Артем Гелун, руководитель отдела системной интеграции, SmartLabs

Под замком

Большинство упоминаний о DRM в настоящее время связаны с медиа — аудио- или видеоконтентом, правами на который и нужно «управлять» (DRM — Digital Rights Management). Это не всегда так (DRM может относиться, например, к компьютерным играм), но, учитывая профиль журнала, в этом тексте речь о DRM также пойдет в контексте медиа, в основном видеоконтента, «живого» или записанного.

Исторически существует две задачи по защите контента: предоставить контент только тому, кто имеет права на его получение, и гарантировать правомерное использование полученного контента. Первую задачу в TV решают системы контроля условного доступа — Conditional Access Systems, CAS. Вторую призваны были решить системы изначально другого класса — DRM.

Одной из первых DRM принято считать систему защиты от копирования Macrovision, даже несмотря на то, что она была аналоговой, а вовсе не digital.

Переход на цифровую доставку контента усугубляет прежде всего проблему защиты от копирования (ведь в отличие от аналогового копирования, при цифровом не происходит потери качества) и приводит к созданию множества различных систем от таких производителей, как Apple, Microsoft, Sony, и многих других, основная задача которых — контролировать копирование контента.

Повсеместное внедрение IPTV и OTT, приход на рынок «CAS с обратным каналом», усложнение моделей продаж контента приводят к смешению понятий CAS и DRM. Например, если клиент посмотрел какой-то фильм в прямом эфире или даже как потоковый EST* на ТВ-приставке, то, в классической терминологии, он использовал CAS. А если этот же клиент посмотрел тот же фильм, но предварительно скачав его на свою ТВ-приставку (предположим, что в нее встроен HDD), и мы хотим защитить этот фильм от копирования на другую приставку, то это уже вроде как DRM…

Поэтому сейчас DRM в чистом виде встречается все реже, часто упоминается как «CAS/DRM» (и даже если аббревиатура CAS нигде рядом не встречается, она подразумевается). Все последующие упоминания о DRM в статье — это упоминания о таком же «CAS/DRM», умеющем как защищать от несанкционированного использования контента, так и контролировать права на использование уже полученного контента.

DRM и безопасность

DRM по факту является системой контроля доступа, такой же, как логин/пароль на рабочую станцию, логин/пароль в личный кабинет какого-нибудь интернет-банка, карточка прохода на рабочее место и т.д. Только субъект контроля у нее — контент.

Как и у любой СКД (системы контроля доступа), у DRM есть несколько задач:

1) Аутентифицировать терминал (абонентское устройство), с помощью которого пользователь хочет получить доступ к контенту, — тип устройства и то, что оно, к примеру, имеет достаточный уровень защиты (например, что вы пользуетесь официальным приложением интернет-банка и оно не будет проводить транзакции без вашего ведома).

2) Аутентифицировать конечного абонента, который хочет получить доступ к контенту — действительно ли он тот, за кого себя выдает? По аналогии со СКУД (системой контроля условного доступа) — проверить пропуск / отпечаток пальца / пароль и т.д.

3) Авторизовать пользователя на данном терминале, т.е. определить, можно ли получить доступ к контенту, и если да, то на каких условиях (по аналогии с интернет-банком — может ли наш пользователь воспользоваться им с помощью приложения для iPhone).

4) Безопасно выдать «лицензию» — т.е. некоторое описание полномочий пользователя и ключи к контенту. Выдать ее так, чтобы она была доступна только аутентифицированному/авторизованному пользователю и, само собой, чтобы исключить утечку ключей (послать на замок последовательность, которая откроет только его и которую не может послать никто другой).

5) Предоставить доступ к контенту на конечном терминале так, чтобы неавторизованные пользователи не могли этим доступом воспользоваться (т.е. чтобы никто не мог пройти в дверь по вашему пропуску вместе с вами).

6) Предоставить информацию оператору о том, как, когда и кем были использованы права на контент. Это может быть как схема с регулярным обновлением лицензий (похожая на heartbeat-запросы), так и прямые отчеты о потреблении контента. Реализуется не всегда, конкретная реализация зависит от DRM (хранить записи камер наблюдения и факты передвижения по помещению).

