# Authentication

https://iuk.one/1033-1007

Clemens H. **Cap**
ORCID: 0000-0003-3958-6136

Department of Computer Science
University of **Rostock**
Rostock,Germany
**clemens.cap@uni-rostock.de**

Version 1

# Overview

1. General Remarks
2. Knowledge Factors
3. Inherence Factors
4. Possession Factors
5. Attribute Based Authentication

# 1. General Remarks

A few thoughts as an introduction to authentication.

1. General Remarks
2. Knowledge Factors
3. Inherence Factors
4. Possession Factors
5. Attribute Based Authentication

# What is Authentication?

## Definition

A person or a system (called prover) proves its identity or a property towards another system or person (called verifier).

**Core aspects:**

1. What is the property?
2. How is the binding of the property to the prover established?
3. How is the property proven?

# Examples of Properties

- **Name:** Maria Müller born Maier.
- **Pseudonym:** Snapchat User DonaldDuck.
- **Face:** Looks like person on reference photo nr. 23.
- **Finger print:** Looks like reference finger print 47.
- **Iris Structure:** Looks like reference iris nr. 99.
- **Genetic Code:** Has a particular genetic feature.
- **Key:** Owns a key which opens lock 24.
- **Passport:** Holds a passport with matching photo and fingerprint.
- **Club Id:** Holds a club id with number and name, but no photo.
- **Simcard:** Owns mobile with SIM-card no. 42 and knows unlock PIN.
- **App:** Owns mobile with unlocked banking app for account 23.
- **SMS:** Capable of receiving an SMS under 0171 2345 6789.
- **Email:** Capable of reading an email at `president@usa.gov` .
- **Account:** Capable fo posting a tweet at handle `@POTUS`.

# Identification and Property Binding

Who might be interested in **separating the binding?**

**Example 1:** Person, who is connected to a property wants to get rid of it.

- Robert Robber wants to prevent identification by fingerprints.
- Nora Nerd wants to view a pay-walled video without payment.

**Example 2:** Person, who is not connected to a property wants to obtain it.

- Bob Bad wants to obtain the key of the bank safe.
- Herbert Hacker wants to receive the baking PINs of Robert Rich on his mobile.

**Delegation:** A right can be delegated.
Bob shall execute company transactions on behalf of his boss Alice during her holidays.
Bob gets this right from Alice for some time but must prove being Bob.
Transaction is recorded as "executed by Bob on behalf of Alice".

# Three Factors of Authentication

**Knowledge Factors:** What I know:

- **Idea:** Directly connected with the brain of a person.
- **Attacks:** Intercepted upon input.
- **Problem:** Often gets written down.
- **Examples:** Passwords, answers to secret or contextual questions.

**Inherence Factors:** What I am:

- **Idea:** Directly connected with the body of a person.
- **Attacks:** Faked or forced signal rendering; systemic attacks.
- **Problem:** Comparison is “soft”.
- **Example:** Biometric features of my body.

**Possession Factors:** What I own.

- **Idea:** A thing which is hard to duplicate.
- **Attacks:** Can be stolen; can be duplicated nevertheless.
- **Problems:** Can be lost.
- **Examples:** Hardware Tokens, Software Tokens, Communications Paths, PIN/TAN-letters.

# Further Aspects

**Further categories:** Some talk about further categories.

**Distinguishability:** Categories not always clearly distinguishable.

**Motivation:**

- Individual factors have different attack modes.
- Combination of factors is more secure against a multivalued attack.
- Need appropriate handling and deployment.

# 2. Knowledge Factors

2.1. Managing a Password

2.2. Improving a Password

What is a knowledge factor of authentication?

1. General Remarks
2. Knowledge Factors
3. Inherence Factors
4. Possession Factors
5. Attribute Based Authentication

# Password Entropy

Humans are bad in generating entropy.

In a 3M password study, the letter e was used 1.5M times and the letter f was used 250K times. Only 8% of 34.000 passwords adhered to a mixed case, numbers & symbols policy.

