

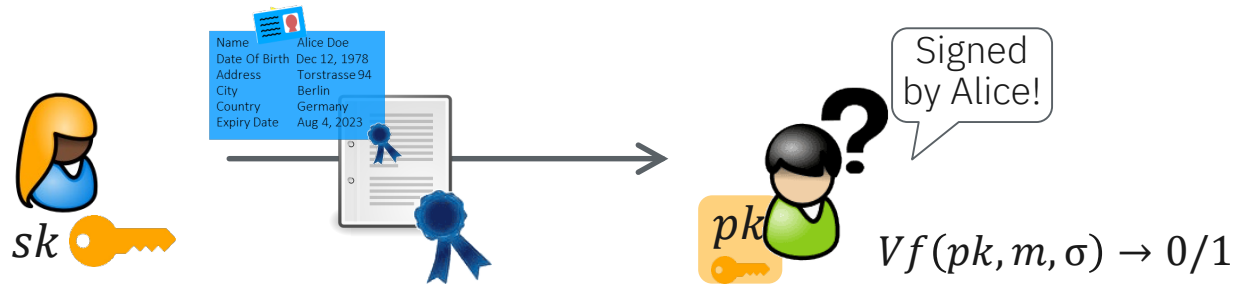


# Privacy-Enhancing Authentication Concepts, Applications & New Advances

Anja Lehmann  
Hasso-Plattner-Institute, University of Potsdam

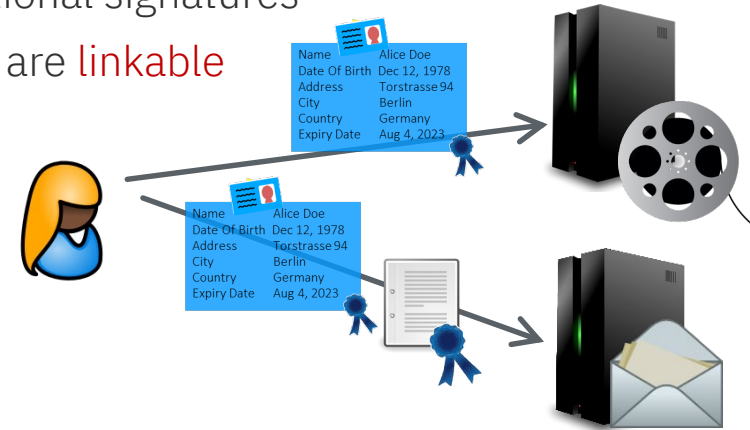
- Authentication: Ensure that information has not been tampered with and has a certain origin
- Cryptographic solution: **digital signatures**

Signature alone is often not sufficient, we also need a **certificate** from a trusted authority that binds the public key to some context



- Many applications: PKI, server-side authentication, certified updates
- Signatures also yield efficient (user) **identification**:
  - Register public key, authenticate by signing a fresh nonce

- Bad for privacy: signatures “leak” identity of the signer
  - Problem when users sign or authenticate with conventional signatures
    - Reveal their identity & all signatures/identifications are **linkable**
    - Could use individual keys everywhere
- Challenge in key management & certification



- Problem for any application that requires authentication of „user-near“ data  
e.g. V2V authentication – linkability allows to trace movements of driver



# Privacy-Enhancing Authentication

- First description of privacy-enhancing authentication by David Chaum

1981. *Untraceable electronic mail, return addresses, and digital pseudonyms* & 1984. *A New Paradigm for Individuals in the Information Age*

- Milestone: 2001 Jan Camenisch & Anna Lysyanskaya

*An Efficient System for Non-transferable Anonymous Credentials with Optional Anonymity Revocation. Eurocrypt'01*



of message content for thousands of years [3]. Recently, some new solutions to the "key distribution problem" (the problem of providing each communicant with a secret key) have been suggested [2, 4], under the name of public key cryptography. Another cryptographic problem, "the traffic analysis problem" (the problem of keeping confidential who converses with whom, and when they converse), will become increasingly important with the growth of electronic mail. This paper presents a solution to the traffic analysis problem that is based on public key cryptography. Baran has solved the traffic analysis problem for networks [1], but requires each participant to trust a common authority. In contrast, systems based on the solution advanced here can be compromised only by subversion or conspiracy of all of a set of authorities. Ideally, each participant is an authority.

