ru_vds
14 часов назад

Как защищают фильмы и доставляют их в кинотеатры

Средний 12 мин 3.2K

Блог компании RUVDS.com, Информационная безопасность*, Криптография*, Работа с видео*, Управление ме

Обзор Перевод

Автор оригинала: Christian

У кинематографической индустрии есть собственные стандарты для защищённого создания и распространения фильмов. Всё необходимое, от форматов файлов и шифрования до проекционных систем, определяется в спецификации DCI (Digital Cinema Initiatives).

Сама спецификация доступна публично, но связана с различными стандартами IEEE

(Institute of Electrical and Electronics Engineers) и SMPTE (Society of Motion Picture and Television Engineers), которые необходимо приобретать за деньги.

В этом посте мы опишем примерный процесс реализации DCI и подробно расскажем, как работает шифрование DCI-фильма. Мы **не** будем рассказывать, как взламывать шифрование; к тому же, на момент написания поста никакое шифрование взломано ещё не было. **С нашей точки зрения, стандарт DCI хорошо защищён**.

Автор поста с 2021 года работает в кинотеатре, ничего не зная о процессах распространения и производства в этой области. Часть информации может быть неполной.

Как всё начиналось

В конце 2023 года был выпущен фильм «Вонка». Некоторые кинотеатры сообщали о том, что не могут запустить эту ленту на своём проекторе.

Причина этого заключалась в истёкшем сроке действия сертификата, используемого дистрибьютором. Этот сертификат использовался для подписывания файлов DCP.

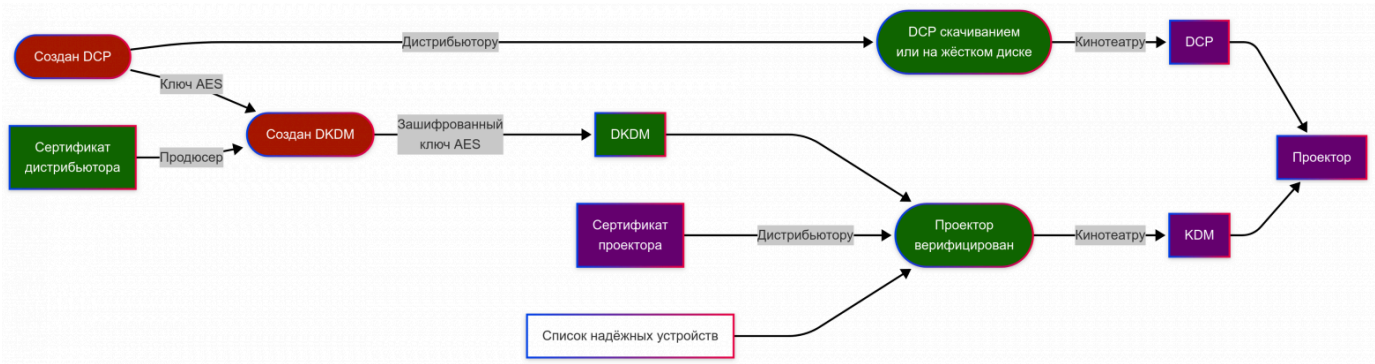
Дистрибьютор опубликовал новые файлы, и фильм начал воспроизводиться.

Автор поста из любопытства начал изучать работу процесса валидации в проекторе.

Глоссарий

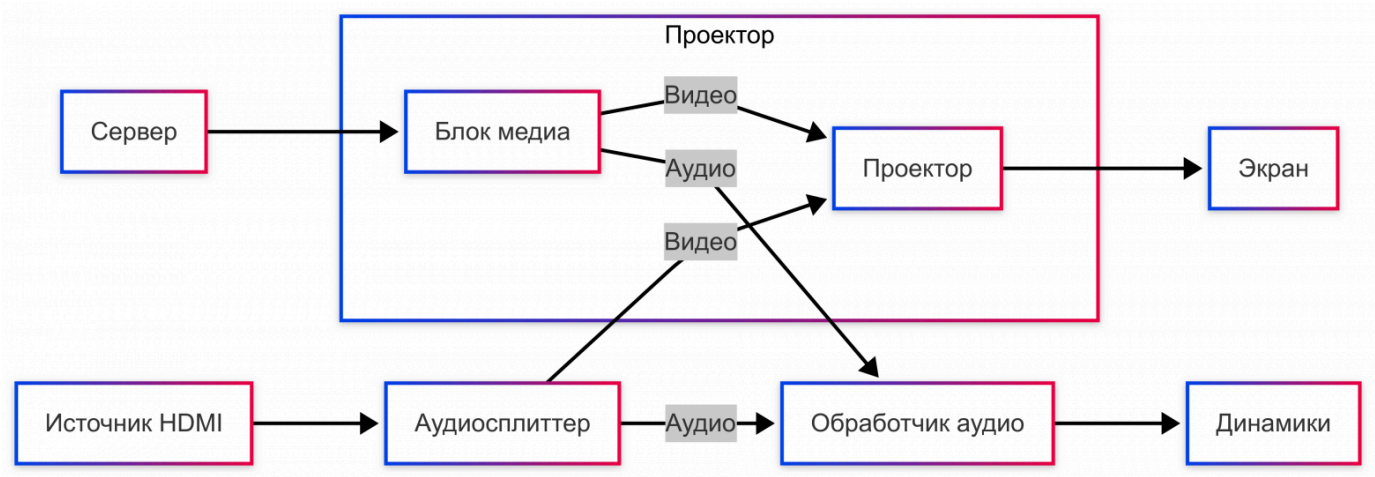
- **DCP**: Digital Cinema Package — папка, в которой содержатся все компоненты фильма: метаданные, субтитры, звук и изображение разбиты на отдельные файлы.
- **CPL**: Composition Playlist — DCP может содержать несколько потоков аудио и видео, которые объединяются в CPL.
- **KDM**: Key Delivery Message — файл XML, содержащий криптографическую информацию, позволяющую воспроизводить фильм на конкретной проекционной системе, сертифицированной по DCI.
- **DKDM**: Distribution Key Delivery Message — похож на KDM, но для системы ремастеринга или дистрибьюции, а не для проекционной системы.

