Privacy-Enhancing Technologies:
DAA, Anonymous Credentials & Pseudonym Systems

Anja Lehmann
IBM Research – Zurich

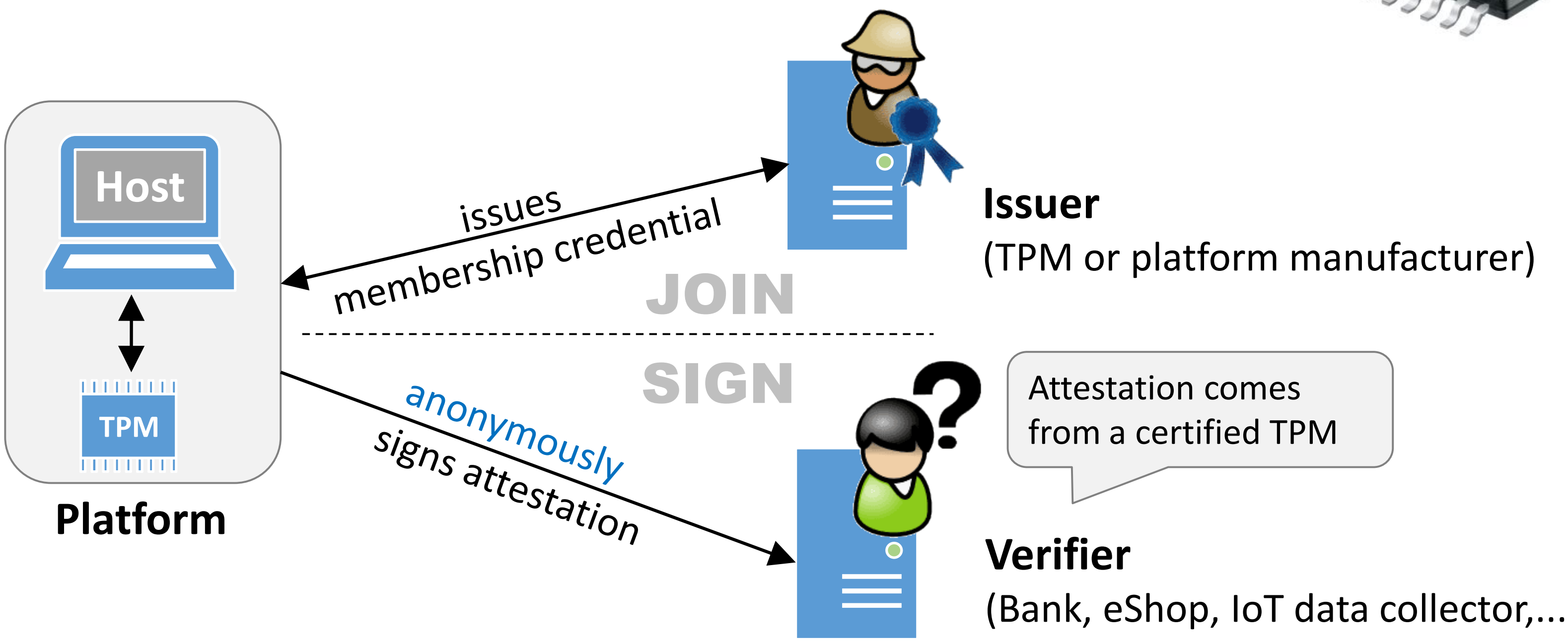
- Direct Anonymous Attestation
  – hardware-based attestation

- Anonymous Credentials
  – privacy-preserving (user) authentication

- Pseudonym Systems
  – privacy-preserving data exchange

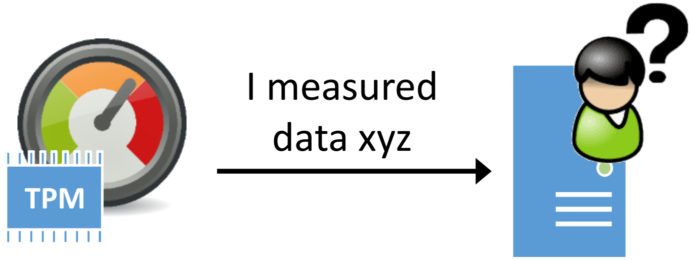
Direct Anonymous Attestation (DAA)

- Hardware-based attestation using a Trusted Platform Module (TPM)
  – Secure crypto processor: creates, stores, uses cryptographic keys
  – Makes remote attestations of host status

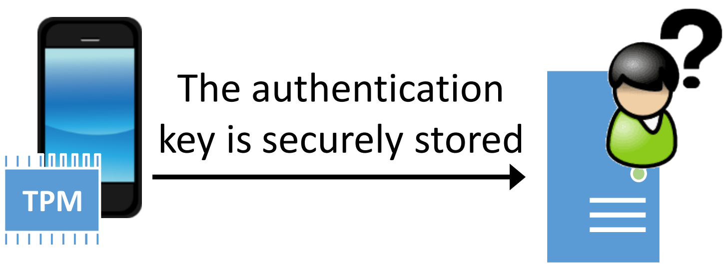


- Standard certificates would make all attestations linkable and reveal TPM‘s ID
- Direct Anonymous Attestation = strong, but privacy-preserving authentication

- First DAA protocol by Brickell, Camenisch, Chen [BCC04]
  - Developed for Trusted Computing Group (TCG) = industry group that standardizes TPM
  - RSA-based
  - Standardized in TPM1.2 (2004) & ISO/IEC 20008-2
- Revised TPM2.0 (2014)
  - Elliptic curve & pairing based
  - Flexible API to support different protocols
  - TPM part & protocols ISO standardized
    - ISO/IEC 20008-2
    - ISO/IEC 11889



- Over 500 million TPMs sold
- Today: Interest in TPM revived
  - Security of mobile and IoT devices
  - SGX & EPID
  - FIDO authentication



DAA | Current Status

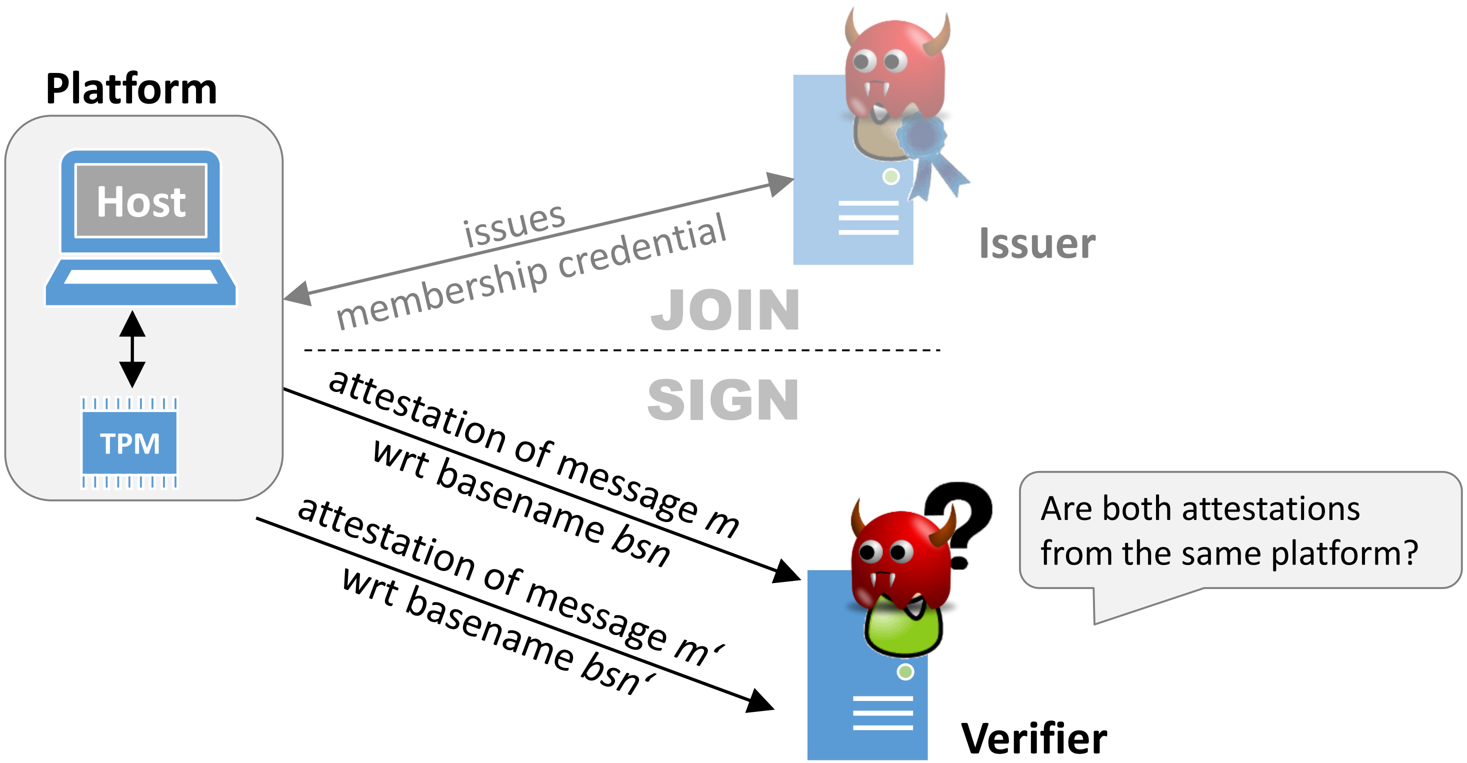
- Ideally: deployed DAA scheme is provably secure

