BS デジタルテレビジョン放送 IP 再送信
Marlin IPTV-ES 運用仕様

Document Version: 1.3
Final

Date: 31 March, 2015

本仕様の内容は予告無しに変更されることがあります。

Contents

163

1. はじめに

1.1. 本書の規定範囲

本書では、BS デジタルテレビジョン放送 IP 再送信において、コンテンツの暗号を復号するための鍵を、以下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]および“BS デジタルテレビジョン放送 IP 再送信 Marlin IPTV-ES Specific Compliance Rules” [RTDBCR]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。

- “Marlin Trust Management Document for IPTV-ES Supplement 1: RTDB/J Support” [MTMDSUP] 2 章で規定する RTDB Client Key 及び RTDB Service Key を用いた“Marlin IPTV End-point Service Specification” [MIPTV], 4.1 項で規定される SAC において、[MIPTV], 4.2.1.2 項で規定される ActionID が「EXTRACT with Indirect Key Delivery（02h）」の Get Permission Request
- [MIPTV], 6.1.2 項で規定される ECM

本書は、Marlin IPTV-ES Device、Marlin IPTV-ES Server（以下、本書ではそれぞれ“受信機”、“DRM サーバ”と記す）、および、ECM を送出するサービス事業者に適用する。
本書の規定に準拠する受信機および DRM サーバが、[MIPTV]と本書以外の任意の運用仕様と組み合わせにも準拠する場合、本書での規定と相反する規定項目がある場合には、本書の規定を優先するものとする。

以下に、本書の規定項目を示す。

- [MIPTV]および[RTDBCR]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 Protocol 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
    - UsageRuleReference の設定
    - 受信機における WorkKey・WorkKeyID・PrivateData・SubscriptionTierBits・ExtractInfo の取得に関する処理
    - DRM サーバにおける WorkKey・WorkKeyID・PrivateData・SubscriptionTierBits・ExtractInfo の送信に関する処理
  - ECM に関する規定（[MIPTV], 6.1.2 項 ECM format）
    - CA_descriptor および ECM の送出
- [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

211 ✧ HTTP による SAC のメッセージの伝送
212 ✧ HTTP ヘッダ
213

214 *1.2. 引用文書*

[RTDBCR]	“BS デジタルテレビジョン放送 IP 再送信 Marlin IPTV-ES Specific Compliance Rules”, Version 1.3
[MIPTV]	“Marlin IPTV End-point Service Specification”, Version 1.0.2
[MP2S]	ISO/IEC 13818-1 “Information technology – Generic coding of moving pictures and associated audio information: Systems” Second edition 2000-12-01
[MTMDSUP]	“Marlin Trust Management Document for IPTV-ES Version 2.0 Supplement 1: RTDB/J Support”, Version 1.0
[RFC2109]	HTTP State Management Mechanism
[RFC2616]	Hypertext Transfer Protocol – HTTP/1.1

215

216 *1.3. 用語の定義*

217 本書で用いる用語を以下に定義する。
218

用語	定義
CA_descriptor	[MP2S], 2.6.16 項で規定される CA_descriptor。
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。
更新開始日時オフセット	WorkKey・WorkKeyID・PrivateData・SubscriptionTierBits・ExtractInfo を更新する運用における、更新の開始日時（更新開始日時）の NotAfter からのオフセット時間（単位は分）。更新開始日時オフセットは、PrivateData（[MIPTV], 4.2.1.5 項で規定される StatusExtension の PrivateData）で指定する。
コンテンツ	BS デジタルテレビジョン放送 IP 再送信において、暗号を復号するための鍵を[MIPTV], 4.2.1.2 項で規定される ActionID が「EXTRACT with Indirect Key Delivery（02h）」の Get Permission Request と [MIPTV], 6.1.2 項で規定される ECM で取得するコンテンツ。

219
220 本書で用いる用語と[MIPTV]の用語との対応を以下に示す。
221

本書	[MIPTV]
受信機	Marlin IPTV-ES Device
DRM サーバ	Marlin IPTV-ES Server
チャネル	Channel
ティアビット	Tier Bits

222

223 *1.4. 略語*

224 本書で用いる略語を以下に示す。
225

略語	正式名称
LSB	Least Significant Bit
MSB	Most Significant Bit

226

227 *1.5. バイトオーダー*

228 本書で規定す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
229 ある。
230

231 *1.6. ビットオーダー*

232 本書で規定するプロトコルのビットオーダーは、“MSB First”である。
233

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 および ECM に関する規定

本章では、Marlin IPTV-ES Service Protocol および ECM 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 項で規定される Get Permission Protocol は、WorkKey・WorkKeyID・PrivateData・SubscriptionTierBits・ExtractInfo の取得に用いる。以降、Get Permission Protocol で取得する一組の WorkKey・WorkKeyID・PrivateData・SubscriptionTierBits・ExtractInfo を示す場合、かぎかっこを付与して「WorkKey・SubscriptionTierBits・ExtractInfo」などと表記する。
本節では、以下の項目を規定する。

- 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
- UsageRuleReference の設定
- 受信機における「WorkKey・SubscriptionTierBits・ExtractInfo」の取得に関する処理
- DRM サーバにおける「WorkKey・SubscriptionTierBits・ExtractInfo」の送信に関する処理

なお、DRM サーバは、同時期に一対となる 2 つの WorkKey（odd）と WorkKey（even）とを常に発行する運用をおこなうこととする。よって、受信機は、「WorkKey・SubscriptionTierBits・ExtractInfo」を取得する場合、一対となる 2 つの WorkKey（odd）と WorkKey（even）とを取得することとする。
以降、2 つの Get Permission Reply で同時期に取得する同一 ServiceProviderID・同一 WorkKeyManagementID の WorkKey（odd）と WorkKey（even）の組、または、Packed Message Reply で同時に取得する同一 ServiceProviderID・同一 WorkKeyManagementID の WorkKey（odd）と WorkKey（even）の組を示す場合、“一対の”という表記を加え、“一対の WorkKey”、“一対の WorkKey（odd）と WorkKey（even）”などと表記する。

3.1.1. メッセージパラメータの設定

受信機および 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
  - 事前に取得する UsageRuleReference を設定する。UsageRuleReference の規定については、3.1.3 項を参照のこと。

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
  - 3.1.2.1 項を参照のこと。
- WorkKeyID
  - 同一 ServiceProviderID かつ同一 WorkKeyManagementID の WorkKey に関し、同時期に発行する一対の WorkKey（odd）と WorkKey（even）の WorkKeyVersion には連続した値を設定する。WorkKeyVersion が 255（FFh）と 0（00h）とである場合は、連続した値とみなす
    なお、WorkKeyID の運用例については、A.4 節を参照のこと。
- PrivateData
  - 同一 ServiceProviderID かつ同一 WorkKeyManagementID の WorkKey に関し、同時期に発行する一対の WorkKey（odd）と WorkKey（even）の PrivateData には、送信する「WorkKey・SubscriptionTierBits・ExtractInfo」の次回の更新有無および次回の更新における NotAfter から更新開始日時までのオフセット時間（更新開始日時オフセット、単位は分）として、同一の値を設定する。
    なお、更新開始日時の詳細については、3.1.4 項の式(3.1)を参照のこと。
    - 次回の更新をおこなう場合、更新開始日時オフセットの値に 0001h（1 分）～FFFFh（65535 分）を設定する。ただし、NotAfter の値が 0000FFFFh（更新開始日時オフセットの最大値）～FFFFFFFEh である場合のみ設定可である。
    - 次回の更新をおこなわない場合、更新開始日時オフセットの値に 0000h を設定する。NotAfter の値が FFFFFFFFh である場合は、更新開始日時オフセットの値に 0000h を設定する。
  - 同一 ServiceProviderID かつ同一 WorkKeyManagementID の WorkKey に関し、同時期に発行する一対の WorkKey（odd）と WorkKey（even）の SubscriptionTierBits には同一の値を設定する。
- NotBefore/NotAfter
  - 同一 ServiceProviderID かつ同一 WorkKeyManagementID の WorkKey に関し、同時期に発行する一対の WorkKey（odd）と WorkKey（even）の NotBefore には、同一の値を設定する。同様に、同時期に発行する一対の WorkKey（odd）と WorkKey（even）の NotAfter には、同一の値を設定する。

3.1.2. メッセージパラメータの検証

DRM サーバおよび受信機は、以下の規定に従い、メッセージ受信時に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。

364 ✧ EXTRACT with Indirect Key Delivery（02h）
365

366 **3.1.2.2. Get Permission Reply parameters**

367 受信機は、[MIPTV], 4.2.4.2 項、4.2.4.4 項および以下の規定に従い、Get Permission
368 Reply のメッセージパラメータを検証する。
369 受信機は、以下の検証の成功後に、一対の「WorkKey・SubscriptionTierBits・
370 ExtractInfo」を使用することとし、なお、受信機は検証に失敗した場合は、一対の
371 WorkKey として取得した 2 つの「WorkKey・SubscriptionTierBits・ExtractInfo」の
372 両方を使用しない。
373 また、受信機は保持する一対の「WorkKey・SubscriptionTierBits・ExtractInfo」と同
374 一 ServiceProviderID・同一 WorkKeyManagementID の一対の「WorkKey・
375 SubscriptionTierBits・ExtractInfo」を取得した場合、以下の検証の成功後に、
376 [MIPTV], 6.1.3 項の規定にしたがい一対の「WorkKey・SubscriptionTierBits・
377 ExtractInfo」を更新する。
378
379 ● WorkKeyID
380 ➢ 一対の WorkKey（odd）・WorkKey（even）として取得した 2 つの
381 WorkKey が、同一 ServiceProviderID かつ同一 WorkKeyManagementID の
382 一対の WorkKey（odd）と WorkKey（even）であることを検証する。
383 したがって、取得した 2 つの WorkKeyID の値が以下の条件のいずれかに該
384 当する場合、検証失敗とする。
385 ✧ ServiceProviderID・ReservedByte（WorkKeyID の上位 3 バイト目）・
386 WorkKeyManagementID のいずれかの値が異なる場合
387 ✧ WorkKeyVersion の値が連続した値でない場合。ただし、2 つの
388 WorkKeyVersion の値が 255（FFh）と 0（00h）である場合は、連続し
389 た値とみなすこと。
390 ➢ PrivateData（更新開始日時オフセット）の値が 0001h～FFFFh（更新され
391 る）である「WorkKey・SubscriptionTierBits・ExtractInfo」を更新する場合
392 において、更新前後の WorkKey（odd）・WorkKey（even）が、同一
393 ServiceProviderID かつ同一 WorkKeyManagementID の WorkKey であるこ
394 とを検証する。
395 したがって、更新前後の WorkKey（odd）または WorkKey（even）の
396 WorkKeyID の値が以下の条件に該当する場合、検証失敗とする。
397 ✧ ServiceProviderID・ReservedByte・WorkKeyManagementID のいずれ
398 かの値が異なる場合
399 ● PrivateData
400 ➢ 一対の WorkKey（odd）・WorkKey（even）として取得した 2 つの
401 WorkKey の PrivateData（更新開始日時オフセット）の値が異なる場合、検
402 証失敗とする。
403 ● SubscriptionTierBits
404 ➢ 一対の WorkKey（odd）・WorkKey（even）として取得した 2 つの
405 WorkKey の SubscriptionTierBits の値が異なる場合、検証失敗とする。
406 ● NotBefore/NotAfter
407 ➢ 一対の WorkKey（odd）・WorkKey（even）として取得した 2 つの
408 WorkKey の NotBefore の値が異なる場合、検証失敗とする。
409 同様に、2 つの WorkKey の NotAfter の値が異なる場合、検証失敗とする。
410 ➢ PrivateData（更新開始日時オフセット）の値が 0001h～FFFFh（更新され
411 る）である WorkKey に関して、NotAfter（単位は分）の値が 0000FFFFh

