Advanced Access Content System (AACS)

Pre-recorded Video Book

Intel Corporation
International Business Machines Corporation
Matsushita Electric Industrial Co., Ltd.
Microsoft Corporation
Sony Corporation
Toshiba Corporation
The Walt Disney Company
Warner Bros.

Revision 0.90
April 14, 2005

Preliminary Draft
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, Matsushita Electric Industrial Co., Ltd., Microsoft Corporation, Sony Corporation, Toshiba Corporation, The Walt Disney Company and Warner Bros. 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 a preliminary draft and is subject to change without notice.

Copyright © 2005 by Intel Corporation, International Business Machines Corporation, Matsushita Electric Industrial Co., Ltd., Microsoft Corporation, Sony Corporation, Toshiba Corporation, The Walt Disney Company, and Warner Bros. Third-party brands and names are the property of their respective owners.

Intellectual Property
Implementation of this specification requires a license from AACS LA LLC.

Contact Information
Please address inquiries, feedback, and licensing requests to AACS LA LLC:

- Licensing inquiries and requests should be addressed to licensing@aacsla.com.
- Feedback on this specification should be addressed to comment@aacsla.com.

The URL for the AACS LA LLC web site is http://www.aacsla.com.
This page is intentionally left blank.
# Table of Contents

Notice ........................................................................................................................................................ iii

Intellectual Property.......................................................................................................... ...................... iii

Contact Information............................................................................................................ .................... iii

CHAPTER 1 INTRODUCTION .............................................................................1

1 INTRODUCTION.............................................................................................1
  1.1 Purpose and Scope.............................................................................................................. ........1
  1.2 Overview....................................................................................................................... ...............1
  1.3 Organization of this Document.................................................................................................. 3
  1.4 References ..................................................................................................................... ..............3
  1.5 Future Directions.............................................................................................................. ..........3
  1.6 Notation .......................................................................................................................................3
  1.7 Terminology ................................................................................................................................3
  1.8 Abbreviations and Acronyms ....................................................................................................4

CHAPTER 2 CONTENT REVOCATION...............................................................5

2 INTRODUCTION.............................................................................................5
  2.1 Scope ...........................................................................................................................................6
  2.2 Content Signing infrastructure .................................................................................................6
  2.3 Content Hash Table.....................................................................................................................6
  2.4 Content Certificate ....................................................................................................................6
  2.5 Creating Content Certificate .....................................................................................................8
  2.6 Verifying Content Certificate ....................................................................................................9
  2.7 Content Revocation List (CRL)...............................................................................................10
CHAPTER 3 CONTENT ENCRYPTION AND DECRYPTION .........................13

3 INTRODUCTION ..............................................................................................13

3.1 Content Encryption (General) ...........................................................................13

3.2 Content Decryption (General) ...........................................................................14

3.3 Calculating the Volume Unique Keys .................................................................14

3.4 AACS Encryption on Pre-Recorded Media .......................................................15

3.5 AACS Decryption on Pre-Recorded Media ........................................................15

CHAPTER 4 SEQUENCE KEY BLOCK ...............................................................17

4 INTRODUCTION ..............................................................................................17

4.1 Sequence Key Block Principles ........................................................................18

4.2 Calculation of the Media Key Variant Data .......................................................19

4.2.1 Sequence Keys .............................................................................................19

4.2.2 Sequence Key Block (SKB) .........................................................................19

4.2.2.1 Verify Media Key Record ..........................................................................20

4.2.2.2 Nonce Record ..........................................................................................22

4.2.2.3 Calculate Variant Data Record .................................................................23

4.2.2.4 Conditionally Calculate Variant Data Record ...........................................25

4.2.2.5 End of Sequence Key Block Record .........................................................27

4.3 Calculation of the Media Key Variant from the Variant Data .........................28
List of Figures

Figure 1-1 – System Overview (Informative) ........................................................................................................ 2
Figure 2-1 – Content Validation and Revocation Overview .................................................................................. 5
Figure 2-2 – Content Signing Process ................................................................................................................... 9
Figure 3-1 – Encryption and Decryption Overview ............................................................................................. 13
Figure 4-1 – Example Sequence Key Block ........................................................................................................ 18
This page is intentionally left blank.
List of Tables

Table 2-1 – Content Certificate for Pre-recorded Video................................................................. 7
Table 2-2 – Content Revocation List for Pre-recorded Video .......................................................... 10
Table 2-3 – Revocation Record for Content Certificate ID .............................................................. 12
Table 4-1 – Verify Media Key Record Format.................................................................................. 20
Table 4-2 – Nonce Record Format.................................................................................................. 22
Table 4-3 – Calculate Variant Data Record Format ......................................................................... 23
Table 4-4 – Conditionally Calculate Variant Data Record Format ...................................................... 25
Table 4-5 – End of Sequence Key Block Record Format ................................................................. 27
Chapter 1
Introduction

1 Introduction

1.1 Purpose and Scope
The Advanced Access Content System (AACS) specification defines an advanced, robust and renewable method for protecting entertainment content, including high-definition audiovisual content. The specification is organized into several “books”. The Introduction and Common Cryptographic Elements book defines cryptographic procedures that are common among the various defined uses of the protection system. This document (the Pre-recorded Video Book) specifies additional details for using the system to protect audiovisual content distributed on pre-recorded (read-only) storage media. Specifications covering other storage types, transmission media and formats are expected to be available in the future (see Section 1.5 below).

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 AACS LA LLC (hereafter referred to as AACS LA) is responsible for establishing and administering the content protection system based, in part, on this specification.

1.2 Overview
In addition to the general objectives described in the Introduction and Common Cryptographic Elements book of this specification, the use of AACS for protecting pre-recorded video content was designed to meet the following specific criteria:

- Provide robust protection for both off-line playback and optional enhanced uses enabled via on-line connection.
- Provide for extended and extensible usage (e.g. jukebox storage, pay for copy).
- Independent of physical storage format to the degree possible.
- Compliant players can authenticate that content came from an authorized, licensed replicator.

Figure 1-1 presents an informative overview of the system, as used for protecting pre-recorded high-definition video content. Actual details and requirements of system operation are described in subsequent chapters.
The owner of audiovisual content that is to be protected provides that content and an associated set of usage rules to a licensed replicator.

The licensing entity provides to the licensed replicator device revocation data in the form of a Media Key Block (MKB). As described in the Introduction and Common Cryptographic Elements book of this specification, the MKB will enable all compliant players, each using its set of device keys, to calculate the Media Key. If a set of device keys is compromised in a way that threatens the integrity of the system, an updated MKB can be provided by the licensing entity that will cause a product with the compromised set of device keys to calculate a different key than is computed by the remaining compliant products. In this way, the compromised device keys are “revoked” by the new MKB.

The licensing entity also provides to the licensed replicator content revocation data in the form of a Content Revocation List (CRL), and a Content Certificate, both cryptographically signed by the AACS LA. The Certificate identifies the content and includes a cryptographic hash thereof (based on a series of hashes), which the licensing entity receives from the replicator prior to signing the Certificate. The CRL identifies content that has been signed and contains a valid certificate but has since been revoked and therefore should not be accessed by a compliant player.

The licensing entity also provides to the licensed replicator device variation data in the form of a Sequence Key Block (SKB), and secret keys, called Media Key Variants, based on both the device variation data and Media Key.

The licensed replicator encrypts the content to be protected, using a one or more secret keys of its own choosing, called Title Keys. The replicator encrypts the Title Keys in one or more keys derived from the Media Key. The replicator denotes one of the Title Keys as the primary Title Key and uses it to create a MAC (Message Authentication Code) on the usage rules. This MAC ensures that any malicious alteration of the usage rules will be detected. The encrypted content, usage rules, MAC of usage rules, MKB, SKB, CRL and Content Certificate are all pre-recorded onto the storage medium.

The AACS LA provides secret device keys and sequence keys to licensed manufacturers for inclusion in compliant playback devices/applications (device key sets and sequence key sets are typically unique per device/application). When a storage medium with protected content is placed in a compliant player, the player uses its Device Keys to process the MKB and calculate the corresponding Media Key.
and its Sequence Keys to process the SKB, the device will calculate a particular Media Key Variant. Assuming
the given set of Device Keys has not been revoked, the calculated key will be one of the keys used by the
licensed replicator. The player uses an inverse procedure to that used by the licensed replicator to derive a Title
Key, reads the Usage Rules from the optical media, and verifies they have not been modified using the MAC. The
player also uses the Title Key to provide access to the protected content in a compliant manner.

The AACS LA also provides its entity public key to licensed manufacturers for inclusion in compliant
devices/applications. Prior to accessing protected content (or unprotected content in the same format), a
compliant player reads the Content Certificate and CRL from the pre-recorded medium, and uses the entity
public key to verify their integrity. If either verification fails, access is aborted.

The compliant player keeps the CRL in non-volatile storage, unless it already has a more up-to-date list. Using
the most up-to-date CRL, the player checks to see if the content is revoked, and if it is, access is aborted.
During playback, the compliant player calculates a series of content hashes using the same method used by the
replicator. If the player’s calculated hash values differ at any point from the replicator-stored values, access is
aborted.