1. Security Model
2. Provably Secure Cryptographic Protocol (secure according to 1.)
3. Secure Implementation (of 2.)

...lets see where we are now, 13 years after DAA was invented

DAA | Security Properties

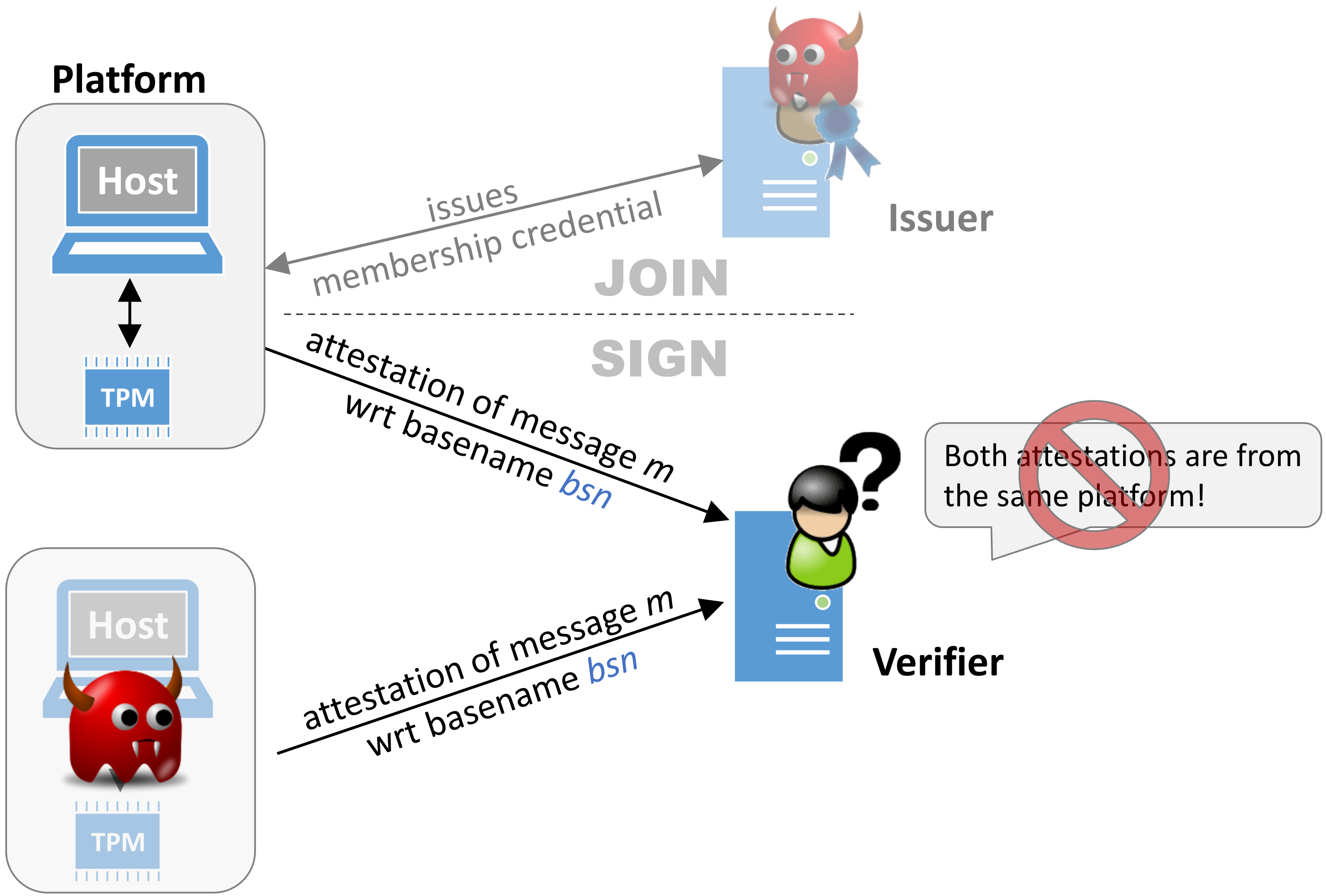
- Anonymity/Unlinkability
  - Attestation doesn‘t leak any information about the platform‘s identity
  - Unlinkability steered via basename



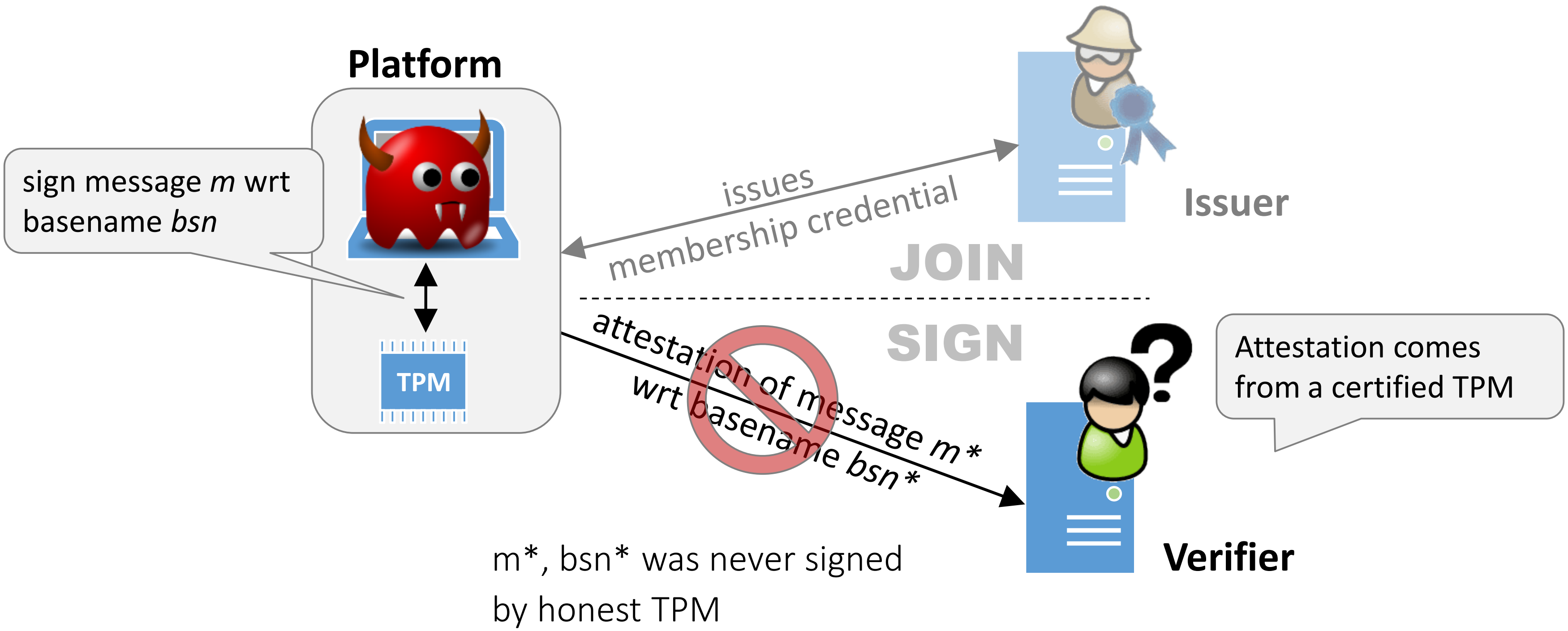
Unlinkable if bsn ≠ bsn‘

DAA | Security Properties

- Non-Frameability
  - Adversary should not be able to impersonate honest platforms



- Unforgeability
  - Adversary should not be able to create attestations without TPM



Simulation-based Definitions

- Brickell, Camenisch, Chen [BCC04]
  - Does not output signatures
  - Prohibits working with signatures in practice
- Chen, Morissey, Smart [CMS08]
  - Output signatures = random values
  - Not realizable by *any* construction
- Camenisch, Drijvers, Lehmann [CDL16a]
  - Security model in UC Framework
  - TPM and host separate parties
  - Signatures modeled as concrete values – for random TPM keys
- Camenisch, Drijvers, Lehmann [CDL17]
  - First definition for privacy against *subverted* TPMs

Game-based Definitions

- Brickell, Chen, Li [BCL09]
  - Trivially forgeable scheme can be proven secure
  - No property for non-frameability
- Chen [C10], Chen, Li [BL10]
  - Extend [BCL09] with non-frameability and signature-based revocation, EPID
  - Same unforgeability flaw as [BCL09]
- Bernhard et al. [BFG+13]
  - Discuss flaws in all previous models
  - Extensive set of definitions for all expected properties
  - But for "pre-DAA", where TPM + host are one party → does not cover honest TPM in corrupt host

All existing security definitions had issues, some of them severe, allowing for insecure schemes!

DAA | Security Models – Open Challenges

1. `Issuer Setup`. On input (SETUP, sid) from issuer $\mathcal{I}$.
   – Verify that $sid = (\mathcal{I}, sid')$.
   – Output (SETUP, sid) to $\mathcal{A}$ and wait for input $(\mathsf{ALG}, sid, \mathsf{sig}, \mathsf{ver}, \mathsf{link}, \mathsf{identify}, \mathsf{ukgen})$ from $\mathcal{A}$.
   – Check that ver, link and $\mathsf{identify}$ are deterministic.
   – Store $(sid, \mathsf{sig}, \mathsf{ver}, \mathsf{link}, \mathsf{identify}, \mathsf{ukgen})$ and output $(\mathsf{SETUPDONE}, sid)$ to $\mathcal{I}$.

Join

2. `Join Request`. On input $(\mathsf{JOIN}, sid, jsid, \mathcal{M}_i)$ from host $\mathcal{H}_j$.
   – Create a join session record $\langle jsid, \mathcal{M}_i, \mathcal{H}_j, status\rangle$ with $status \leftarrow request$.
   – Output $(\mathsf{JOIN}, sid, jsid, \mathcal{H}_j)$ to $\mathcal{M}_i$.
3. $\mathcal{M}$ `Join Proceed`. On input $(\mathsf{JOIN}, sid, jsid)$ from TPM $\mathcal{M}_i$.
   – Update the session record $\langle jsid, \mathcal{M}_i, \mathcal{H}_j, status\rangle$ with $status = request$ to $delivered$.
   – Output $(\mathsf{JOINPROCEED}, sid, jsid, \mathcal{M}_i, \mathcal{H}_j)$ to $\mathcal{A}$, wait for input $(\mathsf{JOINPROCEED}, sid, jsid)$ from $\mathcal{A}$.
   – Abort if $\mathcal{I}$ or $\mathcal{M}_i$ is honest and a record $\langle \mathcal{M}_i, *, *\rangle \in$ `Members` already exists.
   – Output $(\mathsf{JOINPROCEED}, sid, jsid, \mathcal{M}_i)$ to $\mathcal{I}$.
4. $\mathcal{I}$ `Join Proceed`. On input $(\mathsf{JOINPROCEED}, sid, jsid)$ from $\mathcal{I}$.
   – Update the session record $\langle jsid, \mathcal{M}_i, \mathcal{H}_j, status\rangle$ with $status = delivered$ to $complete$.
   – Output $(\mathsf{JOINCOMPLETE}, sid, jsid)$ to $\mathcal{A}$ and wait for input $(\mathsf{JOINCOMPLETE}, sid, jsid, \tau)$ from $\mathcal{A}$
   – If $\mathcal{H}_j$ is honest, set $\tau \leftarrow \perp$. *(strong non-frameability)*
   – Else, verify that the provided tracing trapdoor τ is eligible by checking $\mathsf{CheckTtdCorrupt}(\tau) = 1$.
   – Insert $\langle \mathcal{M}_i, \mathcal{H}_j, \tau\rangle$ into `Members` and output $(\mathsf{JOINED}, sid, jsid)$ to $\mathcal{H}_j$.

Sign

5. `Sign Request`. On input $(\mathsf{SIGN}, sid, ssid, \mathcal{M}_i, m, bsn)$ from $\mathcal{H}_j$.
   – If $\mathcal{H}_j$ is honest and no entry $\langle \mathcal{M}_i, \mathcal{H}_j, *\rangle$ exists in `Members`, abort.
   – Create a sign session record $\langle ssid, \mathcal{M}_i, \mathcal{H}_j, m, bsn, status\rangle$ with $status \leftarrow request$.
   – Output $(\mathsf{SIGNPROCEED}, sid, ssid, m, bsn)$ to $\mathcal{M}_i$.
6. `Sign Proceed`. On input $(\mathsf{SIGNPROCEED}, sid, ssid)$ from $\mathcal{M}_i$.
   – Look up record $\langle ssid, \mathcal{M}_i, \mathcal{H}_j, m, bsn, status\rangle$ with $status = request$ and update it to $status \leftarrow complete$.
   – If $\mathcal{I}$ is honest, check that $\langle \mathcal{M}_i, \mathcal{H}_j, *\rangle$ exists in `Members`.
   – Generate the signature for a fresh or established key: *(strong privacy)*
     • Retrieve (gsk, τ) from $\langle \mathcal{M}_i, \mathcal{H}_j, bsn, gsk, \tau\rangle \in$ `DomainKeys`. If no such entry exists, set $(gsk, \tau) \leftarrow \mathsf{ukgen}()$, check $\mathsf{CheckTtdHonest}(\tau) = 1$, and store $\langle \mathcal{M}_i, \mathcal{H}_j, bsn, gsk, \tau\rangle$ in `DomainKeys`.
     • Compute signature $\sigma \leftarrow \mathsf{sig}(gsk, m, bsn)$, check $\mathsf{ver}(\sigma, m, bsn) = 1$.
     • Check $\mathsf{identify}(\sigma, m, bsn, \tau) = 1$ and that there is no $(\mathcal{M}', \mathcal{H}') \neq (\mathcal{M}_i, \mathcal{H}_j)$ with tracing trapdoor τ' registered in `Members` or `DomainKeys` with $\mathsf{identify}(\sigma, m, bsn, \tau') = 1$.
   – Store $\langle \sigma, m, bsn, \mathcal{M}_i, \mathcal{H}_j\rangle$ in `Signed` and output $(\mathsf{SIGNATURE}, sid, ssid, \sigma)$ to $\mathcal{H}_j$.