412 （更新開始日時オフセットの最大値）よりも小さい場合、または、
413 FFFFFFFFh（期限なし）である場合、検証失敗とする。
414 更新開始日時オフセットの詳細については、3.1.4 項の式(3.1)を参照のこと。
415

416 3.1.3. UsageRuleReference の設定

417 UsageRuleReference の設定は、表 3-1 に示す通りとする。
418 UsageRuleReference は、上位 6 バイトの値のみを規定し、下位 10 バイトの値は規
419 定しない。すなわち、UsageRuleReference の下位 10 バイトの値は、サービス事業
420 者の運用により任意の値を設定して良い。
421 UsageRuleReference の上位 6 バイトは、“ServiceProviderID”、
422 “ReservedByte”、“WorkKeyManagementID”、および“odd/evenID”から構成
423 される。UsageRuleReference の運用例については、A.4 節を参照のこと。
424 なお、表 3-1 のバイトインデックスの値は、UsageRuleReference の最上位バイトか
425 らの相対値である。
426 また、以降では、「WorkKey（odd）・SubscriptionTierBits・ExtractInfo」に対応す
427 る UsageRuleReference を“UsageRuleReference（odd）”、「WorkKey
428 （even）・SubscriptionTierBits・ExtractInfo」に対応する UsageRuleReference を
429 “UsageRuleReference（even）”と表記する。
430

表 3-1　UsageRuleReference の設定

バイトインデックス	パラメータ	パラメータの説明	パラメータ値の規定
0-1	ServiceProviderID	・[MIPTV], 4.2.1.5.1 項で規定される ServiceProviderID である。	・UsageRuleReference に対応する WorkKey の ServiceProviderID と同一の値を設定する。
2	ReservedByte	・[MIPTV], 4.2.1.5.1 項で規定される ReservedByte である。	・UsageRuleReference に対応する WorkKey の ReservedByte と同一の値（00h）を設定する。
3-4	WorkKeyManagementID	・[MIPTV], 4.2.1.5.1 項で規定される WorkKeyManagement ID である。	・UsageRuleReference に対応する WorkKey の WorkKeyManagementID と同一の値を設定する。
5	odd/evenID	・[MIPTV], 4.2.1.5.1 項で規定される WorkKeyVersion の LSB の値である。	・LSB に UsageRuleReference に対応する WorkKey の WorkKeyVersion の LSB と同一の値を設定する。 ・上位 1 ビット目から上位 7 ビット目までは 0b を設定する。

3.1.4.　受信機における「WorkKey・SubscriptionTierBits・ExtractInfo」の取得に関する処理

受信機は、「WorkKey・SubscriptionTierBits・ExtractInfo」の取得に関して、[MIPTV], 4.2.4.4 項および以下の規定に従い処理をおこなう。

- 受信機は、「WorkKey・SubscriptionTierBits・ExtractInfo」の取得時には、事前に取得する UsageRuleReference（odd）・UsageRuleReference（even）を用いて、同一 ServiceProviderID かつ同一 WorkKeyManagementID の一対の「WorkKey（odd）・SubscriptionTierBits・ExtractInfo」と「WorkKey（even）・SubscriptionTierBits・ExtractInfo」とを取得する。
- 受信機は、取得した一対の WorkKey（odd）・WorkKey（even）の更新開始日時オフセット（StatusExtension の PrivateData）の値が 0001h～FFFFh（更新される）である「WorkKey・SubscriptionTierBits・ExtractInfo」の更新の制御をおこなう場合は、以下に従う。
  - 受信機は、WorkKey（odd）または WorkKey（even）の NotAfter および更新開始日時オフセットの値を用いて、式(3.1)に従い更新開始日時（単位は分）を算出する。更新開始日時オフセットの値は、NotAfter（単位は分）から更新開始日時までのオフセット時間（単位は分）を示す。

（更新開始日時）＝（NotAfter）－（更新開始日時オフセット）･･･ (3.1)

453 ➢ 受信機は、式(3.1)により算出した更新開始日時以降、DRM サーバから一対
454 の更新された「WorkKey・SubscriptionTierBits・ExtractInfo」を取得するこ
455 とができる。更新開始日時以降、受信機は速やかに更新することが望まし
456 い。
457 ● 受信機は、取得した一対の WorkKey（odd）・WorkKey（even）の更新開始日
458 時オフセットの値が 0000h（更新されない）である場合は、当該「WorkKey・
459 SubscriptionTierBits・ExtractInfo」の更新の制御はおこなわない。
460

461 ### 3.1.5. DRM サーバにおける「WorkKey・
462 ### SubscriptionTierBits・ExtractInfo」の送信に関する処理

463 DRM サーバは、「WorkKey・SubscriptionTierBits・ExtractInfo」の送信に関して、
464 以下の規定に従い処理をおこなう。
465
466 ● DRM サーバは、同一 ServiceProviderID かつ同一 WorkKeyManagementID の
467 WorkKey に関し、同時期に一対の WorkKey を発行する運用をおこなうこと。す
468 なわち、DRM サーバは、同一 ServiceProviderID かつ同一
469 WorkKeyManagementID の WorkKey に関して、受信機から「WorkKey
470 （odd）・SubscriptionTierBits・ExtractInfo」または「WorkKey（even）・
471 SubscriptionTierBits・ExtractInfo」の要求を受信した場合、対応する
472 「WorkKey・SubscriptionTierBits・ExtractInfo」を送信する。
473 ➢ このとき DRM サーバは、WorkKey（odd）または WorkKey（even）のい
474 ずれか一方は、WorkKey の送信時点で ECM を暗号化する WorkKey を送信
475 する。
476 ● サービス事業者が、ServiceProviderID および WorkKeyManagementID で特定さ
477 れる「WorkKey・SubscriptionTierBits・ExtractInfo」を更新する運用をおこなう
478 場合、DRM サーバは、当該「WorkKey・SubscriptionTierBits・ExtractInfo」の
479 受信機ごとの更新開始日時以降に当該受信機からの要求を受信した場合、少な
480 くとも NotAfter の値を更新した「WorkKey・SubscriptionTierBits・
481 ExtractInfo」を送信する。
482 ➢ DRM サーバは、3.1.4 項の式(3.1)に従い、受信機ごとの更新開始日時を算
483 出する。式(3.1)の NotAfter は、当該「WorkKey・SubscriptionTierBits・
484 ExtractInfo」に関して、当該受信機に対して最後に送信した WorkKey の
485 NotAfter の値を用いる。
486

487 ## *3.2. Get Trusted Time Protocol*

488 [MIPTV], 4.2.2 項で規定される Get Trusted Time Protocol は、Datetime の取得に用
489 いる。
490 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
491 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
492 Trusted Time Reply のメッセージパラメータを検証する。
493 DRM サーバは、[MIPTV], 4.2.4.9 項の規定に従い、Get Trusted Time Request のメ
494 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
495 に従い、Get Trusted Time Reply のメッセージパラメータを設定する。
496

3.3. Packed Message Protocol

[MIPTV], 4.2.3 項で規定される Packed Message Protocol は、以下のパラメータを同時に取得する場合に用いる。

- 1 または複数の“一対の「WorkKey（odd）・SubscriptionTierBits・ExtractInfo」と「WorkKey（even）・SubscriptionTierBits・ExtractInfo」”
- 1 または複数の“一対の「WorkKey（odd）・SubscriptionTierBits・ExtractInfo」と「WorkKey（even）・SubscriptionTierBits・ExtractInfo」”および Datetime

本節では、以下の項目を規定する。

- 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

3.3.1. メッセージパラメータの設定

受信機および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
  - RequestMessageBoxList には、表 3-2 に示す順番で RequestMessage を格納する。
    - ✧ RequestMessage の個数は（2×N）個または（2×N＋1）個とする。ここで、N は 16 以下の自然数である。
      （2×M－1）番目と（2×M）番目の RequestMessage には、同一 ServiceProviderID・同一 WorkKeyManagementID の「WorkKey・SubscriptionTierBits・ExtractInfo」を取得するための Get Permission Request（UsageRuleReference の ServiceProviderID・WorkKeyManagementID が同一の値）を格納する。ここで、M は N 以下の自然数である。
      なお、以降、N または M を用いた RequestMessage の個数および順番の表記については「×」を省略し、（2×N）を（2N）などと記す。
    - ✧ （2M－1）番目の RequestMessage には「WorkKey（odd）・SubscriptionTierBits・ExtractInfo」を、（2M）番目の RequestMessage には「WorkKey（even）・SubscriptionTierBits・ExtractInfo」を取得するための Get Permission Request を格納する。
    - ✧ RequestMessage の個数が（2N＋1）個の場合、（2N＋1）番目の RequestMessage には Get Trusted Time Request を格納する。

表 3-2 RequestMessageBoxList に格納可能な RequestMessage の組み合わせ

Request Message	1 番目の Request	2 番目の Request	…	（2M－1）番目 *5 の	（2M）番目 *5 の Request	…	（2N＋1）番目 *5 の

の個数	Message	Message		Request Message *6	Message *6		Request Message
2N *5	Get Permission Request *1 ActionID：EXTRACT with Indirect Key Delivery（02h） UsageRuleReference：UsageRuleReference（odd）*3	Get Permission Request *1 ActionID：EXTRACT with Indirect Key Delivery（02h） UsageRuleReference：UsageRuleReference（even）*4	…	Get Permission Request *1 ActionID：EXTRACT with Indirect Key Delivery（02h） UsageRuleReference：UsageRuleReference（odd）*3	Get Permission Request *1 ActionID：EXTRACT with Indirect Key Delivery（02h） UsageRuleReference：UsageRuleReference（even）*4	…	
2N+1 *5	Get Permission Request *1 ActionID：EXTRACT with Indirect Key Delivery（02h） UsageRuleReference：UsageRuleReference（odd）*3	Get Permission Request *1 ActionID：EXTRACT with Indirect Key Delivery（02h） UsageRuleReference：UsageRuleReference（even）*4	…	Get Permission Request *1 ActionID：EXTRACT with Indirect Key Delivery（02h） UsageRuleReference：UsageRuleReference（odd）*3	Get Permission Request *1 ActionID：EXTRACT with Indirect Key Delivery（02h） UsageRuleReference：UsageRuleReference（even）*4	…	Get Trusted Time Request *2

*1：メッセージパラメータの設定については、3.1.1 項を参照のこと。
*2：メッセージパラメータの設定については、3.2 項を参照のこと。
*3：「WorkKey（odd）・SubscriptionTierBits・ExtractInfo」に対応する UsageRuleReference（odd/evenID の値が 01h）を示す。詳細は 3.1.3 項を参照のこと。
*4：「WorkKey（even）・SubscriptionTierBits・ExtractInfo」に対応する UsageRuleReference（odd/evenID の値が 00h）を示す。詳細は 3.1.3 項を参照のこと。
*5： N≦16、M≦N（N、M はともに自然数）とする。
*6：（2M－1）番目と（2M）番目の RequestMessage の UsageRuleReference は、同一 ServiceProviderID・同一 WorkKeyManagementID とする。