Technical Note  
Programming Techniques  
and Data Structures

R. Rivest  
Editor

## Untraceable Electronic Mail, Return Addresses, and Digital Pseudonyms

David L. Chaum  
University of California, Berkeley

A technique based on public key cryptography is presented that allows an electronic mail system to hide who a participant communicates with as well as the content of the communication—in spite of an unsecured underlying telecommunication system. The technique does not require a universally trusted authority. One correspondent can remain anonymous to a second, while allowing the second to respond via an untraceable return address.

The technique can also be used to form rosters of untraceable digital pseudonyms from selected applications. Applicants retain the exclusive ability to form digital signatures corresponding to their pseudonyms. Elections in which any interested party can verify that the ballots have been properly counted are possible if anonymously mailed ballots are signed with pseudonyms from a roster of registered voters. Another use allows an individual to correspond with a record-keeping organization under a unique pseudonym which appears in a

The following two sections introduce the notation and assumptions. Then the basic concepts are introduced for some special cases involving a series of one or more authorities. The final section covers general purpose mail networks.

### Notation

Someone becomes a user of a public key cryptosystem (like that of Rivest, Shamir, and Adleman [5]) by creating a pair of keys  $K$  and  $K^{-1}$  from a suitable randomly generated seed. The public key  $K$  is made known to the other users or anyone else who cares to know it; the private key  $K^{-1}$  is never divulged. The encryption of  $X$  with key  $K$  will be denoted  $K(X)$ , and is just the image of  $X$  under the mapping implemented by the cryptographic algorithm using key  $K$ . The increased utility of these algorithms over conventional algorithms results because the two keys are inverses of each other, in the sense that

$$K^{-1}(K(X)) = K(K^{-1}(X)) = X.$$

A message  $X$  is sealed with a public key  $K$  so that only the holder of the private key  $K^{-1}$  can discover its content. If  $X$  is simply encrypted with  $K$ , then anyone could verify a guess that  $Y = X$  by checking whether  $K(Y) = K(X)$ . This threat can be eliminated by attaching a large string of random bits  $R$  to  $X$  before encrypting. The sealing of  $X$  with  $K$  is then denoted  $K(R, X)$ . A user signs some material  $X$  by prepending a large constant  $C$  (all zeros, for example) and then encrypting with its private key, denoted  $K^{-1}(C, X) = Y$ . Anyone can verify that  $Y$  has been signed by the holder of  $K^{-1}$  and determine the sender  $X$ , by forming  $K(Y) = C, X$ , and checking

pseudonym systems

About 146.000 results (0,05 sec)

anonymous credentials

About 257.000 results (0,05 sec)

privacy credentials

About 555.000 results (0,03 sec)

