Usage Rules for User Key are stored as a file in the Protected Area. The file system of the Protected Area and the detail format of the Encrypted User Key and Usage Rules for User Key are described in Section 4.1.

3.4.3.1 Encrypted User Key

Regarding the Encrypted User Key, refer to Section 3.3.3.1 of *SD Memory Card Book Common Part*.

Note: In the case of CPRM for SD-SD, Encrypted User Keys are stored in the Protected Area instead of Encrypted Title Keys.

3.4.3.2 Encrypted Usage Rules for User Key

Usage Rules for User Key consist of the following information:

"User Key Type": a type of User Key.

"User Key ID": an identifier provided by a Service Provider or a Device.

"Hash Value": a value for protecting Usage Rules for Content.

"Check Value": a fixed value placed at the end of the Usage Rules for User Key. This value is used for detecting whether the User Key and Usage Rules for User Key are unexpectedly altered or not.

The detailed format of Usage Rules for User Key is described in Section 5.5.2.

3.4.3.3 Time Stamp file

In order to assist devices to treat Time-Based Usage Rules (TBUR), SD-SD devices will treat Time Stamp files in the Protected Area of the SD Memory Card. In the case where the CKI includes Time-Based Usage Rules, at most three Time Stamp files, denoted TBUR_A.TS, TBUR_B.TS, and TBUR.TS, will appear in the Protected Area as shown in Figure 5-1.

- /SD_SD/TBUR_A.TS
  TBUR_A.TS file is handled by devices that work in Mode A described in section 3.3. This file is encrypted and written with Mode = 1 specified in chapter 3 of *SD Specifications, Part 3: Security Specification,* that is, accessible only to the SD-SD application.
- /SD_SD/TBUR_B.TS
  TBUR_B.TS is handled by devices that work in Mode B described in section 3.3. This file is also encrypted and written with Mode = 1 specified in chapter 3 of *SD Specifications, Part 3: Security Specification,* that is, accessible only to the SD-SD application.
- /TBUR.TS
  TBUR.TS is handled by not only SD-SD devices but other SD applications (e.g. SD-Video). Handling this Time Stamp file is not mandatory. This file is plaintext and written with Mode = 0 specified in chapter 3 of *SD Specifications, Part 3: Security Specification,* that is, accessible to all SD applications.

The detailed format of each Time Stamp file is described in Section 5.6. Those Time Stamp files are used by both devices supporting C2-CPRM and devices supporting AES-CPRM.

3.4.4 User Data Area

Regarding the User Data Area, refer to Section 3.3.4 of *CPRM Specification: SD Memory Card Book Common Part* book.

3.5 SD-SD Data Encryption and Decryption

3.5.1 SD-SD Data Encryption and Decryption Protocol

SD-SD data encryption and decryption protocol supports C2-CPRM and AES-CPRM described in Section 3.7 of this book.

In the case of C2-CPRM, SD-SD data encryption and decryption protocol is based on the protocol described in Section 3.4 of *SD Memory Card Book Common Part* book of this specification. In the case of SD-SD specifications, User Key and Content Key Information are substituted for Title Key and Content in Section 3.4 of the *SD Memory Card Book Common Part* book of this specification, respectively. Additionally, content data encryption and decryption are described in Profile Part of this specification. So, refer to Section 3.4 of the *SD Memory Card Book Common Part* book of this specification. Figure 3-2 illustrates the process for SD-SD data encryption and decryption.

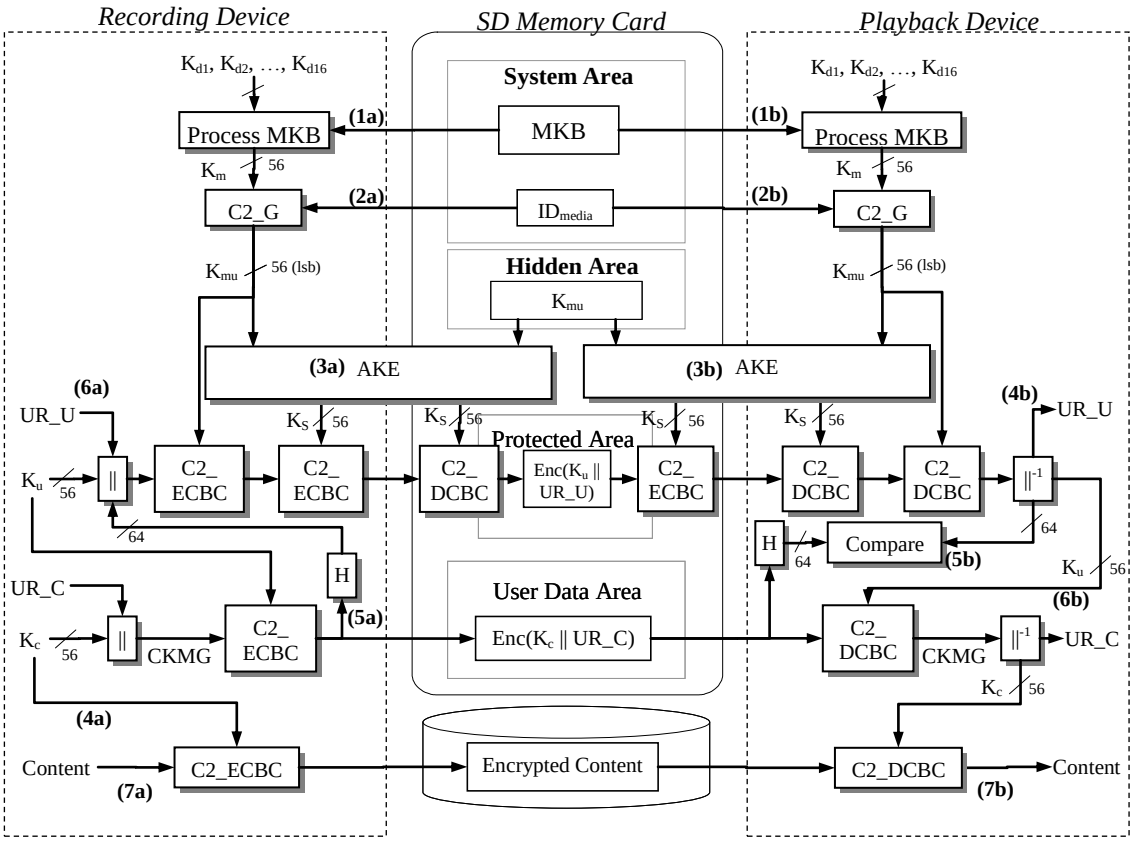


Figure 3-2 C2-CPRM SD-SD Data Encryption and Decryption

The SD Memory Card and the accessing device (Recording Device / Playback Device) authenticate each other, and encrypt or decrypt User Key, Content Key and content as follows:

(1) The accessing device executes Process_MKB

(1a, 1b) Calculate Media Key from MKB using Device Key for MKB (see chapter 3 in the *Introduction and Common Cryptographic Elements* book of this specification)

(2) The accessing device executes the C2_G process

(2a, 2b) Calculate Media Unique Key (K_{mu}) from Media Key (K_m) and Media Identifier (ID_{media}) (see chapter 3 in the *Introduction and Common Cryptographic Elements* book of this specification)

(3) AKE process

(3a, 3b) If the AKE process succeeds, the Session Key (K_s), which is randomly generated in each AKE Process, is shared between Recording Device/Playback Device and the SD Memory Card.

(4a) Content Key Information Encryption Process

The Recording Device concatenates the Content Key (K_c) and UR_C (Usage Rules for Content) and then encrypts them together using the User Key (K_u) with C2_ECBC (the C2 cipher algorithm in C-CBC mode) described in the *Introduction and Common Cryptographic Elements* book of this specification. Note that if K_c and UR_C delivered to the Recording Device are already encrypted in this way, the Recording Device does not perform this encryption step.

(4b) User Key and Usage Rule Entry Encryption Process.

The Encrypted K_u and UR_U are encrypted by the SD Memory Card using the K_s, which is shared at step (3b), using the C2_ECBC. The doubly-encrypted Encrypted K_u and UR_U are sent to the Playback Device. Then, in the Playback Device, they are decrypted by the K_s, which is shared at step (3b), using the C2_DCBC, and those results (Encrypted K_u and UR_U) are decrypted using the K_{mu} with C2_DCBC.

(5a) SD-SD Hash Calculation Process

The Recording Device stores the encrypted K_c and UR_C in the Content Key Manager (CKMG) file and calculates a Hash Value of the updated file with C2_H (C2 hash function) described in the *Introduction and Common Cryptographic Elements* book of this specification. The way of calculating the Hash Value from a CKMG file is specified in Section 3.5.2. The obtained Hash Value is stored in UR_HASH defined in Section 5.5.2. The Recording Device then sends the CKMG file to the SD Memory Card, and stores it in the User Data Area. Note that if the K_c and UR_C are encrypted with K_u whose User Key Type is '1b,' the Recording Device does not perform this hash calculation step.

(5b) Hash verification process

The CKMG file is sent to the Playback Device. The Playback Device calculates a Hash Value of the updated file with C2_H and compares the obtained Hash Value with the value of UR_HASH of UR_U decrypted at step (4b). If two values are different, the Playback Device aborts this process. The way of calculating the Hash Value from a CKMG file is specified in Section 3.5.2. The Playback Device then obtains the encrypted K_c and UR_C from the CKMG file. Note that if the K_c and UR_C are encrypted with K_u whose User Key Type is '1b,' the Playback Device does not perform this hash verification step.

(6a) User Key and Usage Rule Entry Encryption Process

The Recording Device concatenates the K_u and Usage Rules for User Key (UR_U) and then encrypts them together using the K_{mu} with C2_ECBC. Then, the Encrypted K_u and UR_U are further encrypted by the K_s, which is shared at step (3a), using the C2_ECBC. The doubly-encrypted Encrypted K_u and UR_U are sent to the SD Memory Card. In the SD Memory Card, they are decrypted by the K_s, which is shared at step (3a), using the C2_DCBC, and those results (Encrypted K_u and UR_U) are stored in the Protected Area.

(6b) Content Key Information Decryption Process

The Playback Device uses the K_u decrypted at step (4b) to decrypt the encrypted K_c and UR_C using C2_DCBC.

(7a) Content encryption process

The Recording Device shall protect each piece of content by encrypting it using the K_c with C2_ECBC. The Recording Device then sends it to storage (e.g. HDD, SD Memory Card). Note that if the content delivered to the Recording Device is already encrypted in this way, the Recording Device does not perform this encryption step.

(7b) Content decryption process

The Playback Device uses the K_c decrypted at step (6b) to decrypt the encrypted content using C2_DCBC.

Note that the Recording Device does not need to write User Key, Content Key and content to an SD Memory Card at a time. For example, there is a case that a Recording Device stores only one User Key and several Content Keys into an SD Memory Card, but does not store the corresponding content data at the time.

In the case of AES-CPRM, the SD-SD data encryption and decryption protocol is described below. Figure 3-3 illustrates the process for SD-SD data encryption and decryption.

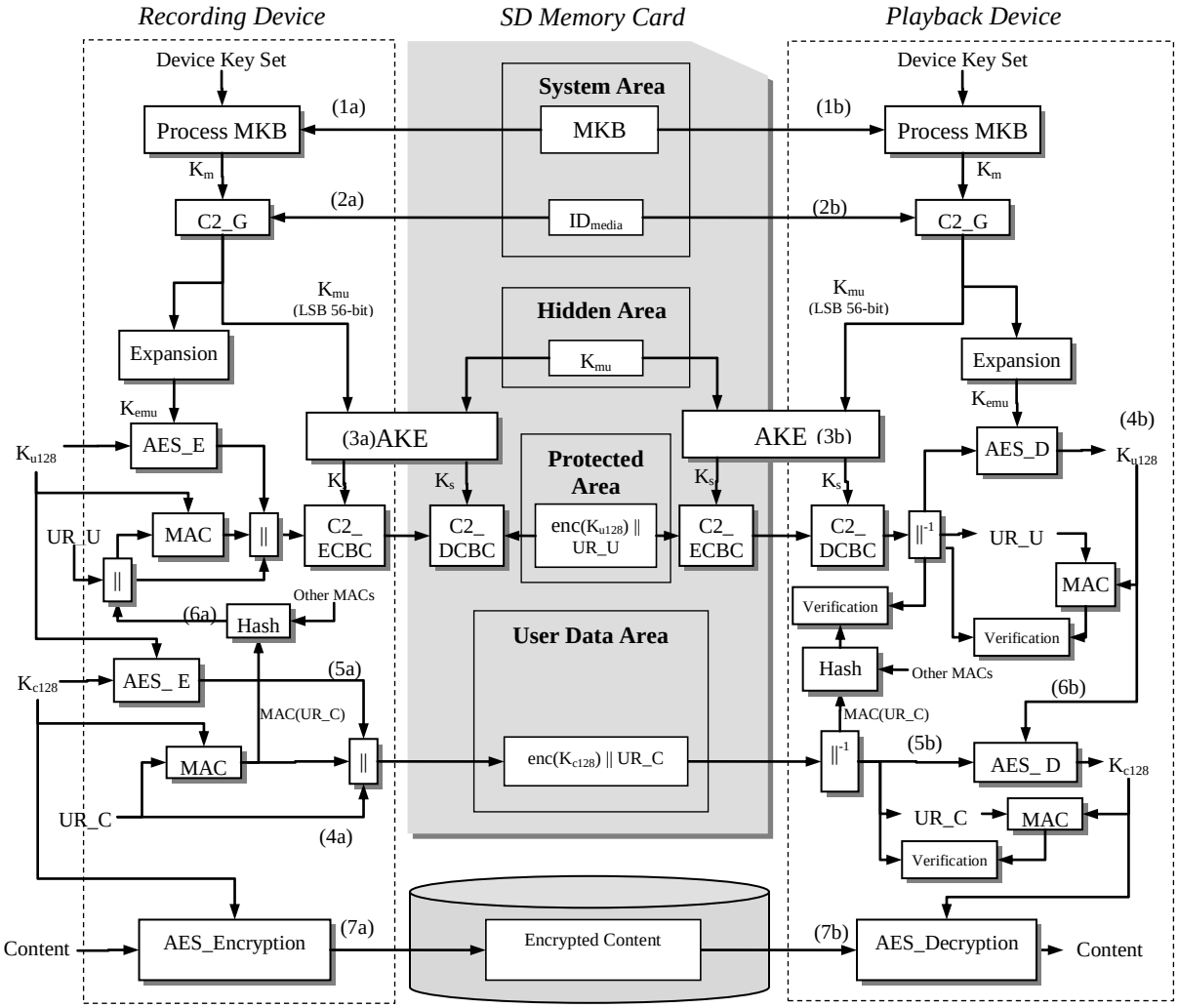


Figure 3-3 AES-CPRM SD-SD Data Encryption and Decryption

The SD Memory Card and the accessing device (Recording Device / Playback Device) authenticate each other, and encrypt or decrypt the 128-bit User Key, 128-bit Content Key and content as follows:

(1) The accessing device executes Process_MKB

(1a, 1b) Calculate Media Key from MKB using Device Key for MKB (see Chapter 3 in the *Introduction and Common Cryptographic Elements* book of this specification)

(2) The accessing device executes the C2_G process and expansion

(2a, 2b) Calculate Media Unique Key (K_{mu}) from Media Key (K_m) and Media Identifier (ID_{media}) (see Chapter 3 in the *Introduction and Common Cryptographic Elements* book of this specification)

K_{mu} is used for authentication and key exchange between device and the SD Memory Card.

Expanded Media Unique Key (K_{emu}) is calculated in the following formula:

$$K_{emu} = AES\_G(K_{mu} \parallel \text{Secret Padding 1, Secret Padding 2})$$

Secret Padding 1 (72 bits) and Secret Padding 2 (128 bits) are provided in separate confidential book(s) corresponding to each profile of SD-SD (e.g. *SD Memory Card Book SD-SD (Separate Delivery) eBook Profile Confidential Part* of this specification). These secret values shall be treated as Confidential Information.

(3) AKE process

(3a, 3b) If the AKE process succeeds, the Session Key (K_s), which is randomly generated in each AKE Process, is shared between Recording Device/Playback Device and the SD Memory Card.

(4a) Content Key Information Encryption Process

The Recording Device creates an entry from 128-bit Content Key and UR_C. As CKI Encryption Process, the 128-bit Content Key is encrypted with the 128-bit User Key by using AES_E. The encrypted 128-bit Content Key is divided and stored into CK128-1 in CKA and CK128-2 in UR_C. After CKI Encryption Process, the UR_CHECK of UR_C is set to the result of CMAC(K_{c128}, the high 40 bytes of UR_C), where CMAC is described in the *Introduction and Common Cryptographic Elements* book of this specification. Note that if K_{c128} and UR_C delivered to the Recording Device are already encrypted in this way, the Recording Device does not perform this step. The Recording Device stores the encrypted K_{c128} and UR_C in the Content Key Manager (CKMG) file and sends the CKMG file to the SD Memory Card to store it in the User Data Area.

(4b) User Key and Usage Rules Entry Decryption Process

The Playback Device reads the entry of 128-bit User Key and Usage Rules for 128-bit User Key from the SD Memory Card. The SD Memory Card encrypts the Encrypted K_{u128} and UR_U with the K_s which is shared at step (3b) by using the C2_ECBC. The encrypted data are sent to the Playback Device. The Playback Device decrypts them with the K_s shared at step (3b) by using the C2_DCBC. The result consists of the Encrypted K_{u128} and UR_U. As UKURE Decryption Process, the Encrypted K_{u128} is decrypted using the K_{emu} with AES_D. K_{u128} is calculated from concatenating the value in UK128-1 in UKA as the high 8 bytes and the value in UK128-2 in UR_U as the low 8 bytes. After UKURE Decryption Process, the Usage Rules for User Key is verified by comparing the value of UR_CHECK in UR_U and the result of CMAC(K_{u128}, the high 40 bytes of UR_U). If those values do not match, the Playback Device shall abort this decryption process.

(5a) SD-SD Hash Calculation Process

The Recording Device calculates a Hash Value of the CKMG file with AES_H (AES hash function) described in the *Introduction and Common Cryptographic Elements* book of this specification. The way of calculating the Hash Value from a CKMG file is specified in Section 3.5.2. The obtained Hash Value is stored in UR_HASH of UR_U defined in Section 5.5.2. Note that if the K_{c128} and UR_C are encrypted with K_{u128} whose User Key Type is '1b,' the Recording Device does not perform this hash calculation step.

(5b) Hash verification process

The Playback Device reads the CKMG file from the SD Memory Card. The Playback Device calculates a Hash Value of the CKMG file with AES_H by using SD-SD Hash Calculation and compares the obtained Hash Value with the value of UR_HASH of UR_U obtained at step (4b). If two values are different, the Playback Device aborts this decryption process. The way of calculating the Hash Value from a CKMG file is specified in Section 3.5.2. The Playback Device then obtains the encrypted K_{c128} and UR_C from the CKMG file. Note that if the K_{c128} and UR_C are encrypted with K_{u128} whose User Key Type is '1b,' the Playback Device does not perform this hash verification step.

(6a) User Key and Usage Rules Entry Encryption Process.

The Recording Device creates an entry of 128-bit User Key and UR_U. As UKURE Encryption Process, the 128-bit User Key is encrypted with the Expanded Media Unique Key calculated in step (2) by using AES_E. The encrypted 128-bit User Key is divided and stored separately in UK128-1 in UKA and UK128-2 in UR_U. After the UKURE Encryption Process, the UR_CHECK of UR_U is set to the result of CMAC(K_{u128}, the high 40 bytes of UR_U). When the Recording Device stores the entry into the SD Memory Card, the Encrypted K_{u128} and UR_U are further encrypted by the K_s shared at step (3a) by using the C2_ECBC. The encrypted data are sent to the SD Memory Card. In the SD Memory Card, they are decrypted with the K_s which is shared at step (3a) by using the C2_DCBC and those results (Encrypted K_{u128} and UR_U) are stored in the Protected Area.

(6b) Content Key Information Decryption Process.

The Playback Device reads the CKI including 128-bit Content Key and Usage Rules for Content from the CKMG file. As CKI Decryption Process, the Encrypted K_{c128} is decrypted using the K_{u128} with AES_D. The K_{c128} is calculated from concatenating the value in CK128-1 in CKA as the high 8 bytes and the value of CKI128-2 in UR_C as the low 8 bytes. After the CKI Decryption Process, the Usage Rules for Content is verified by comparing the value of UR_CHECK in UR_C and the result of CMAC(K_{c128}, the high 40 bytes of UR_C). If those values do not match, the Playback Device shall abort this decryption process.

(7a) Content encryption process

The Recording Device shall protect each piece of content by encrypting it using the K_{c128}. Detail of encryption is described in each CPRM for SD-SD profile part. The Recording Device then sends it to storage (e.g. HDD, SD Memory Card). Note that if the content delivered to the Recording Device is already encrypted in this way, the Recording Device does not perform this encryption step.

(7b) Content decryption process

The Playback Device uses the K_{c128} decrypted at step (6b) to decrypt the encrypted content. Detail of decryption is described in each CPRM for SD-SD profile part.

Note that the Recording Device does not need to write 128-bit User Key, 128-bit Content Key and content data to an SD Memory Card at a time. For example, there is a case that a Recording Device stores only one 128-bit User Key and several 128-bit Content Keys into an SD Memory Card, but does not store the corresponding content data at the time.

3.5.2 Calculation of Hash Value for Content Key Manager

This section shows details of SD-SD Hash Calculation Process in step (5a) of Section 3.5.1. There are two algorithms for calculation of Hash Value. Regarding C2-CPRM and AES-CPRM, refer to Section 3.7.

3.5.2.1 Calculation of Hash Value for C2-CPRM

A Hash Value for Content Key Manager (CKMG) file is calculated with C2_H (the C2 hash function) described in the *Introduction and Common Cryptographic Elements* book of this specification. The length of the Hash Value is 64-bit. Depending on CKI Used flags (A CKI Used flag [j] indicates whether corresponding CKI #j+1 exists, which is defined in *SD Memory Card Specifications, Part 15 SD-SD (Separate Delivery) Specification* issued by the SD Association), an input D to C2_H is obtained by concatenating the last 8 bytes of all used encrypted CKI in ascending order of subscript. The CKI is encrypted in CKI Encryption Process described in Chapter 6. .The following sentences in C code like language illustrates how to set the input D and calculate an output O.

D = NULL

```
i = 0
WHILE (i < 100) {
        IF (CKI Used flag [i] == '1b') D = D || the last 8-byte of encrypted CKI # i+1
        i = i + 1
}
if (D == NULL) O = '00000000h' else O = C2_H(D)
```

Note that NULL || s is equal to s (s: a string)

For example, when CKI Used flag [0], CKI Used flag [1] and CKI Used flag [5] are '1b' and the other flags are set to 0b, the input D is

the last 8-byte of encrypted CKI #1 || the last 8-byte of encrypted CKI #2 || the last 8-byte of encrypted CKI #6.

The last 8 bytes of encrypted CKI is equivalent to the UR_CHECK encrypted with C2_ECBC. As for the detail of CKMG file, refer to *SD Memory Card Specifications, Part 15 SD-SD (Separate Delivery) Specification* issued by the SD Association.

3.5.2.2 Calculation of Hash Value for AES-CPRM

A Hash Value for Content Key Manager (CKMG) file is calculated with AES_H (the AES hash function) described in the *Introduction and Common Cryptographic Elements* book of this specification. The length of the Hash Value is the high 8 bytes of the result of AES_H. Depending on CKI Used flags (A CKI Used flag [j] indicates whether corresponding CKI #j+1 exists, which is defined in *SD Memory Card Specifications, Part 15 SD-SD (Separate Delivery) Specification* issued by the SD Association), an input D to AES_H is obtained by concatenating the last 8 bytes of all used encrypted CKI in ascending order of subscript. The CKI is encrypted in CKI Encryption Process described in Chapter 6. The following sentences in C code like language illustrates how to set the input D and calculate an output O.

```
D = NULL
i = 0
WHILE (i < 100) {
        IF (CKI Used flag [i] == '1b') D = D || the last 8-byte of encrypted CKI # i+1
        i = i + 1
}
IF (D == NULL) O = '00000000h' ELSE O = AES_H(D)
```

Note that NULL || s is equal to s (s: a string)

For example, when CKI Used flag [0], CKI Used flag [1] and CKI Used flag [5] are '1b' and the other flags are set to 0b, the input D is

the last 8-byte of encrypted CKI #1 || the last 8-byte of encrypted CKI #2 || the last 8-byte of encrypted CKI #6.

The last 8-byte of encrypted CKI is specified as the UR_CHECK field and remains in the clear. As for the detail of CKMG file, refer to *SD Memory Card Specifications, Part 15 SD-SD (Separate Delivery) Specification* issued by the SD Association.

3.5.3 User Key Type

SD-SD specification prepares two types of User Key. The type describes whether or not a device executes the Hash Calculation Process described in Section 3.5.2. When the value of User Key Type is '0b,' all CKIs encrypted with the User Key are handled with the hash calculation during all of processes described in Chapter 6. On the other hand, when User Key Type is '1b,' all CKIs encrypted with the User Key are handled without the hash calculation during all of processes. Regarding usage rule, content encrypted with the User Key whose type is '1b' have some restrictions. Use of dynamic Usage Rules for Content is prohibited. As shown in Table 3-1, some fields in Usage Rules for Content are restricted when User Key Type is '1b.' The device shall not set each field to other than the value on this table.

Table 3-1 Restriction of Usage Rules for Content

Field Name or parameters	Restriction
Current Move Control Counter in UR_MCCNTRL	00b
Copy Count Control Counter in UR_MCCNTRL	0000b or 1111b
UR_C_STRTDATE	the value given at the recording
UR_C_ENDDATE	the value given at the recording
UR_C_P_CNT	FFFFh
UR_SPAN	Not applicable
UR_SD_ID	the value given at the recording

These parameters are not updated once the Content Key is stored. Note that a device cannot securely erase a Content Key from CKMG file encrypted with User Key whose type is 1b.

3.6 Accessing the Protected Area

Regarding Accessing the Protected Area, refer to Section 3.5 of *SD Memory Card Book Common Part*.

3.7 C2-CPRM and AES-CPRM

CPRM for SD-SD specification supports both C2-CPRM and AES-CPRM. In the case of C2-CPRM, all keys such as Content Key and User Key are in 56-bit length. The cipher algorithm used in this case is C2 defined in *Introduction and Common Cryptographic Elements* book of this specification. In this book, Content Key and User Key in C2-CPRM are denoted as (1) C2 Content Key and C2 User Key or (2) merely Content Key and User Key. In the case of AES-CPRM, Content Key and User Key are in 128-bit length. The cipher algorithm

used is AES defined in *Introduction and Common Cryptographic Elements* book of this specification. The Content Key and User Key are denoted as 128-bit Content Key and 128-bit User Key.

The 128-bit Content Key shall not be protected by any C2 User Key which is in 56 bits length but be protected by 128-bit User Key. To comply with this rule, it is not allowed that 128-bit Content Key and C2 Content Key are stored in the same Service Directory which is defined in *SD Memory Card Specifications, Part 15: SD-SD (Separate Delivery) Specification*.

Chapter 4
CPRM Related Data in User Data Area

4. CPRM Related Data in User Data Area

4.1 File System of the User Data Area for SD-SD

File system of SD-SD is described in *SD Memory Card Specifications, Part 15 SD-SD (Separate Delivery) Specification*

In this specification, some terms in the SD Association are used. The following terms are defined in the SD Association and used as the same meanings.

Content Key Manager (CKMG): the manager for Content Keys in User Data Area.

User Key Information (UKI): the field in the User Key Manager for storing the information regarding the corresponding User Key.

User Key Manager: the manager in User Data Area for access to User Keys in Protected Area.

USER_KEY_TYPE: the field in the User Key Information for storing the User Key Type.

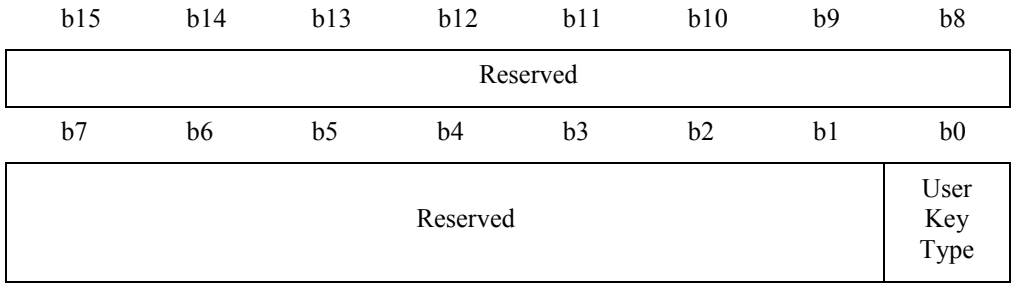
Encrypted Content Key and Usage Rules (ECKUR): Content Key and Usage Rule encrypted with corresponding a User Key. ECKUR is in Content Key Manager

4.2 User Key Type defined in User Key Information

The User Key Type describes whether or not the User Key requires the Hash Value.

(RBP 0 to 1) USER_KEY_TYPE

This field describes the User Key Type.



User Key Type ... 0b: User Key requires the Hash Value

1b: User Key does not require the Hash Value

4.3 Encrypted Content Key and Usage Rule (ECKUR)

As shown in Table 4-1, an ECKUR field contains Content Key Area (CKA) and Usage Rules for Content (UR_C) of the corresponding encrypted content.

Table 4-1 ECKUR

(Description order)

RBP	Field Name	Contents	Number of bytes
0 to 7	CKA	Content Key Area	8 bytes
8 to 63	UR_C	Usage Rules for Content	56 bytes
Total			64 bytes

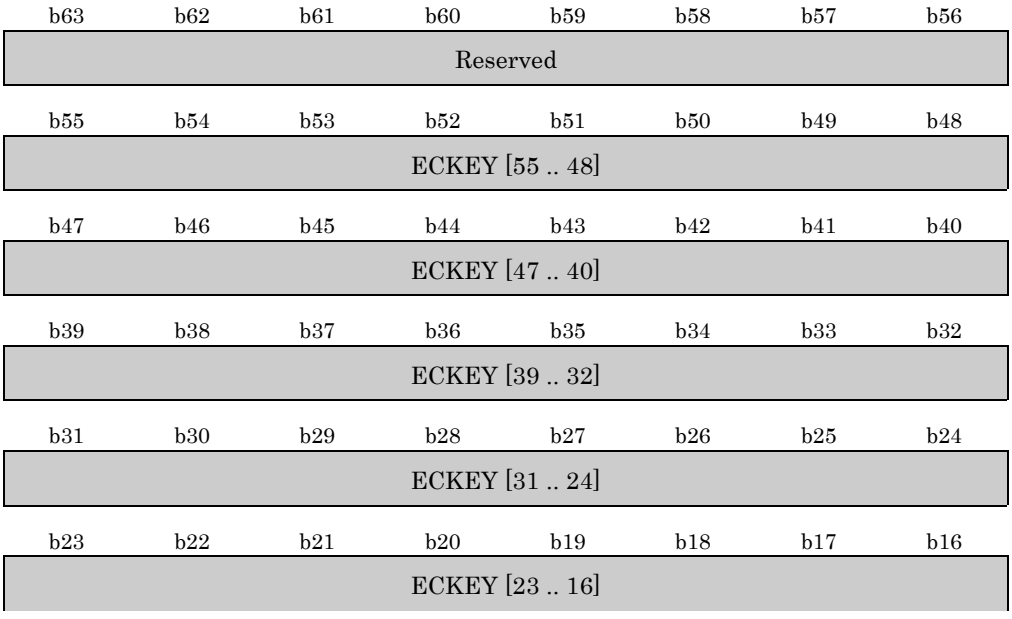
4.3.1 Content Key Area (CKA)

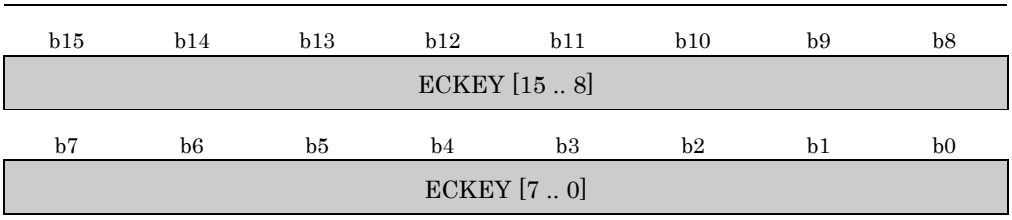
In the case of C2-CPRM, as shown in Table 4-2, CKA contains ECKEY field. This field describes the Content Key of the corresponding encrypted content.

