

Content-Verwertungsmodelle und ihre Umsetzung in mobilen Systemen

Digital Rights Management

Vorlesung im Sommersemester an der
Technischen Universität Ilmenau

von

Privatdozent

Dr.-Ing. habil. Jürgen Nützel

Vorstand der

4FriendsOnly.com Internet Technologies AG

JN (at) 4FO (dot) DE



4FriendsOnly.com
Internet Technologies AG

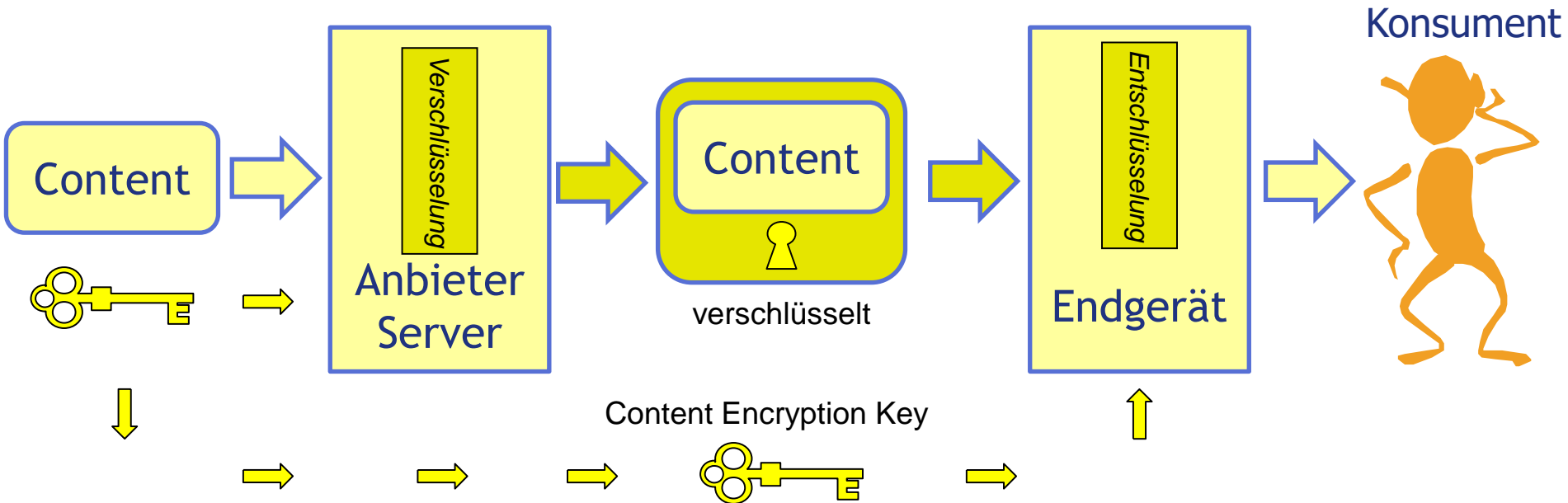
Diese Folien und weitere Informationen unter:
www.juergen-nuetzel.de/content_verwertungsmodelle_mobile_umsetzung.html

Überblick

- Symmetrische Verschlüsselung von Content
- Lizenzen (oder Rechteobjekte)
- Public-Key-Kryptographie
- Zertifikate
- Kryptographische Hash-Funktion
- RSA-Verfahren
- DRM-Referenz-Modell

Verschlüsselung von Content

- Content-Daten werden verschlüsselt



- Anbieter verteilt nur verschlüsselte Nutzdaten
- Verschlüsselung und Entschlüsselung mit dem gleichen Schlüssel, z.B. mit AES (Advanced Encryption Standard)
- Schlüssel wird getrennt und geheim übermittelt

Symmetrische Verschlüsselung [1]

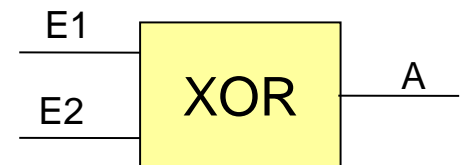
- Einfacher Algorithmus: Bitweise Addition (XOR)

- Beispiel: Verschlüsselung
- Content-Daten = 11 = 1011
- Schlüssel = 9 = 1001

Nutzdaten E1: 1011
Schlüssel E2: 1001
Verschlüsselte -----
Nutzdaten A: 0010
= 2

- Beispiel: Entschlüsselung
- Verschl. Content-Daten = 2 = 0010
- Schlüssel = 9 = 1001
- Content-Daten = ?