Все эти пункты являются неотъемлемыми частями любой «коммерческой» DRM. Этап аутентификации абонента и авторизации зачастую перекладывается на внешние компоненты (какие именно, зависит от метода интеграции конкретной DRM с системами оператора), которые могут проводить аутентификацию и авторизацию абонента (п.п. 2 и 3) самостоятельно и разрешать / не разрешать доступ к контенту.

Теоретически системы, не требующие действительно высокого уровня безопасности, могут не реализовывать часть задач. Например:

- Отсутствие контроля политики использования контента (или вообще возможность использования контента только «сейчас», без возможности хранения копии и копирования) делает из нашего DRM'а CAS.
- Если отказаться от п.п. 4 (при этом используя протокол HTTPS — мы ведь

***EST — Electronic Sell Through,** — т.е. можно купить фильм, скачать его и не зависеть от оператора дальше (смотреть в оффлайне в том числе) — это EST + download. А можно купить фильм, но не иметь ресурсов его куда-то скачать (как на боксе без дисков) и каждый раз смотреть как обычный поток. В данном случае речь идет о потоковом EST.

не хотим, чтобы ключ в ответе было легко перехватить), 5 и упростить п. 1, то можно «вшить» в клиент что-то, что будет получать ключ в чистом виде по уникальной подписи клиента (так называемые «самоподписанные токены»).

- И т.д.

Недостатки DRM

1. Требуется установка специальных плагинов на ПК (в других типах клиентов DRM вшивается в приложение и прозрачна для абонента).

Проблема решается частично путем использования «родных» решений от производителей популярного ПО: Adobe, Google, Microsoft. Таким образом, для работы DRM нужен плагин, но он уже, скорее всего, установлен на вашем ПК — это либо Flash player, либо Chrome browser, либо Silverlight plugin.

2. Много клиентов –> много систем защиты (ни одна система не покрывает всех клиентов).

Этот пункт нерешаем по определению.

3. Много систем –> много форматов хранения, много копий данных.

Как правило, DRM предполагает определенный формат доставки контента. Например, Verimatrix Adaptive CAS требует HLS, Adobe Access — HDS, MS PlayReady — SmoothStreaming и т.д. Но даже если формат доставки один, то под каждую DRM контент должен быть зашифрован со своим ключом.

Хранение контента в нескольких копиях приводит к нерациональному использованию хранилищ (а значит и к увеличению стоимости хранения) и неоптимальной доставке по CDN.

Перепаковка контента «на лету» и тем более его перешифровка — задача ресурсоемкая и реализуемая только на сравнительно небольших нагрузках. Более того, контент должен быть переупакован/перешифрован на стороне CDN, на выходе к абоненту, что зачастую проблематично как технически, так и в вопросе соблюдения требований по доставке шифрованного контента.

Решение появилось сравнительно недавно и активно развивается в настоящее время— это MPEG-DASH/Common Encryption.

4. Много систем –> сложность интеграции.

Решить эту проблему можно с использованием «зонтичных» решений от производителей систем условного доступа, таких как Nagra, Verimatrix и многие другие. Но это приводит к существенному увеличению стоимости решения за счет лицензий на стороннее ПО, и оператору приходится определяться, что для него проще: интегрироваться с несколькими DRM или поставить проверенное решение от одного вендора.

Отклонение от темы: DASH или не DASH

Протокол MPEG-DASH не имеет прямого отношения к DRM, но очень тесно связан с ней, и сейчас существует некоторая путаница вокруг протокола MPEG-DASH, MSE API, Common Encryption (CENC) и EME API. Я бы хотел эти вопросы несколько прояснить.

Изначально речь будет идти о WEB и доставке контента в WEB.