Table 4-2 CKA for C2-CPRM

(Description order)

RBP	Field Name	Contents	Number of bytes
0	Reserved	Reserved	1 byte
1 to 7	ECKEY	Content Key	7 bytes
Total			8 bytes





ECKEY ... Stores the Content Key.

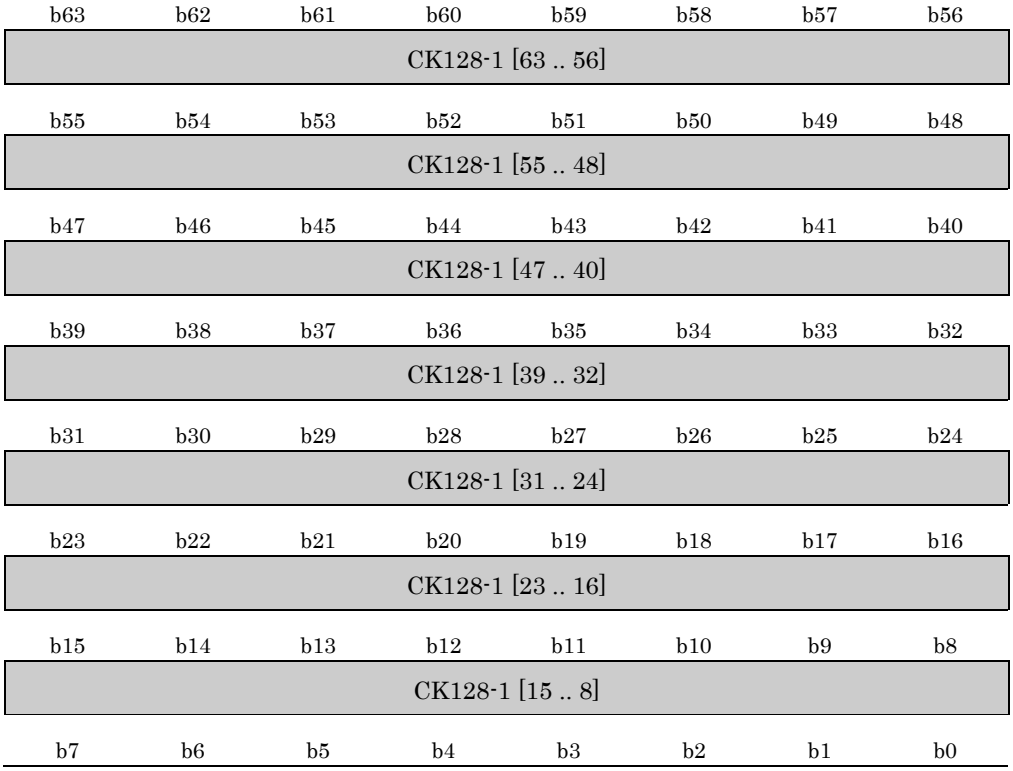
All reserved bits (from b56 to b63) shall be set to '0b.' For forward compatibility, the Playback Device compliant with the current revision of this book shall ignore these reserved bits.

In the case of AES-CPRM, as shown in Table 4-3, CKA contains CK128-1 field. This field describes the high 8 bytes of 128-bit Content Key of the corresponding encrypted content. The low 8 bytes of 128-bit Content Key is stored in CK128-2 field in the Usage Rules for Content.

Table 4-3 CKA for AES-CPRM

(Description order)

RBP	Field Name	Contents	Number of bytes
0 to 7	CK128-1	The high 8 bytes of 128-bit Content Key	8 bytes
Total			8 bytes



CK128-1 [7 .. 0]

CK128-1 ... Stores the high 8 bytes of 128-bit Content Key.

4.3.2 Usage Rules for Content (UR_C)

As shown in Table 4-4, Usage Rules for Content (UR_C) contains Trigger Bits for Playback, Trigger Bits for Copy/Move, Initial Move Control Information, Current Move Control Information, Copy Count Control Information,, Current Start Date of Playback Period, Current End Date of Playback Period, Current Playback Counter, Current TBRU Conditions, Initial Start Date of Playback Period, Initial End Date of Playback Period, Initial Playback Counter, Initial TBUR Conditions, Playback Span, Service Defined ID and Check Value.

Table 4-4 UR_C

(Description order)

RBP	Field Name	Contents	Number of bytes
0	UR_TRIGGER	Trigger Bits for Playback / Trigger Bits for Copy/Move	1 byte
1	UR_MCCNTRL	Initial Move Control Information / Current Move Control Information / Copy Count Control Information	1 byte
2 to 5	Reserved	Reserved	4 bytes
6 to 15	Reserved for Profiles	Reserved for Profiles	10 bytes
16 to 18	UR_C_STRTDATE	Current Start Date of Playback Period	3 bytes
19 to 21	UR_C_ENDDATE	Current End Date of Playback Period	3 bytes
22 to 23	UR_C_P_CNT	Current Playback Counter	2 bytes
24	UR_C_TBUR_CDT	Current TBRU Conditions	1 byte
25	Reserved for CFG	Reserved for Current Field Group	1 byte
26 to 28	UR_I_STRTDATE	Initial Start Date of Playback Period	3 bytes
29 to 31	UR_I_ENDDATE	Initial End Date of Playback Period	3 bytes
32 to 33	UR_I_P_CNT	Initial Playback Counter	2 bytes
34	UR_I_TBUR_CDT	Initial TBUR Conditions	1 byte
35	Reserved for IFG	Reserved for Initial Field Group	1 byte
36	Reserved	Reserved	1 byte
37 to 39	UR_SPAN	Playback Span	3 bytes
40 to 47	UR_SD_ID or CK128-2	Service Defined ID or the low 8 bytes of 128-bit Content Key	8 bytes
48 to 55	UR_CHECK	Check Value	8 bytes

Total	56 bytes

In the following definition of Usage Rule fields, the assigned values are effective only when the ECKUR is used. When the ECKUR is not used, no specific value is assigned to each Usage Rule field.

The usage of Reserved for Profiles is described in a specification for each profile. In this book, it is only described how SD-SD devices treat this field in Recording, Copy and Move processes defined in Chapter 6.

All bits in Reserved for CFG and Reserved for IFG shall be set to '0b.' For forward compatibility, the Playback Device compliant with the current revision of this book shall ignore all the bits except for Copy Process.

All other reserved bits shall be set to '0b.' For forward compatibility, the Playback Device compliant with the current revision of this book shall ignore the aforementioned all other reserved bits.

(RBP 0) UR_TRIGGER

This field describes Trigger Bits for Playback and Trigger Bits for Copy/Move.

b7	b6	b5	b4	b3	b2	b1	b0
Trigger Bits for Playback				Trigger Bits for Copy/Move			
PTB for TBUR	PTB for AES	PTB for future use		CMTB for AES	CMTB for future use		

Playback Trigger Bit for Time Based Usage Rules (PTB for TBUR) ...

0b: The content does not include Time Based Usage Rules

1b: The content includes Time Based Usage Rules. Accessing devices which do not support Clock A or Clock B described in Section 3.3.1 of this book are not allowed to playback the content encrypted by this Content Key.

Playback Trigger Bit for Playback for AES (PTB for AES) ...

0b: The content is encrypted by C2 Content Key.

1b: The Content is encrypted by 128-bit Content Key. Accessing devices which do not support AES-CPRM are not able to playback the content encrypted by this Content Key.

Playback Trigger Bit for future use (PTB for future use) ...

00b: Accessing devices conforming to this specification can control the Playback Process.

01b~11b: Accessing devices conforming to this specification is not allowed to complete the Playback Processes.

Accessing devices conforming to this specification shall always set this Trigger Bits value to ‘0000b,’ ‘1000b,’ ‘0100b’ or ‘1100b’ as appropriate, when writing an encrypted content to an SD Memory Card.

In a future version, the Usage Rules may be expanded, or other information for controlling Playback Processes may be added. Accessing devices of the future version may process the new information for controlling Playback Processes correctly when PTB for future use is set to the value except ‘00b.’

Copy/Move Trigger Bit for AES (CMTB for AES) ... 0b: The content is encrypted by C2 Content Key.

1b: The Content is encrypted by 128-bit Content Key. Accessing devices which do not support AES-CPRM are not able to copy/move the content encrypted by this Content Key.

Copy/Move Trigger Bit for future use (CMTB for future use) ... 000b: Accessing devices conforming to this specification can control the Copy/Move Processes

001b~111b: Accessing devices conforming to this specification is not allowed to complete the Copy/Move Processes.

Accessing devices conforming to this specification shall always set this Trigger Bits value to ‘0000b’ or ‘1000b’ as appropriate, when writing an encrypted content to an SD Memory Card.

In a future version, the Usage Rules may be expanded, or other information for controlling Copy/Move Processes may be added. Accessing devices of the future version shall process the new information for controlling Copy/Move Processes correctly when CMTB for future use is set to the value except ‘000b.’

Note that there is restriction of setting of Trigger Bits. Content Provider or Service Provider shall set one of these combinations described in Table 4-5.

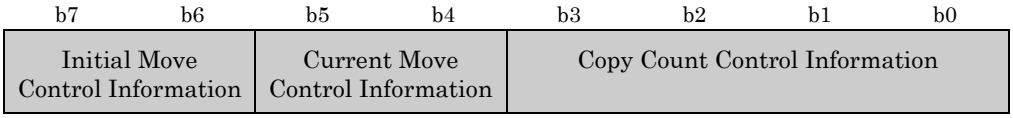
Table 4-5 Combination of Trigger Bits

Content	Field	Value

SD-SD Content protected by C2-CPRM	Without Time Based Usage Rules	Trigger Bits for Playback	0000b
		Trigger Bits for Copy/Move	0000b
	With Time Based Usage Rules	Trigger Bits for Playback	1000b
		Trigger Bits for Copy/Move	0000b
SD-SD Content protected by AES-CPRM	Without Time Based Usage Rules	Trigger Bits for Playback	0100b
		Trigger Bits for Copy/Move	1000b
	With Time Based Usage Rules	Trigger Bits for Playback	1100b
		Trigger Bits for Copy/Move	1000b

(RBP 1) UR_MCCNTRL

This field describes the Initial Move Control Information, Current Move Control Information, and Copy Count Control Information.



Initial Move Control Information ... 00b: Move is never permitted.

01b: Move is permitted once.

11b: Move is permitted unlimited times.

10b: Reserved.

The Initial Move Control Information is set when the corresponding content is distributed. It never changes even when the content is moved. This field is inherited to a replicated content when copying.

Current Move Control Information ... 00b: Move is not permitted.

01b: Move is permitted once.

11b: Move is permitted unlimited times.

10b: Reserved.

The Current Move Control Information changes when the corresponding content is moved. As for the details how conforming devices shall change this field, refer to the Move Process described in Chapter 6.

Copy Count Control Information ... 0000b: Copy is not permitted.

0001b~1110b: Copy is permitted specified times.

1111b: Copy is permitted unlimited times.

The Copy Count Control Information changes when the corresponding content is copied. As for the details how conforming devices shall change this field, refer to the Copy Process described in Chapter 6.

(RBP 6 to 15) Reserved for Profiles

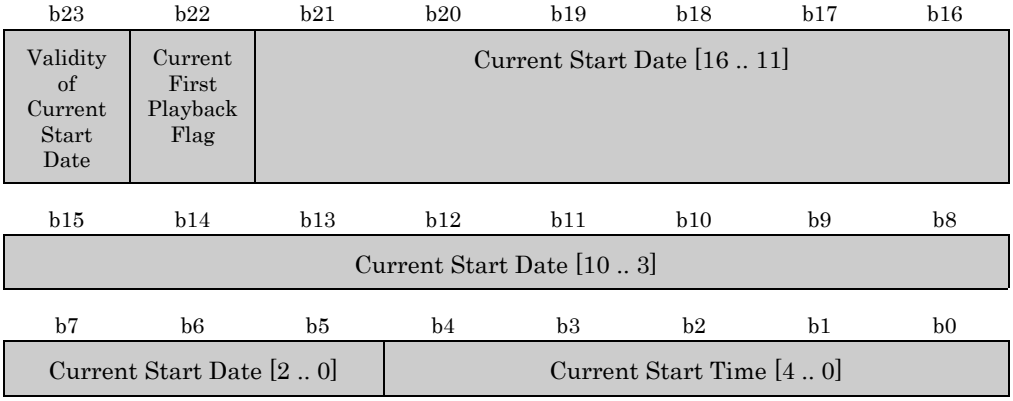
This field describes Trigger Bits for Profile Processes, Current Fields Group for Profile, Initial Fields Group for Profile. The others are reserved. The usage of this field is defined in a specification for each profile.

(Description order)

RBP	Field Name	Contents	Number of bytes
0	UR_PF_TRIGGER	Trigger Bits for Profile Processes	1 byte
1 to 2	UR_PF_CURRENT	Current Fields Group for Profile	2 bytes
3 to 4	UR_PF_INITIAL	Initial Fields Group for Profile	2 bytes
5 to 9	Reserved	Reserved	5 bytes
Total			10 bytes

(RBP 16 to 18) UR_C_STRTDATE

This field describes the current start date and time of permitted playback period.



Validity of Current Start Date ... 0b: The current start date of permitted playback period is not specified.

1b: The current start date of permitted playback period is specified.

Current First Playback Flag ... This field describes whether or not the first playback has been performed when the playback span is specified.

0b: First playback has not been performed.

1b: First playback has been performed and so the Current Start Date and the Current End Date have already been fixed.

Current Start Date ... This field describes the current start date in Modified Julian Date format.

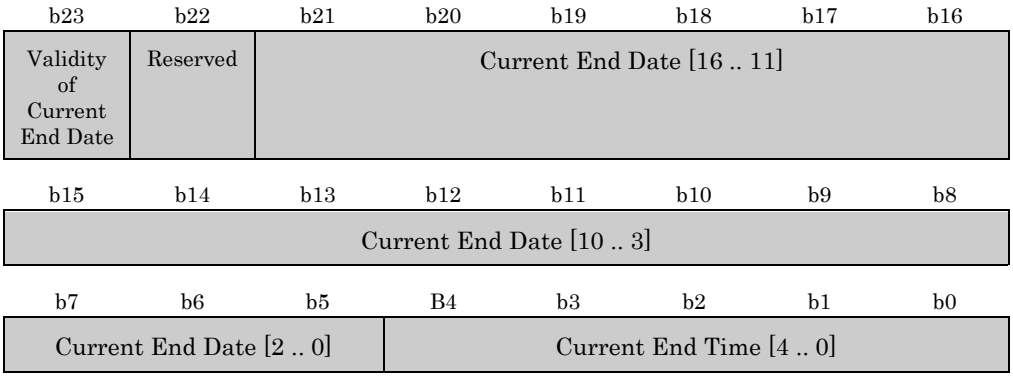
Current Start Time ... This field describes the current start time by the hour.

0~23: Hours from midnight.

Others: Reserved.

(RBP 19 to 21) UR_C_ENDDATE

This field describes the current end date and time of permitted playback period.



Validity of Current End Date ... 0b: The current end date of permitted playback period is not specified.

1b: The current end date of permitted playback period is specified.

Current End Date ... This field describes the current end date in Modified Julian Date format.

Current End Time ... This field describes the current end time by the hour.

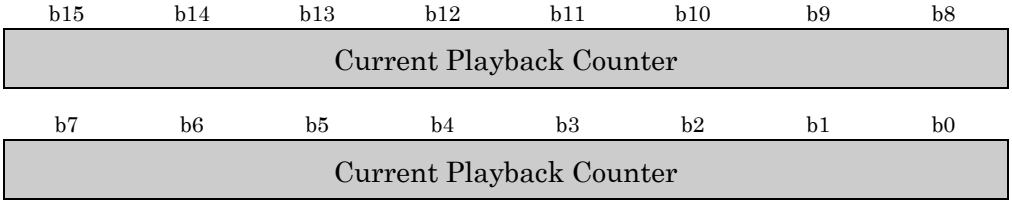
0~23: Hours from midnight.

Others: Reserved.

When the corresponding content is played for the first time, UR_C_STRTDATE and UR_C_ENDDATE may change according to the validity of the playback span. These fields shall not be inherited to a replicated content when copying. For details how conforming devices shall change these fields, refer to the Playback Process described in Section 6.12.

(RBP 22 to 23) UR_C_P_CNT

This field describes the current permitted playback count.



Current Playback Counter ... 0000h: Playback is never permitted.

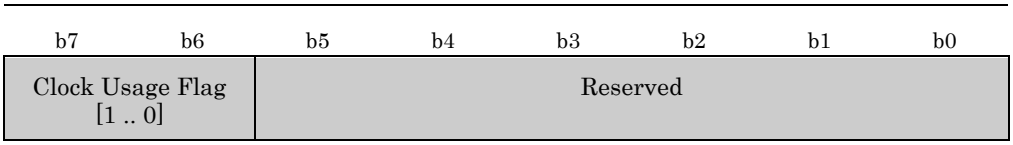
... 0001h~FFFEh: Playback is permitted specified times.

... FFFFh: Playback is permitted unlimited times.

This field may change when the corresponding content is played. This field shall not be inherited to a replicated content when copying. For details on how conforming devices shall change this field, refer to the Playback Process described in Section 6.12.

(RBP 24) UR_C_TBUR_CDT

This field shows current conditions of the CKI with Time-Based Usage Rules in the CKI.



Clock Usage Flag ... This shows a clock mode restriction of playback the content. How to check this field is described in Section 6.12

00b: Devices in Mode A, Mode B1 and Mode B2 are allowed to playback the content.

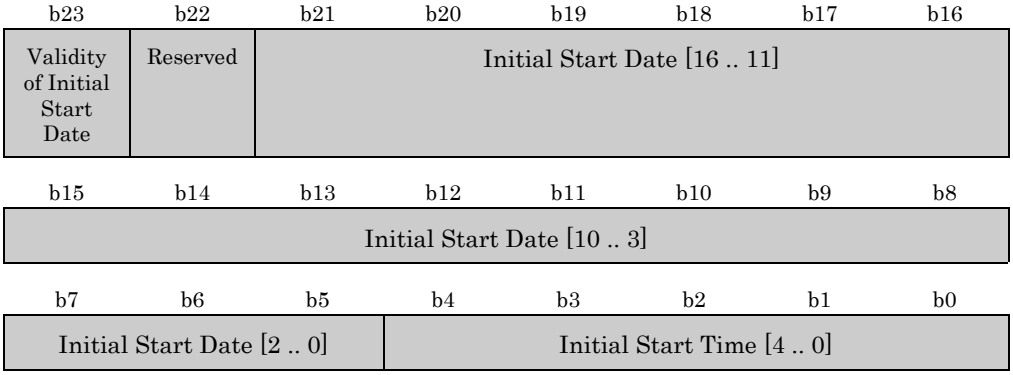
01b: Reserved.

10b: Devices in Mode B1 and Mode B2 are allowed to playback the content. Devices in Mode A are not allowed to playback the content.

11b: Devices in Mode B2 are allowed to playback the content. Devices in Mode A and Mode B1 are not allowed to playback the content.

(RBP 26 to 28) UR_I_STRTDATE

This field describes the initial start date and time of permitted playback period.



Validity of Initial Start Date ... 0b: The initial start date of permitted playback period is not specified.

1b: The initial start date of permitted playback period is specified.

Initial Start Date ... This field describes the initial start date in Modified Julian Date format.

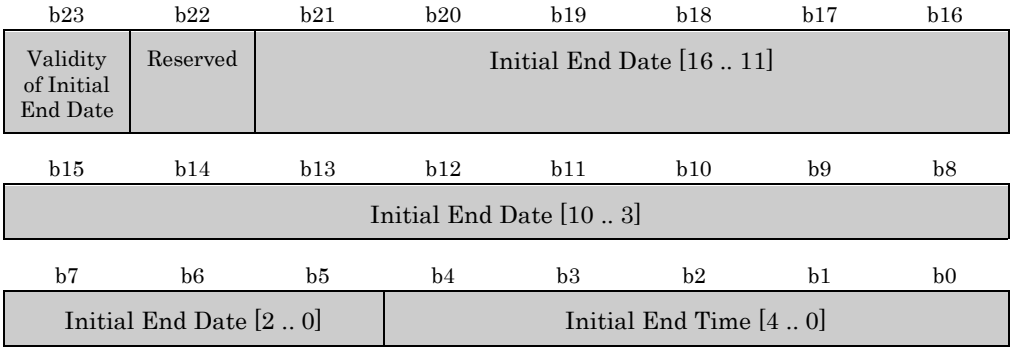
Initial Start Time ... This field describes the initial start time by the hour.

0~23: Hours from midnight.

Others: Reserved.

(RBP 29 to 31) UR_I_ENDDATE

This field describes the initial end date and time of permitted playback period.



Validity of Initial End Date ... 0b: The initial end date of permitted playback period is not specified.

1b: The initial end date of permitted playback period is specified.

Initial End Date ... This field describes the initial end date in Modified Julian Date format.

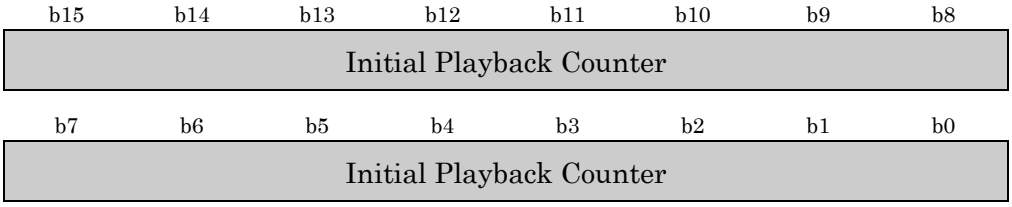
Initial End Time ... This field describes the initial end time by the hour.

0~23: Hours from midnight.

Others: Reserved.

(RBP 32 to 33) UR_I_P_CNT

This field describes the initial permitted playback count.



Initial Playback Counter ... 0000h: Playback is never permitted.

... 0001h~FFFEh: Playback is permitted specified times.

... FFFFh: Playback is permitted unlimited times.

(RBP 34) UR_I_TBUR_CDT

This field shows initial conditions of the CKI with Time-Based Usage Rules in the CKI.

b7	b6	b5	b4	b3	b2	b1	b0
Clock Usage Flag [1 .. 0]		Reserved					

Clock Usage Flag ... This shows a clock mode restriction of playback the content by the copied CKI.

00b: Devices in Mode A, Mode B1 and Mode B2 are allowed to playback the content.

01b: Reserved.

10b: Devices in Mode B1 and Mode B2 are allowed to playback the content. Devices in Mode A are not allowed to playback the content.

11b: Devices in Mode B2 are allowed to playback the content. Devices in Mode A and Mode B1 are not allowed to playback the content.

UR_I_STRTDATE, UR_I_ENDDATE, UR_I_P_CNT and UR_I_TBUR_CDT are not treated at playback. These fields are inherited to a replicated CKI when copying. Specifically at copying, the current field group of replicated CKI is set to these fields.

(RBP 37 to 39) UR_SPAN

This field describes the permitted playback span.

b23	b22	b21	b20	b19	b18	b17	b16
Validity of Span	Span Length	Span Days [16 .. 11]					

b15	b14	b13	b12	b11	b10	b9	b8
Span Days [10 .. 3]							

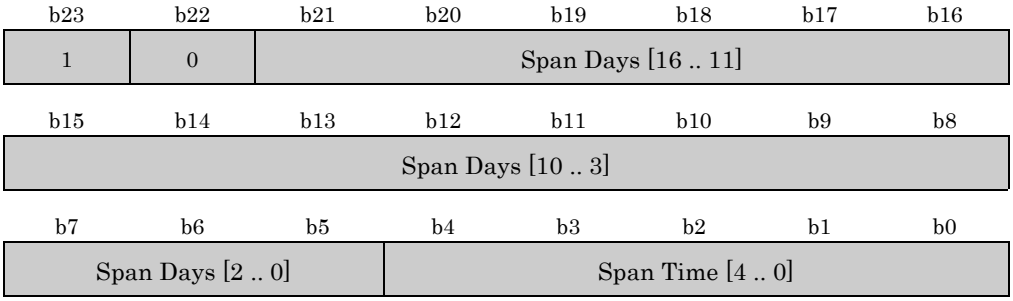
b7	b6	b5	b4	b3	b2	b1	b0
Span Days [2 .. 0]			Span Time [4 .. 0]				

Validity of Span ... 0b: The playback span is not specified.

1b: The playback span is specified.

Span Length ... 0b: The length of span is described as Long Span Mode.
1b: The length of span is described as Short Span Mode.

In the case of Long Span Mode, the rest of the field is described as follows:



Span Days ... This field describes day portion of the permitted playback span.

0~131071: span of days

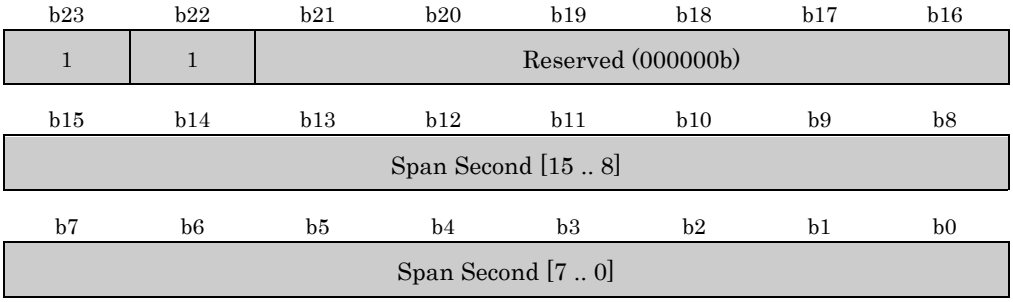
Span Time ... This field describes time portion of the permitted playback span by hours.

0~23: Valid hours.

Others: Reserved.

This field is set when the corresponding content is distributed and shall not be changed. This field shall be inherited to a replicated content when copying.

In the case of Short Span Mode, the rest of the field is described as follows:

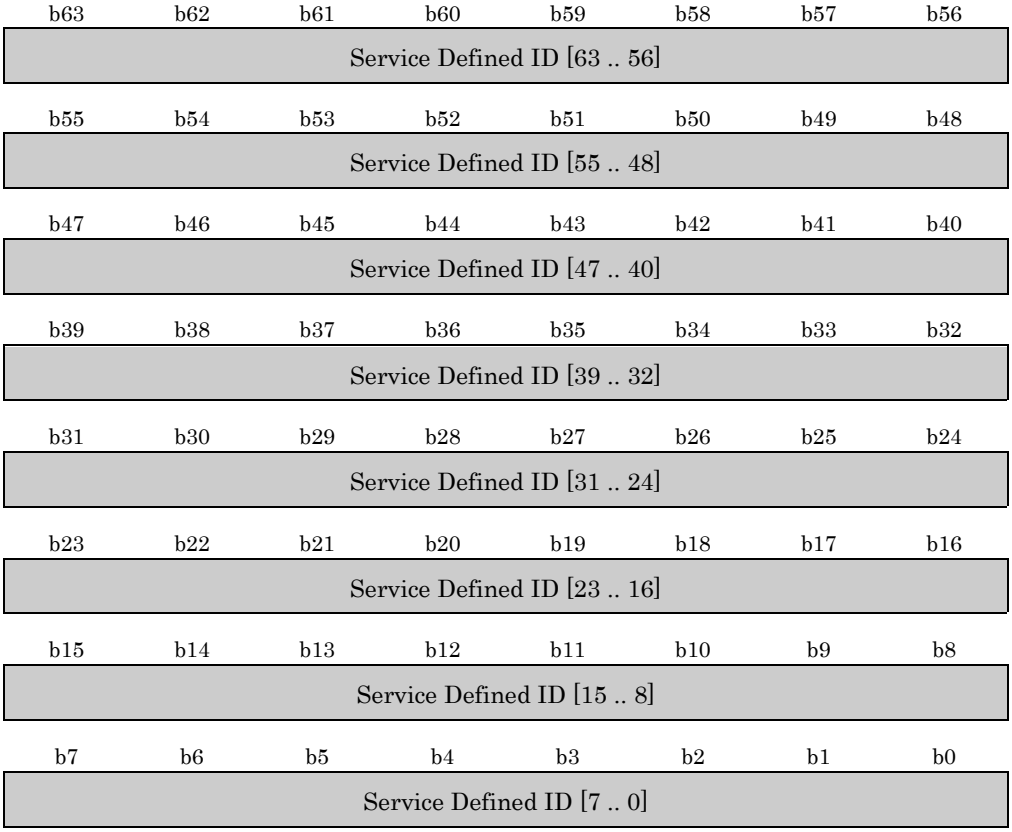


Span Second ... This field describes time portion of the permitted playback span.

0~65535: This value described how many seconds device can playback this content. For example, if the value is set to 330, the Content Key shall be used for playback during 5 minutes and a half after beginning of playback. After the specified span, the playback shall be terminated immediately. How to count the time and check is described in Section 6.12.

