Marlin IPTV-ES 運用仕様
VOD 編

Document Version: 1.4
Final

Date: 25 April, 2012

本仕様の内容は予告無しに変更されることがあります。

目次

1 はじめに

“Marlin IPTV End-point Service Specification” [MIPTV]では、暗号化されたコンテンツを復号するための鍵を受信機が取得するための複数の Key Delivery 方式を規定している。Simple Key Delivery 方式は、様々なサービスへの適用が考えられるが、最も典型的なコンテンツ配信形態としては、コンテンツがストリーム伝送される VOD サービスが想定されるため、本編を VOD 編と呼ぶこととする。

1.1 本書の規定範囲

本書では、暗号を復号するための ContentKey を[MIPTV]の 4.2.1.2 項で規定される ActionID が「EXTRACT with Simple Key Delivery（01h）」の Get Permission Request で取得するコンテンツ（以下、本書では“コンテンツ”と記す）の利用に関し、[MIPTV]に対する詳細規定項目と、[MIPTV]に対する追加規定項目を規定する。本書は、“Marlin IPTV-ES/J Specific Compliance Rules VOD 編” [IPTVCRVOD]に準拠する受信機および DRM サーバに適用する。

以下に本書の規定項目を示す。

- [MIPTV]に対する詳細規定項目
  - SAC に関する規定（[MIPTV], 4.1 節 Secure Authenticated Channel (SAC) Protocol）
    - メッセージパラメータ
    - SAC タイムアウト
    - 1 つの TCP Connection を利用可能な SAC セッション
    - Response & Commit message
  - Service Protocol に関する規定（[MIPTV], 4.2 節 Marlin IPTV-ES Service Protocols over SAC に関する規定）
    - メッセージパラメータの設定
    - メッセージパラメータの検証
- [MIPTV]に対する追加規定項目
  - ネットワーク通信プロトコル（HTTP）に関する規定
    - HTTP による SAC のメッセージの伝送
    - HTTP ヘッダ

1.2 引用文書

[IPTVCRVOD]	“Marlin IPTV-ES/J Specific Compliance Rules VOD 編”, Version 1.4
[MIPTV]	“Marlin IPTV End-point Service Specification”, Version 1.0.2
[RFC2109]	HTTP State Management Mechanism
[RFC2616]	Hypertext Transfer Protocol – HTTP/1.1

1.3 用語の定義

本書で用いる用語を以下のように定義する。

用語	定義
SAC 確立	受信機と DRM サーバとの間で相互認証とセッション鍵の共有を行うこと。
SAC 終了	SAC で用いたメッセージパラメータとセッション鍵を利用できないようにし、SAC で用いた TCP Connection を切断すること。
コンテンツ	暗号を復号するための ContentKey を[MIPTV]の 4.2.1.2 項で規定される ActionID が「EXTRACT with Simple Key Delivery（01h）」の Get Permission Request で取得するコンテンツ。

本書で用いる用語と[MIPTV]の用語との対応を以下に示す。

本書	[MIPTV]
DRM サーバ	Marlin IPTV-ES Server
受信機	Marlin IPTV-ES Device

1.4 略語

本書で用いる略語を以下に示す。

略語	正式名称
MSB	Most Significant Bit

1.5 バイトオーダー

本書で定義されるプロトコルの多バイト数値のバイトオーダーは“Big Endian”である。

1.6 ビットオーダー

本書で定義されるプロトコルのビットオーダーは“MSB First”である。

2 SAC に関する規定

本章では IPTV-ES SAC の運用を規定する。

2.1 メッセージパラメータ

[MIPTV], 4.1.3 項で規定されるプロトコルのメッセージパラメータの運用を以下に示す。

- SenderID
  - 受信機は、DRM サーバの SenderID としていかなる値を受信しても、「NULL value（00h）」を受信したものとして処理する。
- SequenceNumber
  - 受信機は Request message 送信時に SequenceNumber が（2 ^ 24－3）以上になる場合には、[MIPTV], 4.1.4.10.1 項に従い、SAC を終了する。
- TransactionFlag
  - DRM サーバと受信機は、Encrypted command message の TransactionFlag を検証しない。
- Status
  - DRM サーバは、Plain command message の Status として「Certificate issuer mismatch（8005h）」を運用しない。
- SinkCertificate
  - 受信機が送信する SinkCertificate は certificate chain を含む PKIPath とする。
- SourceCertificate
  - DRM サーバが送信する SourceCertificate は certificate chain を含む PKIPath とする。

2.2 SAC タイムアウト

DRM サーバは、メッセージ送信後に 10 秒間はタイムアウトせずにメッセージ受信待ちを行う。
DRM サーバはタイムアウト後に[MIPTV], 4.1.4.10.2 項に従い SAC を終了する。

2.3 1 つの TCP Connection を利用可能な SAC セッション

1 つの TCP Connection を利用可能な SAC セッションは 1 つとする。従って、受信機と DRM サーバは、SAC を終了した時、速やかに TCP Connection を切断する。

2.4 Response & Commit message

Response & Commit message は運用しない。

3 Service Protocol に関する規定

本章では IPTV-ES Service Protocol の運用を規定する。
なお、メッセージパラメータに設定する UsageRuleReference、メッセージ送信先の DRM サーバの URI を受信機が取得する方法については、本書では規定しない。

3.1 Get Permission Protocol

[MIPTV], 4.2 節で規定される Get Permission Protocol は ContentKey 取得に用いる。本節では、メッセージパラメータの設定とメッセージパラメータの検証について規定する。

3.1.1 メッセージパラメータの設定

受信機と DRM サーバは、以下の規定に従いメッセージパラメータを設定する。

3.1.1.1 Get Permission Request parameters

受信機は、[MIPTV], 4.2.1.2 項および以下の規定に従い、Get Permission Request のメッセージパラメータを設定する。

- UsageRuleReference
  - 事前に取得した UsageRuleReference を設定する。

3.1.1.2 Get Permission Reply parameters

DRM サーバは、[MIPTV], 4.2.1.3 項、4.2.1.4 項および以下の規定に従い、Get Permission Reply のメッセージパラメータを設定する。

