DRM Gateway eDRM Specification

Document no	EDGS-151
Version	A9
Created on	2026-04-16

CONFIDENTIAL

Contents

1 Confidentiality Notice

2 About This Document

This document describes and specifies eDRM, (Edgeware DRM) interface

3 History

A brief history of this document:

Version	Date	Description
A1 draft1	2015-09-13	Initial draft
A1 draft2	2016-02-26	Use field encryption instead of `protection_format`
A1	2016-10-24	Allow multiple DRM systems in response with `cenc` encryption Introduced `content_id` and `iv` response fields
A2	2016-12-06	`sample-aes` encryption
A3	2017-01-03	DRM system specific data in input messages
A4	2017-05-08	Multiple keys in response messages
A5	2018-11-20	API endpoint for CCMI URL
A6	2019-05-22	Clarifications on key rotation
A7	2019-09-02	Description of message format; Key per tracks
A8	2024-10-30	HTTP response codes updates
A9	2025-12-19	Drop API v1 Update examples to API v2 Split JSON schema between input and out message

4 Introduction

`eDRM` is a HTTP/REST/JSON based interface for integrating repackagers with external DRM systems. The repackager implements the client side of the interface. The specification hence describes how an `eDRM` server should be implemented.

5 Security

`eDRM` is secured with these steps

- All traffic is encrypted with HTTPS
- Server authenticity is verified with HTTPS
- Client authenticity is verified via a shared secret between the `eDRM` server and client.

6 Interface specification

HTTP 1.1 over TLS v1.1+ (HTTPS) must be used.

6.1 eDRM Resource

The requests are done to a URL that is configured in the client system. This means that it is up to the server implementation to define the resource name, and define a way to separate different eDRM server instances via the URL. The server is also responsible for protecting the request with a shared secret to avoid unauthorized client access.

All examples in this document assume that requests to eDRM use CCMI URLs.

6.2 HTTP request methods

The following HTTP methods should be supported by the server:

6.2.1 POST

Send request for a DRM resource.

6.3 HTTP request headers

6.3.1 Content-Type

All requests contain a `Content-type: application/json` header.

6.3.2 Content-Length

All requests that contain a body must include a `Content-Length: _length of body_`

6.4 HTTP response codes

The following HTTP response codes are allowed to be generated by the server

6.4.1 200 OK

The POST request was successfully completed.

6.4.2 400 Bad request

The syntax of the request is incorrect.

6.4.3 403 Forbidden

The values cannot be set according to request.

6.4.4 404 Not Found

The resource does not exist.

6.4.5 405 Method Not Allowed

The request method is not allowed.

6.4.6 500 Internal Server Error

The server encountered an unexpected condition that prevents it from fulfilling the request.

6.4.7 502 Bad Gateway

The server received an invalid response from the upstream server it accessed in attempting to fulfill the request.

6.4.8 503 Temporarily out of Service

The service is unavailable or overloaded at the moment.

6.4.9 504 Gateway Timeout

This error response is given when the server is acting as a gateway and cannot get a response in time.

6.4.10 506 Variant Also Negotiates

The server has an internal configuration error.

6.5 HTTP response headers

6.5.1 Content-type

All responses that contain a body must include a "Content-type: application/json" header.

6.5.2 Content-Length

All responses that contain a body must include a "Content-Length *length of body*" header.

6.6 Message format

6.6.1 Input message format

API v1 is discontinued. Only API v2 (CCMI API) is accepted.

eDRM message in CCMI API is a request for drm information of an asset. The URL contains resource id of the asset and output profile, while mandatory fields of body are shared secret and position.

Position is used to request the keys for the content that uses time-based key rotation. Position can be either a string or a list of up to two numbers. Two element list represents interval with start and stop time in epoch seconds. Single element list represents open ended interval, with start time in epoch seconds. Empty list is used if the input message does not contain position information. String value is not described by this specification and may be treated as empty list position.

Optionally, key request can contain list of variants (tracks in the asset) with their attributes. For such requests, response can contain different key info for different variants.

