TeleTrusT
Pioneers in IT security.

TeleTrusT-interner Workshop

Nürnberg, 21./22.06.2012

Maik Schott
Otto-von-Guericke-Universität Magdeburg

„Standardisierte Implementierung von elektronischen Signaturen in digitalen Bildern zum Nachweis von Integrität, Authentizität und Urheberschaft“

Standardisierte Implementierung von elektronischen Signaturen in digitalen Bildern zum Nachweis von Integrität, Authentizität und Urheberschaft

- Projekt im Rahmen des Programms „Innovation mit Normen und Standards“
  - Programmkoordination: Deutsches Institut für Normung (DIN)
  - Projektleitung: Otto-von-Guericke-Universität Magdeburg

INS INNOVATION MIT NORMEN UND STANDARDS

Gefördert durch:
Bundesministerium für Wirtschaft und Technologie
aufgrund eines Beschlusses des Deutschen Bundestages

Problemstellung

- Problemstellung:
  - Leichte Manipulierbarkeit/Fälschbarkeit von digitalen Bildern
  - Eingeschränkte Beweiskraft von Bildern
- Aber, Verwendung bei/für:
  - Leistungsbeschreibungen
  - Rechtsangelegenheiten, z.B. Nachweis von Versicherungsschäden
  - Nachweis von Urheberschafts-ansprüchen
  - Verbrechensaufklärung

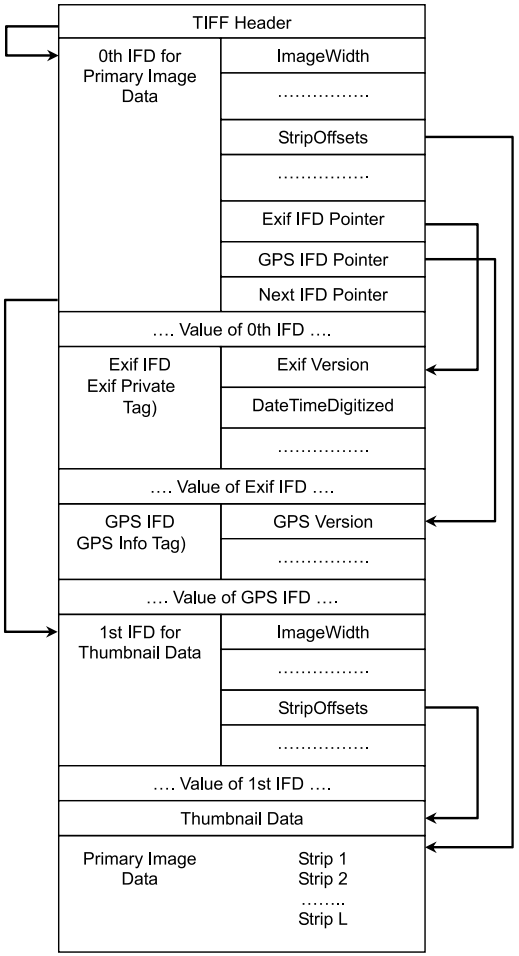
Projektziel

- Daher:
  - Entwicklung einer Methodik zur Verifikation der Integrität und Authentizität von Bildern
    - → digitale Signaturen
  - Erstellung bereits in Digitalkamera
  - Anforderungen an das Verfahren zur Erstellung, Speicherung und Überprüfung
  - Erstellung einer DIN-Norm