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

DRM サーバおよび受信機は、以下の規定に従い、メッセージ受信時に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 Indirect Key Delivery（02h）」の Get Permission Request の RequestMessage が 1 以上格納されている場合、かつ、RequestMessage の組み合わせが表 3-2 の組合せ以外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 項および以下の規定に従い、Packed Message Reply のメッセージパラメータを検証する。

- ReplyMessageBoxList
  - Status が「Success（0000h）」の場合には、3.1.2.2 項にしたがい、ReplyMessageBoxList に格納された ReplyMessage を検証する。ReplyMessageBoxList に格納されたいずれかの ReplyMessage の検証に失敗した場合には、Packed Message Reply 全体を検証失敗とする。

3.4. ECM に関する規定

本節では、[MIPTV], 6.1.2 項で規定される ECM format に関する運用を規定する。

- CA_descriptor および ECM の送出

3.4.1. CA_descriptor および ECM の送出

3.4.1.1. CA_descriptor の送出

- CA_descriptor の descriptor_tag の値は、09h とする。
- CA_descriptor の CA_system_ID の値は 0x000Dh とする。

3.4.1.2. ECM の送出

- ECM の更新
  - ECM の更新間隔は、6 秒以上とする。
- ECM の再送
  - ECM の再送間隔は、最小 100ms・最大 1000ms とする。推奨値は 100ms とする。

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	―		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			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	―	①に移る	①に移る	①に移る	①に移る	①に移る	①に移る

		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	
	場合を除く）							
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

703

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 0 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 0 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

704

705 **A.1.2　メッセージ処理**

706 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。

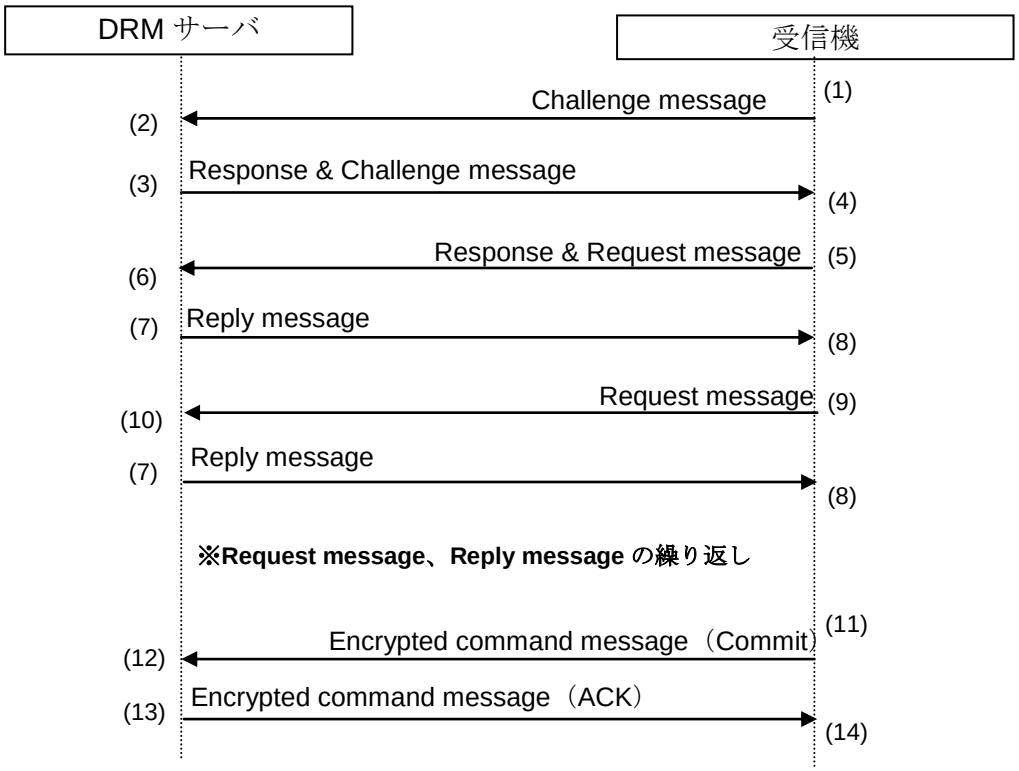

図 A-1　基本シーケンス（複数の Request を連続送信する場合）

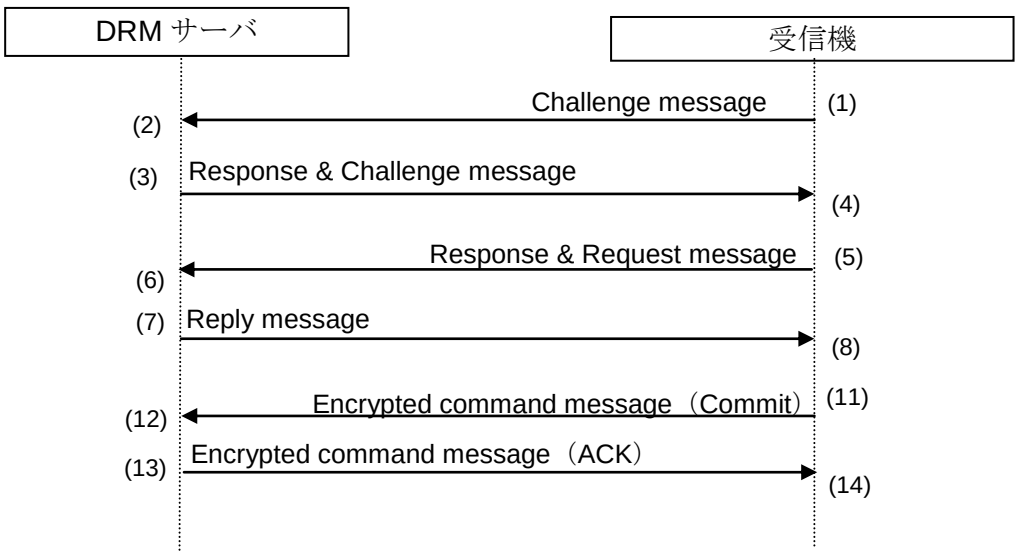

図 A-2　基本シーケンス（一つの Request のみ送信する場合）

(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。
- 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。

A.2 SAC と Service Protocol を用いたシーケンス

本節では、SAC と Service Protocol を用いたシーケンスとして、「WorkKey・SubscriptionTierBits・ExtractInfo」取得を解説する。
図 A-3 に DRM サーバ・受信機間の「WorkKey・SubscriptionTierBits・ExtractInfo」取得シーケンスを示す。

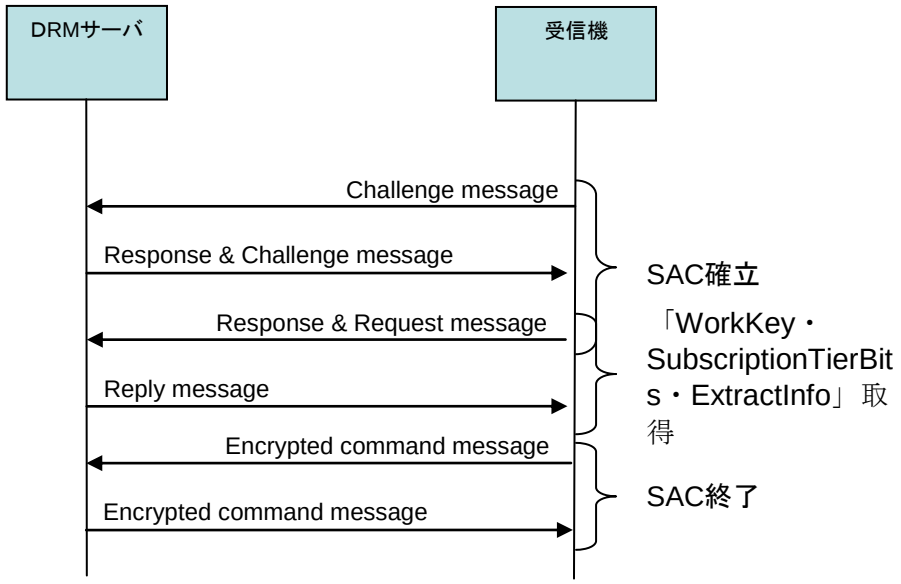

図 A-3 「WorkKey・SubscriptionTierBits・ExtractInfo」取得シーケンス

「WorkKey・SubscriptionTierBits・ExtractInfo」取得シーケンスに先立ち、受信機は UsageRuleReference と DRM サーバの URI を取得・保持する。なお、UsageRuleReference や DRM サーバの URI の取得に関する仕様は本書では規定しない。

1. SAC 確立：[MIPTV], 4.1 節 Secure Authenticated Channel (SAC) Protocol で規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、SAC を確立する。
2. 「WorkKey・SubscriptionTierBits・ExtractInfo」取得： [MIPTV], 4.2.1 項 Get Permission Protocol で規定されるプロトコルにより、受信機は「WorkKey・SubscriptionTierBits・ExtractInfo」取得要求を行い、DRM サーバから「WorkKey・SubscriptionTierBits・ExtractInfo」を取得する。なお、「WorkKey・SubscriptionTierBits・ExtractInfo」取得と Datetime を同時に取得する場合、[MIPTV], 4.2.3 項 Packed Message Protocol で規定されるプロトコルを用いる。
3. SAC 終了：「WorkKey・SubscriptionTierBits・ExtractInfo」取得後、[MIPTV], 4.1 節 Secure Authenticated Channel (SAC) Protocol で規定されるプロトコルにより、DRM サーバは Command が ACK の Encrypted command message を送信した後に、受信機は Command が ACK の Encrypted command message を受信した後に、SAC を終了する。

以上の「WorkKey・SubscriptionTierBits・ExtractInfo」取得シーケンスにより、受信機は「WorkKey・SubscriptionTierBits・ExtractInfo」を取得する。

その後、受信機は、コンテンツサーバから BS デジタルテレビジョン放送 IP 再送信コンテンツをストリーミングで受ける。コンテンツに多重化された ECM は WorkKey で復号され、ECM に含まれるスクランブル鍵を用いてコンテンツは復号されて再生される。この際、ECM に含まれる RenderingObligation に従った処理を行う必要がある。
また、WorkKey の扱いに関しては、[RTDBCR], 4 章の規定を参照のこと。

A.3 コンテンツの利用シーケンス

本節では、コンテンツの利用に関して、受信機と DRM サーバとの間、および受信機とコンテンツを配信するコンテンツサーバとの間のシーケンスの概要について説明する。

コンテンツの利用シーケンスは、以下に示す 3 つの処理により構成される。

① 「WorkKey・SubscriptionTierBits・ExtractInfo」の取得処理
② コンテンツ受信時の ECM 処理
③ 「WorkKey・SubscriptionTierBits・ExtractInfo」の更新処理

なお、以下の処理の事前に契約処理が行われ、受信機が「WorkKey・SubscriptionTierBits・ExtractInfo」の取得のための UsageRuleReference・DRM サーバの URI を取得済みであることを前提とする。