E1	E2	A
0	0	0
0	1	1
1	0	1
1	1	0



Symmetrische Verschlüsselung [2]

- XOR: One-Time-pad

- XOR: Block-Schlüssel

Content-Daten
Bild des Buchstaben X
Länge 25 Bit

1	0	0	0	1
0	1	0	1	0
0	0	1	0	0
0	1	0	1	0
1	0	0	0	1

XOR-Schlüssel
mit der gleichen
Länge: 25 Bit

0	1	1	0	1
1	1	0	1	0
0	0	0	1	0
0	0	1	0	1
0	1	0	1	0

Verschlüsselte
Content-Daten

1	1	1	0	0
1	0	0	0	0
0	0	1	1	0
0	1	1	1	1
1	1	0	1	1

Mehrfache Anwendung
eines XOR-Schlüssels
fester Länge: 5 Bit

0	1	1	0	1
0	1	1	0	1
0	1	1	0	1
0	1	1	0	1
0	1	1	0	1

Verschlüsselte
Content-Daten

1	1	1	0	0
0	0	1	1	1
0	1	0	0	1
0	0	1	1	1
1	1	1	0	0

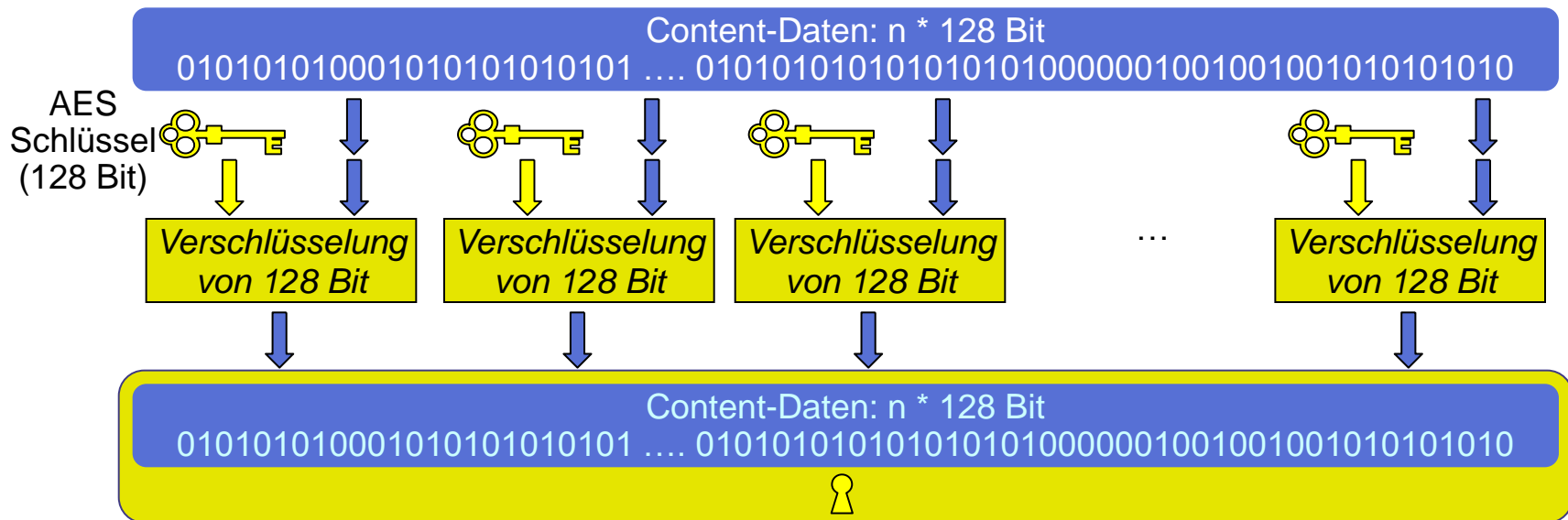
1	0	0	0	1
1	1	1	0	0
0	1	1	0	1

Aber Klartext-Angriff:
Sind nur 5 Bit der geheimen
Content-Daten bekannt,
so kann der 5 Bit Schlüssel
Berechnet werden. Damit
kann der gesamte Content
entschlüsselt werden.