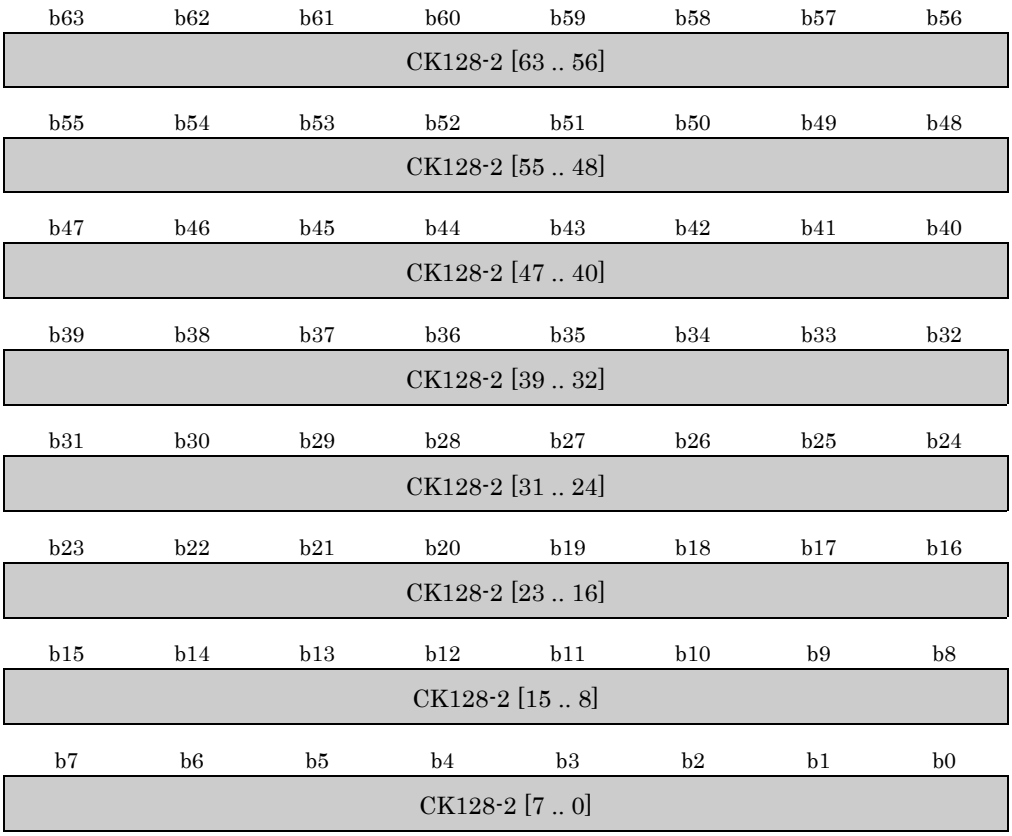
(RBP 40 to 47) UR_SD_ID or CK128-2

In the case of C2-CPRM, this field describes the Service Defined ID.



Service Defined ID ... Stores the Service Defined ID.

In the case of AES-CPRM, this field describes the low 8 bytes of 128-bit Content Key.



CK128-2 ... Stores the low 8 bytes of 128-bit Content Key

(RBP 48 to 55) UR_CHECK

In the case of C2-CPRM, this field stores the 64-bit check value, '0123456789ABCDEFh.'

In the case of AES-CPRM, this field stores the high 8 bytes of CMAC(K_{C128}, the high 40 bytes of this UR_C).

Chapter 5
CPRM Related Data in Protected Area

5. CPRM Related Data in Protected Area

This chapter shows the file system of the Protected Area. The physical allocation of the Protected Area is described in *SD Memory Card Specification –Part3 Security Specification*.

5.1 File System of Protected Area for SD-SD

This section describes the file system of the Protected Area in which Time Stamp files (TBUR.TS, TBUR_A.TS and TBUR_B.TS), the encrypted User Keys and encrypted Usage Rules for User Key (UR_U) are stored.

5.1.1 User Key & Usage Rule Master Manager (UKURMMG)

A single master manager file manages all the C2 User Key & Usage Rule Manager files in the Protected Area and a single master manager file manages all the AES User Key & Usage Rule Manager files in the Protected Area. The file is called User Key & Usage Rule Master Manager (UKURMMG) file.

5.1.2 User Key & Usage Rule Manager (UKURMG)

The User Key and the UR_U are encrypted by a Media Unique Key or Expanded Media Unique Key. The Encrypted User Key and UR_U are stored as a file of the Protected Area. The file is called User Key & Usage Rule Manager (UKURMG) file. In the Protected Area, there can be multiple UKURMG files.

5.1.3 Directory and File Configuration in Protected Area

Figure 5-1 shows an example directory and file configuration of the Protected Area for the SD-SD specifications.

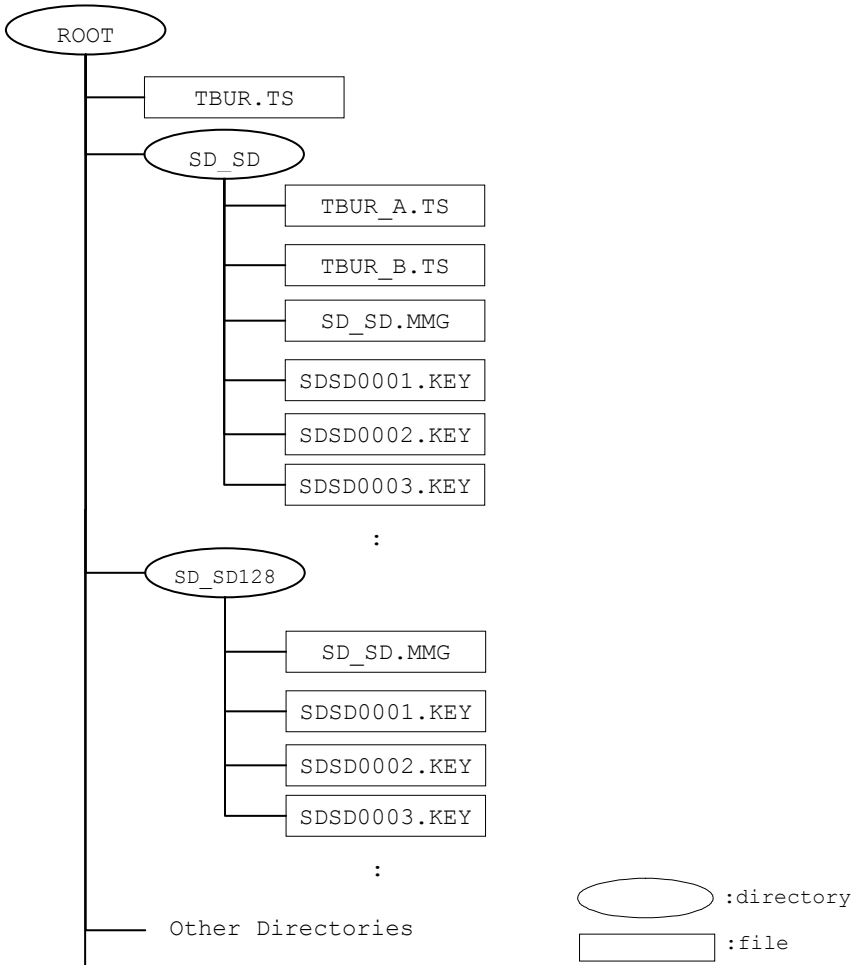


Figure 5-1 Directory and File Configuration

Under the root directory, the following files and directories may be stored:

a file named “TBUR.TS”,

a directory named “SD_SD”,

a directory named “SD_SD128” and

several directories related to SD applications other than SD-SD.

Under the “SD_SD” directory, the following files may be stored:

a file named “TBUR_A.TS,”

a file named “TBUR_B.TS,”

a file named “SD_SD.MMG” and

several files named “SDSDxxxx.KEY” where xxxx is a serial decimal number (0001~0256).

Under the “SD_SD128” directory, the following files may be stored:

a file named “SD_SD.MMG” and

several files named “SDSDxxxx.KEY” where xxxx is a serial decimal number (0001~0256).

The name of the UKURMMG file shall be “SD_SD.MMG.”

- The size of a UKURMG file is fixed. It contains 250 User Key & Usage Rule Entries (UKUREs).
- There can be at most 256 UKURMG files in each of the “SD_SD” directory and the “SD_SD128” directory of the Protected Area.
- The name of a UKURMG file shall be:

  SDSDxxxx.KEY,

  where xxxx is a serial number (0001~0256) assigned to each of the UKURMG files in the “SD_SD” directory or “SD_SD128” directory.
- The UKURE Search Number (UKURE_SRN) is a serial number uniquely associated with each UKURE of all the UKURMG files in each of the “SD_SD” directory and the “SD_SD128” directory. Which directory of those two is used is defined service directory by service directory. The directory contains at most 256 UKURMG files, each of which has 250 UKUREs, and the maximum number of UKURE_SRN is 64,000 (250 x 256). For example, UKURE #1 through UKURE #250 in the SDSD0001.KEY file are associated with UKURE_SRN 1 through 250, UKURE #1 through UKURE #250 in the SDSD0002.KEY file are associated with UKURE_SRN 251 through 500, and so forth.
- All 128-bit User Keys shall be stored under the “SD_SD128” directory. All C2 User Keys shall be stored under the “SD_SD” directory.
- Each encrypted Content Key in the User Data Area is associated with the corresponding UKURE in the Protected Area through its UKURE_SRN.
- The UKURE_SRN of the corresponding UKURE is stored in the User Key Information (UKI) of the User Key Manager file (UKMG) in the User Data Area. Regarding the structure and the file names in the User Data Area, refer to *SD Memory Card Specifications, Part 15: SD-SD (Separate Delivery) Specifications*.
- The UKURE_SRN field has UKURE_SRN, which shall be unique in each of the “SD_SD directory” and “SD_SD128” directory respectively.
- There is a case where the “SD_SD128” directory exists and the “SD_SD” directory does not exist. The case means that no C2 User Keys are stored and only 128-bit User Keys are stored in the SD Memory Card.
- Since both devices supporting C2-CPRM and devices supporting AES-CPRM use Time Stamp files under the “SD_SD” directory, there is a case where the “SD_SD128” and the “SD_SD” directory exist and only Time Stamp file(s) exists in the “SD_SD” directory.

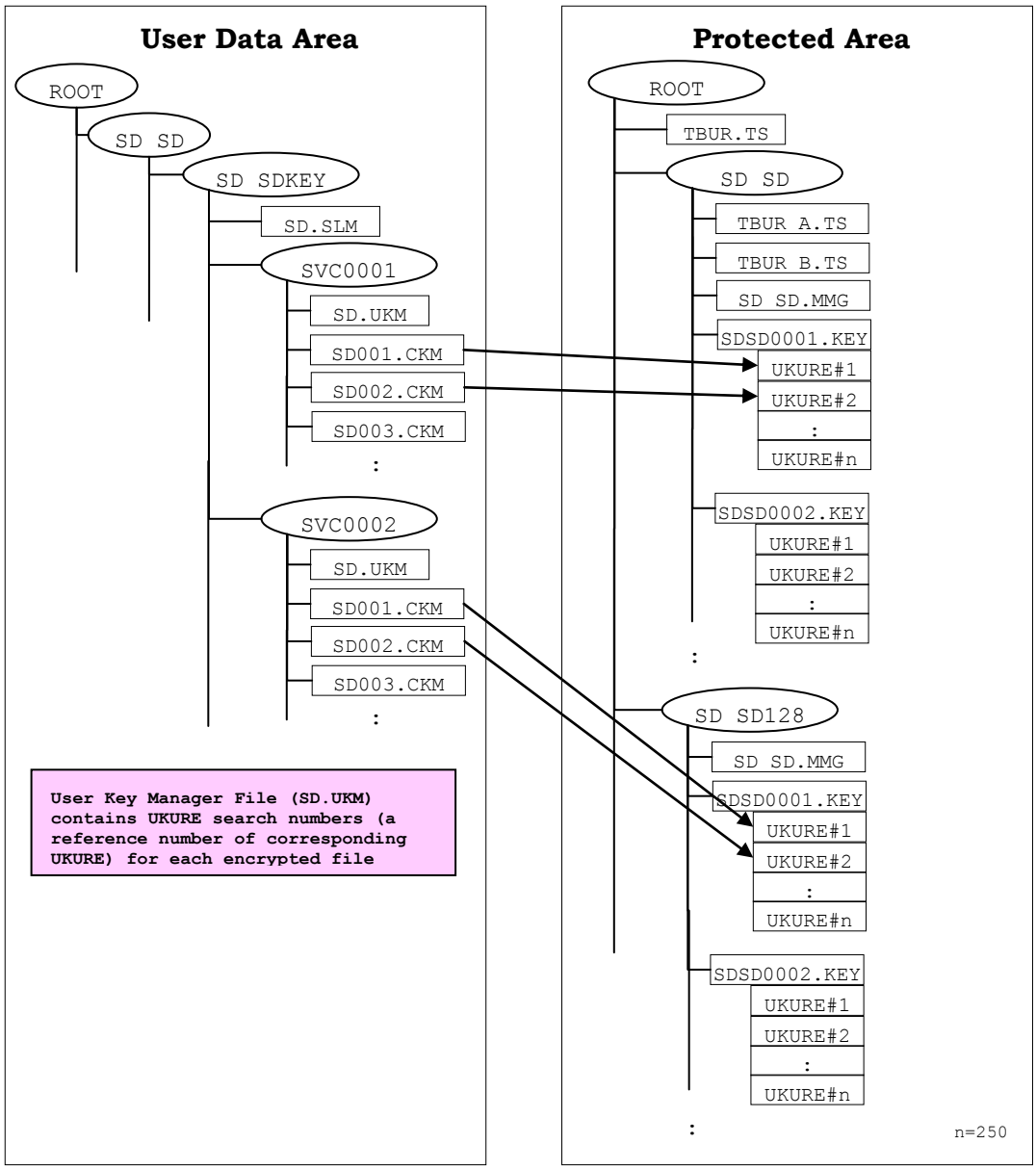


Figure 5-2 Relationship between Directory and Filename

5.2 Structure of User Key & Usage Rule Master Manager

As shown in Table 5-1, the UKURMMG contains Version number, Application ID of UKURMG, and UKURMG Used flag.

Table 5-1 UKURMMG

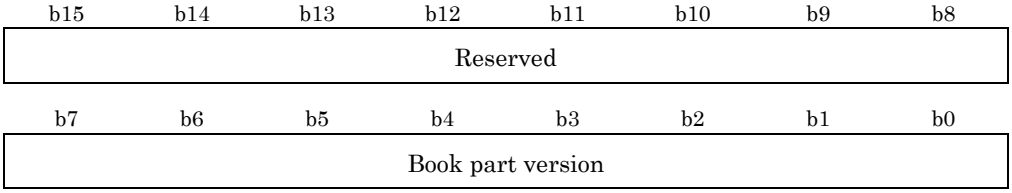
(Description order)

RBP	Field Name	Contents	Number of bytes
0 to 1	VERN	Version number	2 bytes
2 to 3	UKURMG_AP_ID	Application ID of UKURMG	2 bytes
4 to 31	Reserved	Reserved	28 bytes
32 to 63	UKURMG_USED	UKURMG Used flag	32 bytes
Total			64 bytes

All reserved bits shall be set to '0b.' For forward compatibility, the Playback Device compliant with the current revision of this book shall ignore these reserved bits.

(RBP 0 to 1) VERN

This field describes the Version number of the SD-SD specification defined by the SD Association.



Book part version ... 10h: version 1.0

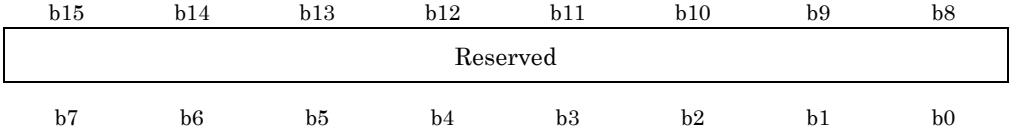
11h: version 1.1

12h: version 1.2

Others: reserved

(RBP 2 to 3) UKURMG_AP_ID

This field describes the Application ID of UKURMG. This value must be '0Ch.'



Application ID

Application ID ... 0Ch: SD-SD

(RBP 32 to 63) UKURMG_USED

This field describes whether each UKURMG has unused UKUREs or not. Notwithstanding bit ordering in Section 1.4, bit positions within UKURMG Used flag are numbered such that the most significant bit is numbered 0 and the least significant bit is numbered 255. Therefore, for example, UKURMG Used flag[0] is stored at b255.

b255	b254	b253	b252	b251	b250	b249	b248
UKURMG Used flag [0 .. 7]							

:

:

b7	b6	b5	b4	b3	b2	b1	b0
UKURMG Used flag [248 .. 255]							

UKURMG Used flag [*j*-1] ... 1b: "SDSD*j*.KEY" exists and all the UKUREs of "SDSD*j*.KEY" are used.

0b: Either "SDSD*j*.KEY" does not exist, or

"SDSD*j*.KEY" exists and has some unused UKUREs.

5.3 Structure of User Key & Usage Rule Manager

Figure 5-3 shows the structure of a User Key & Usage Rule Manager (UKURMG).

(UKURMG)

User Key & Usage Rule Manager Information (UKURMGI)
User Key & Usage Rule Entry #1 (UKURE #1)
User Key & Usage Rule Entry #2 (UKURE #2)
:
User Key & Usage Rule Entry #n (UKURE #n)

(n = 250)

Figure 5-3 User Key & Usage Rule Manager (UKURMG)

A UKURMG file starts with a User Key & Usage Rule Manager Information (UKURMGI), followed by a set of User Key & Usage Rule Entries (UKUREs). UKURE number is from 1 to 250.

5.4 User Key & Usage Rule Manager Information (UKURMGI)

As shown in Table 5-2, the UKURMGI contains UKURE Used flag.

Table 5-2 UKURMGI

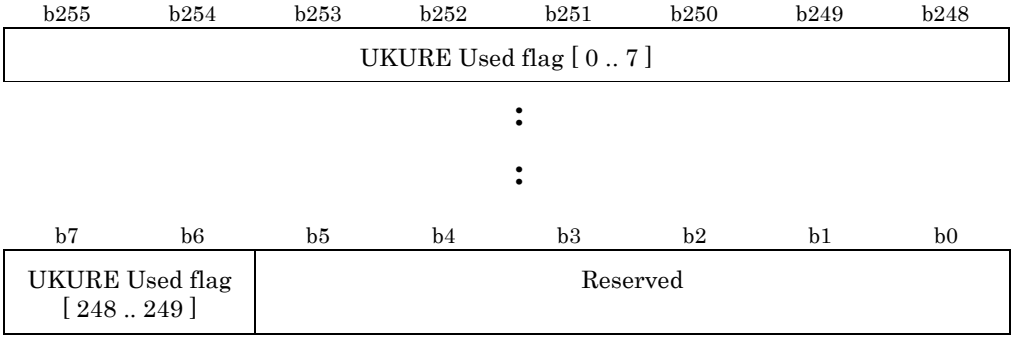
(Description order)

RBP	Field Name	Contents	Number of bytes
0 to 31	UKURE_USED	UKURE Used flag	32 bytes
32 to 383	Reserved	Reserved	352 bytes
Total			384 bytes

All reserved bits shall be set to '0b.'

(RBP 0 to 31) UKURE_USED

This field describes whether each UKURE in this UKURMG is used or not. Notwithstanding bit ordering in Section 1.4, bit positions within UKURE Used flag are numbered such that the most significant bit is numbered 0 and the least significant bit is numbered 249. Therefore, for example, UKURE Used flag[0] is stored at b255.



UKURE Used flag [*j*] ... 0b: UKURE # *j+1* in this UKURMG is not used.

(UKURE # *j+1* is vacant.)

1b: UKURE # *j+1* in this UKURMG is used.

(UKURE # *j+1* is not vacant.)

5.5 User Key & Usage Rule Entry (UKURE)

As shown in Table 5-3, a UKURE consists of User Key Area (UKA) and Usage Rules for User Key (UR_U). In the case of C2-CPRM, the whole UKURE is encrypted using C2_ECBC (both fields are concatenated and then encrypted using C2_ECBC). In the case of AES-CPRM, the 128-bit User Key is protected with the Expanded Media Unique Key and the UKURE is protected with the 128-bit User Key.

Table 5-3 UKURE

(Description order)

RBP	Field Name	Contents	Number of bytes
0 to 7	UKA	User Key Area	8 bytes
8 to 63	UR_U	Usage Rules for User Key	56 bytes
Total			64 bytes

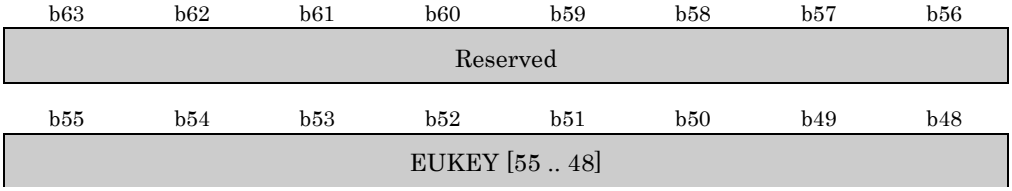
5.5.1 User Key Area (UKA)

In the case of C2-CPRM, as shown in Table 5-4, UKA contains the User Key of the corresponding encrypted content.

Table 5-4 UKA for C2-CPRM

(Description order)

RBP	Field Name	Contents	Number of bytes
0	Reserved	Reserved	1 byte
1 to 7	EUKEY	User Key	7 bytes
Total			8 bytes



b47	b46	b45	b44	b43	b42	b41	b40
EUKEY [47 .. 40]							
b39	b38	b37	b36	b35	b34	b33	b32
EUKEY [39 .. 32]							
b31	b30	b29	b28	b27	b26	b25	b24
EUKEY [31 .. 24]							
b23	b22	b21	b20	b19	b18	b17	b16
EUKEY [23 .. 16]							
b15	b14	b13	b12	b11	b10	b9	b8
EUKEY [15 .. 8]							
b7	b6	b5	b4	b3	b2	b1	b0
EUKEY [7 .. 0]							

EUKEY ... Stores the User Key.

All reserved bits (from b56 to b63) shall be set to '0b.' For forward compatibility, the Playback Device compliant with the current revision of this book shall ignore these reserved bits.

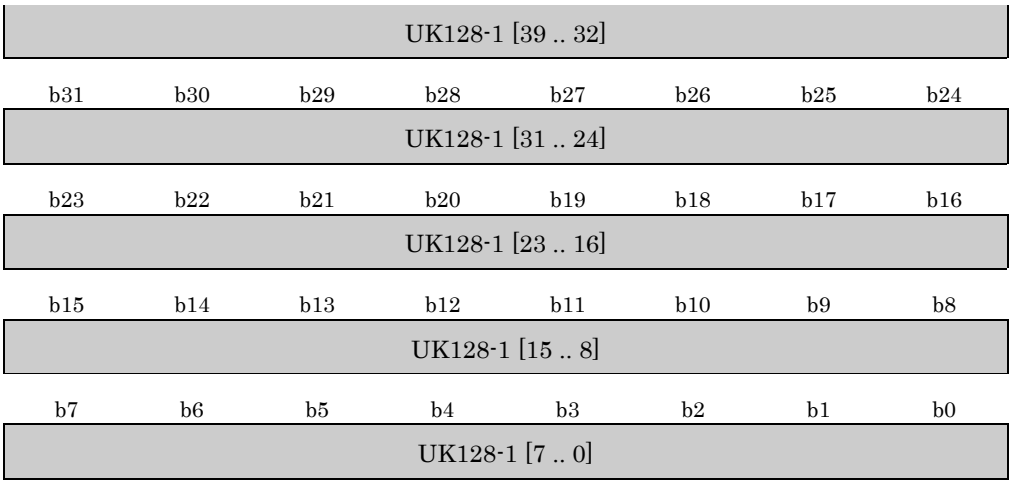
In the case of AES-CPRM, as shown in Table 5-5, UKA contains the high 8 bytes of 128-bit User Key. The low 8 bytes of 128-bit User Key is stored in UK128-2 of Usage Rules for User Key.

Table 5-5 UKA for AES-CPRM

(Description order)

RBP	Field Name	Contents	Number of bytes
0 to 7	UK128-1	High 8 bytes of 128-bit User Key	8 bytes
Total			8 bytes

b63	b62	b61	b60	b59	b58	b57	b56
UK128-1 [63 .. 56]							
b55	b54	b53	b52	b51	b50	b49	b48
UK128-1 [55 .. 48]							
b47	b46	b45	b44	b43	b42	b41	b40
UK128-1 [47 .. 40]							
b39	b38	b37	b36	b35	b34	b33	b32



UK128-1 ... Stores the high 8 bytes of 128-bit User Key.

5.5.2 Usage Rules for User Key (UR_U)

As shown in Table 5-6, Usage Rules for User Key (UR_U) contains Trigger Bits, User Key Type, User Key ID, Start date of valid period of User Key, End date of valid period of User Key, Permitted valid span, Hash Value/the low 8 bytes of 128-bit User Key and Check Value.

Table 5-6 UR_U

(Description order)

RBP	Field Name	Contents	Number of bytes
0	UR_TRIGGER	Trigger Bits	1 byte
1	UR_UKT	User Key Type	1 byte
2 to 17	UR_UK_ID	User Key ID	16 bytes
18 to 20	UR_UK_STARTDATE	Start date of valid period of User Key	3 bytes
21 to 23	UR_UK_ENDDATE	End date of valid period of User Key	3 bytes
24 to 26	UR_UK_SPAN	Permitted valid span of User Key	3 bytes
27 to 31	Reserved	Reserved	5 bytes
32 to 39	Reserved or UR_HASH	Reserved or Hash Value	8 bytes
40 to 47	UR_HASH or UK128-2	Hash Value or the low 8 bytes of 128-bit User Key	8 bytes
48 to 55	UR_CHECK	Check Value	8 bytes
Total			56 bytes

All reserved bits shall be set to ‘0b.’ For forward compatibility, the Playback Device compliant with the current revision of this book shall ignore these reserved bits.

In the following definition of Usage Rule fields, the assigned values are effective only when the UKURE is used. When the UKURE is not used, no specific value is assigned to each Usage Rule field.

(RBP 0) UR_TRIGGER

This field describes Trigger Bits.

b7	b6	b5	b4	b3	b2	b1	b0
Trigger Bits							
TB for TBUR	TB for AES	TB for future use					

Trigger Bit for Time Based Usage Rules (TB for TBUR)		0b: Time Based Usage Rule for User Key is invalid. 1b: Time Based Usage Rule for User Key is valid.
Trigger Bit for AES (TB for AES)		0b: C2-CPRM is used for encryption mechanism. 1b: AES-CPRM is used for encryption mechanism.
Trigger Bits for future use (TB for future use)	...	000000b: Accessing devices conforming to this specification can control all processes. 000001b~111111b: Accessing devices conforming to this specification shall not be permitted the all processes except for Erasing Process. In a future version, the Usage Rules for User Key may be expanded, or other information for controlling these processes may be added. Accessing devices of the future version shall process the new information for controlling these processes correctly when Trigger Bites for future use is set to the value except '000000b.'

(RBP 1) UR_UKT

This field describes whether or not the User Key requires the Hash Value.

b7	b6	b5	b4	b3	b2	b1	b0

Reserved	User Key Type

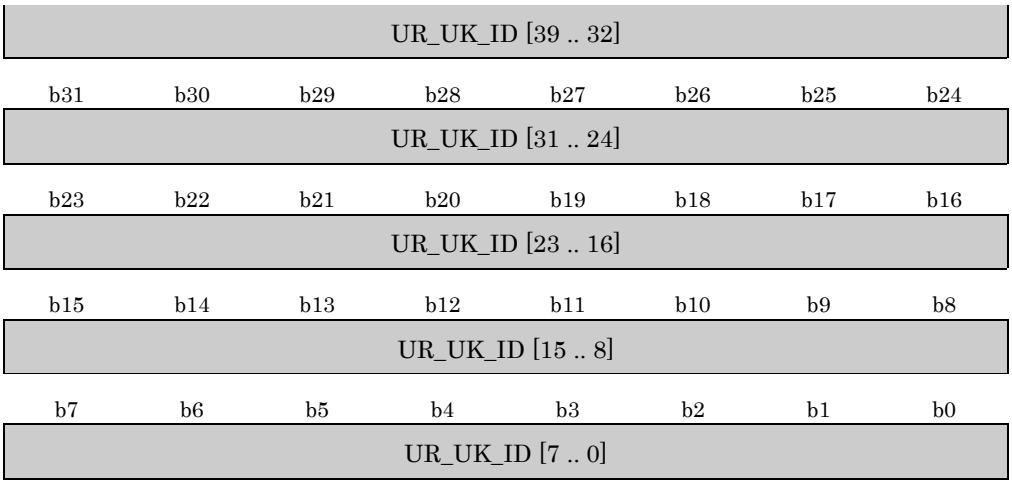
User Key Type ... 0b: User Key requires the Hash Value

1b: User Key does not require the Hash Value

(RBP 2 to 17) UR_UK_ID

This field describes the User Key ID.

b127	b126	b125	b124	b123	b122	b121	b120
UR_UK_ID [127 .. 120]							
b119	b118	b117	b116	b115	b114	b113	b112
UR_UK_ID [119 .. 112]							
b111	b110	b109	b108	b107	b106	b105	b104
UR_UK_ID [111 .. 104]							
b103	b102	b101	b100	b99	b98	b97	b96
UR_UK_ID [103 .. 96]							
b95	b94	b93	b92	b91	b90	b89	b88
UR_UK_ID [95 .. 88]							
b87	b86	b85	b84	b83	b82	b81	b80
UR_UK_ID [87 .. 80]							
b79	b78	b77	b76	b75	b74	b73	b72
UR_UK_ID [79 .. 72]							
b71	b70	b69	b68	b67	b66	b65	b64
UR_UK_ID [71 .. 64]							
b63	b62	b61	b60	b59	b58	b57	b56
UR_UK_ID [63 .. 56]							
b55	b54	b53	b52	b51	b50	b49	b48
UR_UK_ID [55 .. 48]							
b47	b46	b45	b44	b43	b42	b41	b40
UR_UK_ID [47 .. 40]							
b39	b38	b37	b36	b35	b34	b33	b32

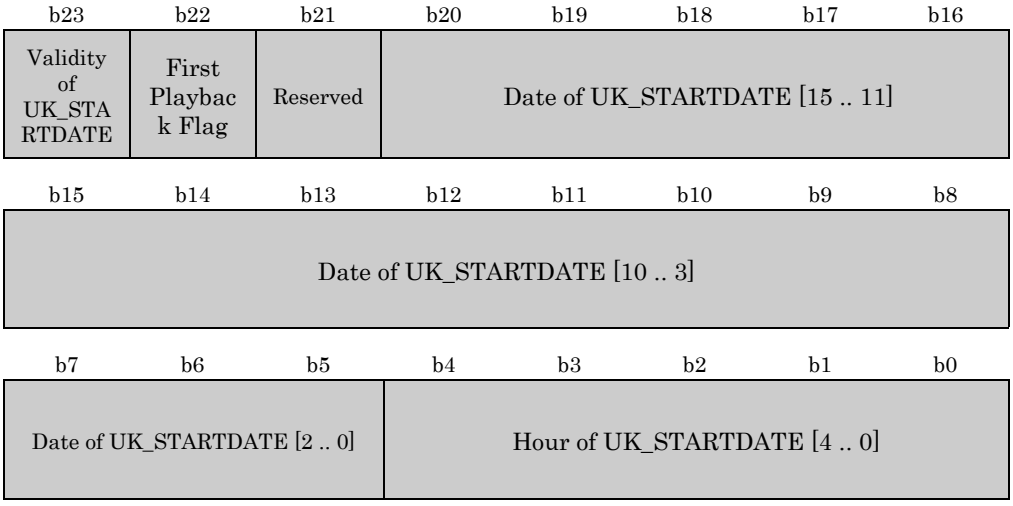


UR_UK_ID ... Stores the User Key ID.