Процесс распространения



Проекционные системы

Большинство проекционных систем состоит из сервера, обработчика аудио и проектора.



На сервере хранятся DCP и KDM, он управляет плейлистами и оборудованием проектора и обработчика аудио.

Многие кинотеатры подключают к проектору и свои системы автоматизации зала (свет, кулисы и так далее). Благодаря вводу в плейлисты команд с указанием времени можно автоматически управлять светом и кулисами экрана. У сервера проектора для этого есть реле на 12 В/24 В.

Эти временные команды также используются для задания нужного соотношения сторон (Aspect Ratio) и громкости для разных DCP из плейлиста.

Сервером можно управлять удалённо через PC, а в больших кинотеатрах — через Theatre Management System.

DCP импортируются с USB/жёстких дисков CRU или скачиваются с сервера через Интернет. На всех этапах они хранятся в зашифрованном виде.

У проектора есть так называемый «блок медиа» (Media Block), занимающийся DRM и

расшифровкой. Он получает данные DCP и KDM для расшифровки каждого кадра в реальном времени.

При воспроизведении DCP проектор отправляет зашифрованное аудио PCM обработчику аудио, который затем отправляет каждый обработанный аудиоканал соответствующим динамикам.

Формат DCP

DCP — это папка, в которой хранятся файлы метаданных XML и множество файлов MXF самого фильма.

Шаблон именования папок

```
AwesomeMovie_FTR-2_S_DE-XX_DE-16_51_4K_20240119_SMPTE_OV
```

- `AwesomeMovie` : краткая версия названия фильма.
- `FTR` : тип медиа, в данном случае «Feature».
- `2` : номер версии.

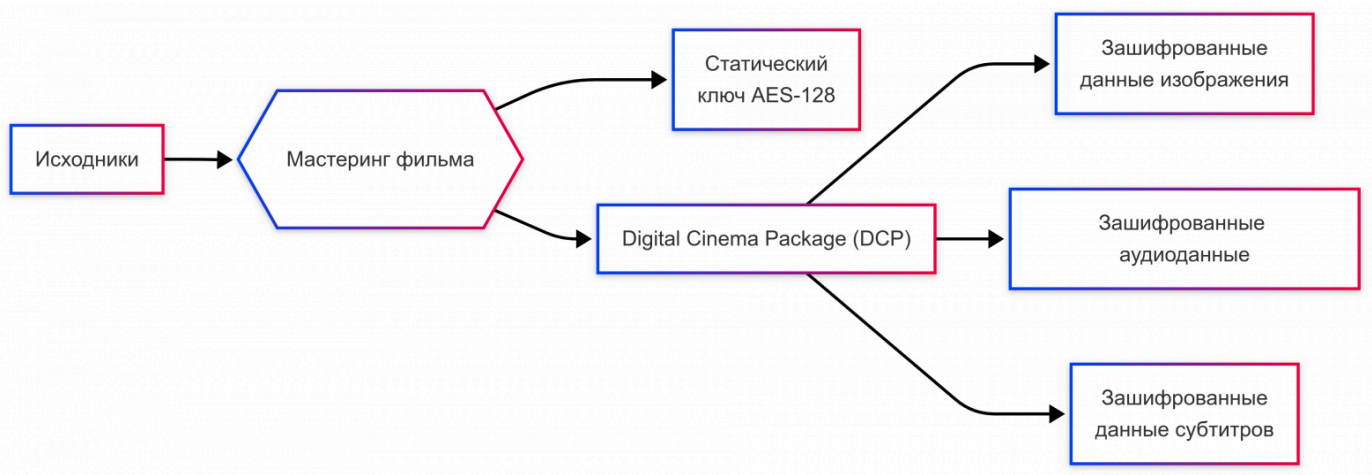
- `S` : соотношение сторон, в данном случае 2.35:1 (Sope).
- `DE` : язык аудио.
- `XX` : язык субтитров, в данном случае субтитры отсутствуют.
- `DE` : территория.
- `16` : возрастной рейтинг.
- `51` : аудиоканалы, в данном случае 5.1 surround sound.
- `4K` : разрешение фильма, в данном случае 4096x1716 пикселей.
- `20240119` : метка времени мастеринга.
- `SMPTE` : стандарт DCP; существует ещё и `Interop` .
- `OV` : тип пакета, оригинальная версия (original version) или `VF` — файл версии (version file).

Источник информации: http://static.kinofreund.com/dcnt/

Процесс мастеринга

В процессе мастеринга генерируется статический 128-битный ключ AES, а исходный носитель преобразуется в файлы MXF, один для изображения, второй для звука.