Symmetrische Verschlüsselung [3]

- Blockweise Verschlüsselung

- XOR kann in der Praxis nur einmalig (one-time pad) angewendet werden (Klartext-Angriff gelingt)
- Gute symmetrische Verfahren wie AES ermöglichen die wiederholte Anwendung des Schlüssels (Erklärung!)
- Blocklänge und Schlüssellänge z.B. 128 oder 256 Bit

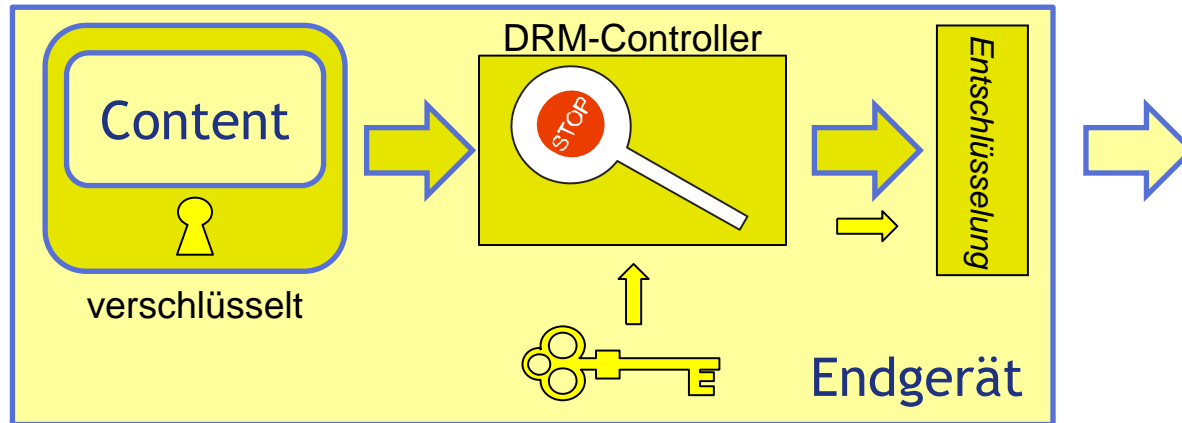


AES - Advanced Encryption Standard

- Standard nach dem Verfahren von Rijndael
 - In 2000 Sieger eines Wettbewerbs der NIST
 - 128 Bit Blocklänge mit 128, 192 oder 256 Bit Schlüssel
 - Realisierung in Hardware und Software sehr schnell
 - Je nach Schlüssellänge: 10, 12 oder 14 Runden
 - Frei von Patenten und unentgeltlich nutzbar
- Tutorial hier: www.realtec.de/privat/arbeiten.shtml

Kontrolle über den Schlüssel

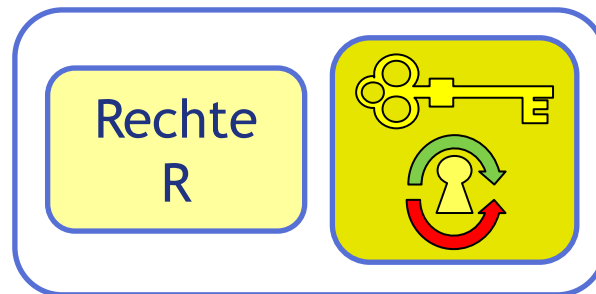
- Schlüssel wird im Endgerät kontrolliert



- Der DRM-Controller kontrolliert die Verwendung des Schlüssels
- Schlüssel muss vor dem Nutzer verborgen bleiben
- DRM-Controller darf nicht vom Nutzer umprogrammiert werden

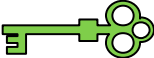
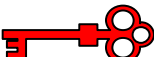
Lizenzen (oder Rechteobjekte)

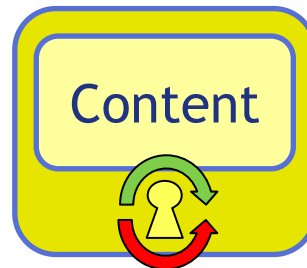
- Lizenzen enthalten den Schlüssel und eine Rechtebeschreibung
 - Verschlüsselte Content-Daten sind ohne Lizenz wertlos
 - Rechtebeschreibung legt die zulässige Nutzungsart (z.B. abspielen) und Nutzungsdauer (z.B. 3 mal oder in den nächsten 48h) fest
 - Verschlüsselte Nutzdaten können und sollen kopiert werden
 - Lizenzen sind an das Endgerät gebunden (Weitergabe unmöglich oder wirkungslos)



Public-Key-Kryptographie

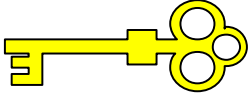
- Grundprinzip

- Es gibt zwei Schlüssel (=Schlüsselpaar)
- Was mit dem einen verschlüsselt wird kann nur mit dem anderen entschlüsselt werden (=asymmetrisch)
- Der eine Schlüssel heißt öffentlich: Public Key 
- Der andere Schlüssel heißt privat: Private Key 