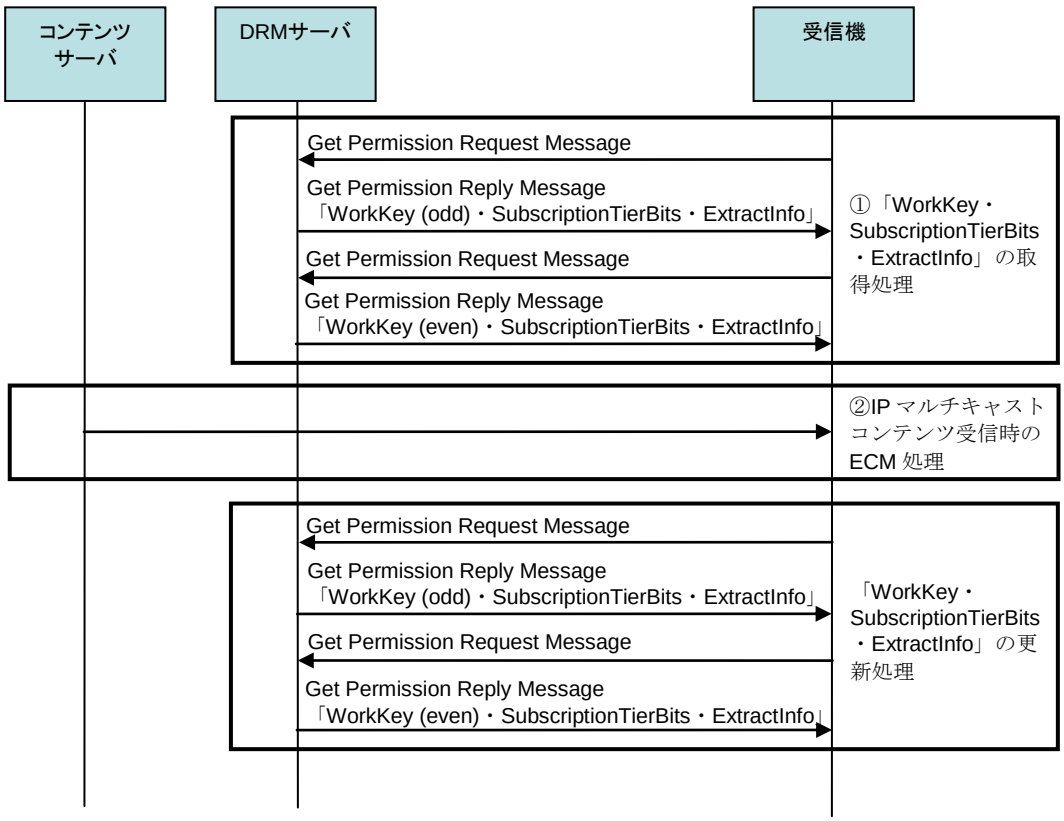

図 A-4 コンテンツの利用シーケンス

A.3.1 「WorkKey・SubscriptionTierBits・ExtractInfo」の取得処理

受信機は、[MIPTV], 4.1 節で規定される Secure Authenticated Channel (SAC) Protocol により SAC を確立した上で、[MIPTV], 4.2.1 項で規定される Get Permission Protocol により、DRM サーバから「WorkKey・SubscriptionTierBits・ExtractInfo」を取得する。
なお、Get Permission Protocol によるメッセージシーケンスは、A.2 節の「WorkKey・SubscriptionTierBits・ExtractInfo」取得シーケンスと同様である。

A.3.1.1ティアビット

「WorkKey・SubscriptionTierBits・ExtractInfo」に関して、DRM サーバと受信機との間の通信回数・受信機での保持数の削減のため、サービス事業者の 1 以上のチャネルに対して同一の WorkKey を適用する運用が想定される。そこで、受信機が有する契約に応じたチャネルの視聴制御のため、ティアビットとよばれる 64 ビットのビット列を用いる。ティアビットの各ビットには、各チャネルに対する契約が対応づけられる。この対応づけは、サービス事業者の運用依存である。
DRM サーバは、WorkKey とともに、ティアビットのうち受信機が有する契約に対応するビットを示す“SubscriptionTierBits”を受信機に送信する。一方、ECM には、ティアビットのうち当該チャネルに対する契約を示す“ChannelTierBits”が設定される。受信機は、SubscriptionTierBits と ChannelTierBits との照合により、受信機が契約を有するチャネルのみを視聴可とするよう制御する。受信機でのティアビットによる視聴制御については A.3.2 項を参照のこと。

A.3.1.2WorkKeyManagementID

A.3.1.1 項で説明したティアビットを用いる運用を実現するため、サービス事業者は、複数チャネルに適用する WorkKey と、当該複数チャネルに対する契約を対応づけたティアビットとの対応を管理する。この WorkKey およびティアビットの管理単位を特定する識別子を“WorkKeyManagementID”とよぶ。WorkKeyManagementID は、サービス事業者ごと（“ServiceProviderID”で識別される）に割り当てる識別子である。
受信機は、ServiceProviderID および WorkKeyManagementID の単位で、DRM サーバから「WorkKey・SubscriptionTierBits・ExtractInfo」を取得する。したがって、受信機が取得時に指定する UsageRuleReference には、ServiceProviderID と WorkKeyManagementID とが含まれる。なお、WorkKeyManagementID および UsageRuleReference の運用例については、A.4 節を参照のこと。

A.3.1.3WorkKey（odd/even）

コンテンツに対して、一定期間などで ECM を暗号化する WorkKey を更新する運用が想定される。
このような運用において、WorkKey の更新時でも受信機が ECM を連続して復号し、継続視聴が可能となるように、DRM サーバは同時期に一対となる 2 つの WorkKey を発行する。この一対となる 2 つの WorkKey を、“WorkKey（odd）”および“WorkKey（even）”とよぶ。WorkKey（odd）と WorkKey（even）は、WorkKeyID に含まれる WorkKeyVersion の LSB の値（odd/evenID）により識別できる。WorkKeyVersion の詳細については[MIPTV], 4.2.1.5.1 項を、運用例については A.4 節を参照のこと。

また、DRM サーバは、一対の WorkKey（odd）・WorkKey（even）として、WorkKey の送出時点で ECM を暗号化する WorkKey と、ECM を暗号化する WorkKey の次回更新後の WorkKey とを送信する。
受信機は、2 つの Get Permission Protocol を用いて、DRM サーバから odd/even の「WorkKey・SubscriptionTierBits・ExtractInfo」を個別に取得する。したがって、DRM サーバが、odd/even のどちらの WorkKey を発行すればよいかを特定できるように、UsageRuleReference には要求する WorkKey の odd/evenID の値が設定される。

A.3.1.4受信機における「WorkKey・SubscriptionTierBits・ExtractInfo」の管理

受信機は、ServiceProviderID・WorkKeyManagementID の単位で、取得した一対の「WorkKey・SubscriptionTierBits・ExtractInfo」を管理する。受信機は「WorkKey・SubscriptionTierBits・ExtractInfo」として、少なくとも WorkKey、WorkKeyID、更新開始日時オフセット（PrivateData）、SubscriptionTierBits、NotBefore/NotAfter を管理する。
[MIPTV], 4.2.4.4 項で規定される通り、受信機は、保持する WorkKey と“ServiceProviderID”、“ReservedByte”、“WorkKeyManagementID”、および“ odd/evenID（WorkKeyVersion の LSB）”の 4 つの値が一致する WorkKey を新たに取得した場合、取得した「WorkKey・SubscriptionTierBits・ExtractInfo」で、保持する「WorkKey・SubscriptionTierBits・ExtractInfo」を更新する。このとき、受信機は、ServiceProviderID・WorkKeyManagementID の単位での「WorkKey・SubscriptionTierBits・ExtractInfo」の管理にあわせて、一対の「WorkKey・SubscriptionTierBits・ExtractInfo」の単位で更新する。ただし、更新前後で値が同一であるパラメータについては更新しなくても良い。
なお、取得した「WorkKey・SubscriptionTierBits・ExtractInfo」を不揮発性記憶領域に記録するか否かについては、受信機の実装依存である。ただし、受信機は、揮発性記憶領域に保持する「WorkKey・SubscriptionTierBits・ExtractInfo」の電源断などによる消失を防止し、DRM サーバからの再取得を不要とするため、不揮発性記憶領域に記録することが望ましい。

A.3.1.5RenderingObligation による EXTRACT・RECORD・EXPORT

受信機は、[MIPTV], 6.1.2 項で規定される ECM の RenderingObligation に基づき、コンテンツの EXTRACT、RECORD、および EXPORT をおこなう。受信機は、[MIPTV], 4.2.1 項で規定される RecordInfo および ExportInfo は取得しない。
RenderingObligation の送出に関する遵守規則、および、RenderingObligation に基づく受信機における EXTRACT・RECORD・EXPORT に関する遵守規則については、[RTDBCR], 2 章を参照のこと。

A.3.1.6Packed Message Protocol による「WorkKey・SubscriptionTierBits・ExtractInfo」の取得

受信機は、[MIPTV], 4.2.3 項で規定される Packed Message Protocol を用いて、1 または複数の“一対の「WorkKey（odd）・SubscriptionTierBits・ExtractInfo」と「WorkKey（even）・SubscriptionTierBits・ExtractInfo」”とを同時に取得することができる。

また、受信機は、1 または複数の“一対の「WorkKey（odd）・SubscriptionTierBits・ExtractInfo」と「WorkKey（even）・SubscriptionTierBits・ExtractInfo」”とともに、Datetime を同時に取得することもできる。

A.3.2 コンテンツ受信時の ECM 処理

受信機は、コンテンツサーバからコンテンツを受信し、ECM を抽出する。受信機は、ECM 毎に ECM の WorkKeyID で指定される WorkKey の NotBefore/NotAfter と [RTDBCR], 3.2 節の受信機内で保持する時刻（以下、本項では“受信機内で保持する時刻”と記す）とを比較し、（NotBefore）≦（受信機内で保持する時刻）≦（NotAfter）を満たす場合、受信機は当該 WorkKey を用いて ECM を復号する。但し、NotBefore の値が FFFFFFFFh の場合は、NotBefore と受信機内で保持する時刻との比較は不要である。同様に NotAfter の値が FFFFFFFFh の場合は、NotAfter と受信機内で保持する時刻との比較は不要である。
復号した ECM の検証後、[MIPTV], 6.1.3 項の規定の通り、受信機は ECM の ChannelTierBits と WorkKey の SubscriptionTierBits とを照合し、同一位置のビットが共に 1b となるビットが存在する場合に視聴可と判定する。
受信機は、視聴可と判定した場合、ECM から抽出した ScrambleKey によりコンテンツを復号し、ECM に設定された RenderingObligation に従った利用をおこなう。

A.3.3 「WorkKey・SubscriptionTierBits・ExtractInfo」の更新処理

受信機は、「WorkKey・SubscriptionTierBits・ExtractInfo」が更新される運用の場合に、保持する「WorkKey・SubscriptionTierBits・ExtractInfo」の更新を制御する。

A.3.3.1 更新の有無と更新開始日時オフセット

DRM サーバは、送信する「WorkKey・SubscriptionTierBits・ExtractInfo」の StatusExtension の PrivateData（上位 23 バイト目から上位 24 バイト目）の値により、次回の更新の有無、および、NotAfter から次回の更新が可能となる日時（更新開始日時）までのオフセット時間を受信機に通知する。この 2 バイトの値を“更新開始日時オフセット”とよぶ。
DRM サーバは、次回の更新をおこなう場合には、送信する「WorkKey・SubscriptionTierBits・ExtractInfo」の更新開始日時オフセットの値に 0001h～FFFFh（単位は“分”）を設定する。
一方、DRM サーバは、契約の解約時など、次回の更新をおこなわない場合には、更新開始日時オフセットの値に 0000h を設定する。