6.6.1.1 JSON schema

```
{
    "$schema": "http://json-schema.org/draft-04/schema#",
    "description": "The schema for eDRM interface version 2 of input message
       ↪ ",
    "definitions": {
        "variant_info": {
            "type": "object",
            "properties": {
                "name": {
                    "type": "string",
                    "description": "Unique name of the variant in the asset."
                },
                "media_type": {
                    "type": "string",
                    "enum": [
                        "video",
                        "audio",
                        "text"
                    ],
                    "description": "Media type of variant"
                },
                "bitrate": {
                    "type": "number",
                    "description": "Variant's bitrate"
                },
                "codec": {
                    "type": "string",
                    "description": "Variant's codec string"
                },
                "width": {
                    "type": "number",
                    "description": "Width of the picture. Applicable only for
                                    variants with media type video"
                },
                "height": {
```

```
                    "type": "number",
                    "description": "Height of the picture. Applicable only
                        ↪ for
                                    variants with media type video"
                }
            },
            "required": [
                "name",
                "media_type"
            ]
        }
    },
    "type": "object",
    "allOf": [
        {
            "properties": {
                "position": {
                    "oneOf": [
                        {
                            "type": "string",
                            "description": "Indicate key position for the
                                ↪ given
                                            resource_id. Set by client."
                        },
                        {
                            "type": "array",
                            "description": "Start and end time for key
                                ↪ rotation
                                            interval. List with one element
                                            specify start position only.",
                            "maxItems": 2,
                            "items": {
                                "type": "number"
                            }
                        }
                    ]
                },
                "shared_secret": {
                    "type": "string",
                    "description": "Set by client to authenticate itself.
                                    Required in CCMI API calls."
                },
                "variants": {
                    "type": "array",
                    "description": "List of variants in asset. Optional in
                                    CCMI API calls.",
                    "items": {
                        "$ref": "#/definitions/variant_info"
                    }
                }
            }
        }
    ]
}
```

6.6.2 Output message format

Response eDRM message for key request should contain a list with one or more key info elements.

Key info element in basic form contains a key id, key and iv triplet and a list of system specific drm info's. In the case that drm info list contains one element, surrounding list of that single drm info element can be omitted.

System specific drm info must have a header data property and it may also contain system id and drm name properties. System id is UUID string identifying DRM system and drm name is its human readable label. For the protection schemes that supports multiple DRM systems (e.g. CENC) each drm info should specify system id property.

Key info can optionally contain key applicability period, given by start and end times.

Additionally, key info element can be only applicable for a certain list of variants. Those variant names are then listed in key info.

Response message can have key info elements with `plaintext` flag set to `true` to signal what variant in the asset should not be encrypted. Such key info element should provide a list of variants, and it may also provide a start and stop times. Key id, key, iv or system specific drm info's should not be present in such key info element.

Every variant provided in input message should be represented in the output message. Each variant should not be present in more then one key info with same start and stop times.

If the response contains one key info element without time or variant information, key info properties can alternatively be present in the root element of the message, and `key_info` list omitted.

Response eDRM message should contain resource id and position that are identical to the values provided in request. If the eDRM server operates with UUIDs internally instead of string based resource ids, it might provide such UUID in content id property in response.

Response message may contain `time_to_next_poll` property that indicates how many seconds the client should wait before requesting new keys. `time_to_next_poll` should be present in response if the input message was for open ended interval, and server responded with key info's that have start and stop times.