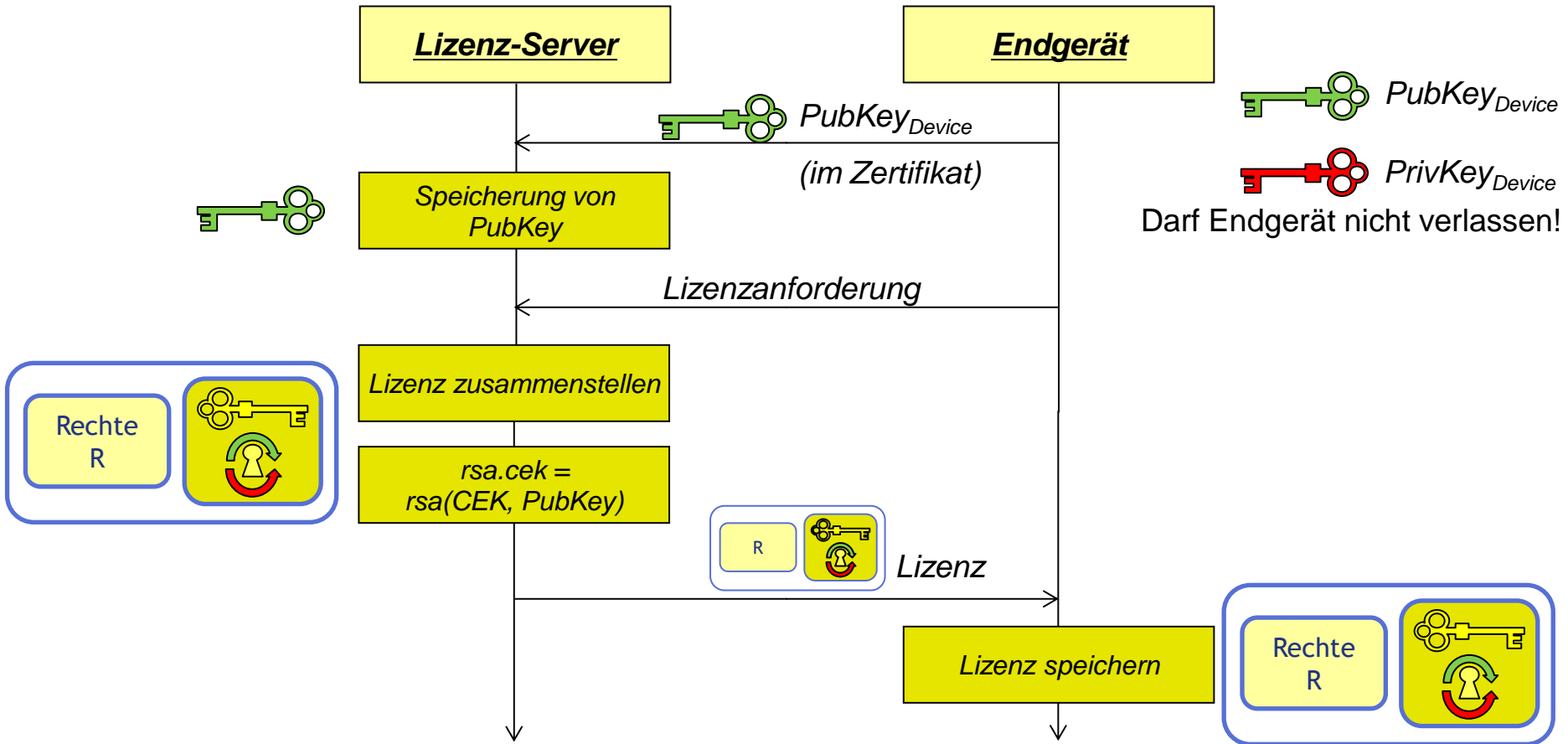
Mit Public Key
verschlüsselt

Anwendungen bei DRM

- Geheime Übertragung des CEK: 
 - Endgerät fordert von einem Lizenz-Server den passenden Schlüssel für die Content-Daten an.
 - Lizenz-Server verschlüsselt den CEK (Content Encryption Key) mit dem öffentlichen Schlüssel des Endgerätes

Geheime Übertragung des CEK

- Sequenz-Diagramm



Allgemeine Anwendungen [1]

