Digital Rights Management

Encrypted Media Extensions

Patrick Bucher

27. Mai 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 BluRay-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 denen (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 erst als *Proposed Recommendation* vor. Dennoch haben *Chrome*, *Firefox*, *Safari*, *Edge* und *Internet Explorer* 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.

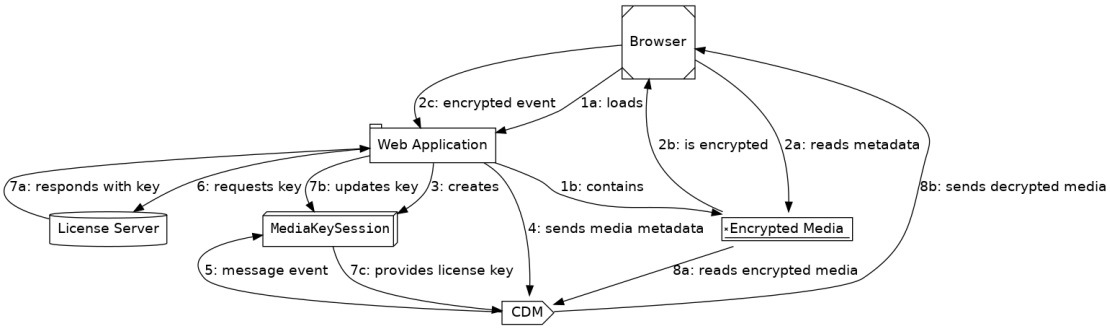

Abbildung 1: Das Abspielen eines EME-geschützten Videos

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, das 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), welche 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 Kritik

Das *Content Decryption Module* umfasst je nach Hersteller und Implementierung andere Funktionalitäten. So könnten sich Videostreaming-Plattformen mit Browserherstellern absprechen und ihre *CDM* nur noch für bestimmte Browser und Plattformen freigeben. Man stelle sich vor, *Netflix* würde seine Videos (gegen entsprechende Gegenleistung von *Apple*) nur noch auf *Mac*

OS und *Safari* abspielen lassen! Tatsächlich arbeitet *Netflix* bereits mit solchen Einschränkungen, indem es 4k-Videos nur auf Intel-Rechnern der neuesten Generation («Kaby Lake») ausführen lässt. Dies wird damit begründet, dass nur diese Chips genug schnell seien, um HEVC-codierte (*High Efficiency Video Codec*, auch als *h265* bekannt) 4k-Videos in Echtzeit zu dekodieren (Warren, 2016). 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 Vidoes Druck auf die Anwender ausüben.

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 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)

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)