6.6.2.1 JSON schema

```
{
    "$schema": "http://json-schema.org/draft-04/schema#",
    "description": "The schema for eDRM interface version 2 of output message
       ↪ ",
    "definitions": {
        "system_info": {
            "type": "object",
            "properties": {
                "system_id": {
                    "type": "string",
                    "description": "Hexencoded 128-bit binary identifying the
                                    protection system."
                },
                "drm": {
                    "type": "string",
                    "description": "Human readable DRM system identifier."
                },
                "header_data": {
                    "type": "string",
                    "description": "System specific data needed by client
                                    to decrypt content. For aes-128 and
                                    sample-aes it is the URL. For playready
                                       ↪ and
                                    cenc it is base64 encoded data object."
                },
                "header_data_expiry_time": {
                    "type": "number",
                    "description": "A relative time (in seconds), starting
                       ↪ from
                                    the response receive time, after which
                                       ↪ the
                                    header data becomes invalid."
                }
            }
        },
```

```
        "system_infos": {
            "oneOf": [
                {
                    "$ref": "#/definitions/system_info"
                },
                {
                    "type": "array",
                    "items": {
                        "$ref": "#/definitions/system_info"
                    }
                }
            ]
        },
        "key_info": {
            "type": "object",
            "properties": {
                "key_id": {
                    "type": "string",
                    "description": "Base64 encoded identification string for
                                    encryption key. Set by client or server."
                },
                "key": {
                    "type": "string",
                    "description": "Base64 encoded encryption key. Set by
                                    client or server."
                },
                "iv": {
                    "type": "string",
                    "description": "Base64 encoded encryption intialization
                                    vector. Set by client or server."
                },
                "start_time": {
                    "type": "number",
                    "description": "Start of validity period for key, in
                                    seconds since epoch start."
                },
                "end_time": {
                    "type": "number",
                    "description": "End of validity period for key, in
                       ↪ seconds
                                    since epoch start."
                },
                "variants": {
                    "type": "array",
                    "description": "List of variant names that this key info
                       ↪ applies to.",
                    "items": {
                        "type": "string"
                    }
                },
                "plaintext": {
                    "type": "boolean",
                    "description": "When set to true, signals that listed
                       ↪ variants
                                    should not be encrypted."
                },
                "playready": {
                    "$ref": "#/definitions/system_infos",
                    "description": "Added by client or server if playready
                                    encryption is required."
                },
                "aes-128": {
```

```
                "$ref": "#/definitions/system_infos",
                "description": "Added by client or server if aes-128
                                encryption is required."
            },
            "sample-aes": {
                "$ref": "#/definitions/system_infos",
                "description": "Added by client or server if sample-aes
                                encryption is required."
            },
            "cenc": {
                "$ref": "#/definitions/system_infos",
                "description": "Added by client or server if cenc
                                encryption is required. Contains list of
                                DRM system specific objects"
            }
        },
        "required": ["key", "key_id"]
    }
},
"type": "object",
"allOf": [
    {
        "properties": {
            "resource_id": {
                "type": "string",
                "description": "External identification string for
                   ↪ content."
            },
            "content_id": {
                "type": "string",
                "description": "Guid based on resource_id. Set by server
                   ↪ ."
            },
            "position": {
                "oneOf": [
                    {
                        "type": "string",
                        "description": "Indicate key position for the
                           ↪ given
                                        resource_id. Set by client."
                    },
                    {
                        "type": "array",
                        "description": "Start and end time for key
                           ↪ rotation
                                        interval. List with one element
                                        specify start position only.",
                        "maxItems": 2,
                        "items": {
                           "type": "number"
                        }
                    }
                ]
            },
            "time_to_next_poll": {
                "type" : "number",
                "description": "Number of seconds until the client should
                   ↪  ask
                                for new keys."
            }
        }
    },
```

```
        {
            "oneOf": [
                {
                    "$ref": "#/definitions/key_info"
                },
                {
                    "properties": {
                        "key_info": {
                            "type": "array",
                            "description": "List of encryption keys with drm
                                            data. If message contains only
                                               ↪ one
                                            key without validity period
                                            timestamps, key_info can be
                                               ↪ omitted
                                            and data for key can be sent in
                                            the root element.",
                            "items": {
                                "$ref": "#/definitions/key_info"
                            }
                        }
                    }
                }
            ]
        }
    ]
}
```

7 Procedures

- An encryption key is requested by repackager through a HTTP POST to the eDRM server with the CCMI URL.
- The HTTP 200 response will include a body with the encryption key and additional data.