The value in this field is specified by a Service Provider or a Device. Refer to *SD Memory Card Specifications, Part 15: SD-SD (Separate Delivery) Specification.*

(RBP 18 to 20) UR_UK_STARTDATE

This field describes the start date and hour of permitted playback period of the User Key.



Validity of UK_STARTDATE ... 0b: The UK_STARTDATE is invalid.

1b: The UK_STARTDATE is valid

First Playback Flag ... 0b: The User Key has not been used for playback.

1b: The User Key has already used for playback.

Date of UK_STARTDATE ... This field describes the start date to permit the usage of the User Key in Modified Julian Date format.

Hour of UK_STARTDATE ... This field describes the start hour of the start date specified by the Date of UK_STARTDATE.

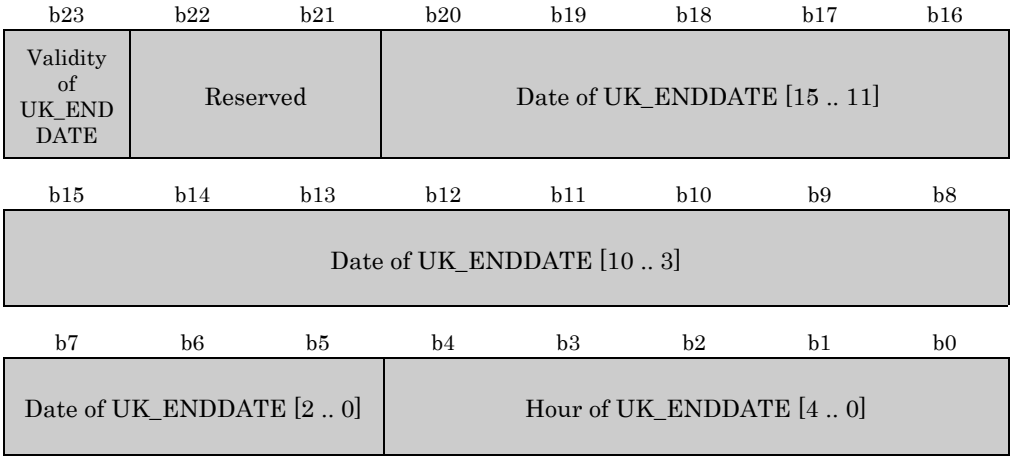
0~23: Hours from midnight.

Others: Reserved.

The reserved bit (b21) shall be set to '0b.' For forward compatibility, the Playback Device compliant with the current revision of this book shall ignore the reserved bit.

(RBP 21 to 23) UR_UK_ENDDATE

This field describes the end date and hour of permitted playback period of the User Key.



Validity of UK_ENDDATE ... 0b: The UK_ENDDATE is invalid.

1b: The UK_ENDDATE is valid.

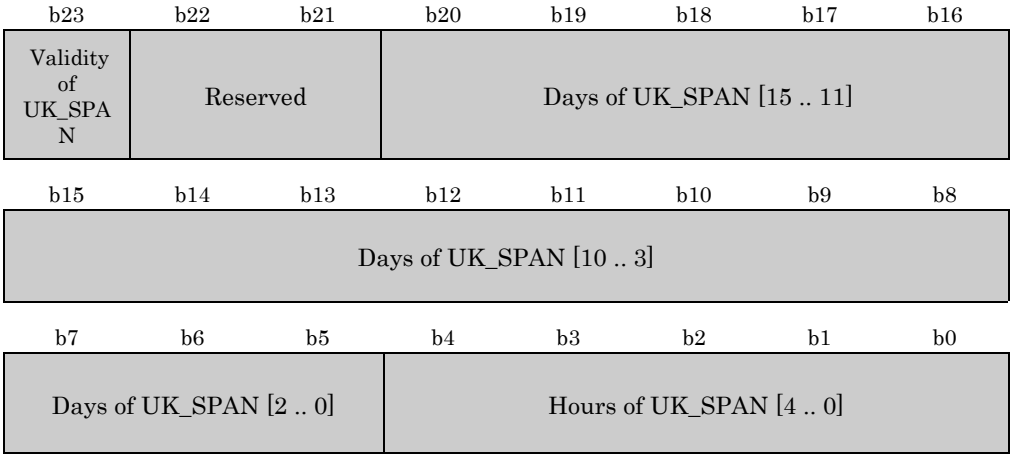
Date of UK_ENDDATE ... This field describes the end date to permit the usage of the User Key in Modified Julian Date format.

Hour of UK_ENDDATE	...	This field describes the end hour of the end date specified by the Date of UK_STARTDATE. 0~23: Hours from midnight. Others: Reserved.

The reserved bits (b22 and b21) shall be set to '00b.' For forward compatibility, the Playback Device compliant with the current revision of this book shall ignore these reserved bits.

(RBP 24 to 26) UR_UK_SPAN

This field describes the permitted playback span of the User Key.



Validity of UK_SPAN	...	0b: The UK_SPAN is invalid. 1b: The UK_SPAN is valid.
Days of UK_SPAN	...	This field describes day portion of the effective span of the User Key. 0~131071: days
Hours of UK_SPAN	...	This field describes hour portion of the effective span of the User Key. 0~23: hours. Others: Reserved.

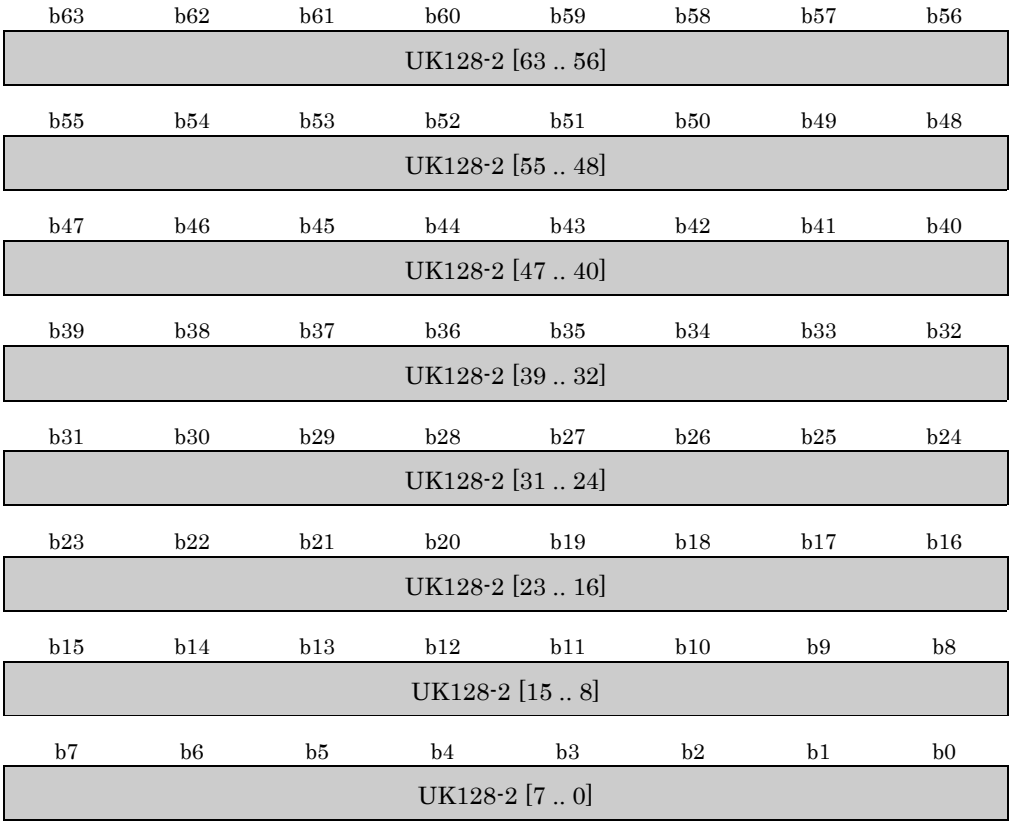
Reserved bits (b22 and b21) shall be set to '00b.' For forward compatibility, the Playback Device compliant with the current revision of this book shall ignore these reserved bits.

(RBP 40 to 47) or (RBP 32 to 39) UR_HASH

This field stores a Hash Value which is derived from CKIs encrypted the User Key. How to calculate this value is described in Section 3.5.2. When the User Key Type is '1b,' this field shall be set to '0000000000000000h.' The UR_HASH is stored in (a) from 40th byte to 47th byte of UR_U in the case of C2-CPRM or (b) from 32nd byte to 39th byte of UR_U in the case of AES-CPRM.

(RBP 40 to 47) UK128-2

This field describes the low 8 bytes of 128-bit User Key. The high 8 bytes of 128-bit User Key is stored in UK128-1 of Usage Rules for User Key.



UK128-2 ... Stores low 8 bytes of 128-bit User Key.

(RBP 48 to 55) UR_CHECK

In the case of C2-CPRM, this field stores the 64-bit check value, '0123456789ABCDEFh.'

In the case of AES-CPRM, this field stores the high 8 bytes of CMAC(K_{u128}, the high 40 bytes of this UR_U).

5.6 Structure of Time Stamp files

In this SD-SD specification, the following Time Stamp files are treated. In this section, structure of these files is described.

/TBUR.TS

/SD_SD/TBUR_A.TS

/SD_SD/TBUR_B.TS

5.6.1 TBUR_A.TS

TBUR_A.TS is handled by devices that work in Mode A described in Section 3.3.1. More precisely, TBUR_A.TS is read and updated by devices that work in Mode A when a CKI with Time-Based Usage Rules is played or recorded. Table 5-7 describes the data format of the TBUR_A.TS.

Table 5-7 TBUR_A.TS

(Description order)

RBP	Field Name	Contents	Number of bytes
0 to 7	AN	Arbitrary number	8 bytes
8 to 11	TS	Time Stamp	4 bytes
12 to 15	TS Verification Data	Time Stamp Verification Data	4 bytes
Total			16 bytes

The data of /SD_SD/TBUR_A.TS file is encrypted using C2_ECBC by the Media Unique Key (K_{mu}) derived from the MKB Process by the Static MKB or both the Static MKB and the MKB Extension file denoted "SD_SDKEY.MKB" (as described in Section 3.9 of *SD Memory Card Book Common Part*).

Namely, /SD_SD/TBUR_A.TS consists of Encrypted Time Stamp Data (D_{tse}) as:

D_{tse} = C2_ECBC(K_{mu}, D_{ts}),

where D_{ts} = AN || TS || TS Verification Data.

Therefore, when using the time of TBUR_A.TS, the device retrieve the time of Time Stamp file to decrypt by the following formula.

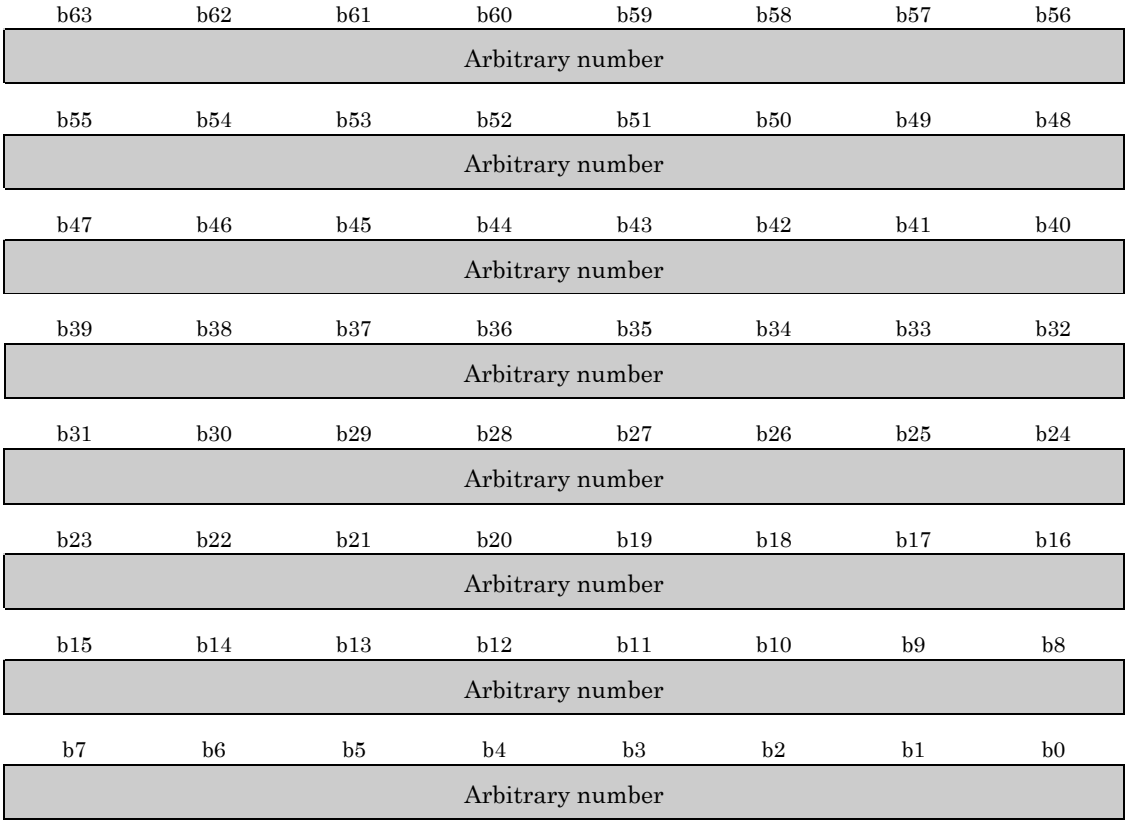
D_{ts} = C2_DCBC(K_{mu}, D_{tse}),

where D_{tse} is the data of TBUR_A.TS

Note that all reserved bits within the TBUR_A.TS shall be set to '0b.' For forward compatibility, the Playback Device compliant with the current revision of this book shall ignore these reserved bits.

(RBP 0 to 7) AN

This field stores the 64-bit arbitrary number. For example, a random number may be stored in this field. When this file is created, the device generates an arbitrary number and stores it in this field. The device is not required to update the value of this field when this file is updated.



Arbitrary number ... This field describes the 64-bit arbitrary number

(RBP 8 to 11) TS

This field describes a Time Stamp indicating the date and time of the last time any content with Time-Based Usage Rules was played or recorded by devices that work in Mode A, a card 'In-Use' flag, and a 'Exception Termination' counter to keep track of how many times unexpected termination of playback has occurred while playing content with Time-Based Usage Rules.

The devices that work in Mode A use the 'In-Use' flag and the 'Exception Termination' counter to keep track of situations where playback is attempted after pulling the card during playback. Note that while pulling the card effectively stops playback it does not allow the Playback Device to update the Time Stamp file. The Playback Device increments the 'Exception Termination' counter when a "Pull Card Attack" is detected. An SD Memory Card is considered to be in-use if

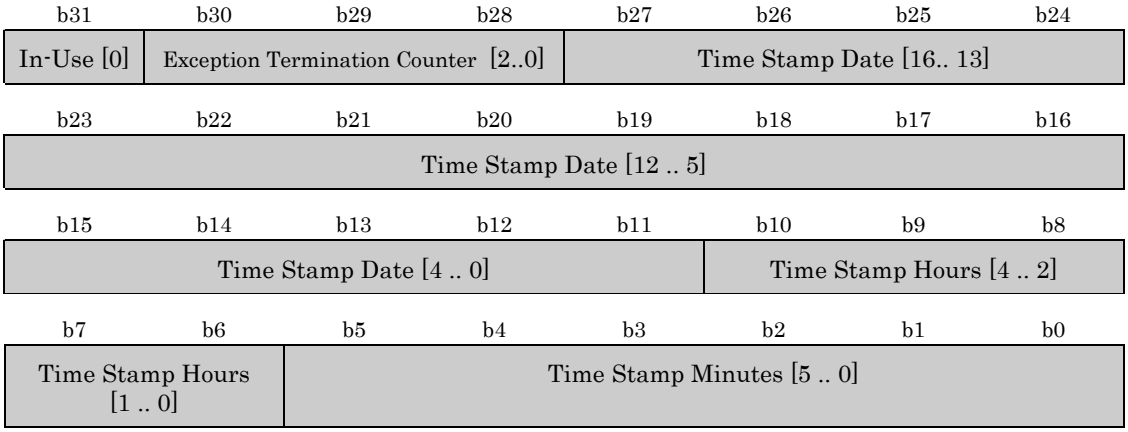
- A device started playing content with Time-Based Usage Rules, or
- A device resumes playback of content with Time-Based Usage Rules.

An SD Memory Card is considered not to be in-use if

- Any devices are playing content that does not have Time-Based Usage Rules, or
- A Mode A device has finished playing content that has Time-Based Usage Rules.

Details on how to update the 'In-Use' flag and 'Exception Termination' counter are explained in Section 6.13 of this book. In the case of the TBUR_A.TS in the \SD_SD directory, this TBUR_A.TS field shall be encrypted according to the process similar to the "Encrypt Title Key and CCI process" described in Section 3.4(4a) of the *CPRM SD Memory Card Book Common Part*.

Note that the SD Memory Card can distinguish between content that has never been played and content for which the first playback has been performed, that is, ‘currently active.’ A ‘currently active’ content is one that is not in the ‘period’ state but in the ‘start’ and ‘end’ state. That is, for active content the Start Date and the End Date have been fixed.



In-Use ... This field describes whether or not a CKI stored in the SD Memory Card is being used for playback by a device which works in Mode A. In other words, playback of content with Time-Based Usage Rules has started playback. At playback of content without Time-Based Usage Rules shall not modify this flag.

0b: the Time Stamp was last updated: a) after stopping playback , b) after playback has reached the end of the content .

1b: the Time Stamp was last updated at the start of a playback

Exception Termination Counter ... This field shows how many times playback of content has been terminated unexpectedly. Normal termination is defined as either using the STOP control function or playback until the end of the content is reached. Pulling the card from a device while playing content with Time-Based Usage Rules is considered an exception. In contrast, playback of content without Time-Based Usage Rules always stops smoothly. That is, pulling the card in this case shall not impact the Exception Termination Counter.

0 (000b)~5 (101b): valid values for this counter. When the number of exception terminations reaches 5, it results on denying playback of currently active content until conditions for compliant playback are met. For details on how to reach such compliant conditions see Section 6.13.1 of this specification.

Others: Reserved

Time Stamp Date ... This field describes the current Time Stamp date in Modified Julian Date format.

Time Stamp Hours ... This field describes the current Time Stamp by the hour.

0 (00000b) ~ 23 (10111b): Hours from midnight.

Others: Reserved.

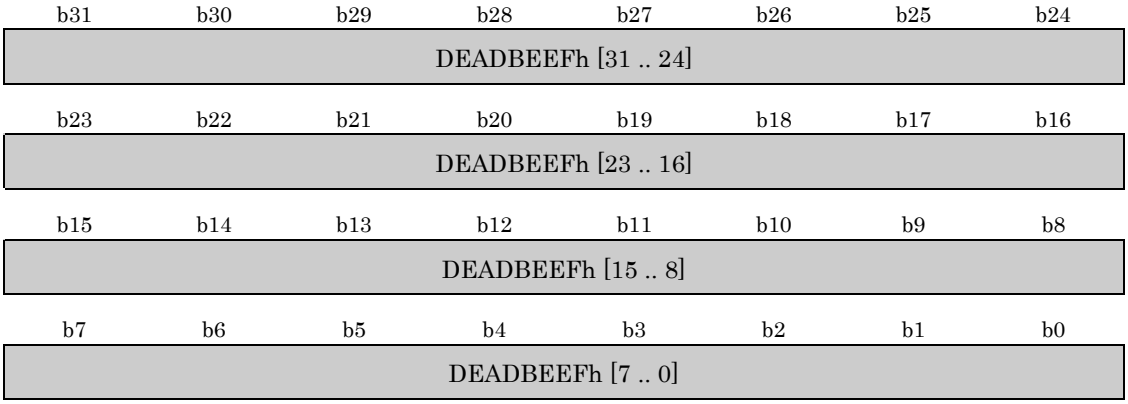
Time Stamp Minutes ... This field describes the current Time Stamp by the minutes.

0 (00000b) ~ 59 (111011b): Minutes after the hour stated in Time Stamp Time Hours field.

Others: Reserved.

(RBP 12 to 15) TS Verification Data

This field stores the 32-bit Verification Data, DEADBEEFh.



Verification Data ... This field describes the 32 bits verification data "DEADBEEFh"

5.6.2 TBUR_B.TS

TBUR_B.TS is handled by devices that work in Mode B described in Section 3.3.1. More precisely, TBUR_B.TS is updated by only devices that work in Mode B when a CKI with Time-Based Usage Rules is recorded. Table 5-8 describes the data format of the TBUR_B.TS.

Table 5-8 TBUR_B.TS

(Description order)

RBP	Field Name	Contents	Number of bytes
0 to 7	AN	Arbitrary Number	8 bytes
8 to 11	TS	Time Stamp	4 bytes
12 to 15	TS Verification Data	Time Stamp Verification Data	4 bytes

Total	16 bytes

The data of /SD_SD/TBUR_B.TS file is encrypted using C2_ECBC by the Media Unique Key (K_{mu}) derived from the MKB Process by the Static MKB or both the Static MKB and the MKB Extension file denoted "SD_SDKEY.MKB" (as described in Section 3.9 of *SD Memory Card Book Common Part*).

Namely, /SD_SD/TBUR_B.TS consists of Encrypted Time Stamp Data (D_{tse}) as:

D_{tse} = C2_ECBC(K_{mu}, D_{ts}),

where D_{ts} = AN || TS || TS Verification Data.

Therefore, when using the time of TBUR_B.TS, the device retrieve the time of Time Stamp file to decrypt by the following formula.

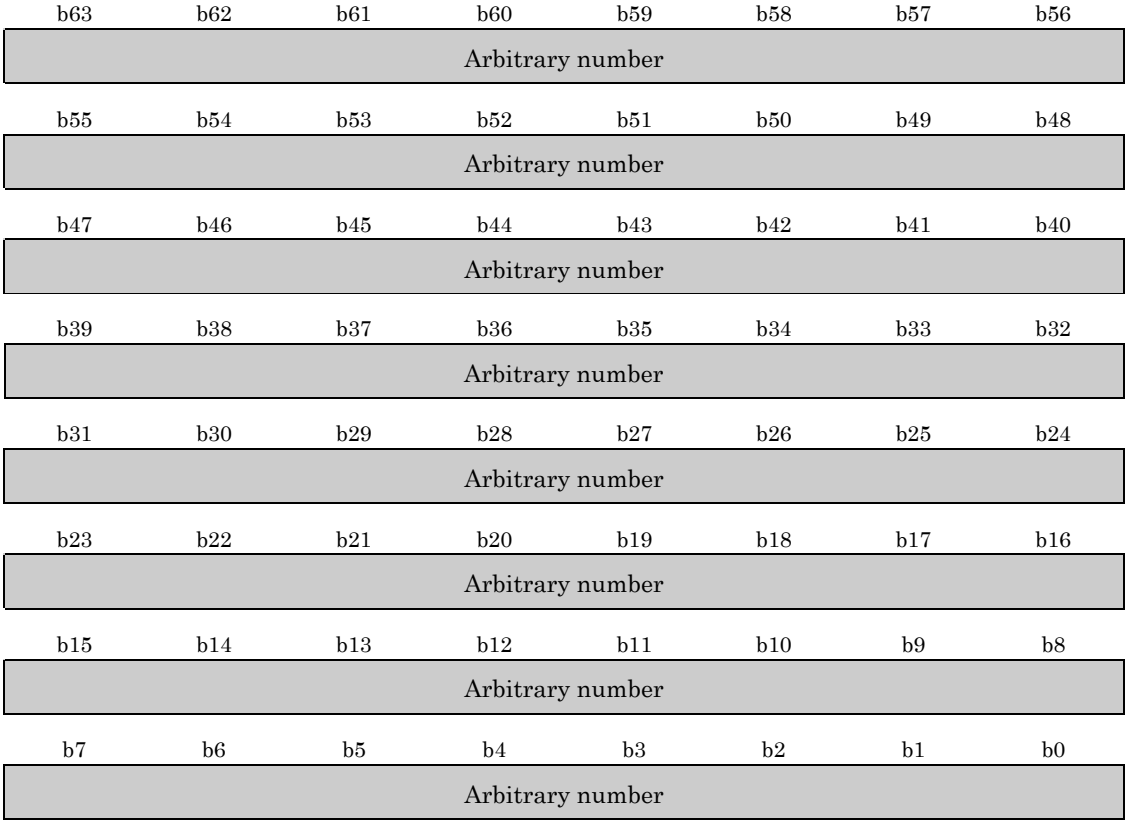
D_{ts} = C2_DCBC(K_{mu}, D_{tse}),

where D_{tse} is the data of TBUR_B.TS

Note that all reserved bits within the TBUR_B.TS shall be set to '0b.' For forward compatibility, the Playback Device compliant with the current revision of this book shall ignore these reserved bits.

(RBP 0 to 7) AN

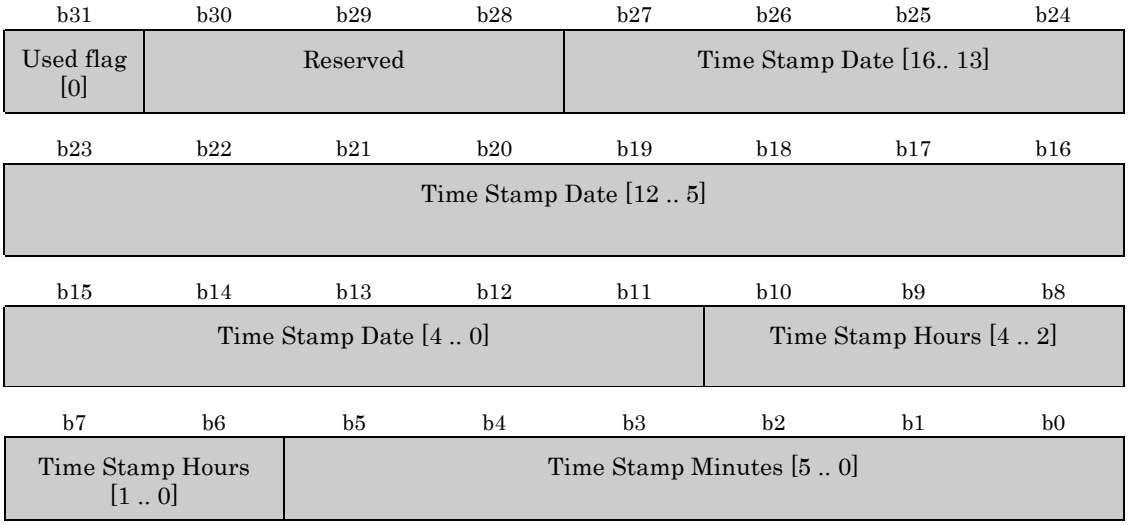
This field stores the 64-bit arbitrary number. For example, a random number may be stored in this field. When this file is created, the device generates an arbitrary number and stores it in this field. The device is not required to update the value of this field when this file is updated.



Arbitrary number ... This field describes the 64-bit arbitrary number

(RBP 8 to 11) TS

This field describes a Time Stamp indicating the date and time of the last time when a CKI with Time-Based Usage Rules is recorded by the devices that works in Mode B.



Used flag ... This field describes if the Time Stamp in TBUR_B.TS file has been used to set Clock B1 by a device that works in Mode B1..

0b: Any device that works in Mode B1 does not use the Time Stamp in TBUR_B.TS to set Clock B1 when this flag = "0b."

1b: the device that works in Mode B1 shall not use the Time Stamp in TBUR_B.TS to set Clock B1 when this flag = "1b."

Time Stamp Date ... This field describes the current Time Stamp date in Modified Julian Date format.

Time Stamp Hours ... This field describes the current Time Stamp by the hour.

0 (00000b) ~ 23 (10111b): Hours from midnight.

Others: Reserved.

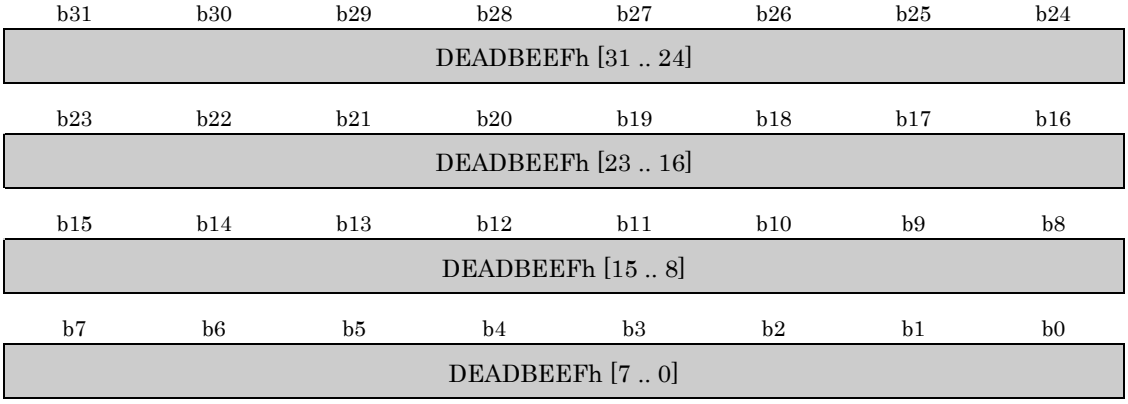
Time Stamp Minutes ... This field describes the current Time Stamp by the minutes.

0 (00000b) ~ 59 (111011b): Minutes after the hour stated in Time Stamp Time Hours field.

Others: Reserved.

(RBP 12 to 15) TS Verification Data

This field stores the 32-bit Verification Data, DEADBEEFh.



Verification Data ... This field describes the verification data "DEADBEEFh"

5.6.3 TBUR.TS

TBUR.TS is handled by any SD applications. More precisely, it is optional to support this file, but recommended, for devices that work in Mode A or Mode B to update TBUR.TS. Table 5-9 describes TBUR.TS.

Table 5-9 TBUR.TS

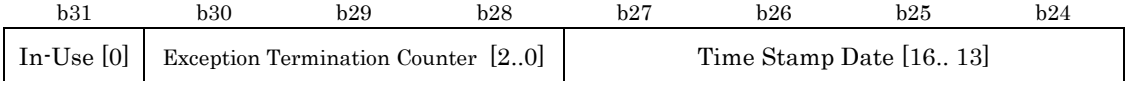
(Description order)

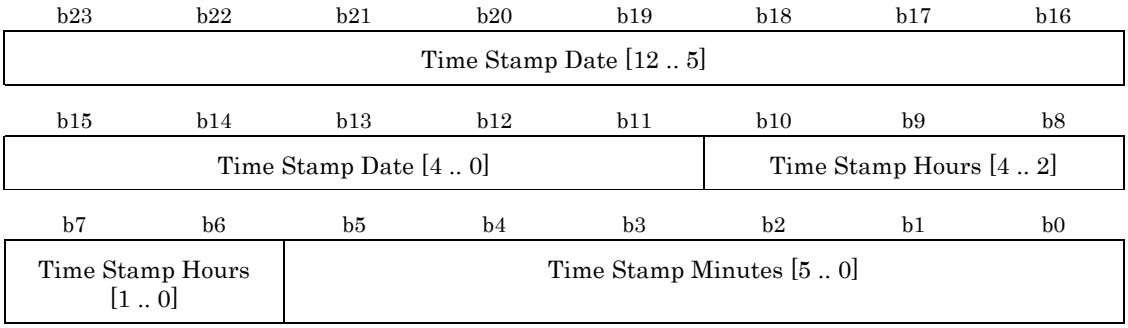
RBP	Field Name	Contents	Number of bytes
0 to 3	TS	Time Stamp	4 bytes
4 to 7	Reserved	Reserved	4 bytes
Total			8 bytes

This TBUR.TS file does not contain an Arbitrary Number and a Verification Data because the file is not encrypted. Note that all reserved bits within the TBUR.TS shall be set to '0b.' Unless otherwise specified, for forward compatibility, devices shall ignore non-zero values in these fields.