- **Beispiel: Verschlüsselte E-Mail oder SSL**
 - Sender einer geheimen E-Mail (z.B.) verschlüsselt diese mit dem öffentlichen Schlüssel des Empfängers
 - Da asymmetrische Verfahren langsam sind, wird der Inhalt (Nutzdaten) mit einem schnellen symmetrischen Algorithmus (z.B. AES) verschlüsselt. Der symmetrische Schlüssel wird mit dem öffentlichen Schlüssel des Empfängers verschlüsselt. (= hybrid)

Allgemeine Anwendungen [2]

- Digitale Signatur

- Integrität von Nachrichten und Authentizität von Kommunikationspartner muss sichergestellt werden
- Ausgetauschte (nicht verschlüsselte) Dokumente (z.B. Rechte in der Lizenz, oder eine App) dürfen nicht verändert werden
- Beteiligte Kommunikationspartner (z.B. Endgerät und Server) müssen sich über den jeweils anderen sicher sein können
- Einsatz von Zertifikaten (z.B. nach X.509)
- Prinzip vereinfacht: Der Sender überträgt das Dokument doppelt. Einmal unverschlüsselt. Ein zweites mal mit seinem privaten Schlüssel verschlüsselt.
- Besser: Sender verschlüsselt mit seinem privaten Schlüssel nur eine Prüfsumme (Hash-Wert, Details später) des Dokumentes.

Authentizität durch Zertifikate

- Wie kann der Lizenz-Server dem Endgerät vertrauen?
 - Dem öffentliche Schlüssel alleine darf man noch nicht trauen
 - Von einer offiziellen Instanz (CA - Certification Authority) ausgestellte Zertifikate bieten Abhilfe
- Was ist ein Zertifikat (nach X.509)?
 - Der öffentliche Schlüssel und
 - Ein Datensatz über den Besitzer des Schlüssel
 - Beides zusammen wurde von einer CA digital signiert
 - Das Zertifikat der CA kann beigefügt sein
- Zertifikatsketten
 - Der Aussteller des Zertifikates besitzt ein eigenes Zertifikat

Kryptographische Hash-Funktion [1]

- ... wird für die digitale Signatur benötigt
 - Wird auch Streuwertfunktion genannt
 - Die Hash-Funktion ist eine Funktion, die zu einer Eingabe aus einer (üblicherweise) großen Quellmenge eine Ausgabe aus einer (im Allgemeinen) kleineren Zielmenge (die Hash-Werte, meist eine Teilmenge der natürlichen Zahlen) erzeugt.
- Dabei muss gelten:
 - Kollisionsfreiheit
 - Es darf nicht effizient möglich sein, zwei Quellelemente mit demselben Hash-Wert zu finden.
 - Unumkehrbarkeit
 - Zu der Funktion gibt es keine effizient berechenbare Umkehrfunktion, mit der es möglich wäre, für ein gegebenes Zielelement ein passendes Quellelement zu finden

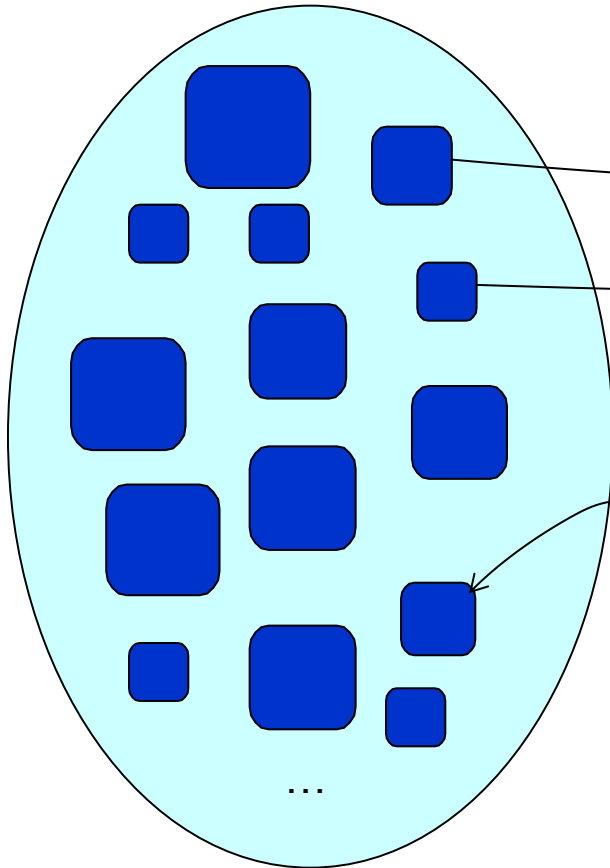
Kryptographische Hash-Funktion [1]