## Cracking Entropy Spaces

- 29 bit: Recommended minimal entropy if online attacks may be expected.
- 56 bit: By special hardware in 1999 in one day (DES).
- 64 bit: By distributed.net in under 5 years.
- 96 bit: If substantial attacks by offline attacker may be expected.

## Too small entropy:

- Average 8-character password contains 18 bits of entropy
- Average user password contains 40 bits of entropy

# What is a good password?

**Questions:**

- Can the password be written on every national keyboard?
- Can the password be written without particular keyboard drivers?
- Did you memorize the password in connection with a keyboard layout?

# Secret Questions

**Idea:**

- Questions asked from a user with answers compared to enrolled answers.

**Problems:**

- Could be known to others.
- Must authenticate user on enrollment using other means.
  Thus: Can it qualify as independent factor?
- Does user remember the correct answer for strict string-comparison?
  Eg: First car? "A blue Toyota", "A Toyota Avensis", "An Avensis", "Avensis"

# Contextual Questions

**Idea:**

- Questions only the user could answer correctly and the verifier can verify based on her role in a transaction.
- Eg: What was the amount we billed you 18 months ago?

**Problems:**

- Answer also known to the verifier; allows misuse.

# Bruce Schneier Password Scheme

**Definition**

Pick a sentence and turn it into a password by picking initials.

**Example:**

- **Sentence:** I have been reading a lot about how passwords are no longer offering good security for me .
- **Password:** `Ihbralahpanlgs4m.`

Bruce Schneier: Chosing Secure Passwords

# How to Chose a Good Password

### Important

- Educate end users about passwords.
- Have and enforce a password policy.
- Have and enforce a verifier policy.

### Why?

- The weakest link is the human.
- Difficult or short validity passwords get written down.
- Attacker do no break passwords but the way people make up passwords.

### The typical password

- Consists of something pronounceable.
- Often plus a suffix (90% of the time) and/or a prefix (10% of the time)
- A set of 1.000 words plus 100 common suffixes breaks 24% of all passwords.

# Password Policies (1)

**Length:**

- At least 10 characters; up to 64 characters should be accepted by system.

**Forbidden Passwords:**

- Equal or reversal of user name.
- Dictionary words.
- Repetitive or sequential characters.
- Blacklisted passwords (eg: `qwertz`).
- Passwords which became public in previous breaches.

**Duration:** Debatable whether this is a good idea.

- Fact: By a 2017 study, 30% of users share passwords with 2 or more people.
- Pro: Passwords are shared and snooped.
- Con: Frequent change leads to breakable schemes or written down passwords
- NIST: Verifiers should not require passwords to be changed periodically.
  Verifiers shall force a change if there is evidence of a compromise.

# Password Policies (2)

**Reuse:**

- Fact: According to two 2017 studies, 87% or 92% of users reuse passwords.
- There is a market for cracked password files.
- Problem: Policy of no-reuse cannot be checked effectively.

**Syntactical Restrictions**

- At least 1 of upper, lower, number, punctuation, special character.
- Prevents most easy passwords.
- May become too complex for the user.
- Better: Minimal password length and testing against forbidden passwords.

# Key Stretching

**Background:**

- **Attack:** Attempt to try out the entire key space.
- **Defense:** Prolong the time to try out a single key.

**Method:**

- **Bad:** Store passwords and do a string comparison.
- **Better:** Store hashed passwords, compute hash and do string comparison.
- **Best:** Hash a password many times in a row (some 100.000 times).

**Effect:**

- Attacker is reduced to one attack every few seconds.
- User has virtually no disadvantage (downgrade fom 10[ms] to 3[s] log-on time.)

## Password Salting

**Background:** 3 Attacks:

- **Preimage:** Attacker has preimage of the hash (eg: rainbow table).
- **Duplication:** Oh, Alice has the same password hash as me.
- **Common:** Hash is in table of hashes of commonly used passwords.