(RBP 0 to 3) TS

This field describes a Time Stamp field indicating the date and time of the last time any content was played or recorded by a compliant device, a card 'In-Use' flag, and a counter to keep track of how many times unexpected termination of playback has occurred while playing content with Time-Based Usage Rules.

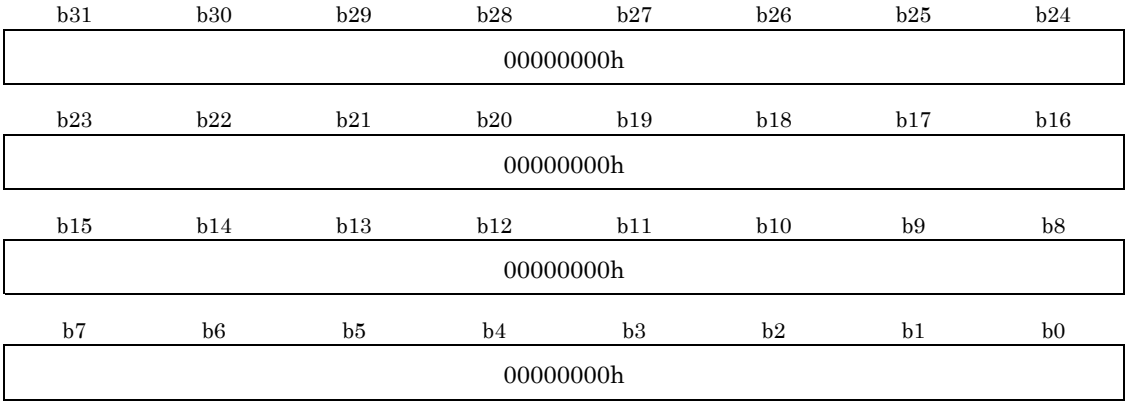




All fields in TS are the same meaning as in the \SD_SD\TBUR_A.TS file, however this file itself is not encrypted.

(RBP 4 to 7) Reserved

This field shall be set to the 32-bit value, "00000000h."



5.7 Update and Recovery of Content Key Manager (CKMG)

This section describes how to update and recover CKMG. For recovery of CKMG, devices shall update CKMG in the following steps.

CKMG Update Process

1. Create updated CKMG file as "SDnnn.BAK" in the same directory which has the original CKMG file.
2. Update corresponding UKURMG including the Hash Value calculated from updated CKMG.
3. Delete the original CKMG file "SDnnn.CKM."
4. Rename "SDnnn.BAK" to "SDnnn.CKM."

In a process described in Chapter 6 and so on, when a calculated Hash Value from CKMG is not equal to corresponding Hash Value in the Protected Area, the device shall execute the following recovery process.

CKMG Recovery Process

1. Check whether corresponding CKMG backup file, "SDnnn.BAK" exists. If does not exist, this process shall abort.
2. Calculate the Hash Value from the backup CKMG file using Hash Calculation Process and check whether the Hash Value is the same value of the corresponding Hash Value stored in the Protected Area. If these values are not identical, this process shall abort.
3. Delete the CKMG file, "SDnnn.CKM" if exists.
4. Rename the backup CKMG file, "SDnnn.BAK" to "SDnnn.CKM".

When this process aborts in step 1 or 2, the process which calls this CKMG Recovery Process shall abort.

This page is intentionally left blank.

Chapter 6
Process Description

6. Process Description

This chapter describes Recording, Erasing, Copy, Move and Playback processes. Even if an SD-SD device does not recognize the profile format of content corresponding to the CKI, the SD-SD device can execute all following processes except Playback Process.

User Key Recording Process

Specifies how a Recording Device writes CPRM protected User Key to an SD Memory Card.

Content Key Recording Process

Specifies how a Recording Device writes CPRM protected Content Key to an SD Memory Card.

Encrypted Content Key Recording Process

Specifies how a Recording Device writes CPRM protected encrypted Content Key to an SD Memory Card.

User Key Erasing Process

Specifies how an Erasing Device erases CPRM protected User Key from an SD Memory Card.

Content Key Erasing Process

Specifies how an Erasing Device erases CPRM protected Content Key from an SD Memory Card.

Content Key Copy Process I (from SD Memory Card to Host)

Specifies how CPRM protected Content Key content on an SD Memory Card is copied securely to a Destination Device (e.g. personal computer).

Content Key Copy Process II (from Host to SD Memory Card)

Specifies how CPRM protected Content Key on a Source Device is copied securely to an SD Memory Card.

Content Key Copy Process III (from SD Memory Card to SD Memory Card)

Specifies how CPRM protected Content Key on an SD Memory Card is copies securely to another SD Memory Card.

Content Key Move Process I (from SD Memory Card to Host)

Specifies how CPRM protected Content Key on an SD Memory Card is copied securely to a Destination Device (e.g. personal computer) and how it is made permanently unusable on the SD Memory Card.

Content Key Move Process II (from Host to SD Memory Card)

Specifies how CPRM protected Content Key on a Source Device is copied securely to an SD Memory Card and how it is made permanently unusable on the Source Device.

Content Key Move Process III (from SD Memory Card to SD Memory Card)

Specifies how CPRM protected Content Key on the first SD Memory Card is copied securely to the second SD Memory Card and how it is made permanently unusable on the first SD Memory Card.

Playback Process

Specifies how the CPRM protected SD-SD content is played back by a Playback Device in conformance with the content's Usage Rules.

User Key Updating Process

Specifies how an Updating Device updates CPRM protected User Key to an SD Memory Card.

When aborting or terminating each process, the processing device shall delete all the temporary images of UKURE/UKURMG/UKURMMG/CKI/CKMG, which are either read from the SD Memory Card or created on the device.

Note that the processing device shall update the files in User Data Area in order to maintain consistency whenever doing each process.

In this section, 'Initial Field Group,' 'Current Field Group,' and 'Fixed Field Group' are used to represent groups of the Usage Rule fields defined as follows:

- 'Initial Field Group' consists of UR_I_STRTDATE, UR_I_ENDDATE, UR_I_P_CNT, UR_I_P_CNT, UR_I_TBUR_CDT and Reserved for IFG.
- 'Current Field Group' consists of UR_C_STRTDATE, UR_C_ENDDATE, UR_C_P_CNT, UR_C_P_CNT, UR_C_TBUR_CDT and Reserved for CFG.
- 'Fixed Field Group' consists of any fields other than Initial Field Group and Current Field Group. Fixed Field Group includes reserved field.

In addition, the following sub-processes are used in each process described in this section. As for the details for these processes, see the corresponding references.

- 'Secure Read Process' is described in *SD Memory Card Book Common Part* Section 3.5.
- 'Secure Write Process' is described in *SD Memory Card Book Common Part* Section 3.5.
- 'UKURE Encryption Process (User Key & Usage Rule Entry Encryption Process)' is described in Section 3.5.1 (step (6a)) of this specification.
- 'UKURE Decryption Process (User Key & Usage Rule Entry Decryption Process)' is described in Section 3.5.1 (step (4b)) of this specification.
- 'CKI Encryption Process (Content Key Information Encryption Process)' is described in Section 3.5.1 (step (4a)) of this specification.
- 'CKI Decryption Process (Content Key Information Decryption Process)' is described in Section 3.5.1 (step (6b)) of this specification.
- 'SD-SD Hash Calculation Process' is described in Section 3.5.1 (step (5a)) of this specification.

For the sake of readability, some flowcharts are added to processes as an example of process flow. A parenthetic number on a box in flowcharts shows which step an action or bifurcation described inside the box is done in.

In these processes, the following implementations are not prohibited in this specification.

- To update two or more CKIs from a CKMG at one process. For example, in deleting process, a device may delete all or selected CKIs in a CKMG at one process.
- Instead of an entire file, to merely read and write a part of the file, where the remaining part of the file is not necessary to do processes and not used for processes

6.1 User Key Recording Process

The Recording Device securely holds information associated with the User Key to be recorded. The information includes the Usage Rules for User Key given by a Service Provider or a Recording Device.

(1) Read the UKURMMG file from the SD Memory Card.

The Recording Device securely reads the UKURMMG file from the SD Memory Card using the Secure Read Process and holds it as the temporary UKURMMG image.

(2) Read or create a UKURMG.

(2.1) Select a UKURMG file that has at least one unused UKURE.

The Recording Device checks the UKURMG Used flag (UKURMG_USED) field in the temporary UKURMMG image. The Recording Device selects the first UKURMG file whose UKURMG Used flag is equal to '0b.' If all the UKURMG Used flags are equal to '1b,' the process shall be aborted.

(2.2) Read the selected UKURMG file from the SD Memory Card or create a temporary UKURMG image on the Recording Device.

The Recording Device checks whether the selected UKURMG file exists or not.

(a) When the selected UKURMG file exists, the Recording Device securely reads the selected UKURMG file from the SD Memory Card using the Secure Read Process and holds it as the temporary UKURMG image.

(b) When the selected UKURMG file does not exist, the Recording Device creates the selected UKURMG image on the Recording Device.

(3) Update the temporary UKURMG and UKURMMG image.

(3.1) Update the UKURE in the temporary UKURMG image.

The Recording Device checks the UKURE Used flag (UKURE_USED) in the temporary UKURMG image, and finds the first unused UKURE.

The Recording Device updates the unused UKURE in the temporary UKURMG image as follows:

The Trigger Bits is set to the value specified by Service Provider or the Recording Device.

The User Key Type is set the value specified by the Service Provider or the Recording Device.

The User Key ID is set the value specified by the Service Provider or the Recording Device.

The UR_UK_STARTDATE is set the value specified by the Service Provider or the Recording Device.

The UR_UK_ENDDATE is set the value specified by the Service Provider or the Recording Device.

The UR_UK_SPAN is set the value specified by the Service Provider or the Recording Device.

The Hash Value is set to the initial value (that is equal to '0000000000000000h').

The Check Value is set to (a) '0123456789ABCDEFh' in the case of C2-CPRM or (b) the high 8 bytes of CMAC(K_{u128}, high 40 bytes of the UR_U which has been set with the fields above) in the case of AES-CPRM.

After all the fields in the unused UKURE are set as above, the Recording Device encrypts the UKURE in the temporary UKURMG image using the UKURE Encryption process.

(3.2) Set the UKURE_USED in the temporary UKURMG image and the UKURMG_USED in the temporary UKURMMG image.

The Recording Device sets the UKURE Used flag associated with the updated UKURE in the temporary UKURMG image to '1b.'

In addition, when all the UKURE Used flags are equal to '1b' (all the UKUREs in the temporary UKURMG image have been used), the UKURMG Used flag associated with the selected UKURMG file in the temporary UKURMMG image shall be set to '1b.'

(4) Write the updated temporary UKURMG and UKURMMG image to the SD Memory Card.

The Recording Device securely writes the updated temporary UKURMG image held in the Recording Device as the updated UKURMG file to the SD Memory Card using the Secure Write Process. When the

selected UKURMG image was created in step (2.2), the updated temporary UKURMG image is written as the selected UKURMG file using the Secure Write Process.

In addition, if the temporary UKURMMG image is updated in step (3.2), the Recording Device securely writes the updated temporary UKURMMG image held in the Recording Device as the updated UKURMMG file to the SD Memory Card using the Secure Write Process.

To protect against the "Pull Card Attack," the Recording Device must assume that the Recording Process has been completely done, even if errors occur in step (4).

6.2 Content Key Recording Process

The Recording Device securely holds information associated with the Content Key to be recorded. The information includes the Usage Rules for Content either given by a Content Provider or according to the default rules defined by Compliance Rules and information which can identify the User Key corresponding to the Content Key (e.g. User Key ID).

(1) Read or create a Content Key Management (CKMG).

(1.1) Select the CKMG file associated with the Content Key to be recorded.

According to the identified User Key, the Recording Device selects the CKMG file.

(1.2) Read the selected CKMG file from the SD Memory Card.

The Recording Device checks whether the selected CKMG file exists or not.

(a) When the selected CKMG file exists, the Recording Device reads the selected CKMG file, holds it as the temporary CKMG image and checks the CKI Used flag (CKI_USED) field in the temporary CKMG image. If all the CKI Used flag are equal to '1b,' the process shall be aborted.

(b) When the selected CKMG file dose not exists, the Recording Device creates the selected CKMG image on the Recording Device.

(2) Determine the UKURMG file and UKURE associated with the Content Key to be recorded by using UKURE_SRN.

(2.1) Obtain UKURE_SRN.

The Recording Device obtains the UKURE_SRN ***s*** associated with the Content Key to be recorded.

(2.2) Determine the UKURMG file and UKURE associated with the Content Key to be recorded.

The Recording Device determines the UKURMG filename and the UKURE using the following formula:

$s = (n - 1) * 250 + m$ (***n***: UKURMG file number, m: UKURE number in a UKURMG)

$1 \le m \le 250,\ 1 \le n \le 256$

For example, when the UKURE_SRN is 1020, the UKURMG filename shall be "SDSD0005.KEY" (the fifth UKURMG file) and the UKURE shall be the twentieth entry in the "SDSD0005.KEY."

(3) Read the UKURMG file from the SD Memory Card.

The Recording Device securely reads the ***n***th UKURMG file from the SD Memory Card using Secure Read Process and holds it as the temporary UKURMG image.

Then, the Recording Device checks the ***m***th UKURE Used flag in the temporary UKURMG image. If it is equal to '0b,' the process shall be aborted.

Otherwise, the Recording Device obtains the ***m***th UKURE in the temporary UKURMG image.

(4) Check the UKURE in the temporary UKURMG image.

The Recording Device decrypts the UKURE using the UKURE Decryption process and securely holds it as the decrypted UKURE image.

In the case of C2-CPRM, the Recording Device checks this decrypted UKURE image as follows:

If the Check Value is not '0123456789ABCDEFh,' the process shall be aborted.

If the TB for AES is not equal to '0b,' this process shall be aborted.

If the TB for future use is not equal to '000000b,' the process shall be aborted.

If the User Key Type in the UKURE is equal to '0b,' the Recording Device checks the Hash Value in UKURE. If the stored Hash Value in UKURE is not equal to the calculated value from the temporary CKMG image using SD-SD Hash Calculation Process, the device shall execute CKMG Recovery Process described in section 5.7.

In the case of AES-CPRM, the Recording Device checks this decrypted UKURE image as follows:

If the Check Value is not equal to the high 8 bytes of calculated CMAC(K_{u128}, high 40 bytes of the UR_U), the process shall be aborted.

If the TB for AES is not equal to '1b', this process shall be aborted.

If the TB for future use is not equal to '000000b,' the process shall be aborted.

If the User Key Type in the UKURE is equal to '0b,' the Recording Device checks the Hash Value in UKURE. If the stored Hash Value in UKURE is not equal to the calculated value from the temporary CKMG image using SD-SD Hash Calculation Process, the device shall execute CKMG Recovery Process described in section 5.7.

(5) Update the temporary CKMG image.

(5.1) Update the CKI in the temporary CKMG image.

The Recording Device checks CKI Used flag (CKI_USED) in the temporary CKMG image, and finds the first unused CKI.

The Recording Device updates the unused CKI in the temporary CKMG image as follows:

The initial Move Control Information, Current Move Control Information, Copy Count Control Information, Initial Field Group, Current Field Group and Fixed Field Group of the CKI are set to the value either specified by the Content Provider or according to the default rules.

The Reserved for Profiles is set to the value either specified by the Content Provider or according to the default rules.

The Trigger Bits for Playback and The Trigger Bits for Copy/Move is set to '0000b.' when the Content Provider specifies the value other than '0000b,' this Trigger Bits field is set to the specific value.

The Check Value is set to (a) '0123456789ABCDEFh' in the case of C2-CPRM or (b).the result of calculated CMAC(K_{c128}, high 40 bytes of the UR_C) in the case of AES-CPRM.

After all the fields in the unused CKI are set as above, the Recording Device encrypts the CKI in the temporary CKMG image using the CKI Encryption process.

(5.2) Set the CKI_USED in the temporary CKMG image.

The Recording Device sets the CKI Used flag associated with the updated CKI in the temporary CKMG image to '1b.'

(6) Check the User Key Type in UKURE.

The Recording Device checks the User Key Type which is obtained in step (4).

(a) When the User Key Type is equal to '0b,' go to step (7).

(b) When the User Key Type is equal to '1b,' go to step (9).

(7) Update the temporary UKURMG image.

The Recording Device updates the UKURE in the temporary UKURMG image. The Hash Value in the UKURE is set to the value that is calculated for the temporary CKMG image using SD-SD Hash Calculation Process. Then the Recording Device encrypts the UKURE in the temporary UKURMG image using the UKURE Encryption process.

(8) Write the updated temporary UKURMG and CKMG images to the SD Memory Card.

The Recording Device securely writes the updated temporary UKURMG and CKMG images held in the Recording Device as the updated UKURMG and CKMG files to the SD Memory Card using the CKMG Update Process in described in section 5.7. Then this process is completed.

To protect against the "Pull Card Attack," the Recording Device must assume that the Recording Process has been completely done, even if errors occur in step (8)

(9) Write the updated temporary CKMG image to the SD Memory Card.

The Recording Device writes the updated temporary CKMG file image held in the Recording Device as the updated CKMG file to the SD Memory Card.

To protect against the "Pull Card Attack," the Recording Device must assume that the Recording Process has been completely done, even if errors occur in step (9).

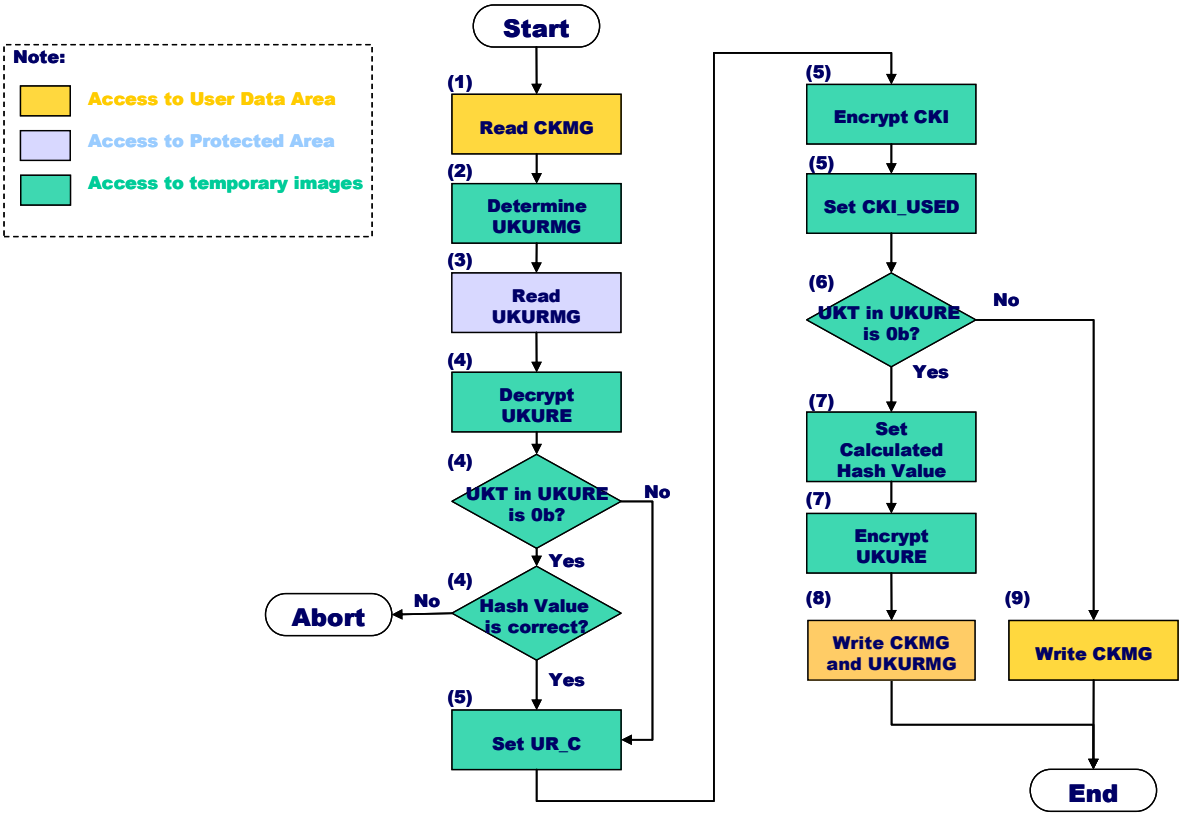


Figure 6-1 Flowchart of Content Key Recording Process

6.3 Encrypted Content Key Recording Process

The Recording Device securely holds information associated with the encrypted Content Key to be recorded. The information includes the Usage Rules for Content given by a Content Provider and information which can identify the User Key corresponding to the encrypted Content Key (e.g. User Key ID).

The Content Key has already encrypted with the corresponding User Key.

(1) Read or create a Content Key Management (CKMG).

(1.1) Select the CKMG file associated with the Content Key to be recorded.

According to the identified User Key, the Recording Device selects the CKMG file.

(1.2) Read the selected CKMG file from the SD Memory Card.

The Recording Device checks whether the selected CKMG file exists or not.

(a) When the selected CKMG file exists, the Recording Device reads the selected CKMG file, holds it as the temporary CKMG image and checks the CKI Used flag (CKI_USED) field in the temporary CKMG image. If all the CKI Used flag are equal to '1b,' the process shall be aborted.

(b) When the selected CKMG file dose not exists, the Recording Device creates the selected CKMG image on the Recording Device.

(2) Read the User Key Type associated with the Content Key to be recorded in the User Data Area.

(2.1) Obtain the User Key Type associated with the Content Key to be recorded from the User Data Area.

The Recording Device obtains the User Key Type associated with the Content Key to be recorded from the User Data Area.

(2.2) Check the User Key Type.

The Recording Device checks the User Key Type.

(a) When the User Key Type is equal to '0b,' then go to step (3).

(b) When the User Key Type is equal to '1b,' then go to step (6).

(3) Determine the UKURMG file and UKURE associated with the Content Key to be recorded by using UKURE_SRN.

(3.1) Obtain UKURE_SRN.

The Recording Device obtains the UKURE_SRN ***s*** associated with the Content Key to be recorded.

(3.2) Determine the UKURMG file and UKURE associated with the Content Key to be recorded.

The Recording Device determines the UKURMG filename and the UKURE using the following formula:

$s = (n - 1) * 250 + m$ (***n***: UKURMG file number, ***m***: UKURE number in a UKURMG)

$1 \leq m \leq 250$, $1 \leq n \leq 256$

For example, when the UKURE_SRN is 1020, the UKURMG filename shall be "SDSD0005.KEY" (the fifth UKURMG file) and the UKURE shall be the twentieth entry in the "SDSD0005.KEY."

(4) Read the UKURMG file from the SD Memory Card.

The Recording Device securely reads the ***n***th UKURMG file from the SD Memory Card using Secure Read Process and holds it as the temporary UKURMG image.

Then, the Recording Device checks the ***m***th UKURE Used flag in the temporary UKURMG image. If it is equal to '0b,' the process shall be aborted.

Otherwise, the Recording Device obtains the ***m***th UKURE in the temporary UKURMG image.

(5) Check the UKURE in the temporary UKURMG image.

The Recording Device decrypts the UKURE using the UKURE Decryption process and securely holds it as the decrypted UKURE image.

In the case of C2-CPRM, the Recording Device checks this decrypted UKURE image as follows:

If the Check Value is not '0123456789ABCDEFh,' the process shall be aborted.

If the TB for AES is not equal to '0b,' this process shall be aborted.

If the TB for future use is not equal to '000000b,' the process shall be aborted.

If the stored Hash value in UKURE is not equal to the calculated value from the temporary CKMG image using SD-SD Hash Calculation Process, the device shall execute CKMG Recovery Process described in section 5.7.

In the case of AES-CPRM, the Recording Device checks this decrypted UKURE image as follows:

If the Check Value is not equal to the value of calculated CMAC(K_{u128}, high 40 bytes of the UR_U), the process shall be aborted.

If the TB for AES is not equal to '1b,' this process shall be aborted.

If the TB for future use is not equal to '000000b,' the process shall be aborted.

If the User Key Type in the UKURE is equal to '0b,' the Playback Device checks the Hash Value in UKURE. If the stored Hash Value in UKURE is not equal to the calculated value from the temporary CKMG image using SD-SD Hash Calculation Process, the device shall execute CKMG Recovery Process described in section 5.7.

(6) Update the temporary CKMG image.

The Recording Device sets the CKI Used flag associated with the updated CKI in the temporary CKMG image to '1b.'

(7) Check the User Key Type associated with the Content Key to be recorded in the User Data Area.

The Recording Device checks the User Key Type which is obtained in step (2.1).

(a) When the User Key Type is equal to '0b,' then go to step (8).

(b) When the User Key Type is equal to '1b,' then go to step (10).

(8) Update the temporary UKURMG image.

The Recording Device updates the UKURE in the temporary UKURMG image. The Hash Value in the UKURE is set to the value that is calculated for the temporary CKMG image using SD-SD Hash Calculation Process. Then the Recording Device encrypts the UKURE in the temporary UKURMG image using the UKURE Encryption process.

(9) Write the updated temporary UKURMG and CKMG images to the SD Memory Card.

The Recording Device securely writes the updated temporary UKURMG and CKMG images held in the Recording Device as the updated UKURMG and CKMG files to the SD Memory Card using the CKMG Update Process in described in section 5.7. Then this process is completed.

To protect against the "Pull Card Attack," the Recording Device must assume that the Recording Process has been completely done, even if errors occur in step (9)

(10) Write the updated temporary CKMG image to the SD Memory Card.

The Recording Device writes the updated temporary CKMG file image held in the Recording Device as the updated CKMG file to the SD Memory Card.

To protect against the "Pull Card Attack," the Recording Device must assume that the Recording Process has been completely done, even if errors occur in step (10).

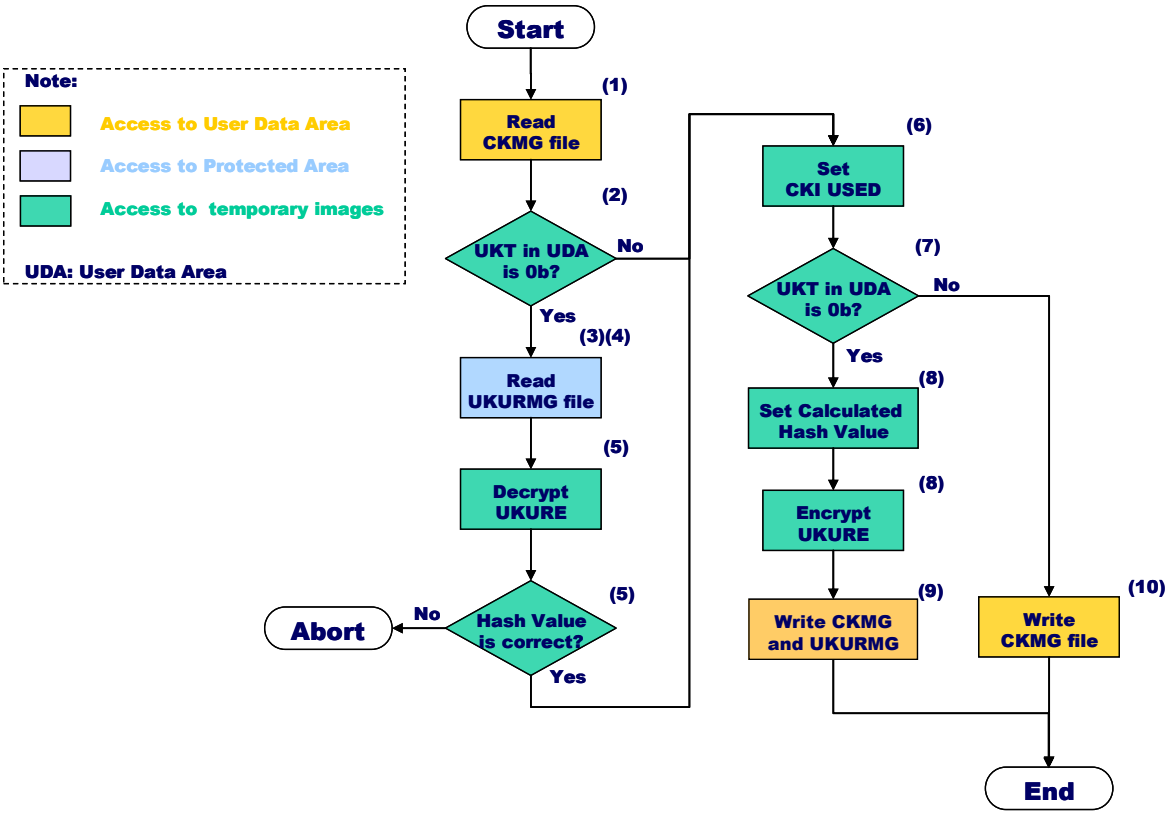


Figure 6-2 Flowchart of Encrypted Content Key Recording Process

6.4 User Key Erasing Process

(1) Determine the UKURMG file and UKURE associated with the User Key to be erased.

(1.1) Obtain UKURE_SRN.

The Erasing Device obtains the UKURE_SRN ***s*** associated with the SD-SD content to be erased.

(1.2) Determine the UKURMG file and UKURE associated with the SD-SD content to be erased.

The Erasing Device determines the UKURMG filename and the UKURE using the following formula:

$s = (n - 1) * 250 + m$ (***n***: UKURMG file number, ***m***: UKURE number in a UKURMG)

$1 \leq m \leq 250$, $1 \leq n \leq 256$

For example, when the UKURE_SRN is 1020, the UKURMG filename shall be "SDSD0005.KEY" (the fifth UKURMG file) and the UKURE shall be the twentieth entry in the "SDSD0005.KEY."

(2) Read the UKURMG file from the SD Memory Card.

The Erasing Device securely reads the ***n***th UKURMG file from the SD Memory Card using Secure Read Process and holds it as the temporary UKURMG image.

Then, the Erasing Device checks the ***m***th UKURE Used flag in the temporary UKURMG image. If it is equal to '0b,' the process shall be aborted.

Otherwise, the Erasing Device obtains the ***m***th UKURE in the temporary UKURMG image.

(3) Update the temporary UKURMG and UKURMMG image.

The Erasing Device overwrites this UKURE in the temporary UKURMG image with "the value for delete (random number)."

