TECHNISCHE UNIVERSITÄT DRESDEN

DRESDEN concept

Betriebssysteme und ***Sicherheit***

Stefan Köpsell, Thorsten Strufe

Modul 3: Mechanismen – Authentisierung und Schlüssel

Disclaimer: large parts from Mark Manulis, Dan Boneh, Stefan Katzenbeisser, Günter Schäfer

Dresden, WS 15/16

Entity Authentication

„Who's Brian of Nazareth? We have an order for him to be released."

Terry Jones, et al.: Life of Brian (1979)

Identifiziere Menschen/Entity Authentication

Ziel: Identifikation eines Subjekts (User oder Process!) und Identitätsverifikation

Klassen der Authentisierung:

- *User authentication (login)*
- *Computer network authentication (identity management)*
- *Identity verification service*

Kardinalität bei der Autenthisierung:

- One-way authentication
  - Computer authentisieren User
  - ATM authentisieren Karteninhaber
  - Browser authentisiert Webserver
- Zwei-Wege-authentisierung (mutual authentication)
  - ePass <--> reader
  - UMTS cellphone < -- > network
  - Online bank < -- > account holder (w/ certificates)

Faktoren zur Authentisierung

Nutzung unterschiedlicher Faktoren:

- Wissen (knowledge factors)
  - Passworte
  - Antworten zu „security questions“
  - …
- Besitz (possession factors)
  - Security token
  - Smart card
  - Schlüssel/Zertifikate
  - …
- Sein (Inherence factors)
  - Biometrische Artefakte
  - Unterschriften
  - …
- *Manchmal andere Eigenschaften (z.B. Standort)*

Authenticating a device
Conventional: knowlede (key)
Advanced: Possession (smart card)
Recent: Inherence (PUF)
Sometimes: location (ATM)

Entity Authentication

Verifikation der Faktoren:

- ***direkt*** (Alice vs. Bob) oder
- vermittelt durch einen ***Arbiter*** („TTP“, *Kerberos, Shibboleth*)

Grundsätzliche Anforderungen:

- Stärke des Geheimnisses basiert auf seiner Entropie (Passworte, Biometrie)
- Bereitstellung und Verwaltung: Faktoren müssen geheim bleiben (*impersonation*), anpassbar und widerrufbar sein
- Monitoring, Entdeckung und Reaktion auf Mißbrauchsversuche

Mehr-Faktor Authentisierung:

- Kombination unterschiedlicher Faktoren (*Beispiele?*)
  - Bank-Karte (Besitz) und PIN (Wissen)
  - Passwort (Wissen) und mobileTAN (Besitz der SIM)
- Erfordert Unabhängigkeit der Faktoren
- Erhöht Sicherheit um die Sicherheit des schwächsten Faktors (Sicherheitsfragen?)
- *(aber: nicht mit fall-back authentication zu verwechseln – die ist so sicher wie der schwächste Faktor…)*

Schwierigkeiten bei der Authentisierung

Verifikation von Faktoren über Netze ist schwer, mit Krypto aber möglich

Entity authentication ist mehr als der reine Austausch von Nachrichten:

- Bob weiß bei Erhalt der Authentisierungsnachricht von Alice nicht, ob:
  - Alice *in diesem Moment an der Kommunikation teilnimmt,* oder ob
  - Eve eine alte Nachricht von Alice wiedereinspielt (*replay)*
- Dies ist besonders wichtig wenn nur bei Session-Setup authentisiert wird:
  - Beispiel: Übertragung eines PIN (evtl. verschlüsselt) beim login
- Zwei grundsätzliche Hilfsmittel zur Ermittlung der Aktualität:
  - *Zeitstempel* (erfordern lose synchronisierte Uhren)
  - *Zufallszahlen/Nonces* (challenge-response Austausch)

Case Studies zur Authentisierung

Authentisierung „in RL“

- Ausweise

User authentication

- Wissensbasiert: Passworte
- Inherence-based: Biometrie

Authentisierung von/durch Individue

Wie authensieren Sie „in RL“ (@home, Bank,…)

“What one *is* (inherence)
- Handgeometrie
- Fingerabdruck
- Bild
- Unterschrift
- Retina
- Stimme
- Schreib-Charakteristika

has (possession)
- Papierdokument
- Schlüssel
- Magnetstreifen-Karte
- smart card (chip card)
- SIM

knows (knowledge)”
- Passworte, Passphrasen
- Antworten zu Fragen
- Erfolgreich übermittelte Nachricht

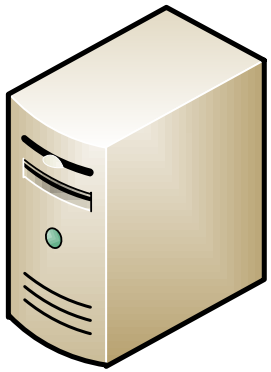
→ **eID-card**

Knowledge-based authentication: passwords

Simple approach

User

(dog, bone)



Server

yes

grant access

?