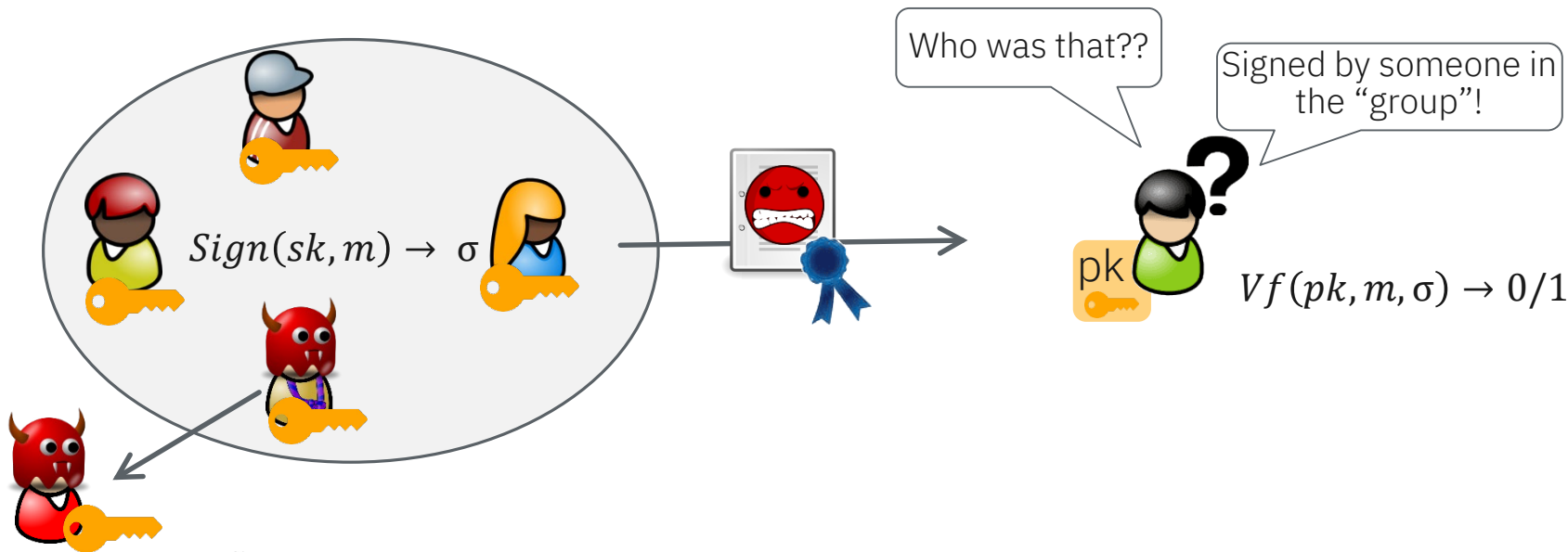
group signatures



About 2.420.000 results (0,03 sec)



ions  
approach taken here is based on two important ons:  
ations  
February 1981  
Volume 24  
Number 2  
of the ACM