Verify & Link

7. `Verify`. On input $(\mathsf{VERIFY}, sid, m, bsn, \sigma, \mathtt{RL})$ from some party $\mathcal{V}$.
   – Retrieve all tuples $(\tau_i, \mathcal{M}_i, \mathcal{H}_j)$ from $\langle \mathcal{M}_i, \mathcal{H}_j, \tau_i\rangle \in$ `Members` and $\langle \mathcal{M}_i, \mathcal{H}_j, *, *, \tau_i\rangle \in$ `DomainKeys` where $\mathsf{identify}(\sigma, m, bsn, \tau_i) = 1$. Set $f \leftarrow 0$ if at least one of the following conditions hold:
     • More than one τ_i was found.
     • $\mathcal{I}$ is honest and no pair $(\tau_i, \mathcal{M}_i, \mathcal{H}_j)$ was found.
     • $\mathcal{M}_i$ or $\mathcal{H}_j$ is honest but no entry $\langle *, m, bsn, \mathcal{M}_i, \mathcal{H}_j\rangle \in$ `Signed` exists. *(strong unforgeability)*
     • There is a $\tau' \in \mathtt{RL}$ where $\mathsf{identify}(\sigma, m, bsn, \tau') = 1$ and no pair $(\tau_i, \mathcal{M}_i, \mathcal{H}_j)$ for an honest $\mathcal{H}_j$ was found.
   – If $f \neq 0$, set $f \leftarrow \mathsf{ver}(\sigma, m, bsn)$.
   – Add $\langle \sigma, m, bsn, \mathtt{RL}, f\rangle$ to `VerResults` and output $(\mathsf{VERIFIED}, sid, f)$ to $\mathcal{V}$.
8. `Link`. On input $(\mathsf{LINK}, sid, \sigma, m, \sigma', m', bsn)$ from a party $\mathcal{V}$.
   – Output $\perp$ to $\mathcal{V}$ if at least one signature (σ, m, bsn) or (σ', m', bsn) is not valid (verified via the `verify` interface with $\mathtt{RL} = \emptyset$).
   – For each τ_i in `Members` and `DomainKeys` compute $b_i \leftarrow \mathsf{identify}(\sigma, m, bsn, \tau_i)$ and $b'_i \leftarrow \mathsf{identify}(\sigma', m', bsn, \tau_i)$ and do the following:
     • Set $f \leftarrow 0$ if $b_i \neq b'_i$ for some i.
     • Set $f \leftarrow 1$ if $b_i = b'_i = 1$ for some i.
   – If f is not defined yet, set $f \leftarrow \mathsf{link}(\sigma, m, \sigma', m', bsn)$.

`functionality_simple.tex`

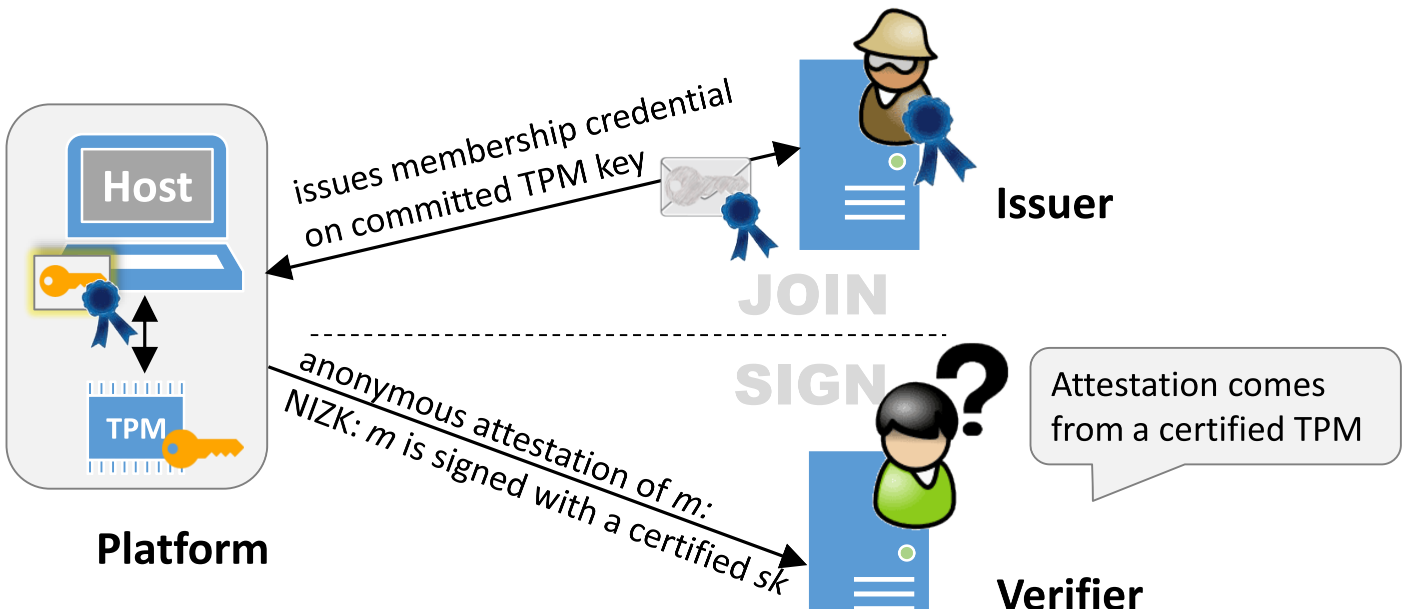
Simpler & more modular security models and proofs?

DAA | Current Status

- What is needed to make DAA a provably secure real-world protocol?

1. Security Model ✓
2. Provably Secure Cryptographic Protocol (secure according to 1.)
3. Secure Implementation (of 2.)

DAA | Protocols – Common Approach

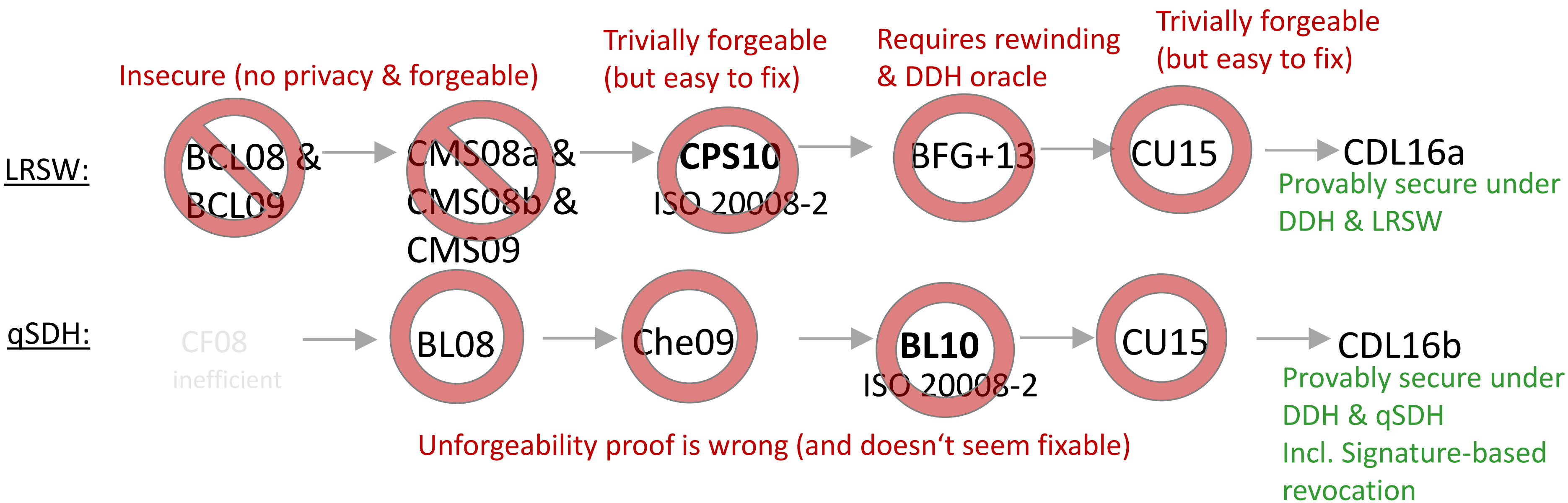


TPM creates & stores secret key
Host stores membership credential

- DAA protocols mainly differ on how the membership credential & NIZK is computed
- First protocol [BCC04] based on RSA, standardized in TPM1.2 (very slow)
- Subsequent DAA protocols & TPM2.0 based on elliptic curves and pairings

DAA | Protocols for TPM2.0

- TPM2.0 offers generic APIs to support various schemes,
  e.g., DAA based on LRSW (CL-signature) & qSDH (BBS+ signature)



- All existing schemes are either insecure, or cannot be proven secure in strongest model
  - (1, 1, 1, 1) is a valid credential on *any* key in [CPS10] – ISO 20008 standardized

- Revised provably secure protocols [CDL16a, CDL16b]
  - as efficient as existing schemes – mainly details had to be fixed

DAA | Current Status

1. Security Model ✓
2. Provably Secure Cryptographic Protocol (secure according to 1.) ✓
3. Secure Implementation (of 2.)

Efficient protocol, lightweight part for TPM .... Done!

TPM accessible only via few, limited APIs

DAA | TPM2.0 Interfaces (High-Level Idea)

TPM

`TPM.Create()`

draw $sk \in Z_q$, store sk

output $pk \leftarrow g^{sk}$

`TPM.Hash(t,m)`

output $c \leftarrow H(t,m)$

`TPM.Commit(P)`

choose $r \in Z_q$, store (ctr, r)

$t \leftarrow P^r$

output (ctr, t)

`TPM.Sign(c,ctr)`

get (ctr, r)

output $s \leftarrow r + c \cdot sk$

(Signature) Proof-of-Knowledge:

Prover		Verifier
sk		$pk = g^{sk}$
choose $r \in Z_q$		
$t \leftarrow g^r$	$\xrightarrow{t}$	
	$\xleftarrow{c}$	$c \leftarrow H(t,m)$
$s \leftarrow r + c \cdot sk$	$\xrightarrow{s}$	
		$g^s = t \cdot pk^c$?