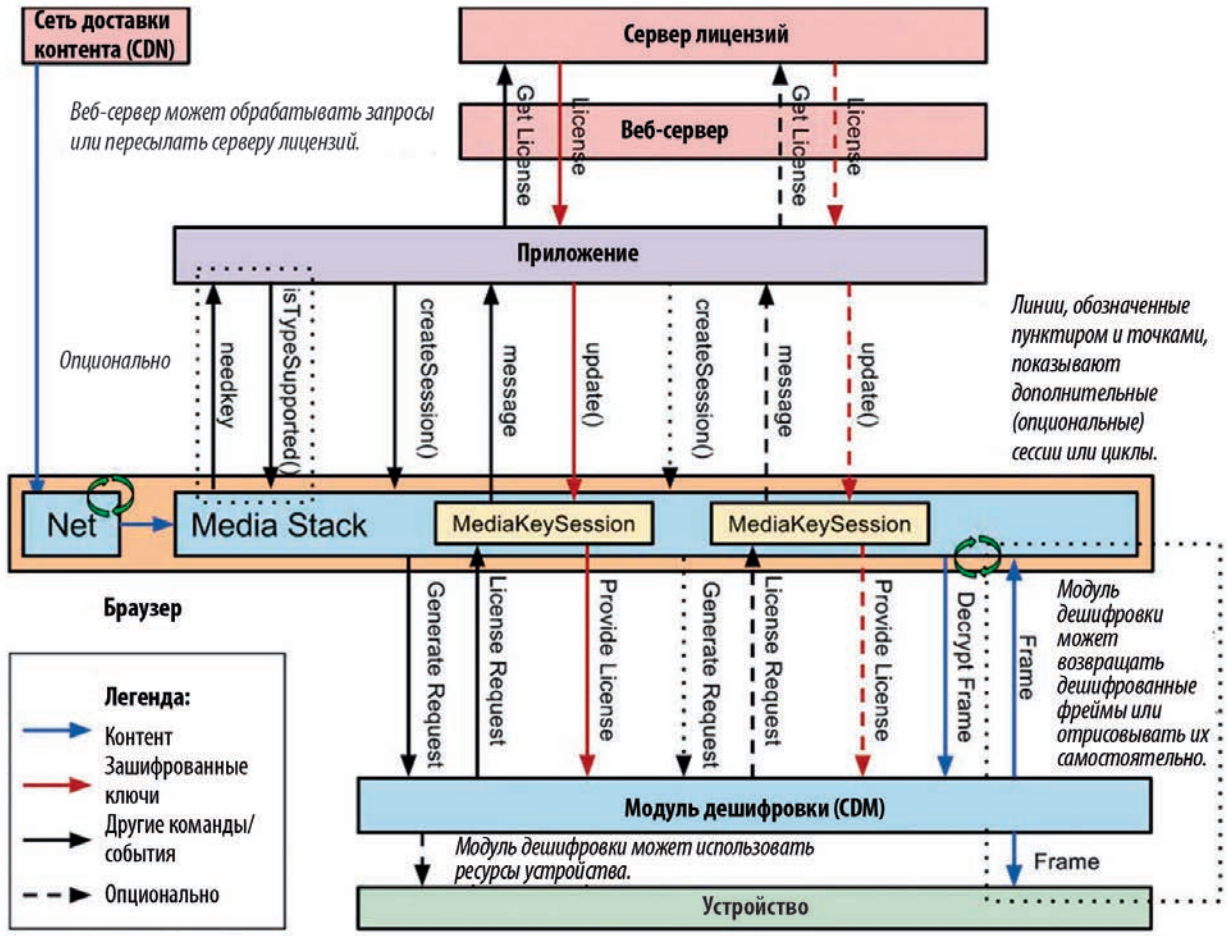
В HTML5 есть встроенный тэг <video>. Но возможности его весьма ограничены. В зависимости от конкретного браузера он, например, может воспроизвести обычный скачиваемый mp4-файл, но может не работать с протоколами типа DASH или HLS.

Чтобы поддержать в браузерах то, что не поддерживается «из коробки», в W3C разработали API MSE — Media Source Extensions, поддерживаемый большей частью браузеров и предоставляющий «прослойку» между желаемым протоколом и возможностями конкретного браузера. Например, реализация MSE для DASH обеспечивает парсинг manifest-файлов, переключение битрейтов, подгрузку видеофайлов и передачу этих видеофайлов «родному» плееру. Можно сказать, что MSE — это реализация конкретного протокола для браузера (на Java Script). Также могут быть реализованы и HLS, и SmoothStreaming, и многие другие протоколы (если это, конечно, кому-то надо и низкоуровневый протокол доставки не противоречит политике безопасности браузера).

Теоретически этим можно было бы и ограничиться: предобработка любого протокола позволяет делать с контентом все что угодно, в том числе и расшифровывать защищенное видео. Но если криптографию реализовывать на Java Script, то никакой «секретной» части в клиенте оставить не получится — все как на ладони, и логику клиента можно было бы повторить на любом устройстве, расшифровать контент и получить его «чистую» копию.