Для мастеринга может использоваться инструмент DCP-o-matic. Существуют и коммерческие продукты.



Видеопоток кодируется как единое изображение JPEG2000 на каждый кадр. Каждый кадр кодируется тем же самым статическим ключом AES.

Аудиопоток (скорее всего) объединяется в один поток BWF (Broadcast Wave Format) на каждый кадр и тоже шифруется отдельно. (Пока мне не удалось найти об этом никакой информации.)

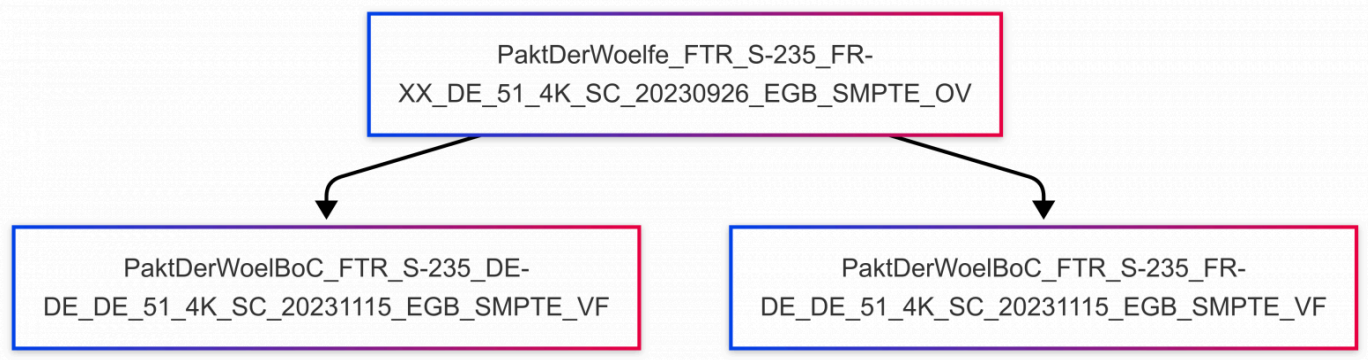
DCP может иметь размер до 200 ГБ и даже больше. Некоторые новые релизы могут доходить до терабайта, если на одном жёстком диске поставляется несколько версий (разные языки, субтитры, 2D/3D).

Субтитры поставляются в файле XML или записываются непосредственно в кадры картины. Если они поставляются в виде файла XML, то проектор рендерит субтитры при помощи файла шрифта TTF.

```
<?xml version="1.0" encoding="UTF-8"?>
<DCSubtitle Version="1.0">
    <SubtitleID>xxxxxxxx-xxxx-xxxx-xxxx-xxxxxxxxxxxx</SubtitleID>
    <MovieTitle>MovieTitle</MovieTitle>
    <ReelNumber>1</ReelNumber>
    <Language>de</Language>
    <LoadFont Id="Font1" URI="Arial.ttf" />
    <Font Id="Font1" Color="ffffffff" Effect="border" EffectColor="ff000000" Italic="nc
        <Subtitle SpotNumber="1" TimeIn="00:07:45:094" TimeOut="00:07:48:000" FadeUpTir
            <Text HPosition="0" VAlign="bottom" VPosition="7">WÜSTE VON NEVADA - HEUTE<
        </Subtitle>
    </Font>
</DCSubtitle>
```

Вспомогательный DCP (VF, Version File)

Вспомогательный DCP позволяет использовать ту же оригинальную картинку DCP (OV, Original File) и заменить аудиодорожки или субтитры. Это обеспечивает поддержку нескольких языков без необходимости отправки в кинотеатры сотен гигабайтов дублированных видеоданных.



Пример: один файл версии для немецкого с немецкими субтитрами и одна для французского с немецкими субтитрами.

Можно даже отправлять разные версии монтажа для разных регионов.

Программное обеспечение

ПО — это две опенсорсных реализаций на C++, обе активно развиваются, одна создаётся самой киноотраслью!

cth103/libdcp — это библиотека C++, написанная создателями DCP-o-matic:

- https://git.carlh.net/gitweb/?p=dcpomatic.git;a=shortlog
- https://dcpomatic.com/ (исходный код)

asdcplib — библиотека C++, написанная компаниями, активно работающими в сфере кинематографической индустрии:

> Проект asdcplib изначально передавался по FTP. В 2005-2008 годах проект находился на SourceForge, когда он перешёл на распространение только релизов через CineCert. На декабрь 2019 года его новым домом стал Github.
>
> — https://github.com/cinecert/asdcplib

Почти всё ПО, как коммерческое, так и опенсорсное, использует какую-то из этих двух библиотек.