Grundlagen
Kamera-Bildformate: Raw

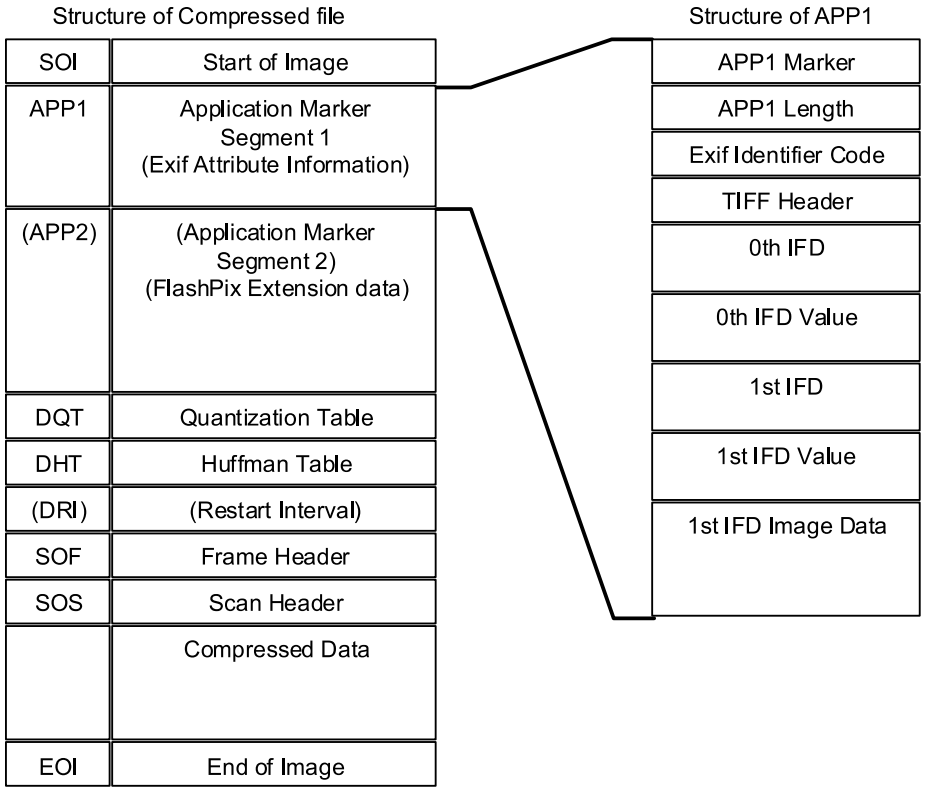
- Raw:
  - Adobe: .dng
  - Canon: .crw, .cr2
  - Epson: .erf
  - Fujifilm: .raf
  - Hasselblad: .3fr, .fff
  - Kodak: .dcr, .kdc, .raw
  - Konica Minolta: .mrw
  - Leica: .raw
  - Nikon: .nef
  - Olympus: .orf
  - Panasonic: .raw, .rw2
  - Pentax: .pef
  - Samsung: .srw
  - Sigma: .x3f
  - Sony: .arw, .srf, .sr2
  - …
  - aber: alles TIFF-Varianten mit propietären Erweiterungen
- JFIF/JPEG

Grundlagen
Kamera-Bildformate: Raw/TIFF



- Bytereihenfolge: Little-Endian *und* Big-Endian
- Besteht aus Verzeichnissen (IFD)
  - Jedes Top-Level-IFD *üblicherweise* ein Bild
  - Verzeichniseinträge = Tags
  - Unterverzeichnisse
- Für Raw-Bilder: meist Untermenge TIFF/EP

Grundlagen
Kamera-Bildformate: JFIF/JPEG und EXIF



- JPEG = Bilddatenformat
- JFIF = Dateiformat
- Besteht aus Einzelsegmenten
- Digitalkameras verwenden Teilmenge Exif(-JPEG)

Grundlagen
Kamera-Bildformate: EXIF-TIFF

- APP1-Segment: Metadaten = TIFF (!)
  - IFD0: allgemeine Bildmetadaten
  - IFD1: EXIF-Metadaten (Magic Word: EXIF)
  - IFD2: GPS-Metadaten (Magic Word: GPS)
- APP2-Segment: Flashpix-Erweiterung für eingebettetes Audio
- APP14-Segment: Photoshop-Metadaten
- …
- Segmente können mehrfach vorkommen mit verschiedenen Semantiken und meist TIFF

Grundlagen
Kamera-Bildformate: Stereoskopie, Panorama, usw.

- Stereo Still Image Format
  - Zwei JPEG-Dateien (.jpg und .ssi)
  - Metadaten in APP3 (Magic Word: Stim)
- Multi-Picture Format
  - Konkatenierte JPEG-Dateien
  - Metadaten in APP2 (Magic Word: MPF)

- Flexibles Verfahren benötigt um alle Fälle abdecken zu können

Metadaten für Bildsignatur

Metadaten für Bildsignatur (Signatur-IFD) Überblick

Feldname	Tag	Typ	Anzahl	Optional (J/N)
TIFF-Erweiterung				
ImageSignatureOffset	????	LONG	1	N
Bildsignatur-IFD				
Version	0	BYTE	4	N
PreviousIFD	1	UNDEFINED	variabel	J
PreviousIFDPath	2	UNDEFINED	variabel	J
OriginalByteOrder	3	SHORT	1	N
HashAlgorithm	4	ASCII	variabel	N
LocalHashPath	5	SHORT	variabel	N
HashPathsGroups	6	UNDEFINED	variabel	N
SignatureFormat	7	SHORT	1	N
TimeSources	8	ASCII	variabel	J
ExternalSignedFiles	9	UNDEFINED	variabel	J
Signature	8000_h	UNDEFINED	variabel	N

Metadaten für Bildsignatur (Signatur-IFD) Tags (I)

- `PreviousIFD` und `PreviousIFDPath`:
  - Bei Signaturauffrischung oder Änderung der signierten Daten, Erstellung neuer Signatur-Metadaten
  - Link auf frühere Fassung
- `OriginalByteOrder`
  - Bildprogramme können beim Abspeichern die Bytereihenfolge des APP1-TIFF-Segments ändern
  - Bytereihenfolge zum Signierungszeitpunkt muss gespeichert werden