**Method:**

- For every user-id $u$ storing a password $p$, also generate a nonce $n$.
- Instead of $h(p)$ store $n$ and $h(n, p)$ with the user.
- For verification, calculate $h(n, q)$ with $n$ from table and $q$ from prover.

**What about pepper?**

- Some sources also recommend pepper.
- Stackexchange arguing against pepper
- W article recommending pepper

# Task: Salting and Nonces

**We learned:** A salt must be a “nonce”.

**Questions:**

- What *exactly* is a nonce?
- Why is it important to use a nonce as salt?
- Which attacks would become possible when we reuse salt on a node or use the same salt for all users?

# 3. Inherence Factors

This is biometric authentication – and we have an entire separate unit on this.

1. General Remarks
2. Knowledge Factors
3. Inherence Factors
4. Possession Factors
5. Attribute Based Authentication

# 4. Possession Factors

4.1. TAN

4.2. Power of Proof

4.3. One Time Passwords

4.4. Hardware Token

Authentication by something you own and guard carefully.

1. General Remarks
2. Knowledge Factors
3. Inherence Factors
4. **Possession Factors**
5. Attribute Based Authentication

# Wide Range

Authentication by what I own.

**Example:** Something which cannot be (easily) duplicated.

- Cannot be duplicated as such.
  Eg: passport; USB token with on board generated secret.
- Cannot be duplicated against will of owner.
  Eg: USB token after download of keys, TAN generator after enrollment.

**Example:** Something which is typically safeguarded by the owner.

- (Separate) communication path.
- Can be duplicated but is safeguarded, eg: TAN letter.

# TAN Variants

**TAN:** User enters an arbitrary unused transaction number from a paper-based list.

**iTAN:** TAN plus verifier sends an index, prover returns the TAN indexed by that index.

**iTAN with confirmation:** iTAN plus verifier replies with a confirmation code also provided in the list and prover must check confirmation code.

**mTAN:** Verifier sends an SMS code to mobile of prover to be entered by the prover.

**photoTAN:** Verifier shows QR code, prover scans QR code with mobile and enters TAN generated by mobile into mobile app.

**photoTAN plus browser:** Verifier shows QR code, prover scans QR code with mobile and enters TAN generated by mobile into app on different hardware device.

**App TAN:** Verifier shows QR code, prover scans QR code with mobile and entry of TAN is done by internal app-to-app communication on mobile.

**TAN generator:** Specific hardware TAN generator scans QR code with camera and outputs a TAN, since it has no IP connection of its own.

# Task: TAN Attacks

**Task 1:** Find possible attacks against TAN variants.

**Task 2:** Analyze, which TAN variant is safe against which attack vector.

# Power of Proof (1)

**Problem:** In cases where the recipient of an authentication knows the expected answer the recipient can claim proper authentication.

**Example:** TAN-based authentication.

- Bank employee Mallory looks into TAN letter of Bob before sending it out.
- Mallory can make a transaction on behalf of Bob.

**Example:** Password-based authentication.

- Mark Zuckerberg has access to password database & server of facebook.
- He can authenticate as every one of his users.

**Counterexample:** Biometric Templates.

- Apple cannot biometrically unlock my iPhone, since they (claim) only to store templates of fingerprints.

**Counterexample:** Digital signature using asymmetric algorithm.

- If only Alice has access to the private key, no

# Power of Proof (2)

Power of proof towards **cooperating second party**
Works fine so long as

- I trust the second party not to abuse the scheme.
- Cooperating party properly safeguards my credentials.
- I have no dispute with that second party.

Power of proof **towards third party** works fine

- if an asymmetric algorithm is used
- only one party has access to the private key

# OTP: One Time Passwords

**Enrollment:**

- Device learns to which account it belongs.
- Usually this sets a secret key in the device.