- ... wird für die digitale Signatur benötigt
 - Wird auch Streuwertfunktion genannt
 - Die Hash-Funktion ist eine Funktion, die zu einer Eingabe aus einer (üblicherweise) großen Quellmenge eine Ausgabe aus einer (im Allgemeinen) kleineren Zielmenge (die Hash-Werte, meist eine Teilmenge der natürlichen Zahlen) erzeugt.
- Dabei muss gelten:
 - Kollisionsfreiheit
 - Es darf nicht effizient möglich sein, zwei Quellelemente mit demselben Hash-Wert zu finden.
 - Unumkehrbarkeit
 - Zu der Funktion gibt es keine effizient berechenbare Umkehrfunktion, mit der es möglich wäre, für ein gegebenes Zielelement ein passendes Quellelement zu finden

Kryptographische Hash-Funktion [2]

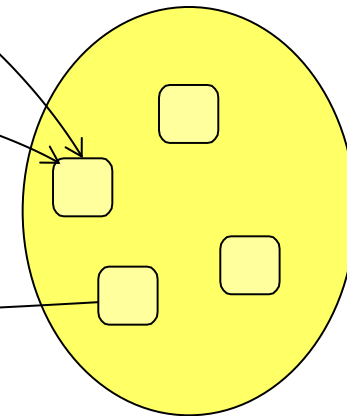
Quellmenge

Elemente können beliebige große Dokumente sein



Zielfmenge

Elemente sind Zahlen (z.B. 160 Bit)



Wenige Kollisionen

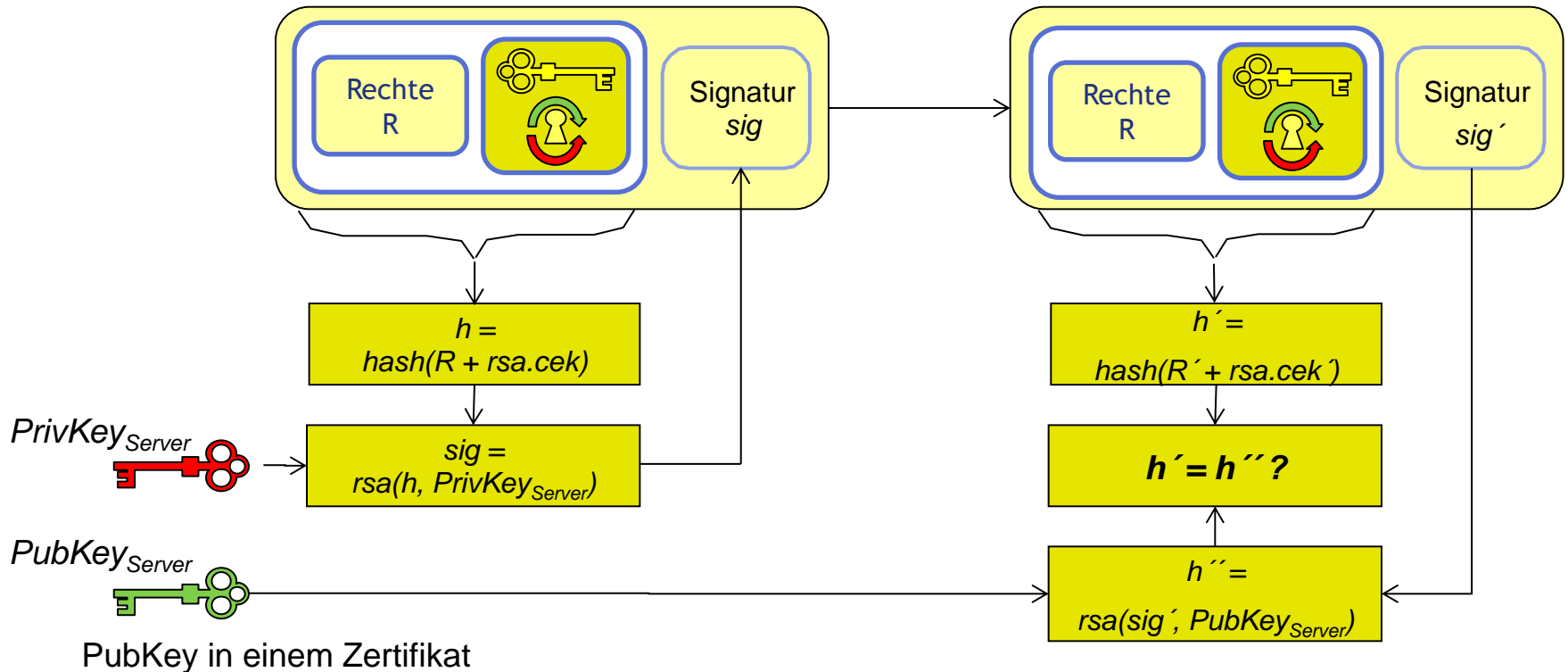
Unumkehrbarkeit

z.B. SHA256 oder MD5



Sicherung der Integrität & Authentizität

- ... der Lizenz durch digitale Signatur

Mit dem Private Key des Lizenz-Servers wird ein über die Rechte und Schlüssel errechneter Hash-Wert verschlüsselt