The system enables certain enhanced uses of pre-recorded video content, at the election of the content owner,
through the use of a robust on-line connection. For example, a home video server might connect with a service
provider to obtain authorization to make a protected local copy of a given pre-recorded Title for “jukebox”
purposes. Such authorization might be provided free-of-charge to the owner of the optical media, with any
additional authorized copies incurring a charge. Thus, this and other enhanced uses may entail business
interaction between content owners and service providers, as indicated by the dashed line in the figure above.

1.3 Organization of this Document
This document is organized as follows:

- Chapter 1 provides an introduction and overview.
- Chapter 2 describes procedures related to the authentication and revocation of pre-recorded content.
- Chapter 3 describes procedures for the production (encryption) and off-line playback (decryption) of
protected pre-recorded content.
- Chapter 4 describes the Sequence Key Block. NOTE: This section has not yet received final review by the
AACS Founders and may be subject to change.

1.4 References
This specification shall be used in conjunction with the following publications. When the publications are
superseded by an approved revision, the revision shall apply.
AACS LA, License agreement

1.5 Future Directions
With its advanced, robust cryptography, key management and renewal mechanisms, it is expected that this
technology will develop and expand, through additions to this specification, to address content protection for
additional storage types, application formats and usage models, as authorized by AACS LA.

1.6 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.

1.7 Terminology
Except where specifically noted otherwise, this document uses the same terminology as described in the
Introduction and Common Cryptographic Elements book of this specification.
1.8 Abbreviations and Acronyms

Except where specifically noted otherwise, this document uses the same abbreviations and Acronyms as described in the Introduction and Common Cryptographic Elements book of this specification.
Chapter 2
Content Revocation

2 Introduction

This chapter describes a robust mechanism whereby content on individual media can be revoked to prevent playback of unauthorized content. This is accomplished by applying cryptographic signatures to authorized content and storing those signatures on the media with the content. The signature is validated before allowing playback. A Content Revocation List (CRL) is also embedded onto media and then stored in non-volatile memory by players and contains a list of content that contains a valid signature but has since been revoked. The license agreement describes the conditions under which such revocation can occur. Figure 2-1 presents an overview of the approach, and details are provided in subsequent sections.

Figure 2-1 – Content Validation and Revocation Overview
2.1 Scope
The content verification and revocation scheme described in this chapter is in principle applicable to any pre-
recorded content having a format covered by this specification. The AACS license agreement requires that
AACS licensed products always verify that content encrypted with AACS conform to this scheme before
providing access to such content. It is anticipated that license agreements of pre-recorded video formats will
also impose such requirements regarding pre-recorded content in the respective formats that are not encrypted
by AACS, particularly for new formats. For backward-compatibility reasons, it is not expected that such
requirements will be applied to previously existing formats, such as DVD-Video formatted content that are
either unprotected or protected using CSS. Content that is encrypted by AACS, or that is not encrypted by
AACS but is in a format for which products are otherwise required to verify conformance to the scheme
described here before providing access to such content, are referred to in this specification as Certifiable;
content that does in fact conform to the scheme are referred to as Certified.

2.2 Content Signing infrastructure
Licensed Players will contain the Entity Public Key that will be used as the root of trust for validating content
signatures. Media containing Titles protected with AACS will contain the following items in addition to the
Titles themselves:

- Content Certificate
- Content Hash Table
- Content Revocation List

The Content Certificate and Content Hash Table are together used to validate the authenticity of the content and
prevent playback of that content if the signature is not valid. The Content Revocation List is a mechanism to
prevent playback of content that contains a valid signature but is not valid content.

The Content Certificate, Content Hash Table, and Content Revocation List must be stored on the pre-recorded
media with the signed content. The details and file formats for storing this information is contained in the
format specific books of this specification.

2.3 Content Hash Table
Replicators shall include with each Certified Title that they produce a Content Hash Table (CHT). The CHT
consists of a series of cryptographic hash values calculated by the replicator, which cover the complete content.
For storage media that contain more than one physical layer, the storage medium may contain a separate
Content Hash Table for each layer. Details of CHT format, calculation and storage are specific to each
supported format, and are provided in the Format-specific books of this specification.

The inclusion of a Content Hash Table, together with the use of its digest in signing the Content Certificate,
enables licensed products to verify a correspondence between the Content Certificate and the pre-recorded
content with which it is used. If this verification fails, access to such content shall be aborted. When providing
access to Certified Content, the licensed product shall calculate the series of content hash values using the same
algorithm used by the replicator, and shall abort such access if at any time a calculated hash value does not
match the corresponding hash value provided by the replicator in the CHT. Unless the licensed product has a
robust means of detecting a change of the pre-recorded storage medium, it must re-verify that the hash of the
CHT is the same as the Content Hash Table Digest contained in the Content Certificate whenever a CHT is re-
read from the medium.

2.4 Content Certificate
Licensed replicators shall include with any Certified Title that they produce a signed Content Certificate
covering that content. The Content Certificate is stored as unencrypted data on the same storage medium as the
Certified Content, as described for each supported content format elsewhere in this specification. For storage
media that contain more than one physical layer, the storage medium may contain a separate Content Certificate
for each layer. Where a storage medium contains Certified Content in more than one content format, a Content
Certificate shall be included for each format. Table 2-1 shows the format of the Content Certificate.
### Table 2-1 – Content Certificate for Pre-recorded Video

<table>
<thead>
<tr>
<th>Bit</th>
<th>0</th>
<th>1</th>
<th>2</th>
<th>3</th>
<th>4</th>
<th>5</th>
<th>6</th>
<th>7</th>
</tr>
</thead>
<tbody>
<tr>
<td>0</td>
<td></td>
<td></td>
<td></td>
<td></td>
<td></td>
<td></td>
<td></td>
<td>0 Certificate Type: 00₁₆</td>
</tr>
<tr>
<td>1</td>
<td></td>
<td></td>
<td></td>
<td></td>
<td></td>
<td></td>
<td></td>
<td>(reserved)</td>
</tr>
<tr>
<td>2</td>
<td></td>
<td></td>
<td></td>
<td></td>
<td></td>
<td></td>
<td></td>
<td>Total_Number_of_Digests</td>
</tr>
<tr>
<td>3</td>
<td></td>
<td></td>
<td></td>
<td></td>
<td></td>
<td></td>
<td></td>
<td>Layer_Number</td>
</tr>
<tr>
<td>4</td>
<td></td>
<td></td>
<td></td>
<td></td>
<td></td>
<td></td>
<td></td>
<td>Total_Number_of_Layers</td>
</tr>
<tr>
<td>5</td>
<td></td>
<td></td>
<td></td>
<td></td>
<td></td>
<td></td>
<td></td>
<td>Content Provider ID</td>
</tr>
<tr>
<td>6</td>
<td></td>
<td></td>
<td></td>
<td></td>
<td></td>
<td></td>
<td></td>
<td>Content Sequence Number</td>
</tr>
<tr>
<td>7</td>
<td></td>
<td></td>
<td></td>
<td></td>
<td></td>
<td></td>
<td></td>
<td>Minimum CRL Version</td>
</tr>
<tr>
<td>8</td>
<td></td>
<td></td>
<td></td>
<td></td>
<td></td>
<td></td>
<td></td>
<td>Replicator ID</td>
</tr>
<tr>
<td>9</td>
<td></td>
<td></td>
<td></td>
<td></td>
<td></td>
<td></td>
<td></td>
<td>Content Hash Table Digest #1</td>
</tr>
<tr>
<td>10</td>
<td></td>
<td></td>
<td></td>
<td></td>
<td></td>
<td></td>
<td></td>
<td></td>
</tr>
<tr>
<td>11</td>
<td></td>
<td></td>
<td></td>
<td></td>
<td></td>
<td></td>
<td></td>
<td></td>
</tr>
<tr>
<td>12</td>
<td></td>
<td></td>
<td></td>
<td></td>
<td></td>
<td></td>
<td></td>
<td>Content Hash Table Digest #N</td>
</tr>
<tr>
<td>13</td>
<td></td>
<td></td>
<td></td>
<td></td>
<td></td>
<td></td>
<td></td>
<td></td>
</tr>
<tr>
<td>14</td>
<td></td>
<td></td>
<td></td>
<td></td>
<td></td>
<td></td>
<td></td>
<td></td>
</tr>
<tr>
<td>15</td>
<td></td>
<td></td>
<td></td>
<td></td>
<td></td>
<td></td>
<td></td>
<td></td>
</tr>
<tr>
<td>16</td>
<td></td>
<td></td>
<td></td>
<td></td>
<td></td>
<td></td>
<td></td>
<td></td>
</tr>
<tr>
<td>17</td>
<td></td>
<td></td>
<td></td>
<td></td>
<td></td>
<td></td>
<td></td>
<td></td>
</tr>
<tr>
<td>18</td>
<td></td>
<td></td>
<td></td>
<td></td>
<td></td>
<td></td>
<td></td>
<td></td>
</tr>
<tr>
<td>19</td>
<td></td>
<td></td>
<td></td>
<td></td>
<td></td>
<td></td>
<td></td>
<td></td>
</tr>
<tr>
<td>20</td>
<td></td>
<td></td>
<td></td>
<td></td>
<td></td>
<td></td>
<td></td>
<td></td>
</tr>
<tr>
<td>21</td>
<td></td>
<td></td>
<td></td>
<td></td>
<td></td>
<td></td>
<td></td>
<td></td>
</tr>
<tr>
<td>22</td>
<td></td>
<td></td>
<td></td>
<td></td>
<td></td>
<td></td>
<td></td>
<td></td>
</tr>
<tr>
<td>23</td>
<td></td>
<td></td>
<td></td>
<td></td>
<td></td>
<td></td>
<td></td>
<td></td>
</tr>
<tr>
<td>24</td>
<td></td>
<td></td>
<td></td>
<td></td>
<td></td>
<td></td>
<td></td>
<td>Signature Data</td>
</tr>
<tr>
<td>25</td>
<td></td>
<td></td>
<td></td>
<td></td>
<td></td>
<td></td>
<td></td>
<td></td>
</tr>
<tr>
<td>26</td>
<td></td>
<td></td>
<td></td>
<td></td>
<td></td>
<td></td>
<td></td>
<td></td>
</tr>
<tr>
<td>27</td>
<td></td>
<td></td>
<td></td>
<td></td>
<td></td>
<td></td>
<td></td>
<td></td>
</tr>
<tr>
<td>28</td>
<td></td>
<td></td>
<td></td>
<td></td>
<td></td>
<td></td>
<td></td>
<td></td>
</tr>
<tr>
<td>29</td>
<td></td>
<td></td>
<td></td>
<td></td>
<td></td>
<td></td>
<td></td>
<td></td>
</tr>
<tr>
<td>30</td>
<td></td>
<td></td>
<td></td>
<td></td>
<td></td>
<td></td>
<td></td>
<td></td>
</tr>
<tr>
<td>31</td>
<td></td>
<td></td>
<td></td>
<td></td>
<td></td>
<td></td>
<td></td>
<td></td>
</tr>
</tbody>
</table>