7.1 Examples

7.1.1 Request an encryption key for HLS

```
POST /__cl/s:esf/__c/content/__op/hls-drm/__f/index.m3u8 HTTP/1.1
Content-Type: application/json
Content-Length: ...
{
    "shared_secret": "shared-secret",
    "position": "0",
    "variants": [
        ...
    ]
}

HTTP/1.1 200 OK
Content-Type: application/json
Content-Length: ...
{
    "resource_id": "content",
    "position": "0",
    "encryption": "aes-128",
    "content_id": "4da98ac7-a4a2-5c1a-92d9-d7343378b69f",
    "key": "bReUiCnBVvG71WkTVi0UEw==",
    "key_id": "bReUiCnBVvG71WkTVi0UEw==",
    "aes-128": {
```

```
        "drm": "Verimatrix",
        "header_data": "https://keyserver/url-for-key_id"
    }
}
```

7.1.2 Request a Playready encryption key for Smooth Streaming

```
POST /__cl/s:esf/__c/content/__op/mss-drm/__f/Manifest HTTP/1.1
Content-Type: application/json
Content-Length: ...
{
    "shared_secret": "shared-secret",
    "position": "0",
    "variants": [
        ...
    ]
}

HTTP/1.1 200 OK
Content-Type: application/json
Content-Length: ...
{
    "resource_id": "content",
    "position": "0",
    "encryption": "playready",
    "content_id": "4da98ac7-a4a2-5c1a-92d9-d7343378b69f",
    "key": "OiobaN0r2bLusl6ExHdmaA==",
    "key_id": "EAAAABAAEAAQABAAAAAAAQ==",
    "playready": {
        "drm": "Playready",
        "system_id": "9a04f079-9840-4286-ab92-e65be0885f95",
        "header_data": "+gIAAAEAAQDwA..."
    }
}
```

7.1.3 Request an encryption key and DRM information for CENC

```
POST /__cl/s:esf/__c/content/__op/dash-drm/__f/manifest.mpd HTTP/1.1
Content-Type: application/json
Content-Length: ...
{
    "shared_secret": "shared-secret",
    "position": "0",
    "variants": [
        ...
    ]
}

HTTP/1.1 200 OK
Content-Type: application/json
Content-Length: ...
{
    "resource_id": "content",
    "position": "0",
    "encryption": "cenc",
    "content_id": "4da98ac7-a4a2-5c1a-92d9-d7343378b69f",
    "key": "FmY0xnWCPCNaSpRG+tUuTQ==",
    "key_id": "nrQFDeRLSAKTLifXUIPiZg==",
    "cenc": [
```

```
        {
            "drm": "Axinom Widevine",
            "system_id": "edef8ba9-79d6-4ace-a3c8-27dcd51d21ed",
            "header_data": "AAAANHBzc2gAAAAA7e+
                ↪ LqXnWSs6jyCfc1R0h7QAAABQIARIQnrQFDeRLSAKTLifXUIPiZg=="
        },
        {
            "drm": "Playready",
            "system_id": "9a04f079-9840-4286-ab92-e65be0885f95",
            "header_data": "AAADdnBzc2g..."
        },
        {
            "drm": "ClearKey",
            "system_id": "1077efec-c0b2-4d02-ace3-3c1e52e2fb4b",
            "header_data": "
                ↪ AAAANHBzc2gBAAAAEHfv7MCyTQKs4zweUuL7SwAAAAGetAUN5EtIApMuJ9dQg
                ↪ +JmAAAAAA=="
        }
    ]
}
```

All items in the cenc array contain pssh boxes specific to that DRM system. All DRM systems use the same key and key_id.

7.1.4 Request an encryption key, iv and DRM information for Fairplay DRM

```
POST /__cl/s:esf/__c/content/__op/hls-drm-fairplay/__f/index.m3u8 HTTP/1.1
Content-Type: application/json
Content-Length: ...
{
    "shared_secret": "shared-secret",
    "position": "0",
    "variants": [
        ...
    ]
}

HTTP/1.1 200 OK
Content-Type: application/json
Content-Length: ...
{
    "resource_id": "content",
    "position": "0",
    "encryption": "sample-aes",
    "content_id": "4da98ac7-a4a2-5c1a-92d9-d7343378b69f",
    "key": "hcClETJTqsiAW2Ks0NwFeg==",
    "key_id": "hpLAGRouSuGWLo33AUSMrg==",
    "iv": "AAAAAAAAAAAAAAAAAAAAAA==",
    "sample-aes": {
        "drm": "Test AES",
        "header_data": "skd://eyJMaWNlbnNl..."
    }
}
```

Header data with skd: scheme contains Fairplay key uri.