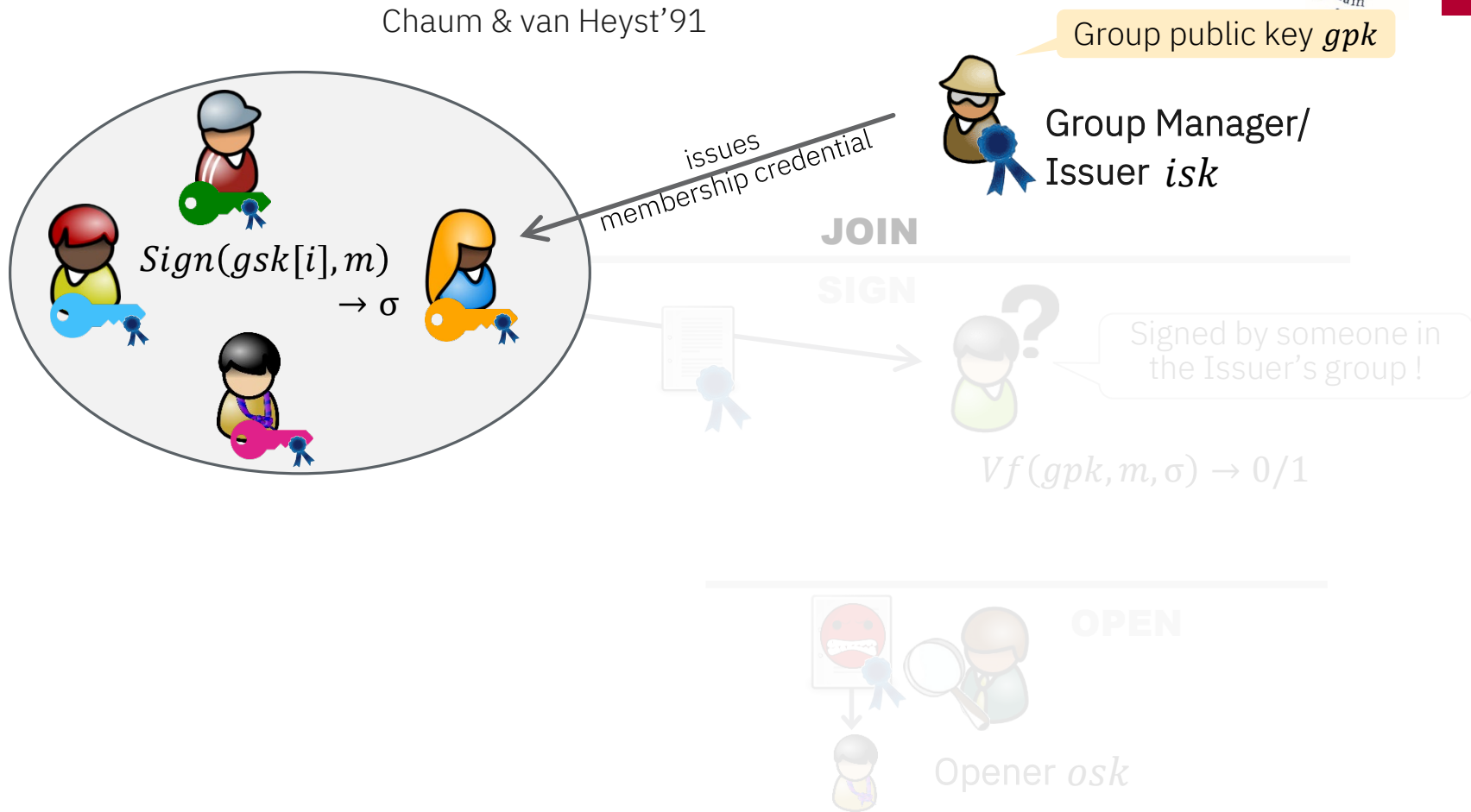
- Group Signatures
  - General idea, constructions & new advances
  - Adding pseudonyms & attributes → anonymous credentials
- New approaches to balance privacy & accountability/utility
- Where do we stand in terms of real-world usage?

