Digital Rights Management

Encrypted Media Extensions

Patrick Bucher

25. Juni 2017

Inhaltsverzeichnis

1 DRM: Digital Rights Management

DRM steht für *Digital Rights Management*. Der Duden definiert DRM als «Gesamtheit der Strategien und Maßnahmen zur Kontrolle der Nutzung digitaler Medien» (Duden, 2015). Es geht also um die Verwaltung von Rechten im Zusammenhang mit digitalen Medien. Einem Lizenzeigner sollen bestimmte Nutzungsrechte gewährt werden. Personen ohne eine solche Lizenz sollen von der Nutzung ausgeschlossen werden. Darum steht DRM für Anhänger der «Free Software»-Bewegung auch für *Digital Restriction Management* (*What is DRM?*, 2017). Die beiden Begriffe beschreiben das gleiche, einfach aus einer anderen Perspektive.

DRM-Verfahren kommen etwa in folgenden Bereichen zum Einsatz:

- auf digitalen Datenträgern wie CD, DVD und Blu-ray-Disc als Kopierschutzmechanismus
- auf eBook-Plattformen wie Adobe Digital Editions oder Amazon Kindle zur Verwaltung der Leserechte
- bei digitalen Schnittstellen wie DVI und HDMI zur Verhinderung unautorisierten Abspielens von Videos

Hier soll es um DRM im Zusammenhang mit dem Abspielen von Videos im Webbrowser gehen, genauer um die *Encrypted Media Extensions*.

2 EME: Encrypted Media Extensions

2.1 Motivation

Wollte man sich vor einigen Jahren noch ein Video im Browser anschauen, war man auf Plugins wie RealPlayer oder Adobe Flash angewiesen. HTML5 brachte dann das `<video>`-Element mit, und die Browser implementierten Funktionen zum Abspielen von Videos. Das funktioniert für kostenlose Inhalte mittlerweile hervorragend. Das Problem ist aber, dass kostenpflichtige Angebote wie Netflix und Zattoo ihre Videos nur denjenigen (ohne Einschränkung) zeigen wollen, die vorher dafür bezahlt haben. Das funktioniert mit dem durch HTML5 spezifizierten `<video>`-Element nicht. Darum setzen kommerzielle Angebote weiterhin auf Technologien wie Microsoft Silverlight und Adobe Flash. Der HTML5-Standard muss also um entsprechende DRM-Mechanismen ergänzt werden, um kostenpflichtige Videoinhalte künftig ganz ohne Browser-Plugins abspielen zu können.

2.2 Status

Eine W3C-Arbeitsgruppe spezifizierte die *Encrypted Media Extensions* als optionale Erweiterung für HTML5-Videos (Dorwin, Smith, Watson & Bateman, 2017). Ein Browser, der die EME nicht implementiert, kann somit trotzdem noch standardkonform sein. Die EME beschreiben kein DRM-System, sondern eine API, womit DRM implementiert werden kann (Dutton, 2014). Die EME sind noch kein etablierter Standard, sondern liegen als *Proposed Recommendation* vor: die zweithöchste Stufe des fünfstufigen W3C-Standardisierungsprozederes. Webbrowser wie Chrome, Firefox, Safari, Edge und Internet Explorer haben die EME bereits teilweise implementiert. Netflix funktioniert auf Chromebooks (die auf Linux basieren und auf denen Silverlight nicht funktioniert) bereits seit 2013 über die EME (Park & Watson, 2013).

2.3 Komponenten

Da die EME kein komplettes DRM-System, sondern bloss eine API spezifizieren, werden darin keine Vorgaben über die eingesetzten kryptografischen Mechanismen und Verfahren gemacht. Dies ist Gegenstand einer anderen W3C-Spezifikation, der *Web Cryptography API*, die derzeit als *Recommendation* vorliegt (Watson, 2017).

Die Schlüssel müssen auf einem *License Server* abgelegt sein, und die *Web Application* (d.h. die Webseite mit den Videos) muss wissen, wo der Schlüssel für welches verschlüsselte Video zu finden ist. Die eigentliche Rechteverwaltung – wer Zugriff auf welche Videos hat – ist Sache der *Web Application* bzw. deren Entwickler.

Das Herzstück der EME ist das *Content Decryption Module* (CDM), welches die eigentliche Entschlüsselung (und optional auch die Dekodierung) der Videodatei vornimmt. Es kann als Bestandteil des Browsers implementiert oder als Plugin nachinstallierbar sein – oder sogar als Hardware oder Firmware umgesetzt werden.