Each Content Certificate includes:

- A 1-byte Certificate Type value, where 00₁₆ shall be used to indicate a first-generation AACS Content Certificate
- A 2-byte Total_Number_of_Digests field indicates the number of Content Hash Table Digests contained within the Content Certificate.
- A 1-byte Layer_Number field indicates the layer of the optical media for which this Content Certificate is created. NOTE: This field may not be used by every Format. Additional details on the meaning and use of this field may be provided in the Format specific books of this specification.
- A 1-byte Total_Number_of_Layers field indicates the total layers on the optical media.
- A 2-byte Content Provider ID, assigned by the AACS LA.
- A 4-byte Content Sequence Number assigned by the AACS LA to uniquely identify the Certified Content amongst that content provider’s content. The combination of the Content Provider ID and the Content Sequence Number is referred to as the Content Certificate ID. In other words, the Content Certificate ID is a 6-byte number.
Advanced Access Content System: Pre-recorded Video Book

- A 2-byte Minimum CRL Version value, assigned by the AACS LA to indicate the minimum Content Revocation List Version number that must accompany the Certified Content.
- A 2-byte Replicator ID assigned by the AACS LA to uniquely identify the Replicator.
- A series of 8-byte Content Hash Table Digests, containing the digests of the Content Hash Tables. The digest consists of the least significant 64 bits of the resulting digest from SHA-1 as described in the Introduction and Common Cryptographic Elements book of this specification.
- A 40 byte Signature Data, calculated using the Entity Private Key, over the entire data up to and including Content_Hash_Table_Digest#N.

The licensed replicator shall sign the encrypted content for each Title to be protected. The resulting Content Certificate and Content Hash Table will be included with the content on the pre-recorded medium. The AACS LA will provide to the licensed replicator a current version of the Content Revocation List which will also be included on the pre-recorded medium. The creation of the content certificate is performed by the licensed replicator. The signing of the certificate is performed by a secure facility operated by the licensing entity. The licensed replicator will submit the certificate to the secure facility and received the signed certificate back from that facility.

The AACS LA provides its Entity Public Key (which corresponds to the Entity Private Key) to each licensed manufacturer for inclusion in each licensed device or application produced. Licensed products shall treat the Entity Public Key as Integrity Protected, as defined in the license agreement. Prior to providing access to Certified Content, licensed products shall verify the signature of the Content Certificate. If at any point in the process the verification fails, such access shall be aborted. Unless the licensed product has a robust means of detecting change of the pre-recorded storage medium, it must, whenever a Content certificate is re-read from the medium, either re-verify the signature of the Certificate or robustly verify that the Certificate is the same as one whose signature it already verified, before using the Content Hash Table Digest contained therein for comparisons with the CHT.

Prior to providing access to Certified Content, licensed products shall also read and process a Content Revocation List having a List Version value equal to or greater than the Minimum CRL Version value, as described below.

### 2.5 Creating Content Certificate

A licensed replicator shall create the content certificate by the following procedure.

1. The digests of the individual units of encrypted content are computed as follows:
   \[ C_d = \text{SHA-1}(\text{Hash\_Unit})_{lsb\_64} \]
   Where Hash\_Unit is defined in the format specific Pre-recorded books of this specification and SHA-1 is the SHA hashing function as defined in Introduction and Common Cryptographic Elements book.
2. Each instance of \( C_d \) is stored in one or more Content Hash Tables (CHT) as defined in the format specific Pre-recorded books of this specification.
3. The digest of each Content Hash Table is computed as follows:
   \[ \text{CHT}_d = \text{SHA-1}(\text{CHT})_{lsb\_64} \]
4. Each instance of \( \text{CHT}_d \) is stored in a Content Certificate (CC) and the Content Certificate is cryptographically signed as follows:
   \[ \text{CC}_\text{sig} = \text{AACS\_Sign}(\text{AACS\_LA}_\text{priv}, \text{CC}) \]
   With AACS\_Sign and AACS\_LA\_priv as defined in Introduction and Common Cryptographic Elements book and the format and layout of the Content Certificate as defined in Section 2.4.

This step will be performed at a secure facility operated by the AACS LA where AACS\_LA\_priv is securely stored. The previous steps are all performed by the licensed replicator. This process is demonstrated in Figure 2-2.
Figure 2-2 – Content Signing Process

2.6 Verifying Content Certificate

As a condition of playing, copying or other use of content stored on a pre-recorded media, a licensed product shall verify the integrity of that content by the following procedure, once for each starting use of that content.

1. Selects a subset of hash units randomly from all content hash units, from which the content hash value is calculated. The licensed product can select the subset of hash units using either of the following two procedures:
   a) Selects 7 hash units randomly from all content hash units
   b) Selects the first hash unit and additionally selects at least 1% of the remaining content hash units from the position where playback begins until the end of the Title, where the hash units are randomly selected and evenly distributed throughout all content hash units. As an example, the licensed product could use a pseudo randomly generated value modulo 100 to determine which one of the next 100 hash units to verify.

2. Calculates hash value for each of the selected content hash units.
   \[ C_d = [SHA-1(Hash\_Unit)]_{lsb\_64} \]

Where Hash\_Unit is defined in the Format-specific Pre-Recorded books of this specification and SHA-1 is the SHA hashing function as defined in *Introduction and Common Cryptographic Elements* book.

3. Reads Content Hash Table (or tables), which includes the content hash values corresponding to the selected content hash units and calculates the Content Hash Table digest.
   \[ CHT_d = [SHA-1(CHT)]_{lsb\_64} \]

4. Reads Content Certificate, which includes the Content Hash Table digest (or digests) that should match the digests calculated in step 3.

5. Verifies the Signature of the Content Certificate, where the Content Hash Table digest (or digests) has been replaced with the digests calculated in step 3.
   \[ AACS\_Verify(AACS\_LA_{pub}, CC_{sig}, CC) \]
With AACS_Verify and AACS_LApub as defined in *Introduction and Common Cryptographic Elements* book.

6. The licensed product will not proceed with content playback if the signature fails to correctly verify.

7. For each selected hash unit, verifies that the calculated $C_d$ from step 2 is equal to the corresponding $C_d$ that was contained in the selected Content Hash Tables. The licensed product will not proceed with content playback if the digest of any of the selected hash units fails to match the expected digest value.

This procedure may be performed either before or after playback begins. If it is performed after playback begins and the selection process described in step 1.a is utilized, steps 1 through 7 shall be completed within the first 300 seconds of playback. If it is performed after playback begins and the selection process described in step 1.b is utilized, steps 3 through 6 shall be completed within the first 30 seconds of playback.

### 2.7 Content Revocation List (CRL)

Under certain circumstances described in the license agreement, Certified Content may be revoked. When this occurs, corresponding revocation information is added to a Content Revocation List, which the AACS LA signs using its Entity Private Key and provides to all licensed replicators. Licensed replicators shall include the most recent CRL on each pre-recorded medium that they produce containing Certified Content, in a manner consistent with normal production cycles as described in the license agreement, and as described for each supported content format elsewhere in this specification. Table 2-2 shows the format of the CRL.