Распространение

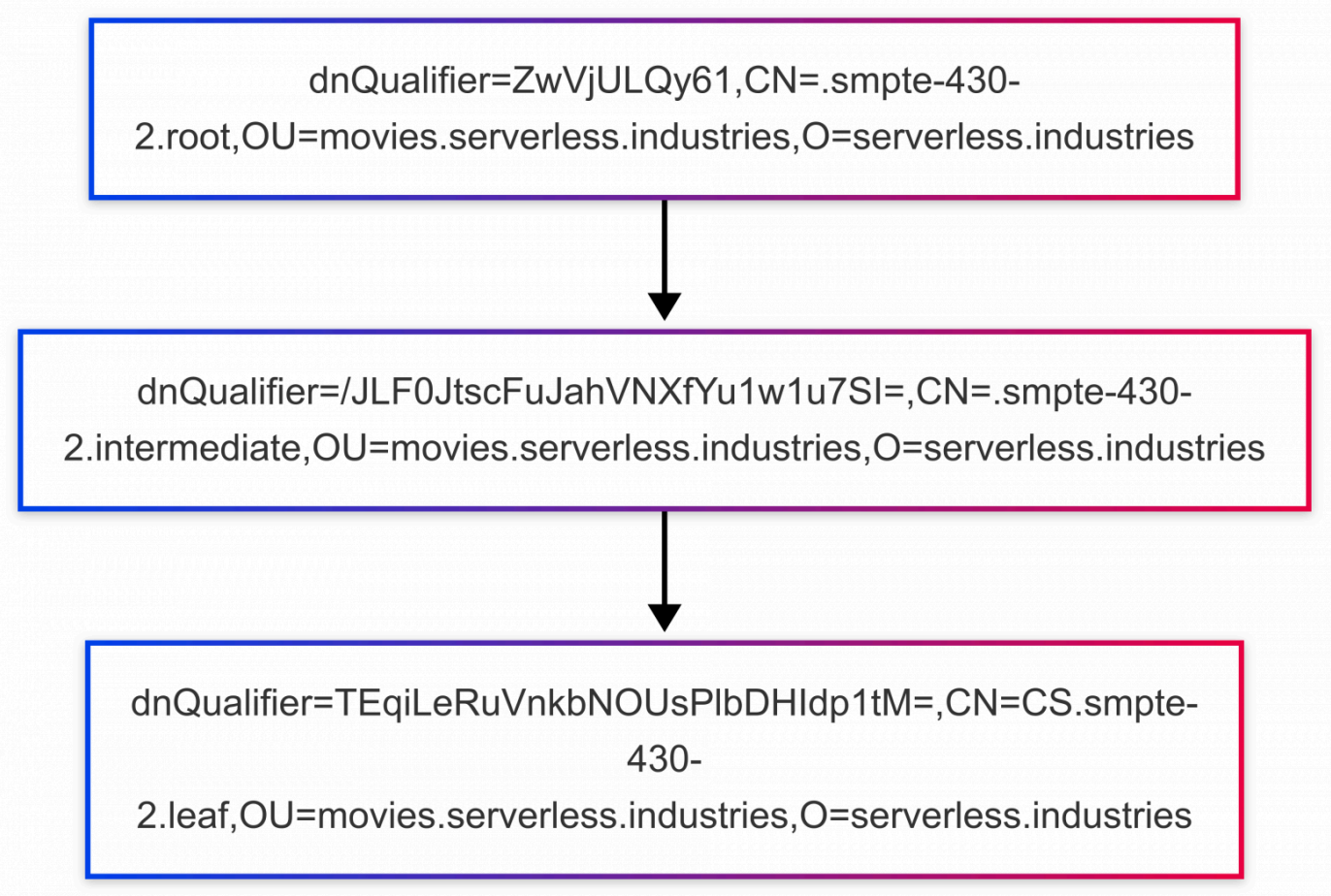
Симметричный ключ шифрования DCP как-то нужно защищать. Для этого компания-продюсер создаёт файл XML DKDM, содержащий ключ AES, зашифрованный публичным ключом сертификата (Certificate Public Key) дистрибьютора.

Дистрибьютор может использовать этот DKDM, чтобы создавать файлы XML KDM для кинотеатров. Для этого он расшифровывает ключ AES DCP и шифрует его снова публичным ключом сертификата целевой проекционной системы.

Цепочки сертификатов

И проекторы, и дистрибьюторы используют для подписывания и шифрования/расшифровки данных цепочки сертификатов SSL.

Цепочка всегда состоит Root Certificate Authority (CA), Intermediate CA и Leaf Certificate.



Поле `dnQualifier` — это цифровая сигнатура публичного ключа сгенерированного сертификата:

```
cat leaf.pem | openssl x509 -pubkey -noout | \
    openssl base64 -d | dd bs=1 skip=24 2>/dev/null | \
    openssl sha1 -binary | openssl base64
```

Скрипт создания цепочки сертификатов: create-smpte-chain.sh

Key Delivery Message

KDM — это файл XML, содержащий ключ AES DCP, зашифрованный публичным ключом проекторов.

- Различные схемы XML: SMPTE 430-1, SMPTE 430-3.
- Подписывание документа XML-DSig.
- Один или несколько элементов `<enc:CipherValue>` содержат ключи AES и метаданные для множественных DCP OV/VF; и те, и другие зашифрованы.