**Requirements:** Attacker must not be able to

- repeat enrollment for her fake account
- intercept enrollment
- extract secret key from the device
- derive secret key from device output

**Derivation:**

- **Counter** based
- **Chain** based
- **Time** based

# OTP: Chain Based Mechanism

**Mechanism:**

- User generates random number $s$.
- User builds hash chain $h(\sigma), h(h(\sigma)), h(h(h(\sigma))), \dots h^500(\sigma)$.
- Server stores $s = h^{500}(\sigma)$
- User sends $u = h^{499}(\sigma)$.
- Server checks $s = h(u)$ and stores $u$ as new $s$.
- User sends $u = h^{498}(\sigma)$
- Server checks $s = h(u)$ and stores $z$ as new $s$.
- Before expiry of the chain: Negotiate new chain.

# OTP: Counter Based Mechanism

**Mechanism:**

- User and server share a common secret $\sigma$.
- $n$ is a counter for the logins.
- Expected password is $h(\sigma \cdot n)$.
- Problem: For some reason, counter at user and counter at server get out of synch.
- Solution: Resynchronization mechanism using look-ahead.

**Examples:**

- Standard: RFC-4226
- Product: YubiKey YubiKey Neo, Google Authenticator, Google Authenticator

# OTP: Time Based Mechanism: TOTP

**Mechanism:**

- User and server share a common secret $\sigma$.
- Expected password is $h(\sigma \cdot \lfloor \frac{\text{now} - t_0}{\tau} \rfloor)$.
- $h$ is a hash based message authentication.
- Parameters chosen as to get new password every 30 seconds.
- Problem: 30 second window for an attacker; need second factor.
- Problem: Needs a trustworthy time base.

**Examples:**

- Standards: RFC-6238
- Products: Google Authenticator, Google Authenticator

# General Idea

**Mechanism:**

- Token generates public, private key pair.
- Public key is sent to server where it is stored with the user id.
- Private key remains inside token.
- For verification: Server sends a nonce as challenge.
- Token replies with a signature as response.

**Problem:** Limited storage for private keys on token.

**Solution:**

- Delegate storage of private key to server as well.
- Server stores it in an encrypted form, only the token can decrypt.

**Examples:**

- Standards: FIDO Specification Overview
- Products: YubiKey

# Standards

**Proliferation** of standards:

| | |
|---|---|
| **TPM**: | Trusted Platform Module |
| **FIDO, FIDO2**: | Fast IDentity Online |
| **U2F**: | Universal Second Factor |
| **UAF**: | Universal Authentication Framework |
| **CTAP, CTAP2**: | Client to Authenticator Protocol |
| **WebAuthn:** | Browser based API |

**Problems:**

- Adoption in different speeds.
- Partial incompatibility of platforms.
- User does not have “one token for all”.

# Use Modes

**Token Activity:** Does user know when and why the device becomes active?

| | |
|---|---|
| Bad: | Interaction confirmed only on browser/mobile UI. |
| Better: | Interaction confirmed on hardware token. |
| Best: | Interaction details shown on browser/mobile UI and hardware token. |

**Usage modes:**

| | |
|---|---|
| **Stationary:** | Permanently stays in the device as second factor. |
| **Mobile:** | User takes it with her. |

# 5. Attribute Based Authentication

Proving attributes instead of identities.

1. General Remarks
2. Knowledge Factors
3. Inherence Factors
4. Possession Factors
5. Attribute Based Authentication

# Attribute Authentication

**As a person:**

- Proves that the user logged on as `donald` knows the password of that account.
- Organizational frameworks ensure that `donald` is

```
Donald Duck, born 1. 1. 1934 in Wise Little Hen, living in Duckburg,
Calisota, US, social security number 47-11.
```

**In a property:**

- Often not a person but a property is required.
- Eg: Is entitled to open this door.
- Eg: Is at least 18 years of age.
- Eg: Is owner of the website `iuk.one`.

# Proof of Website Ownership

**Example:**

- Google provides website crawler and search information to site owners.
- How does Google authenticate site ownership?

**Methods:**

| | |
|---|---|
| **File upload:** | Upload file with a name provided by Google to website. |
| **DNS edit:** | Add specific TXT or CNAME record to the DNS record. |
| **HTML tag:** | Insert specific META-tag into file index.html. |
| **Analytics code:** | Inserting specific Google analytics code. |

# Proof of Domain Ownership (1)

**EV: Extended Validation Certificates**

- Highest level of organizational security.