- Status
  - 本書 3.1.2.1 項を参照のこと。

3.1.2 メッセージパラメータの検証

受信機と DRM サーバは、メッセージ受信時に以下の規定に従い、メッセージパラメータを検証する。

3.1.2.1 Get Permission Request parameters

DRM サーバは、[MIPTV], 4.2.4.1 項および以下の規定に従い、Get Permission Request のメッセージパラメータを検証する。

- ActionID
  - ActionID が以下に示す値の場合には、検証失敗としない。
    - ✧ EXTRACT with Simple Key Delivery（01h）

3.1.2.2 Get Permission Reply parameters

受信機は、[MIPTV], 4.2.4.2 および 4.2.4.3 項の規定に従い、Get Permission Reply のメッセージパラメータを検証する。

3.2 Get Trusted Time Protocol

[MIPTV], 4.2.2 項で規定される Get Trusted Time Protocol は Datetime 取得に用いる。
受信機は、[MIPTV], 4.2.2.2 項の規定に従い、Get Trusted Time Request のメッセージパラメータを設定する。また、受信機は、[MIPTV], 4.2.4.10 項の規定に従い、Get Trusted Time Reply のメッセージパラメータを検証する。
DRM サーバは、[MIPTV], 4.2.4.9 項の規定に従い、Get Trusted Time Request のメッセージパラメータを検証する。また、DRM サーバは、[MIPTV], 4.2.2.3 項の規定に従い、Get Trusted Time Reply のメッセージパラメータを設定する。

3.3 Packed Message Protocol

[MIPTV], 4.2.3 項で規定される Packed Message は ContentKey と Datetime の同時取得に用いる。
本節では、メッセージパラメータとメッセージパラメータの検証について規定する。

3.3.1 メッセージパラメータの設定

受信機と DRM サーバは、以下の規定に従いメッセージパラメータを設定する。

3.3.1.1 Packed Message Request parameters

受信機は、[MIPTV], 4.2.3.2 項および以下の規定に従い、Packed Message Request のメッセージパラメータを設定する。

- RequestMessageBoxList
  - 表 3-1 に示す順番にメッセージを格納する。

表 3-1 RequestMessageBoxList に格納可能な RequestMessage の組み合わせ

1 番目の RequestMessage	2 番目の RequestMessage
Get Permission Request（EXTRACT with Simple Key Delivery（01h））	Get Trusted Time Request

3.3.1.2 Packed Message Reply parameters

DRM サーバは、[MIPTV], 4.2.3.3 項の規定に従い、Packed Message Reply のメッセージパラメータを設定する。

3.3.2 メッセージパラメータの検証

受信機と DRM サーバはメッセージ受信時に以下の規定に従いメッセージパラメータを検証する。

3.3.2.1 Packed Message Request parameters

DRM サーバは、[MIPTV], 4.2.4.11 項および以下の規定に従い Packed Message Request のメッセージパラメータを検証する。

- RequestMessageBoxList
  - RequestMessageBoxList に ActionID が「EXTRACT with Simple Key Delivery（01h）」の Get Permission Request の RequestMessage が 1 以上格納されている場合、かつ、RequestMessageBoxList に格納された RequestMessage の組み合わせが、表 3-1 以外の組み合わせの場合には検証失敗とし、Packed Message Reply Parameter の Status を「Message format error（8009h）」とする。

3.3.2.2 Packed Message Reply parameters

受信機は、[MIPTV], 4.2.4.12 項の規定に従い、Packed Message Reply のメッセージパラメータを検証する。

4 ネットワーク通信プロトコル（HTTP）に関する規定

本章では、SAC のメッセージの伝送に用いるネットワーク通信プロトコル（HTTP）の運用を規定する。
ネットワーク通信プロトコルは、[RFC2616]で規定される HTTP/1.1 および[RFC2109]で規定される Cookie に準拠し、以下に示す運用とする。Cookie は SAC のセッションを識別するために用いる。なお、Cookie と Set-Cookie における attribute のうち、設定および解釈を必須とするのは NAME のみとする。

4.1 HTTP による SAC のメッセージの伝送

HTTP メッセージには SAC のメッセージを 1 個のみ格納して送信する。

4.2 HTTP ヘッダ

本節では、必須となる HTTP ヘッダについて規定する。
以下に示す HTTP ヘッダ以外は実装依存であり、受信しても解釈しなくてよい。

4.2.1 サイズ

受信機と DRM サーバは、以下に示す規定に従う。
なお、本項におけるヘッダとは、HTTP ヘッダを含む HTTP メッセージにおける start-line から空行までを示す。

- ヘッダ 1 行のサイズ
  - 受信機と DRM サーバは、ヘッダ 1 行（CR＋LF を含む）のサイズの上限は 256byte とし、それを超える場合は、複数行に分割する。
  - 受信機と DRM サーバは、1 行が 256byte を超えるヘッダを含むメッセージは受信できなくてもよい。
- ヘッダ全体のサイズ
  - 受信機と DRM サーバは、ヘッダ全体のサイズが 4096byte を超えるメッセージを受信できなくてもよい。但し、プロキシにより 1Kbyte 程度のヘッダが付加されても受信できるように、5Kbyte 程度のヘッダは受信できるよう受信機の実装は考慮されるべきである。
  - DRM サーバは、送信するレスポンスヘッダ全体のサイズが 4096byte を超えないようにすべきである。但し、受信したリクエストヘッダに表 4-1 に記載されている以外のヘッダ項目が設定されている場合は、送信するレスポンスヘッダ全体のサイズが 4096byte を超えてもよい。

4.2.2 メソッド

受信機は、メソッドとして POST のみ運用する。

4.2.3 リクエストヘッダ

DRM サーバは、リクエストヘッダとして表 4-1 に記載されたもののみ解釈を必須とする。受信機は、表 4-1 の運用に従う。

表 4-1　解釈を必須とする HTTP リクエストヘッダ

ヘッダ		運用
Request	Cookie	受信機は、SAC を終了する場合に Cookie を削除する。 受信機は、Challenge message 以外を送信する場合は Set-Cookie で指定された Cookie をつけて、Challenge message 送信時には Cookie をつけない。
	Host	
General	Connection	Close のみ運用し、受信機のリクエストヘッダの設定処理において、HTTP のリクエストを送信する際に HTTP のレスポンス受信後に TCP connection を切断することが確定している場合には必ず用いる。
Entity	Content-Length	
	Content-Type	Application/octet-stream のみ運用する。