Table 2-2 – Content Revocation List for Pre-recorded Video

<table>
<thead>
<tr>
<th>Byte</th>
<th>Bit</th>
<th>7</th>
<th>6</th>
<th>5</th>
<th>4</th>
<th>3</th>
<th>2</th>
<th>1</th>
<th>0</th>
</tr>
</thead>
<tbody>
<tr>
<td>0</td>
<td>List Type: $0_{16}$ (reserved)</td>
<td></td>
<td></td>
<td></td>
<td></td>
<td></td>
<td></td>
<td></td>
<td></td>
</tr>
<tr>
<td>1</td>
<td>List Version</td>
<td></td>
<td></td>
<td></td>
<td></td>
<td></td>
<td></td>
<td></td>
<td></td>
</tr>
<tr>
<td>2</td>
<td>Number of Segments (N)</td>
<td></td>
<td></td>
<td></td>
<td></td>
<td></td>
<td></td>
<td></td>
<td></td>
</tr>
<tr>
<td>3</td>
<td>Segment Size #1 ($S_1$)</td>
<td></td>
<td></td>
<td></td>
<td></td>
<td></td>
<td></td>
<td></td>
<td></td>
</tr>
<tr>
<td>4+8</td>
<td>Revocation Record Set #1</td>
<td></td>
<td></td>
<td></td>
<td></td>
<td></td>
<td></td>
<td></td>
<td></td>
</tr>
<tr>
<td>4+S</td>
<td>Entity Signature #1</td>
<td></td>
<td></td>
<td></td>
<td></td>
<td></td>
<td></td>
<td></td>
<td></td>
</tr>
<tr>
<td>X+3</td>
<td>Segment Size #N ($S_N$)</td>
<td></td>
<td></td>
<td></td>
<td></td>
<td></td>
<td></td>
<td></td>
<td></td>
</tr>
<tr>
<td>X+4</td>
<td>Revocation Record Set #N</td>
<td></td>
<td></td>
<td></td>
<td></td>
<td></td>
<td></td>
<td></td>
<td></td>
</tr>
</tbody>
</table>

---

0.90 Preliminary Draft Page 10
A CRL shall consist of:

- A 4-bit List Type value, where \(0_{16}\) shall be used to indicate a first-generation AACS Content Revocation List
- A 2-byte List Version value, which is assigned by the AACS LA to identify the version number of the CRL, and shall start at \(0000_{16}\) and increase by an increment of one with each new version
- A 1-byte Number of Segments field, which indicates the number of CRL Segments that follow, and shall be at least 1.
- One or more variable-size CRL Segments, each consisting of the following:
  - A 4-byte Segment Size field, which indicates the size of the CRL Segment in bytes, starting with the Segment Size field itself, and ending with the Entity Signature field of that CRL Segment. The Segment Size value for the first segment (CRL Segment \#1) shall be no more than 128K bytes less the CRL Header.
  - A variable-size Revocation Record Set field, which consists of one or more Revocation Records, as described below.
  - A 40-byte Signature field, which contains the result of the AACS LA signing all fields above the Signature field, including the CRL Header and previous CRL Segments, using the Entity Private Key

Prior to providing access to Certified Content, licensed products shall read at least the CRL Header and CRL Segment \#1 of the CRL provided on the pre-recorded medium with that content, and use the Entity Public Key to verify the Entity Signature found in that first segment. Licensed products that allocate more than the minimum non-volatile storage for the CRL shall also read subsequent CRL Segments, up to the amount of storage provided, if present, and verify their corresponding Entity Signature. NOTE: when reading more than one CRL Segment, only the signature of the last Segment must be checked since that signature includes all previous fields including previous Segments and the CRL Header. Licensed products shall verify that the CRL’s List Version value is equal to or greater than the Minimum CRL Version value of the Content Certificate. If any of the above-mentioned verifications fails, such access shall be aborted.

The signature of a CRL segment is verified as follows:

\[
\text{AACS\_Verify} (\text{AACS\_LA\_pub}, \text{CRL\_Seg\_sig}, \text{CRL\_Seg})
\]

With AACS\_Verify and AACS\_LA\_pub as defined in Introduction and Common Cryptographic Elements book.

A licensed product must retain in non-volatile storage, the CRL data with the highest valued List version which it encounters and has verified. Therefore, when a CRL (or a portion thereof) is successfully verified as described above, the licensed product shall compare its List Version value to the List Version value of its persistently stored CRL data, if any. If

- no CRL data was previously stored, or
- if the List Version value of the previously stored CRL data is lower than that of the newly read CRL, or
- if the List Version values are the same but the newly read CRL data is larger (more complete) than the previously stored CRL data and the product is able to store more CRL data than it previously stored,

then the licensed product shall replace the previously stored CRL data, if any, with the newly read CRL data. Then, using its persistently stored CRL data, the licensed product shall examine the list to see if the Content Certificate ID to be accessed is revoked and if so, such access shall be aborted.
When persistently storing CRL data, licensed products shall have at least 128K bytes of non-volatile storage for that purpose. That size is sufficient to store the List Version field of the CRL and the Revocation Record Set field of CRL Segment #1. Licensed products that allocate more than 128K bytes of non-volatile storage shall store other fields of the CRL which fit in the additional storage, including the Revocation Record Set fields of other CRL Segments, if present. Where a licensed product persistently stores some but not all Revocation Record Set fields of a CRL, it shall give priority to storing the Revocation Record Set fields of the lowest-numbered CRL Segments. In all cases, licensed products shall treat CRL List Version and Revocation Record Set fields as Integrity Protected, as defined in the license agreement. Table 2-3 shows the format of a Revocation Record for Content Certificate ID.

### Table 2-3 – Revocation Record for Content Certificate ID

<table>
<thead>
<tr>
<th>Bit</th>
<th>7</th>
<th>6</th>
<th>5</th>
<th>4</th>
<th>3</th>
<th>2</th>
<th>1</th>
<th>0</th>
</tr>
</thead>
<tbody>
<tr>
<td>0</td>
<td></td>
<td></td>
<td></td>
<td></td>
<td>Record Type: 00₁₆</td>
<td></td>
<td></td>
<td>Range (bits 11-8)</td>
</tr>
<tr>
<td>1</td>
<td></td>
<td></td>
<td></td>
<td></td>
<td></td>
<td>Range (bits 7-0)</td>
<td></td>
<td></td>
</tr>
<tr>
<td>2</td>
<td></td>
<td></td>
<td>Content Certificate ID</td>
<td></td>
<td></td>
<td></td>
<td></td>
<td></td>
</tr>
<tr>
<td>:</td>
<td></td>
<td></td>
<td></td>
<td></td>
<td></td>
<td></td>
<td></td>
<td></td>
</tr>
<tr>
<td>7</td>
<td></td>
<td></td>
<td></td>
<td></td>
<td></td>
<td></td>
<td></td>
<td></td>
</tr>
</tbody>
</table>

A Revocation Record for Content Certificate ID shall consist of:

- A 4-bit Record Type value, where 00₁₆ shall be used to indicate a Revocation Record for Content Certificate ID.
- A 12-bit Range value indicates the range of revoked Content Certificate ID’s starting from the ID contained in the record. A value of zero in the Range field indicates that only one ID is being revoked.
- A 6-byte Content Certificate ID value (the concatenation of the 2-byte Content Provider ID and the 4-byte Content Sequence Number) indicating the lowest numbered Content Certificate ID to be revoked.

If a licensed product encounters a Revocation Record with a Record Type value it does not recognize, the record shall be ignored. Other valid records in the CRL shall still be processed, however. Revocation Records within a CRL will not necessarily occur in order of their Content Certificate ID field values.
Chapter 3
Content Encryption and Decryption

3 Introduction
This chapter specifies the procedures for encryption and decryption of pre-recorded video content protected by AACS. Figure 3-1 presents an overview, and the remainder of the chapter describes the encryption and decryption processes in detail.

![Figure 3-1 – Encryption and Decryption Overview](image)

3.1 Content Encryption (General)
The owner of content that is to be protected provides the content in the form of one or more Titles, and their associated Usage Rules, to the licensed replicator.

The licensed replicator shall select a secret, random Title Key for each Title to be protected. Each Title Key shall be used to encrypt the content of its corresponding Title, as specified for each supported content format elsewhere in this specification. At the replicator’s discretion, a given Title may be encrypted using the same Title Key for all instances of pre-recorded media, or different Title Keys may be used for different instances.

The licensed replicator shall also assign a secret, unpredictable (e.g., random) identifier to the protected Title or set of protected Titles to be included together on a pre-recorded medium. This identifier, referred to as the Volume ID, is used as a safeguard against “bit-by-bit copying” of protected content, and is therefore stored on the pre-recorded medium in a manner that cannot be duplicated by consumer recorders, as specified for each supported storage format elsewhere in this specification. At the licensed replicator’s discretion, the same Volume ID may be used for all instances of pre-recorded media containing a given protected Title or set of protected Titles, or different values may be assigned for different instances.

For each protected Title or set of protected Titles to be included together on a pre-recorded medium, the AACS LA provides to the licensed replicator a Media Key Block (MKB), a secret Media Key, a Sequence Key Key Block

0.90 Preliminary Draft
(SKB), and a corresponding set of secret Media Key Variants. The Media Key Block will enable all compliant devices, each using their set of secret Device Keys, to calculate the same Media Key as described in the *Introduction and Common Cryptographic Elements* book of this specification. If a set of Device Keys is compromised in a way that threatens the integrity of the system, an updated MKB can be released that will cause a device with the compromised set of Device Keys to calculate a different Media Key than is computed by the remaining compliant devices. In this way, the compromised Device Keys are “revoked” by the new MKB.

For each protected Title, the licensed replicator calculates a cryptographic hash of the Media Key and/or the Media Key Variants and the Volume ID, and uses the result to encrypt the Title’s Title Key. The encrypted content, Encrypted Title Keys, Usage Rules and MKB are stored on the pre-recorded medium as specified for each supported storage/content format elsewhere in this specification.

