Fair DRM – Ermöglichen von Privatkopien und Schutz der digitalen Ware

André Adelsbach, Ulrich Greveler, Jörg Schwenk*

31. Januar 2005

Zusammenfassung

Wir beschreiben ein *faires* DRM-Verfahren, mit dessen Hilfe es möglich ist, die nach gängigem Rechtsempfinden erlaubten sieben Privatkopien anzufertigen, bei dem aber ab der achten (nicht mehr erlaubten) Privatkopie eine hohe Wahrscheinlichkeit besteht, wegen Piraterie zur Rechenschaft gezogen zu werden. Zur Realisierung dieses Verfahrens werden Secret-Sharing-Schemata eingesetzt, die es ermöglichen, dass die Identität eines Nutzers erst dann von einer zentralen Instanz aufgedeckt werden kann, wenn ein Nutzungsrechte-Verstoß vorliegt. Solange ein Nutzer sich an die Lizenzbedingungen hält, bleibt seine Anonymität gewahrt. Die Anzahl der zulässigen Privatkopien (z. B. 7) ist parametrisierbar.

1 Einführung

In der deutschen Rechtsprechung existieren heute nebeneinander ein Recht auf Privatkopien digitaler Datenträger (wir gehen in diesem Artikel von einem Durchschnittswert von sieben Privatkopien aus [10]), und ein Verbot des Umgehens des Kopierschutzes dieser Datenträger.

Um diesen Widerspruch aufzulösen und die Attraktivität digitaler Inhalte zu erhöhen, könnten die Hersteller der Kopierschutzverfahren auch spezielle „Privatkopie“-Geräte anbieten, die die Anfertigung der gesetzlich erlaubten Privatkopien ermöglichen und gleichzeitig die Anfertigung weiterer Kopien unterbinden. Eine solche Lösung erscheint aber aus praktischer Sicht undurchführbar, da zur Anfertigung von Privatkopien für jedes eingesetzte Verfahren ein solches Gerät öffentlich zugänglich (beispielsweise in öffentlichen

*Horst-Görtz-Institut für IT-Sicherheit, Ruhr-Universität, 44780 Bochum, e-mail: {andre.adelsbach, ulrich.greveler, joerg.schwenk}@nds.rub.de

Bibliotheken) aufgestellt werden müsste. Des Weiteren müssten diese Geräte aufwändige Methoden zur Benutzer- und Inhalteverwaltung einsetzen, um die Erzeugung weiterer Privatkopien zu verhindern. Eine Alternative dazu bietet das Konzept der so genannten „Authorised Domains“ [4]. In diesem Konzept stellen die Geräte eines Benutzers einen Bereich dar, in dem legal erworbene digitale Inhalte frei ausgetauscht und genutzt werden können. Ein Transfer digitaler Inhalte auf Geräte außerhalb dieses Bereiches wird durch die konformen Geräte verhindert.

Im Gegensatz zu diesen präventiven Mechanismen verfolgen wir in diesem Artikel einen anderen Ansatz: Es soll gar nicht mehr verhindert werden, Privatkopien zu erstellen, aber bei Überschreiten der erlaubten Anzahl von Privatkopien (und nur dann) soll der Nutzer mit Sanktionen rechnen müssen. Daher kann unser Ansatz, ähnlich wie Fingerprinting [1] oder das Lightweight DRM System [5] als abschreckende Maßnahme interpretiert werden, die vor einer illegalen Verbreitung digitaler Inhalte abschrecken, da ein Fehlverhalten nachweisbar wird. Im Gegensatz zu diesen existierenden Verfahren ermöglicht unser Verfahren zusätzlich eine effektive Limitierung der Anzahl durch konforme Geräte angefertigter Privatkopien.

Zu diesem Zweck werden jeder Lizenz als Shares eines Secret-Sharing Verfahrens als Zusatzdaten angefügt, die die Identität und ein Zahlungsversprechen des Käufers des digitalen Inhaltes codieren. Wird diese Lizenz auf einem Gerät verwendet, das hardware- oder softwarekompatibel zu diesem fairen DRM-System ist (nur auf solchen Geräten ist ein Abspielen überhaupt möglich[1]), so kann dieses Gerät ein Share S der Identität und des Zahlungsversprechens berechnen. Dieses Share wird an eine Verwertungsgesellschaft übermittelt.