7.1.5 Request encryption keys for a time interval

```
POST /__cl/cg:live/__c/content/__op/dash-drm/__f/manifest.mpd HTTP/1.1
Content-Type: application/json
```

```
Content-Length: ...
{
    "shared_secret": "shared-secret",
    "position": [
        1766370975,
        1766371085
    ],
    "variants": [
        ...
    ]
}

HTTP/1.1 200 OK
Content-Type: application/json
Content-Length: ...
{
    "resource_id": "content",
    "position": [
        1766370975,
        1766371085
    ],
    "encryption": "cenc",
    "content_id": "d7e1918c-de83-52a1-a74a-e6fc3999e811",
    "key": "FmY0xnWCPCNaSpRG+tUuTQ==",
    "key_id": "nrQFDeRLSAKTLifXUIPiZg==",
    "cenc": [
        {
            "drm": "Axinom Widevine",
            "system_id": "edef8ba9-79d6-4ace-a3c8-27dcd51d21ed",
            "header_data": "AAAANHBzc2gAAAAA7e+
               ↪ LqXnWSs6jyCfc1R0h7QAAABQIARIQnrQFDeRLSAKTLifXUIPiZg=="
        },
        {
            "drm": "Playready",
            "system_id": "9a04f079-9840-4286-ab92-e65be0885f95",
            "header_data": "AAADdnBzc2g..."
        },
        {
            "drm": "ClearKey",
            "system_id": "1077efec-c0b2-4d02-ace3-3c1e52e2fb4b",
            "header_data": "
               ↪ AAAANHBzc2gBAAAAEHfv7MCyTQKs4zweUuL7SwAAAAGetAUN5EtIApMuJ9dQg
               ↪ +JmAAAAAA=="
        }
    ]
}
```

7.1.6 Request encryption keys for an open time interval (live window)

```
POST /__cl/cg:live/__c/content/__op/dash-drm/__f/manifest.mpd HTTP/1.1
Content-Type: application/json
Content-Length: ...
{
    "shared_secret": "shared-secret",
    "position": [
        1766368843
    ],
    "variants": [
        ...
    ]
}
```

```
HTTP/1.1 200 OK
Content-Type: application/json
Content-Length: ...
{
    "resource_id": "content",
    "position": [
        1766368843
    ],
    "encryption": "cenc",
    "content_id": "c71e94fb-049e-58b7-b7a5-0763fa657fa0",
    "key": "FmY0xnWCPCNaSpRG+tUuTQ==",
    "key_id": "nrQFDeRLSAKTLifXUIPiZg==",
    "cenc": [
        {
            "drm": "Axinom Widevine",
            "system_id": "edef8ba9-79d6-4ace-a3c8-27dcd51d21ed",
            "header_data": "AAAANHBzc2gAAAAA7e+
                ↪ LqXnWSs6jyCfc1R0h7QAAABQIARIQnrQFDeRLSAKTLifXUIPiZg=="
        },
        {
            "drm": "Playready",
            "system_id": "9a04f079-9840-4286-ab92-e65be0885f95",
            "header_data": "AAADdnBzc2g..."
        },
        {
            "drm": "ClearKey",
            "system_id": "1077efec-c0b2-4d02-ace3-3c1e52e2fb4b",
            "header_data": "
                ↪ AAAANHBzc2gBAAAAEHfv7MCyTQKs4zweUuL7SwAAAAGetAUN5EtIApMuJ9dQg
                ↪ +JmAAAAAA=="
        }
    ]
}
```