A.3.3.2 「WorkKey・SubscriptionTierBits・ExtractInfo」の更新

「WorkKey・SubscriptionTierBits・ExtractInfo」が更新される運用（更新開始日時オフセットの値が 0001h～FFFFh）の場合、受信機は更新開始日時以降に更新することができる。更新のための「WorkKey・SubscriptionTierBits・ExtractInfo」の取得は、A.3.1 項と同等である。
一方、DRM サーバは、更新開始日時以降に受信機から「WorkKey・SubscriptionTierBits・ExtractInfo」の取得要求を受信した場合、少なくとも NotAfter の値を更新した「WorkKey・SubscriptionTierBits・ExtractInfo」を送信する。

「WorkKey・SubscriptionTierBits・ExtractInfo」の更新に関して、DRM サーバ・受信機の実装は以下の事項が考慮されることが望ましい。

- DRM サーバ
  - 受信機が短期間に繰り返し更新しないように、DRM サーバが更新後の NotAfter に設定する値は、（「WorkKey・SubscriptionTierBits・ExtractInfo」の送信日時）＋（更新開始日時オフセット）よりも大きな値とすることが望ましい。
- 受信機
  - NotAfter と更新開始日時オフセットとに基づいて更新するか否かについては、受信機の実装依存であるが、更新された場合に継続して視聴できるようにするため、受信機は更新開始日時以降、速やかに更新することが望ましい。
  - DRM サーバと受信機とで保持する時刻に誤差が生じている場合などは、DRM サーバが NotAfter を更新する前に、受信機が更新のために「WorkKey・SubscriptionTierBits・ExtractInfo」を取得してしまう場合が考えられる。このような場合、受信機は短期間に繰り返し更新しないように、間隔をあけて再度更新するなどの制御をおこなうことが望ましい。

A.4 WorkKeyID および UsageRuleReference の運用例

本節では、WorkKeyID および UsageRuleReference の運用例を示す。

A.4.1 ティアビット・WorkKeyID・UsageRuleReference の関係

サービス事業者が運用するティアビット、WorkKeyID、および UsageRuleReference の関係について説明する。

A.4.1.1WorkKeyID とティアビットとの関係

WorkKeyID は、“ServiceProviderID”、“ReservedByte”、“WorkKeyManagementID”、および“WorkKeyVersion”から構成される。WorkKeyID のうち、DRM サーバおよび受信機における WorkKey の管理単位となる ID が、ServiceProviderID および WorkKeyManagementID である。サービス事業者は、64 ビットからなるティアビットを管理し、当該ティアビットに対して WorkKeyManagementID を対応づける。したがって、サービス事業者は、64 ビットからなるティアビットを運用する個数分だけ、WorkKeyManagementID を管理する。たとえば、サービス事業者が 64 ビットからなるティアビットを 1 個用いる運用をおこなう場合、当該サービス事業者は 1 つの WorkKeyManagementID を管理する。また、64 ビットからなるティアビットを 2 個用いて、64 を超える契約を当該 2 個のティアビットに対応づける運用をおこなう場合、サービス事業者は 2 つの WorkKeyManagementID を管理する。

また、DRM サーバは、同一 ServiceProviderID・同一 WorkKeyManagementID の WorkKey について、WorkKey 自体を更新する場合に限り、WorkKeyVersion を更新する。すなわち、SubscriptionTierBits または ExtractInfo の値を更新する場合であっても、WorkKey 自体を更新しない限り、WorkKeyVersion は更新しない。

WorkKeyID の概要については A.3.1.2 項および A.3.1.3 項を、詳細については [MIPTV], 4.2.1.5.1 項を参照のこと。

A.4.1.2UsageRuleReference と WorkKeyID との関係

UsageRuleReference は、“ServiceProviderID”、“ReservedByte”、“WorkKeyManagementID”、および“odd/evenID”から構成される。
ServiceProviderID・WorkKeyManagementID には、UsageRuleReference に対応する WorkKey の ServiceProviderID・WorkKeyManagementID と同一の値が設定される。
また、odd/evenID には、UsageRuleReference に対応する WorkKey の WorkKeyVersion の LSB と同一の値が設定される。

A.4.1.3ティアビットと WorkKeyID・UsageRuleReference の値との関係の例

サービス事業者が運用するティアビットと、WorkKeyID・UsageRuleReference の値との関係の例を、表 A-3 に示す。
ただし、WorkKeyID および UsageRuleReference の ReservedByte は固定値であるので、表 A-3 では記載を省略する。
また、3.1.3 項で規定する通り、UsageRuleReference の下位 10 バイトの値はサービス事業者の運用依存であるため、表 A-3 では上位 6 バイトの値のみを記載する。

表 A-3　サービス事業者が運用するティアビットと WorkKeyID・UsageRuleReference の値との関係の例

サービス事業者	サービス事業者が運用するティアビット		WorkKeyID の値：16 進表記			UsageRuleReference の値：16 進表記		
			ServiceProviderID	WorkKeyManagementID	WorkKeyVersion	ServiceProviderID	WorkKeyManagementID	odd/evenID
サービス事業者 1	1 個 (*1)	一つ目のティアビット	0001	0001	01, 03, …(odd)	0001	0001	01(odd)
					02, 04, …(even)			00(even)
サービス事業者 2	2 個 (*2)	一つ目のティアビット	0002	0001	01, 03, …(odd)	0002	0001	01(odd)
					02, 04, …(even)			00(even)
		二つ目のティアビット		0002	01, 03, …(odd)		0002	01(odd)
					02, 04, …(even)			00(even)

*1：64 ビットからなるティアビットを 1 個用いる運用をおこなうサービス事業者の例である。
*2：64 ビットからなるティアビットを 2 個用いて、64 を超える契約を当該 2 個のティアビットに対応づける運用をおこなうサービス事業者の例である。

A.4.2 WorkKey を更新する運用における WorkKeyID と UsageRuleReference との関係の例

DRM サーバは、ServiceProviderID・WorkKeyManagementID ごとに WorkKey のバージョン（WorkKeyVersion）を管理する。DRM サーバは、受信機から WorkKey を要求された場合、一対の WorkKey（odd）・WorkKey（even）として、WorkKey の送信時点で ECM を暗号化する WorkKey と、ECM を暗号化する WorkKey の次回更新後の WorkKey とを送信する。よって、DRM サーバは、ServiceProviderID・WorkKeyManagementID ごとに少なくとも一対の WorkKeyVersion を管理する。

WorkKey を更新する運用において、ECM を暗号化する WorkKey の WorkKeyID の値、Get Permission Protocol で DRM サーバから送信する WorkKey の WorkKeyID

の値、Get Permission Protocol で WorkKey を要求する際の UsageRuleReference の値の関係の例を、表 A-4 に示す。

表 A-4 において、Get Permission Protocol で送信する WorkKey の WorkKeyVersion は、ECM を暗号化する WorkKey の更新以降に送信する値を示すものである。DRM サーバは、ECM を暗号化する WorkKey の更新後、Get Permission Protocol で送信する一対の WorkKey（odd）・WorkKey（even）も速やかに更新する。

なお、A.4.1.3 項の表 A-3 と同様に、表 A-4 において WorkKeyID および UsageRuleReference の ReservedByte は記載を省略する。また、UsageRuleReference の下位 10 バイトの値についても、同様に記載を省略する。

表 A-4　WorkKey を更新する運用における WorkKeyID と UsageRuleReference との値の関係の例

ECM を暗号化する WorkKey の更新 (*1)	ECM を暗号化する WorkKey の WorkKeyID の値：16 進表記			Get Permission Protocol で送信する WorkKey の WorkKeyID の値：16 進表記			UsageRuleReference の値：16 進表記		
	ServiceProviderID	WorkKeyManagementID	WorkKeyVersion	ServiceProviderID	WorkKeyManagementID	WorkKeyVersion	ServiceProviderID	WorkKeyManagementID	odd/even ID
運用開始	0001	0001	01(odd)	0001	0001	01(odd)	0001	0001	01(odd)
						02(even)			00(even)
1 回目の更新後			02(even)			03(odd)			01(odd)
						02(even)			00(even)
2 回目の更新後			03(odd)			03(odd)			01(odd)
						04(even)			00(even)
・・・			・・・			・・・			・・・
254 回目の更新後			FF(odd)			FF(odd)			01(odd)
						00(even)			00(even)
255 回目の更新後			00(even)			01(odd)			01(odd)
						00(even)			00(even)

*1：256 回目以降の更新では、表 A-4 における“運用開始～255 回目の更新後”の繰り返しとなる。

A.5 「WorkKey・SubscriptionTierBits・ExtractInfo」の更新運用の例

本節では、図 A-5 を用いて「WorkKey・SubscriptionTierBits・ExtractInfo」の更新運用の例を示す。

以下、図 A-5 のイベントの囲い数字、および、DRM サーバ・事業者サーバの括弧書きの数字に対応して、新規契約から解約までの DRM サーバ・事業者サーバ・受信機の処理の概要について説明する。なお、事業者サーバは、契約・解約などをおこなうサービス事業者のポータルサーバなどである。

本節では、以下を前提とした更新運用の例を示す。

- 特定の ServiceProviderID・WorkKeyManagementID の「WorkKey・SubscriptionTierBits・ExtractInfo」に関して、特定の受信機についての例を示す。
- WorkKey の期限（NotBefore/NotAfter）を定期的に更新する、期限延長の運用を基本とする。したがって、更新の度に「WorkKey・SubscriptionTierBits・ExtractInfo」の全てが更新されるとは限らないことに留意されたい。
- 期限の更新周期は 1 年とする。DRM サーバは、受信機ごとの契約月の 1 年後に更新をおこなうため、NotAfter を契約月の 1 年後の月末日に設定する。なお、NotBefore には期限なし（FFFFFFFFh）を設定する。
  図 A-5 に示す NotAfter の年月日の表記は、受信機が取得する WorkKey の NotAfter の値を示し、点線の矢印は、受信機が当該 NotAfter の値を保持する期間を示す。
- 期限延長における更新開始日時は NotAfter の 14 日前とし、更新開始日時オフセットに 14 日（20160 分）を設定する場合の例を示す。なお、以下では、更新開始日時から NotAfter までを“更新期間”と記す。
  図 A-5 に示す更新期間の年月日の表記は、受信機が NotAfter の値と更新開始日時オフセットとから算出する更新期間（更新開始日時～NotAfter）を示し、点線の矢印は、受信機が当該更新期間の値を保持する期間を示す。
- ECM を暗号化する WorkKey の WorkKeyVersion の初期値は 1 とする。DRM サーバは、図 A-5 に示す“⑦ ECM を暗号化する WorkKey の更新”において、WorkKeyVersion を 1 から 2 に更新する。
- 事業者サーバは、契約時や解約時などに「WorkKey・SubscriptionTierBits・ExtractInfo」の取得のための UsageRuleReference・DRM サーバの URI（DRM サーバ URI）を含むファイルを提供する。以下、当該ファイルを“メタファイル”と記す。