Wenn ein Kunde nur maximal 7 Kopien erstellt, genauer gesagt, wenn der Content nur maximal auf sieben verschiedenen Fair DRM -kompatiblen Geräten abgespielt wird, so erhält auch die Verwertungsgesellschaft maximal 7 Shares. $(8, n)$-Secret-Sharing-Verfahren zeichnen sich nun dadurch aus, dass man aus 7 Shares informationstheoretisch keinerlei Wissen über die Identität/Zahlungsversprechen des Käufers erlangen kann. Der Datenschutz ist in diesem Fall perfekt gewährleistet, auch wenn die Verwertungsgesellschaft nicht vertrauenswürdig sein sollte.

Ab der 8. Privatkopie, genauer gesagt ab dem 8. Fair-DRM Gerät, auf dem der digitale Inhalt abgespielt wird, besteht nun eine signifikante, vom Parameter n abhängige Wahrscheinlichkeit, dass 8 verschiedene Shares an die Verwertungsgesellschaft übermittelt werden. In diesem Fall kann die Identität des Käufers rekonstruiert werden.

Bei Verteilung eines geschützten Inhaltes über eine Internet-Tauschbörse nähert sich die Wahrscheinlichkeit, die Identität des ursprünglichen Käufers ermitteln zu können, dem Wert 1: Der ursprüngliche Käufer verliert hier jegliche Kontrolle darüber, auf wie vielen Playern der Content abgespielt wird. Somit können unehrliche Käufer mit an Sicherheit

[1] Dies kann einfach erreicht werden, indem die Inhalte zusätzlich verschlüsselt werden und der notwendige Entschlüsselungsschlüssel nur lizenzierten Geräten bekannt gemacht wird.

grenzender Wahrscheinlichkeit identifiziert werden.

2 Modell

In diesem Abschnitt modellieren wir das Fair-DRM System: Dazu definieren wir zunächst die Einheiten/Rollen unseres Systems und beschreiben das Vertrauensmodell. Des Weiteren geben wir einen Überblick über die notwendigen Transaktionen im vorgeschlagenen Fair-DRM System und führen die wesentlichen Sicherheitsanforderung ein.

2.1 Rollen und Transaktionen

Wir unterscheiden folgende Parteien/Rollen in unserem Fair-DRM System:

1. Inhalteanbieter besitzen Rechte auf digitale Inhalte und bieten Inhalte zur Nutzung an.

2. Lizenzierungsstellen verkaufen Lizenzen, welche ihren jeweiligen Besitzer zur Nutzung digitaler Inhalte autorisieren. Die Lizenz kann dabei weitere Nutzungsbedingungen umfassen (z. B. Pay-per-View).

3. Verwertungsgesellschaften erhalten Informationen über die Nutzung digitaler Inhalte und entlohnen die Inhalteanbieter. Die Entlohnung eines Inhalteanbieters ist dabei abhängig von der Nutzung der von ihm geschaffenen Inhalte. Verwertungsgesellschaften können in der Praxis die Aufgaben der Lizenzierungsstellen übernehmen. Zur Verdeutlichung des Systems haben wir diese Funktionen getrennt.

4. Die Pseudonymisierungsstelle zertifiziert Benutzerpseudonyme. Während der Zertifizierung eines Pseudonyms werden Voraussetzungen für eine Deanonymisierung eines Pseudonyms festgelegt. Kann eine Partei nachweisen, dass diese Voraussetzungen vorliegen, so legt die Pseudonymisierungsstelle die Identität des Benutzers offen, der die Zertifizierung des Pseudonyms beantragt hat. Die Funktion der Pseudonymisierungsstelle kann auch von der Verwertungsgesellschaft übernommen werden, sofern die Benutzer der Verwertungsgesellschaft hinreichend vertrauen.

5. Benutzer beziehen digitale Inhalte von Inhalteanbietern und erwerben eine zugehörige Lizenz von einer Lizenzierungsstelle, um diese Inhalte abspielen (nutzen) zu können. Zum Schutz ihrer Privatsphäre können Benutzer Lizenzen und Inhalte unter ihren Pseudonymen erwerben.