4.2.4 レスポンスヘッダ

受信機は、レスポンスヘッダとして表 4-2 に記載されたもののみ解釈を必須とする。DRM サーバは表 4-2 の運用に従う。
なお、DRM サーバは HTTP のリクエストが正常である場合はステータスコードとして「200 OK」のみを送信する。受信機は、ステータスコードとして「200 OK」以外はエラーとする。なお、HTTP がエラーの場合でも、SAC は終了しなくてよい。

表 4-2　解釈を必須とする HTTP レスポンスヘッダ

ヘッダ		運用
Response	Set-Cookie	Set-Cookie の値は SAC のセッションを識別可能な情報を格納する。 Cookie の最大数は 1 個とする。 Set-Cookie は複数行に分割しない。
General	Cache-Control	No-cache のみ運用する。
	Connection	Close のみ運用し、TCP connection 切断時には必ず用いる。
Entity	Content-Length	
	Content-Type	Application/octet-stream のみ運用する。

4.3 メッセージ処理中の受信機から HTTP のリクエストを受信した時の DRM サーバの処理

受信したメッセージ処理中に、メッセージを送信した受信機から HTTP のリクエストを受信した場合、DRM サーバは [MIPTV], 4.1 節に規定する SAC の処理は行わずに、ステータスコードが「200 OK」以外の HTTP レスポンスを送信する。

A Appendix（Informative）

A.1 SAC 処理の例

A.1.1 状態遷移

SAC 処理に関わる受信機の状態遷移を表 A-1 に、DRM サーバの状態遷移を表 A-2 に示す。

SAC を行う 1 対の受信機と DRM サーバは、状態遷移表に従って状態を遷移する。本書の規定外となる状態とイベントの組み合わせは、表中“ー”で示す。なお、本書に規定されていないが、表 A-1 に受信機の SAC タイムアウトを記述した。

表 A-1　受信機の状態遷移表

		受信機の状態						
		①SAC 開始前	②Challenge message 送信後のメッセージ受信待ち	③Response & Request message 送信後のメッセージ受信待ち		④Request message 送信後のメッセージ受信待ち		⑤Encrypted command message 送信後のメッセージ受信待ち
				送信する Request がある	送信する Request がない	送信する Request がある	送信する Request がない	
SAC 開始指示		A.1.2 項(1)の処理を実行して、②に移る	—	—	—	—	—	—
メッセージ受信イベント	Response & Challenge message 受信	—	A.1.2 項(4)の処理を実行する • メッセージ検証に成功した場合は③に移る • メッセージ検証に失敗した場合は①に移る	①に移る	①に移る	①に移る	①に移る	①に移る
	Reply message 受信	—	①に移る	A.1.2 項(8)の処理を実行する • メッセージ検証に成功した場合は④に移る • メッセージ検証に失敗した場合は①に移る	A.1.2 項(8)の処理を実行する • メッセージ検証に成功した場合は⑤に移る • メッセージ検証に失敗した場合は①に移る	A.1.2 項(8)の処理を実行する • メッセージ検証に成功した場合は④に移る • メッセージ検証に失敗した場合は①に移る	A.1.2 項(8)の処理を実行する • メッセージ検証に成功した場合は⑤に移る • メッセージ検証に失敗した場合は①に移る	①に移る
	Encrypted command message 受信	—	①に移る	①に移る	①に移る	A.1.2 項(14)の処理を実行して、①に移る	A.1.2 項(14)の処理を実行して、①に移る	A.1.2 項(14)の処理を実行して、①に移る
	Plain command message 受信	—	A.1.2 項(15)の処理を実行して、①に移る	A.1.2 項(15)の処理を実行して、①に移る	A.1.2 項(15)の処理を実行して、①に移る	①に移る	①に移る	①に移る
	[MIPTV], 4.1.4.1 項の Message header 検証に失敗（Payload Type が上記のメッセージの場合を除く）	—	①に移る	①に移る	①に移る	①に移る	①に移る	①に移る
SAC タイムアウト		—	①に移る	①に移る	①に移る	①に移る	①に移る	①に移る

表 A-2　DRM サーバの状態遷移表

		DRM サーバの状態		
		㋐SAC 開始前のメッセージ受信待ち	㋑Response & Challenge message 送信後のメッセージ受信待ち	㋒Reply message 送信後のメッセージ受信待ち
メッセージ受信イベント	Challenge message 受信	A.1.2 項(2)の処理を実行する • メッセージ検証に成功した場合は㋑に移る • メッセージ検証に失敗した場合は㋐に移る	新たな SAC として A.1.2 項(2)の処理を実行する。Response & Challenge message を送信した SAC は終了する。 • メッセージ検証に成功した場合は㋑に移る • メッセージ検証に失敗した場合は㋐に移る	新たな SAC として A.1.2 項(2)の処理を実行する。Reply message を送信した SAC は終了する。 • メッセージ検証に成功した場合は㋑に移る • メッセージ検証に失敗した場合は㋐に移る
	Response & Request message 受信	㋐に移る	A.1.2 項(6)の処理を実行する • メッセージ検証に成功した場合は㋒に移る • メッセージ検証に失敗した場合は㋐に移る	㋐に移る
	Request message 受信	㋐に移る	㋐に移る	A.1.2 項(10)の処理を実行する • メッセージ検証に成功した場合は㋒に移る • メッセージ検証に失敗した場合は㋐に移る
	Encrypted command message 受信	㋐に移る	㋐に移る	A.1.2 項(12)の処理を実行して、㋐に移る
	[MIPTV], 4.1.4.1 項の Message header 検証に失敗（Payload Type が上記のメッセージの場合を除く）	㋐に移る	㋐に移る	㋐に移る
SAC タイムアウト		—	㋐に移る	㋐に移る