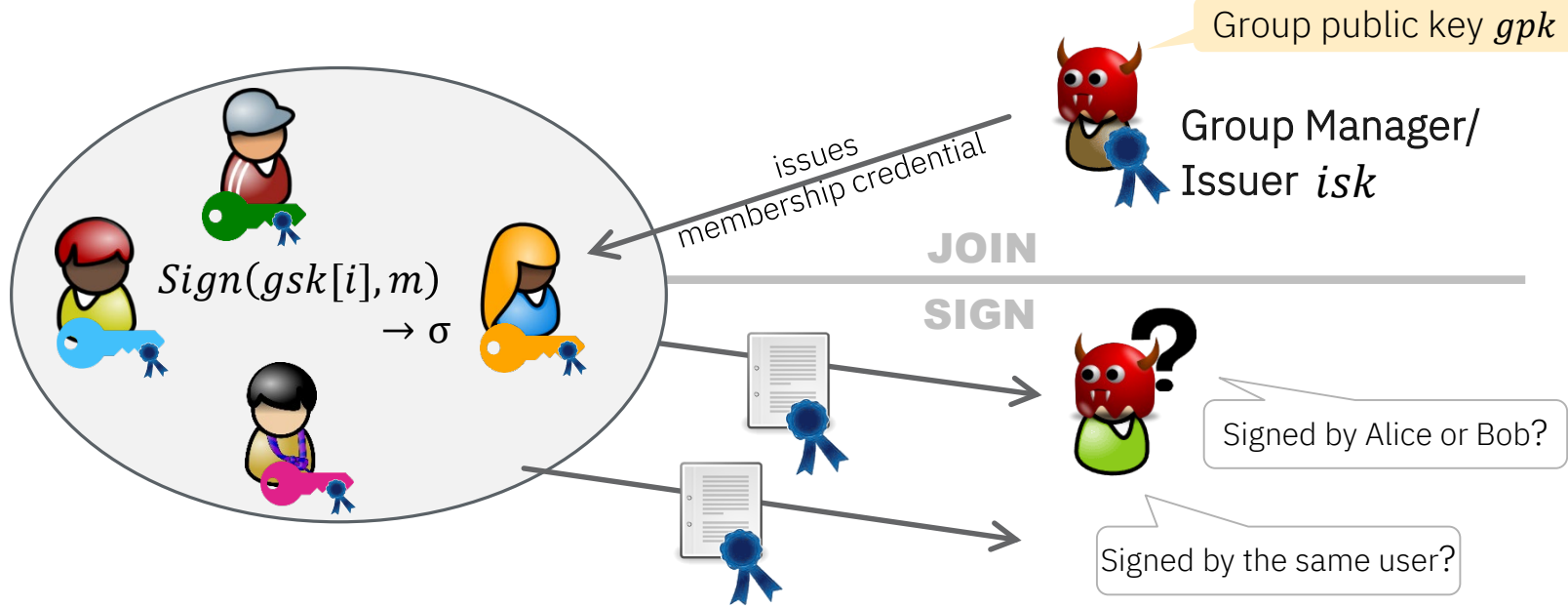


- Privacy  : Doesn't leak any information about signer
- Security  : Access to "group" not controlled  
No way to reveal signer in case of abuse (bug or feature?)