### 3.2 Content Decryption (General)

The AACS LA provides a set of 253 secret Device Keys, denoted \(K_{d,0}, K_{d,1}, \ldots, K_{d,n-1}\), to the licensed manufacturer for inclusion into each compliant device or application produced. Device Key sets may either be unique per licensed product, or used commonly by multiple products; the license agreement describes the details and requirements associated with these two alternatives. A licensed product shall treat its Device Keys as highly confidential, as defined in the license agreement.

The licensed product reads the MKB from the pre-recorded medium, and uses its Device Keys to process the MKB and thereby calculate the Media Key, as described in the *Introduction and Common Cryptographic Elements* book of this specification. If the given set of Device Keys has not been revoked, then the calculated Media Key will be the same Media Key that was used by the licensed replicator as described above.

The AACS LA also provides a set of 256 Sequence Keys to the licensed manufacturer for inclusion in each compliant device or application produced. Like Device Keys, Sequence Keys may be either unique per licensed product, or used in common by multiple products. If a product’s Device Keys are unique, then its Sequence Keys will also be unique; if a product’s Device Keys are in common, then its Sequence Keys will also be in common. In either event, the licensed product reads the SKB from the prerecorded medium and uses its Sequence Keys and its previously calculated Media Key to calculate its Media Key Variant.

For each protected Title the licensed product then calculates a cryptographic hash of the calculated Media Key and/or the Media Key Variants and the Volume ID, and uses the result to decrypt the Title’s Encrypted Title Key. The result is then used to decrypt the Title, as specified for each supported format elsewhere in this specification.

The licensed product verifies that the Content Certificate ID has not been included on a revocation list before allowing playback of that Title.

### 3.3 Calculating the Volume Unique Keys

The Volume Unique Key (\(K_{vu}\)) and/or the Volume Variant Unique Keys (\(K_{vvu}\)) are used to encrypt and decrypt the Title Keys stored on the pre-recorded media, in a manner that is described in the given Format-specific book of this specification.

The Volume Unique Key is calculated as follows:

1. Calculate the Media Key (\(K_m\)):

The \(K_m\) and the Media Key Block are delivered directly to licensed replicators. The licensed replicator embeds the MKB onto the media to enable licensed players to derive \(K_m\).

2. Calculate the Volume Unique Key (\(K_{vu}\)):

The licensed replicator chooses a Volume ID (\(ID_v\)) to be placed on the pre-recorded medium and calculates a Volume Unique Key (\(K_{vu}\)) as follows:

\[
K_{vu} = AES-G(K_m, ID_v)
\]

where AES-G represents the AES-based one-way function defined in the *Introduction and Common Cryptographic Elements* book.

The Volume Variant Unique Key is calculated as follows:
1. Calculate the Media Key Variant ($K_{mv}$):
The $K_{mv}$’s, and the Sequence Key Block are delivered directly to licensed replicators. The licensed replicator embeds SKB onto the media to enable licensed players to derive its particular $K_{mv}$, as described in Chapter 4.

2. Calculate the Volume Variant Unique Key ($K_{vvu}$):
The licensed replicator calculates a Volume Variant Unique Key ($K_{vvu}$) as follows:

$$K_{vvu} = AES-G(K_{mv}, ID_v).$$

### 3.4 AACS Encryption on Pre-Recorded Media

The following steps detail the minimum procedures for encrypting AACS content on pre-recorded media as illustrated in Figure 3-1 above.

1. **Generate the Title Key**
The licensed replicator generates a statistically unique 128-bit Title Key ($K_t$).

2. **Encrypt the content**
The Title Key is used to encrypt the Content (C) as follows:

$$C_e = AES-128CBC(E(K_t, C))$$

where AES-128CBC represents encryption by the AES algorithm in CBC mode as defined in the *Introduction and Common Cryptographic Elements* book.

3. **Sign the content**
The encrypted content is signed using the procedure defined in Section 2.5.

4. **Encrypt the Title Key(s)**
The Title Key(s) is encrypted ($K_{te}$) as follows:

$$K_{te} = AES-128E(K_u, K_t)$$

where AES-128E represents encryption by the AES algorithm in ECB mode as defined in the *Introduction and Common Cryptographic Elements* book and the $K_u$’s is one of the volume unique keys defined in Section 3.3. There may be more than one encryption at this step, if a Title Key is encrypted in all of the Volume Variant Unique Keys.

5. **Generate a Message Authentication Code of the Usage Rules using the Title Key**
The licensed replicator uses the primary Title Key to generate a MAC on the Usage Rules as described in the *Introduction and Common Cryptographic Elements* book.

$$MAC_{ur} = CMAC(K_t, Usage\_Rules)$$

6. **Transfer the data**
The licensed replicator stores the encrypted Title Keys ($K_{te}$), Usage Rules, the Usage Rules MAC, Content Certificate, and the encrypted AACS content to the pre-recorded media in the location and manner as specified for each supported Format elsewhere in this specification.

### 3.5 AACS Decryption on Pre-Recorded Media

The following steps detail the minimum procedures for decrypting AACS content on pre-recorded media as illustrated in Figure 3-1 above.

1. **Decrypt the Title Key(s)**
The Title Key(s) is decrypted ($K_t$) as follows:

$$K_t = AES-128D(K_u, K_{te})$$

where AES-128D represents decryption by the AES algorithm in ECB mode as defined in the *Introduction and Common Cryptographic Elements* book and $K_u$ is one of the volume unique keys defined in Section 3.3.

2. **Verify the Usage Rules**
The Usage Rule MAC is checked using the Title Key calculated in step 1. If the verification fails, the content shall not be decrypted nor played.
MAC_{ur} == CMAC(K_o, Usage_Rules)

3. Verify content is not revoked

The Content Certificate ID is checked for revocation status as defined in Section 2.7. If the content has been revoked, the content shall not be decrypted nor played.

4. Verify content signature

The encrypted content is verified as defined in Section 2.6. If the verification fails, the content shall not be further decrypted nor played.

5. Decrypt the content

The Title Key is used to decrypt the Content (C) as follows:

\[ C = \text{AES-128CBCD}(K_t, C_e) \]

where AES-128CBCD represents decryption by the AES algorithm in CBC mode as defined in the *Introduction and Common Cryptographic Elements* book.

Steps 4 and 5 can be performed in parallel as defined in Section 2.6.
Chapter 4
Sequence Key Block

4 Introduction

This chapter describes the use of device Sequence Keys and the Sequence Key Block (SKB) on pre-recorded media. Fundamentally, AACS protection depends on Device Keys and the tree-based Media Key Block, which allows unlimited, precise revocation without danger of collateral damage to innocent devices. Because of the inherent power of the revocation of the AACS system, it is possible that attackers may forgo building clones or non-compliant devices and instead devote themselves to attacks where they try to hide the underlying compromised device(s). These attacks are both more expensive and more legally risky for the attackers, because the attacks require them to have an active server serving either content keys or the content itself, on an instance-by-instance basis.

It is possible that non-technical means could stop these “anonymous attack” servers. Nonetheless, AACS has developed a technology to accurately determine the underlying compromised devices that these servers are using, so the AACS revocation mechanism can be brought to bear. The technology is called Sequence Keys and the Sequence Key Block.

Sets of Sequence Keys are assigned to individual devices by the AACS LA out of a matrix of keys. The AACS LA also assigns the Sequence Key Blocks to be used on the pre-recorded media. In this respect, the key management aspects of Sequence Keys are identical to the technology developed by 4C Entity, LLC, called Content Protection for Recordable Media (CPRM). Sequence Key Blocks are very similar to CPRM Media Key Blocks; the only differences arise from the different ciphers used (AES instead of C2). However, unlike CPRM MKBs, the AACS SKBs are not part of the fundamental cryptographic protection of the content. The fundamental protection of AACS is the Media Key; the SKB merely allows different variants of the Media Key to be calculated by different devices.

The remainder of this section describes the Sequence Key Block in detail.
4.1 Sequence Key Block Principles

This is an informative section intended to give an overview of how Sequence Key Blocks work.

The AACS licensing agency will generate Sequence Keys organized in a large matrix. The matrix has 256 columns and not more than 65,536 rows. Each cell is a different Sequence Key. A single device has one key in each column. Thus, each device has 256 Sequence Keys.

Attackers would prefer to use already-compromised Sequence Keys if they could, so that no new forensic information could be deduced by the licensing agency. Therefore, it is important that compromised keys are no longer usable by the attackers. The problem is that many thousands of devices might share a single compromised key. Therefore, revocation of a single key is impossible. On the other hand, revocation of a unique set of keys is very possible; in fact, that is precisely what the SKB achieves. The fundamental principle is that no two devices have many keys in common, so even if the system has been heavily attacked and a significant fraction of the Sequence Keys is compromised, all innocent devices will have many columns in which they have uncompromised keys. The purpose of the Sequence Key Block is to give all innocent devices a column they can use to calculate the correct answer, while at the same time preventing compromised devices (who have compromised keys in all columns) from getting to the same answer.

In an SKB there are actually many correct answers, one for each variation in the content. For the purpose of explanation, however, it is helpful to imagine that a single SKB is producing a single answer. We will call that answer the output key. Then the SKB mechanism is completely identical to the CPRM/CPPM mechanism.