The Erasing Device shall set the UKURE Used flag associated with the UKURE in the temporary UKURMG image to '0b.'

In addition, the Erasing Device checks all the UKURE Used flags in the temporary UKURMG image.

(a) When all the UKURE Used flags are equal to '0b,' the Erasing Device deletes the selected UKURMG file from the Protected Area of the SD Memory Card, and then considers this process to be successfully terminated.

(b) When all the UKURE Used flags other than the one associated with the overwritten UKURE are equal to '1b,' the Erasing Device securely reads the UKURMMG file from the SD Memory Card using Secure Read Process and holds it as the temporary UKURMMG image. Then the Erasing Device sets the ***n***th UKURMG Used flag in the temporary UKURMMG image to '0b.'

(4) Write the updated temporary UKURMG and UKURMMG image to the SD Memory Card.

The Erasing Device securely writes the updated temporary UKURMG image as the updated UKURMG file to the SD Memory Card using the Secure Write Process. Then the Erasing Device securely reads the updated UKURMG file from the SD Memory Card using the Secure Read Process and verifies that the value of the ***m***th UKURE in the UKURMG file is equal to "the value for delete (random number)" used in step (3). If the verification of the UKURMG file fails, the Erasing Device shall abort this process.

In addition, if the temporary UKURMMG image was updated in step (3b), the Erasing Device securely writes the updated temporary UKURMMG image as the updated UKURMMG file to the SD Memory Card using the Secure Write Process.

6.5 Content Key Erasing Process

(1) Determine the CKMG file and CKI associated with the Content Key to be erased.

The Erasing Device determines the CKMG filename and the CKI.

(2) Read the CKMG file from the SD Memory Card.

The Erasing Device reads the CKMG file from the SD Memory Card and holds it as the temporary CKMG image. Then, the Erasing Device checks the corresponding CKI Used flag in the temporary CKMG image. If it is equal to '0b,' the process shall be aborted. Otherwise, the Erasing Device obtains the selected CKI in the temporary CKMG image.

(3) Read the User Key Type associated with the Content Key to be erased in the User Data Area.

(3.1) Obtain the User Key Type associated with the Content Key to be erased in the User Data Area.

The Erasing Device obtains the User Key Type associated with the Content Key to be erased in the User Data Area.

(3.2) Check the User Key Type.

The Erasing Device checks the User Key Type.

(a) When the User Key Type is equal to '0b,' go to step (4).

(b) When the User Key Type is equal to '1b,' go to step (7).

(4) Determine the UKURMG file and UKURE associated with the Content Key to be erased.

(4.1) Obtain UKURE_SRN.

The Erasing Device obtains the UKURE_SRN ***s*** associated with the SD-SD content to be erased.

(4.2) Determine the UKURMG file and UKURE associated with the SD-SD content to be erased.

The Erasing Device determines the UKURMG filename and the UKURE using the following formula:

$s = (n - 1) * 250 + m$ (n: UKURMG file number, m: UKURE number in a UKURMG)

$1 \leq m \leq 250$, $1 \leq n \leq 256$

For example, when the UKURE_SRN is 1020, the UKURMG filename shall be "SDSD0005.KEY" (the fifth UKURMG file) and the UKURE shall be the twentieth entry in the "SDSD0005.KEY."

(5) Read the UKURMG file from the SD Memory Card.

The Erasing Device securely reads the ***n***th UKURMG file from the SD Memory Card using Secure Read Process and holds it as the temporary UKURMG image.

Then, The Erasing Device checks the ***m***th UKURE Used flag in the temporary UKURMG image. If it is equal to '0b,' the process shall be aborted.

Otherwise, The Erasing Device obtains the ***m***th UKURE in the temporary UKURMG image.

(6) Check the UKURE in the temporary UKURMG image.

The Erasing Device decrypts the UKURE using the UKURE Decryption process and securely holds it as the decrypted UKURE image. The Erasing Device shall validate the Hash Value in the image. If the Hash Value in the decrypted UKURE is not equal to the Hash Value from the temporary CKMG image using SD-SD Hash Calculation Process, the device shall execute CKMG Recovery Process described in section 5.7.

(7) Update the temporary CKMG image.

The Erasing Device shall set the CKI Used flag associated with the CKI in the temporary CKMG image to '0b.'

(8) Check the User Key Type associated with the Content Key to be erased in the User Data Area.

The Erasing Device checks the User Key Type which is obtained in step (3.1).

(a) When the User Key Type is equal to '0b,' go to step (9).

(b) When the User Key Type is equal to '1b,' go to step (11).

(9) Update the temporary UKURMG image.

The Erasing Device updates the UKURE in the temporary UKURMG image. The Hash Value in the UKURE is set to the value that is calculated for the temporary CKMG image using SD-SD Hash Calculation Process. Then the Erasing Device encrypts the UKURE in the temporary UKURMG image using the UKURE Encryption process.

(10) Write the updated temporary UKURMG and CKMG images to the SD Memory Card.

The Erasing Device securely writes the updated temporary UKURMG and CKMG images held in the Erasing Device as the updated UKURMG and CKMG files to the SD Memory Card using the CKMG Update Process in described in section 5.7. Then this process is completed.

(11) Write the updated temporary CKMG image to the SD Memory Card.

The Erasing Device securely writes the updated temporary CKMG image as the updated CKMG file to the SD Memory Card.

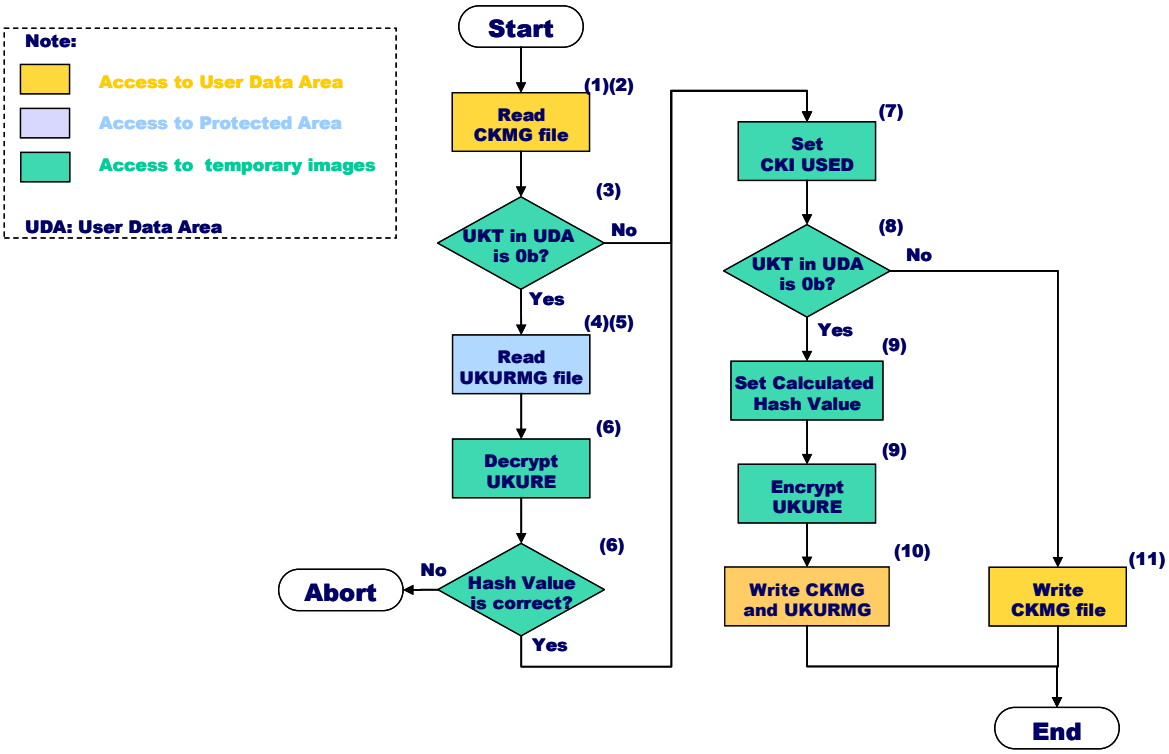


Figure 6-3 Flowchart of Content Key Erasing Process

6.6 Content Key Copy Process I (from SD Memory Card to Host)

(1) Determine the CKMG file and CKI associated with the Content Key to be copied.

The Destination Device determines the CKMG filename and the CKI.

(2) Read the CKMG file from the SD Memory Card.

The Destination Device reads the CKMG file from the SD Memory Card and holds it as the temporary CKMG image.

Then, the Destination Device checks the corresponding CKI Used flag in the temporary CKMG image. If it is equal to '0b,' the process shall be aborted.

Otherwise, the Destination Device obtains the selected CKI in the temporary CKMG image.

(3) Determine the UKURMG file and UKURE associated with the Content Key to be copied by using UKURE_SRN.

(3.1) Obtain UKURE_SRN.

The Destination Device obtains the UKURE_SRN ***s*** associated with the Content Key to be copied.

(3.2) Determine the UKURMG file and UKURE associated with the Content Key to be copied.

The Destination Device determines the UKURMG filename and the UKURE using the following formula:

$s = (n - 1) * 250 + m$ (***n***: UKURMG file number, ***m***: UKURE number in a UKURMG)

$1 \leq m \leq 250$, $1 \leq n \leq 256$

For example, when the UKURE_SRN is 1020, the UKURMG filename shall be "SDSD0005.KEY" (the fifth UKURMG file) and the UKURE shall be the twentieth entry in the "SDSD0005.KEY."

(4) Read the UKURMG file from the SD Memory Card.

The Destination Device securely reads the ***n***th UKURMG file from the SD Memory Card using Secure Read Process and holds it as the temporary UKURMG image.

Then, the Destination Device checks the ***m***th UKURE Used flag in the temporary UKURMG image. If it is equal to '0b,' the process shall be aborted.

Otherwise, the Destination Device obtains the ***m***th UKURE in the temporary UKURMG image.

(5) Check the UKURE in the temporary UKURMG image.

The Destination Device decrypts the UKURE using the UKURE Decryption process and securely holds it as the decrypted UKURE image.

In the case of C2-CPRM, the Destination Device checks this decrypted UKURE image as follows:

If the Check Value is not '0123456789ABCDEFh,' the process shall be aborted.

If the TB for AES is not equal to '0b,' this process shall be aborted.

If the TB for future use is not equal to '000000b,' the process shall be aborted.

If the User Key Type in the UKURE is equal to '0b,' the Destination Device checks the Hash Value in UKURE. If the stored Hash Value in UKURE is not equal to the calculated value from the temporary CKMG image using SD-SD Hash Calculation Process, the device shall execute CKMG Recovery Process described in section 5.7.

If the User Key Type in the UKURE is equal to '1b,' the process shall be aborted.

In the case of AES-CPRM, the Destination Device checks this decrypted UKURE image as follows:

If the Check Value is not equal to the value of calculated CMAC(K_{u128}, high 40 bytes of the UR_U), the process shall be aborted.

If the TB for AES is not equal to '1b,' this process shall be aborted.

If the TB for future use is not equal to '000000b,' the process shall be aborted.

If the User Key Type in the UKURE is equal to '0b,' the Playback Device checks the Hash Value in UKURE. If the stored Hash Value in UKURE is not equal to the calculated value from the temporary CKMG image using SD-SD Hash Calculation Process, the device shall execute CKMG Recovery Process described in section 5.7.

(6) Check the CKI in the temporary CKMG image.

The Destination Device decrypts the CKI using the CKI Decryption process and securely holds it as the decrypted CKI image.

In the case of C2-CPRM, the Destination Device checks this decrypted CKI image as follows:

If the Check Value is not '0123456789ABCDEFh,' the process shall be aborted.

If the CMTB for AES is not equal to '0b,' this process shall be aborted.

If the CMTB for future use is not equal to '000b,' the process shall be aborted.

If the Copy Count Control Information is equal to '0000b,' the process shall be aborted.

After all the fields in the CKI are checked as above, the Destination Device additionally checks this decrypted CKI image. If the Copy Count Control Information is equal to '1111b,' then go to step (11).

In the case of AES-CPRM, the Destination Device checks this decrypted CKI image as follows:

If the Check Value is not equal to the value of calculated CMAC(K_{c128}, high 40 bytes of the UR_C), the process shall be aborted.

If the CMTB for AES is not equal to '1b,' this process shall be aborted.

If the CMTB for future use is equal to '000b,' the process shall be aborted.

If the Copy Count Control Information is equal to '0000b,' the process shall be aborted.

After all the fields in the CKI are checked as above, the Destination Device additionally checks this decrypted CKI image. If the Copy Count Control Information is equal to '1111b,' then go to step (11).

(7) Update the CKI in the temporary CKMG image.

The Destination Device decrement the value of Copy Control Information of the decrypted CKI image.

Then, the Destination Device encrypts this decrypted CKI image using the CKI Encryption process, and sets the CKI in the temporary CKMG image to the resulting value.

(8) Update the temporary UKURMG image.

The Destination Device updates the UKURE in the temporary UKURMG image. The Hash Value in the UKURE is set to the value that is calculated for the temporary CKMG image using SD-SD Hash Calculation Process. Then the Destination Device encrypts the UKURE in the temporary UKURMG image using the UKURE Encryption process.

(9) Write the updated temporary UKURMG and CKMG images to the SD Memory Card.

The Destination Device securely writes the updated temporary UKURMG and CKMG images held in the Destination Device as the updated UKURMG and CKMG files to the SD Memory Card using the CKMG Update Process in described in section 5.7. Then the Destination Device securely reads the updated UKURMG file from the SD Memory Card using Secure Read Process and verifies that the update ***m***th UKURE in the UKURMG file has completed successfully.

(10) Update the Usage Rules for copied content on the Destination Device.

The Destination Device temporary holds the decrypted CKI image as the associated the Content Key and Usage Rules for copied content on the Destination Device.

The Destination Device updates those Usage Rules for copied content fields as follows:

The Copy Count Control Information is set to '0000b.'

The Current Move Control Information field is set to the same value as that of the Initial Move Control Information field of the decrypted CKI.

Each field in the Current Field Group is set to the same value as that of each corresponding field in the Initial Field Group of the decrypted CKI. (Here, the Current First Playback Flag in the Current Start Date of Playback is set to be '0b.')

When all of the above steps are executed successfully, the Destination Device securely holds the User Key, the Content Key, the Usage Rules for User Key, the Usage Rules for Content and the Usage Rules for copied content.

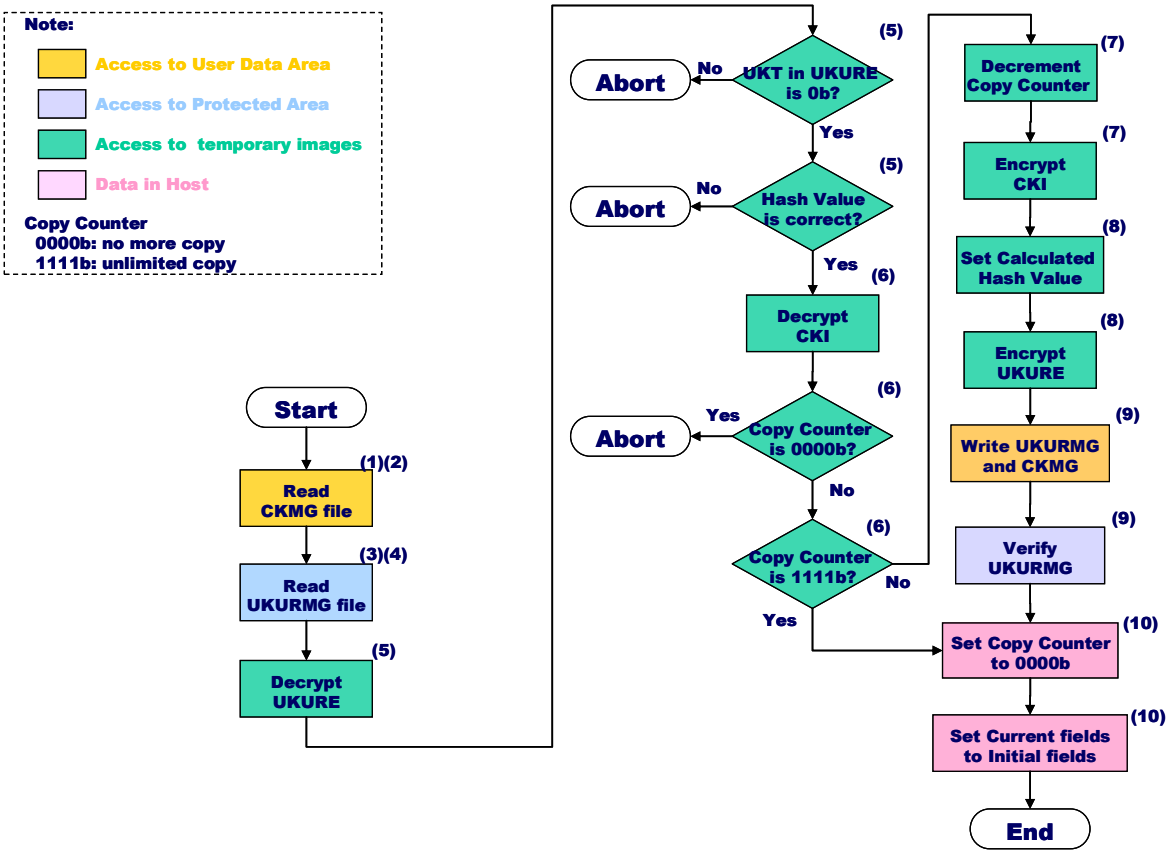


Figure 6-4 Flowchart of Content Key Copy Process I

6.7 Content Key Copy Process II (from Host to SD Memory Card)

The Source Device securely holds information associated with the Content Key to be copied. The information includes the Usage Rules for Content given by a Content Provider and information which can identify the User Key corresponding to the Content Key (e.g. User Key ID).

(1) Check the Usage Rules for Content on the Source Device.

The Source Device checks the Usage Rules for Content securely held in it

If the Copy Count Control Information is equal to '0000b,' then the process shall be aborted.

(2) Read or create a Content Key Management (CKMG).

(2.1) Select the CKMG file associated with the Content Key to be copied.

According to the identified User Key, the Recording Device selects the CKMG file.

(2.2) Read the selected CKMG file from the SD Memory Card.

The Source Device checks whether the selected CKMG file exists or not.

(a) When the selected CKMG file exists, the Source Device reads the selected CKMG file, holds it as the temporary CKMG image and checks the CKI Used flag (CKI_USED) field in the temporary CKMG image. If all the CKI Used flag are equal to '1b,' the process shall be aborted.

(b) When the selected CKMG file dose not exists, the Source Device creates the selected CKMG image on the Source Device.

(3) Determine the UKURMG file and UKURE associated with the Content Key to be copied by using UKURE_SRN.

(3.1) Obtain UKURE_SRN.

The Source Device obtains the UKURE_SRN ***s*** associated with the Content Key to be copied.

(3.2) Determine the UKURMG file and UKURE associated with the Content Key to be copied.

The Source Device determines the UKURMG filename and the UKURE using the following formula:

$s = (n - 1) * 250 + m$ (***n***: UKURMG file number, ***m***: UKURE number in a UKURMG)

$1 \leq m \leq 250$, $1 \leq n \leq 256$

For example, when the UKURE_SRN is 1020, the UKURMG filename shall be "SDSD0005.KEY" (the fifth UKURMG file) and the UKURE shall be the twentieth entry in the "SDSD0005.KEY."

(4) Read the UKURMG file from the SD Memory Card.

The Source Device securely reads the ***n***th UKURMG file from the SD Memory Card using Secure Read Process and holds it as the temporary UKURMG image.

Then, the Source Device checks the ***m***th UKURE Used flag in the temporary UKURMG image. If it is equal to '0b,' the process shall be aborted.

Otherwise, the Source Device obtains the ***m***th UKURE in the temporary UKURMG image.

(5) Check the UKURE in the temporary UKURMG image.

The Source Device decrypts the UKURE using the UKURE Decryption process and securely holds it as the decrypted UKURE image.

In the case of C2-CPRM, the Source Device checks this decrypted UKURE image as follows:

If the Check Value is not '0123456789ABCDEFh,' the process shall be aborted.

If the TB for AES is not equal to '0b,' this process shall be aborted.

If the TB for future use is not equal to '000000b,' the process shall be aborted.

If the User Key Type in the UKURE is equal to '0b,' the Source Device checks the Hash Value in UKURE. If the stored Hash Value in UKURE is not equal to the calculated value from the temporary CKMG image using SD-SD Hash Calculation Process, the device shall execute CKMG Recovery Process described in section 5.7.

In the case of AES-CPRM, the Source Device checks this decrypted UKURE image as follows:

If the Check Value is not equal to the value of calculated CMAC(K_{u128}, high 40 bytes of the UR_U), the process shall be aborted.

If the TB for AES is not equal to '1b,' this process shall be aborted.

If the TB for future use is not equal to '000000b,' the process shall be aborted.

If the User Key Type in the UKURE is equal to '0b,' the Playback Device checks the Hash Value in UKURE. If the stored Hash Value in UKURE is not equal to the calculated value from the temporary CKMG image using SD-SD Hash Calculation Process, the device shall execute CKMG Recovery Process described in section 5.7.

(6) Update the temporary CKMG image.

(6.1) Update the CKI in the temporary CKMG image.

The Source Device checks CKI Used flag (CKI_USED) in the temporary CKMG image, and finds the first unused CKI.

The Source Device updates the unused CKI in the temporary CKMG image as follows:

The Copy Count Control Information of the CKI is set to '0000b.'

The Initial Move Control Information field of the CKI is set to the same value as that of the Initial Move Control Information field of the Usage Rules for Content held in the Source Device. The Current Move Control Information field of the CKI is set to the same value as that of the Initial Move Control Information field of the Usage Rules for Content held in the Source Device.

Each field in the Initial Field Group of the CKI is set to the same value as that of each corresponding field in the Initial Field Group of the Usage Rules for Content held in the Source Device.

Each field in the Current Field Group of the CKI is set to the same value as that of each corresponding field in the Initial Field Group of the Usage Rules for Content held in the Source Device. (Here, the Current First Playback Flag in the Current Start Date of Playback of the CKI is set to be '0b.')

Each field in the Fixed Field Group of the CKI is set to the same value as that of each corresponding field in the Fixed Field Group of the Usage Rules for Content held in the Source Device.

The Check Value of the CKI is set to '0123456789ABCDEFh.'

After all the fields in the unused CKI are set as above, the Source Device encrypts the CKI in the temporary CKMG image using the CKI Encryption process.

(6.2) Set the CKI_USED in the temporary CKMG image.

The Source Device sets the CKI Used flag associated with the updated CKI in the temporary CKMG image to '1b.'

(7) Check the User Key Type associated with the Content Key to be copied in the UKURE.

The Source Device checks the User Key Type which is obtained in step (5).

(a) When the User Key Type is equal to '0b,' go to step (8).

(b) When the User Key Type is equal to '1b,' go to step (9).

(8) Update the temporary UKURMG image

The Source Device updates the UKURE in the temporary UKURMG image. The Hash Value in the UKURE is set to the value that is calculated for the temporary CKMG image using SD-SD Hash Calculation Process. Then the Source Device encrypts the UKURE in the temporary UKURMG image using the UKURE Encryption process.

(9) Update the Usage Rules for Content on the Source Device

If the Copy Count Control Information held in the Source Device is not equal to '1111b,' the Source Device decrements the value of the Copy Count Control Information held in it. If User Key Type is equal to '1b' in step (7), then go to step (11).

(10) Write the updated temporary UKURMG and CKMG images to the SD Memory Card.

The Source Device securely writes the updated temporary UKURMG and CKMG images held in the Source Device as the updated UKURMG and CKMG files to the SD Memory Card using the CKMG Update Process in described in section 5.7. Then this process is completed.

To protect against the "Pull Card Attack," the Source Device must assume that the Source Process has been completely done, even if errors occur in step (10)

(11) Write the updated temporary CKMG image to the SD Memory Card.

The Source Device writes the updated temporary CKMG file image held in the Source Device as the updated CKMG file to the SD Memory Card.

To protect against the "Pull Card Attack," the Source Device must assume that the Source Process has been completely done, even if errors occur in step (11).

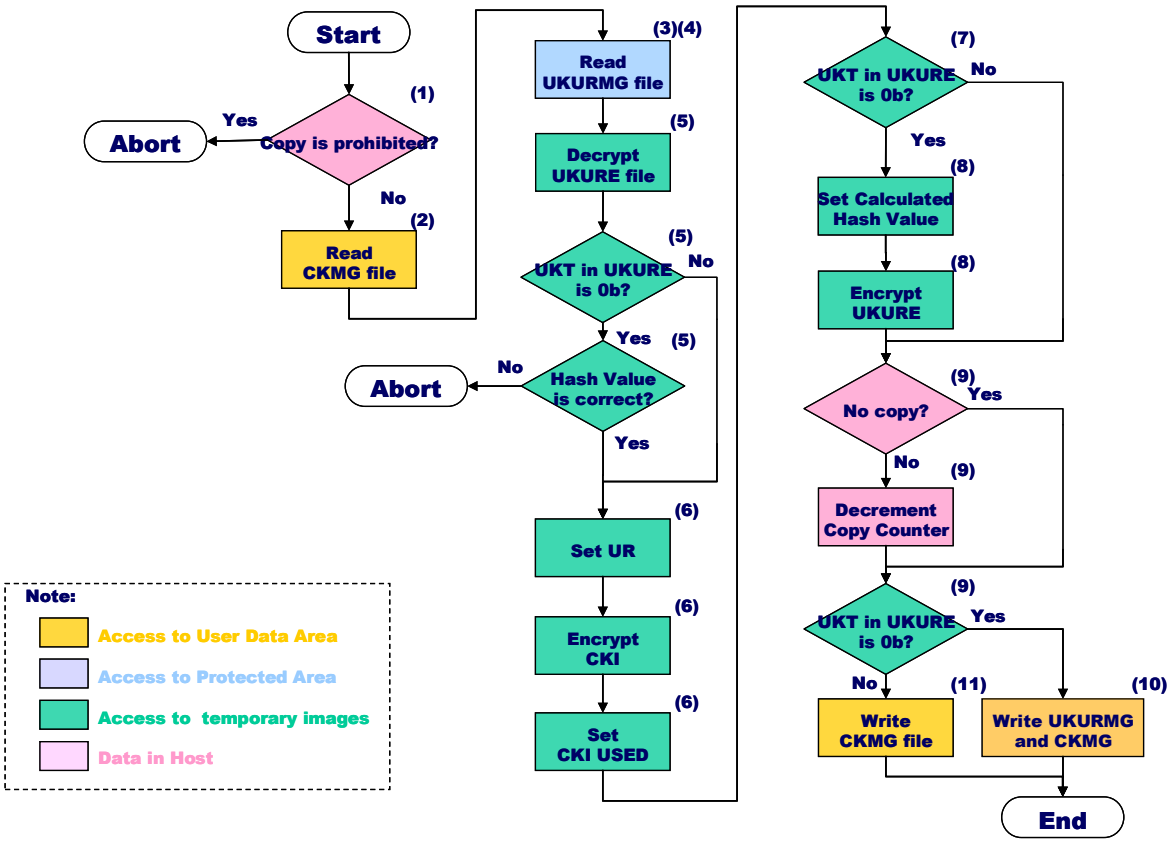


Figure 6-5 Flowchart of Content Key Copy Process II

6.8 Content Key Copy Process III (from SD Memory Card to SD Memory Card)

In SD-SD specification, compliant devices may copy a Content Key from an SD Memory Card to another SD Memory Card. The device can copy a Content Key to execute the following two steps:

(1) The device copies a Content Key from SD Memory Card to the device in accordance with Content Key Copy Process I.

(2) The device writes a Content Key from the device to the destination SD Memory Card in accordance with Content Key Recording Process.

The device shall hold the Content Key and its usage rule securely and shall not use the Content Key to decrypt any content before step (2) is completed.

On the condition that a device has two SD Memory Card interfaces and two SD Memory Cards are inserted to the interfaces in the device respectively, when the device copies a Content Key from an SD Memory Card to another SD Memory Card, the device can copy the Content Key to execute the following four steps based on two processes, Content Key Copy Process I and II:

(1) The device executes from step (1) to (8) of Content Key Copy Process I,
In this step, the device securely reads the files including the original Content Key and its usage rule. The device also prepares to update the Content Key and its Usage Rule.

(2) The device executes from step (2) to (8) of Content Key Copy Process II.
In this step, the device securely reads the files where the copied Content Key is to be stored. The device also prepares to write the Content Key and its Usage Rule.

(3) The device executes step (9) of Content Key Copy Process I.
In this step, the device securely writes the files including the original Content Key and its Usage Rule.

(4) The device executes step (10) and (11) of Content Key Copy Process II.
In this step, the device securely writes the files including the copies of the Content Key.

Note that completion of step (1) and (3) herein is regarded as step (1) and (9) in Content Key Copy Process II respectively. To protect against the "Pull Card Attack," the device must assume that this Copy Process III has been completely done, even if errors occur in step (4) herein.

6.9 Content Key Move Process I (from SD Memory Card to Host)

(1) Determine the CKMG file and CKI associated with the Content Key to be moved.

The Destination Device determines the CKMG filename and the CKI.

(2) Read the CKMG file from the SD Memory Card.

The Destination Device reads the CKMG file from the SD Memory Card and holds it as the temporary CKMG image.

Then, the Destination Device checks the corresponding CKI Used flag in the temporary CKMG image. If it is equal to '0b,' the process shall be aborted.

Otherwise, the Destination Device obtains the selected CKI in the temporary CKMG image.

(3) Determine the UKURMG file and UKURE associated with the Content Key to be moved by using UKURE_SRN.

(3.1) Obtain UKURE_SRN.

The Destination Device obtains the UKURE_SRN ***s*** associated with the Content Key to be moved.

(3.2) Determine the UKURMG file and UKURE associated with the Content Key to be moved.

The Destination Device determines the UKURMG filename and the UKURE using the following formula:

$s = (n - 1) * 250 + m$ (***n***: UKURMG file number, ***m***: UKURE number in a UKURMG)

$1 \leq m \leq 250$, $1 \leq n \leq 256$

For example, when the UKURE_SRN is 1020, the UKURMG filename shall be "SDSD0005.KEY" (the fifth UKURMG file) and the UKURE shall be the twentieth entry in the "SDSD0005.KEY."

(4) Read the UKURMG file from the SD Memory Card.

The Destination Device securely reads the ***n***th UKURMG file from the SD Memory Card using Secure Read Process and holds it as the temporary UKURMG image.

Then, the Destination Device checks the ***m***th UKURE Used flag in the temporary UKURMG image. If it is equal to '0b,' the process shall be aborted.

Otherwise, the Destination Device obtains the ***m***th UKURE in the temporary UKURMG image.

(5) Check the UKURE in the temporary UKURMG image.

The Destination Device decrypts the UKURE using the UKURE Decryption process and securely holds it as the decrypted UKURE image.

In the case of C2-CPRM, the Destination Device checks this decrypted UKURE image as follows:

If the Check Value is not '0123456789ABCDEFh,' the process shall be aborted.

If the TB for AES is not equal to '0b,' this process shall be aborted.