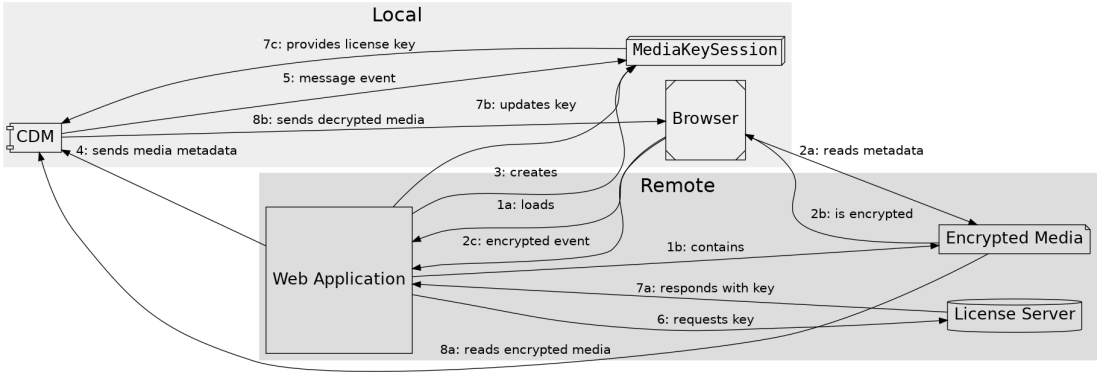

Abbildung 1: Das Abspielen eines EME-geschützten Videos

2.4 Ablauf

Gelangt der Benutzer auf eine Seite mit einem EME-geschützen Video, passiert folgendes:

- Der *Browser* lädt eine *Web Application* (1a). Diese enthält ein DRM-geschütztes Dokument (die *Encrypted Media*), sprich ein Video (1b).
- Der *Browser* liest die Metadaten der *Encrypted Media* aus (2a) und stellt fest, dass diese verschlüsselt ist (2b). Darauf löst der *Browser* den `encrypted`-Event aus, der von der *Web Application* entgegengenommen wird (2c).
- Die *Web Application* erstellt eine `MediaKeySession` (3), welche als Container für die Lizenzschlüssel fungiert, und sendet die Metadaten des Videos an das *CDM* (4).
- Das *CDM* löst einen `message`-Event aus (5), welcher von der `MediaKeySession` abgefangen wird. Diese stellt nun über die *Web Application* eine Key-Anfrage an den *License Server* (6).
- Der *License Server* antwortet und schickt den Lizenzschlüssel an die *Web Application* zurück (7a). Diese aktualisiert die `MediaKeySession` mit dem neuen Lizenzschlüssel (7b), welcher nun dem *CDM* zur Verfügung steht.
- Das *CDM* kann nun die *Encrypted Media* entschlüsseln (8a) und im Browser abspielen lassen (8b).

Abbildung 1 veranschaulicht diesen Prozess.

2.5 Fallbeispiel: Netflix 4k-Videos

Seit Ende 2016 bietet Netflix vereinzelte Filme und Serien in 4k-Auflösung (ca. 4000×2000 Bildpunkte) an (Warren, 2016). Um in den Genuss solch hochauflösender Videos zu kommen benötigt man aber mehr als einen 4k-Monitor, eine schnelle Internetverbindung und ein Netflix-Abo: Die 4k-Auflösung funktioniert nur auf Intel-Rechnern der neuesten Generation («Kaby

Lake»). Angeblich seien nur diese Chips schnell genug, um *HEVC*-codierte (*High Efficiency Video Codec*, auch als *h265* bekannt) 4k-Videos in Echtzeit zu dekodieren.

Die Begründung wirkt fadenscheinig, zumal Konkurrenzprodukte wie Radeon RX400 und GeForce GTX 1000 die zum Dekodieren erforderliche Performance problemlos erreichen. Tatsächlich dürfte ein noch nicht implementierter Kopierschutzmechanismus in den entsprechenden Treibern ausschlaggebend sein (Fischer, 2016). Die 4k-Auflösung steht nur dann zur Verfügung, wenn man Netflix über den Webbrowser Edge von Microsoft oder über die passende Windows-App verwendet (Hachman, 2017). Diese Einschränkung hat nichts mit der Hardware-Performance zu tun, vielmehr haben Microsoft und Netflix ein CDM ausgearbeitet, das sie anderen Anbietern nur eingeschränkt zur Verfügung stellen. Mitgliedern des *Windows Insider Programs* steht mittlerweile ein experimenteller Treiber für GeForce-Grafikkarten zur Verfügung, womit sich die 4k-Videos von Netflix abspielen lassen sollen (Nvidia-Support, 2017). Linux-Anwender werden noch länger auf die Serie *Gilmore Girls* in 4k-Auflösung verzichten müssen.

Das CDM scheint also aus einer Software- (Browser, App) und einer Hardware- bzw. Treiber-Komponente zu bestehen. Weder von Netflix noch von den Grafikkartenherstellern lassen sich genauere Informationen darüber in Erfahrung bringen. Dass die 4k-Auflösung bisher nur mit Software von Microsoft zur Verfügung steht, deutet auf die Verwendung von *PlayReady 3.0* – einem DRM-Verfahren aus dem Hause Microsoft – hin, das bereits in Intel- und (experimentell) Nvidia-GeForce-Grafiktreibern implementiert ist, jedoch noch nicht in den Radeon-Grafiktreibern von AMD.

