SYMBIOSE VON KRYPTOGRAPHIE UND DIGITALEN WASSERZEICHEN: EFFIZIENTER SCHUTZ DES URHEBERRECHTES DIGITALER MEDIEN

MARTIN KUTTER UND FRANCK LEPRÉVOST

ZUSAMMENFASSUNG. In diesem Artikel stellen wir ein komplettes und funktionelles System vor, welches zum Schutz des Urheberrechtes digitaler Daten benutzt werden kann. Dies wird erreicht durch die Kombination von Kryptographie und digitaler Waßerzeichen. Das Konzept des vorgestellten Systems ist anwendbar für verschiedene Multimediadaten, obwohl wir uns in diesem Artikel auf digitale Bilder konzentrieren.

1. EINLEITUNG

Mit dem digitalen Zeitalter und der weltweiten Verteilung digitaler Daten über das Internet wird der Schutz des Urheberrechts immer wichtiger. Bis zum jetzigen Moments gibt es noch keine Technologien die es ermöglichen das Urheberrechte von Multimediadaten, zum Beispiel Bilder, Video und Ton, in effizienter Weise zu schützen.

Beginn der neunziger Jahre wurde die Idee der digitalen Waßerzeichen [1, 2, 16, 15] geboren. Das Konzept der digitalen Waßerzeichen besteht darin, Information, zum Beispiel in Form einer Nummer, so in Zieldaten einzufügen, daß die Veränderungen in der Zieldatei nicht wahrnehmbar sind. Zusätzlich besteht das Ziel darin, daß die eingefügte Information robust ist. In anderen Worten heißt das, wenn die Daten modifiziert werden, zum Beispiel durch verlustbehaftete Kompreßion, sollte es immer noch möglich sein, die eingefügte Information zu extrahieren. Für eine Übersicht der Technologien für digitaler Waßerzeichen verweisen wir auf die Publikation von Swanson et al [14]. Anfangs wurde geglaubt, daß der ausschließliche Gebrauch der digitale Waßerzeichen auch für den Schutz des Urheberrechtes benützt werden könne. Craver et al [3] zeigten dann jedoch, daß diese ursprüngliche Annahme für die meisten Verfahren nicht gilt. Zum einen besteht das Problem darin, daß nicht bewiesen werden kann, wer zuerst ein digitales Waßerzeichen eingefügt hat, und zum anderen fehlt meistens ein rechtlich gültige Verbindung zwischen dem Waßerzeichen und der Zieldaten.

2. KONZEPT

2.1. **Idee.** Wie bereits in der Einleitung erwähnt, um digitale Waßerzeichen zum Schutz des Urheberrechtes zu gebrauchen müßen mindestens die Folgenden zwei Bedingungen erfüllt sein: 1) das Waßerzeichen und die Zieldaten müßen eine eindeutige Verbindung aufweisen; 2) das Waßerzeichen muß eine zeitliche Identifikation ermöglichen. Bevor wir näher auf diese Bedingungen und unsere Lösung eingehen, stellen wir das angewandte Konzepte des digitalen Waßerzeichens vor. Das digitale Waßerzeichen besteht in unserem Fall aus einer Nummer *DI*. Wir schlagen die Benützung des sogenannten *License Plates* (ISO Multimedia License Plate ISO 10919-4) vor. Das *License Plate* ist eine 64-Bit Nummer und enthällt Informationen wie Herkunftsland (16-Bit), Eigentümer Identifikation (16-Bits) und Daten Identifikation (32-Bits). Es ist natürlich klar, daß diese Eigentümer Identifikation rechtlich beglaubigt sein muß und eindeutig ist.

Das Waßerzeichen wird in die Zieldateien eingefügt unter Verwendung eines geheimen Schlüßels. Ein eingefügtes Waßerzeichen kann nur unter Verwendung des selben geheimen Schlüßels extrahiert werden. Damit die Sicherheit das Wasserzeichens gewährleistet werden kann, muß der Schlüßel ein Mindestlänge aufweisen. Es ist klar, dass ein 32–Bit Schlüßel heutzutage nicht genügt. Wir schlagen deshalb die Benützung eines 128–Bit Schlüßels vor. Um das Waßerzeichen mit der Zieldatei eindeutig zu verbinden, machen wir den Schlüßel von der Zieldatei abhängig unter der Verwendung einer *Hash*–Funktion. Im weiteren werden dann das Waßerzeichen bei einer rechtlich beglaubigten Person (Trusted Third Party – TTP) registriert um den Zeitpunkt des digital Waßerzeichenprozeßes festzuhalten.