Metadaten für Bildsignatur (Signatur-IFD) Tags (II)

- `HashAlgorithm`
  - Verwendeter Hash-Algorithmus als ASN.1-Objekt-ID
- `LocalHashPath`
  - Liste der Metadatenfelder dieses Signatur-IFDs die signiert werden sollen
- `HashPathGroups`
  - Angabe der Blöcke und Metadaten innerhalb der Datei die signierte werden sollen
  - Jede Gruppe wird separat signiert

Metadaten für Bildsignatur (Signatur-IFD) Tags (III)

HashPath	Bedeutung
file:Dateiname.erw	Die Datei „Dateiname.erw“ soll gehasht werden
imagedata:	Die Bilddaten sollen gehasht werden (nicht anwendbar auf TIFF-Dateien, da diese mehrere besitzen können)
jpeg:FFE2	die JPEG-APP2-Segmente sollen gehasht werden
jpeg:FFE2.MPF	das JPEG-APP2-Segment mit der ID „MPF<NUL>“ soll gehasht werden
0394	Hashe das IFD-Feld mit dem Tag 0394 (Copyright) in IFD0
0106/03D0	Betrachte das IFD-Feld mit dem Tag 0106_h (ExifIFDPointer) als Zeiger auf ein Offset an dem sich ein Unter-IFD (hier das Exif-IFD) befindet und hashe dort das Feld $03D0_h$ (DateTimeOriginal)
0106	Hashe das IFD-Feld mit dem Tag 0106_h, d.h. nur das Offset würde gehasht werden, nicht jedoch die Daten die sich an jenem Offset befinden.
0106/	Hashe alle Felder im durch 0106_h bezeichneten Unter-IFD (hier das Exif-ID). Dies erfolgt nicht rekursiv, d.h. für Felder in diesem IFD die auf einen
.next/0201[2000]	Besuche das nächste IFD von IFD0, d.h. IFD1, und betrachte das Tag 0201_h (JPEGInterchangeFormat) als Zeiger auf ein Offset von Daten mit einer Länge von 2000 Bytes (hier speziell ein JPEG-Thumbnail)
.next/0201	Würde demgegenüber nur das Offset hashen, nicht die jedoch die Daten die sich an jenem Offset befinden.
0111[0:1000]<TAB>0111[1:2000]	Die Bilddaten können in mehrere Strip genannte Teile aufgespalten sein, die sich an unterschiedlichen Positionen befinden. Diese werden durch ein Tag 0111_h (StripOffsets) referenziert, das eine Liste von Zeigern auf Offsets enthält (hier: 2). Jedes Strip kann dabei eine unterschiedliche Größe haben, so dass für jedes Offset die Größe separat angegeben werden muss. Hier ist der erste Strip 1000 Bytes groß und der zweite Strip 2000 Bytes.
jpeg:FFE3.Stim/000A	Gehe in APP3-Tag mit der ID Stim<NUL> (Stereo-Still Image) und hashe dort das Feld $000A_h$ (RepresentativeImage).

Metadaten für Bildsignatur (Signatur-IFD) Tags (IV)

- `SignatureFormat`:
  - DER-kodiert nach CMS/PKCS#7
  - XML-Signatur
- `TimeSources`
  - Vertrauen in eine Signatur hängt auch von Qualität und Vertrauenswürdigkeit der Zeitquelle ab
  - Da verschiedene denkbar und Vertrauen nicht maschinell bestimmbar → Freitext mit vordefinierten Werten
    - „Internal“, „Internal (fixed)“, „GPS“, „Radio“ („Radio (DCF77)“), „TSP“

Metadaten für Bildsignatur (Signatur-IFD) Tags (V)

- `ExternalSignedFiles`
  - Externe Dateien die zum Bild gehören, aber ihre eigene Signatur mitbringen
- `Signature`
  - Signatur (DER-CMS, XMLSig)
  - Anforderungen (Muss):
    - mindestens einen Signierer, d.h. ein SignerInfo-Objekt
    - Zertifikate des verwendeten privaten Schlüssels
    - die signierte Nachricht (d.h. die Hashes) im EncapsulatedContentInfo-Objekt
    - der Zeitpunkt der Signatur als SigningTime-Objekt als signiertes Attribut
  - Anforderungen (Soll):
    - ein RevocationInfoChoices-Objekt für Zertifikatswiderrufe
    - ein TimeStampToken-Objekt als unsigniertes Attribut für einen TSP-Zeitstempel
    - Padding, z.B. (RSASSA-)PSS