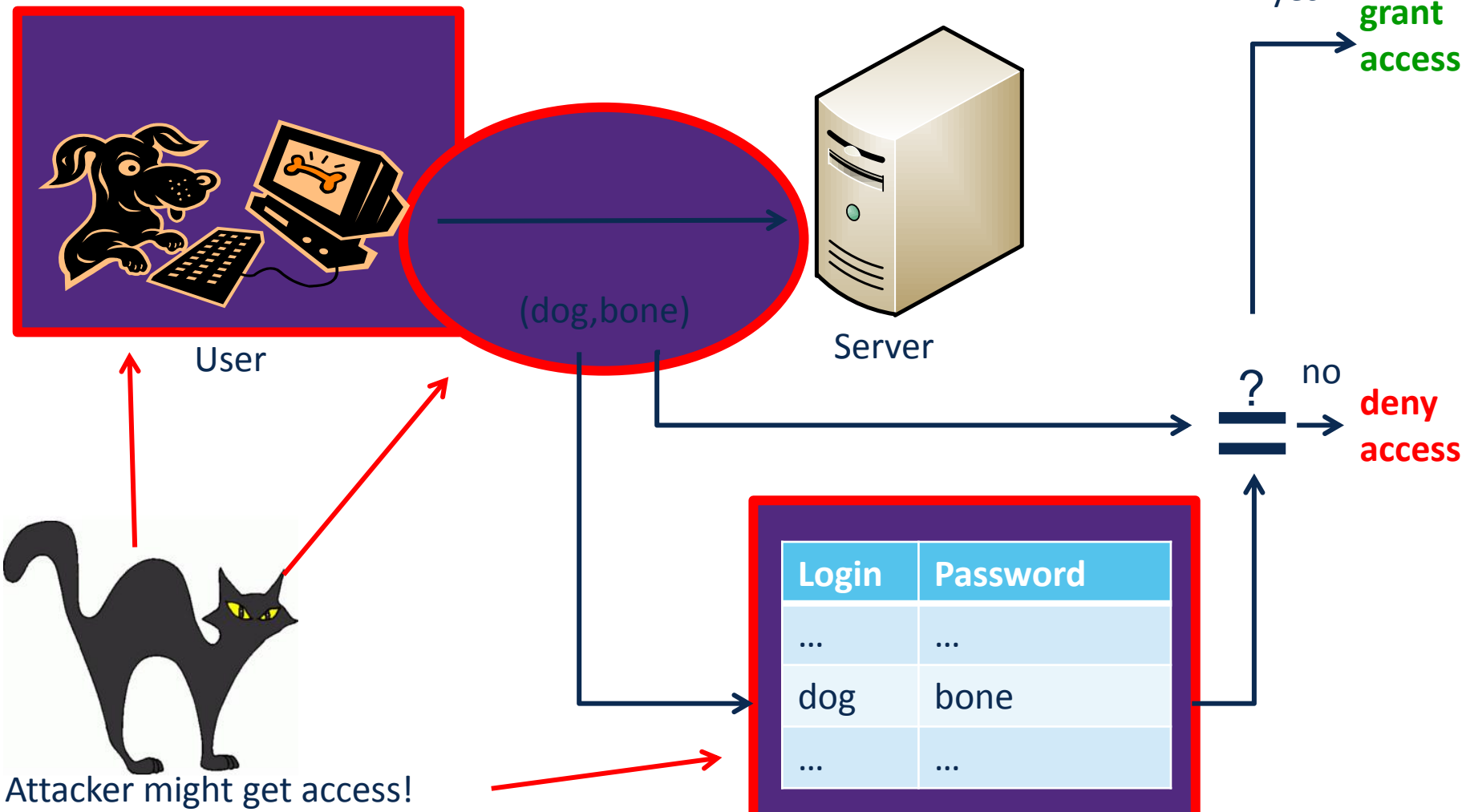
no

deny access

Login	Password
…	…
dog	bone
…	…

Password based authentication

Simple approach – **security problems**

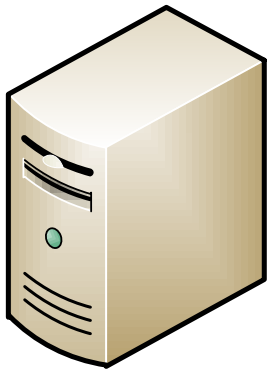


Password based authentication

Enhanced approach using one way (hash) functions

User

(dog,bone)



Server

calculates *h*(bone)

?

yes

grant access

no

deny access

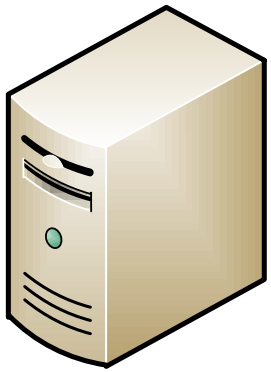
Login	Password
…	…
dog	***h*(bone)**
…	…

Password based authentication

Enhanced approach using one way (hash) functions

User

(dog,bone)



Server

calculates *h*(bone)

?

yes

grant access

no

deny access

Login	Password
…	…
dog	*h*(bone)
…	…

Slightly reduced risk, if attacker gets access.