2.2. **System Beschreibung.** Wir stellen nun das Gesamtschema vor. Alle Teile kryptogaphischer Natur werden im Abschnitt 3 erläutert, und die Technologie des verwendeten digitalen Waßerzeichen–Verfahren ist in Abschnitt 4 erklärt.

Stichwörter. Digitale Waßerzeichen, Digitale Signaturen, Watermarking, Fingerprinting, Kryptographie, Zeitstempel.

Alice möchte das Urheberrecht ihres Bildes I schützen bevor sie jenes auf dem Internet freigibt. Sie öffnet zuerst eine sichere Verbindung mit der TTP unter Verwendung des *Diffie–Hellman* Protokolls. Dieses Protokoll ermöglicht den sicheren Austausch eines Geheimnis K auf einem öffentlichen Kanal. Wenn Alice viele Bilder schützen möchte, zum Beispiel ein Video, ist es von Vorteil die Protokolle von Secret Key Kryptographie und Public Key Kryptographie zu kombinieren. Schematisch gesehen wird das Geheimnis K zwischen Alice und der TTP als Schlüßel für ein symmetrisches Verschlüsselunsprotokoll (wie zum Beispiel *IDEA* oder der zukünftige Standard *AES*) eingesetzt. Der Vorteil liegt darin, daß symmetrische Protokolle generell bis zu 1000 mal schneller sind als ihre asymmetrischen Homologe.

Nach dem Aufbau der sicheren Verbindung werden folgende Aktionen ausgeführt:

- Alice übermittelt der TTP den Hash–Wert $H_I = h(I)$ des Bildes I.
- TTP erzeugt einen Digital Zeitstempel [13] ZS_{H_I} und sendet jenen zu Alice.
- Alice erzeugt den Schlüßel [1] $K_P = h(ZS_{H_I}||h(ps_{Alice}))$ für das digitale Wasserzeichen–Verfahren, wobei ps_{Alice} der private Schlüßel von Alice repräsentiert. Der Operator $||$ hängt die beiden Teile aneinander.
- Alice generiert das License Plate LP für das Bild mit seiner Identifikationsnummer, dem Ursprungsland und einer Bildidentifikationsnummer. Danach schützt sie das Bild indem sie mittels eines digitalen Wasserzeichen–Verfahren das LP in das Bild einfügt, unter Verwendung des Schlüßels K_P.
- Nun übermittelt Alice das LP der TPP.
- TPP registriert LP, ZS, H_I und den Zeitpunkt und gibt Alice die Freigabe für das Bild (1–Bit).

Es ist natürlich auch denkbar, daß Alice der TTP das geschützte Bild sendet. Nach erfolgter Registrierung legt die TTP das Bild auf dem Internetserver.

Bemerkung: mit dem beschriebenen Verlauf ist zu keinem Zeitpunkt ein ungeschütztes Bild im Umlauf.

3. Public Key Kryptographie, Secret Key Kryptographie und Erzeugung der digitalen Unterschrift

Das Ziel dieser Sektion ist es, alle oben erwähnten Sicherheitskomponenten zu beschreiben: Secret Key Kryptographie, Public Key Kryptographie und die Erzeugung von zufälligen Zahlen. Siehe [9] für einen Überblick der aktuellen Situation dieses Gebiets.

3.1. Hash-Funktionen.

Eine *Hash-Funktion* ist eine Funktion, die als Input eine Datei F irgendwelcher Länge nimmt, und deren Ergebnis ein Wort W einer beliebigen kurzen Länge ist. Die Erzeugung des Worts W muß sehr schnell sein. Eine Hash-Funktion muß auch zusätzliche Eigenschaften erfüllen. Es muß z.B. praktischerweise unmöglich sein, die Hash-Funktion umzukehren: gegeben seien ein Wort W und eine Hash-Funktion H, es ist unmöglich innerhalb einer vernünftigen Zeit eine Datei F zu finden, so daß $H(F) = W$ gilt. MD5 (Output = 128 bits), SHS-1 oder RIPEMD-160 (Output = 160 bits) sind unter anderem heutzutage häufig benutzte Hash-Funktionen.