6. Endbenutzergeräte spielen digitale Inhalte ab, sofern eine zugehörige Lizenz verfügbar ist.

2.2 Vertrauensmodell und Sicherheitsanforderungen

In diesem Beitrag liegt unser Fokus auf den Transaktionen zwischen Lizensierungsstellen/Verwertungsgesellschaften und Benutzern (Konsumenten). Zur einfacheren Darstellung nehmen wir daher an, dass Inhalteanbieter darauf vertrauen können, dass Lizenzierungsstellen ausgestellte Lizenzen korrekt abrechnen und dass die Verwertungsgesellschaften die Tantiemen korrekt verteilen. Endbenutzergeräte werden als tamper-resistant angenommen, d.h. die Geräte halten sich immer an die Spezifikation des Fair-DRM Systems und die enthaltenen Geheimnisse können nicht ausgelesen oder modifiziert werden.

Sicherheitsanforderungen werden sowohl von Seite der Inhalteanbieter, als auch von Seite der Benutzer an das Fair-DRM System gestellt. Die Inhalteanbieter fordern im Wesentlichen, dass sie die Nutzung Ihrer Inhalte wirksam steuern können und angemessen entschädigt werden, wenn ein Benutzer mehr als 7 Privatkopien verwendet. Die Benutzeranforderungen bestehen darin, dass sie Inhalte anonym nutzen wollen und eine Deanonymisierung nur dann tolerieren, wenn sie mehr als die erlaubte Anzahl an Privatkopien gemacht haben. Des Weiteren möchten Benutzer nicht für mehr Privatkopien zahlen, als sie auch tatsächlich angefertigt und verwendet haben.

3 Threshold-Verfahren

Das in diesem Paper vorgeschlagene Verfahren greift auf Threshold-Verfahren als kryptographischen Baustein zurück. Threshold-Verfahren (auch Secret-Sharing-Verfahren genannt) ermöglichen es, ein Geheimnis auf eine Weise in Teilgeheimnisse (*shares*) zu zerlegen, dass aus Kenntnis einzelner oder weniger dieser Teilgeheimnisse keine Informationen über das Geheimnis selbst gewonnen werden können; wird jedoch ein definierter Schwellenwert (*threshold*) überschritten, kann das Geheimnis einfach errechnet werden. Ein (t, n)-Threshold-Verfahren liefert für einen Eingabewert (das Geheimnis) eine Menge von n paarweise verschiedenen Shares mit der Eigenschaft, dass die Kenntnis von t oder weniger Shares keine Information über den Eingabewert liefern, jedoch lässt sich bereits aus beliebigen $t + 1$ Shares der Eingabewert effizient berechnen. Geeignete Verfahren werden z. B. von A. Shamir in [11] beschrieben. Dieses Verfahren beruht auf der Bestimmbarkeit von Polynomen über endlichen Körpern und zeichnet sich insbesondere durch seine Einfachheit, Effizienz sowie informationstheoretische Sicherheit aus. Die Größe des Körpers N bestimmt dabei die Größe der Kodierbaren Geheimnisse und ist gleichzeitig eine obere Schranke für Anzahl erzeugbarer Shares.

Die genannten Eigenschaften von Threshold-Verfahren können auf den Aspekt Privatkopien angewendet werden, indem auf ein $(8, n)$-Threshold-Verfahren zurückgegriffen wird, das es ermöglicht, ein Geheimnis (hier die Identität eines Nutzers) zu wahren, solange nicht mehr als sieben Shares (die in die Privatkopien eingebettet werden) bekannt sind; ab der achten Kopie sind auch mehr als sieben dieser generierten Shares bekannt und die geheime Identität kann mit hoher Wahrscheinlichkeit errechnet werden.

Um Threshold-Verfahren hier anwenden zu können, wird ein Mechanismus konstruiert, der die Übermittlung der Shares beim Nutzen (Abspielen) des Inhaltes an eine zentrale Instanz (eine Verwertungsgesellschaft) sicher stellt. Diese Instanz kann die Identität eines Nutzers aufdecken, wenn mehr als sieben Privatkopien abgespielt wurden. Solange der Schwellenwert nicht erreicht wird, bleibt der Nutzer gegenüber dieser Instanz anonym.