430
431

A.1.2 メッセージ処理

本項では、メッセージ送受信時に行うメッセージ処理を示す。
以降で説明するメッセージ処理の基本シーケンスについて、図 A-1 と図 A-2 に示す。
図中の()つきの番号は、各メッセージの送受信処理の種類を示す。以下、各処理について説明する。

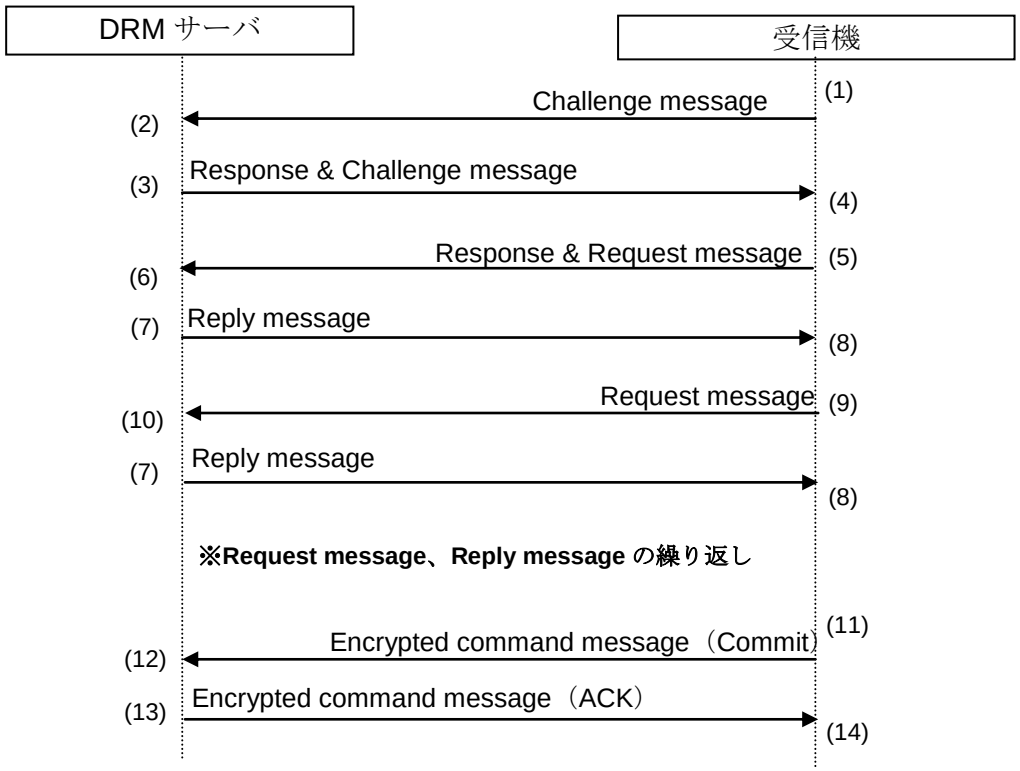


図 A-1　基本シーケンス（複数の Request を連続送信する場合）

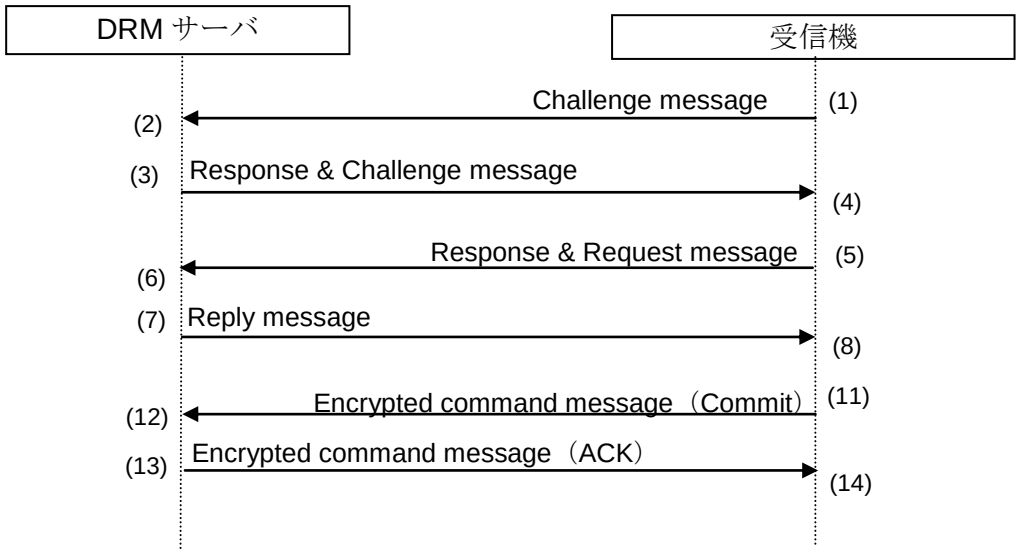


図 A-2　基本シーケンス（一つの Request のみ送信する場合）

(1) Challenge message 送信時の受信機処理

- Challenge message を作成する。
- DRM サーバに Challenge message を送信する。
- Challenge message 送信後のメッセージ受信待ちの状態に移る。

(2) Challenge message 受信時の DRM サーバ処理

- [MIPTV], 4.1.4.2 項の規定に従い、Challenge message の検証を行う。
  - 検証が成功した場合、(3)Response & Challenge message 送信時の DRM サーバ処理を実行する。
  - 検証が失敗した場合、(16)Plain command message 送信時の DRM サーバ処理を実行する。

(3) Response & Challenge message 送信時の DRM サーバ処理

- Response & Challenge message を作成する。Response & Challenge message のメッセージパラメータに関する処理を以下に示す。
  - Signature：SinkRandomNumber と SourceEC-DHPhase1Value に対して生成する。
- 受信機にResponse & Challenge messageを送信する。
- Response & Challenge message送信後のメッセージ受信待ちの状態に移る。