3.2. Secret Key Kryptographie.

Wir befassen uns hier nur mit den sogenannten Block-Ciphers (im Gegenteil zu den Stream-Ciphers).

DES ("Data Encryton Standard") ist der am häufigsten benutzte symmetrische Secret-Key Algorithmus. Das NIST (National Institute of Standards and Technology) hat ihn 1977 als FIPS 46-2 (Federal Information Processing Standard) definiert. Es ist ein Protokoll mit 16 Kreisen, hat ein Schlüssellänge von 56–Bits und eine Blocklänge von 64 bits. DES wurde jedoch kürzlich von der Electronic Frontier Foundation ([4]) geknackt, und es ist deshalb heutzutage sehr empfehlenswert einen anderen Algorithmus zu benutzen. In diesem Rahmen hat das NIST die Entwicklung des Nachfolgestandards AES ("Advanced Encryption Standard") gefordert ([12]). Zur Zeit sind noch 15 Angebote im Rennen (von anfangs 21). Der gewählte AES wird im 2001 ein FIPS.

Man benutzt auch sehr oft den Algorithmus IDEA. Insbesondere findet man diesen Algorithmus im Produkt PGP (Pretty Good Privacy, siehe unten).

[1] Wie bereits erwähnt, um sogenannte *Bruce Force* Attacken zu vermeiden sollte der Schlüßel eine Mindestlänge von 128–Bits haben. Kurzfristig gesehen ist eine Länge von <60–Bits, und längerfristig eine Länge von <90–Bits auf jeden Fall zu vermeiden. Um auf der sicheren Seite zu sein, ist somit eine Hash–Funktion mit einem 128–Bit Resultat von Vorteil.

3.3. **Public Key Kryptographie: IEEE-P1363.** Das Problem mit den symmetrischen Algorithmen besteht darin, daß die zwei Teilnehmer sich für einen gemeinsamen Schlüssel einigen müssen. Public Key Kryptographie kann dieses Problem lösen. Als Beispiel beschreiben wir das Diffie-Hellman Protokoll für die elliptischen Kurven mit ungeraden Charakteristiken (siehe beispielsweise [11]):

Sei $p > 2$ eine Primzahl der Länge 160 Bits und E eine über $\mathbf{F}_p$ definierte kryptographisch–geeignete elliptische Kurve. Grob gesagt, jede elliptische Kurve E ist kryptographisch–geeignet, wenn die Spur S des Frobenius $\neq 0, 1$. Die Kurven mit $S = 0, 1$ entsprechen sogenannten *supersingulären* bzw. *anomalen* elliptischen Kurven. Man definiert $G = E(\mathbf{F}_p)$ als die Gruppe der rationalen Punkte der Kurve über $\mathbf{F}_p$ unter der Voraussetzung, daß diese Gruppe monogen ist . Das heißt, daß $G =< P >$ für ein Element $P \in G$ gilt. Veröffentlicht sind p, E, P. Zwei Teilnehmer Alice und Bob wählen zufällige **geheime** Zahlen x_A, x_B der Länge 160 Bits (dafür können sie z.B. einen pseudo–zufällige Zahlen Generatoren (siehe Abschnitt 3.5) anweden. Sie können aber auch die Hash-Funktion eines Gedichtes anwenden: in diesem Fall brauchen sie nirgendwo den geheimen Schlüssel zu speichern!). Alice (bzw. Bob) veröffentlicht $y_A = x_A P$ (bzw. $y_B = x_B P$) in einer Datei (wie ein Telefonbuch). Das Geheimnis K, das die beiden Teilen, ist $K = x_B y_A = x_A y_B = x_A x_B P$. Um dieses Geheimnis K zu erfahren, müßte ein (passiver) Angreifer in der Lage sein, das sogenannte Diskret-Log Problem in G zu lösen. Wenn E und p richtig gewählt sind, kostet diese Arbeit ungefähr soviel MIPS/year wie das RSA-1024 Bits zu knacken. Man kann andere Gruppen G benutzen, wie z.B. $\mathbf{F}_p^*$ oder die Gruppe der rationalen Punkten der Jacobischen Varietät einer hyperelliptischen Kurve über $\mathbf{F}_p$.