Поэтому для шифрования сделали второй API: EME — Encrypted Media Extensions. Если упрощенно, то он является интерфейсом между любым браузерным JS-кодом (например, DASH MSE) и конкретной безопасной (читай — на C/ассемблере, обфусцированной, использующей по возможности SoC и т.д.) реализацией подсистемы шифрования — CDM (Content Decryption Module). Например, WideVine, PlayReady или другой.

Но EME определяет только интерфейс для взаимодействия «плеера» (например,



MSE, но это может быть и вообще что-то нестандартное) и подсистемы шифрования (CDM). О том, как именно должно производиться шифрование, речь не идет. Для этого есть другие стандарты, например, Common Encryption for ISO Base Media File Format Files, в простонародье CENC, или Common Encryption.

CDM должен как реализовывать API EME, так и быть совместимым с тем стандартом шифрования, который предполагается использовать в MSE (например, Common Encryption). Если мы используем CDM, работающий с каким-то протоколом на основе CENC, то данные Initialization Data (терминология EME) должны представлять собой PSSH-бокс (терминология CENC). Хотя формат, содержимое PSSH-бокса зависит от целевого CDM, поэтому тут встает вопрос о курице и яйце.

Плюс к этому безопасный CDM должен не только расшифровывать и отдавать «на сторону» чистый контент, но и контролировать воспроизведение этого контента. Опять же, в случае с DASH/CENC речь идет не о парсинге mpd-файлов или даже mp4-контейнера с целью найти PSSH в нем, а о воспроизведении шифрованного elementary stream (который и расшифровывается, и воспроизводится внутри CDM, без выдачи наружу и без возможности перехвата).

Чтобы чуть уменьшить разброд и шатание, W3C выпустила отдельный драфт — ISO Common Encryption EME Stream Format and Initialization Data, по сути, связывающую EME и CENC.

При этом никто не обязывает использовать какой-то конкретный транспорт (например, DASH) со всеми этими EME, MSE, CDM и пр. Таким образом, если на основе какого-то альтернативного транспорта можно использовать стандартные API, то никто не мешает использовать этот альтернативный транспорт.

Стоит заметить, что и MSE, и EME до сих пор находятся в статусе драфтов.

Ну а как же это ложится на небраузерные модели клиентов? Как правило, интегратору, разрабатывающему приложение, использующее какую-то DRM, нужно реализовать (или подключить уже реализованную) функциональность каждого из описанных выше модулей: MSE (для реализации протокола доставки), EME (фактически сделать низкоуровневую связку с используемым CDM) и т.д. Жизнь разработчику, на первый взгляд, это вряд ли упрощает, но зато систематизирует и разделяет разные задачи по разным компонентам и позволяет тем же вендорам CDM выпускать библиотеки под разные устройства с одним (или очень схожим) API и логикой работы.

О достаточной защите

Идеальной защиты, как известно, не существует. Но защита считается достаточной тогда, когда защищаемый ресурс имеет меньшую стоимость, чем та, которую нужно заплатить (прямо или косвенно) за взлом самой защиты.

С этой точки зрения live-контент можно считать условно безопасным даже при минимальных мерах, так как источников его получения масса, куда более дешевых, чем взлом криптозащиты. Конечно, взлом любого нормального DRM в этой ситуации будет нецелесообразным — слишком сложно, а значит дорого.

А вот для VoD ситуация обстоит несколько иначе. VoD-контент значительно более «разношерстный», и защиты, которая достаточна для SD-фильма двухлетней давности (который давно доступен в других источниках и закрывать который нужно скорее не для того, чтобы защитить от копирования, а чтобы не провоцировать пиратство), будет явно недостаточно для HD-/UHD-новинки, которая только вышла в прокат в кинотеатрах и еще даже не выпущена на других электронных носителях.

«Большая двойка» в DRM (Microsoft и Google), а также некоторые другие вендоры адаптировали модель безопасности своих DRM к такой разношерстности кон-

тента и ввели сертификацию устройств на различные уровни безопасности, которые определяют как вопросы доставки ключей и шифрования, так и вопрос того, как устройство будет уникально идентифицировано. К примеру, для соответствия «званию самого безопасного» устройство должно получить сертификат еще на заводе, вся криптография в устройстве должна реализовываться только в защищенной памяти, расшифрованный контент должен доставляться только на встроенный экран или по HDMI с поддержкой HDCP, аналоговые выходы должны быть отключены и т.д. При этом другое устройство может целиком расшифровывать контент на CPU и хранить ключи в общей RAM (например, CDM в браузере).