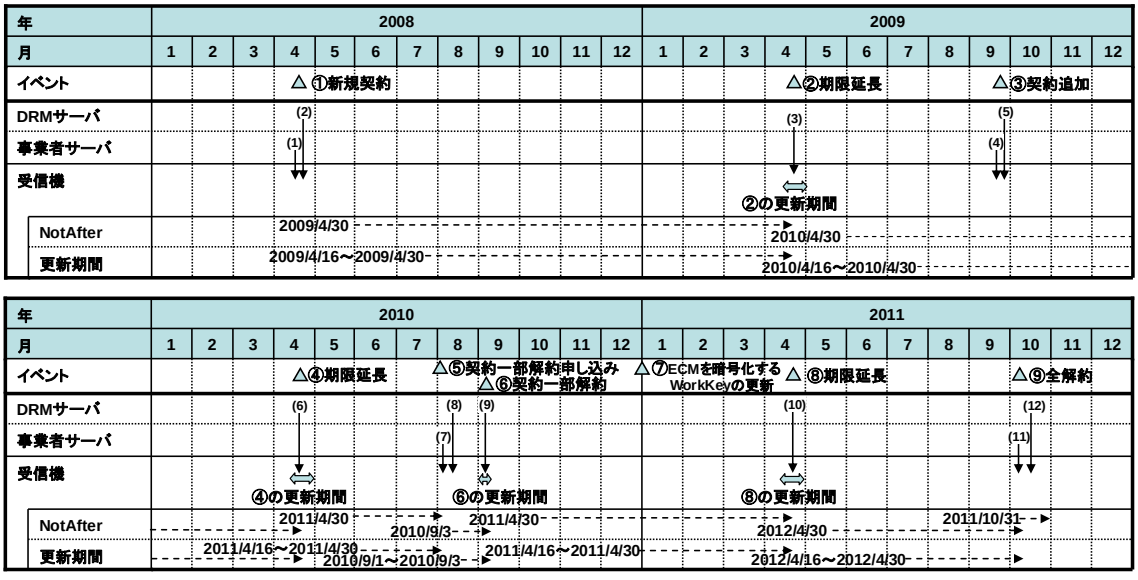

図 A-5 「WorkKey・SubscriptionTierBits・ExtractInfo」の更新運用の例

① 新規契約

受信機は、2008/4 に事業者サーバに新規に契約を申し込む。受信機は、事業者サーバから取得したメタファイルを用いて、DRM サーバから一対の「WorkKey・SubscriptionTierBits・ExtractInfo」を取得する。

(1) 事業者サーバからのメタファイルの取得

- 受信機は、事業者サーバに新規契約の要求を送信する。
- 事業者サーバは、受信機からの契約申し込みを受け付け、DRM サーバに契約内容を送信するなど、新規契約の受け付け処理を完了する。
- 受信機は、事業者サーバからメタファイルを取得し、不揮発性記憶領域に記録する。

(2) DRM サーバからの「WorkKey・SubscriptionTierBits・ExtractInfo」の取得

- 受信機は、メタファイルで指定される UsageRuleReference（odd）を設定した Get Permission Request message を作成して、メタファイルの DRM サーバ URI で指定される DRM サーバに送信する。
- DRM サーバは、UsageRuleReference（odd）に対応する「WorkKey（odd）・SubscriptionTierBits・ExtractInfo」を設定した Get Permission Reply message を作成して、受信機に送信する。
  - NotAfter には、加入月の 1 年後の月末の日時（2009/4/30）を設定する。
  - 送信する「WorkKey・SubscriptionTierBits・ExtractInfo」を次回に更新することを通知するため、更新開始日時オフセットの値には 14 日（20160 分）を設定する。
  - WorkKey（odd）には、WorkKey の送信時点で ECM を暗号化する WorkKey（WorkKeyVersion が 1）を設定する。
- 受信機は、DRM サーバから受信した Get Permission Reply message から「WorkKey（odd）・SubscriptionTierBits・ExtractInfo」を取得する。
- 受信機は、メタファイルで指定される UsageRuleReference（even）を設定した Get Permission Request message を作成して、メタファイルの DRM サーバ URI で指定される DRM サーバに送信する。
- DRM サーバは、UsageRuleReference（even）に対応する「WorkKey（even）・SubscriptionTierBits・ExtractInfo」を設定した Get Permission Reply message を作成して、受信機に送信する。
  - SubscriptionTierBits・ExtractInfo には、WorkKey（odd）の SubscriptionTierBits・ExtractInfo と同一の値を設定する。また、更新開始日時オフセットには、WorkKey（odd）の更新開始日時オフセットと同一の値を設定する。
  - WorkKey（even）には、ECM を暗号化する WorkKey の次回更新後の WorkKey（WorkKeyVersion が 2）を設定する。
- 受信機は、DRM サーバから受信した Get Permission Reply message から「WorkKey（even）・SubscriptionTierBits・ExtractInfo」を取得する。
- 受信機は、取得した一対の「WorkKey・SubscriptionTierBits・ExtractInfo」を不揮発性記憶領域に記録する。
- 受信機は、更新開始日時オフセットの値が 14 日（更新される）であることから、NotAfter と更新開始日時オフセットとを用いて次回の更新期間（②の更新期間：2009/4/16～2009/4/30）を算出し、次回更新の制御をおこなう。

[備考]

- DRM サーバ
  - DRM サーバは、WorkKey（odd）または WorkKey（even）のいずれかとして、WorkKey の送信時点で ECM を暗号化する WorkKey を送信する。他方は、ECM を暗号化する WorkKey の次回更新後の WorkKey を送信する。
  - DRM サーバは、受信機が有する契約の終了日時を経過するまでは、受信機に「WorkKey・SubscriptionTierBits・ExtractInfo」を送信する。送信回数は制限しない。
  - DRM サーバは、更新開始日時オフセットまたは NotAfter の値を受信機ごとなどに異なる値とすることで、更新時の受信機からのアクセスを分散させることができる。
- 受信機
  - メタファイルから取得した UsageRuleReference・DRM サーバ URI は、以降の「WorkKey・SubscriptionTierBits・ExtractInfo」の更新時に必要となるため、受信機は不揮発性記憶領域に記録しておくことが望ましい。また、この場合、受信機は新たにメタファイルを取得した場合、当該メタファイルに含まれる DRM サーバ URI の値で、不揮発性記憶領域に記録した DRM サーバ URI の値を更新することが望ましい。
  - 受信機は、同一 ServiceProviderID・同一 WorkKeyManagementID の odd/even の「WorkKey・SubscriptionTierBits・ExtractInfo」を一対の組として取得する。メタファイルでは、一対の「WorkKey・SubscriptionTierBits・ExtractInfo」に対応する 2 つの UsageRuleReference（odd）・UsageRuleReference（even）を組として、1 組以上の UsageRuleReference が指定される（1 サービス事業者が 64 ビットからなるティアビットを複数個用いる運用をおこなう場合などに、2 組以上の UsageRuleReference が指定される場合がある）。

受信機は、更新開始日時オフセットの値が 0000h（更新されない）である場合は、更新をおこなわない。

② 期限延長

受信機は、②の更新期間（2009/4/16～2009/4/30）に、期限延長された一対の「WorkKey・SubscriptionTierBits・ExtractInfo」を取得する。

(3) DRM サーバからの「WorkKey・SubscriptionTierBits・ExtractInfo」の取得

- 受信機は、不揮発性記憶領域に記録したメタファイルで指定される UsageRuleReference（odd）を設定した Get Permission Request message を作成して、メタファイルの DRM サーバ URI で指定される DRM サーバに送信する。
- DRM サーバは、UsageRuleReference（odd）に対応する「WorkKey（odd）・SubscriptionTierBits・ExtractInfo」を設定した Get Permission Reply message を作成して、受信機に送信する。
  - NotAfter には、1 年後の月末の日時（2010/4/30）を設定する。
  - 引き続き次回に更新をおこなうことを通知するため、更新開始日時オフセットの値には 14 日（20160 分）を設定する。
  - WorkKey（odd）には、WorkKey の送信時点で ECM を暗号化する WorkKey（WorkKeyVersion が 1）を設定する。
- 受信機は、DRM サーバから受信した Get Permission Reply message から「WorkKey（odd）・SubscriptionTierBits・ExtractInfo」を取得する。

- 受信機は、メタファイルで指定される UsageRuleReference（even）を設定した Get Permission Request message を作成して、メタファイルの DRM サーバ URI で指定される DRM サーバに送信する。
- DRM サーバは、UsageRuleReference（even）に対応する「WorkKey（even）・SubscriptionTierBits・ExtractInfo」を設定した Get Permission Reply message を作成して、受信機に送信する。
  - SubscriptionTierBits・ExtractInfo には、WorkKey（odd）の SubscriptionTierBits・ExtractInfo と同一の値を設定する。また、更新開始日時オフセットには、WorkKey（odd）の更新開始日時オフセットと同一の値を設定する。
  - WorkKey（even）には、ECM を暗号化する WorkKey の次回更新後の WorkKey（WorkKeyVersion が 2）を設定する。
- 受信機は、DRM サーバから受信した Get Permission Reply message から「WorkKey（even）・SubscriptionTierBits・ExtractInfo」を取得する。
- 受信機は、取得した一対の「WorkKey・SubscriptionTierBits・ExtractInfo」で、不揮発性記憶領域に記録した同一 ServiceProviderID・同一 WorkKeyManagementID の一対の「WorkKey・SubscriptionTierBits・ExtractInfo」を更新する。
- 受信機は、更新開始日時オフセットの値が 14 日（更新される）であることから、NotAfter と更新開始日時オフセットとを用いて次回の更新期間（④の更新期間：2010/4/16～2010/4/30）を算出し、次回更新の制御をおこなう。

[備考]

- DRM サーバ
  - DRM サーバは、更新開始日時（上記では 2009/4/16）以降に受信機から「WorkKey・SubscriptionTierBits・ExtractInfo」の要求を受信した場合、NotAfter を更新（上記では 2010/4/30）した「WorkKey・SubscriptionTierBits・ExtractInfo」を送信する。
- 受信機
  - メタファイルの DRM サーバ URI が変更される場合を考慮し、受信機は、期限延長時などにメタファイルを取得し、取得したメタファイルに設定された DRM サーバ URI の値で、不揮発性記憶領域に記録した DRM サーバ URI の値を更新することが望ましい。

③ 契約追加

受信機は、2009/9 に事業者サーバに契約の一部追加を申し込む。受信機は一部の契約が追加された一対の「WorkKey・SubscriptionTierBits・ExtractInfo」を取得する。

(4) 事業者サーバからのメタファイルの取得

- 受信機は、事業者サーバに対して契約の一部追加の要求を送信する。
- 事業者サーバは、受信機からの契約追加の申し込みを受け付け、DRM サーバに契約内容を送信するなど、契約追加の受け付け処理を完了する。
- 受信機は、事業者サーバからメタファイルを取得し、受信したメタファイルで不揮発性記憶領域に記録したメタファイルを更新する。

(5) DRM サーバからの「WorkKey・SubscriptionTierBits・ExtractInfo」の取得

- 受信機は、メタファイルで指定される UsageRuleReference（odd）を設定した Get Permission Request message を作成して、メタファイルの DRM サーバ URI で指定される DRM サーバに送信する。