2.6 Kritik

Das CDM umfasst je nach Hersteller und Implementierung andere Funktionalitäten. Wie das Netflix-4k-Beispiel zeigt, können sich Videostreaming-Plattformen mit Browser- und Hardware-Herstellern absprechen und ihre CDM nur noch für bestimmte Plattformen freigeben. Solche Absprachen könnten zu einem handfesten Marktvorteil führen, wobei in diesem Beispiel Microsoft zuungunsten von Apple (Mac OS, Safari), Google (Chrome) und Mozilla (Firefox) – bzw. Intel zuungunsten von AMD und Nvidia profitiert. Das offen konzipierte und auf Standards basierende Web könnte so bald von einzelnen Herstellern kontrolliert werden, zumindest was das Videostreaming anbelangt.

Ein CDM kann nicht nur zum Entschlüsseln, sondern auch zum Dekodieren von Videos verwendet werden. So könnten neue, hoch leistungsfähige Videocodecs nur den Benutzern zur Verfügung gestellt werden, die bereit sind ein proprietäres CDM zu installieren. Damit könnten Anbieter geschützter Videos Druck auf die Anwender ausüben. Glücklicherweise scheinen sich die meisten Anbieter von Videoplattformen (Netflix, Amazon), Webbrowsern (Microsoft, Firefox, Google) und Hardware (Intel, Nvidia, AMD) auf einen offenen und gebührenfreien Video-Codec einigen zu können (*Open. Fast. Royalty-free.*, 2015). Einzig Apple, das mit eigener Hardware, einem eigenen Betriebssystem, einem eigenen Browser und eigenen Vertriebskanälen ein geschlossenes «Ökosystem» bildet, könnte zum Spielverderber werden.

Viele Anbieter werden ihre CDM als Binärmodule und nicht quelloffen (mit entsprechender Open-Source-Lizenz) zur Verfügung stellen. Das ist gerade bei den Open-Source-Browsern Chrome und Firefox ein Problem. Für Firefox soll dieses Problem entschärft werden, indem das CDM als Plugin nachgeladen und nicht direkt im Browser implementiert wird.

Für Sicherheitsforscher stellen CDM ein gewaltiges Problem dar, da das Reverse-Engineering dieser Komponenten unter den *Digital Millenium Copyright Act* fällt (*DMCA*; Umgehung von Kopierschutzmechanismen) und somit verboten ist.

Wer sich in Zukunft kostenpflichtige Videos im Webbrowser anschauen will, der wird sich vorerst zwischen den proprietären Lösungen Adobe Flash und Microsoft Silverlight einerseits und den proprietären EME-Browsermodulen (mit entsprechender Grafikkarte und Software) andererseits entscheiden müssen. Der Einwand, dass wer sich proprietäre Videos anschauen will auch proprietäre Software ausführen soll, greift zu kurz, zumal die EME-Spezifikation der Grundidee des «freien» Webs zuwiderläuft.

Literatur

Dorwin, D., Smith, J., Watson, M. & Bateman, A. (2017). *Encrypted Media Extensions.* `https://www.w3.org/TR/encrypted-media/`. W3C. (Zugriff am 26. April 2017)

Duden (Hrsg.). (2015). *Duden: Deutsches Universalwörterbuch* (8. Aufl.). Bibliographisches Institut.

Dutton, S. (2014). *EME WTF?* `https://www.html5rocks.com/en/tutorials/eme/basics/`. HTML5 Rocks. (Zugriff am 22. Mai 2017)

Fischer, M. (2016). *Netflix 4K auf dem PC ausprobiert: Kaby Lake funktioniert, GeForce 1000 und Radeon RX 400 nicht.* `https://heise.de/-3505452`. Heise Online. (Zugriff am 12. Juni 2017)

Hachman, M. (2017). *Tested: Microsoft Edge is the only browser to run Netflix in 4K.* `http://www.pcworld.com/article/3181818/browsers/tested-microsoft-edge-is-the-only-browser-to-run-netflix-in-4k.html`. PCWorld. (Zugriff am 12. Juni 2017)

Nvidia-Support. (2017). *Preview of 4K UHD Netflix content on NVIDIA GPUs.* `https://nvidia.custhelp.com/app/answers/detail/a_id/4457`. Nvidia. (Zugriff am 12. Juni 2017)

Open. Fast. Royalty-free. (2015). `http://aomedia.org/about-us/`. Alliance for Open Media. (Zugriff am 12. Juni 2017)

Park, A. & Watson, M. (2013). *HTML5 Video at Netflix.* `https://medium.com/netflix-techblog/html5-video-at-netflix-721d1f143979`. Medium.com. (Zugriff am 25. Mai 2017)

Warren, T. (2016). *4K Netflix arrives on Windows 10, but probably not for your PC.* `https://www.theverge.com/2016/11/21/13703152/netflix-4k-pc-windows-support`. The Verge. (Zugriff am 26. Mai 2017)

Watson, M. (2017). *Web Cryptography API.* `https://www.w3.org/TR/WebCryptoAPI/`. W3C. (Zugriff am 25. Mai 2017)

What is DRM? (2017). `https://www.defectivebydesign.org/what_is_drm_digital_restrictions_management`. Free Software Foundation. (Zugriff am 22. Mai 2017)