Расшифровка `<enc:CipherValue>` с помощью `openssl`:

```
cat KDM_KaizoTrap_FTR-1_F_XX-XX_20_2K_20240119_SMPTE_OV_My_nonexistent_cinema_Nonexiste
    xq -r '.DCinemaSecurityMessage.AuthenticatedPrivate."enc:EncryptedKey"[0]."enc:Ciph
    base64 -d | \
    openssl pkeyutl -decrypt -inkey leaf.key -pkeyopt rsa_padding_mode:oaep > kdminfo.b
```

Вывод `hexdump -C kdminfo.bin`:

```
00000000  f1 dc 12 44 60 16 9a 0e  85 bc 30 06 42 f8 66 ab  |...D`.....0.B.f.|
00000010  28 fd 80 bf a9 bb f1 dc  48 d9 87 e9 5c c9 a5 41  |(.......H...\..A|
00000020  ab 8e 14 82 88 40 c3 90  2a 0b 46 21 b4 10 49 6b  |.....@..*.F!..Ik|
00000030  b8 a4 a9 4f 4d 44 41 4b  2e 2e 05 b5 bb ed 45 79  |...OMDAK......Ey|
00000040  96 cc 9c d6 00 ed db 4c  32 30 32 34 2d 30 31 2d  |.......L2024-01-|
00000050  32 38 54 31 39 3a 30 30  3a 30 30 2b 30 31 3a 30  |28T19:00:00+01:0|
00000060  30 32 30 32 34 2d 30 32  2d 30 34 54 32 30 3a 30  |02024-02-04T20:0|
00000070  30 3a 30 30 2b 30 31 3a  30 30 4c 3a b6 ed 71 eb  |0:00+01:00L:..q.|
00000080  29 25 90 48 6b 4f 96 2f  44 f6                    |)%.HkO./D.|
0000008a
```

Далее при помощи `dd` файл можно нарезать на отдельные поля:

```
# закодированная в base64 сигнатура
dd if=kdminfo.bin bs=1 skip=16 count=20 status=none | base64

# Ключ AES в шестнадцатеричном виде
dd if=kdminfo.bin bs=1 skip=122 count=16 status=none | hexdump -C

# строка даты в простом тексте ASCII
dd if=kdminfo.bin bs=1 skip=97 count=25 status=none; echo
```

Формат:

Начало	Длина	Описание
0	16	ID структуры, двоичный, статический `0xf1dc124460169a0e85bc300642f866ab`

Начало	Длина	Описание
16	20	Сигнатура подписавшей сертификат стороны, двоичная, `KP2Av6m78dxI2YfpXMmlQauOFII=`
36	16	Composition Playlist UUID, `8840c390-2a0b-4621-b410-496bb8a4a94f`
52	4	Тип ключа, ASCII, `MDAK` = Main Sound
56	16	UUID ключа, `2e2e05b5-bbed-4579-96cc-9cd600eddb4c`
72	25	Строка даты, до которой сертификат невалиден, ASCII, `2024-01-28T19:00:00+01:00`
97	25	Строка даты, после которой сертификат невалиден, ASCII, `2024-02-04T20:00:00+01:00`
122	16	Ключ дешифровки AES, двоичный, `0x4c3ab6ed71eb292590486b4f962f44f6`

- Validate signer certificate: должен соответствовать сигнатуре leaf certificate, используемого для подписания XML-структуры KDM.

```
openssl asn1parse -in leaf.pem -noout -strparse 4 -out - | \
    openssl dgst -sha1 -binary | \
    openssl base64
```

- Validate Composition Playlist UUID: должен быть указан в XML-поле KDM `<CompositionPlaylistId>` .
- Validate Key UUID: должен быть указан в XML-поле KDM `<KeyId>` .
- Validate Key Type: должен совпадать с типом соответствующего Key UUID `<TypedKeyId>` block.
- Validate Dates: должны быть указаны в XML-полях KDM `<ContentKeysNotValidBefore>` и `<ContentKeysNotValidAfter>` .

(Это подмножество проверок задокументировано в DCI Compliance Test Plan)

У самого проектора нет доверенного хранилища CA для валидации KDM и DCP. Весь

процесс зависит от сигнатуры, хранящейся в `<enc:CipherValue>`, и от того, что кинотеатр-получатель не имеет доступа к Projectors Private Key.

Список доверенных устройств

Можно ли самостоятельно изготовить проектор DCI/DCP?

И да, и нет.

ПО для этого существует: теоретически, требуются только PC Linux и любая проекционная/звуковая система.

Дистрибьюторы используют так называемый «Trusted Device List», предоставляемый сертифицированными по DCI производителями проекционных систем. Отсутствующие в этом списке проекторы не получают DCP/KDM от многих дистрибьюторов.

Кроме того, сертифицированная по DCI проекционная система должна устанавливаться авторизованной компанией.

- Запрос KDM по серийному номеру: многие дистрибьюторы просто просят указать модель и серийный номер проектора, после чего они могут запросить в своей системе дистрибьюции сертификат проекционной системы.
- Запрос KDM по сертификату: кинотеатр должен предоставить сертификат своего проектора, который валидируется с корневым сертификатом производителя проекционных систем.

Если кинотеатр «не играет по правилам», то проекционная система стоимостью от 30 тысяч евро превратится в дорогостоящий кирпич.

Формат файлов MXF

MXF используется всей индустрией кино и широковещания.

Стандарт SMPTE по нему тоже можно приобрести только за деньги.

Так как стандарт довольно сложен, я для удобства воспользовался MXFInspect и парсил только один кадр/триплет.

Кодирование BER

Basic **E**ncoding **R**ules:

- Если первый бит байта равен `1`, это означает использование BER.
- Биты 2-8 содержат количество байтов, используемых для длины.
- Байты длины содержат длину данных.