- DRM サーバは、UsageRuleReference（odd）に対応する「WorkKey（odd）・SubscriptionTierBits・ExtractInfo」を設定した Get Permission Reply message を作成して、受信機に送信する。
  - SubscriptionTierBits には、②で設定した SubscriptionTierBits のうち、追加した契約に対応するビットを 1b に変更したビット列を設定する。
  - NotAfter には、引き続き②で設定した NotAfter（2010/4/30）と同一の値を設定する。また、更新開始日時オフセットにも、引き続き②で設定した更新開始日時オフセット（14 日）と同一の値を設定する。
  - WorkKey（odd）には、WorkKey の送信時点で ECM を暗号化する WorkKey（WorkKeyVersion が 1）を設定する。
- 受信機は、DRM サーバから受信した Get Permission Reply message から「WorkKey（odd）・SubscriptionTierBits・ExtractInfo」を取得する。
- 受信機は、メタファイルで指定される UsageRuleReference（even）を設定した Get Permission Request message を作成して、メタファイルの DRM サーバ URI で指定される DRM サーバに送信する。
- DRM サーバは、UsageRuleReference（even）に対応する「WorkKey（even）・SubscriptionTierBits・ExtractInfo」を設定した Get Permission Reply message を作成して、受信機に送信する。
  - SubscriptionTierBits・ExtractInfo には、WorkKey（odd）の SubscriptionTierBits・ExtractInfo と同一の値を設定する。また、更新開始日時オフセットには、WorkKey（odd）の更新開始日時オフセットと同一の値を設定する。
  - WorkKey（even）には、ECM を暗号化する WorkKey の次回更新後の WorkKey（WorkKeyVersion が 2）を設定する。
- 受信機は、DRM サーバから受信した Get Permission Reply message から「WorkKey（even）・SubscriptionTierBits・ExtractInfo」を取得する。
- 受信機は、取得した一対の「WorkKey・SubscriptionTierBits・ExtractInfo」で、不揮発性記憶領域に記録した同一 ServiceProviderID・同一 WorkKeyManagementID の一対の「WorkKey・SubscriptionTierBits・ExtractInfo」を更新する。
- 受信機は、更新開始日時オフセットの値が 14 日（更新される）であることから、NotAfter と更新開始日時オフセットとを用いて次回の更新期間（④の更新期間：2010/4/16～2010/4/30）を算出し、次回更新の制御をおこなう。

④ 期限延長
受信機は、④の更新期間（2010/4/16～2010/4/30）に、期限延長された一対の「WorkKey・SubscriptionTierBits・ExtractInfo」を取得する。

(6) DRM サーバからの「WorkKey・SubscriptionTierBits・ExtractInfo」の取得

- 受信機は、不揮発性記憶領域に記録したメタファイルで指定される UsageRuleReference（odd）を設定した Get Permission Request message を作成して、メタファイルの DRM サーバ URI で指定される DRM サーバに送信する。
- DRM サーバは、UsageRuleReference（odd）に対応する「WorkKey（odd）・SubscriptionTierBits・ExtractInfo」を設定した Get Permission Reply message を作成して、受信機に送信する。

- NotAfter には、1 年後の月末の日時（2011/4/30）を設定する。
- 引き続き次回に更新をおこなうことを通知するため、更新開始日時オフセットの値には 14 日（20160 分）を設定する。
- WorkKey（odd）には、WorkKey の送信時点で ECM を暗号化する WorkKey（WorkKeyVersion が 1）を設定する。

- 受信機は、DRM サーバから受信した Get Permission Reply message から「WorkKey（odd）・SubscriptionTierBits・ExtractInfo」を取得する。
- 受信機は、メタファイルで指定される UsageRuleReference（even）を設定した Get Permission Request message を作成して、メタファイルの DRM サーバ URI で指定される DRM サーバに送信する。
- DRM サーバは、UsageRuleReference（even）に対応する「WorkKey（even）・SubscriptionTierBits・ExtractInfo」を設定した Get Permission Reply message を作成して、受信機に送信する。
  - SubscriptionTierBits・ExtractInfo には、WorkKey（odd）の SubscriptionTierBits・ExtractInfo と同一の値を設定する。また、更新開始日時オフセットには、WorkKey（odd）の更新開始日時オフセットと同一の値を設定する。
  - WorkKey（even）には、ECM を暗号化する WorkKey の次回更新後の WorkKey（WorkKeyVersion が 2）を設定する。
- 受信機は、DRM サーバから受信した Get Permission Reply message から「WorkKey（even）・SubscriptionTierBits・ExtractInfo」を取得する。
- 受信機は、取得した一対の「WorkKey・SubscriptionTierBits・ExtractInfo」で、不揮発性記憶領域に記録した同一 ServiceProviderID・同一 WorkKeyManagementID の一対の「WorkKey・SubscriptionTierBits・ExtractInfo」を更新する。
- 受信機は、更新開始日時オフセットの値が 14 日（更新される）であることから、NotAfter と更新開始日時オフセットとを用いて次回の更新期間（⑧の更新期間：2011/4/16～2011/4/30）を算出し、次回更新の制御をおこなう。

⑤ 契約一部解約申し込み

受信機は、2010/8 初旬に事業者サーバに一部の契約の解約を申し込む。DRM サーバは、解約するチャネルが当月末まで視聴可能となるように、SubscriptionTierBits は解約前と同一で、期限が来月初めに設定された一対の「WorkKey・SubscriptionTierBits・ExtractInfo」を受信機に送信する。

受信機は、当該「WorkKey・SubscriptionTierBits・ExtractInfo」を取得し、NotAfter に基づき、一部の契約が解約された一対の「WorkKey・SubscriptionTierBits・ExtractInfo」を、来月初めに取得するための制御をおこなう。

(7) 事業者サーバからのメタファイルの取得

- 受信機は、事業者サーバに対して契約の一部解約の要求を送信する。
- 事業者サーバは、受信機からの契約一部解約の申し込みを受け付け、DRM サーバに解約内容を送信するなど、契約一部解約の受付処理を完了する。
- 受信機は、事業者サーバからメタファイルを取得し、受信したメタファイルで不揮発性記憶領域に記録したメタファイルを更新する。

(8) DRM サーバからの「WorkKey・SubscriptionTierBits・ExtractInfo」の取得

- 受信機は、メタファイルで指定される UsageRuleReference（odd）を設定した Get Permission Request message を作成して、メタファイルの DRM サーバ URI で指定される DRM サーバに送信する。
- DRM サーバは、UsageRuleReference（odd）に対応する「WorkKey（odd）・SubscriptionTierBits・ExtractInfo」を設定した Get Permission Reply message を作成して、受信機に送信する。
  - SubscriptionTierBits には、引き続き、③および④で設定した契約の一部解約前の SubscriptionTierBits を設定する。
  - 2010/9/1 からの一部解約のための更新を指定し、かつ、更新期間を 3 日間とするため、NotAfter には 2010/9/3、更新開始日時オフセットには 3 日（4320 分）を設定する。
  - WorkKey（odd）には、WorkKey の送信時点で ECM を暗号化する WorkKey（WorkKeyVersion が 1）を設定する。
- 受信機は、DRM サーバから受信した Get Permission Reply message から「WorkKey（odd）・SubscriptionTierBits・ExtractInfo」を取得する。
- 受信機は、メタファイルで指定される UsageRuleReference（even）を設定した Get Permission Request message を作成して、メタファイルの DRM サーバ URI で指定される DRM サーバに送信する。
- DRM サーバは、UsageRuleReference（even）に対応する「WorkKey（even）・SubscriptionTierBits・ExtractInfo」を設定した Get Permission Reply message を作成して、受信機に送信する。
  - SubscriptionTierBits・ExtractInfo には、WorkKey（odd）の SubscriptionTierBits・ExtractInfo と同一の値を設定する。また、更新開始日時オフセットには、WorkKey（odd）の更新開始日時オフセットと同一の値を設定する。
  - WorkKey（even）には、ECM を暗号化する WorkKey の次回更新後の WorkKey（WorkKeyVersion が 2）を設定する。
- 受信機は、DRM サーバから受信した Get Permission Reply message から「WorkKey（even）・SubscriptionTierBits・ExtractInfo」を取得する。
- 受信機は、取得した一対の「WorkKey・SubscriptionTierBits・ExtractInfo」で、不揮発性記憶領域に記録した同一 ServiceProviderID・同一 WorkKeyManagementID の一対の「WorkKey・SubscriptionTierBits・ExtractInfo」を更新する。
- 受信機は、更新開始日時オフセットの値が 3 日（更新される）であることから、NotAfter と更新開始日時オフセットとを用いて次回の更新期間（⑥の更新期間：2010/9/1～2010/9/3）を算出し、次回更新の制御をおこなう。

⑥ 契約一部解約

受信機は、⑥の更新期間（2010/9/1～2010/9/3）に、一部の契約が解約された一対の「WorkKey・SubscriptionTierBits・ExtractInfo」を取得する。

(9) DRM サーバからの「WorkKey・SubscriptionTierBits・ExtractInfo」の取得

- 受信機は、不揮発性記憶領域に記録したメタファイルで指定される UsageRuleReference（odd）を設定した Get Permission Request message を作成して、メタファイルの DRM サーバ URI で指定される DRM サーバに送信する。

- DRM サーバは、UsageRuleReference（odd）に対応する「WorkKey（odd）・SubscriptionTierBits・ExtractInfo」を設定した Get Permission Reply message を作成して、受信機に送信する。
  - SubscriptionTierBits には、⑤で設定した SubscriptionTierBits のうち、解約した契約に対応するビットを 0b に変更したビット列を設定する。
  - NotAfter には、⑧の更新期間での期限延長を継続するため、④の期限延長で設定した NotAfter（2011/4/30）と同一の値を設定する。また、更新開始日時オフセットにも、引き続き④で設定した更新開始日時オフセット（14 日）と同一の値を設定する。
  - WorkKey（odd）には、WorkKey の送信時点で ECM を暗号化する WorkKey（WorkKeyVersion が 1）を設定する。
- 受信機は、DRM サーバから受信した Get Permission Reply message から「WorkKey（odd）・SubscriptionTierBits・ExtractInfo」を取得する。
- 受信機は、メタファイルで指定される UsageRuleReference（even）を設定した Get Permission Request message を作成して、メタファイルの DRM サーバ URI で指定される DRM サーバに送信する。
- DRM サーバは、UsageRuleReference（even）に対応する「WorkKey（even）・SubscriptionTierBits・ExtractInfo」を設定した Get Permission Reply message を作成して、受信機に送信する。
  - SubscriptionTierBits・ExtractInfo には、WorkKey（odd）の SubscriptionTierBits・ExtractInfo と同一の値を設定する。また、更新開始日時オフセットには、WorkKey（odd）の更新開始日時オフセットと同一の値を設定する。
  - WorkKey（even）には、ECM を暗号化する WorkKey の次回更新後の WorkKey（WorkKeyVersion が 2）を設定する。
- 受信機は、DRM サーバから受信した Get Permission Reply message から「WorkKey（even）・SubscriptionTierBits・ExtractInfo」を取得する。
- 受信機は、取得した一対の「WorkKey・SubscriptionTierBits・ExtractInfo」で、不揮発性記憶領域に記録している同一 ServiceProviderID・同一 WorkKeyManagementID の一対の「WorkKey・SubscriptionTierBits・ExtractInfo」を更新する。
- 受信機は、更新開始日時オフセットの値が 14 日（更新される）であることから、NotAfter と更新開始日時オフセットとを用いて次回の更新期間（⑧の更新期間：2011/4/16～2011/4/30）を算出し、次回更新の制御をおこなう。

[備考]

- DRM サーバ
  - DRM サーバは、「WorkKey・SubscriptionTierBits・ExtractInfo」の送信に対して、受信機から以下の SAC 層のメッセージを受信することにより、受信機での「WorkKey・SubscriptionTierBits・ExtractInfo」の更新を確認できる。
    - ✧ Command に「Commit」が設定された Encrypted command message
    - ✧ Request message

    したがって、DRM サーバは、odd/even の一対の「WorkKey・SubscriptionTierBits・ExtractInfo」の送信に対して、上記メッセージを受信することにより契約の一部解約処理を完了できる。なお、上記は Get Permission Protocol と Packed Message Protocol とで共通である。