Bestehende Verfahren von Kameraherstellern

Bildsignaturen in der Industrie
Canon

- Bildsignaturverfahren: OSK-E3 mit Bildsignatur ODD (Original Decision Data)
- Funktionsweise:
  - Einteilung in Bildregionen
  - MD5-Hash über jede Region und SHA1-HMAC über alle Hashs
  - Schlüssel gleich für jede Kamera eines Modells
  - Version 3:
    - Salted-HMAC (Salt = Schließanzahl des Verschlusses)
  - Version 3 Revision:
    - SHA-256 und Salt aus mehreren IDs
- Version 2 und 3 wurden 2010 gebrochen, die Revision teilweise

Bildsignaturen in der Industrie
Nikon

- Nikon Image Authentication
- SHA1 je über Bilddaten und Metadaten
- Verschlüsselung mit RSA-1024
- Speicherung als Exif-Tag
- Privater Schlüssel unsicher gespeichert
  - Kompromittierung 2011

Bildsignaturen in der Industrie
Kodak und Kappa

- Kodak Picture Authentication
  - SHA1 über Bild- und Metadaten
  - Verschlüsselung mit DSA
  - Schlüsselpaar wird einmalig in der Kamera generiert
  - Kompromittierung nicht bekannt
- Kappa: PS/DX 4S/40S
  - SHA-512/RSA-1024 (Datenblatt-Stand: Nov. 2011)
  - Signaturchip (BSI-zertifiziert, CC EAL5+)

Schlüsselverwaltung und -anforderungen

Schlüsselverwaltung und -anforderungen
Allgemeine Fragen

- Wer oder was ist der Signierer bzw. auf wen ist das Zertifikat ausgestellt?
  - = wer oder was soll authentifiziert werden?
  - Kamera oder Person?
  - Einfluss darauf, ob qualifizierte elektronische Signatur möglich
- Interne oder externe Schlüsselspeicherung?

Schlüsselverwaltung und -anforderungen
Identifizierung der Kamera (I)

- Identifizierung der Kamera
  - pro Kamera(exemplar bzw. -modell) nur ein Zertifikat
  - Vorteil:
    - Sicherstellung das Bild von diesem Kameraexemplar bzw. –model aufgenommen wurde
  - Nachteil:
    - Kameras sind keine natürlichen oder juristischen Personen nach SigG
    - → Keine qualifizierte elektronische Signatur möglich

Schlüsselverwaltung und -anforderungen
Identifizierung der Kamera (II)

- Interne Schlüsselspeicherung
  - Festverdrahtung nicht wünschenswert, da damit Austausch abgelaufener, ungültiger oder kompromittierter Schlüsse, Zertifikate und Algorithmen unmöglich
  - Kameras können zudem längere Zeit auf Lager liegen

Schlüsselverwaltung und -anforderungen Identifizierung der Kamera (III)

- Externe Schlüsselspeicherung
  - Sicherer Speicher notwendig → z.B. Smartcard
  - Problemverlagerung
  - Vorteile:
    - Smartcards (nach DIN V 66291) besitzen bereits Kryptofunktionen und erfüllen Anforderungen an SigG/SigV → Komplette Auslagerung der Kryptofunktionen auf Smartcards möglich
    - Schlüssel-/Algorithmenaustausch leicht möglich
  - Nachteile:
    - (Authentifizierung mit PIN entspräche Benutzerauthentifizierung)
    - Schlüssel-/Algorithmenaustausch leicht möglich → z.B. von fremder Kamera

Schlüsselverwaltung und -anforderungen
Identifizierung der Nutzer

- Identifizierung der Nutzer
  - Kamera wird mit Schlüsseln der Benutzer bestückt
  - externe oder interne Speicherung
  - Vorteile:
    - Nachvollziehbarkeit wer ein Photo aufgenommen hat
    - Qualifizierte elektronische Signaturen möglich
  - Nachteile:
    - Siehe vorige Folie (insbesonders bei externer Speicherung)

Schlüsselverwaltung und -anforderungen Umsetzung

- Beide Ansätze möglich, da CMS-Signaturen mehrere Signierer unterstützen
- privater Schlüssel einer Kamera muss per TPM (vorzugsweise mit Kryptofunktionen) gespeichert werden
- Ausgestellung durch Hersteller
- Aber: Kompromittierung lässt sich nicht verhindern, wenn gesamte Funktionalität und Schlüssel beim Nutzer liegen

Schlüsselverwaltung und -anforderungen
Fazit

- Bedrohungsverhinderung / Lehren aus bestehenden Bildsignaturverfahren:
  - Soll: Schlüsselpaar für jedes Kameraexemplar
    - Muss: nur für eine beschränkte Anzahl an Exemplaren
  - Kompromittierung eines Schlüssels kompromittiert nicht das gesamte System
  - Auslieferung nur mit sicheren Algorithmen (Algorithmenkatalog)