Индикатор BER	Длина	Содержимое
`0x83 = 10000011b` = 3 байта на длину	`0x004F0C` = `20236d`	20236 байтов данных

Формат триплета (одного кадра фильма)

Структура триплета кадра

Начало	Тип	Длина	Описание
0	DAT	16	Ключ зашифрованного триплета
16	BER	1	BER `0x83 = 10000011b` = следующее поле имеет длину 3 байта
17	LEN	3	`0x004F0C = 20236d` = длина данных
20	BER	1	BER `0x83 = 10000011b` = следующее поле имеет длину 3 байта
21	LEN	3	`0x000010 = 16d` = длина следующего поля
24	DAT	16	Ссылка на криптографический контекст
40	BER	1	BER `0x83 = 10000011b` = следующее поле имеет длину 3 байта
41	LEN	3	`0x000008 = 8d` = длина следующего поля
44	DAT	8	Смещение в текстовом виде

Начало	Тип	Длина	Описание
52	BER	1	BER `0x83 = 10000011b` = следующее поле имеет длину 3 байта
53	LEN	3	`0x000010 = 16d` = длина следующего поля
56	DAT	16	Ключ исходников
72	BER	1	BER `0x83 = 10000011b` = следующее поле имеет длину 3 байта
73	LEN	3	`0x000008 = 8d` = длина следующего поля
76	DAT	8	Длина исходников
84	BER	1	BER `0x83 = 10000011b` = следующее поле имеет длину 3 байта
85	LEN	3	`0x004e90 = 20112d` = длина следующего поля
88	DAT	20112	Зашифрованное значение исходников

Нарезаем данные

```
# вырезаем первый кадр из MXF bodypartition
# смещения взяты из MXFInspect
dd if=file.mxf bs=1 count=20256 skip=16524 of=block.bin

# вырезаем данные из кадра
dd if=block.bin skip=88 count=20112 bs=1 of=block-encrypted-data.bin
```

Дешифруем данные кадра

Ключ шифрования DCP: `4c3ab6ed71eb292590486b4f962f44f6`

IV кадра

В каждом кадре используется уникальный IV (Initialization Vector, вектор инициализации), гарантирующий, что блочное шифрование AES всегда будет генерировать разные шифрованные тексты, что усложнит брутфорс. Это работает аналогично парольной соли.

```
# данные начинаются с 16 байтов iv и 16 байтов cv
# байты iv AES
```

```
dd if=block-encrypted-data.bin bs=1 count=16 of=iv.bin
```

Вывод `hexdump -C iv.bin` :

```
00000000  16 6e 7b d1 67 81 44 2e  7a ca de 3c 46 cc d7 39  |.n{.g.D.z..<F..9|
00000010
```

Чтобы понять, почему IV так важен, загуглите «ECB Penguin».

Валидируем CV

Если содержимое данных неизвестно, мы не можем никак узнать, была ли расшифровка успешной; поэтому используется CV (Check Value, контрольное значение). Check Value зашифровано тем же ключом AES + IV, но его текстовое содержимое известно заранее.

```
# данные начинаются с 16 байтов iv и 16 байтов cv
# байты cv AES
dd if=block-encrypted-data.bin bs=1 count=16 skip=16 of=cv.bin
```

```
cat cv.bin | openssl enc -aes128 -d \
    -K 4c3ab6ed71eb292590486b4f962f44f6 \
    -iv 166e7bd16781442e7acade3c46ccd739 \
    -nosalt -nopad | hexdump -C
```

```
00000000  43 48 55 4b 43 48 55 4b  43 48 55 4b 43 48 55 4b  |CHUKCHUKCHUKCHUK|
00000010
```

Если результат равен `0x4348554B4348554B4348554B4348554B` , то ключ правильный.

Дешифруем данные

```
# Удаляем IV и CV из зашифрованного блока данных
dd bs=1 skip=32 \
    if=block-encrypted-data.bin \
    of=block-encrypted-data-nocryptinfo.bin
```

```
# Дешифруем блок данных
cat block-encrypted-data-nocryptinfo.bin | \
    openssl enc -aes128 -d \
        -K 4c3ab6ed71eb292590486b4f962f44f6 \
        -iv 166e7bd16781442e7acade3c46ccd739 \
        -nosalt -nopad > block-decrypted-data.bin
```

MXF, созданные программой DCP-o-Matic, содержат в hexdump `libdcp` ; это означает, что расшифровка была успешной.

Вывод `hexdump -C block-decrypted-data.bin | head` :

```
00000000  3a f0 b7 f5 53 49 eb b7  c0 c0 cb a5 c9 2f 35 19  |:...SI......./5.|
00000010  00 00 00 00 00 00 00 00  00 00 07 ce 00 00 04 38  |...............8|
00000020  00 00 00 00 00 00 00 00  00 03 0b 01 01 0b 01 01  |................|
00000030  0b 01 01 ff 52 00 12 01  04 00 01 01 05 03 03 00  |....R...........|
00000040  00 77 88 88 88 88 88 ff  5c 00 23 22 97 20 96 f0  |.w......\.#". ..|
00000050  96 f0 96 c0 8f 00 8f 00  8e e0 87 50 87 50 87 68  |...........P.P.h|
00000060  70 05 70 05 70 47 77 d3  77 d3 77 62 ff 55 00 13  |p.p.pGw.w.wb.U..|
00000070  00 50 00 00 00 4d 40 00  00 00 00 49 00 00 00 00  |.P...M@....I....|
00000080  49 ff 64 00 0a 00 01 6c  69 62 64 63 70 ff 90 00  |I.d....libdcp...|
00000090  0a 00 00 00 00 4d 40 00  03 ff 93 ef fe 2c 71 ff  |.....M@......,q.|
```

Магические сигнатуры

Сначала расшифрованный кадр невозможно открыть в GIMP.