4 Das Fair-DRM Verfahren

Im Folgenden nehmen wir an, dass der Benutzer B ein Schlüsselpaar (sk_B, pk_B) und ein entsprechendes Zertifikat $cert_B$ auf seine wahre Identität B besitzt. Des Weiteren nehmen wir an, dass jedes Endbenutzergerät einen symmetrischen *Player Identification Key* (PIK) besitzt.

4.1 Setup (Registrierung und Zertifizierung eines Pseudonyms)

Vor dem Erwerb einer Lizenz beantragt der Benutzer zunächst ein neues Pseudonym in Form eines Schlüsselpaares (sk_P, pk_P) und beantragt ein zugehöriges Zertifikat.[2]

- Dazu sendet er eine signierte Anfrage (engl. request)

  $$Req_{Pseud} = Sign(sk_B, pk_P||terms)$$

  an eine Pseudonymisierungsstelle. In *terms* wird der Haftungsumfang und die Voraussetzungen für eine Haftung und Deanonymisierung festgelegt.

- Die Pseudonymisierungsstelle antwortet mit einem entsprechenden Zertifikat $cert_P$.

Der öffentliche Schlüssel pk_P (bzw. das zugehörige Zertifikat) wird im Folgenden als Pseudonym des Benutzers B dienen.

4.2 Zugriff auf digitale Inhalte

Digitale Inhalte werden nur in verschlüsselter Form verbreitet und können daher frei (sogar per Superdistribution über P2P-Netze) verteilt werden. Der Benutzer B kann sich nun aus beliebigen Quellen verschlüsselte Inhalte auf eines seiner Abspielgeräte laden. Um Inhalte durch ein konformes Endbenutzergerät abspielen zu können, muss er, wie im Folgenden beschrieben wird, eine zugehörige Lizenz erwerben. Diese enthalten unter anderem den benötigten Entschlüsselungsschlüssel und legen die erlaubten Nutzungsarten des Inhaltes verbindlich fest.

[2]Bei geringen Anforderungen bezüglich Anonymität bzw. Unverkettbarkeit unterschiedlicher Transaktionen kann ein Pseudonym auch für den Bezug mehrerer Lizenzen verwendet werden.

4.3 Erwerb einer Lizenz

Der Benutzer B sendet zunächst einen mit Pseudonym pk_P signierte Anfrage Req an die Lizenzierungsstelle

$$Req = Sign(sk_P, < DOI||cert_P||terms >)$$

Hier bezeichnet DOI einen Bezeichner (z. B. den Digital Object Identifier [7] oder einen kryptographischen Hash des Inhaltes), der den Inhalt eindeutig identifiziert, für den eine Lizenz angefordert wird. $cert_P$ ist das Pseudonym des Benutzers B unter dem die Lizenz angefordert wird. Die Zeichenkette $terms$ beinhaltet ein Zahlungsversprechen an den Inhalteanbieter / Verwertungsgesellschaft für den Fall, dass zu viele (Privat)kopien angefertigt und genutzt werden.

Die Lizenzierungsstelle sendet daraufhin eine signierte Lizenz an den pseudonymen Benutzer. Diese Lizenz kann mehrere mögliche Formen haben:

1. *Die Lizenz enthält ein zufällig gewähltes Polynom Pol.*

   $$Lizenz = Sign(sk_L, DOI||Pol||Rights||Enc(pk_P, key))$$

   Pol ist hier ein zufälliges Polynom vom Grad $k-1$ $(= 7)$ mit $Pol(0) = Req$ über einem endl. Körper mit $N > n$ Elementen, wie es im Secret-Sharing Verfahren von Shamir beschrieben wird. Dieses Polynom wird bei der Wiedergabe von den konformen Geräten an zufälligen (aber geräte-spezifischen) Positionen ausgewertet, um die Shares von Req zu berechnen.