Problems of hashed password storage

Possible attack:

- pre-computation (rainbow tables)

Login	Password
…	…
dog	*h*(bone)
…	…

countermeasures:

- Hash rounds
  - store: h^{1000}(pw)
  - linear overhead per string (computation, storage)
- Salting
  - Use (long) random value
  - store: h(salt,pw), salt

```
$ sudo less /etc/shadow | grep strufe
strufe:$6$m40rV3LS$kAf.4WUEwr7[…]
```

- ➔ pre-computation has to be done for each possible salt,|rounds|

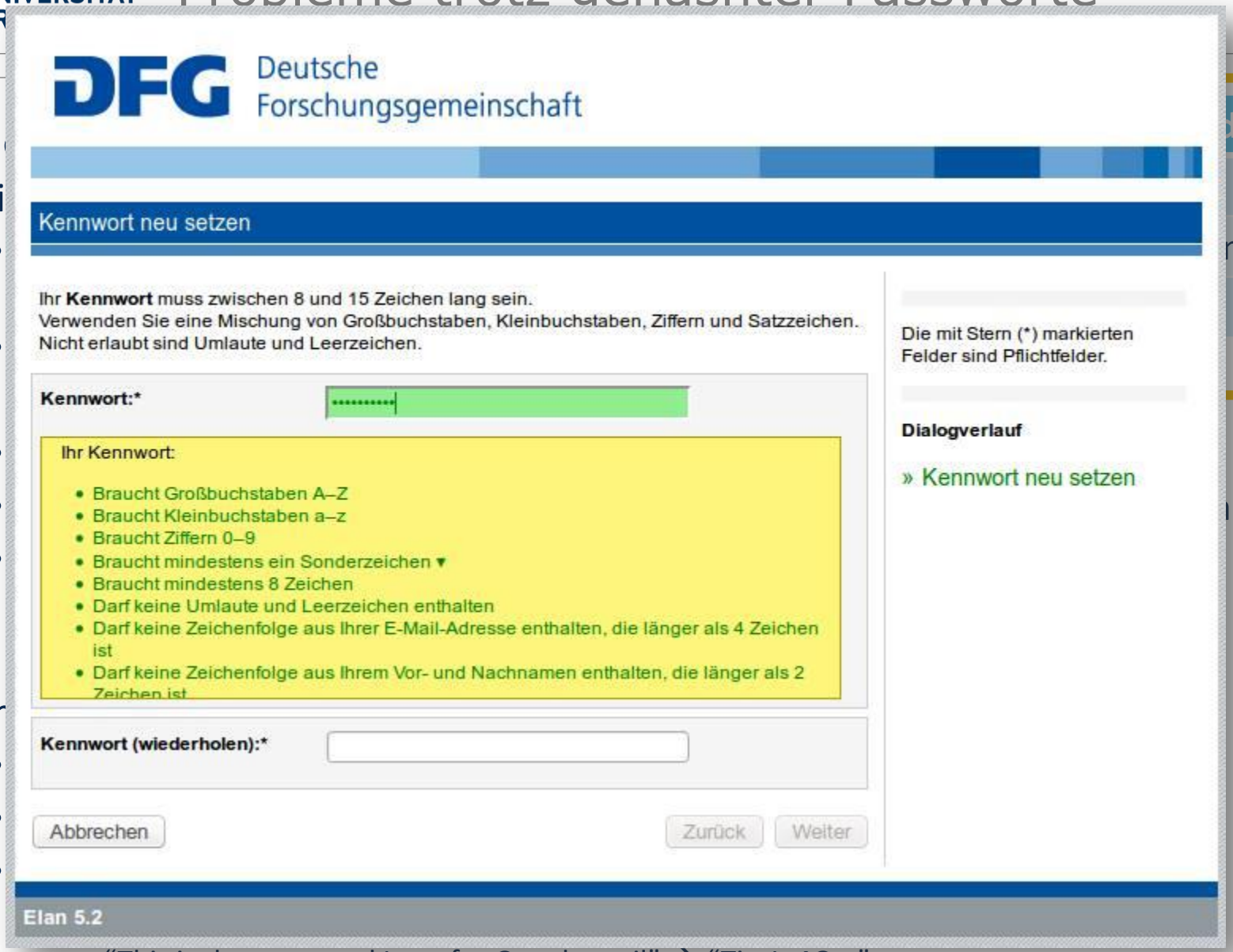
Probleme trotz gehashter Passworte

Verbl

- Di
  -
  -
  -
  -
  -

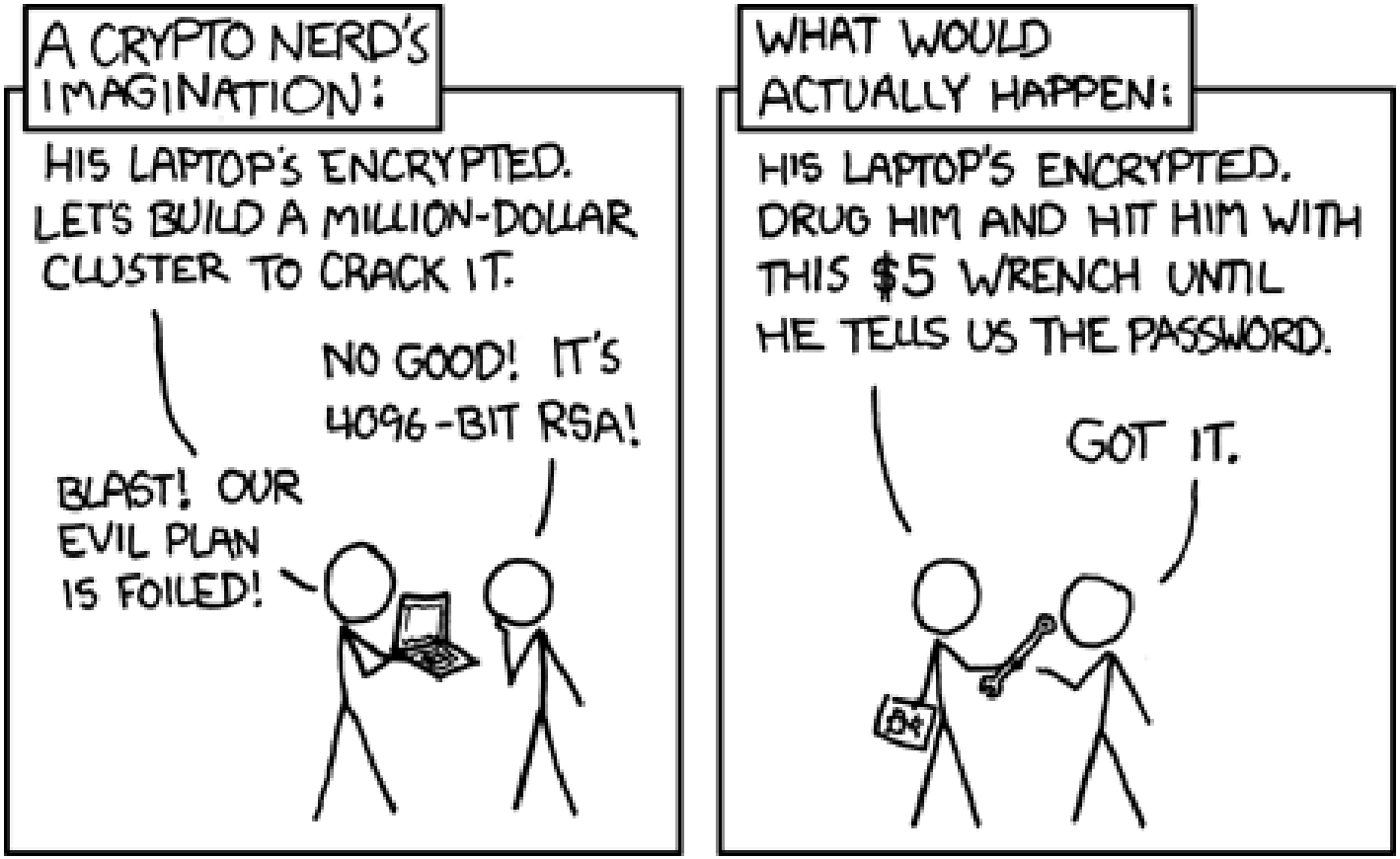
Lösur

  -
  -
  -



  - “This is the password I use for Google mail” → “Titplu4Gm”

Verbleibende Probleme…

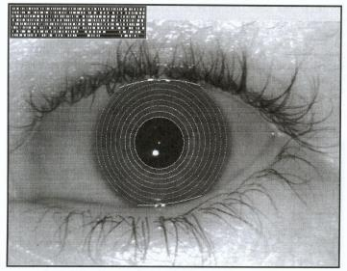


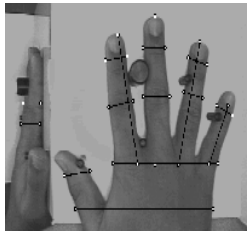
Inherence factors: Biometry

Identifying an individual by physiological characteristics

Requirements:

- **Universality**: Applies to all individuals
- **Uniqueness**: Different for all individuals
- **Stable**: Property does not change over lifetime
- **Measurable** reliably, easily with (cheap) sensors
- **Acceptance** by users
- Resistant to **forgery**





Inherence factors: Biometry

Two-stage process:

- Enrollment: Registration of users
- Verification: Capture and compare biometric data with samples stored during enrollment

Verification is always noisy (statistical hypothesis tests)

- False positives: unauthorized individual is authenticated (FAR)
  → Security at risk!
- False negatives: authorized individual is rejected (FRR)
  → Problems for acceptance and usability
- FAR & FRR can be adapted by choice of features/characteristics

Advantages:

(+) Cannot be lost/forgotten

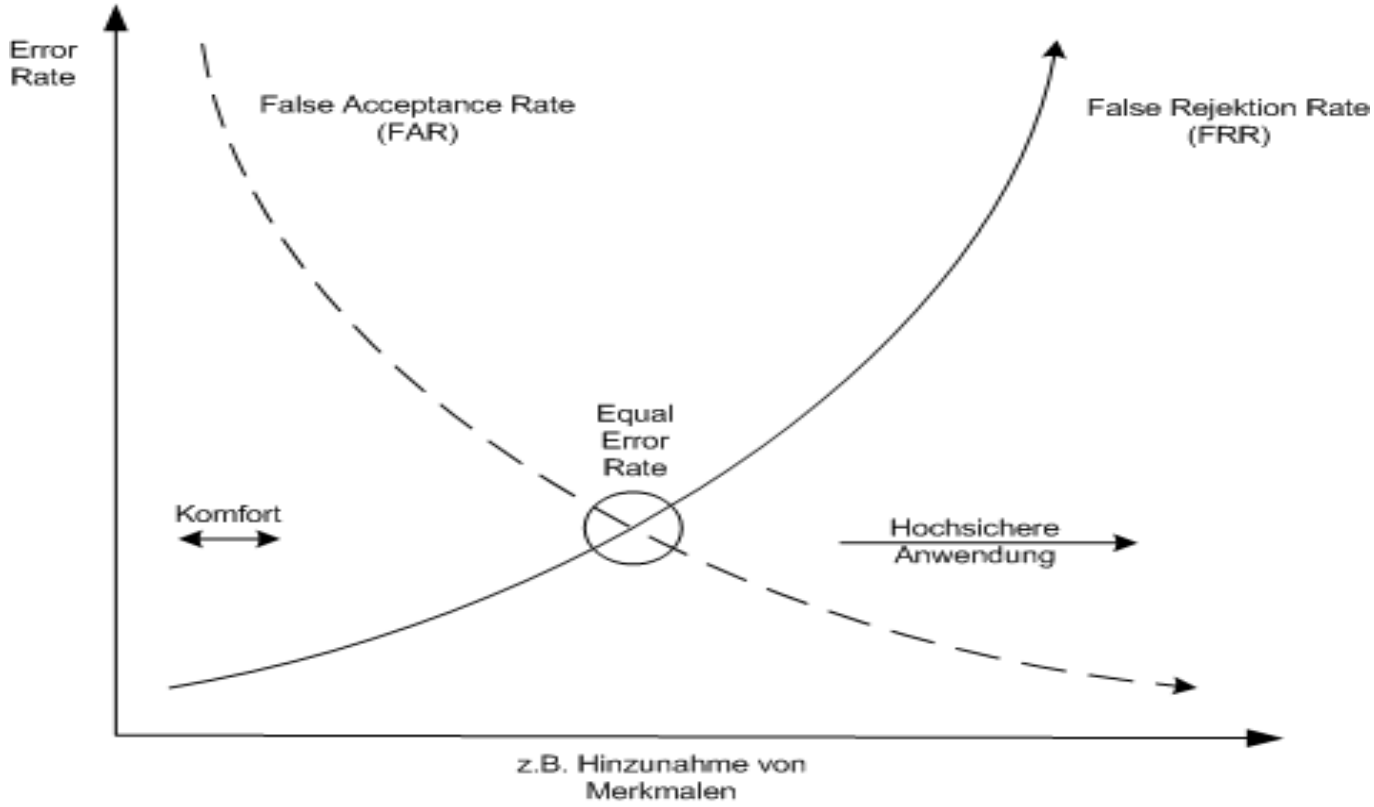
(+) Hard to copy

Disadvantages:

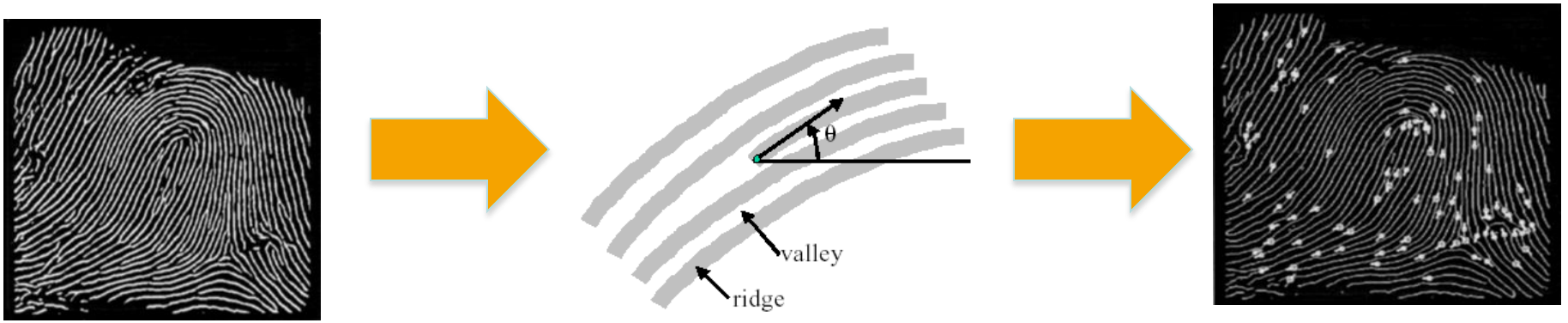
(-) Very hard to change/reset

(-) Error rate

Inherence factors: Biometry



Inherence factors: Biometry



- Storage of **minutiae**: Branches, endpoints, etc.
- Pattern (coordinates and angle) of minutiae considered to be unique
- Compare measured minutiae with samples stored at enrollment
- Problem: Some minutiae may be missing (noisy, incomplete sensing)!

- Main biometric factor (Face recognition, retina scans increasingly used)

Schlüssel-Verteilung

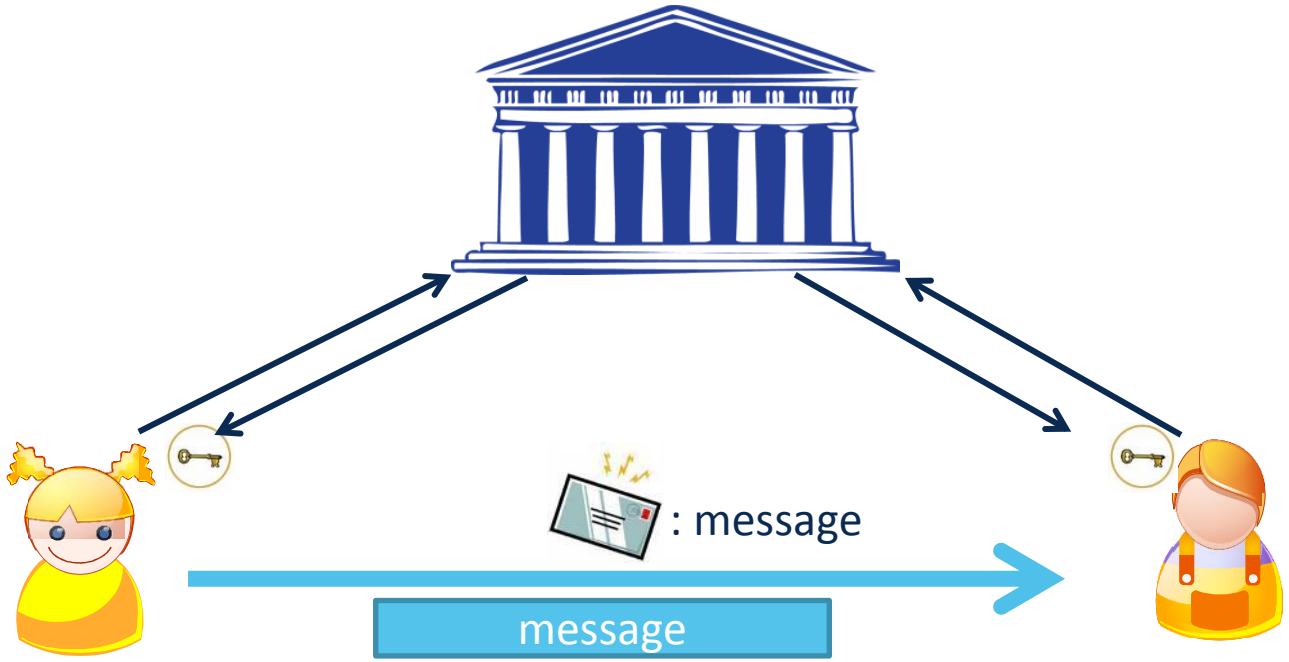
Einrichtung eines sicheren Kanals…

„Schlüsselaustausch/-verteilung“:

- Bedarf ***sicheren,*** direkten Kanals!