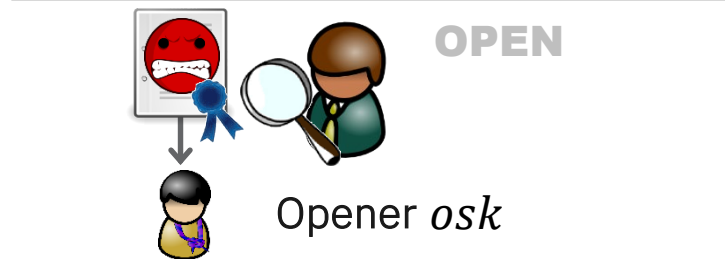
# Group Signatures | High-Level Idea

Chaum & van Heyst'91



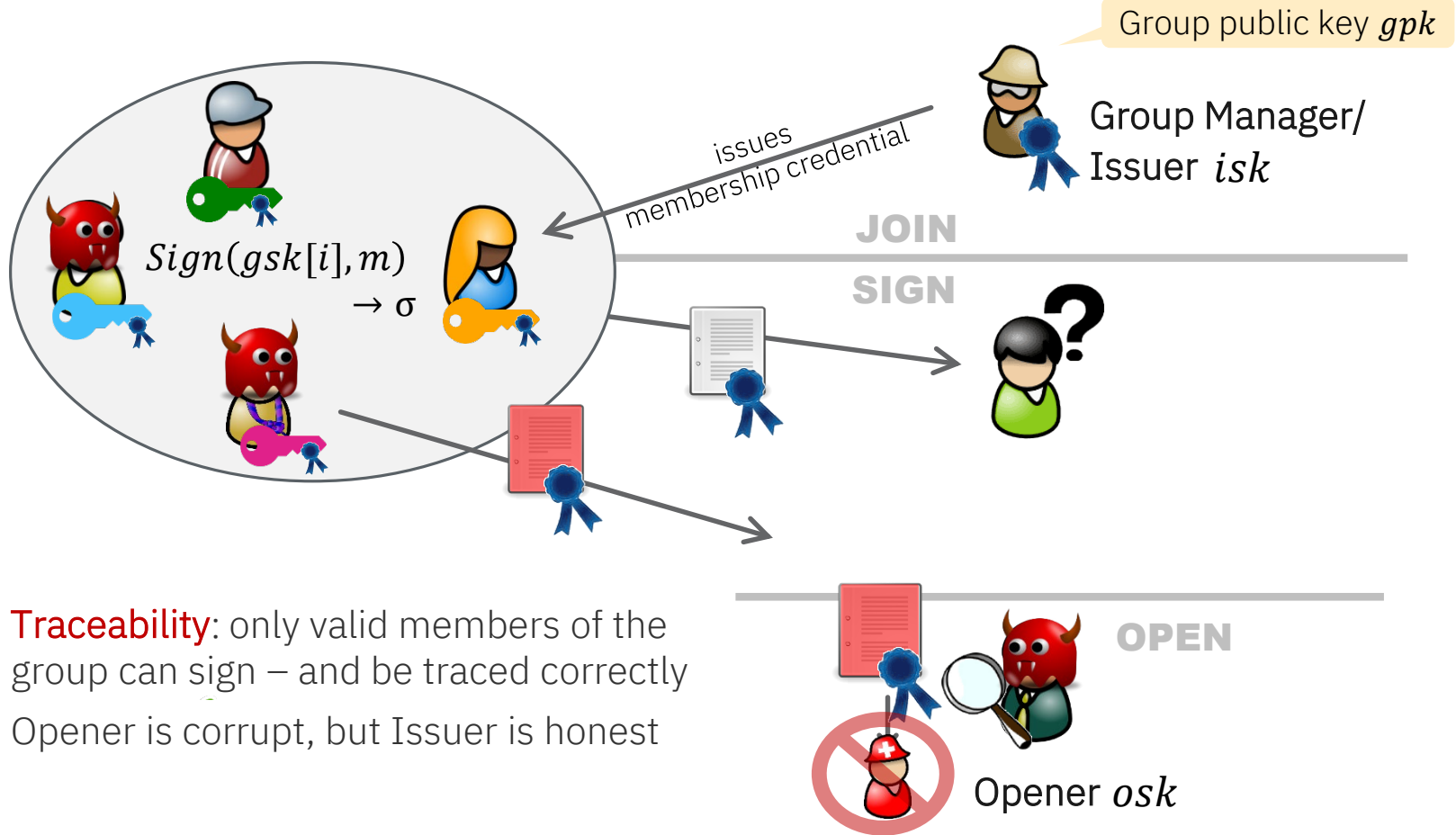


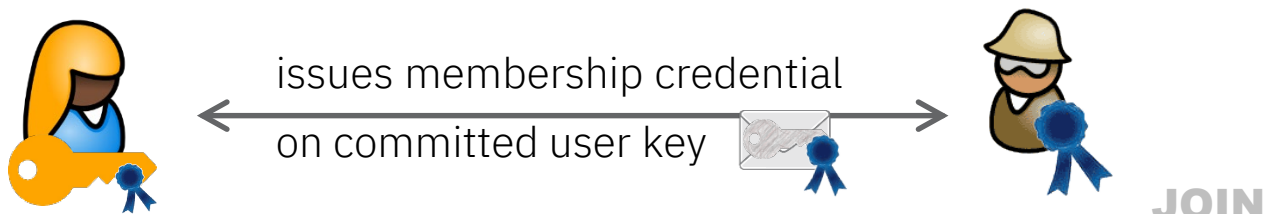
- **Anonymity:** Signatures don't leak info about signer = unlinkability  
Issuer can be corrupt  
Opener must be honest