As shown in Figure 4-1 above, the SKB begins with a first column, called the “unconditional” column. By “column”, we mean a column of Sequence Keys in the matrix will be used to encrypt. (To be precise, the key used to encrypt is derived from the Sequence Key, not the Sequence Key itself.) The first column will have an encryption of the output key (denoted ‘K’ in the figure) in every uncompromised Sequence Key’s cell. Devices that do not have compromised keys in that column immediately decrypt the output key. Devices, both innocent and otherwise, that do have compromised keys instead decrypt a key called a link key that allows them to process a further column in the SKB. To process the further column they need both the link key and their Sequence Key in that column. Thus the subsequent columns are called “conditional” columns because they can only be processed by the device if it were given the necessary link key in a previous column.

Figure 4-1 – Example Sequence Key Block

As shown in Figure 4-1 above, the SKB begins with a first column, called the “unconditional” column. By “column”, we mean a column of Sequence Keys in the matrix will be used to encrypt. (To be precise, the key used to encrypt is derived from the Sequence Key, not the Sequence Key itself.) The first column will have an encryption of the output key (denoted ‘K’ in the figure) in every uncompromised Sequence Key’s cell. Devices that do not have compromised keys in that column immediately decrypt the output key. Devices, both innocent and otherwise, that do have compromised keys instead decrypt a key called a link key that allows them to process a further column in the SKB. To process the further column they need both the link key and their Sequence Key in that column. Thus the subsequent columns are called “conditional” columns because they can only be processed by the device if it were given the necessary link key in a previous column.
The subsequent additional conditional columns are produced the same way as the first column: They will have an encryption of the output key in every uncompromised Sequence Key’s cell. Devices with a compromised key will get a further link key to another column instead of the output key. However, after some number of columns depending on the actual number of compromised keys, the AACS licensing agency will know that only compromised devices would be getting the link key—all innocent devices would have found the output key in this column or in a previous column. At this point, rather than encrypting a link key, the agency encrypts a 0, and the SKB is complete. All innocent devices will have decrypted the output key, and all compromised devices have ended up decrypting 0.

How do the devices know they have a link key versus the output key? The short answer is they do not, at least not at first. Each conditional column has a header of known data (DEADBEEF16) encrypted in the link key for that column. The device decrypts the header with the key it currently has. If the header decrypts correctly, the device knows it has a link key and processes the column. If it does not decrypt correctly, the device knows it has either the output key or a link key for a different column. When the device reaches the end of the SKB without decrypting 0, it knows it must have an output key. Note that this device logic allows the licensing agency to send different populations of devices to different columns by having more than one link key output from a single column. For example, in the figure, column (1) links to both column (2) and column (5). This flexibility can help against certain types of attacks.

The preceding description is equally accurate for an AACS SKB as it is for a CPRM/CPPM Media Key Block, with the exception that in the AACS SKB there is not a single output key, but multiple output keys called Variant Data.

4.2 Calculation of the Media Key Variant Data

4.2.1 Sequence Keys

Each AACS compliant device capable of playing pre-recorded content is given a set of secret Sequence Keys when manufactured. These are in addition to the Device Keys that all AACS devices require. These Sequence Keys are provided by the AACS LA and are for use in processing the Sequence Key Block. The result of the calculation is Variant Data which is then combined with the Media Key from the Media Key Block to generate the Media Key Variant, as explained in section 4.3. Key sets may either be unique per device, or used commonly by multiple devices. The AACS license agreement describes the details and requirements associated with these two options.

Each device receives 256 64-bit Sequence Keys, which are referred to as $K_{s\_i}$ (i=0,1,…,255). For each Sequence Key there is an associated Column and Row value, referred to as $C_{s\_j}$ (i=0,1,…,n-1) and $R_{s\_j}$ (j=0,1,…,m-1) respectively. Column and Row values start at 0. For a given device, no two Sequence Keys will have the same associated Column value (in other words, a device will have at most one Sequence Key per Column). It is possible for a device to have some Sequence Keys with the same associated Row values.

A device uses a Sequence Key $K_{s\_j}$ together with the Media Key $K_m$ to calculate the Media Sequence Key $K_{m\_s\_j}$ as follows:

$$K_{m\_s\_j} = AES-G(K_m, K_{s\_j} || 0302153EE3EC752416)$$

The Media Sequence Keys serve the role that the device keys in CPRM. In other words, the device does not use its Sequence Key directly to decrypt; instead, it combines it with the Media Key first as shown above. That means that a given SKB is associated with a given MKB (because the SKB depends on the Media Key for correct processing).

A device shall treat its Sequence Keys as highly confidential, and their associated Row values as confidential, as defined in the AACS license agreement.

4.2.2 Sequence Key Block (SKB)

The SKB is generated by the AACS LA and allows all compliant devices, each using their set of secret Sequence Keys and the Media Key, to calculate the Variant Data, $D_v$, which in turn allows them to calculate the Media Key Variant. If a set of Sequence Keys is compromised in a way that threatens the integrity of the system, an updated SKB can be released that causes a device with one or more compromised sets of Sequence
Keys to calculate invalid Variant Data. In this way, the compromised Sequence Keys are “revoked” by the new SKB.

An SKB is formatted as a sequence of contiguous Records. Each Record begins with a one-byte Record Type field, followed by a three-byte Record Length field. The Record Type field value indicates the type of the Record, and the Record Length field value indicates the number of bytes in the Record, including the Record Type and the Record Length fields themselves. Record lengths are always multiples of 4 bytes. The Record Type and Record Length fields are never encrypted. Subsequent fields in a Record may be encrypted (by the AES cipher in ECB mode), depending on the Record Type.

Using its Sequence Keys, a device calculates \( D_v \) by processing Records of the SKB one-by-one, in order, from first to last. Except where explicitly noted otherwise, a device must process every Record of the SKB. The device must not make any assumptions about the length of Records, and must instead use the Record Length field value to go from one Record to the next. If a device encounters a Record with a Record Type field value it does not recognize, it ignores that Record and skips to the next. For some Records, processing will result in the calculation of a \( D_v \) value. Processing of subsequent Records may update the \( D_v \) value that was calculated previously. After processing of the SKB is completed, the device uses the most recently calculated \( D_v \) value as the final value for \( D_v \).

If a device correctly processes an SKB using Sequence Keys that are revoked by that SKB, the resulting final \( D_v \) will have the special value 0000000000000000000016. This special value will never be an SKB’s correct final \( D_v \) value, and can therefore always be taken as an indication that the device’s Sequence Keys are revoked. Device behavior in this situation is implementation defined. As an example, a device could exhibit a special diagnostic code, as information to a service technician.

The following subsections describe the currently defined Record types, and how a device processes each.

### 4.2.2.1 Verify Media Key Record

<table>
<thead>
<tr>
<th>Bit</th>
<th>7</th>
<th>6</th>
<th>5</th>
<th>4</th>
<th>3</th>
<th>2</th>
<th>1</th>
<th>0</th>
</tr>
</thead>
<tbody>
<tr>
<td>0</td>
<td></td>
<td></td>
<td></td>
<td></td>
<td></td>
<td></td>
<td></td>
<td></td>
</tr>
<tr>
<td>1</td>
<td></td>
<td></td>
<td></td>
<td></td>
<td></td>
<td></td>
<td></td>
<td></td>
</tr>
<tr>
<td>2</td>
<td></td>
<td></td>
<td></td>
<td></td>
<td></td>
<td></td>
<td></td>
<td></td>
</tr>
<tr>
<td>3</td>
<td></td>
<td></td>
<td></td>
<td></td>
<td></td>
<td></td>
<td></td>
<td></td>
</tr>
<tr>
<td>4</td>
<td></td>
<td></td>
<td></td>
<td></td>
<td></td>
<td></td>
<td></td>
<td></td>
</tr>
<tr>
<td>...</td>
<td></td>
<td></td>
<td></td>
<td></td>
<td></td>
<td></td>
<td></td>
<td></td>
</tr>
<tr>
<td>19</td>
<td></td>
<td></td>
<td></td>
<td></td>
<td></td>
<td></td>
<td></td>
<td></td>
</tr>
</tbody>
</table>

**Table 4-1 – Verify Media Key Record Format**

A properly formatted SKB shall have exactly one Verify Media Key Record as its first Record. Bytes 4 through 19 of the Record contain the ciphertext value

\[
D_x = \text{AES-128E} (K_m, 0123456789ABCDEF16 || \text{XXXXXXXXXXXXXXXXX16})
\]

where \text{XXXXXXXXXXXXXXXXX16} is an arbitrary 8-byte value, and \( K_m \) is the correct final Media Key value.

The presence of the Verify Media Key Record in an SKB is mandatory, but the use of the Record by a device is not mandatory. The device may use the Verify Media Key Record to make sure that it has the correct Media Key for processing the SKB. The Media Key comes from calculation based on the separate Media Key Block. Since MKBs and SKBs are associated, the device can, in effect, verify it has the correct MKB/SKB pair. The device action in the case of a mismatched MKB/SKB pair is manufacturer-specific. In any event, the device will not be able to process the content correctly.