7.1.7 Request encryption keys for tracks in an asset

```
POST /__cl/s:esf/__c/content/__op/dash-drm-multikeys/__f/manifest.mpd HTTP
    ↪ /1.1
Content-Type: application/json
Content-Length: ...
{
    "shared_secret": "shared-secret",
    "position": "0",
    "variants": [
        {
            "name": "000001152790_1",
            "media_type": "video",
            "width": 854,
            "height": 480,
            "codec": "avc1.4d401f",
            "bitrate": 752000
        },
        {
            "name": "000001152791_1",
            "media_type": "video",
            "width": 1024,
            "height": 576,
            "codec": "avc1.640029",
            "bitrate": 904000
        },
```

```
        {
            "name": "000001152792_1",
            "media_type": "video",
            "width": 1280,
            "height": 720,
            "codec": "avc1.640029",
            "bitrate": 1404000
        },
        {
            "name": "000001152795_1",
            "media_type": "video",
            "width": 1920,
            "height": 1080,
            "codec": "avc1.640029",
            "bitrate": 3872000
        },
        {
            "name": "000001152805_2",
            "media_type": "audio",
            "codec": "mp4a.40.2",
            "bitrate": 96000
        },
        {
            "name": "271648_0",
            "media_type": "text",
            "codec": "wvtt",
            "bitrate": 1400
        }
    ]
}

HTTP/1.1 200 OK
Content-Type: application/json
Content-Length: ...
{
    "resource_id": "content",
    "position": "0",
    "encryption": "cenc",
    "content_id": "4da98ac7-a4a2-5c1a-92d9-d7343378b69f",
    "key_info": [
        {
            "key": "+5f6+rqidg+YaZG/0IyQcA==",
            "key_id": "NxKm0GF+Q8q2VTR1oqyRNQ==",
            "cenc": {
                "drm": "Axinom Multiple Keys",
                "system_id": "edef8ba9-79d6-4ace-a3c8-27dcd51d21ed",
                "header_data": "AAAANHBzc2gAAAAA7e+
                    ↪ LqXnWSs6jyCfc1R0h7QAAABQIARIQNxKm0GF+Q8q2VTR1oqyRNQ=="
            },
            "variants": [
                "000001152790_1"
            ]
        },
        {
            "key": "GnN0tBgC2M7a0eD6iIXv+A==",
            "key_id": "ECiOkEeeR4Wf0GSXx21Vdw==",
            "cenc": {
                "drm": "Axinom Multiple Keys",
                "system_id": "edef8ba9-79d6-4ace-a3c8-27dcd51d21ed",
                "header_data": "AAAANHBzc2gAAAAA7e+
                    ↪ LqXnWSs6jyCfc1R0h7QAAABQIARIQECiOkEeeR4Wf0GSXx21Vdw=="
            },
```

```
            "variants": [
                "000001152791_1"
            ]
        },
        {
            "key": "/lAasS+wBU1gDBmKhG6OPA==",
            "key_id": "wrIh6BP+TeqEsJMx4RI1SQ==",
            "cenc": {
                "drm": "Axinom Multiple Keys",
                "system_id": "edef8ba9-79d6-4ace-a3c8-27dcd51d21ed",
                "header_data": "AAAANHBzc2gAAAAA7e+
                    ↪ LqXnWSs6jyCfc1R0h7QAAABQIARIQwrIh6BP+TeqEsJMx4RI1SQ=="
            },
            "variants": [
                "000001152792_1"
            ]
        },
        {
            "key": "WubwO/iIMZ+LcK+7OoQu6A==",
            "key_id": "Iv5Uh382Su6gKPA+w1qBlQ==",
            "cenc": {
                "drm": "Axinom Multiple Keys",
                "system_id": "edef8ba9-79d6-4ace-a3c8-27dcd51d21ed",
                "header_data": "AAAANHBzc2gAAAAA7e+
                    ↪ LqXnWSs6jyCfc1R0h7QAAABQIARIQIv5Uh382Su6gKPA+w1qBlQ=="
            },
            "variants": [
                "000001152795_1"
            ]
        },
        {
            "key": "a3gnC/B6lKmjiFNyzC4n4Q==",
            "key_id": "d+FtF72rTxmAe2WBvVOEHQ==",
            "cenc": {
                "drm": "Axinom Multiple Keys",
                "system_id": "edef8ba9-79d6-4ace-a3c8-27dcd51d21ed",
                "header_data": "AAAANHBzc2gAAAAA7e+
                    ↪ LqXnWSs6jyCfc1R0h7QAAABQIARIQd+FtF72rTxmAe2WBvVOEHQ=="
            },
            "variants": [
                "000001152805_2"
            ]
        },
        {
            "plaintext": true,
            "variants": [
                "271648_0"
            ]
        }
    ]
}
```