Чтобы программы могли корректно открывать файл, в начале многих файлов содержатся так называемые «магические сигнатуры».

Это магическое значение в `block-decrypted-data.bin` по неизвестным причинам некорректно. При извлечении кадра JPEG2000 из незашифрованного MXF сигнатура корректна. В качестве простого хака я просто скопировал из незашифрованного кадра MXF в дешифрованный:

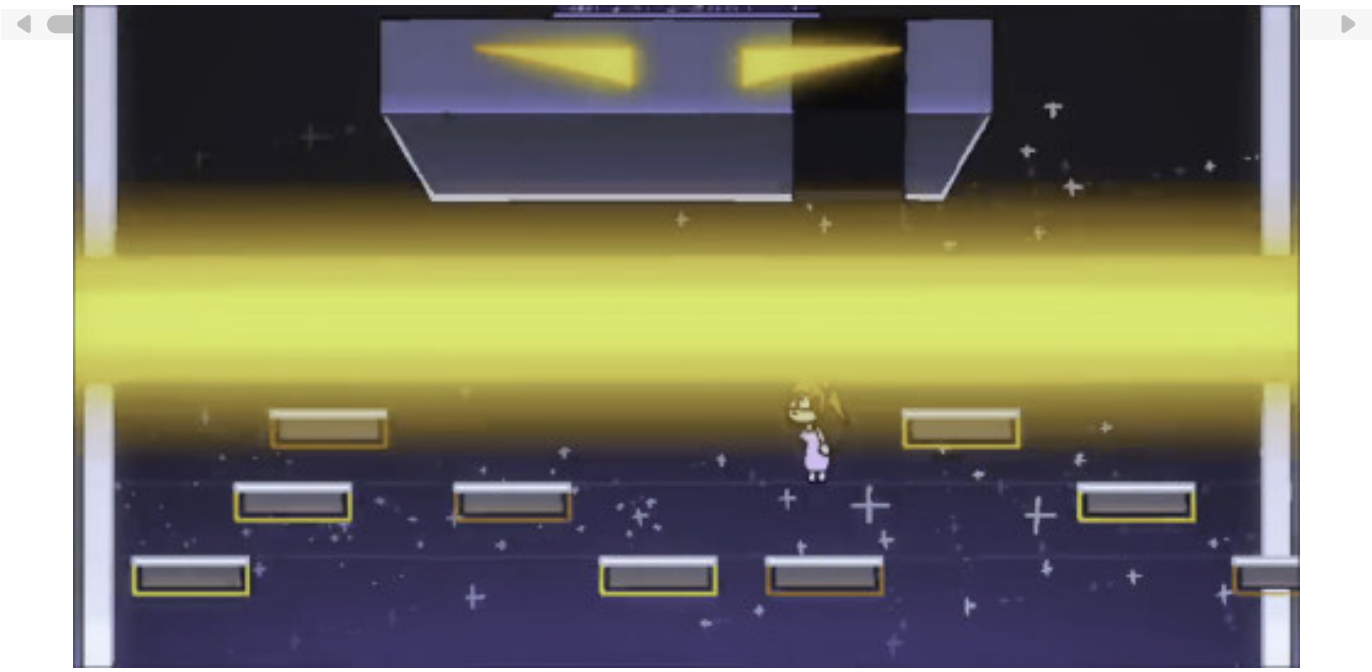
```
# снова смещение из MXFInspect
dd if=j2c_caa6b39e-bd57-4676-9797-112e96a6f0c3.mxf bs=1 skip=16524 count=20085 of=frame

# незашифрованный кадр не содержит криптографической информации,
# поэтому нам просто нужно нарезать первый заголовок BER
```