Это позволило снизить в целом «порог вхождения» при интеграции устройств, требования к ним, не скомпрометировав себя перед мейджорами. Более того, это позволило задавать перечень устройств по известным критериям, на которых контент доступен, к примеру, при первом прокате (параллельно с кинотеатрами), и расширять этот перечень тогда, когда контент становится менее интересным для пиратов, например при выходе BluRay-копии (соответственно, взлом DRM становится экономически неоправданным).

Портрет пользователя OTT TB

Каждый год компания SPB TV проводит онлайн-опросы среди конечных пользователей сервиса SPB TV и партнерских проектов. Задача — понять интересы и потребности аудитории. В 2015 году были собраны ответы 60 000 респондентов на основании базы пользователей в 45 миллионов человек. Среди самых интересных данных — ответы пользователей на вопрос, какие услуги они хотят получить и за что готовы платить.

Главную роль играет контент и базовые услуги. 64 % респондентов готовы платить за сервис в том случае, если он полностью удовлетворяет их запросам в сфере видео. 45 % согласны платить, чтобы не видеть рекламу. Лишь 27 % готовы подписаться на платные телеканалы. 26 % готовы платить за доступ к ТВ-контенту в записи.

Только 19% опрошенных заявили, что не будут платить за ТВ- или видеосервис ни при каких условиях. Для сравнения, в 2013 году так ответили 27% респондентов.

Большинство респондентов (44 %) готовы дополнительно платить за высокое качество видео — HD и Ultra HD. Однако у OTT-ТВ-аудитории есть определенный список требований к технологическим возможностям сервиса, за которые дополнительно платить, в большинстве своем, они не готовы. Самые важные из требований: удобный и красивый интерфейс, полная программа передач, звук 5.1, персональный профиль, доступность сервиса на компьютере, мобильном и Smart TV с функциями отложенного просмотра и синхронизации просмотра между устройствами.

Задача поставщика решений и устройств для интерактивного ТВ (IPTV, OTT, кабельного) – предоставить оператору все необходимое (инфраструктуру, ПО, оборудование), включая надежные системы защиты контента. Мы в компании СмартЛабс традиционно интегрируем наши решения с самыми современными и надежными системами CAS/DRM от ведущих поставщиков, таких как Widevine/Google, Verimatrix, Secure Media, PlayReady. В частности, СмартЛабс первый в России системный интегратор DRM-решений Widevine/Google, которые очень популярны на Западе и высоко оцениваются мейджорами. При этом мы не ограничиваемся только решениями и приставками собственного производства и имеем опыт успешной интеграции с компонентами сторонних производителей.

Розыгрыш призов от журнала ТЕЛЕСПУТНИК

В октябре мы разыграли призы среди самых преданных читателей-подписчиков нашего издания.

СПИСОК ПОБЕДИТЕЛЕЙ:

Азарных Михаил Георгиевич, Калужская обл., п.о. Малые Зимницы
Приставка Nemo Box HD, тест в №1(231)

Бочаров Алексей, Санкт-Петербург
Dr. HD цифровой HD TV ресивер со спектроанализатором Grand Triple, тест в №3(221)

Дрезин Сергей Владимирович, Чувашия, г. Шумерля
Цифровой спутниковый ресивер Skyway DROID 2, тест в №10(228)

Есипко Игорь Анатольевич, РСО Алания, г. Владикавказ
Цифровой кабельный приемник Lumax DVC-2300 HD, тест в №1(231)

Ковалев Анатолий Алексеевич, Красноярский край, г. Минусинск
Мультимедийный плеер EVA Vision Mini, тест опубликован на сайте www.telemultimedia.ru

Погорелов Геннадий Дмитриевич, Белгородская обл., г. Короча
DVB-T2 шлюз для домашней сети VBox TV Gateway XTi-3342, тест в №2(2332)

Редакция журнала поздравляет победителей и благодарит компании EVAA, TelCo Group, Nemo TV, SAT.COM.RU, Skyway, Vbox Communications, любезно предоставившие призы для розыгрыша