⑦ ECM を暗号化する WorkKey の更新
DRM サーバは、現在送出中の ECM を暗号化する WorkKey が更新された場合、Get Permission Protocol で受信機に送信する WorkKey を更新する。
DRM サーバは、ECM を暗号化する WorkKey を、WorkKeyVersion が 1 の WorkKey（odd）から WorkKeyVersion が 2 の WorkKey（even）に更新する。以降、DRM サーバは、WorkKeyVersion が 2 の WorkKey（even）と、WorkKeyVersion が 3 の WorkKey（odd）とを受信機に送信する。

[備考]
- DRM サーバは、ECM を暗号化する WorkKey の更新後、Get Permission Protocol で送信する一対の WorkKey（odd）・WorkKey（even）も速やかに更新する。
- DRM サーバが WorkKey を更新するタイミングは、サービス事業者の運用依存とする。
- DRM サーバは、WorkKey 自体を定期的に更新する運用をおこなう場合、ECM を暗号化する WorkKey の更新前に、全ての受信機が更新後に ECM の暗号化に用いる WorkKey を取得できるように各受信機の更新期間を設定する。

⑧ 期限延長
受信機は、⑧の更新期間（2011/4/16～2011/4/30）に、期限延長された一対の「WorkKey・SubscriptionTierBits・ExtractInfo」を取得する。

(10) DRM サーバからの「WorkKey・SubscriptionTierBits・ExtractInfo」の取得
- 受信機は、不揮発性記憶領域に記録したメタファイルで指定される UsageRuleReference（odd）を設定した Get Permission Request message を作成して、メタファイルの DRM サーバ URI で指定される DRM サーバに送信する。
- DRM サーバは、UsageRuleReference（odd）に対応する「WorkKey（odd）・SubscriptionTierBits・ExtractInfo」を設定した Get Permission Reply message を作成して、受信機に送信する。
  - NotAfter には、1 年後の月末の日時（2012/4/30）を設定する。
  - 引き続き次回に更新することを通知するため、更新開始日時オフセットの値には 14 日（20160 分）を設定する。
  - WorkKey（odd）には、ECM を暗号化する WorkKey の次回更新後の WorkKey（WorkKeyVersion が 3）を設定する。
- 受信機は、DRM サーバから受信した Get Permission Reply message から「WorkKey（odd）・SubscriptionTierBits・ExtractInfo」を取得する。
- 受信機は、メタファイルで指定される UsageRuleReference（even）を設定した Get Permission Request message を作成して、メタファイルの DRM サーバ URI で指定される DRM サーバに送信する。
- DRM サーバは、UsageRuleReference（even）に対応する「WorkKey（even）・SubscriptionTierBits・ExtractInfo」を設定した Get Permission Reply message を作成して、受信機に送信する。
  - SubscriptionTierBits・ExtractInfo には、WorkKey（odd）の SubscriptionTierBits・ExtractInfo と同一の値を設定する。また、更新開始日時オフセットには、WorkKey（odd）の更新開始日時オフセットと同一の値を設定する。

- WorkKey（even）には、WorkKey の送信時点で ECM を暗号化する WorkKey（WorkKeyVersion が 2）を設定する。
- 受信機は、DRM サーバから受信した Get Permission Reply message から「WorkKey（even）・SubscriptionTierBits・ExtractInfo」を取得する。
- 受信機は、取得した一対の「WorkKey・SubscriptionTierBits・ExtractInfo」で、不揮発性記憶領域に記録した同一 ServiceProviderID・同一 WorkKeyManagementID の一対の「WorkKey・SubscriptionTierBits・ExtractInfo」を更新する。
- 受信機は、更新開始日時オフセットの値が 14 日（更新される）であることから、NotAfter と更新開始日時オフセットとを用いて次回の更新期間（2012/4/16～2012/4/30、図 A-5 に図示せず）を算出し、次回更新の制御をおこなう。

⑨ 全解約

受信機は、2011/10 初旬に事業者サーバに当該 WorkKey に関する全契約の解約を申し込む。受信機は、解約日を期限とした一対の「WorkKey・SubscriptionTierBits・ExtractInfo」を取得する。

(11) 事業者サーバからのメタファイルの取得

- 受信機は、事業者サーバに対して全契約の解約の要求を送信する。
- 事業者サーバは、受信機からの全解約申し込みを受け付け、DRM サーバに解約内容を送信するなど、全解約の受付処理を完了する。
- 受信機は、事業者サーバからメタファイルを取得し、受信したメタファイルで不揮発性記憶領域に記録したメタファイルを更新する。

(12) DRM サーバからの「WorkKey・SubscriptionTierBits・ExtractInfo」の取得

- 受信機は、メタファイルで指定される UsageRuleReference（odd）を設定した Get Permission Request message を作成して、メタファイルの DRM サーバ URI で指定される DRM サーバに送信する。
- DRM サーバは、UsageRuleReference（odd）に対応する「WorkKey（odd）・SubscriptionTierBits・ExtractInfo」を設定した Get Permission Reply message を作成して、受信機に送信する。
  - NotAfter には、解約月の月末の日時（2011/10/31）を設定する。
  - 当該受信機の以降の更新を停止するため、更新開始日時オフセットの値には 0000h を設定する。
  - WorkKey（odd）には、ECM を暗号化する WorkKey の次回更新後の WorkKey（WorkKeyVersion が 3）を設定する。
- 受信機は、DRM サーバから受信した Get Permission Reply message から「WorkKey（odd）・SubscriptionTierBits・ExtractInfo」を取得する。
- 受信機は、メタファイルで指定される UsageRuleReference（even）を設定した Get Permission Request message を作成して、メタファイルの DRM サーバ URI で指定される DRM サーバに送信する。
- DRM サーバは、UsageRuleReference（even）に対応する「WorkKey（even）・SubscriptionTierBits・ExtractInfo」を設定した Get Permission Reply message を作成して、受信機に送信する。
  - SubscriptionTierBits・ExtractInfo には、WorkKey（odd）の SubscriptionTierBits・ExtractInfo と同一の値を設定する。また、更新開始日時オフセットには、WorkKey（odd）の更新開始日時オフセットと同一の値を設定する。

- WorkKey（even）には、WorkKey の送信時点で ECM を暗号化する WorkKey（WorkKeyVersion が 2）を設定する。
- 受信機は、DRM サーバから受信した Get Permission Reply message から「WorkKey（even）・SubscriptionTierBits・ExtractInfo」を取得する。
- 受信機は、取得した一対の「WorkKey・SubscriptionTierBits・ExtractInfo」で、不揮発性記憶領域に記録した同一 ServiceProviderID・同一 WorkKeyManagementID の一対の「WorkKey・SubscriptionTierBits・ExtractInfo」を更新する。
- 受信機は、更新開始日時オフセットの値が 0000h（更新されない）であることから、次回以降の更新はおこなわない。

[備考]

- DRM サーバ
  - DRM サーバは、「WorkKey・SubscriptionTierBits・ExtractInfo」の送信に対して、受信機から以下の SAC 層のメッセージを受信することにより、受信機の「WorkKey・SubscriptionTierBits・ExtractInfo」の更新を確認できる。
    - Command に「Commit」が設定された Encrypted command message
    - Request message

    したがって、DRM サーバは、odd/even の一対の「WorkKey・SubscriptionTierBits・ExtractInfo」の送信に対して、上記メッセージを受信することにより解約処理を完了できる。なお、上記は Get Permission Protocol と Packed Message Protocol とで共通である。
- 受信機
  - 「WorkKey・SubscriptionTierBits・ExtractInfo」の削除は、受信機の実装依存とする。ただし、更新開始日時オフセットの値が 0000h（更新されない）である場合、NotAfter を経過した「WorkKey・SubscriptionTierBits・ExtractInfo」を削除することが望ましい。このとき、不揮発性記憶領域に記録した、対応するメタファイルも削除することが望ましい。

A.6 メッセージの例

A.6.1 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.6.2 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.6.2.1Challenge message

1001byte の SinkCertificate を格納した Challenge message の例を表 A-5 に示す。

表 A-5　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.6.2.2Response & Challenge message

1171byte の SourceCertificate を格納した Response & Challenge message の例を表 A-6 に示す。

表 A-6　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.6.2.3Response & Request message

GetPermission Request を格納した Response & Request message の例を表 A-7 に示す。

表 A-7　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.6.3.1.2 項を参照のこと
204-235	MessageDigest	（省略）

A.6.2.4Request 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-8 に示す。

表 A-8　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.6.3.1.2 項を参照のこと
92-123	MessageDigest	（省略）

A.6.2.5Reply message

Get Permission Reply を格納した Reply message の例を表 A-9 に示す。

表 A-9　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.6.3.1.3 項を参照のこと
62-93	MessageDigest	（省略）

A.6.2.6Plain command message

Status として Error other than the below を格納した Plain command message の例を表 A-10 に示す。

表 A-10　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.6.2.7Encrypted 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-11 に示す。

表 A-11　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.6.3 Service Protocol のメッセージ例

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.6.3.1Get 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.6.3.1.1.　DeviceInformation

[MIPTV], 3.2.2 項で規定される DeviceInformation の例を表 A-12 に示す。この情報は Get Permission Request メッセージに格納される。

表 A-12　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）

1840

1841 ***A.6.3.1.2.　Get Permission Request message***

1842 [MIPTV], 4.2.1.2 項で規定される Get Permission Request message の例を、表 A-13
1843 に示す。

表 A-13　Get Permission Request message の例

バイト インデックス	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.6.3.1.1 項参照
16-31	UsageRuleReference	00010000010100000000000000000000
32	ActionID	02（固定値）
33	ActionParameter	FF（固定値）
34-35	SpecificCRID	0002（固定値）
36	PrivateDataTag	00（固定値）
37-63	PrivateData	全て 00（固定値）

A.6.3.1.3. Get Permission Reply message

[MIPTV], 4.2.1.3 項で規定される Get Permission Reply message の例を、表 A-14 に示す。

表 A-14　Get Permission Reply message の例

バイト インデックス	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	WorkKey	00112233445566778899AABBCCDDEEFF
22-27	WorkKeyID	000100000101
28-29	PrivateData	4EC0
30-37	SubscriptionTierBits	8000000000000000
38-39	ExtractInfoSize	0A（固定値）
40-43	NotBefore	FFFFFFFF
44-47	NotAfter	4B3CCAAF
48-49	RenderingObligation	0000（固定値）

A.6.3.2Get 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.6.3.2.1 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-15 に示す。

表 A-15　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.6.3.2.2　Get TrustedTime 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-16 に示す。

表 A-16　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.6.3.3Packed 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.6.3.3.1　Packed Message Request message

[MIPTV], 4.2.3.2 項で規定される Packed Message Request に格納するリクエストメッセージが A.6.3.1.2 項の場合の例を、表 A-17 に示す。

表 A-17　Packed Message Request message の例

バイト インデックス	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	NumberOfRequest MessageBoxes	0002
6-7	RequestMessageSize	0040
8-71	RequestMessage	A.6.3.1.2 項参照
72-73	RequestMessageSize	0040
74-137	RequestMessage	A.6.3.1.2 項参照

A.6.3.3.2　Packed Message Reply message

[MIPTV], 4.2.3.3 項で規定される Packed Message Reply message で、A.6.3.3.1 項に対して応答する場合の例を、表 A-18 に示す。

表 A-18　Packed Message Reply message の例

バイト インデックス	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	NumberOfReplyMessage Boxes	0002
8-9	ReplyMessageSize	0032
10-59	ReplyMessage	A.6.3.1.3 項参照
34-35	ReplyMessageSize	0032
36-85	ReplyMessage	A.6.3.1.3 項参照

1880