- Both revised protocols are not compatible with current TPM2.0 interfaces
- Protocols designed to avoid a static Diffie-Hellman oracle – but TPM2.0 is one

DAA | TPM2.0 Interfaces – Static Diffie-Hellman Oracle

TPM

`TPM.Create()`

draw $sk \in Zq$, store sk

output $pk \leftarrow g^{sk}$

`TPM.Hash(t,m)`

output $c \leftarrow H(t,m)$

`TPM.Commit(P)`

choose $r \in Zq$, store (ctr, r)

$t \leftarrow P^r$

output (ctr, t)

`TPM.Sign(c,ctr)`

get (ctr, r)

output $s \leftarrow r + c \cdot sk$

TPM2.0 interfaces provide static Diffie-Hellman Oracle

- DH oracle via Commit, Hash & Sign query:

$$P^{sk} \leftarrow (P^s / t)^{1/c}$$

For arbitrary P chosen by (corrupt) host

- Get TPM to compute $g^{sk}, g^{sk^2}, g^{sk^3} \cdots g^{sk^n}$
- Static DH oracle significantly reduces security level,

  e.g., 256bit BN curve: 128bit security reduced to 85bit

TPM interfaces should be revised to remove the static DH oracle!

DAA | TPM2.0 Interfaces w/o Static DH Oracle

TPM

`TPM.Create()`

draw sk $\in$ Zq, store sk

output pk $\leftarrow g^{sk}$

`TPM.Hash(t,m)`

output c $\leftarrow$ H(t,m)

`TPM.Commit(bsn)`

choose r $\in$ Zq , store (ctr, r)

P $\leftarrow$ H(bsn), t $\leftarrow P^r$

output (ctr, t)

`TPM.Sign(c,ctr)`

get (ctr, r)

output s $\leftarrow$ r + c ·sk

- Simple TPM2.0 fix removes static DH oracle
  - But not „compatible“ with protocol design

    vs
  - New protocols required unrealistic APIs
- Re-revised provably-secure LRSW/qSDH-DAA
  - Required low-level redesign of CL-issuance

DAA | TPM2.0 Interfaces w/o Static DH Oracle

TPM

`TPM.Create()`

draw sk ∈ Zq, store sk

output pk ← g^{sk}

`TPM.Hash(t,m)`

output c ← H(t,m)

`TPM.Commit(bsn)`

choose r ∈ Zq , store (ctr, r)

P ← H(bsn), t ← P^r

output (ctr, t)

`TPM.Sign(c,ctr)`

get (ctr, r)

output s ← r + c ·sk

Revised TPM2.0 interfaces w/o static DH ✓

Re-revised provably secure LRSW/qSDH-DAA ✓

Are we done now?

Are the TPM-based contributions unforgeable & anonymous ?

- Chen, Li [CL13]
  - Proof that TPM2.0 generated SPKs are unforgeable
- Xi et al. [XYZF14]
  - Proof by [CL13] is wrong
  - Unforgeability cannot be proven – but simple fix
  - Proposed fix introduces subliminal channel!

DAA | TPM2.0 Interfaces w/o Static DH Oracle & Provably-Secure

TPM

TPM.Create()

draw sk ∈ Zq, store sk

output pk ← g^{sk}

TPM.Hash(t,m)

output c ← H(t,m)

TPM.Commit(bsn)

random nT, hT ← H(nT)

choose r ∈ Zq , store (ctr, r, **nT**)

P ← H(bsn), t ← P^r

output (ctr, t, **hT**)

TPM.Sign(c,ctr,nH)

get (ctr, r, **nT**)

c' ← H(nH ⊕ nT , c)

output **nT**, s ← r + **c'** ·sk

Revised TPM2.0 interfaces w/o static DH

Re-revised provably secure LRSW/qSDH-DAA

Are we done now?

The TPM-based contributions are unforgeable & anonymous

- Camenisch et al. [CCD+17]
  - Revised fix by Xi et al.
  - Abstract SPK interface from TPM commands: unforgeable, anonymous and device-bound

Full formal model & framework for trusted hardware?

Validation of hardware properties?

IBM

1. Security Model
2. Provably Secure Cryptographic Protocol (secure according to 1.)
3. Secure Implementation (of 2.)

<u>Current Status:</u>

- Revision of TPM2.0 APIs under review by TCG
- Working on revision of ISO standardized protocols
  - Corrigendum to avoid trivial forgeries in LRSW-DAA scheme
- Working with Intel on revision of EPID spec
- FIDO key attestation using DAA: spec and reference implementation published

- Direct Anonymous Attestation
  – hardware-based attestation

Privacy-Enhancing Credentials

- Anonymous Credentials
  – privacy-preserving (user) authentication

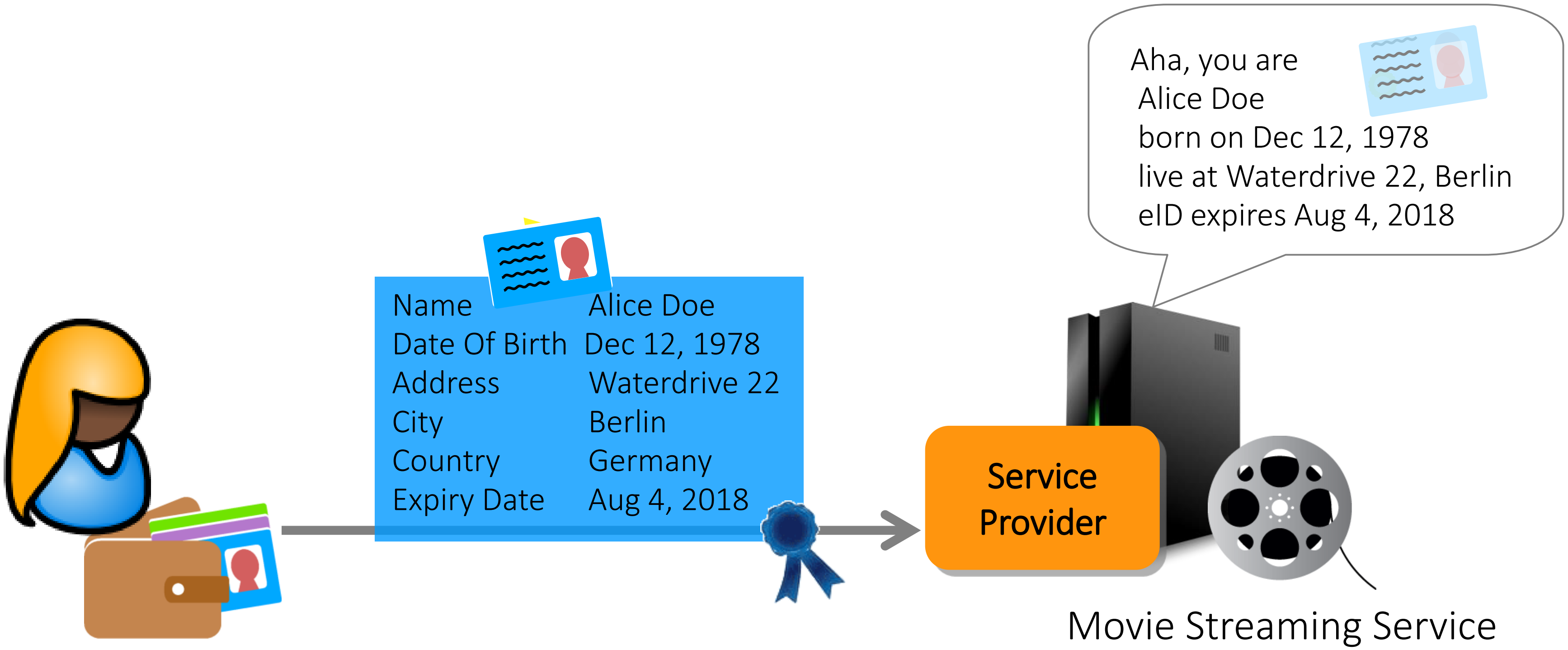
Privacy-Preserving Credentials

- Pseudonym Systems
  – privacy-preserving data exchange

Privacy-Enhancing Credentials | Motivation

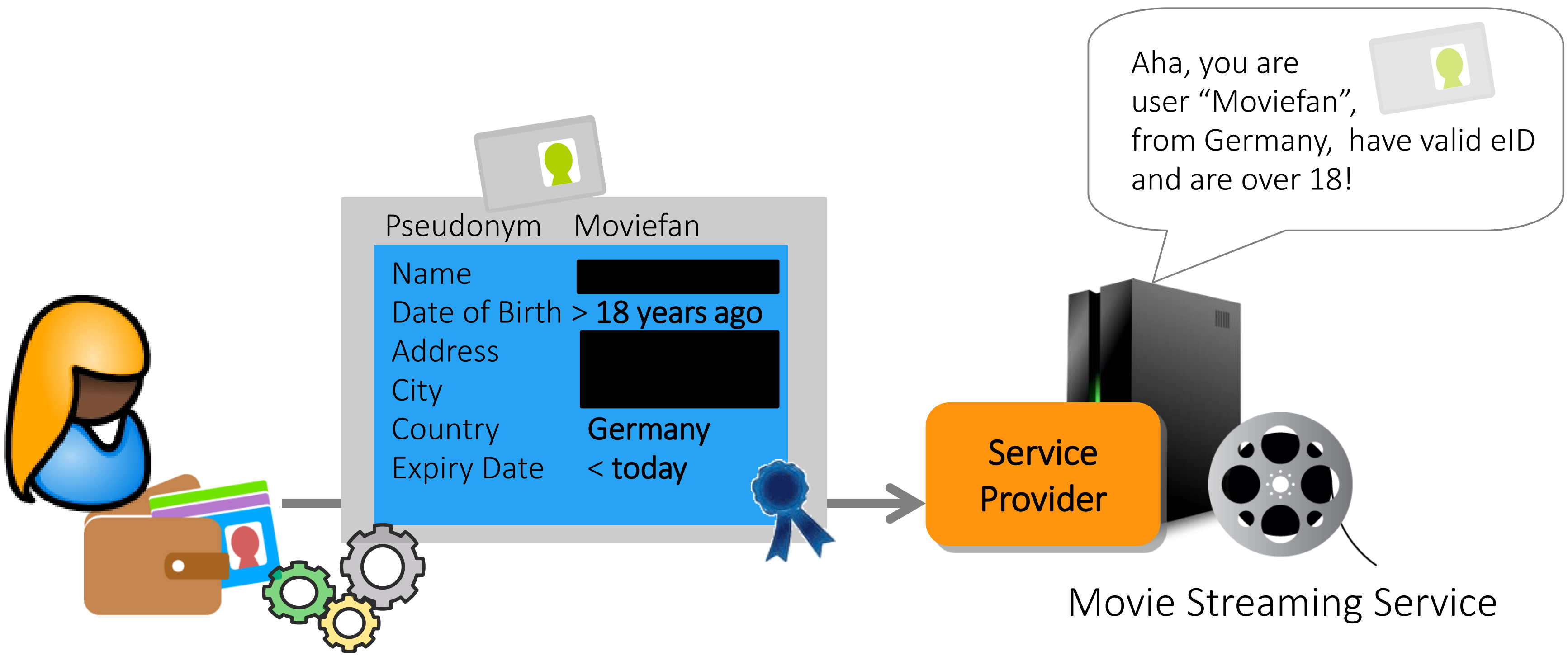
- Strong (user) authentication via certificates / attribute-based credentials
  - Many European countries have or will introduce eID cards
  - Desirable for security, but detrimental for privacy
  - Existing schemes require full information disclosure & user is linkable in all transactions