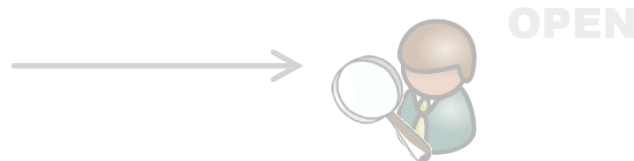
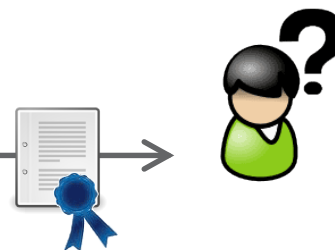


# Group Signatures | Traceability (~Unforgeability)

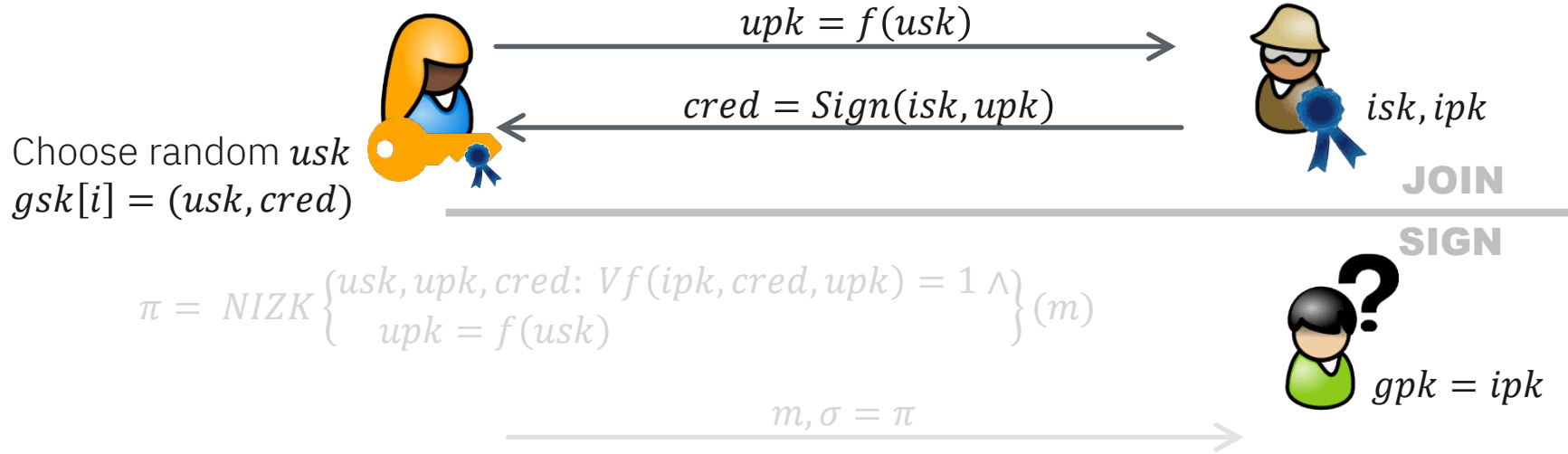




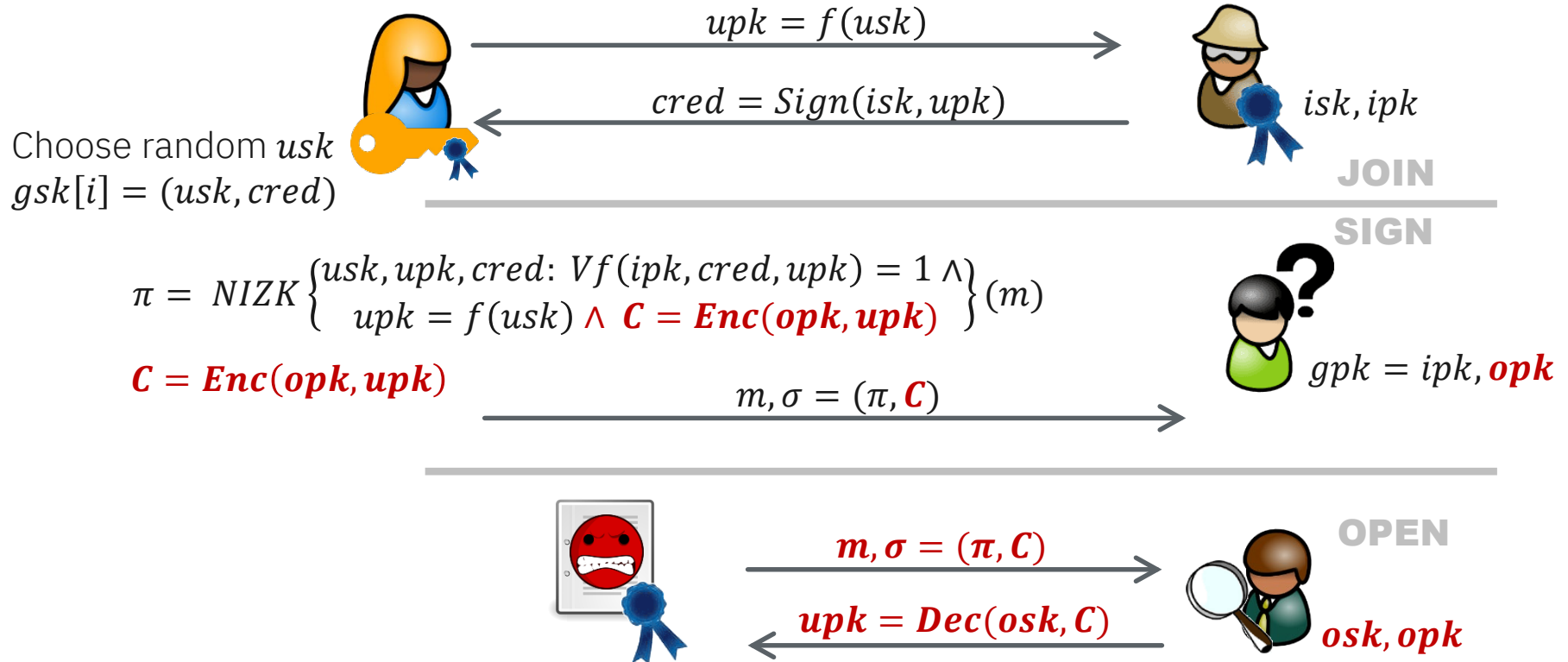
proof of knowledge of user key  