(4) Response & Challenge message 受信時の受信機処理

- [MIPTV], 4.1.4.3項の規定に従い、Response & Challenge messageの検証を行う。
  - 検証が成功した場合、(5)Response & Request message 送信時の受信機処理を実行する。
  - 検証が失敗した場合、SAC を終了して、SAC 開始前の状態に移る。

(5) Response & Request message 送信時の受信機処理

- セッション鍵を生成する。
- Response & Request message を作成する。SequenceNumber、TransactionFlag、Request、MessageDigest は生成したセッション鍵で暗号化する。Response & Request message のメッセージパラメータに関する処理を以下に示す。
  - Signature：SourceRandomNumber と SinkEC-DHPhase1Value に対して生成する。
  - SequenceNumber：1 を Response & Request message に設定し、保持する。
  - TransactionFlag：「even（00h）」を Response & Request message に設定し、保持する。
  - Request：Service Protocol のメッセージを設定する。
  - MessageDigest：暗号化前の MessageDigest を除く Response & Request message のパラメータから生成する。
- Response & Request message 作成後に保持している SequenceNumber を 1 増加する。
- Response & Request messageをDRMサーバに送信する。

- Response & Request message送信後のメッセージ受信待ちの状態に移る。

(6) Response & Request message 受信時の DRM サーバ処理

- [MIPTV], 4.1.4.4 項の規定に従い、Response & Request message の検証を行い、セッション鍵を生成する。
  - 検証が成功した場合、以下の処理を行い、(7)Reply message 送信時の DRM サーバ処理を実行する。
    - SequenceNumber として、[MIPTV], 4.1.4.4 項で SequenceNumber の確認に用いた値よりも 1 大きい値である 2 を保持する。
    - Response & Request message の TransactionFlag を保持する。
    - Response & Request message から Service Protocol のメッセージを抽出する。
  - 検証が失敗した場合、(16)Plain command message 送信時の DRM サーバ処理を実行する。

(7) Reply message 送信時の DRM サーバ処理

- Reply message を作成する。SequenceNumber、TransactionFlagRecordFlag、Reply、MessageDigest はセッション鍵で暗号化する。Reply message のメッセージパラメータに関する処理を以下に示す。
  - SequenceNumber：保持している SequenceNumber を用いる。
  - TransactionFlagRecordFlag：00h を設定する。
  - Reply：Service Protocol のメッセージを設定する。
  - MessageDigest：暗号化前の MessageDigest を除く Reply message のパラメータから生成する。
- Reply message 作成後に保持している SequenceNumber を 1 増加する。
- Reply messageを受信機に送信する。
- Reply message送信後のメッセージ受信待ちの状態に移る。

(8) Reply message 受信時の受信機処理

- [MIPTV], 4.1.4.6 項の規定に従い、Reply message の検証を行う。
  - 検証が成功した場合、以下の処理を行い、送信する Request がある場合は(9)の Request message 送信時の受信機処理を実行し、送信する Request がない場合は(11)Encrypted command message 送信時の受信機処理を実行する。
    - 保持している SequenceNumber を 1 増加する。
    - TransactionFlag の反転と保持を行う。現在、保持している TransactionFlag が even（00h）の場合は「odd（01h）」を、odd（01h）の場合は「even（00h）」を保持する。
    - Reply message から Service Protocol のメッセージを抽出する。
  - 検証が失敗した場合、SAC を終了して、SAC 開始前の状態に移る。

(9) Request message 送信時の受信機処理

- Request message を作成する。SequenceNumber、TransactionFlag、Request、MessageDigest はセッション鍵で暗号化する。Request message のメッセージパラメータに関する処理を以下に示す。

- ✧ SequenceNumber：保持している SequenceNumber を用いる。
- ✧ TransactionFlag：保持している TransactionFlag を用いる。
- ✧ Request：Service Protocol のメッセージを設定する。
- ✧ MessageDigest：暗号化前の MessageDigest を除く Request message のパラメータから生成する。

➢ Request message 作成後に保持している SequenceNumber を 1 増加する。
➢ Request message を DRM サーバに送信する。
➢ Request message 送信後のメッセージ受信待ちの状態に移る。

(10) Request message 受信時の DRM サーバ処理

➢ [MIPTV], 4.1.4.5項の規定に従い、Request messageの検証を行う。
- ✧ 検証が成功した場合、以下の処理を行い、(7)Reply message 送信時の DRM サーバ処理を実行する
  - ● 保持している SequenceNumber を 1 増加する。
  - ● 保持している TransactionFlag を受信した Request message の TransactionFlag に変更して保持する。
  - ● Request message から Service Protocol のメッセージを抽出する。
- ✧ 検証が失敗した場合、以下の処理を行い、(17)Command が「ERROR」の Encrypted command message 送信時の DRM サーバ処理を実行する。
  - ● 保持している SequenceNumber を 1 増加する。

(11) Encrypted command message 送信時の受信機処理

➢ Encrypted command messageを作成する。SequenceNumber、TransactionFlag、Command、Status、MessageDigestはセッション鍵で暗号化する。Encrypted command messageのメッセージパラメータに関する処理を以下に示す。
- ✧ SequenceNumber：保持している SequenceNumber を用いる。
- ✧ Command：「Commit」を設定する。
- ✧ Status：「Success（0000h）」を設定する。
- ✧ MessageDigest：暗号化前の MessageDigest を除く Encrypted command message のパラメータから生成する。

➢ Encrypted command message 作成後に保持している SequenceNumber を 1 増加する。
➢ Encrypted command messageをDRMサーバに送信する。
➢ Encrypted command message送信後のメッセージ受信待ちの状態に移る。

(12) Encrypted command message 受信時の DRM サーバ処理

➢ [MIPTV], 4.1.4.8 項の規定に従い、Encrypted command message の検証を行う。
- ✧ 検証が成功した場合、以下の処理を行い、(13)Command が「ACK」の Encrypted command message 送信時の DRM サーバ処理を実行する。
  - ● 保持している SequenceNumber を 1 増加する
- ✧ 検証が失敗した場合、以下の処理を行い、(17)Command が「ERROR」の Encrypted command message 送信時の DRM サーバ処理を実行する。