→ This is a privacy and security problem!
  - Acquired personal data requires protection
  - Linkability enables tracking & profiling of users



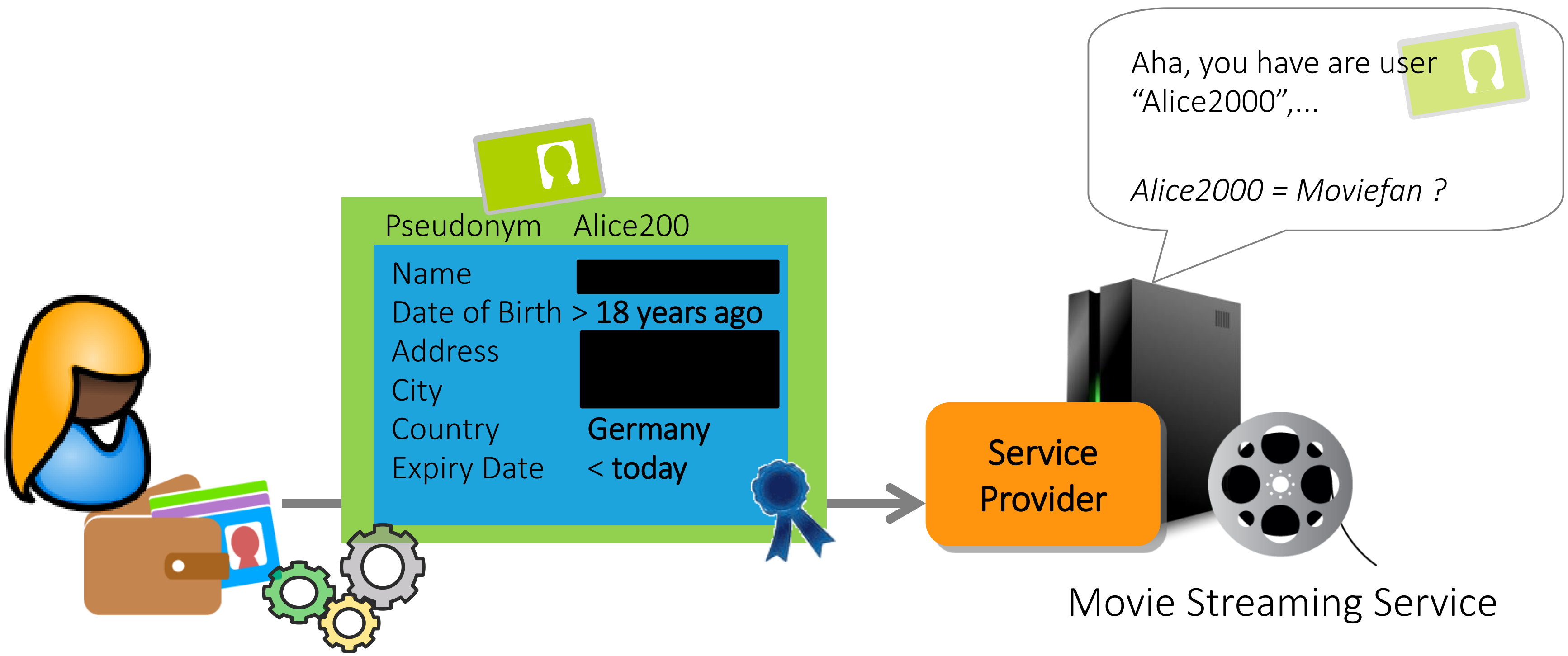
Privacy-Enhancing Credentials | Overview

- Strong yet privacy-friendly (user) authentication
- Envisioned by Chaum in 1981, first full scheme by Camenisch & Lysyanskaya in 2001
  - User / service provider can selectively decide which attributes to reveal / request
  - User can prove predicates over the attributes, e.g., “I'm over 18”



Privacy-Enhancing Credentials | Overview

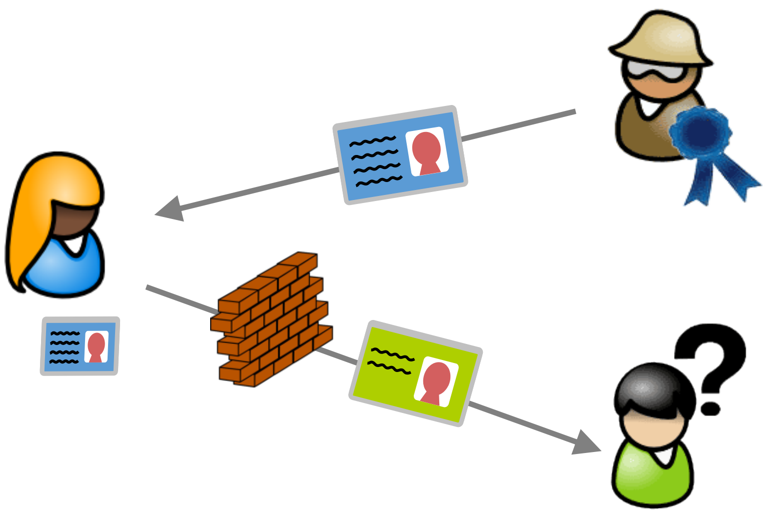
- Strong yet privacy-friendly (user) authentication
- Envisioned by Chaum in 1981, first full scheme by Camenisch & Lysyanskaya in 2001
  - User / service provider can selectively decide which attributes to reveal / request
  - User can prove predicates over the attributes, e.g., “I'm over 18”
  - Unlinkable authentication as default, linkability as an option



Privacy-Enhancing Credentials | Protocols

- Most prominent core-credential/signature schemes:

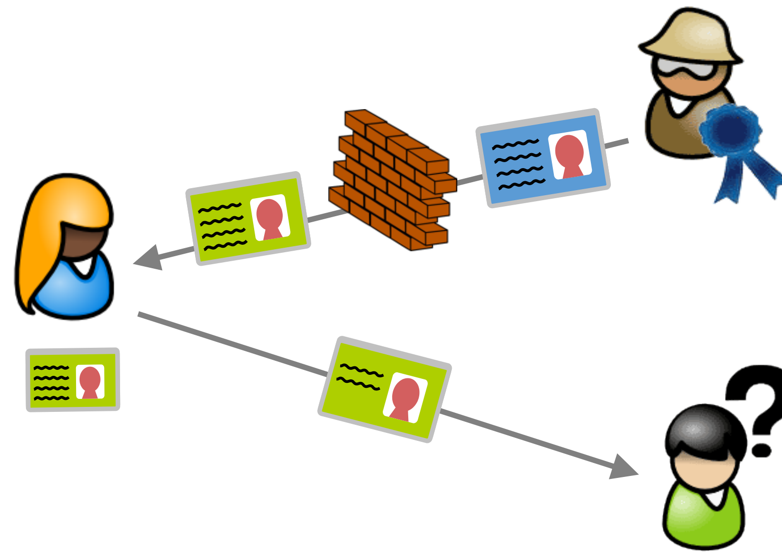
Identity Mixer (IBM)



Multi-use credentials

Zero-Knowledge Proofs
Strong RSA, pairings (LRSW, qSDH)

U-Prove (Microsoft)



One-time use credentials
(multi-use via batch-issuance)

Blind Signatures
RSA, DL

Privacy-Enhancing Credentials | Extended Features

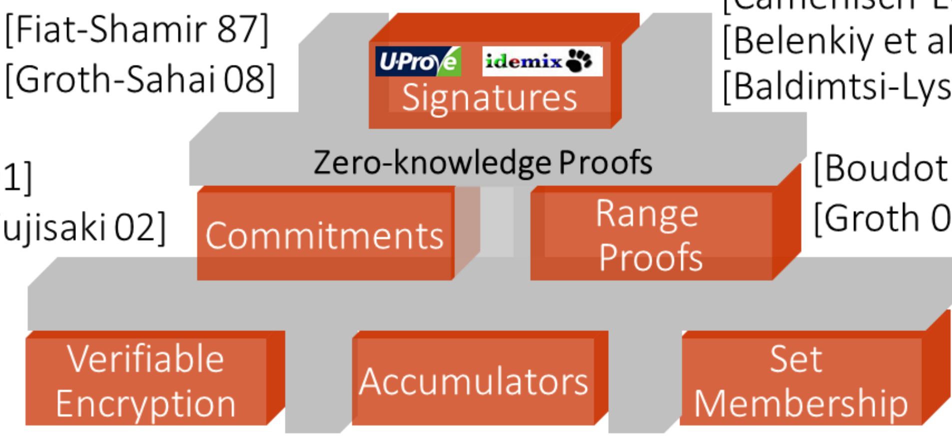
- Many more extensions & properties:
  - Revocation, multi-credential proofs, issuance with carry-over attributes, conditional disclosure, „symmetric“ credentials
- Various cryptographic realizations
- Provable security for (some) building blocks, and different combinations / properties
- No security model covering *all* features exists yet

Σ protocols
[Fiat-Shamir 87]
[Groth-Sahai 08]

[Brands 99]
[Camenisch-Lysyanskaya 02]
[Camenisch-Lysyanskaya 04]
[Belenkiy et al 09]
[Baldimtsi-Lysyanskaya13]

[Pedersen 91]
[Damgard-Fujisaki 02]

[Boudot 00]
[Groth 05]



[Cramer-Shoup 98]
[Camenisch-Damgard 00]
[Camenisch-Shoup 03]

[Camenisch-Lysyanskaya 02]
[Nguyen 05]
[Camenisch-Kohlweiss-Soriente 09]

[Camenisch-Chaabouni-Shelat 08]

Privacy-Enhancing Credentials | Generic Framework

- Hard to deploy: complex, multitude of protocols, features, inconsistent naming, ...
- ABC4Trust (EU project 2010-2014): joint project with Microsoft, ALX, Miracle, ULD, ...
  - Goal: generic and easy-to-use credential framework