Wieviele Schlüssel-Austausch-Transaktionen gibt es bei N Teilnehmern?

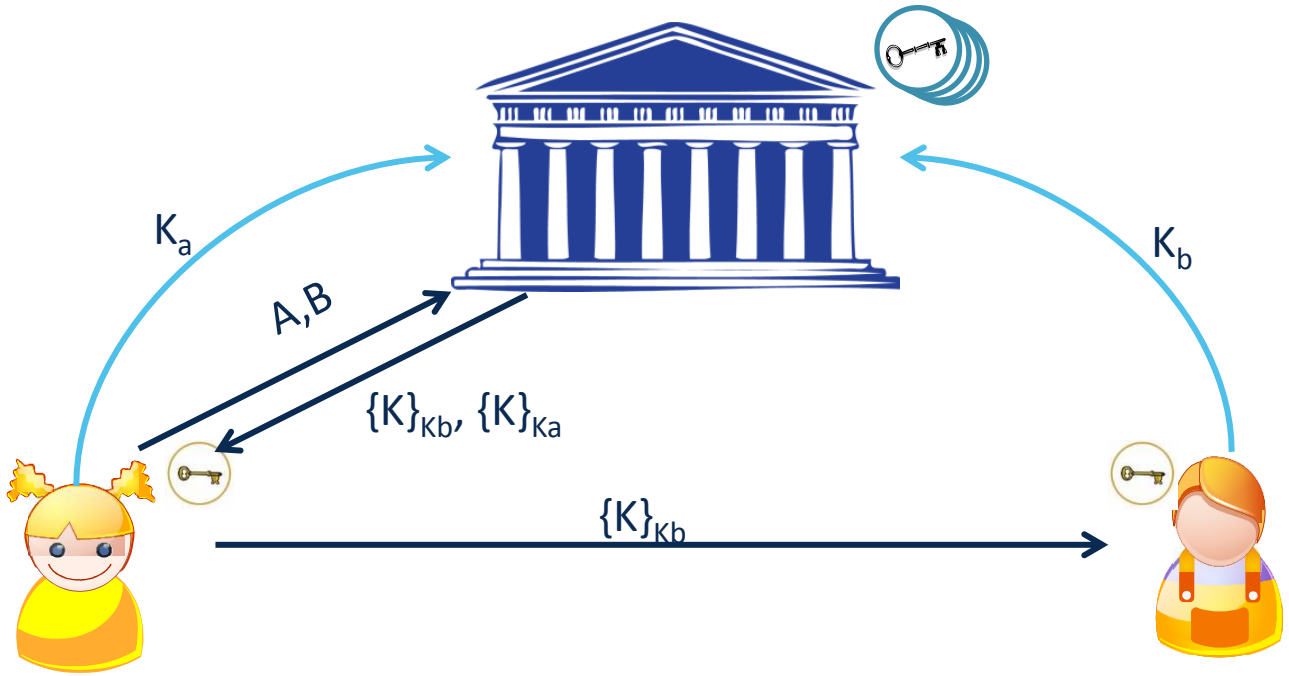
Trusted 3rd Parties/Key Distribution Centers



Key Distribution:

- Paarweise Schlüssel mit TTP
- Aber was ist auszutauschen?

KDC: a Very Simple Protocol



Simple Key Exchange:

- TTP kennt / generiert alle Schlüssel
- Eve kann Verschlüsselung nicht brechen, aber
- Mallory kann aktiv eingreifen *(wie?)*

Dolev/Yao Angreifer

Volle Kontrolle über den Übertragungskanal:

- Nachrichten abfangen / mithören (passiv)
- Nachrichten verzögern
- Nachrichtenübermittlung unterdrücken
- Nachrichten erneut senden
- Nachrichten verändern
- Nachrichten durch andere Nachrichten ersetzen
- Vollkommen neue Nachrichten fälschen

Aber:

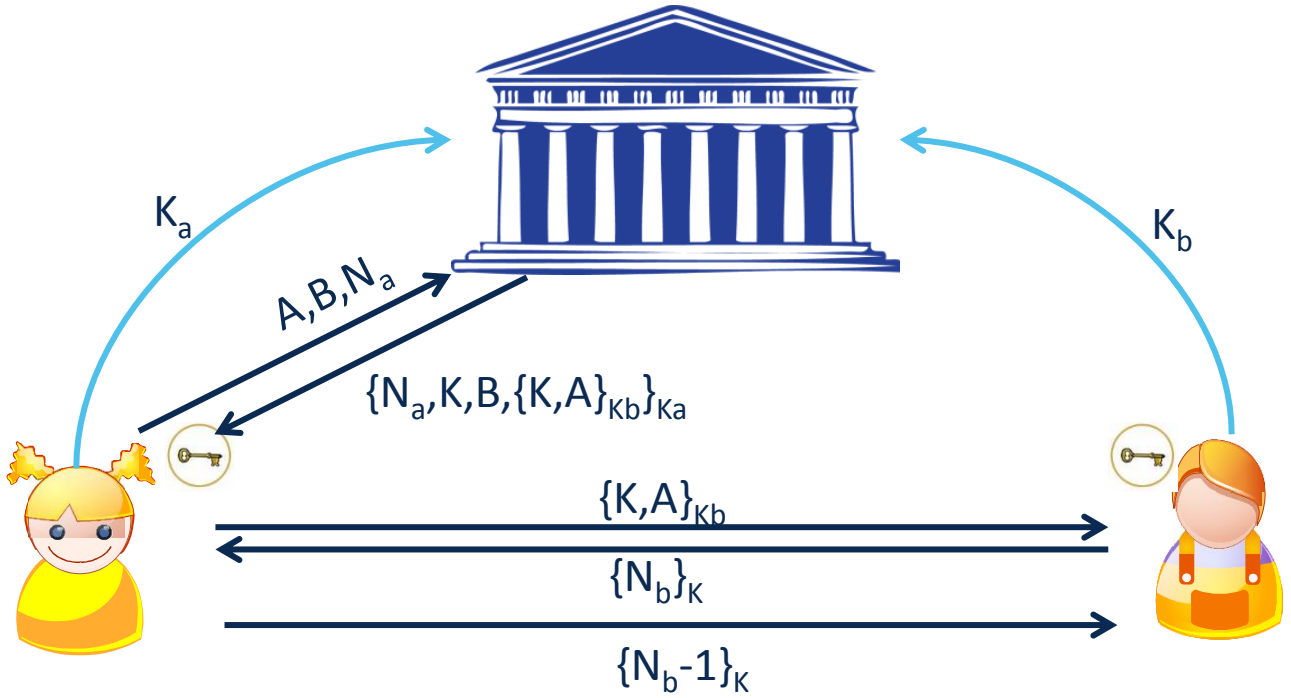
- Mallory kann kryptographische Primitive ***nicht*** brechen!

Angriffe auf Schlüsseletablierung

- Man-in-the-middle attack

- Replay attack

Schroeder-Needham Schlüsseltausch



- Impersonation/MitM durch verschlüsselte Adressierung verhindert (authentisiert Alice und Bob)
- Replay durch Nonce verhindert
- Falls alter Schlüssel gebrochen wurde kann Malory sich als Alice ausgeben, Angriff kann durch Zeitstempel verhindert werden

Key *Agreement*

Ralph Merkle, Martin Hellmann, Whitfield Diffie

Berechnung des Schlüssel (statt Tausch)

Ziel:

Auszutauschende Information soll vollkommen öffentlich sein

Primäre Idee (Merkle, ’74):

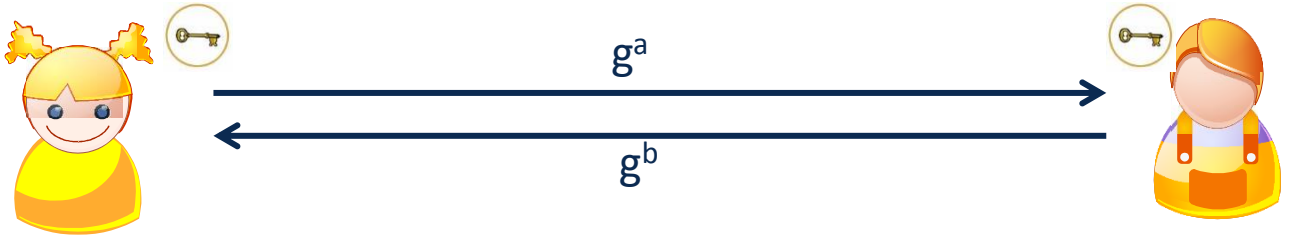
- Alice generiert 2^{32} Puzzle („brechen“: P_i und Schlüssel) (O(n))
- Bob wählt zufällig P_j aus 2^{32} und „berechnet“ Schlüssel (O(n))
- Bob schickt P_j an Alice, Schlüssel etabliert.

Welchen Aufwand hat Mallory?

Can‘t we do better? Diffie-Hellmann

Zyklische Gruppe G mit Generator g und φ(G)=p (prim)

Alice wählt a $\xleftarrow{R}$ {1,…,(p-1)}, Bob wählt b $\xleftarrow{R}$ {1,…,(p-1)}



Alice berechnet $(g^a)^b$ = g^{ab} = Bob berechnet $(g^b)^a$

Computational Diffie Hellmann Problem (CDH):

- *Given p,g, g^a, g^b*
- *Output g^{ab}*

Was ist mit MitM?