If everything is correct, the device should observe the condition:
\[\text{AES}_{128}(K_m, D_x)_{\text{msb}_64} = 0123456789\text{ABCDEF}_{16}\]

where $K_m$ is the Media Key value.
4.2.2.2 Nonce Record

Table 4-2 – Nonce Record Format

<table>
<thead>
<tr>
<th>Bit Byte</th>
<th>7</th>
<th>6</th>
<th>5</th>
<th>4</th>
<th>3</th>
<th>2</th>
<th>1</th>
<th>0</th>
</tr>
</thead>
<tbody>
<tr>
<td>0</td>
<td></td>
<td></td>
<td></td>
<td></td>
<td></td>
<td></td>
<td></td>
<td></td>
</tr>
<tr>
<td>1</td>
<td></td>
<td></td>
<td></td>
<td></td>
<td></td>
<td></td>
<td></td>
<td></td>
</tr>
<tr>
<td>2</td>
<td></td>
<td></td>
<td></td>
<td></td>
<td></td>
<td></td>
<td></td>
<td></td>
</tr>
<tr>
<td>3</td>
<td></td>
<td></td>
<td></td>
<td></td>
<td></td>
<td></td>
<td></td>
<td></td>
</tr>
<tr>
<td>4</td>
<td></td>
<td></td>
<td></td>
<td></td>
<td></td>
<td></td>
<td></td>
<td></td>
</tr>
<tr>
<td>...</td>
<td></td>
<td></td>
<td></td>
<td></td>
<td></td>
<td></td>
<td></td>
<td></td>
</tr>
<tr>
<td>19</td>
<td></td>
<td></td>
<td></td>
<td></td>
<td></td>
<td></td>
<td></td>
<td></td>
</tr>
</tbody>
</table>

A properly formatted SKB shall have exactly one Nonce Record. The nonce number X is used in the Variant Data calculation as described below. The nonce record will always precede the Calculate Variant Data Record and the Conditionally Calculate Variant Data Records in the SKB, although it may not immediately precede them.
### 4.2.2.3 Calculate Variant Data Record

Table 4-3 shows the format of a Calculate Variant Data Record.

<table>
<thead>
<tr>
<th>Bit</th>
<th>Byte</th>
<th>7</th>
<th>6</th>
<th>5</th>
<th>4</th>
<th>3</th>
<th>2</th>
<th>1</th>
<th>0</th>
</tr>
</thead>
<tbody>
<tr>
<td>0</td>
<td></td>
<td></td>
<td></td>
<td></td>
<td></td>
<td></td>
<td></td>
<td></td>
<td></td>
</tr>
<tr>
<td>1</td>
<td></td>
<td></td>
<td></td>
<td></td>
<td></td>
<td></td>
<td></td>
<td></td>
<td></td>
</tr>
<tr>
<td>2</td>
<td></td>
<td></td>
<td></td>
<td></td>
<td></td>
<td></td>
<td></td>
<td></td>
<td></td>
</tr>
<tr>
<td>3</td>
<td></td>
<td></td>
<td></td>
<td></td>
<td></td>
<td></td>
<td></td>
<td></td>
<td></td>
</tr>
<tr>
<td>4</td>
<td></td>
<td></td>
<td></td>
<td></td>
<td></td>
<td></td>
<td></td>
<td></td>
<td></td>
</tr>
<tr>
<td>...</td>
<td></td>
<td></td>
<td></td>
<td></td>
<td></td>
<td></td>
<td></td>
<td></td>
<td></td>
</tr>
<tr>
<td>7</td>
<td></td>
<td></td>
<td></td>
<td></td>
<td></td>
<td></td>
<td></td>
<td></td>
<td></td>
</tr>
<tr>
<td>8</td>
<td></td>
<td></td>
<td></td>
<td></td>
<td></td>
<td></td>
<td></td>
<td></td>
<td></td>
</tr>
<tr>
<td>9</td>
<td></td>
<td></td>
<td></td>
<td></td>
<td></td>
<td></td>
<td></td>
<td></td>
<td></td>
</tr>
<tr>
<td>10</td>
<td></td>
<td></td>
<td></td>
<td></td>
<td></td>
<td></td>
<td></td>
<td></td>
<td></td>
</tr>
<tr>
<td>11</td>
<td></td>
<td></td>
<td></td>
<td></td>
<td></td>
<td></td>
<td></td>
<td></td>
<td></td>
</tr>
<tr>
<td>12</td>
<td></td>
<td></td>
<td></td>
<td></td>
<td></td>
<td></td>
<td></td>
<td></td>
<td></td>
</tr>
<tr>
<td>...</td>
<td></td>
<td></td>
<td></td>
<td></td>
<td></td>
<td></td>
<td></td>
<td></td>
<td></td>
</tr>
<tr>
<td>19</td>
<td></td>
<td></td>
<td></td>
<td></td>
<td></td>
<td></td>
<td></td>
<td></td>
<td></td>
</tr>
<tr>
<td>20</td>
<td></td>
<td></td>
<td></td>
<td></td>
<td></td>
<td></td>
<td></td>
<td></td>
<td></td>
</tr>
<tr>
<td>...</td>
<td></td>
<td></td>
<td></td>
<td></td>
<td></td>
<td></td>
<td></td>
<td></td>
<td></td>
</tr>
<tr>
<td>29</td>
<td></td>
<td></td>
<td></td>
<td></td>
<td></td>
<td></td>
<td></td>
<td></td>
<td></td>
</tr>
<tr>
<td>30</td>
<td></td>
<td></td>
<td></td>
<td></td>
<td></td>
<td></td>
<td></td>
<td></td>
<td></td>
</tr>
<tr>
<td>...</td>
<td></td>
<td></td>
<td></td>
<td></td>
<td></td>
<td></td>
<td></td>
<td></td>
<td></td>
</tr>
<tr>
<td>39</td>
<td></td>
<td></td>
<td></td>
<td></td>
<td></td>
<td></td>
<td></td>
<td></td>
<td></td>
</tr>
<tr>
<td>40</td>
<td></td>
<td></td>
<td></td>
<td></td>
<td></td>
<td></td>
<td></td>
<td></td>
<td></td>
</tr>
<tr>
<td>...</td>
<td></td>
<td></td>
<td></td>
<td></td>
<td></td>
<td></td>
<td></td>
<td></td>
<td></td>
</tr>
<tr>
<td>Length-1</td>
<td></td>
<td></td>
<td></td>
<td></td>
<td></td>
<td></td>
<td></td>
<td></td>
<td></td>
</tr>
</tbody>
</table>

**Table 4-3 – Calculate Variant Data Record Format**

A properly formatted SKB shall have exactly one Calculate Variant Data Record. Devices must ignore any Calculate Variant Data Records encountered after the first one in an SKB. The use of the reserved fields is currently undefined, and they shall be ignored. The Generation field shall contain 0001₁₆ for the first generation. The Column field indicates the associated Column value for the Sequence Key to be used with this Record, as described below. Bytes 20 and higher contain Encrypted Key Data (possibly followed by some padding bytes at the end of the Record, not shown in Table 4-3). The first ten bytes of the Encrypted Key Data correspond to Sequence Key Row 0; the next ten bytes correspond to Sequence Key Row 1; and so forth.

Before processing the Record, the device checks that both of the following conditions are true:

- Generation == 0001₁₆
- the device has a Sequence Key with associated Column value C_{d,i} == Column, for some i.

If either of these conditions is false, the device ignores the rest of the Record.
Otherwise, using the value $i$ from the condition above, the value $X$ from the Nonce Record, and $r = R_d, c = C_d, i$, the device calculates:

$$D_v = [\text{AES-G}(K_{ms_i}, X \oplus f(c,r))]_{\text{msb,80}} \oplus D_{ke,r}$$

where $K_{ms_i}$ is the device’s $i$-th Media Sequence Key’s value and $D_{ke,r}$ is the 80-bit value starting at byte offset $r \times 10$ within the Record’s Encrypted Key Data. $f(c,r)$ represents the 128-bit value:

$$f(c,r) = 0000_{16} \ || \ c \ || \ 0000_{16} \ || \ r \ || \ 0000000000000000_{16}$$

where $c$ and $r$ are left-padded to lengths 16 bits, by prepending zero-valued bits to each as needed. The resulting $D_v$ becomes the current Variant Data value.

It is not necessary for a first generation device to verify that Record Length is sufficient to index into the Encrypted Key Data. First generation devices are assured that the Encrypted Key Data contains a value corresponding to their Device Key’s associated Row value.
4.2.2.4 Conditionally Calculate Variant Data Record

Table 4-4 shows the format of a *Conditionally Calculate Variant Data* Record.