- 保持している SequenceNumber を 1 増加する

(13) Command が「ACK」の Encrypted command message 送信時の DRM サーバ処理

- Encrypted command messageを作成する。SequenceNumber、TransactionFlag、Command、Status、MessageDigestはセッション鍵で暗号化する。Encrypted command messageのメッセージパラメータに関する処理を以下に示す。
  - SequenceNumber：保持している SequenceNumber を用いる。
  - Command：「ACK」を設定する。
  - Status：「Success（0000h）」を設定する。
  - MessageDigest：暗号化前の MessageDigest を除く Encrypted command message のパラメータから生成する。
- Encrypted command messageを受信機に送信する。
- SACを終了して、SAC開始前の状態に移る。

(14) Encrypted command message 受信時の受信機処理

- [MIPTV], 4.1.4.8項の規定に従い、Encrypted command messageの検証を行う。
- SACを終了して、SAC開始前の状態に移る。

(15) Plain command message 受信時の受信機処理

- [MIPTV], 4.1.4.7 項の規定に従い、Plain command message の検証を行う。
- SAC を終了して、SAC 開始前の状態に移る。

(16) Plain command message 送信時の DRM サーバ処理

- Plain command message を作成する。Plain command message のメッセージパラメータに関する処理を以下に示す。
  - Command：「ERROR」を設定する。
  - Status：メッセージの検証で確定した Status を設定する。
- Plain command messageを受信機に送信する。
- SACを終了して、SAC開始前のメッセージ受信待ちの状態に移る。

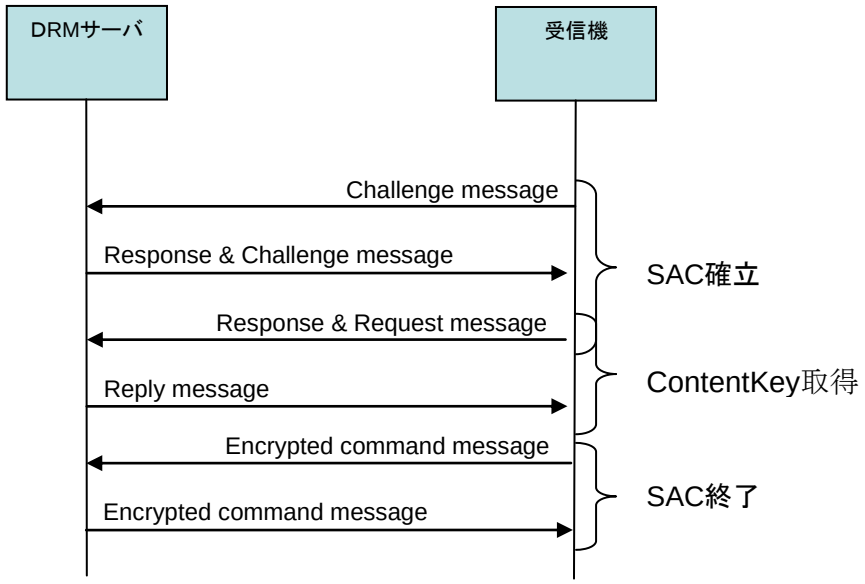
(17) Command が「ERROR」の Encrypted command message 送信時の DRM サーバ処理

- Encrypted command messageを作成する。SequenceNumber、TransactionFlag、Command、Status、MessageDigestはセッション鍵で暗号化する。Encrypted command messageのメッセージパラメータに関する処理を以下に示す。
  - SequenceNumber：保持している SequenceNumber を用いる。
  - Command：「ERROR」を設定する。
  - Status：メッセージの検証で確定した Status を設定する。
  - MessageDigest：暗号化前の MessageDigest を除く Encrypted command message のパラメータから生成する。
- Encrypted command messageを受信機に送信する。

650 ➢ SACを終了して、SAC開始前のメッセージ受信待ちの状態に移る。
651

652 ## *A.2 SAC と Service Protocol を用いたシーケンス*

653 本節では、SAC と Service Protocol を用いたシーケンスとして、ContentKey 取得を
654 解説する。
655 図 A-3 に DRM サーバ・受信機間の ContentKey 取得シーケンスを示す。
656 なお、本 ContentKey 取得シーケンスは、ContentKey で暗号化され、ユニキャスト
657 やマルチキャストで伝送されるコンテンツを、受信時に視聴するために、都度
658 ContentKey を取得する形態への適用を想定するものである。
659
660



661

図 A-3 ContentKey 取得シーケンス

662
663 ContentKey 取得シーケンスに先立ち、受信機は UsageRuleReference と DRM サー
664 バの URI を取得・保持する。なお、UsageRuleReference や DRM サーバの URI の
665 取得に関する仕様は本書では規定しない。
666
667 1. SAC 確立：[MIPTV], 4.1 節 Secure Authenticated Channel (SAC) Protocol で規
668 定されるプロトコルにより、受信機は DRM サーバとの間で相互認証を行い、
669 SAC を確立する。
670 2. ContentKey 取得： [MIPTV], 4.2.1 項 Get Permission Protocol で規定されるプ
671 ロトコルにより、受信機は ContentKey 取得要求を行い、DRM サーバから
672 ContentKey と RenderingObligation を取得する。なお、ContentKey 取得と
673 Datetime を同時に取得する場合、[MIPTV], 4.2.3 項 Packed Message Protocol
674 で規定されるプロトコルを用いる。
675 3. SAC 終了：ContentKey 取得後、[MIPTV], 4.1 節 Secure Authenticated Channel
676 (SAC) Protocol で規定されるプロトコルにより、DRM サーバは Command が
677 ACK の Encrypted command message を送信した後に、受信機は Command が
678 ACK の Encrypted command message を受信した後に、SAC を終了する。
679