1) Identify crucial features & if necessary adapt underlying technologies

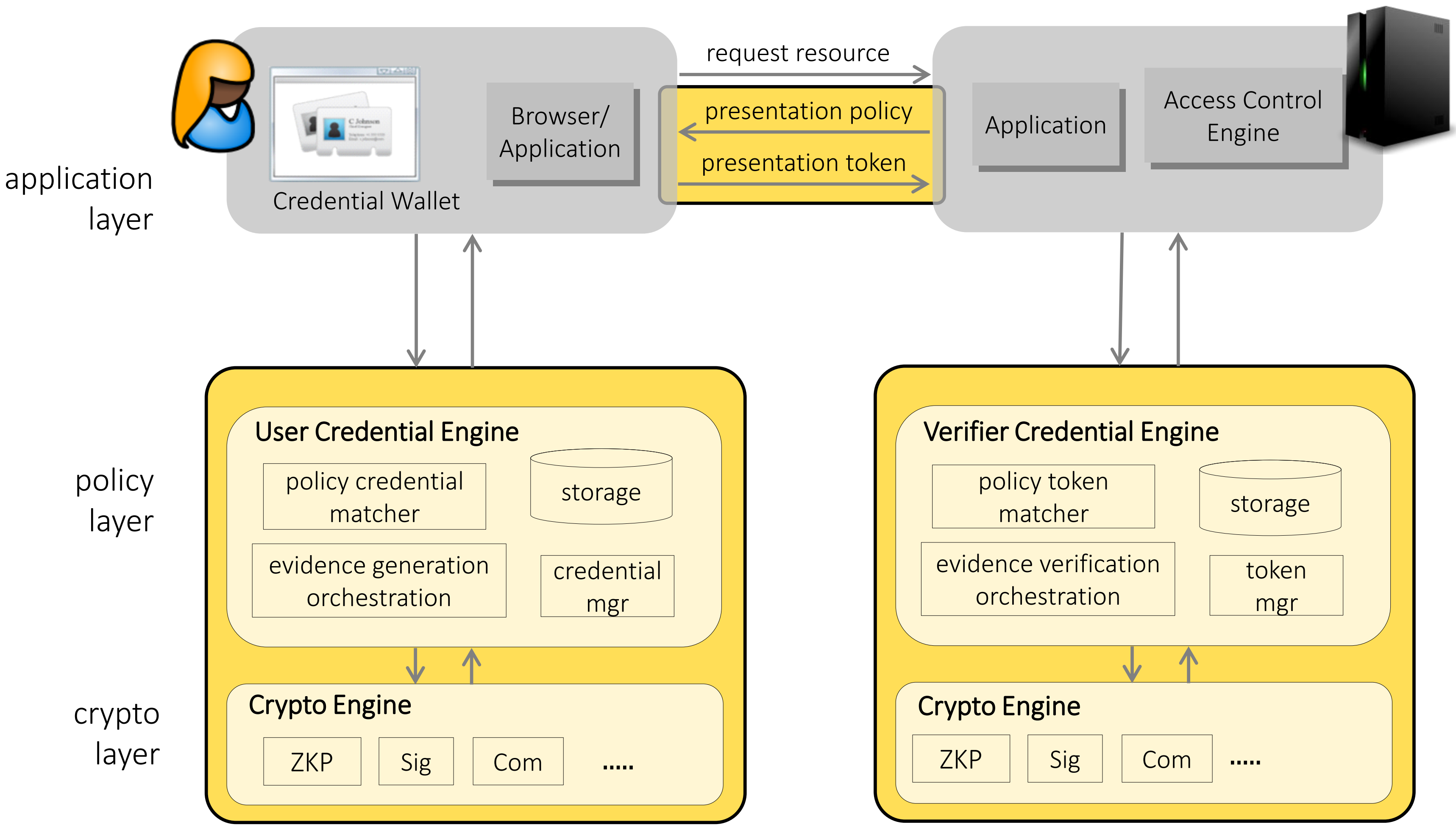
2) Formal security model for complex & generic credential system [CKL+15]
   - (Scope-exclusive) pseudonyms, selective disclosure, equality predicates, key-binding, carry-over attributes, revocation
   - Formal definitions of generic building blocks & generic construction

3) Provide unified and simple access to credential systems

4) Demonstrate practicality via real-world piloting

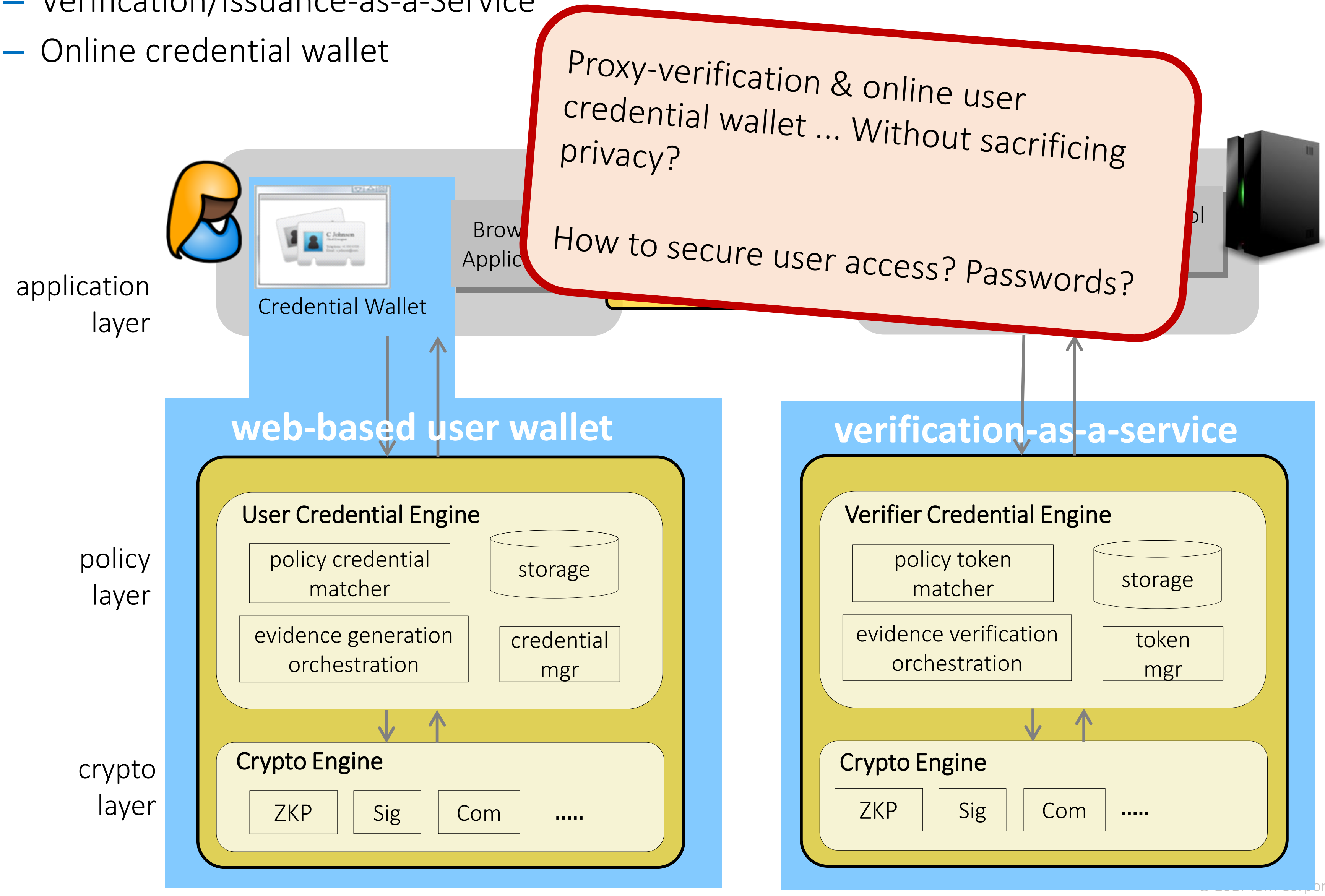
Privacy-Enhancing Credentials | Generic Framework

- Technology-independent & „easy-to-use“ framework
  - Comprehensive & standardized language framework
  - Technology-agnostic credential & policy handling on top of crypto engine
  - Generic, automated crypto engine



Privacy-Enhancing Credentials | Deployment

- Deployment still too complicated – simpler via „cloudification“
  - Verification/Issuance-as-a-Service
  - Online credential wallet

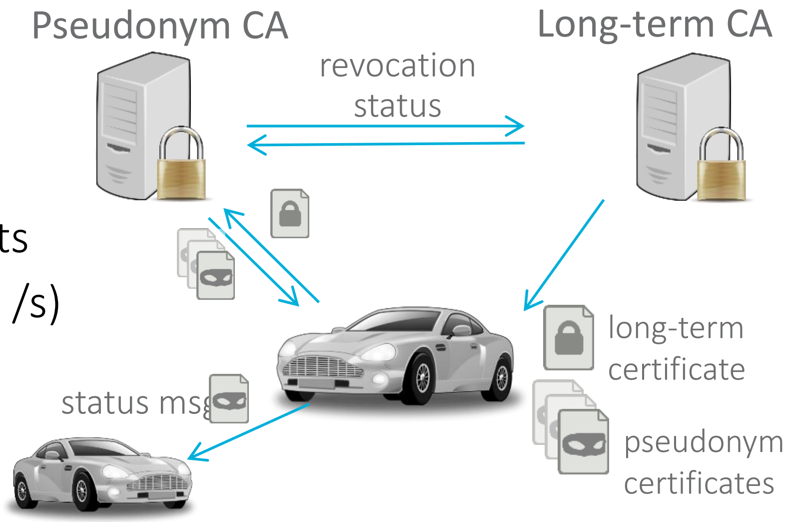


Privacy-Enhancing Credentials | Future Directions

- Use-case specific, tailored versions instead of generic solution - the „DAA approach“

- For instance, V2X communication: between vehicles (V2V) and infrastructure (V2I)
  - Security needs: authentication & privacy
  - Current approach: pseudonym CA
  - Privacy credentials instead of pseudonym CA
  - Stringent computational/bandwidth requirements (300 bytes max, 10 signatures/s, 100 verifications /s)

  → more research needed to meet size & efficiency requirements



- " Functionality-preserving" applications of privacy-preserving authentication
  - E.g., Data exchange and correlation of pseudonymized data

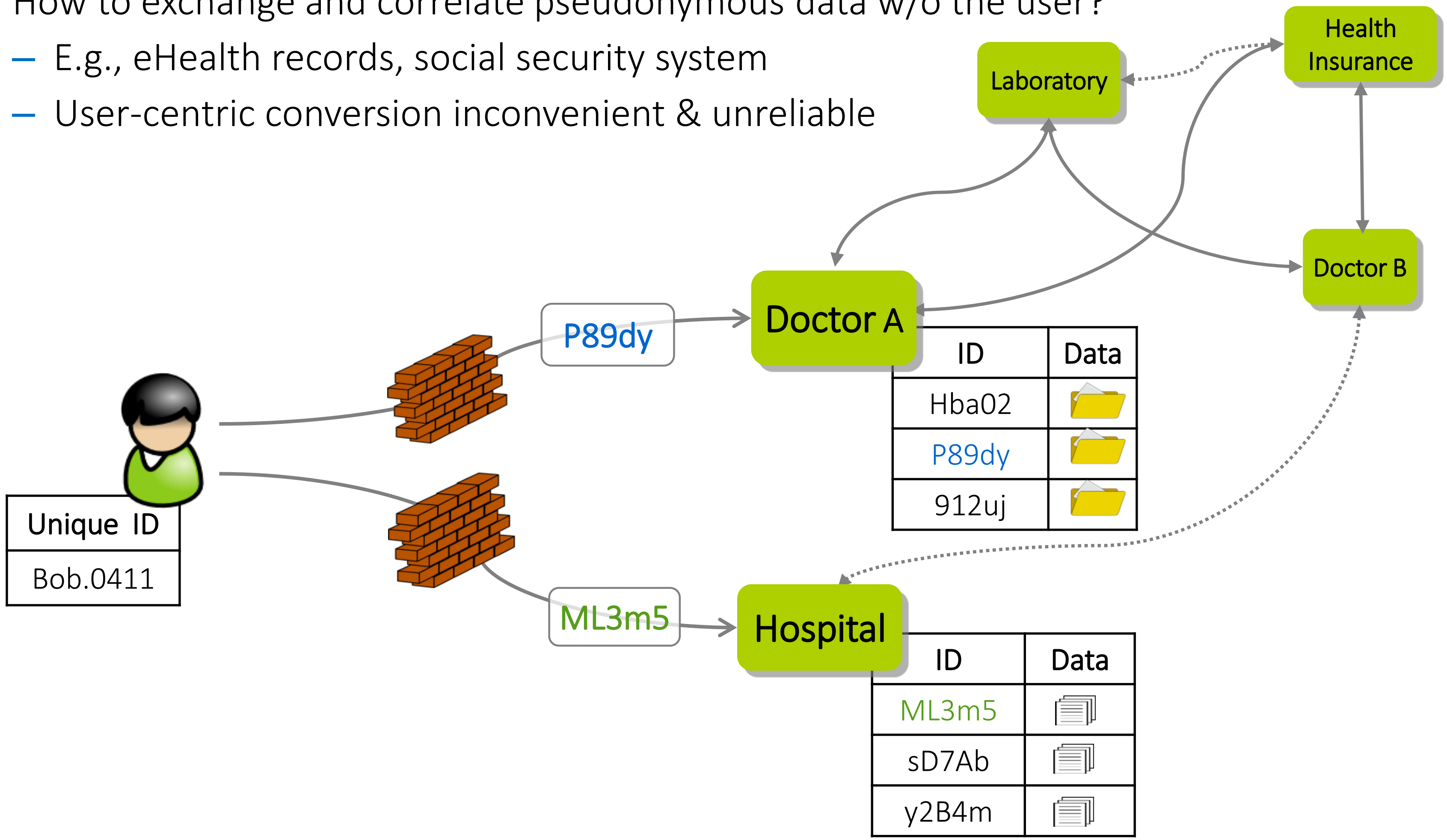
- Direct Anonymous Attestation
  – hardware-based attestation