Das RSA-Verfahren

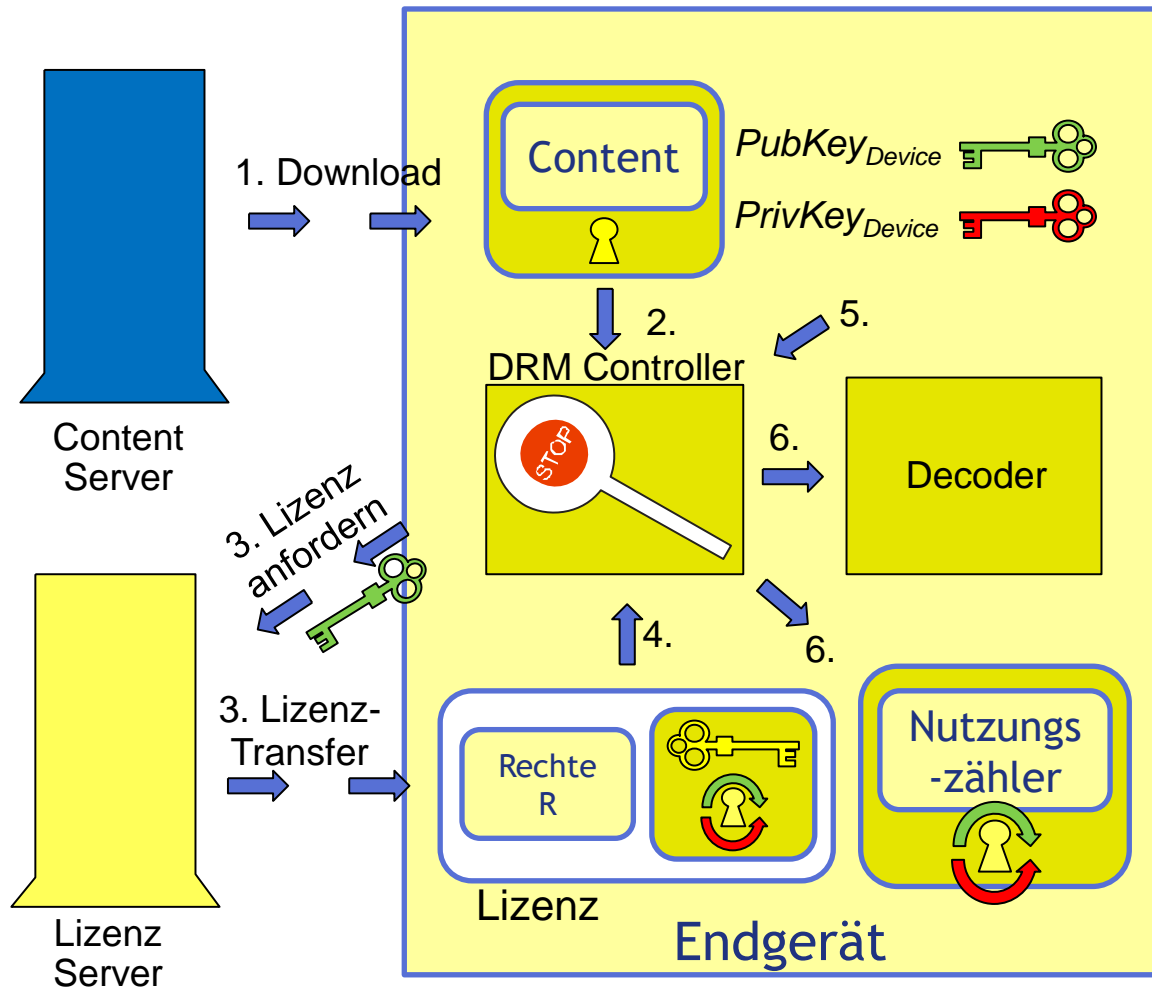
- RSA (Rivest, Shamir, Adleman)
- Idee: Multiplikation ist einfach. Die Umkehrung, die Faktorisierung, ist schwierig
- Nehme zwei etwa gleich lange Primzahlen: p und q
(Beispiel: $p=11$, $q=13$)
- Berechne $n = p \cdot q$ ($n = 143$)
- Berechne $\varphi(n) = (p-1) \cdot (q-1)$ ($=120$)
- Wähle e (23) mit $\text{ggT}(e, \varphi(n)) = 1$
- Berechne d so, dass $e \cdot d \equiv 1 \pmod{\varphi(n)}$ gilt,
 $e \cdot d = k \cdot \varphi(n) + 1$
($d = 47$ mit $k = 9$)
- Public Key: (n, e) (143, 23) 
- Private Key: (n, d) (143, 47) 
- Verschlüsselung mit Public Key:
 $C \equiv K^e \pmod{n}$
 $2 \equiv 723 \pmod{143}$
- Entschlüsselung mit Private Key:
 $K \equiv C^d \pmod{n}$
 $7 \equiv 247 \pmod{143}$
- Damit n im praktischen Anwendungsfall nicht in p und q faktorisiert werden kann, muss n aktuell eine 1024 bis 2048 bit lange Zahl sein !!