2. *Die Lizenz enthält verschlüsselte Shares.*

   $$Lizenz = Sign(sk_L, DOI||S_1, \ldots, S_n||Rights||K_1, \ldots, K_n)$$

   Hier bezeichnen $S_1, ..., S_n$ Shares eines (k, n)-Threshold-Verfahrens (über einem Körper der Größe N) von Req, die unter verschiedenen Player Identifikation Keys $(PIK_1, \ldots, PIK_n)$ verschlüsselt sind:

   $$S_i := Enc(PIK_i, s_i), K_i := Enc(PIK_i, key).$$

Lizenzen der ersten Form besitzen einen wesentlichen Vorteil gegenüber Lizenzen der zweiten Form: der Sicherheitsparameter n kann sehr groß gewählt werden, ohne die Lizenzen wesentlich zu verlängern, wohingegen die Lizenzen der zweiten Form linear in diesem Sicherheitsparameter wachsen. Der Nachteil der ersten Form besteht darin, dass jedes Gerät Zugriff auf das vollständige Polynom / Geheimnis $(Pol(0))$ hat und prinzipiell alle n Shares selbst erzeugen kann. Da die Benutzer den Geräten jedoch ohnehin vollständig (bzgl. Anonymität) vertrauen müssen, ist dies kein wesentlicher Nachteil. In beiden Fällen ist zu beachten, dass der Parameter N hinreichend groß gewählt wird, um das Geheimnis Req zu kodieren, z.B. als Primzahl der Länge $|Req|$.

4.4 Wiedergabe digitaler Inhalte

Der Benutzer lädt sowohl den verschlüsselten Inhalt, als auch eine zugehörige Lizenz auf das Gerät. Optional kann er dazu aufgefordert werden, sich unter dem Pseudonym pk_P gegenüber dem Gerät zu authentifizieren. Nun überprüft das Gerät die Lizenz (Signatur der Lizenzierungsstelle, Nutzungsrechte sowie DOI). Sind alle Überprüfungen erfolgreich, besitzt der Benutzer das Recht zur Wiedergabe des Inhaltes. Bevor nun mit der eigentlichen Wiedergabe begonnen wird, berechnet das Gerät ein zugehöriges Share. Dies funktioniert, abhängig von der Form der Lizenz, wie folgt:

1. *Auswertung des Polynoms an zufälliger Stelle.* Das Gerät wertet das Polynom $Pol()$ an der Stelle
$$i := Hash(Lizenz||Geraete - ID)$$
aus. Der Wert S, der an die Verwertungsgesellschaft gesendet wird, ist wie folgt aufgebaut:
$$S := DOI||Hash(Lizenz)||i||Pol(i)$$
Durch die Berechnung des Auswertepunktes i in Abhängigkeit von Lizenz und $Geraete - ID$ wird sichergestellt, dass ein Gerät immer das gleiche Share zu einer Lizenz erzeugt. Unterschiedliche Geräte berechnen jedoch mit überwältigender Wahrscheinlichkeit verschiedene Shares, weil sie das Polynom an unterschiedlichen Positionen auswerten (Kollisions-Resistenz der Hashfunktion).

2. *Entschlüsselung der verschlüsselten Shares.* Das Gerät besitzt einen Geräteschlüssel PIK_i, mit dem er das zugehörige verschlüsselte Share S_i entschlüsseln kann.
$$S := DOI||Hash(Lizenz)||Dec(PIK_i, S_i).$$

Der berechnete Wert S wird vom Gerät signiert und an die Verwertungsgesellschaft gesendet. Die Verwertungsgesellschaft bestätigt den Empfang von S mit einer signierten Empfangsquittung. Nach Erhalt einer gültigen Quittung startet das Gerät die Wiedergabe, indem der verschlüsselte Inhalt mit dem Schlüssel aus der Lizenz entschlüsselt wird. Um den Grad der Interaktion zu reduzieren könnten Endgeräte eine bestimmte Anzahl von Shares sammeln und diese gemeinsam quittieren lassen. Bleibt eine solche Sammelquittung aus, müßte das Endgerät die Wiedergabe weiterer Inhalte verweigern.