以上の ContentKey 取得シーケンスにより、受信機は ContentKey と対応する RenderingObligation を取得する。
その後、受信機は、コンテンツサーバからコンテンツをストリーミングで受ける。
コンテンツは ContentKey で復号して再生される。この際、RenderingObligation に従った処理を行う必要がある。
なお、受信機は取得した ContentKey、Datetime を、取得した時点で利用可能である。
従って、SAC 終了前に、取得済みの ContentKey、Datetime を利用してもよい。
また、ContentKey の扱いに関しては、[IPTVCRVOD], 4 章の規定を参照のこと。

A.3 メッセージの例

A.3.1 HTTP のメッセージの例

本項では受信機と DRM サーバとの通信の HTTP ヘッダの例を SAC の処理毎に示す。
本項で示す例では、以下を想定している。

- DRM サーバのホスト名は www.iptv.jp
- Cookie は 0000000000000001

なお、Cookie を用いない場合は、以降に示す例の Set-Cookie および Cookie は不要である。

受信機から Challenge message を送信する場合

```
POST / HTTP/1.1
Host: www.iptv.jp
Content-type: application/octet-stream
Content- Length: 1039
```

DRM サーバから Response & Challenge message を送信する場合

```
HTTP/1.1 200 OK
Set-Cookie: JSESSIONID=0000000000000001
Cache-Control: no-cache
Content-type: application/octet-stream
Content- Length: 1321
```

受信機から Challenge message 以外のメッセージを送信する場合

```
POST / HTTP/1.1
Host: www.iptv.jp
Cookie: JSESSIONID=0000000000000001
Content-type: application/octet-stream
Content- Length: 236
```

DRM サーバから Response & Challenge message 以外のメッセージを送信する場合

```
HTTP/1.1 200 OK
Cache-Control: no-cache
Content-type: application/octet-stream
Content- Length: 124
```

A.3.2 SAC のメッセージの例

本項では SAC のメッセージの例を示す。
なお、サイズの大きいパラメータ（SinkCertificate、SourceCertificate など）と演算により生成するパラメータ（SourceEC-DHPhase1Value、Signature など）は、値

の記述を省略した。また、以下の表中のハッチング部に記載されている値は暗号化前の値を示す。

A.3.2.1 Challenge message

1001byte の SinkCertificate を格納した Challenge message の例を表 A-3 に示す。

表 A-3　Challenge message の例

Byte index	パラメータ名	値：16 進表記
0-3	ProtocolID	49505456（固定値）
4-5	ProtocolVersion	0100（固定値）
6-13	SenderID.	1001000000000000
14-15	PayloadType	0001（固定値）
16-19	PayloadSize	0000034C
20-35	SinkRandomNumber	（省略）
36-37	SinkCertificateSize	03E9
38-1038	SinkCertificate	（省略）

A.3.2.2 Response & Challenge message

1171byte の SourceCertificate を格納した Response & Challenge message の例を表 A-4 に示す。

表 A-4　Response & Challenge message の例

Byte index	パラメータ名	値：16 進表記
0-3	ProtocolID	49505456（固定値）
4-5	ProtocolVersion	0100（固定値）
6-13	SenderID	0000000000000000（固定値）
14-15	PayloadType	0002（固定値）
16-19	PayloadSize	000003BC
20-35	SourceRandomNumber	（省略）
36-91	SourceEC-DHPhase1Value	（省略）
92-147	Signature	（省略）
148-149	SourceCertificateSize	0493
150-1320	SourceCertificate	（省略）

A.3.2.3 Response & Request message

GetPermission Request を格納した Response & Request message の例を表 A-5 に示す。

表 A-5　Response & Request message の例

Byte index	パラメータ名	値：16 進表記
0-3	ProtocolID	49505456（固定値）
4-5	ProtocolVersion	0100（固定値）
6-13	SenderID	1001000000000000
14-15	PayloadType	0003（固定値）
16-19	PayloadSize	000000D8
20-75	SinkEC-DHPhase1Value	（省略）
76-131	Signature	（省略）
132-135	EncryptedDataSize	00000064
136-138	SequenceNumber	000001（固定値）
139	TransactionFlag	00（固定値）
140-203	Request	A.3.3.1.2 項を参照のこと
204-235	MessageDigest	（省略）

A.3.2.4 Request message

複数の Request を連続送信する場合の GetPermission Request を格納した最初の Request message の例を表 A-6 に示す。

表 A-6　Request message の例

Byte index	パラメータ名	値：16 進表記
0-3	ProtocolID	49505456（固定値）
4-5	ProtocolVersion	0100（固定値）
6-13	SenderID	1001000000000000
14-15	PayloadType	0004（固定値）
16-19	PayloadSize	00000068
20-23	EncryptedDataSize	00000064
24-26	SequenceNumber	000003
27	TransactionFlag	01
28-91	Request	A.3.3.1.2 項を参照のこと
92-123	MessageDigest	（省略）

A.3.2.5 Reply message

Get Permission Reply を格納した Reply message の例を表 A-7 に示す。

表 A-7　Reply message の例

Byte index	パラメータ名	値：16 進表記
0-3	ProtocolID	49505456（固定値）
4-5	ProtocolVersion	0100（固定値）
6-13	SenderID	0000000000000000（固定値）
14-15	PayloadType	0005（固定値）
16-19	PayloadSize	0000004A
20-23	EncryptedDataSize	00000046
24-26	SequenceNumber	000002
27	TransactionFlagRecordFlag	00
28-61	Reply	A.3.3.1.3 項を参照のこと
62-93	MessageDigest	（省略）

A.3.2.6 Plain command message

Status として Error other than the below を格納した Plain command message の例を表 A-8 に示す。

表 A-8　Plain command message の例

Byte index	パラメータ名	値：16 進表記
0-3	ProtocolID	49505456（固定値）
4-5	ProtocolVersion	0100（固定値）
6-13	SenderID.	0000000000000000（固定値）
14-15	PayloadType	0006（固定値）
16-19	PayloadSize	00000004（固定値）
20-21	Command	0002（固定値）
22-23	Status	8001

A.3.2.7 Encrypted command message

Status として Error other than the below を格納した複数の Request を連続送信する場合の最初の Request message に対する Encrypted command message の例を表 A-9 に示す。