Bemerkung: Es wird einen Standard für Public Key Kryptographie (RSA, elliptische Kurven über endlichen Körpern der Charakteristik $p \geq 2$) geben: IEEE-P1363 ([6]). Franck Leprévost gehört zu den Experten, die von der IEEE-Standard Association berufen wurden, um diesen internationalen Standard zu überprüfen.

3.4. **Secret Key Kryptographie und Public Key Kryptographie zusammen.** Wie oben beschrieben gibt mehrere Produkte auf dem Markt die Secret Key Kryptographie und Public Key Kryptographie zusammen mischen. Ein Beispiel ist *PGP* (Pretty Good Privacy): dieses Produkt kombiniert RSA mit IDEA.

3.5. **Pseudo–Zufälliger Zahlen Generator.** Man benutzt oft die sogenannten Pseudo-zufällige Zahlen Generatoren (PZZG), z.B. für die Erzeugung von geheimen Schlüsseln. Dafür gibt es einen Standard: Der FIPS-186 (siehe beispielsweise [10]). Anderseits kann man auch die Ergebnisse von Hash-Funktionen anwenden, um pseudo–zufällige Zahlen zu erzeugen.

4. Digitales Wasserzeichen

In dem hier vorgestellten System benützen wir ein digitales Waßerzeichen Verfahren das von Kutter [8, 7] entwickelt wurde. Das Verfahren beruht auf *Spread Sprectrum* Modulation im blauen Farbbereich des Bildes. Jedes Bit b_i, $i = \{1, \cdots, N\}$, des N–Bit Waßerzeichen wird durch eine zweidimensionale Funktion $p_i(x, y)$ repräsentiert. Das Waßerzeichen w ist gegeben durch die Summe aller Funktionen, $w = \sum_i p_i$. Das Einfügen des Waßerzeichen in das Bild geschieht dann durch komponentenweise Addition:

$$B(x, y) \leftarrow B(x, y) + w(x, y),$$

wobei $B(x, y)$ den Blauwert des Bildes an der Position (x, y) repräsentiert. Die zweidimensionalen Funktionen p_i sind definiert als:

$$p_i(x, y) = b_i \alpha(x, y) \phi_i(x, y).$$

$\phi_i(x, y)$ ist eine zweidimensionale Modulationsfunktion, $\alpha(x, y)$ eine zweidimensionale Funktion mit dem Zweck das Waßerzeichen so anzupaßen, daß es visuell nicht wahrnehmbar ist, und b_i ist eine Projektion des Bitwertes von $\{0, 1\}$ nach $\{-1, 1\}$. Die Modulationsfunktionen werden generiert mittels zweidimensionalen pseudo–zufälligen Sequenzen S_i welche durch den geheimen Schlüßel P_K initialisiert werden. Damit das sichtbare Rauschen besser kontrollierbar ist, dürfen sich die Sequenzen nicht überschneiden, das heißt $S_i \cap S_j = \emptyset, \forall i \neq j$. Zusätzlich führen wir die Dichte D ein. Die Dichte definiert den Anteil aller Position im Bild, welche für das Einfügen des Waßerzeichens benützt werden. Die Modulationsfunktionen sind nun definiert als:

$$\phi_i(x, y) = \begin{cases} s_i(x, y) & \text{wenn } (x, y) \in S_i \\ 0 & \text{ansonsten} \end{cases}$$

Die statistische Verteilung der $s_i(x,y) \in \{-1,1\}$ ist einheitlich und hängt auch vom Schlüßel ab. Ein Problem ist es nun, die Sequenzen S_i so zu generieren, daß sie kryptographisch sicher sind. Wir beginnen mit der Initialisierung des pseudo–zufälligen Nummerngenerator (siehe Abschnitt 3) mit dem Schlüßel P_k. Danach wird ein zweidimensionales Feld zick–zack förming durchlaufen. Für jede Position werden dann drei Zufallsnummern generiert, $Z_1 \in [0,1]$, $Z_2 \in \{1,\cdots,N\}$ und $Z_3 \in \{-1,1\}$. Wenn nun für die Position (x,y) die Bedingung $Z_1 < D$ erfüllt ist, dann wird (x,y) zu S_{Z_2} hinzugefügt und $s_{Z_2}(x,y) = z_3$ generiert. Mit dieser Methode können zweidimensionale Funktionen beliebiger größe unter Einhaltung der gewünschten Dichte und der erforderliche kryptographischen Sicherheit generiert werden. Das lesen des eingefügten Wasserzeichens geschieht durch Korrelation zwischen einer Estimation des eingefügten Wasserzeichens und den einzelnen Modulationsfunktionen.