Ein 1024 bit RSA-Schlüsselpaar

- $n =$
151117088560515543543583509112099097962003663556607044995
537346278481881284114992437661794727300361132467861422736
444261887801298612841233509930473048074186874048225374579
833810514168500718834144275902347213750223932752522075922
296123467024433402797906496071473309891192170853187418104
035346071158728163015279
- $e = 65537$ (fast immer gleich, damit öffentlich)
- $d =$
284377840489622381024919574878239494509268227511731493039
040861139893043910613882498997957546392931297851005527657
440389682508112332296402471576746586187747826535831626494
005466669015130749910740387014636406766438399916759421316
216341535604374551482729571882157734877943009193315973748
61906548026671091235657

Referenz-Modell für DRM-Systeme

• DRM-Referenz-Modell



1. Download des Contents
2. Content wird geöffnet
3. DRM-Controller fordert eine Lizenz an
4. Lizenz wird geöffnet
5. Der private Geräteschlüssel wird benötigt
6. Nutzungszähler werden geprüft und angepasst. Entschlüsselter Content wird decodiert

Zusammenfassung

- **Content ist symmetrisch verschlüsselt**
 - Content-Daten sind ohne Schlüssel (CEK) wertlos
 - Im unverschlüsselten Teil steht die Adresse des Lizenz-Servers
- **Schlüssel wird in der Lizenz transportiert**
 - Lizenz enthält Rechtebeschreibung
 - Rechte werden im DRM-Controller ausgewertet
- **Asymmetrische Kryptographie**
 - Nachrichten (z.B. Rechtebeschreibungen) werden von beiden Seiten signiert
 - Zertifikate werden eingesetzt
 - Schlüssel (CEK) in der Lizenz wird vom Server mit dem öffentlichen Schlüssel des Endgerätes verschlüsselt
 - Der private Endgeräteschlüssel ist der Sicherheitsanker

Weitere Informationen [1]

- Jürgen Nützel: Die informatorischen Aspekte virtueller Güter und Waren, Oktober 2006 im Unversitätsverlag Ilmenau, www.juergen-nuetzel.de/habilitation.html
- Jürgen Nützel: Digital Rights Management (Seite 28 - 49), in Die Privatkopie, herausgegeben von Frank Fechner, 2007, Universitätsverlag Ilmenau, ISBN 978-3-939473-06-0, https://www.db-thueringen.de/receive/dbt_mods_00007543
- <http://de.wikipedia.org/wiki/One-Time-Pad>
- http://de.wikipedia.org/wiki/Advanced_Encryption_Standard
- Tutorial zu AES: www.realtec.de/privat/arbeiten.shtml
- Reinhard Wobst: Abenteuer Kryptologie. 3. Auflage, Addison-Wesley, München 2003

Weitere Informationen [2]

- http://de.wikipedia.org/wiki/Digitale_Signatur
- The Internet Society: RFC 3280 Internet X.509 Public Key Infrastructure, <http://www.ietf.org/rfc/rfc3280.txt>
- http://de.wikipedia.org/wiki/Digitales_Zertifikat
- <http://de.wikipedia.org/wiki/RSA-Kryptosystem>

Weitere Infos und Kontakt

Privatdozent Dr.-Ing. habil. Jürgen Nützel

JN (at) 4FO (dot) DE

www.juergen-nuetzel.de

www.4fo.de



4FriendsOnly.com
Internet Technologies AG