4.5 Rekonstruktion der Shares und De-Anonymisierung

Die Verwertungsgesellschaft verwaltet alle empfangenen Shares, sortiert nach ihrem Präfix $DOI||Hash(Lizenz)$, in einer Datenbank, wobei doppelte Shares aus der Datenbank entfernt werden können. Sobald die Verwertungsgesellschaft mehr als 7 unterschiedliche Shares mit identischem Präfixen $DOI||Hash(Lizenz)$ empfangen hat, kann sie die Shares zu dem ursprünglichen Request

$$Req = Sign(sk_P, < DOI||cert_P||terms >)$$

zusammensetzen. Mit diesem Request beantragt sie nun die Deanonymisierung des Pseudonyms pk_P bei der Pseudonymisierungsstelle und fordert das Zahlungsversprechen des Benutzers B ein. Auf eine Deanonymisierung kann prinzipiell auch verzichtet werden, wenn entsprechende (anonyme) Zahlungssysteme verwendet werden. Die erhaltene Zahlung kann nun an den Inhalteanbieter weitergeleitet werden, um ihn angemessen für die Nutzung seiner Werke zu entlohnen.

5 Sicherheitsbetrachtung

5.1 Perfekter Datenschutz

Solange der Nutzer höchstens 7 Privatkopien anfertigt (egal ob für sich selbst oder für andere), d.h. der Content höchstens auf 7 verschiedenen Playern abgespielt wird, kann die Verwertungsgesellschaft aus den 7 empfangenen Shares die Identität nicht rekonstruieren: Alle möglichen Werte für *Req* sind gleich wahrscheinlich in einem $(8, n)$-Threshold-Verfahren. Der Datenschutz ist für den Nutzer perfekt gewährleistet.

Wird eine 8. Kopie erstellt, so sind mit Wahrscheinlichkeit (hier sei k die Anzahl der erstellten Kopien einschließlich der selbstgenutzten Kopie des Käufers)

$$P_{(k=8)}(Reconstruct(s_i = s)) = \frac{n(n-1)\dots(n-7)}{n^8}$$

bereits alle 8 Shares paarweise verschieden, und die Identität des Nutzers kann ermittelt werden. Dieser Wert ist höher, je größer n ist. Beispielsweise ist diese Wahrscheinlichkeit größer als 97 Prozent, wenn wir $n = 1000$ wählen (oder größer als 99 Prozent wenn wir für $n = 10000$ wählen).

5.2 Sicherheit der PIKs

Angriffe auf die Vertraulichkeit der *PIKs* brechen die Sicherheitsanforderungen des Benutzers (Datenschutz und gerechte Abrechnung) nicht: Der Kunde hat weiterhin volle Kontrolle über den Content und könnte nur dann fälschlicherweise belastet werden, wenn er eine der 7 erlaubten Kopien an einen Angreifer weitergibt, der mehrere Shares entschlüsseln und an die Verwertungsgesellschaft senden würde. Den gleichen Effekt kann der Angreifer aber wesentlich leichter erzielen, wenn er die ihm anvertraute Kopie einfach mindestens 8 mal auf verschiedenen Playern abspielt.

Gelingt es einem Angreifer, den Content von den verschlüsselten Shares zu trennen, so ist das System gebrochen. Um dies zu verhindern, müssen die Shares über so genannte Hooks (vgl. MPEG21 in [6]) mit dem Content verbunden werden. Folgende Hooks sind möglich und werden in der Praxis eingesetzt:

- **Verschlüsselung des Content.** Der Content wird so verschlüsselt, dass er nur von Fair-DRM kompatiblen Clients entschlüsselt werden kann. Diese Clients testen gleichzeitig das Vorhandensein der verschlüsselten Shares, und verweigern die Wiedergabe, wenn diese fehlen. Dies entspricht der von verfolgten Strategie. Die volle Stärke dieses Ansatzes kann sich allerdings nur in sicheren Endgeräten entfalten, z. B. in PC, auf denen Trusted Computing möglich ist [13].
- **Robuste digitale Wasserzeichen.** Die verschlüsselten Shares werden mit einem digitalen Wasserzeichenverfahren eingebettet. Robust bedeutet hierbei, dass eine Entfernung dieser Daten einen erheblichen Qualitätsverlust des Content nach sich zieht. Die Entwicklung robuster Wasserzeichenverfahren wird von vielen Arbeitsgruppen weltweit vorangetrieben. Eine gute Einführung in digitale Wasserzeichen bietet [3].