<table>
<thead>
<tr>
<th>Bit</th>
<th>Byte</th>
<th>7</th>
<th>6</th>
<th>5</th>
<th>4</th>
<th>3</th>
<th>2</th>
<th>1</th>
<th>0</th>
</tr>
</thead>
<tbody>
<tr>
<td>0</td>
<td></td>
<td></td>
<td></td>
<td></td>
<td></td>
<td></td>
<td></td>
<td></td>
<td></td>
</tr>
<tr>
<td>1</td>
<td></td>
<td></td>
<td></td>
<td></td>
<td></td>
<td></td>
<td></td>
<td></td>
<td></td>
</tr>
<tr>
<td>2</td>
<td></td>
<td></td>
<td></td>
<td></td>
<td></td>
<td></td>
<td></td>
<td></td>
<td></td>
</tr>
<tr>
<td>3</td>
<td></td>
<td></td>
<td></td>
<td></td>
<td></td>
<td></td>
<td></td>
<td></td>
<td></td>
</tr>
<tr>
<td>4</td>
<td></td>
<td></td>
<td></td>
<td></td>
<td></td>
<td></td>
<td></td>
<td></td>
<td></td>
</tr>
<tr>
<td>...</td>
<td></td>
<td></td>
<td></td>
<td></td>
<td></td>
<td></td>
<td></td>
<td></td>
<td></td>
</tr>
<tr>
<td>7</td>
<td></td>
<td></td>
<td></td>
<td></td>
<td></td>
<td></td>
<td></td>
<td></td>
<td></td>
</tr>
<tr>
<td>8</td>
<td></td>
<td></td>
<td></td>
<td></td>
<td></td>
<td></td>
<td></td>
<td></td>
<td></td>
</tr>
<tr>
<td>9</td>
<td></td>
<td></td>
<td></td>
<td></td>
<td></td>
<td></td>
<td></td>
<td></td>
<td></td>
</tr>
<tr>
<td>10</td>
<td></td>
<td></td>
<td></td>
<td></td>
<td></td>
<td></td>
<td></td>
<td></td>
<td></td>
</tr>
<tr>
<td>11</td>
<td></td>
<td></td>
<td></td>
<td></td>
<td></td>
<td></td>
<td></td>
<td></td>
<td></td>
</tr>
<tr>
<td>12</td>
<td></td>
<td></td>
<td></td>
<td></td>
<td></td>
<td></td>
<td></td>
<td></td>
<td></td>
</tr>
<tr>
<td>...</td>
<td></td>
<td></td>
<td></td>
<td></td>
<td></td>
<td></td>
<td></td>
<td></td>
<td></td>
</tr>
<tr>
<td>19</td>
<td></td>
<td></td>
<td></td>
<td></td>
<td></td>
<td></td>
<td></td>
<td></td>
<td></td>
</tr>
<tr>
<td>20</td>
<td></td>
<td></td>
<td></td>
<td></td>
<td></td>
<td></td>
<td></td>
<td></td>
<td></td>
</tr>
<tr>
<td></td>
<td>Encrypted Conditional Data ((D_{e}))</td>
<td></td>
<td></td>
<td></td>
<td></td>
<td></td>
<td></td>
<td></td>
<td></td>
</tr>
<tr>
<td>29</td>
<td></td>
<td></td>
<td></td>
<td></td>
<td></td>
<td></td>
<td></td>
<td></td>
<td></td>
</tr>
<tr>
<td>30</td>
<td></td>
<td></td>
<td></td>
<td></td>
<td></td>
<td></td>
<td></td>
<td></td>
<td></td>
</tr>
<tr>
<td></td>
<td>Doubly Encrypted Variant Data for Row 0 ((D_{kde,0}))</td>
<td></td>
<td></td>
<td></td>
<td></td>
<td></td>
<td></td>
<td></td>
<td></td>
</tr>
<tr>
<td>39</td>
<td></td>
<td></td>
<td></td>
<td></td>
<td></td>
<td></td>
<td></td>
<td></td>
<td></td>
</tr>
<tr>
<td></td>
<td>Doubly Encrypted Variant Data for Row 1 ((D_{kde,1}))</td>
<td></td>
<td></td>
<td></td>
<td></td>
<td></td>
<td></td>
<td></td>
<td></td>
</tr>
<tr>
<td>39</td>
<td></td>
<td></td>
<td></td>
<td></td>
<td></td>
<td></td>
<td></td>
<td></td>
<td></td>
</tr>
<tr>
<td>40</td>
<td></td>
<td></td>
<td></td>
<td></td>
<td></td>
<td></td>
<td></td>
<td></td>
<td></td>
</tr>
<tr>
<td></td>
<td>Doubly Encrypted Variant Data Length-1</td>
<td></td>
<td></td>
<td></td>
<td></td>
<td></td>
<td></td>
<td></td>
<td></td>
</tr>
<tr>
<td></td>
<td>.</td>
<td></td>
<td></td>
<td></td>
<td></td>
<td></td>
<td></td>
<td></td>
<td></td>
</tr>
<tr>
<td></td>
<td>.</td>
<td></td>
<td></td>
<td></td>
<td></td>
<td></td>
<td></td>
<td></td>
<td></td>
</tr>
<tr>
<td></td>
<td>.</td>
<td></td>
<td></td>
<td></td>
<td></td>
<td></td>
<td></td>
<td></td>
<td></td>
</tr>
</tbody>
</table>

A properly formatted SKB may have zero or more *Conditionally Calculate Media Key* Records. Bytes 4 through 19 of the Record contain Encrypted Conditional Data \((D_{e})\). If decrypted successfully, as described below, bytes 4 through 7 contain the value DEADBEEF\(_{16}\), bytes 8-9 contain the associated Column value for the Sequence Key to be used with this Record, and bytes 10-11 contain a Generation value of 0001\(_{16}\) for the first generation. Bytes 20 and higher contain Doubly Encrypted Variant Data (possibly followed by some padding bytes at the end of the Record, not shown in Table 4-4). The first ten bytes of the Doubly Encrypted Key Data correspond to Sequence Key Row 0, the next ten bytes correspond to Sequence Key Row 1, and so forth.

Upon encountering a Conditionally Calculate Variant Data Record, the device first calculates its current Media Key Variant, as follows:

\[
K_{mv} = AES-G(K_{m}, D_{v} || 041826fa7749_{16})
\]

Where \(D_{v}\) is its current Variant Data calculated from a previous Calculate Variant Data Record or Conditionally Calculate Variant Data Record.

Using its current \(K_{mv}\) value, the device calculates Conditional Data \((D_{c})\) as:
D_c = AES-128D(K_{mv}, D_{ce}).

Before continuing to process the Record, the device checks that all of the following conditions are true:

\[ [D_c]_{msb\_32} == \text{DEADBEEF}_{16} \]
and
\[ [D_c]_{79:64} == 0001_{16} \]
and
the device has a Sequence Key with associated Column value \( C_{d, i} == [D_c]_{95:80} \) for some \( i \).

If any of these conditions is false, the device ignores the rest of the Record.

Otherwise, using the value \( i \) from the condition above, \( X \) from the Nonce Record, the device’s current Variant Data \( D_v \), and \( r = R_{d, i} \), \( c = C_{d, i} \), the device calculates:

\[ D_v = [\text{AES-G}(K_{ms, i}, X \oplus f(c, r)) \oplus D_v]_{msb\_80} \oplus D_{kde, r} \]

where \( D_{kde, r} \) is the 80-bit value starting at byte offset \( r \times 10 \) within the Record’s Doubly Encrypted Key Data, \( f(c, r) \) represents the 128-bit value:

\[ f(c, r) = 0000_{16} || c || 0000_{16} || r || 0000000000000000_{16} \]

where \( c \) and \( r \) are left-padded to lengths 16 bits, by prepending zero-valued bits to each as needed. The resulting \( D_v \) becomes the current Variant Data value.

This record is always a multiple of 4 bytes.
4.2.2.5 End of Sequence Key Block Record

Table 4-5 – *End of Sequence Key Block Record Format*

<table>
<thead>
<tr>
<th>Byte</th>
<th>Bit</th>
<th>7</th>
<th>6</th>
<th>5</th>
<th>4</th>
<th>3</th>
<th>2</th>
<th>1</th>
<th>0</th>
</tr>
</thead>
<tbody>
<tr>
<td>0</td>
<td></td>
<td></td>
<td></td>
<td></td>
<td></td>
<td></td>
<td></td>
<td></td>
<td>Record Type: 02₁₀</td>
</tr>
<tr>
<td>1</td>
<td></td>
<td></td>
<td></td>
<td></td>
<td></td>
<td></td>
<td></td>
<td></td>
<td>Record Length</td>
</tr>
<tr>
<td>2</td>
<td></td>
<td></td>
<td></td>
<td></td>
<td></td>
<td></td>
<td></td>
<td></td>
<td></td>
</tr>
<tr>
<td>3</td>
<td></td>
<td></td>
<td></td>
<td></td>
<td></td>
<td></td>
<td></td>
<td></td>
<td></td>
</tr>
<tr>
<td>4</td>
<td></td>
<td></td>
<td></td>
<td></td>
<td></td>
<td></td>
<td></td>
<td></td>
<td>Signature Data</td>
</tr>
<tr>
<td>…</td>
<td></td>
<td></td>
<td></td>
<td></td>
<td></td>
<td></td>
<td></td>
<td>Length-1</td>
<td></td>
</tr>
</tbody>
</table>

A properly formatted SKB shall contain an *End of Sequence Key Block* Record. When a device encounters this Record it stops processing the SKB, using whatever Dᵥ value it has calculated up to that point as the final Dᵥ for that SKB.

The End of Sequence Key Block Record contains the AACS LA’s signature on the data in the Sequence Key Block up to, but not including, this record. Devices may ignore the signature data. However, if any device checks the signatures and determines that the signature does not verify or is omitted, it must refuse to use the Variant Data.

The length of this record is always a multiple of 4 bytes.
4.3 Calculation of the Media Key Variant from the Variant Data

When the device has finished processing the SKB, and if it has not been revoked, it will have an 80-bit valid Variant Data $D_v$. The device calculates the Media Key Variant from the Variant Data as follows:

$$K_{mv} = AES-G(K_m, D_v || 041826fa774916)$$

In addition, the low-order 10 bits of the Variant Data identify the Variant Number for the device to use in playing the content, from 0 to 1023. This number usually denotes the particular Title Key file the device should use to decrypt the content, although the meaning and use of the Variant Number is format-specific.