If the TB for future use is not equal to '000000b,' the process shall be aborted.

If the User Key Type in the UKURE is equal to '0b,' the Destination Device checks the Hash Value in UKURE. If the stored Hash Value in UKURE is not equal to the calculated value from the temporary CKMG image using SD-SD Hash Calculation Process, the device shall execute CKMG Recovery Process described in section 5.7.

If the User Key Type in the UKURE is equal to '1b,' the process shall be aborted.

In the case of AES-CPRM, the Recording Device checks this decrypted UKURE image as follows:

If the Check Value is not equal to the value of calculated CMAC(K_{u128}, high 40 bytes of the UR_U), the process shall be aborted.

If the TB for AES is not equal to '1b,' this process shall be aborted.

If the TB for future use is not equal to '000000b,' the process shall be aborted.

If the User Key Type in the UKURE is equal to '0b,' the Playback Device checks the Hash Value in UKURE. If the stored Hash Value in UKURE is not equal to the calculated value from the temporary CKMG image using SD-SD Hash Calculation Process, the device shall execute CKMG Recovery Process described in section 5.7.

(6) Check the CKI in the temporary CKMG image.

The Destination Device decrypts the CKI using the CKI Decryption process and securely holds it as the decrypted CKI image.

In the case of C2-CPRM, the Destination Device checks this decrypted CKI image as follows:

If the Check Value is not '0123456789ABCDEFh,' the process shall be aborted.

If the CMTB for AES is not equal to '0b,' this process shall be aborted.

If the CMTB for future use is not equal to '000b,' the process shall be aborted.

If the Current Move Control Information is equal to '00b,' the process shall be aborted.

In the case of AES-CPRM, the Destination Device checks this decrypted CKI image as follows:

If the Check Value is not equal to the value of calculated CMAC(K_{c128}, high 40 bytes of the UR_C), the process shall be aborted.

If the CMTB for AES is not equal to '1b,' this process shall be aborted.

If the CMTB for future use is equal to '000b,' the process shall be aborted.

If the Current Move Control Information is equal to '00b,' the process shall be aborted.

(7) Update the temporary CKMG image.

The Destination Device shall set the CKI Used flag associated with the CKI in the temporary CKMG image to '0b.'

(8) Update the temporary UKURMG image.

The Destination Device updates the UKURE in the temporary UKURMG image. The Hash Value in the UKURE is set to the value that is calculated for the temporary CKMG image using SD-SD Hash

Calculation Process. Then the Destination Device encrypts the UKURE in the temporary UKURMG image using the UKURE Encryption process.

(9) Write the updated temporary UKURMG and CKMG images to the SD Memory Card.

The Destination Device securely writes the updated temporary UKURMG and CKMG images held in the Destination Device as the updated UKURMG and CKMG files to the SD Memory Card using the CKMG Update Process in described in section 5.7. Then the Destination Device securely reads the updated UKURMG file from the SD Memory Card using Secure Read Process and verifies that the update ***m***th UKURE in the UKURMG file has completed successfully.

(10) Update the Usage Rules for Content on the Destination Device.

The Destination Device temporary holds the decrypted CKI image as the associated the Content Key and Usage Rules for moved content on the Destination Device.

When the Current Move Control Information in the decrypted CKI image is equal to '01b,' the Destination Device sets the value of the Current Move Control Information field to '00b.'

When all of the above steps are executed successfully, the Destination Device securely holds the User Key, the Content Key, the Usage Rules for User Key, the Usage Rules for Content and the Usage Rules for moved content.

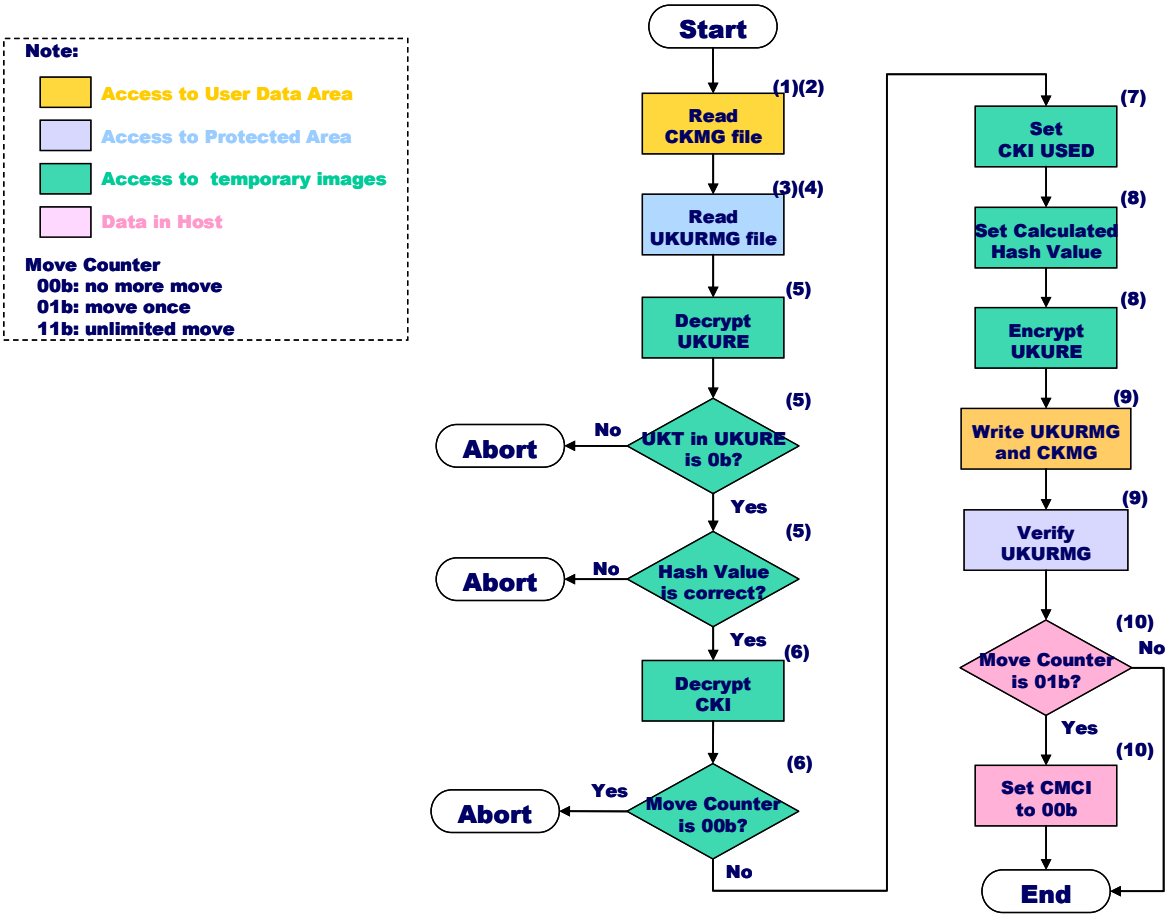


Figure 6-6 Flowchart of Content Key Move Process I

6.10 Content Key Move Process II (from Host to SD Memory Card)

The Source Device securely holds information associated with the Content Key to be moved. The information includes the Usage Rules for Content given by a Content Provider and information which can identify the User Key corresponding to the Content Key (e.g. User Key ID).

(1) Check the Usage Rules for Content on the Source Device.

The Source Device checks the Usage Rules for Content securely held in it. If the Current Move Control Information is equal to '00b,' then the process shall be aborted.

(2) Read or create a Content Key Management (CKMG).

(2.1) Select the CKMG file associated with the Content Key to be moved.

According to the identified User Key, the Recording Device selects the CKMG file.

(2.2) Read the selected CKMG file from the SD Memory Card.

The Source Device checks whether the selected CKMG file exists or not.

(a) When the selected CKMG file exists, the Source Device reads the selected CKMG file, holds it as the temporary CKMG image and checks the CKI Used flag (CKI_USED) field in the temporary CKMG image. If all the CKI Used flag are equal to '1b,' the process shall be aborted.

(b) When the selected CKMG file dose not exists, the Source Device creates the selected CKMG image on the Source Device.

(3) Determine the UKURMG file and UKURE associated with the Content Key to be moved by using UKURE_SRN.

(3.1) Obtain UKURE_SRN.

The Source Device obtains the UKURE_SRN ***s*** associated with the Content Key to be moved.

(3.2) Determine the UKURMG file and UKURE associated with the Content Key to be moved.

The Source Device determines the UKURMG filename and the UKURE using the following formula:

$s = (n - 1) * 250 + m$ (***n***: UKURMG file number, ***m***: UKURE number in a UKURMG)

$1 \leq m \leq 250,\ 1 \leq n \leq 256$

For example, when the UKURE_SRN is 1020, the UKURMG filename shall be "SDSD0005.KEY" (the fifth UKURMG file) and the UKURE shall be the twentieth entry in the "SDSD0005.KEY."

(4) Read the UKURMG file from the SD Memory Card.

The Source Device securely reads the ***n***th UKURMG file from the SD Memory Card using Secure Read Process and holds it as the temporary UKURMG image.

Then, the Source Device checks the ***m***th UKURE Used flag in the temporary UKURMG image. If it is equal to '0b,' the process shall be aborted.

Otherwise, the Source Device obtains the ***m***th UKURE in the temporary UKURMG image.

(5) Check the UKURE in the temporary UKURMG image.

The Source Device decrypts the UKURE using the UKURE Decryption process and securely holds it as the decrypted UKURE image.

In the case of C2-CPRM, the Source Device checks this decrypted UKURE image as follows:

If the Check Value is not '0123456789ABCDEFh,' the process shall be aborted.

If the TB for AES is not equal to '0b,' this process shall be aborted.

If the TB for future use is not equal to '000000b,' the process shall be aborted.

If the User Key Type in the UKURE is equal to '0b,' the Source Device checks the Hash Value in UKURE. If the stored Hash Value in UKURE is not equal to the calculated value from the temporary CKMG image using SD-SD Hash Calculation Process, the device shall execute CKMG Recovery Process described in section 5.7.

If the User Key Type in the UKURE is equal to '1b,' the process shall be aborted.

In the case of AES-CPRM, the Source Device checks this decrypted UKURE image as follows:

If the Check Value is not equal to the value of calculated CMAC(K_{u128}, high 40 bytes of the UR_U), the process shall be aborted.

If the TB for AES is not equal to '1b,' this process shall be aborted.

If the TB for future use is not equal to '000000b,' the process shall be aborted.

If the User Key Type in the UKURE is equal to '0b,' the Playback Device checks the Hash Value in UKURE. If the stored Hash Value in UKURE is not equal to the calculated value from the temporary CKMG image using SD-SD Hash Calculation Process, the device shall execute CKMG Recovery Process described in section 5.7.

(6) Update the temporary CKMG image.

(6.1) Update the CKI in the temporary CKMG image.

The Source Device checks CKI Used flag (CKI_USED) in the temporary CKMG image, and finds the first unused CKI.

The Source Device updates the unused CKI in the temporary CKMG image as follows:

The Copy Count Control Information of the CKI is set to the same value as that of the Copy Count Control Information of the Usage Rules for Content held in the Source Device.

The Initial Move Control Information field of the CKI is set to the same value as that of the Initial Move Control Information field of the Usage Rules for Content held in the Source Device.

The Current Move Control Information field of the CKI is set to the same value as that of the Initial Move Control Information field of the Usage Rules for Content held in the Source Device.

Each field in the Initial Field Group of the CKI is set to the same value as that of each corresponding field in Initial Field Group of the Usage Rules for Content held in the Source Device.

Each field in the Current Field Group of the CKI is set to the same value as that of each corresponding field in Current Field Group of the Usage Rules for Content held in the Source Device.

Each field in the Fixed Field Group of the CKI is set to the same value as that of each corresponding field in Fixed Field Group of the Usage Rules for Content held in the Source Device.

The Check Value of the CKI is set to (a) '0123456789ABCDEFh' in the case of C2-CPRM or (b) the value of calculated CMAC(K_{c128}, high 40 bytes of the UR_C) in the case of AES-CPRM.

After all the fields in the unused CKI are set as above, the Source Device encrypts the CKI in the temporary CKMG image using the CKI Encryption process.

(6.2) Set the CKI_USED in the temporary CKMG image.

The Source Device sets the CKI Used flag associated with the updated CKI in the temporary CKMG image to '1b.'

(7) Update the temporary UKURMG image

The Source Device updates the UKURE in the temporary UKURMG image. The Hash Value in the UKURE is set to the value that is calculated for the temporary CKMG image using SD-SD Hash Calculation Process. Then the Source Device encrypts the UKURE in the temporary UKURMG image using the UKURE Encryption process.

(8) Make the original Content Key held in the Source Device unusable.

The Source Device makes the original Content Key held in it permanently unusable.

(9) Write the updated temporary UKURMG and CKMG images to the SD Memory Card.

The Source Device securely writes the updated temporary UKURMG and CKMG images held in the Source Device as the updated UKURMG and CKMG files to the SD Memory Card using the CKMG Update Process in described in section 5.7.

To protect against the "Pull Card Attack," the Source Device must assume that the Source Process has been completely done, even if errors occur in step (9)

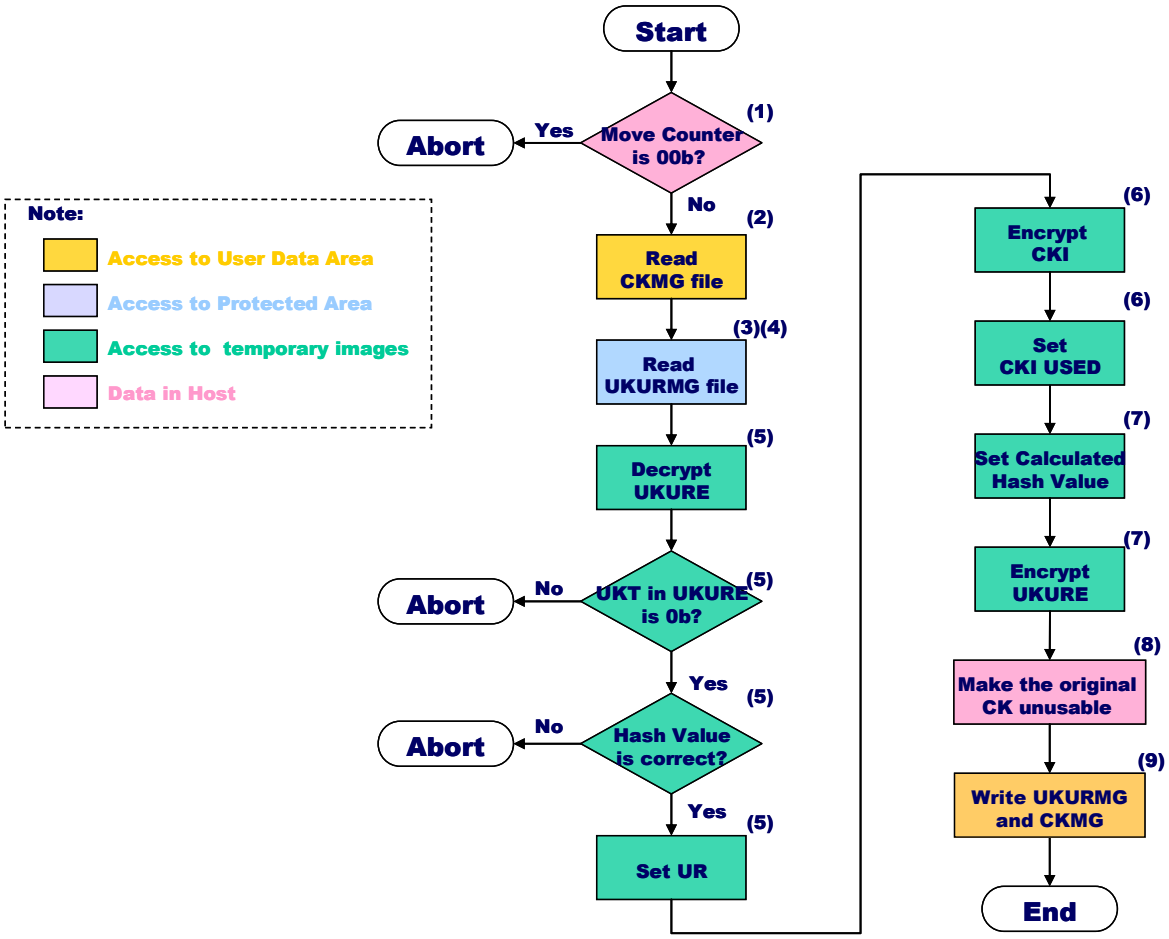


Figure 6-7 Flowchart of Content Key Move Process II

6.11 Content Key Move Process III (from SD Memory Card to SD Memory Card)

In the SD-SD specification, compliant devices may move a Content Key from an SD Memory Card to another SD Memory Card. The device can move a Content Key to execute the following two steps:

(1) The device moves a content key from SD Memory Card to the device in accordance with Content Key Move Process I.

(2) The device writes a content key from the device to the destination SD Memory Card in accordance with Content Key Recording Process.

The device shall hold the Content Key and its Usage Rule securely and shall not use the Content Key to decrypt any content before step (2) is completed.

On the condition that a device has two SD Memory Card interfaces and two SD Memory Cards are inserted to the interfaces in the device respectively, when the device moves a Content Key from an SD Memory Card to another SD Memory Card, the device can move the Content Key to execute the following four steps based on two processes, Content Key Move Process I and II:

(1) The device executes from step (1) to (8) of Content Key Move Process I,
In this step, the device securely reads the files including the original Content Key and its Usage Rule. The device also prepares to update the Content Key and its Usage Rule.

(2) The device executes from step (2) to (7) of Content Key Move Process II.
In this step, the device securely reads the files where the moved Content Key is to be stored. The device also prepares to write the Content Key and its Usage Rule.

(3) The device executes step (9) of Content Key Move Process I.
In this step, the device securely writes the files including the original Content Key and its Usage Rule.

(4) The device executes step (9) of Content Key Move Process II.
In this step, the device securely writes the files including the copies of the content key.

Note that completion of step (1) and (3) herein is regarded as step (1) and (8) in Content Key Move Process II respectively. To protect against the "Pull Card Attack," the device must assume that the Move Process III has been completely done, even if errors occur in step (4) herein.

6.12 Playback Process

This process shows how to playback the SD-SD content. In this process, the Playback Device may encounter an abort in this process because of time based usage rules in the UKURE. In the case, the Playback Device may update the UKURE by User Key Update Process defined in Section 6.14 of this book.

(1) Determine the CKMG file and CKI associated with the Content to be played back.

The Playback Device determines the CKMG filename and the CKI.

(2) Read the CKMG file from the SD Memory Card.

The Playback Device reads the CKMG file from the SD Memory Card and holds it as the temporary CKMG image.

Then, the Playback Device checks the corresponding CKI Used flag in the temporary CKMG image. If it is equal to '0b,' the process shall be aborted.

Otherwise, the Playback Device obtains the selected CKI in the temporary CKMG image.

(3) Determine the UKURMG file and UKURE associated with the SD-SD content to be played back.

(3.1) Obtain UKURE_SRN.

The Playback Device obtains the UKURE_SRN ***s*** associated with the SD-SD content to be played back.

(3.2) Determine the UKURMG file and UKURE associated with the SD-SD content to be played back.

The Playback Device determines the UKURMG filename and the UKURE using the following formula:

s = (***n*** - 1) * 250 + ***m*** (***n***: UKURMG file number, ***m***: UKURE number in a UKURMG)

$1 \leq \boldsymbol{m} \leq 250$, $1 \leq \boldsymbol{n} \leq 256$

For example, when the UKURE_SRN is 1020, the UKURMG filename shall be "SDSD0005.KEY" (the fifth UKURMG file) and the UKURE shall be the twentieth entry in the "SDSD0005.KEY."

(4) Read the UKURMG file from the SD Memory Card.

The Playback Device securely reads the ***n***th UKURMG file from the SD Memory Card using Secure Read Process and holds it as the temporary UKURMG image.

Then, The Playback Device checks the ***m***th UKURE Used flag in the temporary UKURMG image. If it is equal to '0b,' the process shall be aborted.

Otherwise, The Playback Device obtains the ***m***th UKURE in the temporary UKURMG image.

(5) Check the UKURE in the temporary UKURMG image.

The Playback Device decrypts the UKURE using the UKURE Decryption process and securely holds it as the decrypted UKURE image.

In the case of C2-CPRM, the Playback Device checks this decrypted UKURE image as follows:

If the Check Value is not '0123456789ABCDEFh,' the process shall be aborted.

If the TB for AES is not equal to '0b,' this process shall be aborted.

If the TB for future use is not equal to '000000b,' the process shall be aborted.

If the User Key Type in the UKURE is equal to '0b,' the Playback Device checks the Hash Value in UKURE. If the stored Hash Value in UKURE is not equal to the calculated value from the temporary CKMG image using SD-SD Hash Calculation Process, the device shall execute CKMG Recovery Process described in section 5.7.

In the case of AES-CPRM, the Playback Device checks this decrypted UKURE image as follows:

If the Check Value is not equal to the value of calculated CMAC(K_{u128}, high 40 bytes of the UR_U), the process shall be aborted.

If the TB for AES is not equal to '1b,' this process shall be aborted.

If the TB for future use is not equal to '000000b,' the process shall be aborted.

If the User Key Type in the UKURE is equal to '0b,' the Playback Device checks the Hash Value in UKURE. If the stored Hash Value in UKURE is not equal to the calculated value from the temporary CKMG image using SD-SD Hash Calculation Process, the device shall execute CKMG Recovery Process described in section 5.7

(6) Check the CKI in the temporary CKMG image.

The Playback Device decrypts the CKI using the CKI Decryption process and securely holds it as the decrypted CKI image.

In the case of C2-CPRM, the Playback Device checks this decrypted CKI image as follows:

If the Check Value is not '0123456789ABCDEFh,' the process shall be aborted.

If the PTB for AES is not equal to '0b,' this process shall be aborted.

If the PTB for future use is not equal to '00b,' the process shall be aborted.

If the Current Playback Counter is equal to '0000h,' then the process shall be aborted

In the case of AES-CPRM, the Playback Device checks this decrypted CKI image as follows:

If the Check Value is not equal to the value of calculated CMAC(K_{c128}, high 40 bytes of the UR_C), the process shall be aborted.

If the PTB for AES is not equal to '1b,' this process shall be aborted.

If the PTB for future use is equal to '00b,' the process shall be aborted.

If the Current Playback Counter is equal to '0000h,' then the process shall be aborted

(7) Check if Time-Based Usage Rules are set in the CKI and UKURE

The Playback Device checks if Time-Based Usage Rules are valid in the CKI and UKURE. Specifically, at least one of the following conditions is true, Time-Based Usage Rules are valid. Note that if the device works in a Mode, the device shall not change the Mode to another during this process.

(a) Validity of Current Start Date in the CKI is '1b'

(b) Validity of Current End Date in the CKI is '1b'

(c) Validity of Span in the CKI is '1b' and Span Length in the CKI is '0b'

(d) Current Playback Counter in the CKI is not 'FFFFh'

(e) Validity of Span in the CKI is '1b' and Span Length in the CKI is '1b'

(f) Validity of UK_STARTDDATE in the UKURE is '1b'

(g) Validity of UK_ENDDATE in the UKURE is '1b'

(h) Validity of UK_SPAN in the UKURE is '1b'

Then the device goes to the following sub-steps.

(7.1) When all of conditions above are false (not true), go to the last step (15).

(7.2) When one of the conditions is true and the Playback Device does not support Time-Based Usage Rules or the device does not work in neither Mode A, Mode B1 nor Mode B2, this process shall be aborted.

(7.3) When either (d) or (e) is true and all of (a), (b), (c), (f), (g) and (h) are false (not true), that is, rules require no date and time clock, go to the step (11).

(7.4) When the Clock Usage Flag in UR_C_TBUR_CDT is 01b, this process shall be aborted.

(7.5) When the Clock Usage Flag in UR_C_TBUR_CDT is 11b and the device does not work in Mode B2, this process shall be aborted.

(7.6) When the Clock Usage Flag in UR_C_TBUR_CDT is 10b and the device does not work in Mode B2 nor Mode B1, this process shall be aborted.

(7.7) Otherwise (rules require date and time clock), go to the following steps from (8) to (13).

(8) Obtain the current date and time

The Playback Device obtains the current date and time by referring to its internal Clock. If the Playback Device cannot obtain the current date and time, then the process shall be aborted.

(9) Update the decrypted CKI image and UKURE image (Phase 1).

In this phase, the CKI is checked if playback of the CKI with span-limited is done for the first time. The Playback Device checks the Current First Playback Flag field and the Validity of Span field. When the Current First Playback Flag field in the CKI is equal to '0b,' the Validity of Span field in the CKI is equal to '1b' and the Span Length is equal to '0b,' execute sub-steps from (9.1) to (9.3). When the First Playback Flag field in the UKURE is equal to '0b' and the Validity of UK_SPAN field in the UKURE is equal to '1b,' execute sub-steps from (9.4) to (9.6). Otherwise skip these sub-steps and go to step (10).

(9.1) Update the Current Start Date of Playback Period.

a) When the Validity of Current Start Date field is equal to '0b,' the Playback Device sets the Current Start Date of Playback Period field of the decrypted CKI image to the current date and time and sets the Validity of Current Start Date field to '1b.'

b) When the Validity of Current Start Date field is equal to ‘1b,’ the Playback Device compares the current date and time with the date and time of the Current Start Date of Playback Period field.

- If the current date and time precedes the Current Start Date of Playback Period, then the process shall be aborted.
- If the current date and time does not precede the Current Start Date of Playback Period, then the Playback Device sets the Current Start Date of Playback Period field of the decrypted CKI image to the current date and time.

(9.2) Update the Current End Date of Playback Period.

a) When the Validity of Current End Date field is equal to ‘0b,’ the Playback Device calculates the end date and time by adding the value specified in the Playback Span field to the current date and time, sets the Current End Date of the Playback Period field of the decrypted CKI image to the calculated end date and time, and sets the Validity of Current End Date field to ‘1b.’

b) When the Validity of Current End Date field is equal to ‘1b,’ the Playback Device compares the current date and time with the date and time of the Current End Date of the Playback Period field.

- If the current date and time does not precede the Current End Date of the Playback Period, then the process shall be aborted.
- If the current date and time precedes the Current End Date of the Playback Period, then the Playback Device calculates the end date and time by adding the value specified in the Playback Span field to the current date and time. If the calculated end date and time precedes the Current End Date of the Playback Period field of the decrypted CKI image, the Playback Device sets the Current End Date of the Playback Period field of the decrypted CKI image to the calculated end date and time.

(9.3) The Playback Device sets the Current First Playback Flag field to ‘1b.’

(9.4) Update the UR_UK_STARTDATE

When the Validity of UK_STARTDATE field is equal to ‘0b,’ the Playback Device sets fields of the decrypted UKURE image by the following settings:

- The Date of UK_STARTDATE field is set to the date of the current date and time
- The Hour of UK_STARTDATE field is set to the hour of the current date and time
- The Validity of UK_STARTDATE field is set to ‘1b.’

When the Validity of UK_STARTDATE field is equal to ‘1b,’ the Playback Device compares the current date and time with the date and time of the UK_STARTDATE field.

- If the current date and time precedes the UK_STARTDATE, then the process shall be aborted.
- If the current date and time does not precede the UK_STARTDATE, then the Playback Device sets (1) the Date of UK_STARTDATE field of the decrypted CKI image to the date of the current date and time and (2) the Hour of UK_STARTDATE field of the decrypted CKI image to the hour of the current date and time.

(9.5) Update the UR_UK_ENDDATE

When the Validity of UK_ENDDATE is equal to ‘0b,’ the Playback Device calculates the end date and time by adding the value specified in the UR_UK_SPAN field to the current date and time. Then the Playback Device set fields of the decrypted UKURE image by the following settings:

- The Date of UK_ENDDATE field is set to the date of the calculated date and time
- The Hour of UK_ENDDATE field is set to the hour of the calculated date and time
- The Validity of UK_ENDDATE field is set to ‘1b.’

When the Validity of UK_ENDDATE is equal to ‘1b,’ the Playback Device compares the current date and time with the date and time of the UR_UK_ENDDATE.

- If the current date and time does not precede the UR_UK_ENDDATE, then the process shall be aborted.
- If the current date and time precedes the UR_UK_ENDDATE, then the Playback Device calculates the end date and time by adding the value specified in the UR_UK_SPAN field to the current date and time. If the calculated date and time precedes the UR_UK_ENDDATE of the decrypted CKI image, the Playback Device sets (1) the Date of UK_ENDDATE of the decrypted CKI image to the date of the calculated end date and time and (2) the Hour of UK_ENDDATE of the decrypted CKI image to the hour of the calculated end date and time.

(9.6) The Playback Device sets the First Playback Flag field to ‘1b.’

(10) Check the CKI and UKURE in the temporary CKMG image (Phase 2).

In this phase, it checks if the current time is in the designated period.

(10.1) If the Validity of Current Start Date field is equal to ‘1b’ and the current date and time precedes the Current Start Date of Playback Period field, then the process shall be aborted.

(10.2) If the Validity of Current End Date field is equal to ‘1b’ and the current date and time does not precede the Current End Date of Playback Period field, then the process shall be aborted.

(10.3) If the Validity of UK_STARTDATE field is equal to ‘1b’ and the current date and time precedes the UR_UK_STARTDATE, then the process shall be aborted.

(10.4) If the Validity of UK_ENDDATE field is equal to ‘1b’ and the current date and time does not precede the UR_UK_ENDDATE, then the process shall be aborted.

(11) Update the temporary CKI (Phase 3).

In this phase, a process about the number of playback times is done. Specifically, if the Current Playback Counter of the decrypted CKI image is not equal to ‘FFFFh,’ the Playback Device decrements the value of the Current Playback Counter and encrypts the CKI in the temporary CKMG image using the CKI Encryption process.

(12) Check the User Key Type in UKURE.

The Recording Device checks the User Key Type which is obtained in step (5).

(a) When the User Key Type is equal to ‘0b,’ go to step (13).

(b) When the User Key Type is equal to ‘1b,’ go to step (15).

(13) Update the temporary UKURMG image.

If the decrypted CKI image has not been updated in step (9), (11), then go to step (15).

The Playback Device updates the UKURE in the temporary UKURMG image. The Hash Value in the UKURE is set to the value that is calculated for the temporary CKMG image using SD-SD Hash Calculation Process. Then the Playback Device encrypts the UKURE in the temporary UKURMG image using the UKURE Encryption process.

(14) Write the temporary UKURMG and CKMG images to the SD Memory Card.

The Playback Device securely writes the updated temporary UKURMG and CKMG images as the updated UKURMG and CKMG files to the SD Memory Card using the CKMG Update Process in described in section 5.7. Then the Playback Device securely reads the updated UKURMG file from the SD Memory Card using the Secure Read Process and verifies that the update of the ***m***th UKURE in the UKURMG file has completed successfully.

If the verification of the UKURMG file fails, the Playback Device shall abort this process.

(15) Start Playback

The Playback Device starts to play the SD-SD content. If the Validity of Span is set to '1b' and the Span Length is set to '1b,' the Playback Device shall measure time from start of playback. The Playback Device shall abort the playback after the time of playback exceeds the second described in Span Second field. If starting another playback process from the step (1) again, the Playback Device could playback SD-SD content. Therefore it is recommended that the Short Span Mode is used with playback counter.