5. Diskussion

Das vorgestellte System ermöglicht es durch die Kombination von Kryptographie und digitaler Wasserzeichen das Urheberrecht von digitalen Medien zu schützen. Die vorgeschlagene Implementation beschränkt sich auf digitale Bilder, jedoch ist das Konzept ohne weiteres auch für Video und Ton anwendbar. Ähnliche Systeme wurden bereits Vorgeschlagen bei Herrigel[5], jedoch ist unser System weniger komplex, macht Gebrauch von neuen Standarden und benötigt kürzere Wasserzeichen, was wiederum die Robustheit erhöht.

Literatur

[1] G. Caronni. Ermitteln unauthorisierter verteiler von maschinenlesbaren daten. Technical report, ETH Zürich, Switzerland, August 1993.

[2] G. Caronni. Assuring ownership rights for digital images. In *Proceedings VIS 95, Session "Reliable IT Systems"*. Vieweg, 1995.

[3] S. Craver, N. Memon, B. Yeo, und M. Yeung. Can invisible watermarks resolve rightful ownerships. In *Proceedings of Electronic Imaging*, Band 3022, Seiten 310–321, San Jose, CA, USA, Februar 1997. SPIE.

[4] Electronic Frontier Foundation. Cracking DES, Secrets of Encryption Research, Wiretap Politics & Chip Design. *O'Reilly*, 1998.

[5] A. Herrigel. Copyright protection for multimedia–data based on asymmetric cryptography techniques. In *Spie Proceedings onf Electronic Image Capture and Publishing*, Mai 1998.

[6] IEEE-P1363. *http://grouper.ieee.org/groups/1363/index.html.*

[7] M. Kutter. Watermarking resisting to translation, rotation, and scaling. In *Proceedings of SPIE International Symposium on Voice, Video, and Data Communications*, November 1998.

[8] M. Kutter, F. Jordan, und Frank Bossen. Digital watermarking of color images using amplitude modulation. *Journal of Electronic Imaging*, 7(2):326–332, April 1998.

[9] F. Leprévost. AES und IEEE-P1363, die kryptographischen Standards des 21. Jahrhunderts. In *Tagungsband des 6. Deutschen IT-Sicherheitskongress des BSI*, Mai 1999.

[10] A. J. Menezes, P. C. van Oorschot, und S. A. Vanstone. Handbook of Applied Cryptography. *CRC Press series on discrete mathematics and its applications*, 1996.

[11] A. J. Menezes. Elliptic Curve Public Key Cryptosystems. *Kluwer Academic Publishers*, 1993.

[12] NIST AES Home Page. *http://csrc.nist.gov/encryption/aes/aes_home.htm.*

[13] B. Schneider. *Applied Cryptography.* Wiley, 1996.

[14] M.D. Swanson, M. Kobayashi, und A.H. Tewfik. Multimedia data embedding and watermarking techniques. *Proceedings of the IEEE*, 86(6):1064–1087, Juni 1998.

[15] K. Tanaka, Y. Nakamura, und K. Matsui. Embedding secret information into a dithered multilevel image. In *Proceeding of the 1990 IEEE Military Communications Conference*, Seiten 216–220, September 1990.

[16] K. Tanaka, Y. Nakamura, und K. Matsui. Embedding the attribute information into a dithered image. *Systems and Computers in Japan*, 21(7), 1990.

M. K.: EPFL, DE-LTS, Ecublens, CH-1015 Lausanne

E-mail address: Martin.Kutter@epfl.ch, http://ltswww.epfl.ch:1248/kutter

F. L.: CNRS Paris und Technische Universität Berlin, Fachbereich Mathematik MA 8-1, Strasse des 17. Juni 136, D-10623 Berlin

E-mail address: leprevot@math.tu-berlin.de, http://www.math.tu-berlin.de/∼leprevot