表 A-9　Encrypted command message の例

Byte index	パラメータ名	値：16 進表記
0-3	ProtocolID	49505456（固定値）
4-5	ProtocolVersion	0100（固定値）
6-13	SenderID.	0000000000000000（固定値）
14-15	PayloadType	0007（固定値）
16-19	PayloadSize	0000002C（固定値）
20-23	EncryptedDataSize	00000028（固定値）
24-26	SequenceNumber	000004
27	TransactionFlag	00
28-29	Command	0002
30-31	Status	8001
32-63	MessageDigest	（省略）

A.3.3 Service Protocol のメッセージの例

本項では[MIPTV], 4.2 節で規定される Service Protocol のメッセージの例を示す。

A.3.3.1 Get Permission Protocol

本項では、[MIPTV], 4.2.1 項で規定される Get Permission Protocol のメッセージ例を示す。

A.3.3.1.1 DeviceInformation

[MIPTV], 3.2.2 項で規定される DeviceInformation の例を表 A-10 に示す。この情報は Get Permission Request メッセージに格納される。

表 A-10　DeviceInformation の例

Byte index	パラメータ名	値：16 進表記
0	Marlin IPTV-ES SpecificationVersionMajor	01（固定値）
1	Marlin IPTV-ES SpecificationVersionMinor	00（固定値）
2	Capabilities	00 （固定値）
3-4	Manufacturer	1001
5-6	ManufacturerModel	0000（固定値）
7	ManufacturerModelVersion Major	00（固定値）
8	ManufacturerModelVersion Minor	00（固定値）
9-11	Reserved	000000（固定値）

A.3.3.1.2 Get Permission Request

[MIPTV], 4.2.1.2 項で規定される Get Permission Request メッセージの例を表 A-11 に示す。

表 A-11 Get Permission Request メッセージの例

Byte index	パラメータ名	値：16 進表記
0-1	ProtocolVersion	0100（固定値）
2-3	MessageID	0001（固定値）
4-15	DeviceInformation	A.3.3.1.1 項参照
16-31	UsageRuleReference	全て 00
32	ActionID	01（固定値）
33	ActionParameter	FF（固定値）
34-35	SpecificCRID	0000（固定値）
36	PrivateDataTag	00（固定値）
37-63	PrivateData	全て 00（固定値）

A.3.3.1.3 Get Permission Reply

[MIPTV], 4.2.1.3 項で規定される Get Permission Reply メッセージの例を表 A-12 に示す。

表 A-12 Get Permission Reply メッセージの例

Byte index	パラメータ名	値：16 進表記
0-1	ProtocolVersion	0100（固定値）
2-3	MessageID	0002（固定値）
4-5	Status	0000
6-21	ContentKey	全て 00
22-23	ExtractInfoSize	0A（固定値）
24-27	NotBefore	00000000（固定値）
28-31	NotAfter	00000000（固定値）
32-33	RenderingObligation	A.3.3.1.4 項参照

A.3.3.1.4 Output Control Information

[MIPTV], 4.2.1.4.1 項で規定される RenderingObligation (Output Control Information) の例を表 A-13 に示す。

表 A-13　RenderingObligation の例

Bit index	パラメータ名	値：2 進表記
0-1	DigitalRecordingControlData	11（固定値）
2-3	CopyControlType	01（固定値）
4-5	APSControlData	01
6	ImageConstraintToken	1（固定値）
7	RetentionMode	0（固定値）
8-10	RetentionState	111（固定値）
11	EncryptionMode	1（固定値）
12-15	UserDefined	全て 0（固定値）

A.3.3.2 Get Trusted Time Protocol

本項では、[MIPTV], 4.2.2 項で規定される Get Trusted Time Protocol のメッセージ例を示す。

A.3.3.2.1 Get TrustedTime Request

[MIPTV], 4.2.2.2 項で規定される Get Trusted Time Request メッセージの例を表 A-14 に示す。

表 A-14　Get Trusted Time Request メッセージの例

Byte index	パラメータ名	値：16 進表記
0-1	ProtocolVersion	0100（固定値）
2-3	MessageID	0003（固定値）

A.3.3.2.2 Get Trusted Time Reply

[MIPTV], 4.2.2.3 で規定される Get Trusted Time Reply メッセージの例を表 A-15 に示す。

表 A-15　Get Trusted Time Reply メッセージの例

Byte index	パラメータ名	値：16 進表記
0-1	ProtocolVersion	0100（固定値）
2-3	MessageID	0004（固定値）
4-5	Status	0000
6-9	Datetime	4B3CCABD

A.3.3.3 Packed Message Protocol

本項では、[MIPTV], 4.2.3 項で規定される Packed Message Protocol のメッセージ例を示す。

A.3.3.3.1 Packed Message Request

[MIPTV], 4.2.3.2 項で規定される Packed Message Request に格納する Request が A.3.3.1.2 項と A.3.3.2.1 項の場合の例を表 A-16 に示す。

表 A-16 Packed Message Request メッセージの例

Byte index	パラメータ名	値：16 進表記
0-1	ProtocolVersion	0100（固定値）
2-3	MessageID	0101（固定値）
4-5	NumberOfRequestMessageBoxes	0002（固定値）
6-7	RequestMessageSize	0040（固定値）
8-71	RequestMessage	A.3.3.1.2 項参照
72-73	RequestMessageSize	0004（固定値）
74-77	RequestMessage	A.3.3.2.1 項参照

A.3.3.3.2 Packed Message Reply

[MIPTV], 4.2.3.3 項で規定される Packed Message Reply メッセージで A.3.3.3.1 項に対して応答する場合の例を表 A-17 に示す。

表 A-17 Packed Message Reply メッセージの例

Byte index	パラメータ名	値：16 進表記
0-1	ProtocolVersion	0100（固定値）
2-3	MessageID	0102（固定値）
4-5	Status	0000
6-7	NumberOfReplyMessageBoxes	0002（固定値）
8-9	ReplyMessageSize	0022（固定値）
10-43	ReplyMessage	A.3.3.1.3 項参照
44-45	ReplyMessageSize	000A（固定値）
46-55	ReplyMessage	A.3.3.2.2 項参照