In step (6), the Playback Device may check two or more CKIs from a CKMG at one process. Devices can store the valid CKIs temporarily and securely. When the device plays back the content corresponding to the stored CKIs, the device can playback the content with skipping from step (1) to step (5). Temporarily and securely stored CKIs shall be deleted under the following conditions:

The SD Memory Card is pulled out.

The CKI in the SD Memory Card is deleted.

The Playback Device's power is off.

The Playback Device application is terminated

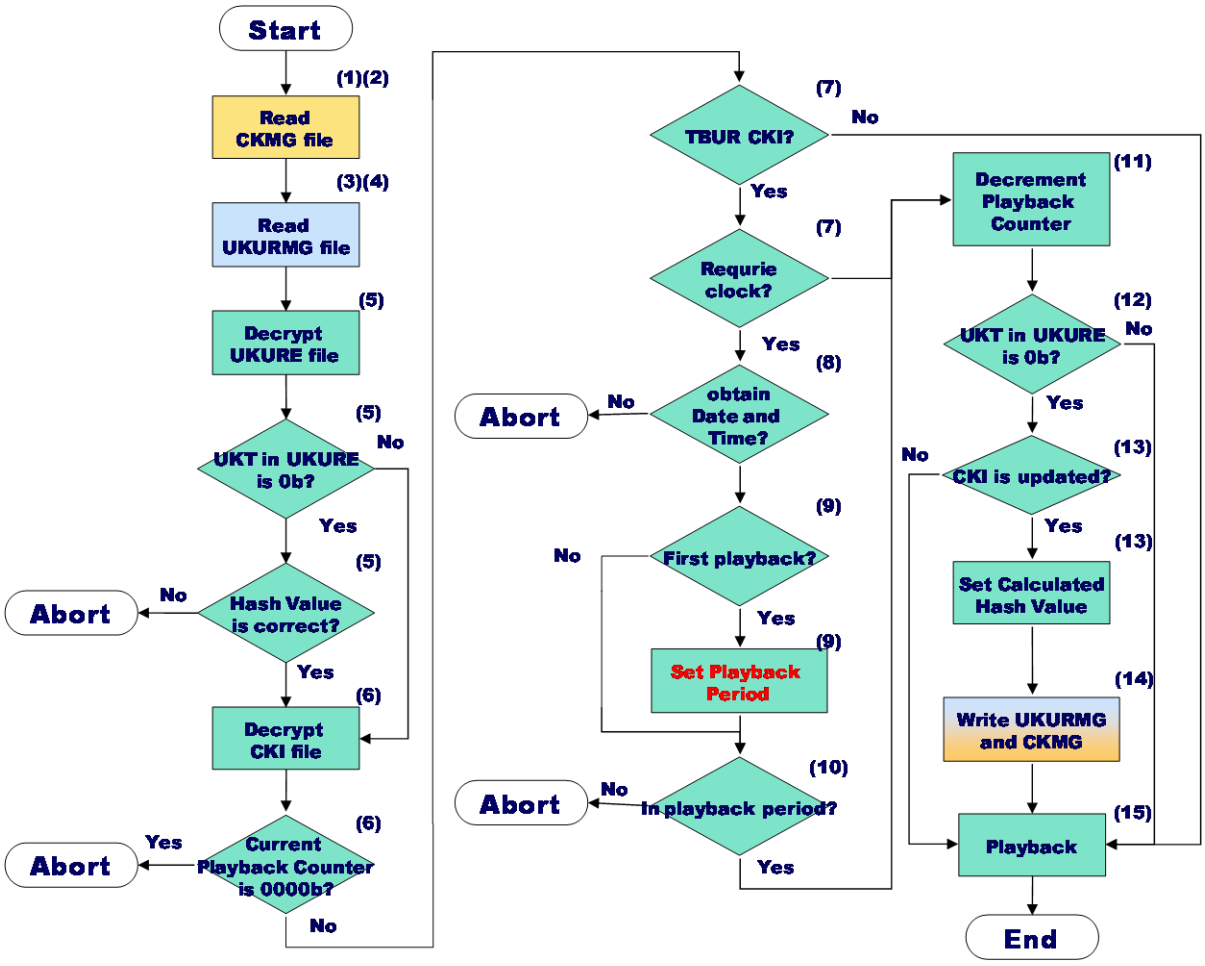


Figure 6-8 Flowchart of Playback Process

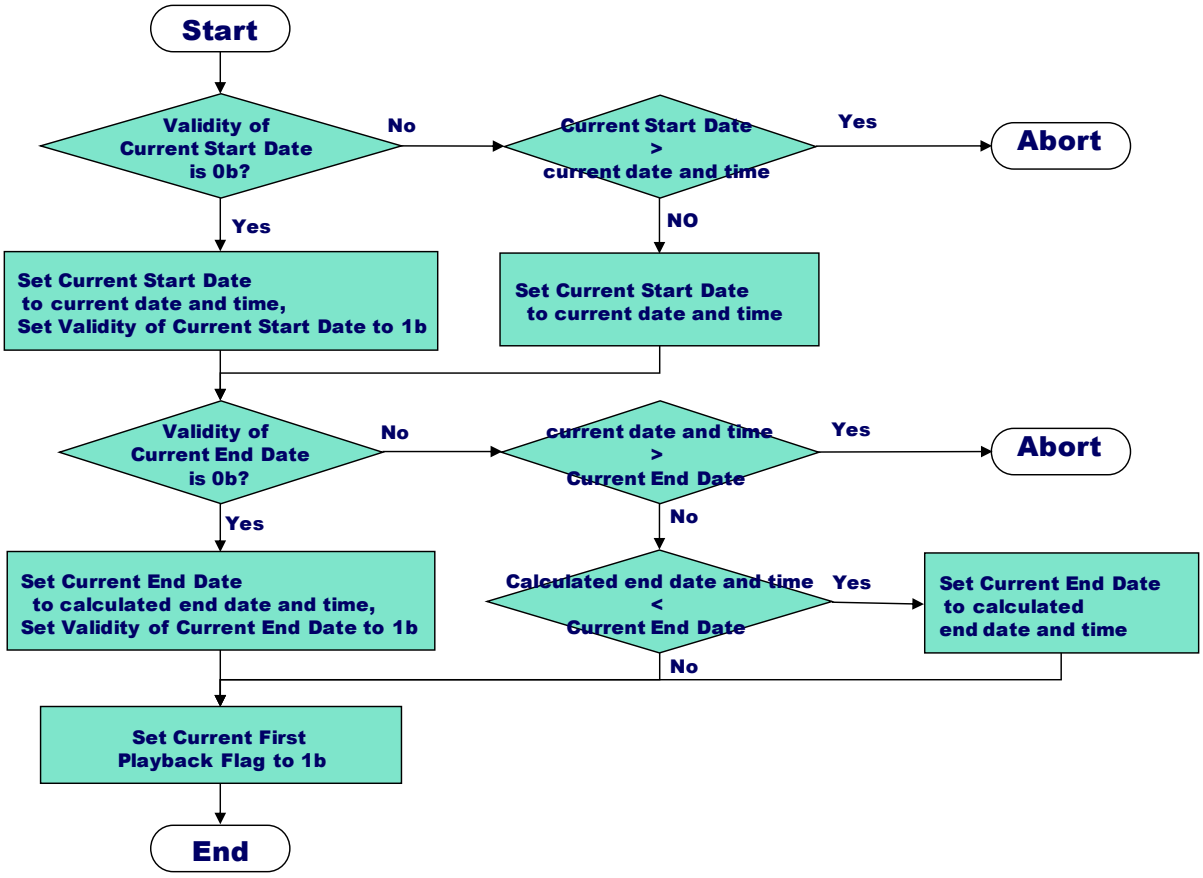


Figure 6-9 Flowchart of Set Playback Period

6.13 Process of Time Stamp files

This section shows how to treat Time Stamp files. The devices supporting Time-Based Usage Rules shall be compliant with rules described in the following sub-sections. The devices which do not supporting Time-Based Usage Rules shall not execute processes in the following sub-sections. Devices supporting Time-Based Usage Rules shall treat Time Stamp files in order to keep the accuracy of the Time Stamps before completing each process if necessary. Conditions and treatments are described in the following Sections.

6.13.1 Processing the Time Stamp files in Mode A

When accessing a CKI with Time-Based Usage Rules on the SD Memory Card, the device which works in Mode A shall process Time Stamp files as follows.

6.13.1.1 Processing Timestamp at Accessing to SD Memory Card in Mode A

The device which works in Mode A shall treat the Time Stamp files in the recording destination SD Memory Card in the case of the following process.

- Content Key Recording Process
- Content Key Erasing Process
- Encrypted Content Key Recording Process
- Content Key Copy Process II (from Host to SD Memory Card)
- Content Key Copy Process III (from SD Memory Card to SD Memory Card)
- Content Key Move Process I (from SD Memory Card to Host)
- Content Key Move Process II (from Host to SD Memory Card)
- Content Key Move Process III (from SD Memory Card to SD Memory Card)

Details are described in the following steps and executed before the process completes. The device is also allowed to execute the following steps in the other cases or anytime.

(1) Read or prepare the Time Stamp files

The device reads the TBUR_A.TS file from the Protected Area securely and holds it as the temporary image. If the file does not exist, the device creates the file data as the temporary image and skips the next step (2) and (3). The fields of the created file are set to the following values:

- AN is set to an arbitrary number, for example, a random number
- TS is set to all zero bits
- TS Verification Data is set to DEADBEEFh

In order to take more secure time into the ecosystem, the device is allowed to read the TBUR.TS file from the Protected Area securely and hold it as the temporary image. If the file does not exist, the device creates the file data as the temporary image. The fields of the data are set to the following values:

- TS is set to all zero bits
- Reserved is set to all zero bits

(2) Decrypt the TBUR_A.TS

The device decrypts the TBUR_A.TS image with the Media Unique Key of the SD Memory Card. How to decrypt is described in Section 5.6.1.

(3) Check the integrity of the TBUR_A.TS

The device checks if the TS Verification Data is equal to DEADBEEFh. If the check is false, the device regards the time of the TBUR_A.TS as an incorrect value and skips all of the following steps.

(4) Compare and adjust the times

The device compares (a) the time of Clock A in the device, (b) the time of the TBUR_A.TS file in the temporary image and (c) the time of the TBUR.TS file in the temporary image if the file is read in step (1).

(4.1) If the time of Clock A is ahead of the others, the device replaces the time of the TBUR_A.TS file with the time of Clock A and replaces the time of TBUR.TS file with the time of Clock A if any.

(4.2) If the time of the TBUR_A.TS file is ahead of the others, the device replaces the time of TBUR.TS file with the time of the TBUR_A.TS if any and the device shall not change the Exception Termination Counter in step (5).

(4.3) If the time of the TBUR.TS file is ahead of the others, the device replaces the time of the TBUR_A.TS file with the time of the TBUR.TS if any and the device shall not change the Exception Termination Counter in step (5).

(5) Set the TBUR_A.TS

The device sets the following parameters

- In-Use is set to 0b
- Exception Termination Counter is set to 000b on the condition that step (4.2) and (4.3) do not occur

(6) Encrypt the TBUR_A.TS

The device encrypts the TBUR_A.TS image with the Media Unique Key of the SD Memory Card. How to encrypt is described in Section 5.6.1.

(7) Record the updated file

The device records the TBUR_A.TS onto the SD Memory Card securely if the TBUR_A.TS is updated or newly created. If the device holds the TBUR.TS and the file is updated, the device also records the TBUR.TS onto the SD Memory Card securely.

6.13.1.2 Processing Timestamp at Start Playback in Mode A

At playback of content with Time-Based Usage Rules, devices that work in Mode A shall execute the following actions.

When the device do Playback Process described in Section 6.12 and before starting playback of the content, devices that work in Mode A shall complete the following steps:

(1) Read or prepare the Time Stamp files

The device reads the TBUR_A.TS file from the Protected Area securely and holds it as the temporary image. If the file does not exist, the device creates the file data as the temporary image and skips the next step (2). The fields of the created file are set to the following values:

- AN is set to an arbitrary number, for example, a random number
- TS is set to all zero bits
- TS Verification Data is set to DEADBEEFh

In order to take more secure time into the ecosystem, the device is allowed to read the TBUR.TS file from the Protected Area securely and hold it as the temporary image. If the file does not exist, the device creates the file data as the temporary image. The fields of the data are set to the following values:

- TS is set to all zero bits
- Reserved is set to all zero bits

(2) Decrypt the TBUR_A.TS

The device decrypts the TBUR_A.TS image with the Media Unique Key of the SD Memory Card. How to decrypt is described in Section 5.6.1.

(3) Check and set the fields of the TBUR_A.TS

The device checks the following conditions.

- If the TS Verification Data is not equal to DEADBEEFh, the device regards the time of the TBUR_A.TS as incorrect value and does not use the time for comparison in the next step (4) and shall create the new TBUR_A.TS as the temporary image. The fields of the created file are set to the following values:
  - TS is set to all zero bits
  - TS Verification Data is set to DEADBEEFh
- If the In-Use is 1b, the device increases Exception Termination Counter. Then if the Exception Termination Counter reaches 6 (110b), the device shall abort this process and the playback.
- If the In-Use is 0b, the In-Use is set to 1b.

(4) Compare and adjust the times

The device compares (a) the time of Clock A in the device, (b) the time of the TBUR_A.TS file in the temporary image and (c) the time of the TBUR.TS file in the temporary image if the file is read in step (1).

(4.1) If the time of Clock A is ahead of the others, the device updates the TBUR_A.TS file. The setting of the time of the TBUR_A.TS file, In-Use and ETC depends on the following conditions:

(a) In the case that In-Use is 0 and ETC is either of 0, 1, 2, 3, 4 or 5:
The time of the TBUR_A.TS is set to the time of Clock A,
In-Use is set to 1 and
ETC is set to 0.

(b) In the case that In-Use is 1 and ETC is either of 0, 1, 2, 3 or 4:
The time of the TBUR_A.TS is set to the time of Clock A,
In-Use is set to 1 (not changed) and
ETC increases.

(c) In the case that In-Use is 1 and ETC is 5:
The device shall abort the Playback process after step (6) in this process. The time of the TBUR_A.TS is set to the time of Clock A plus the time length of the content to be played,
In-Use is set to 0 and
ETC is set to 5 (not changed).

In case that the TBUR.TS is read, the device shall update the TBUR.TS file. The setting of the file is based on the following conditions:

(d) In the case that In-Use is 0 and ETC is either of 0, 1, 2, 3, 4 or 5:
The time of the TBUR.TS is set to the time of Clock A,
In-Use is set to 1 and
ETC is set to 0.

(e) In the case that In-Use is 1 and ETC is either of 0, 1, 2, 3 or 4:
The time of the TBUR.TS is set to the time of Clock A,
In-Use is set to 1 (not changed) and
ETC increases.

(f) In the case that In-Use is 1 and ETC is 5:
The device shall abort the Playback process after step (6) in this process. The time of the TBUR.TS is set to the time of Clock A plus the time length of the content to be played,
In-Use is set to 0 and
ETC is set to 5 (not changed).

(4.2) If the time of the TBUR_A.TS file or the TBUR.TS file is ahead of the others, the device shall abort this process and the Playback process.

(5) Encrypt the TBUR_A.TS

The device encrypts the TBUR_A.TS image with the Media Unique Key of the SD Memory Card. How to encrypt is described in Section 5.6.1.

(6) Record the updated file

The device records the TBUR_A.TS securely if the TBUR_A.TS is updated. If the device holds the TBUR.TS and the file is updated, the device also records the TBUR.TS in the SD Memory Card securely.

Note that if the condition is either (c) of step (4.1) or step (4.2) in this process, the device shall not execute the Playback process.

6.13.1.3 Processing Timestamp at Stop Playback in Mode A

When the device has updated the TBUR_A file in Playback Process described in Section 6.12 and after stopping or finishing the playback of the content, devices that work in Mode A shall execute the following steps:

(1) Read or prepare the Time Stamp files

The device reads the TBUR_A.TS file from the Protected Area securely and holds it as the temporary image. If the file does not exist, the device creates the file data as the temporary image and skips the next step (2) and (3). The fields of the created file are set to the following values:

- AN is set to an arbitrary number, for example, a random number
- TS is set to all zero bits
- TS Verification Data is set to DEADBEEFh

In order to take more secure time into the ecosystem, the device is allowed to read the TBUR.TS file from the Protected Area securely and hold it as the temporary image. If the file does not exist, the device creates the file data as the temporary image. The fields of the data are set to the following values:

- TS is set to all zero bits
- Reserved is set to all zero bits

(2) Decrypt the TBUR_A.TS

The device decrypts the TBUR_A.TS image with the Media Unique Key of the SD Memory Card. How to decrypt is described in Section 5.6.1.

(3) Check and set the fields of the TBUR_A.TS

If the TS Verification Data is not equal to DEADBEEFh, the device regards the time of the TBUR_A.TS as incorrect value and does not use the time for comparison in the next step (4) and shall create the new TBUR_A.TS as the temporary image. The fields of the created file are set to the following values:

(4) Compare and adjust the times

The device compares (a) the time of Clock A in the device, (b) the time of the TBUR_A.TS file in the temporary image and (c) the time of the TBUR.TS file in the temporary image if the file is read in step (1). The device updates the TBUR_A.TS file and the TBUR.TS file if the TBUR.TS file is read. The setting of the files depends on the following conditions:

- In the case that the time of Clock A is ahead of the others, the updates of the TBUR_A.TS file and TBUR.TS file are the following:
  The time of the TBUR_A.TS file and TBUR.TS file is set to the time of Clock A,
  In-Use is set to 0 and
  ETC is set to 0.

- If the time of the TBUR_A.TS file is ahead of the others, the updates of the TBUR_A.TS file and TBUR.TS file are the following:
  The time of the TBUR_A.TS file and TBUR.TS file is set to the time of TBUR_A.TS,
  In-Use is set to 0 and
  ETC is unchanged.
- If the time of the TBUR.TS file is ahead of the others, the updates of the TBUR_A.TS file and TBUR.TS file are the following:
  The time of the TBUR_A.TS file and TBUR.TS file is set to the time of TBUR.TS,
  In-Use is set to 0 and
  ETC is unchanged.

(5) Encrypt the TBUR_A.TS

The device encrypts the TBUR_A.TS image with the Media Unique Key of the SD Memory Card. How to encrypt is described in Section 5.6.1.

(6) Record the updated file

The device records the TBUR_A.TS securely if the TBUR_A.TS is updated. If the device holds the TBUR.TS and the file is updated, the device also records the TBUR.TS in the SD Memory Card securely.

6.13.2 Processing the Time Stamp files in Mode B

At accessing to a CKI with Time-Based Usage Rules onto SD Memory Card, the device which works in Mode B2 shall record the new Time Stamp files however the device that works in Mode B1 shall not treat the Time Stamp files. The device in Mode B2 is allowed to replace the time of TBUR_A.TS file with the time of the device in order to adjust an incorrect Time Stamp file. How to proceed is described in Section 6.13.2.2.

6.13.2.1 Processing Timestamp at Accessing to SD Memory Card in Mode B2

The device which works in Mode B2 shall treat the time stamp files in the recording destination SD Memory Card in the case of the following process:

- Content Key Recording Process
- Content Key Erasing Process
- Encrypted Content Key Recording Process
- Content Key Copy Process II (from Host to SD Memory Card)
- Content Key Copy Process III (from SD Memory Card to SD Memory Card)
- Content Key Move Process I (from SD Memory Card to Host)
- Content Key Move Process II (from Host to SD Memory Card)
- Content Key Move Process III (from SD Memory Card to SD Memory Card)

Details are described in the following steps and executed before the process completes. The device is also allowed to execute the following steps in the other cases or anytime.

(1) Read or Prepare the Time Stamp files

The device reads the TBUR_B.TS file from the Protected Area securely and holds it as the temporary image. If the file does not exist, the device creates the file data as a temporary image and skips the next step (2) and (3). The fields of the created file are set to the following values:

- AN is set to an arbitrary number, for example, a random number
- Used flag is set to 0b
- The time of TS is set to the time of Clock B2

- TS Verification Data is set to DEADBEEFh

In order to take more secure time into the ecosystem, the device is allowed to read the TBUR.TS file from the Protected Area securely and hold it as the temporary image. If the file does not exist, the device creates the file data as the temporary image. The fields of the data are set to the following values:

- The time of TS is set to the time of Clock B2
- Reserved is set to all zero bits

(2) Decrypt the TBUR_B.TS

The device decrypts the TBUR_B.TS image with the Media Unique Key of the SD Memory Card. How to decrypt is described in Section 5.6.1.

(3) Check and update the TBUR_B.TS

The fields of the file are set to the following values:

- Used flag is set to 0b
- The time of TS is set to the time of Clock B2
- TS Verification Data is set to DEADBEEFh

If the device hold the TBUR.TS and the time of Clock B2 is ahead of the time of TS in the TBUR.TS, the device sets the time of TS to the time of Clock B2.

(4) Encrypt the TBUR_B.TS

The device encrypts the TBUR_B.TS image with the Media Unique Key of the SD Memory Card. How to encrypt is described in Section 5.6.1.

(5) Record the TBUR_B.TS file

The device records the TBUR_B.TS securely. If the device holds the TBUR.TS, the device also records the TBUR.TS in the SD Memory Card securely.

6.13.2.2 Processing correction of TBUR_A file and TBUR file in Mode B2

The device which works in Mode B2 is allowed to adjust the incorrect TBUR_A.TS file and TBUR file by the steps below. The device is allowed to execute these steps anytime:

(1) Prepare the Time Stamp files

The device creates the TBUR_A.TS file or TBUR.TS file data as a temporary image. The fields of the created file are set to the following values:

- AN is set to an arbitrary number, for example, a random number (TBUR_A.TS only)
- In-Use is set to 0b
- ETC is set to 000b
- The time of TS is set to the time of Clock B2
- TS Verification Data is set to DEADBEEFh (TBUR_A.TS only)
- Reserved is set to all zero bits (TBUR.TS only)

(2) Encrypt the TBUR_A.TS

The device encrypts the TBUR_A.TS image with the Media Unique Key of the SD Memory Card. How to encrypt is described in Section 5.6.1.

(3) Record the TBUR_A.TS file or TBUR.TS file

The device records the TBUR_A.TS or TBUR.TS file securely.

6.13.2.3 Processing Timestamp at Event in Mode B1

This process is used for adjustment of Clock B1. Devices in Mode B1are allowed to execute this process at any time. However, when the following conditions occur,

- An SD Memory Card is inserted
- An application or device starts

the devices that works in Mode B1 shall execute the following steps:

(1) Read the Time Stamp file

The device reads the TBUR_B.TS file from the Protected Area securely and holds it as the temporary image. If the file does not exist, this process shall abort.

(2) Decrypt the TBUR_B.TS

The device decrypts the TBUR_B.TS image with the Media Unique Key of the SD Memory Card. How to decrypt is described in Section 5.6.1.

(3) Check and set the fields of the TBUR_B.TS

If the TS Verification Data is not equal to DEADBEEFh, the device regards the time of the TBUR_B.TS as an incorrect value and shall create the new TBUR_B.TS as the temporary image and skip the next step (4). The fields of the created file image are set to the following values:

- Used flag is set to 1b
- The time of TS is set to the time of Clock B1
- TS Verification Data is set to DEADBEEFh

(4) Compare and adjust the times

The device compares (a) the time of Clock B in the device, (b) the time of the TBUR_B.TS file in the temporary image.

(4.1) If the Time Stamp in the TBUR_B.TS is ahead of the time of Clock B1, the device shall adjust the time of Clock B1 identical to the Time Stamp in TBUR_B.TS and set the Used flag in TBUR_B.TS to ‘1b.’

(4.2) If the time of Clock B1 is ahead of the Time Stamp in the TBUR_B.TS and the Used flag in TBUR_B.TS is ‘0b,’ the device shall adjust the time of Clock B1 identical to the Time Stamp in TBUR_B.TS and set the Used flag to ‘1b.’

(5) Encrypt the TBUR_B.TS

The device encrypts the TBUR_B.TS image with the Media Unique Key of the SD Memory Card. How to encrypt is described in Section 5.6.1.

(6) Record the updated file

The device records the TBUR_B.TS securely if the TBUR_B.TS is updated.

6.14 User Key Update Process

The Updating Device securely receives and holds information associated with the User Key to be updated. The information includes the Usage Rules for User Key given by a Service Provider. How to transmit the information from the Service Provider to the Updating Device is out of this specification’s scope.

(1) Determine the UKURMG file and UKURE to be updated.

(1.1) Obtain UKURE_SRN.

The Updating Device obtains the UKURE_SRN ***s*** associated with the User Key to be updated.

(1.2) Determine the UKURMG file and UKURE associated with the User Key to be updated.

The Updating Device determines the UKURMG filename and the UKURE using the following formula:

$s = (n - 1) * 250 + m$ (***n***: UKURMG file number, ***m***: UKURE number in a UKURMG)

$1 \leq m \leq 250$, $1 \leq n \leq 256$

For example, when the UKURE_SRN is 1020, the UKURMG filename shall be "SDSD0005.KEY" (the fifth UKURMG file) and the UKURE shall be the twentieth entry in the "SDSD0005.KEY."

(2) Read the UKURMG file from the SD Memory Card.

The Updating Device securely reads the ***n***th UKURMG file from the SD Memory Card using Secure Read Process and holds it as the temporary UKURMG image.

Then, The Updating Device checks the ***m***th UKURE Used flag in the temporary UKURMG image. If it is equal to '0b,' the process shall be aborted.

Otherwise, The Updating Device obtains the ***m***th UKURE in the temporary UKURMG image.

(3) Check the UKURE in the temporary UKURMG image.

The Updating Device decrypts the UKURE using the UKURE Decryption process and securely holds it as the decrypted UKURE image.

In the case of C2-CPRM, the Updating Device checks this decrypted UKURE image as follows:

If the Check Value is not '0123456789ABCDEFh,' the process shall be aborted.

If the TB for AES is not equal to '0b,' this process shall be aborted.

If the TB for future use is not equal to '000000b,' the process shall be aborted.

If the User Key Type in the UKURE is equal to '0b,' the Updating Device checks the Hash Value in UKURE. If the stored Hash Value in UKURE is not equal to the calculated value from the temporary CKMG image using SD-SD Hash Calculation Process, the Updating Device shall execute CKMG Recovery Process described in Section 5.7.

In the case of AES-CPRM, the Updating Device checks this decrypted UKURE image as follows:

If the Check Value is not equal to the high 8 bytes of calculated CMAC(K_{u128}, high 40 bytes of the UR_U), the process shall be aborted.

If the TB for AES is not equal to '1b,' this process shall be aborted.

If the TB for future use is not equal to '000000b,' the process shall be aborted.

If the User Key Type in the UKURE is equal to '0b,' the Updating Device checks the Hash Value in UKURE. If the stored Hash Value in UKURE is not equal to the calculated value from the temporary CKMG image using SD-SD Hash Calculation Process, the Updating Device shall execute CKMG Recovery Process described in Section 5.7

(4) Update the temporary UKURMG image.

The Updating Device updates the UKURE in the temporary UKURMG image as follows:

The Trigger Bits is set to the value specified by Service Provider.

The User Key Type is set the value specified by the Service Provider.

The User Key ID is set the value specified by the Service Provider.

The UR_UK_STARTDATE is set the value specified by the Service Provider.

The UR_UK_ENDDATE is set the value specified by the Service Provider.

The UR_UK_SPAN is set the value specified by the Service Provider.

The Check Value is set to (a) '0123456789ABCDEFh.' in the case of C2-CPRM or (b) the high 8 bytes of CMAC(K_{u128}, high 40 bytes of the UR_U which has been updated with the fields above) in the case of AES-CPRM.

After all the fields in the UKURE are set as above, the Updating Device encrypts the UKURE in the temporary UKURMG image using the UKURE Encryption process.

(5) Write the updated temporary UKURMG image to the SD Memory Card.

The Updating Device securely writes the updated temporary UKURMG image held in the Updating Device as the updated UKURMG file to the SD Memory Card using the Secure Write Process.

To protect against the "Pull Card Attack," the Updating Device must assume that this process has been completely done, even if errors occur in step (5).

Chapter 7
MKB Extension

7. MKB Extension

The MKB Extension file configuration in the User Date Area for SD-SD is as follows:

In the case of C2-CPRM, the MKB Extension file used by the devices accessing the data stored in “SD_SD” in the Protected Area is located under the “SD_SD” directory in the User Data Area and the name of the MKB Extension file is “SD_SDKEY.MKB.” In the case of AES-CPRM, the MKB Extension file used by the devices accessing the data for AES-CPRM is located under the “SD_SD” directory in the User Data Area and the name of the MKB Extension file is “SD_SD128.MKB.” To be clear, the Media Unique Key calculated from the Extended MKB which consists of the Static MKB and the MKB Extension called “SD_SDKEY.MKB” is used for encryption of data stored in the “SD_SD” directory in the Protected Area and the Media Unique Key calculated from the Extended MKB which consists of the Static MKB and the MKB Extension called “SD_SD128.MKB” is used for encryption of data stored in the “SD_SD128” directory in the Protected Area.

When writing the MKB Extension file, the device shall either (1) write only “SD_SDKEY.MKB” and re-encrypt all User Keys stored in “SD_SD” directory in the Protected Area or (2) write both “SD_SDKEY.MKB” and “SD_SD128.MKB” and re-encrypt all User Keys stored in both “SD_SD” and “SD_SD128” directory in the Protected Area.

In addition to update the User Keys, when writing an MKB Extension file, the device shall re-encrypt the TBUR_A.TS and TBUR_B.TS as described in the following steps if those exist. Note that this process shall be done by all type of devices despite support of Time-Based Usage Rules.

(1) Securely read the TBUR_A.TS and TBUR_B.TS from the SD Memory Card.

(2) Calculate the old K_{mu} by the Static MKB and MKB Extension (if it exists.)

(3) Decrypt the TBUR_A.TS and TBUR_B.TS.

(4) Update time information in the files read in step (3). When the device cannot get the current time or does not support Time-Based Usage Rules, the update of time information in the file shall not be updated.

(5) Calculate the new K_{mu} by the Static MKB and MKB Extension.

(6) Encrypt the TBUR_A.TS and TBUR_B.TS with the new K_{mu}.

(7) Securely write the updated and encrypted TBUR_A and TBUR_B.TS in step (6).

This page is intentionally left blank.

Chapter 8
MKB Update

8. MKB Update

This chapter describes modification of the procedure of the dynamic MKB update defined in Section 3.10 of *SD Memory Card Book Common Part*. This modification shall apply to the Updating Device to update A_MKB for SD-SD.

In the procedure, the Title Key shall be replaced with the User Key. Therefore the Updating Device shall re-encrypt User Keys instead of Title Keys. When updating, the Updating Device shall re-encrypt all User Keys, the TBUR_A.TS file and the TBUR_B.TS file stored in both “SD_SD” and “SD_SD128” directory in the Protected Area.

This page is intentionally left blank.