- Anonymous Credentials
  – privacy-preserving (user) authentication

- Pseudonym Systems
  – privacy-preserving data exchange

Pseudonym System | Motivation

- How to exchange and correlate pseudonymous data w/o the user?
  - E.g., eHealth records, social security system
  - User-centric conversion inconvenient & unreliable

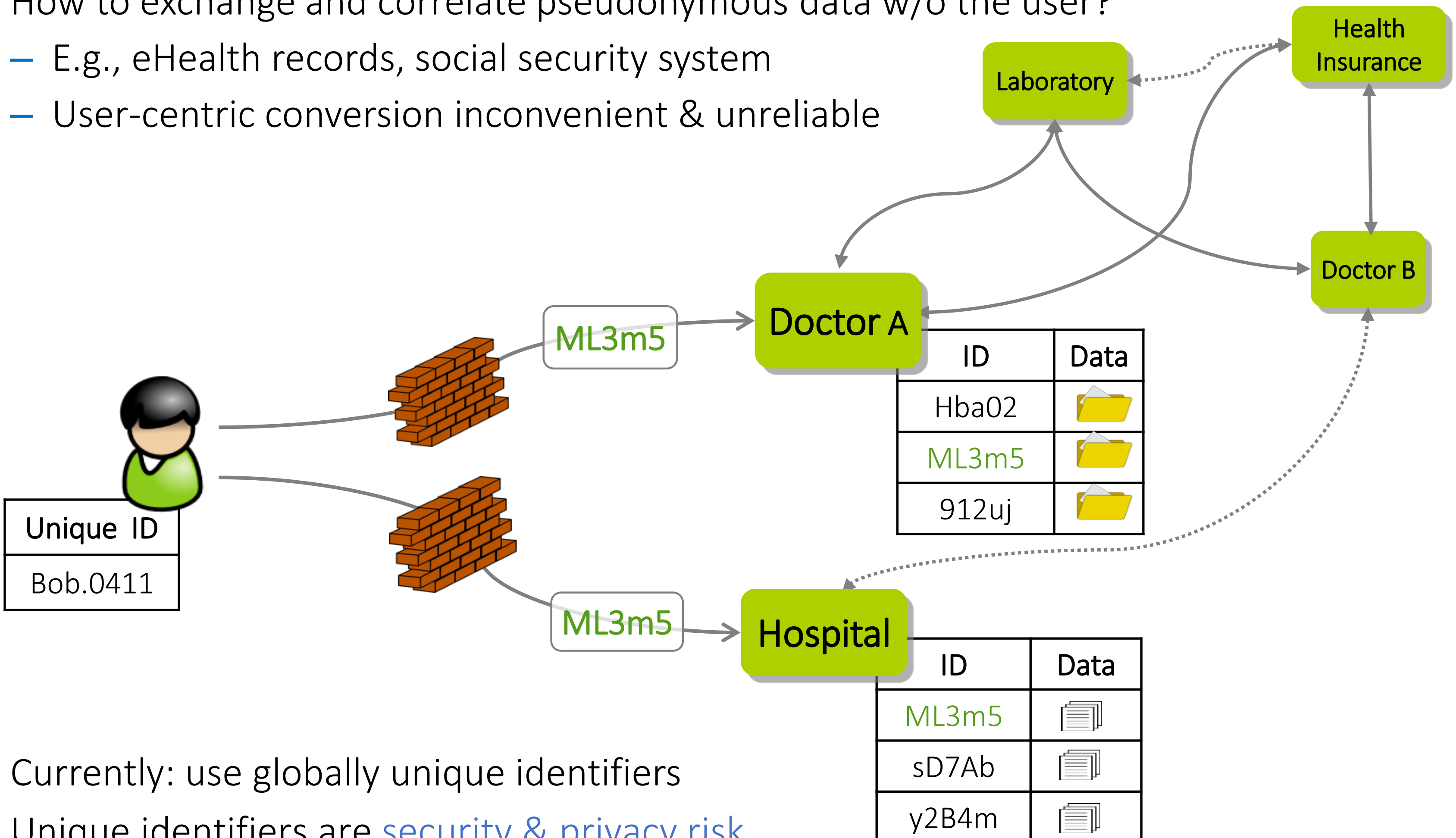


ID	Data
Hba02	
P89dy	
912uj	

ID	Data
ML3m5	
sD7Ab	
y2B4m	

Pseudonym System | Unique Pseudonyms

- How to exchange and correlate pseudonymous data w/o the user?
  - E.g., eHealth records, social security system
  - User-centric conversion inconvenient & unreliable



- Currently: use globally unique identifiers
- Unique identifiers are security & privacy risk
  - No control about data exchange & usage
  - If associated data is lost, all pieces can be linked together
  - User is fully traceable

Pseudonym System | *Trusted* Converter

- Central entity – the converter – can link & convert pseudonyms
- Design of newly introduced Japan eID / social security number system (?)
- Austrian eID allows pseudonyms = H(Enc(k; uid), Doctor A)
  - Converter could keep hash table / use encryption

Doctor A

ID	Data
Hba02	
P89dy	
912uj	

Record of P89dy from Hospital?

Converter

Main ID	Doctor A	Hospital
Alice.1210	Hba02	7twnG
Bob.0411	P89dy	ML3m5
Carol.2503	912uj	sD7Ab

Record of ML3m5 ?

Hospital

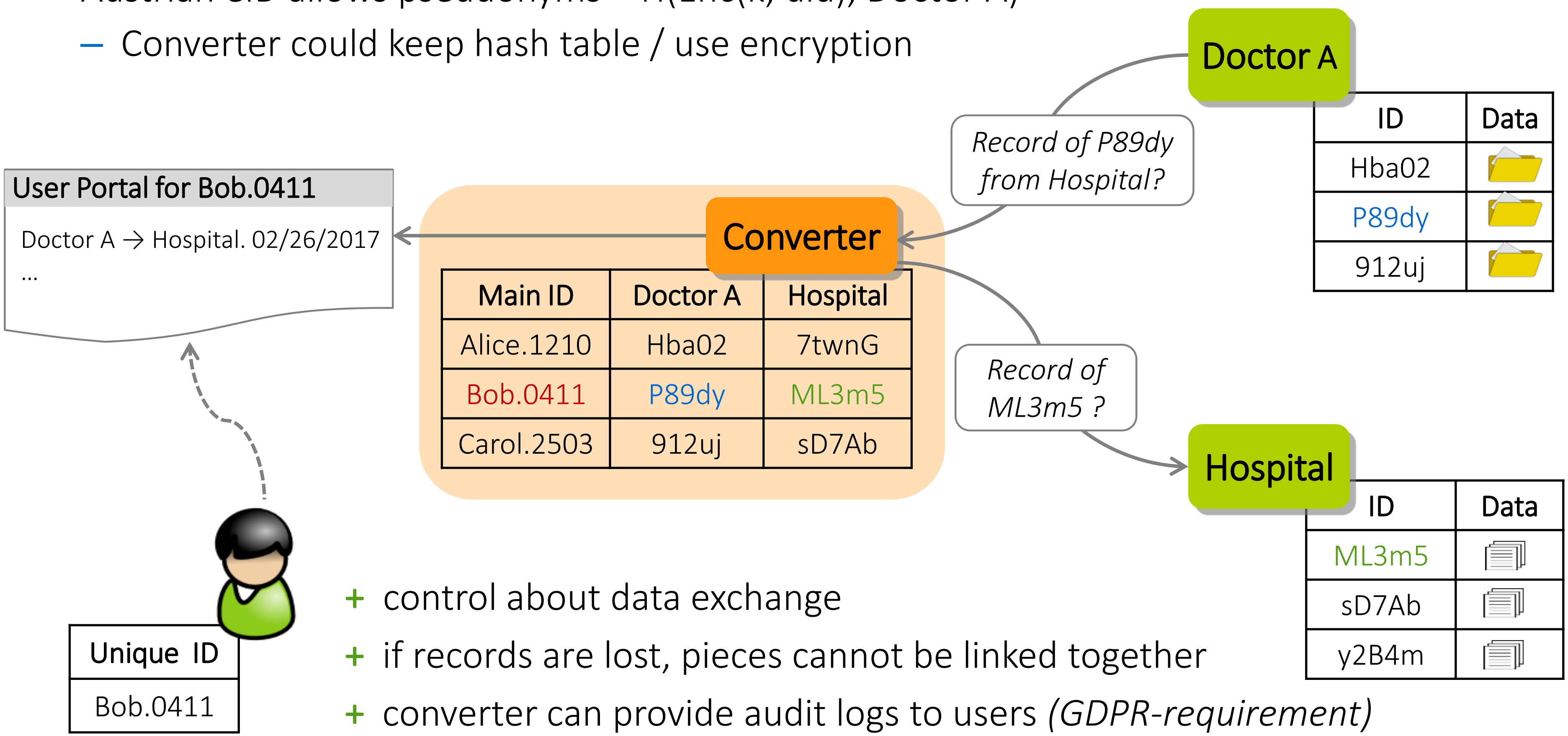
ID	Data
ML3m5	
sD7Ab	
y2B4m	

\+ control about data exchange

\+ if records are lost, pieces cannot be linked together

Pseudonym System | *Trusted* Converter

- Central entity – the converter – can link & convert pseudonyms
- Design of newly introduced Japan eID / social security number system (?)
- Austrian eID allows pseudonyms = H(Enc(k; uid), Doctor A)
  - Converter could keep hash table / use encryption



Main ID	Doctor A	Hospital
Alice.1210	Hba02	7twnG
Bob.0411	P89dy	ML3m5
Carol.2503	912uj	sD7Ab