5.3 Äquivalenzklassen von Playern

Für Variante 2 des Verfahrens sind noch besondere Vorkehrungen notwendig, um die Sicherheit des Verfahrens zu gewährleisten. In diesem Verfahren gehören zwei Player zur gleichen Äquivalenzklasse i, wenn sie den gleichen Geräteschlüssel PIK_i besitzen. Könnte ein Angreifer die Äquivalezklasse eines Players ermitteln, so könnte er unbegrenzt viele Kopien eines Werkes auf Playern aus maximal 7 Äquivalenzklassen abspielen. Folgende Maßnahmen müssen getroffen werden, um dies zu verhindern:

- Die PKIs müssen so im ausführbaren Code des Players versteckt sein, dass nicht einfach durch Durchsuchen des ausführbaren Codes die Äquivalenzklasse ermittelt werden kann.
- Die Übertragung der Shares an die Clearingstelle muss verschlüsselt erfolgen (z. B. mit SSL), sonst kann man durch Abspielen des Content die Äquivalenzklasse des Players ermitteln.
- *Randomisierung der Client-Software.* Eine Möglichkeit, die Bestimmung von Äquivalenzklassen unmöglich zu machen, bestünde im Prinzip darin, die n Äquivalenzklassen möglichst ähnlich zu machen. Dieser Ansatz ist nur durchführbar, wenn es um die Gleichheit bestimmter Prüfsummen des ausführbaren Codes geht, scheitert aber schon an einer einfachen Hashwertbildung: man müsste eine n-fache Kollision bezüglich jeder gängigen Hashfunktion wie MD5, SHA-1 und RIPEMD erzeugen, sonst sind die Äquivalenzklassen anhand ihres Hashwerts unterscheidbar. Es bleibt

also nur der andere Ausweg, nämlich potenziell unendlich viele Äquivalenzklassen zu erzeugen, indem jeder ausführbare Player-Code neben dem jeweils verwendeten *PIK* noch eine individuelle Zufallszahl enthält. In diesem Fall liefert jede Prüfsummenbildung unterschiedliche Werte, auch bei Hashfunktionen. Dies kann im Prinzip durch eine Online-Installation der Software erreicht werden, bei der mindestens eine Komponente von einem Installationsserver an den Client übertragen wird, oder durch Berechnung eines Computer-spezifischen Parameters, der während der Installation berechnet wird.

6 Performance des *Fair-DRM*-Verfahrens

Zur Steigerung der Performance eines DRM-Systems in der praktischen Implementierung ist es wünschenswert, getrennte Server für die Bereitstellung des eigentlichen Content und die Vergabe der Lizenzen vorzusehen. Auf diese Weise kann der verschlüsselte (und umfangreiche) Content auf viele Server redundant verteilt bzw. automatisch zwischengespeichert (*cached*) werden, während die zugehörigen Lizenzen zur Nutzung des Content auf einem dedizierten Server generiert und an den Nutzer ausgegeben werden.

Das in diesem Beitrag vorgeschlagene Verfahren sieht vor, dass die einzelnen Shares $S_1, ..., S_n$ nur in die Lizenzinformation eingefügt werden, so dass die gewünschte Trennung von Content- und Lizenzserver erzielt werden kann. Die Zusammenführung der Daten und ggf. auch die Einbettung eines Wasserzeichens erfolgt ausschließlich auf Client-Seite, so dass keine weiteren Server-Kapazitäten benötigt werden.

Die Nutzung des Secret-Sharing-Verfahrens macht es für den Client erforderlich, ein Polynom an zufälliger Stelle auszuwerten, wenn die Nutzung des Content erfolgen soll. Die zentrale Instanz ihrerseits muss zur erfolgreichen De-Anonymisierung das gewählte Polynom anhand der übermittelten Werte bestimmen (dies ist jedoch nur erforderlich, nachdem der Threshold überschritten wurde). Die beiden genannten mathematischen Operationen sind in wenigen Langzahl-Arithmetik-Rechenschritten durchführbar [11], so dass keine Performance-Engpässe auf Client- oder Server-Seite zu erwarten sind.