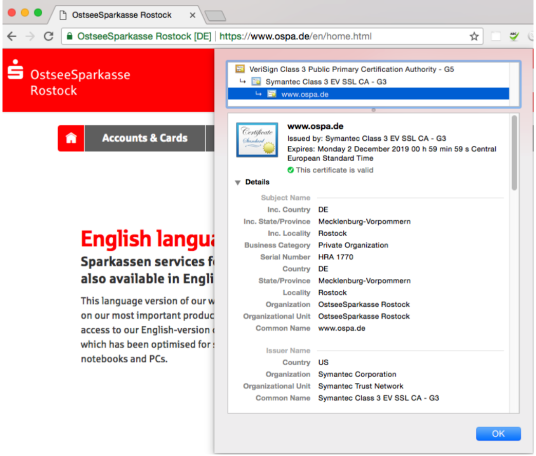
**Verification** may comprise:

- Legal, physical and operational existence of the entity.
- Checking official records and governmental lists of organizations.
- Entity has exclusive right to use the domain.
- Entity has properly authorized the issuance of the EV SSL Certificate.
- Entity is registered with Dun & Bradstreet
- Confirmation letter by financial or legal institutions.

**Provides:**

- Special trust labeling in the browser.
- More details in the certificate for display in the browser.
- Certificate should support OCSP status protocol.

# Example: Certificate of Domain Ownership



OstseeSparkasse Rostock
OstseeSparkasse Rostock [DE] | https://www.ospa.de/en/home.html
OstseeSparkasse
Rostock
Accounts & Cards
OK

English langu
Sparkassen services f
also available in Engli

This language version of our w
on our most important produc
access to our English-version
which has been optimised for
notebooks and PCs.

VeriSign Class 3 Public Primary Certification Authority - G5
↳ Symantec Class 3 EV SSL CA - G3
↳ www.ospa.de

**www.ospa.de**
Issued by: Symantec Class 3 EV SSL CA - G3
Expires: Monday 2 December 2019 00 h 59 min 59 s Central European Standard Time
✓ This certificate is valid

▼ **Details**

| | |
|---|---|
| Subject Name | |
| Inc. Country | DE |
| Inc. State/Province | Mecklenburg-Vorpommern |
| Inc. Locality | Rostock |
| Business Category | Private Organization |
| Serial Number | HRA 1770 |
| Country | DE |
| State/Province | Mecklenburg-Vorpommern |
| Locality | Rostock |
| Organization | OstseeSparkasse Rostock |
| Organizational Unit | OstseeSparkasse Rostock |
| Common Name | www.ospa.de |
| Issuer Name | |
| Country | US |
| Organization | Symantec Corporation |
| Organizational Unit | Symantec Trust Network |
| Common Name | Symantec Class 3 EV SSL CA - G3 |

**Fig. 1:** Screenshot of OSPA certificate. In 2016 the browsers showed a more prominent markup for EV in the location bar; today this is displayed less prominently.

# Proof of Domain Ownership (2)

**OV: Organizational Validation Certificates**

- Medium level of organizational security.

**Verification** may comprise:

- Legal, physical and operational existence of the entity
- Identity of the entity matches official records

**Provides:**

- No specific trust labeling in the browser
- Name of the organization provided as domain owner

# Proof of Domain Ownership (3)

**DV: Domain Validation**

- Lowest level of organizational security.

**Provides:**

- No specific trust labeling in the browser.
- Organizational unit is "DV Validated SSL" or similar but not the name of organization.
- Most references go to the certificate authority.

# Task: Domain Verification

Check the current certificate of

- https://www.ospa.de/
- https://www.uni-rostock.de/
- https://iuk.one/
- https://www.spiegel.de/

**Tasks:**

- What is the identity of the domain owner?
- How do you know?
- What is the level of the domain verification?
- What is the quality of the certificate?
  For checking certificates, see 🔗 and 🔗

# Proof of Authorized Device

**Goals:**

- Prevent unqualified media players from playing protected media.
- Prevent duplication of media at the level of digital streams.
- Cannot protect against analogue gap.