ID	Data
Hba02	
P89dy	
912uj	

ID	Data
ML3m5	
sD7Ab	
y2B4m	

Unique ID
Bob.0411

\+ control about data exchange

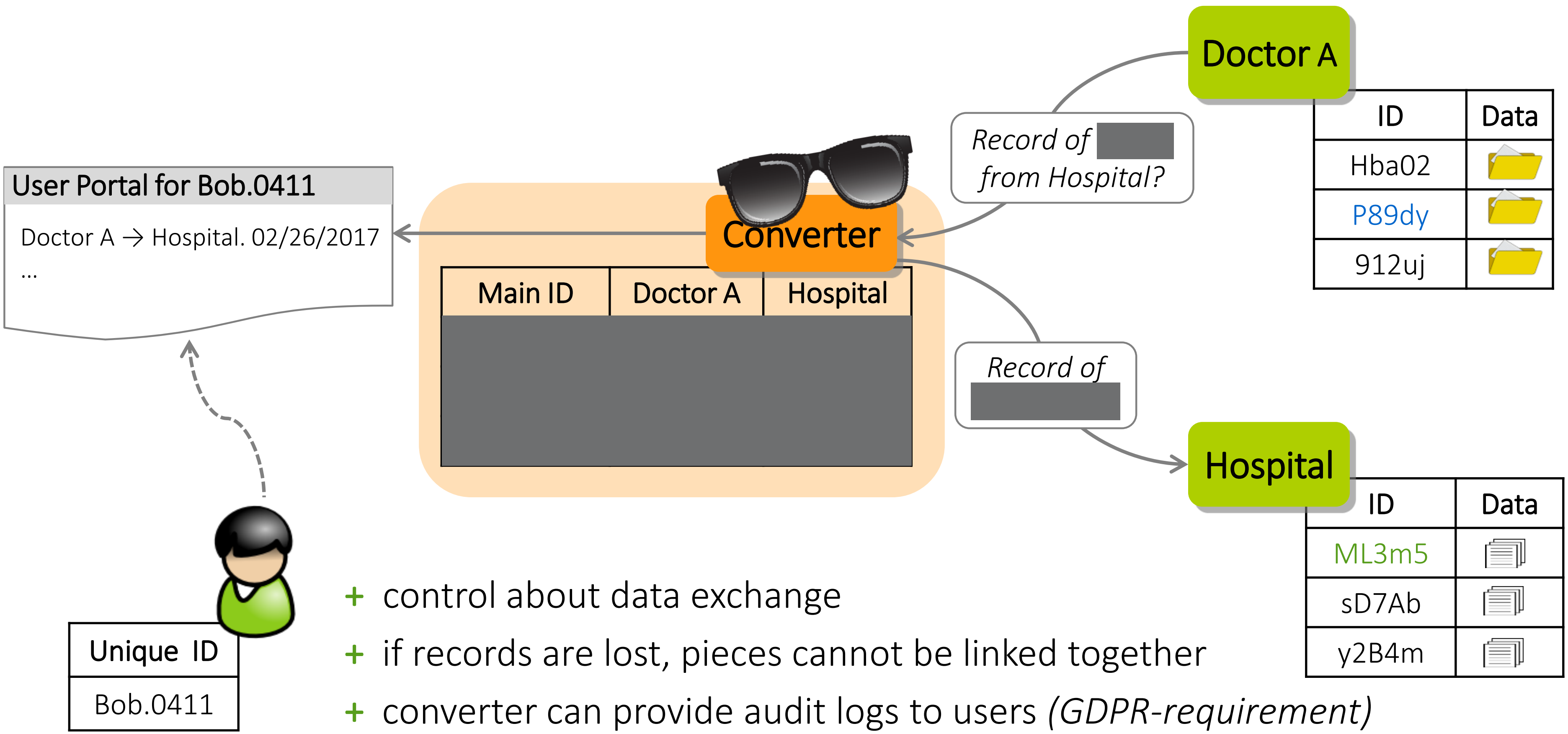
\+ if records are lost, pieces cannot be linked together

\+ converter can provide audit logs to users *(GDPR-requirement)*

– **converter learns all request & knows all correlations**

Pseudonym System | Privacy-Preserving Solution

- Central entity – the converter – can link & convert pseudonyms
  ...but does so in a *blind* way



\+ control about data exchange

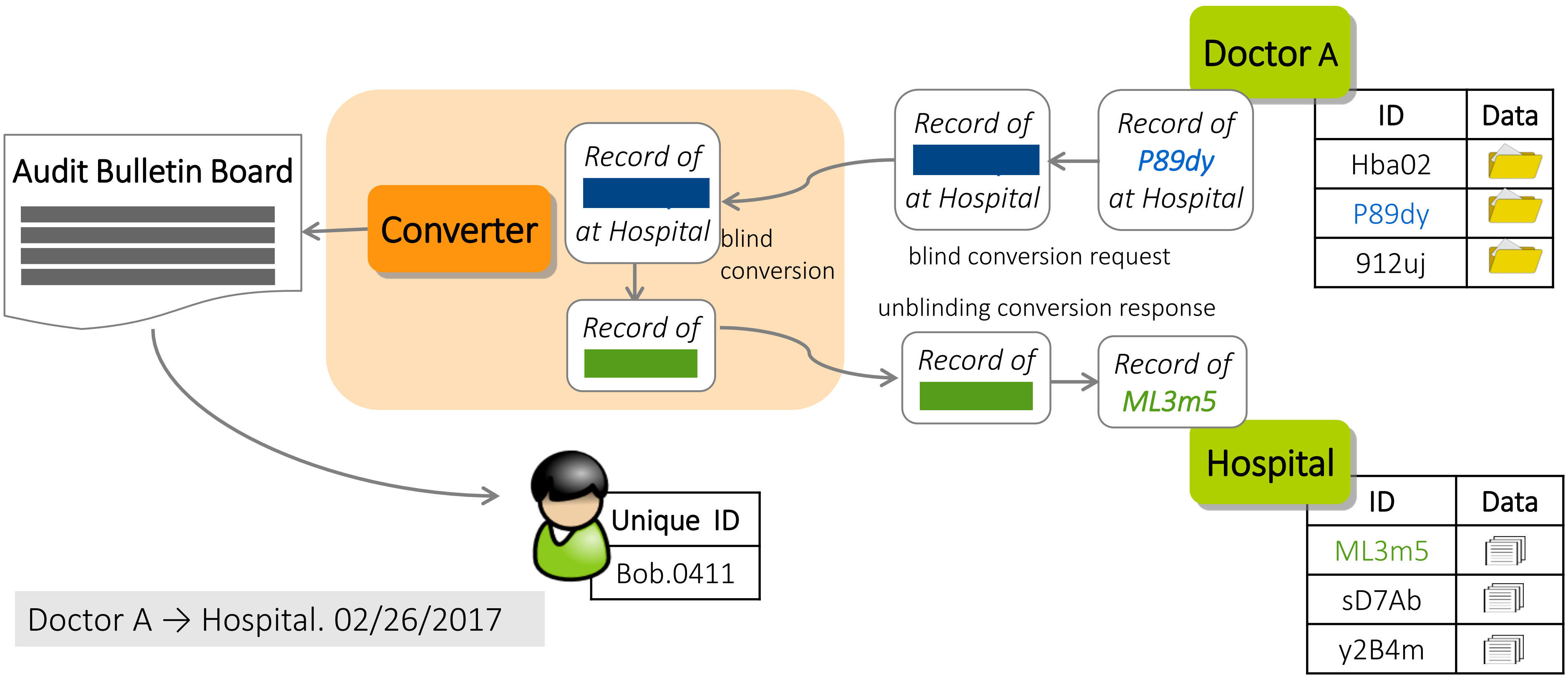
\+ if records are lost, pieces cannot be linked together

\+ converter can provide audit logs to users *(GDPR-requirement)*

– **converter learns all request & knows all correlations**

Pseudonym System | Privacy-Preserving Solution

- Converter can link & convert pseudonyms, but does so in a blind yet consistent way
- Every conversion triggers blind generation of audit log entry



- Core model and protocol [CL15, CL17], many open research problems
- Similar idea by Verheul and Jacobs: polymorphic pseudonyms
  - Implementation and pilot in medical research project at the Radboud University

Summary

- Mature privacy-enhancing solutions exist – privacy and functionality are not exclusive
- GDPR creates a great practical demand for privacy-preserving mechanisms to manage and protect personal data – data minimisation, auditability, ...

- Open challenges:
  - Provably secure protocols
    - Simpler security models and proofs
    - Security models & frameworks for leveraging trusted hardware
  - Use-case specific protocols (e.g., V2X communication)
    - Better efficiency
  - Usability
    - Simple, yet secure and privacy-friendly deployment models
  - Long-term privacy
    - Quantum-safe protocols

Thanks! Questions?

anj@zurich.ibm.com

References

[BCC04] Brickell, Camenisch, Chen. Direct anonymous attestation. ACM CCS 04
[BCL08] Brickell, Chen, Li. A new direct anonymous attestation scheme from bilinear maps. Trust 2008
[BCL09] Brickell, Chen, Li. Simplied security notions of DAA and a concrete scheme from pairings. Int. J. Inf. Sec., 2009.
[BFG+13] Bernhard, Fuchsbauer, Ghadafi, Smart, Warinschi. Anonymous attestation with user-controlled linkability. Int. J. Inf. Sec., 2013.
[BL07] Brickell, Li. Enhanced privacy id: a direct anonymous attestation scheme with enhanced revocation capabilities. WPES 2007
[BL10] Brickell, Li. A pairing-based DAA scheme further reducing TPM resources. Trust 2010
[BL11] Brickell, Li. Enhanced privacy ID from bilinear pairing for hardware authentication and attestation. IJIPSI, 1(1):3{33, 2011.
[CCD+17] Camenisch, Chen, Drijvers, Lehmann, Novick, Urian. One TPM to Bind Them All: Fixing TPM2.0 for Provably Secure Anonymous Attestation. IEEE S&P 2017
[CDL16a] Camenisch, Drijvers, Lehmann. Universally composable direct anonymous attestation. PKC 2016
[CDL16b] Camenisch, Drijvers, Lehmann. Anonymous attestation using the strong diffie hellman assumption revisited. Trust 2016
[CDL17] Camenisch, Drijvers, Lehmann. Anonymous attestation with subverted TPMs. CRYPTO 2017
[CKL+15] Camenisch, Krenn, Lehmann, Mikkelsen, Neven, Pedersen. Formal Treatment of Privacy-Enhancing Credential Systems. SAC 2015
[CL15] Camenisch, Lehmann. (Un)linkable Pseudonyms for Governmental Databases. CCS 2015
[CL17] Camenisch, Lehmann. Privacy-Preserving User-Auditable Pseudonym Systems IEEE. EuroS&P 2015
[C10] Chen. A DAA scheme requiring less TPM resources. Inscrypt 2010
[CF08] X. Chen, Feng. Direct anonymous attestation for next generation TPM. JCP, 3(12):43{50, 2008.
[CMS08a] Chen, Morrissey, Smart. Pairings in trusted computing (invited talk). PAIRING 2008.
[CMS08b] Chen, Morrissey, Smart. On proofs of security for DAA schemes. Provable Security 2008.
[CMS09] Chen, Morrissey, Smart. DAA: Fixing the pairing based protocols. ePrint Archive, Report 2009/198.
[CPS10] Chen, Page, Smart. On the design and implementation of an efficient DAA scheme. CARDIS 2010