7 Ausblick

Unser Modell sieht drei verschiedene zentrale Instanzen (Pseudonymisierungsstelle, Lizenzstelle, Verwertungsstelle) vor, um den Kunden gegen einen Missbrauch des Verfahrens durch betrügerische Anbieter zu schützen. In der Praxis wird ein Anbieter versuchen, alle drei Funktionen gleichzeitig auszuüben (Kosten, Kundenbindung). Hier könnte sich der Einsatz von *Verifiable Secret Sharing*-Verfahren [2, 12] lohnen, die auch schon bei elektronischen Münzen [8] zur Lösung des *doublespending*-Problems eingesetzt wurden.

Unser Ansatz ist zu unterscheiden von *Digital Fingerprints*, wie sie z. B. in dem maßgeblichen Artikel von Boneh und Shaw [1] definiert sind. Bei diesem Ansatz erhält jeder Kunde eine individuell markierte Kopie des Content, so dass beim Auffinden einer illegalen Kopie ermittelt werden kann, von wem diese Kopie stammt. Man unterscheidet hier also zwischen legalen Kopien (in der Regel nur lokal verwendet, unbegrenzte Anzahl) und illegalen Kopien (aufgefunden im Internet, illegale Zweitvermarktung, schon die erste solche Kopie ist illegal). Wir treffen diese Unterscheidung dagegen allein anhand der Anzahl der Kopien: 7 sind erlaubt, die 8. ist nicht erlaubt. Um dies zu erreichen müssen Fair-DRM Endgeräte vertrauenswürdig sein. Eine offene Frage ist, ob sich dieses notwendige Vertrauen reduzieren läßt.

Ein weiter interessanter Aspekt betrifft die Integration des Fair-DRM Ansatzes mit Broadcast Encryption Techniken [9], die verwendet werden, um kompromittierte Endgeräte aus dem System auszuschließen. Die dadurch gewonnene Erneuerbarkeit (*renewability*) des Systems würde zu dessen langfristigen Sicherheit beitragen.

Literatur

[1] Dan Boneh and James Shaw. Collusion-secure fingerprinting for digital data. In Don Coppersmith, editor, *Advances in Cryptology – CRYPTO '95*, volume 963 of *Lecture Notes in Computer Science*, pages 452–465. International Association for Cryptologic Research, Springer-Verlag, Berlin Germany, 1995.

[2] Jan Camenisch, Ueli M. Maurer, and Markus Stadler. Digital payment systems with passive anonymity-revoking trustees. In *ESORICS*, pages 33–43, 1996.

[3] Ingemar Cox, Matthew L. Miller, and Jefferey A. Bloom. *Digital Watermarking*. Morgan Kaufmann Publisher, 2002.

[4] DVB Project. The dvb glossary. `http://www.dvb.org/documents//dvb_glossary.pdf`, August 2004.

[5] Rüdiger Grimm and Patrick Aichroth. Privacy protection for signed media files: a separation-of-duty approach to the lightweight drm (lwdrm) system. In *MM&Sec '04: Proceedings of the 2004 multimedia and security workshop on Multimedia and security*, pages 93–99. ACM Press, 2004.

[6] Moving Picture Experts Group. Multimedia framework (mpeg-21), iso/iec 21000. `http://www.chiariglione.org/mpeg/`, 2003.

[7] The International DOI Foundation (IDF). Digital object identifier. `http://www.doi.org`, 2004.

[8] Dennis Kügler and Holger Vogt. Off-line payments with auditable tracing. In *Financial Cryptography – FC 2002*, Lecture Notes in Computer Science. Springer-Verlag, 2002.

[9] Dalit Naor, Moni Naor, and Jeff Lotspiech. Revocation and tracing schemes for stateless receivers. pages 41–62.

[10] remus Projekt. remus web-dok. 2/2004. Online verfügbar von `http://remus.jura.uni-sb.de/web-dok/2004/20040002.html`.

[11] Adi Shamir. How to share a secret. *Communications of the ACM*, 22(11):612–613, November 1979.

[12] Markus A. Stadler. Publicly verifiable secret sharing. *Lecture Notes in Computer Science*, 1070:190, 1996.

[13] Trusted Computing Group (TCG). `https://www.trustedcomputinggroup.org/`.