**Task:** One device authenticates itself at another connected device

**Issue 1:** What does it mean to be a connected device?

- Man in the middle scenario.

**Issue 2:** Software players

- Many attacks succeeded in re-engineering a software player.

**Issue 3:** Kerchkoff principle

- Wealth of examples of crypto failings due to violation of Kerchkoff principle.

# Proof of Authorized Device, Examples

**CSS:** Content Scramble System for DVDs

- Introduced 1996.
- Compromised 1999 based on cryptanalysis of leaked source code.

**AACS:** Advanced Access Content System for HD DVD and Blueray

- 2005 released.
- 2006 first keys appear as extracted from software players.

**HDCP** High-bandwidth Digital Content Protection for Display Port, DVI and HDMI.

- 2001 broken but not published due to legal proceedings.
- 2010 master key is released.
- 2011 broken and published by Ruhr-Universität Bochum.

# Proof of Same Browser

**Goals:**

- Authenticate browser once, then recognize that it is the same.
- Track users while surfing the net.

**Techniques:**

- Bearer Token. (see different unit!)
- Certified public key.
- Browser Fingerprinting.

# Proof of Same Browser: Certified Public Keys

**Mechanism:**

- Browser generates a public, private key pair.
- Browser sends public key to server, maybe with additional authentication.
- Server returns certificate for public key.
- Browser stores private key in keystore.
- IndexedDB has special keystore where private keys can be specially protected.

# Proof of Same Browser: Browser Fingerprinting

**Cave:**

- Not cryptographically secured!
- Not a security feature but a tracking and privacy issue.

**Mechanism:** Checking browser properties.

- Installed plugins and fonts.
- Headers (languages, user agent, OS).
- Canvas and WebGL implementation properties.
- Graphics drivers specifics

- Article on Browser Fingerprinting
- Browser fingerprinting demo website
- Browser fingerprinting website focusing on browser headers
- Browser fingerprinting website focusing on graphics properties

**Conclusion:** Together with contextual information, more or less every browser is unique.

# Attribute and Identity Based Encryption

Two novel and completely different approaches in the wake of authentication.

**Identity-based** encryption:

- Public and private keys depend on identity string, eg. email or passport number.
- Makes public key infrastructure redundant.

**Attribute-based** encryption:

- Ciphertext and decryption key depend on attributes.
- Only users whose decryption key contain specific attributes can decrypt.
- Allows anonymous decryption and signatures based on attributes.

# Appendix

# Contents of Appendix

# List of Figures

# Terms of Use (1)

Die hier angebotenen Inhalte unterliegen deutschem Urheberrecht. Inhalte Dritter werden unter Nennung der Rechtsgrundlage ihrer Nutzung und der geltenden Lizenzbestimmungen hier angeführt. Auf das Literaturverzeichnis wird verwiesen. Das Zitatrecht in dem für wissenschaftliche Werke üblichen Ausmaß wird beansprucht. Wenn Sie eine Urheberrechtsverletzung erkennen, so bitten wir um Hinweis an den auf der Titelseite genannten Autor und werden entsprechende Inhalte sofort entfernen oder fehlende Rechtsnennungen nachholen. Bei Produkt- und Firmennamen können Markenrechte Dritter bestehen. Verweise und Verlinkungen wurden zum Zeitpunkt des Setzens der Verweise überprüft; sie dienen der Information des Lesers. Der Autor macht sich die Inhalte, auch in der Form, wie sie zum Zeitpunkt des Setzens des Verweises vorlagen, nicht zu eigen und kann diese nicht laufend auf Veränderungen überprüfen.