7.1.8 Request encryption keys with key rotation

```
POST /__cl/s:esf/__c/content/__op/dash-drm-rotation/__f/manifest.mpd HTTP/1.1
Content-Type: application/json
Content-Length: ...
{
    "shared_secret": "shared-secret",
    "position": [
        1766375672
```

```
    ],
    "variants": [
        ...
    ]
}

HTTP/1.1 200 OK
Content-Type: application/json
Content-Length: ...
{
    "resource_id": "content",
    "position": [
        1766375672
    ],
    "encryption": "cenc",
    "content_id": "d7e1918c-de83-52a1-a74a-e6fc3999e811",
    "time_to_next_poll": 53,
    "key_info": [
        {
            "key": "/lAasS+wBU1gDBmKhG6OPA==",
            "key_id": "wrIh6BP+TeqEsJMx4RI1SQ==",
            "cenc": {
                "drm": "Axinom Rotation",
                "system_id": "edef8ba9-79d6-4ace-a3c8-27dcd51d21ed",
                "header_data": "AAAANHBzc2gAAAAA7e+
                    ↪ LqXnWSs6jyCfc1R0h7QAAABQIARIQwrIh6BP+TeqEsJMx4RI1SQ=="
            },
            "start_time": 1766375640,
            "end_time": 1766375700
        },
        {
            "key": "WubwO/iIMZ+LcK+7OoQu6A==",
            "key_id": "Iv5Uh382Su6gKPA+w1qBlQ==",
            "cenc": {
                "drm": "Axinom Rotation",
                "system_id": "edef8ba9-79d6-4ace-a3c8-27dcd51d21ed",
                "header_data": "AAAANHBzc2gAAAAA7e+
                    ↪ LqXnWSs6jyCfc1R0h7QAAABQIARIQIv5Uh382Su6gKPA+w1qBlQ=="
            },
            "start_time": 1766375700,
            "end_time": 1766375760
        },
        {
            "key": "a3gnC/B6lKmjiFNyzC4n4Q==",
            "key_id": "d+FtF72rTxmAe2WBvVOEHQ==",
            "cenc": {
                "drm": "Axinom Rotation",
                "system_id": "edef8ba9-79d6-4ace-a3c8-27dcd51d21ed",
                "header_data": "AAAANHBzc2gAAAAA7e+
                    ↪ LqXnWSs6jyCfc1R0h7QAAABQIARIQd+FtF72rTxmAe2WBvVOEHQ=="
            },
            "start_time": 1766375760,
            "end_time": 1766375820
        },
        {
            "key": "ce29zFawfKfRheNlbokcbA==",
            "key_id": "sUYQ5SzYSUugipBHBBbVOA==",
            "cenc": {
                "drm": "Axinom Rotation",
                "system_id": "edef8ba9-79d6-4ace-a3c8-27dcd51d21ed",
                "header_data": "AAAANHBzc2gAAAAA7e+
                    ↪ LqXnWSs6jyCfc1R0h7QAAABQIARIQsUYQ5SzYSUugipBHBBbVOA=="
```

```
            },
            "start_time": 1766375820,
            "end_time": 1766375880
        },
        {
            "key": "FqcpMo2M/H0XRqViJp6e8Q==",
            "key_id": "iskPK7gBQZG9DUL015lm8g==",
            "cenc": {
                "drm": "Axinom Rotation",
                "system_id": "edef8ba9-79d6-4ace-a3c8-27dcd51d21ed",
                "header_data": "AAAANHBzc2gAAAAA7e+
                    ↪ LqXnWSs6jyCfc1R0h7QAAABQIARIQiskPK7gBQZG9DUL015lm8g=="
            },
            "start_time": 1766375880,
            "end_time": 1766375940
        }
    ]
}
```

It is recommended to return at least the keys for the current and upcoming intervals and to set `time_to_next_poll` to the time remaining in the current interval. By having the key for the upcoming interval available some downtime of the eDRM server can be tolerated.