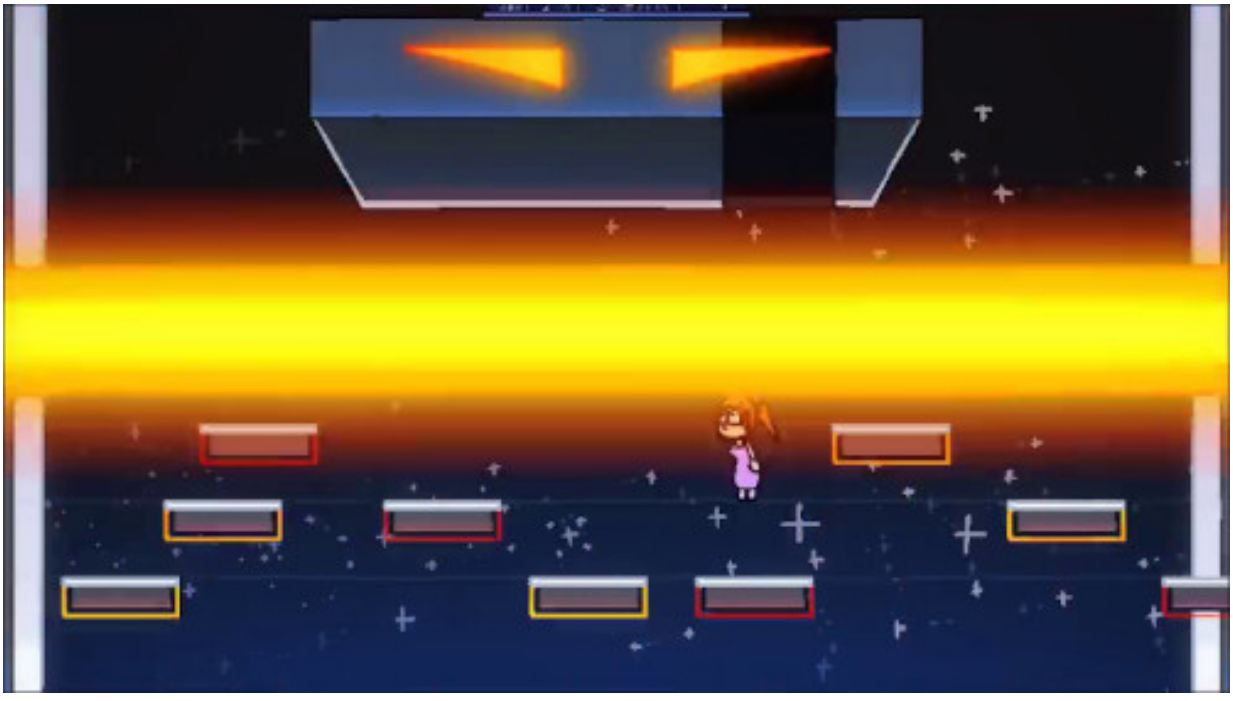
```
dd if=framedata.bin.j2k bs=1 skip=20 count=16 of=j2k_header.bin

# возвращаемся к расшифрованному кадру mxf
# заменяем сигнатуру
cp j2k_header.bin frame.j2k
dd if=block-decrypted-data.bin bs=1 skip=16 >> image.j2k
```

Теперь файл должен открываться в GIMP. Редактор GIMP отображает кадр в неправильных цветах, потому что в файлах MXF используется собственное цветовое пространство.



Вот оригинальный кадр с правильными цветами:



Источник: Kaizo Trap by Guy Collins Animation

Почему это безопасно

Вся система полагается на то, что ключи AES никогда не используются повторно, а также на применение защищённых приватных ключей.

- Данные KDM зашифрованы 2048-битным ключом RSA.
- Данные MXF зашифрованы AES-128.
- Ключи расшифровки всегда хранятся в оборудовании типа TPM.
- Каждый DCP использует уникальные ключи шифрования.
- У каждого проектора свои уникальные сертификаты/ключи.
- Требуется проектор, сертифицированный по DCI.
- Дистрибьюторы могут верифицировать принадлежность сертификата к сертифицированному по DCI проектору.

Можно же просто всё записать

В зашифрованных DCP используются водяные знаки Forensic Watermarks, содержащие серийный номер проекционной системы, поэтому если записанная копия фильма появится онлайн, то кинотеатру придётся отвечать на очень серьёзные вопросы, и он больше никогда не получит фильмов.

Спиратенные копии фильмов, скорее всего, берутся из других источников.

Как всё продвигается

DCI выпустил версию 1.4.4 спецификации, которая теперь позволяет воспроизводить DCP с устаревшим сертификатом подписывающей стороны.

Производители оборудования уже начали работу над обновлением ПО.

Источники

- DCI Specification
- DCI Compliance Test Plan
- OpenSSL Foo
- Key Types
- DC Tools by WolfgangW
- MXFInspect
- MXF Triplets
- SMPTE Standard
- MXF Triplet Encryption
- Showing jpeg2000 in Gimp / parsing jpeg2000 data from mxf
- List of magic signatures
- Forensic Watermarks in DCPs

Теги: пиратство, шифрование, интеллектуальная собственность, кинотеатры, кинопиратство, ruvds_перевод

Хабы: Блог компании RUVDS.com, Информационная безопасность, Криптография, Работа с видео, Управление медиа

584 Карма 293.3 Рейтинг

@ru_vds
Пользователь

Подписаться

Комментарии 6

Публикации

ЛУЧШИЕ ЗА СУТКИ ПОХОЖИЕ

Maximov_psy
12 часов назад

Что я узнал, проконсультировав 100 айтишников

12 мин 8.6K

+47 49 53

ru_vds
14 часов назад

Как защищают фильмы и доставляют их в кинотеатры

Средний 12 мин 3.2K

Обзор Перевод

+36 30 6

MrSotnik
17 часов назад

Новый язык от 1С: Зачем? Кому? Стоит ли лезть?

5 мин 22K

+36 34 76

RationalAnswer
22 часа назад

Тотальный блэкаут на юге Европы, а также чудеса нейро-лизоблюдия от ChatGPT

11 мин 13K

Дайджест

+34 17 70

alizar
18 часов назад

Ян Лекун, создатель LeNet, формата DjVu и адвокат опенсорса

Средний 7 мин 1.8K

Обзор

+28 13 6

El_Gato_Grande
18 часов назад

Как ИИ-контент проклял интернет и почему это закономерно

8 мин 4.4K

+25 10 9

BaDInMe
18 часов назад

ML-обработка видео в web-браузере для видеоконференций SaluteJazz

Средний 14 мин 239

Кейс

+21 5 0

full_moon
17 часов назад