Alle sonstigen, hier nicht angeführten Inhalte unterliegen dem Copyright des Autors, Prof. Dr. Clemens Cap, ©2020. Wenn Sie diese Inhalte nützlich finden, können Sie darauf verlinken oder sie zitieren. Jede weitere Verbreitung, Speicherung, Vervielfältigung oder sonstige Verwertung außerhalb der Grenzen des Urheberrechts bedarf der schriftlichen Zustimmung des Rechteinhabers. Dieses dient der Sicherung der Aktualität der Inhalte und soll dem Autor auch die Einhaltung urheberrechtlicher Einschränkungen wie beispielsweise Par 60a UrhG ermöglichen.

Die Bereitstellung der Inhalte erfolgt hier zur persönlichen Information des Lesers. Eine Haftung für mittelbare oder unmittelbare Schäden wird im maximal rechtlich zulässigen Ausmaß ausgeschlossen, mit Ausnahme von Vorsatz und grober Fahrlässigkeit. Eine Garantie für den Fortbestand dieses Informationsangebots wird nicht gegeben.

Die Anfertigung einer persönlichen Sicherungskopie für die private, nicht gewerbliche und nicht öffentliche Nutzung ist zulässig, sofern sie nicht von einer offensichtlich rechtswidrig hergestellten oder zugänglich gemachten Vorlage stammt.

**Use of Logos and Trademark Symbols:** The logos and trademark symbols used here are the property of their respective owners. The YouTube logo is used according to brand request 2-9753000030769 granted on November 30, 2020. The GitHub logo is property of GitHub Inc. and is used in accordance to the GitHub logo usage conditions `https://github.com/logos` to link to a GitHub account. The Tweedback logo is property of Tweedback GmbH and here is used in accordance to a cooperation contract.

# Terms of Use (2)

**Disclaimer:** Die sich immer wieder ändernde Rechtslage für digitale Urheberrechte erzeugt für mich ein nicht unerhebliches Risiko bei der Einbindung von Materialien, deren Status ich nicht oder nur mit unverhältnismäßig hohem Aufwand abklären kann. Ebenso kann ich den Rechteinhabern nicht auf sinnvolle oder einfache Weise ein Honorar zukommen lassen, obwohl ich – und in letzter Konsequenz Sie als Leser – ihre Leistungen nutzen.

Daher binde ich gelegentlich Inhalte nur als Link und nicht durch Framing ein. Lt EuGH Urteil 13.02.2014, C-466/12 ist das unbedenklich, da die benutzten Links ohne Umgehung technischer Sperren auf im Internet frei verfügbare Inhalte verweisen.

Wenn Sie diese Rechtslage stört, dann setzen Sie sich für eine Modernisierung des völlig veralteten Vergütungssystems für urheberrechtliche Leistungen ein. Bis dahin klicken Sie bitte auf die angegebenen Links und denken Sie darüber nach, warum wir keine für das digitale Zeitalter sinnvoll angepaßte Vergütungssysteme digital erbrachter Leistungen haben.

Zu Risiken und Nebenwirkungen fragen Sie Ihren Rechtsanwalt oder Gesetzgeber.

Weitere Hinweise finden Sie im Netz hier und hier oder hier.

# Citing This Document

If you use contents from this document or want to cite it,
please do so in the following manner:

Clemens H. Cap: Authentication. Electronic document. https://iuk.one/1033-1007 19. 6. 2021.

**Bibtex Information:** https://iuk.one/1033-1007.bib

```
@misc{doc:1033-1007,
   author      = {Clemens H. Cap},
   title       = {Authentication},
   year        = {2021},
   month       = {6},
   howpublished = {Electronic document},
   url         = {https://iuk.one/1033-1007}
}
```

**Typographic Information:**
Typeset on June 19, 2021
This is pdfTeX, Version 3.14159265-2.6-1.40.21 (TeX Live 2020) kpathsea version 6.3.2
This is pgf in version 3.1.5b
This is preamble-slides.tex myFormat©C.H.Cap

# List of Slides

## 5. Attribute Based Authentication

**Legend:**

⧉ continuation slide
○ slide without title header
🖼 image slide