& membership credential

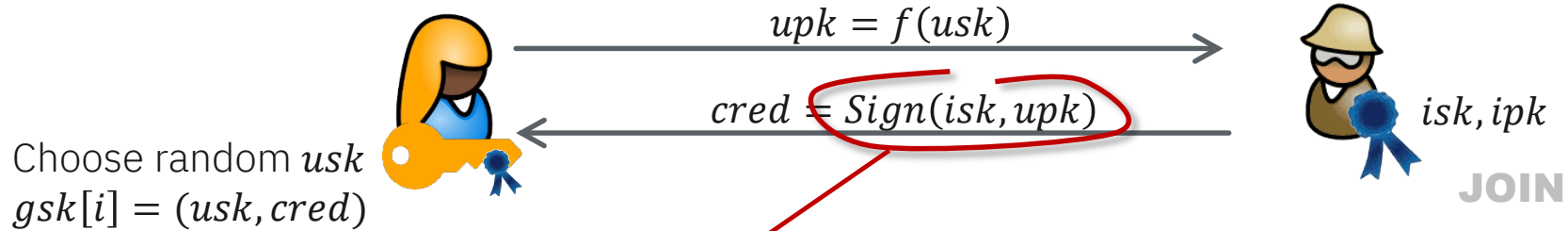


# Group Signatures | Constructions



# Group Signatures | Constructions

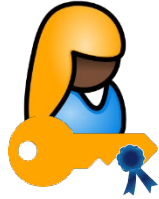




- Schemes mainly differ in the signature scheme that is used for the membership credential
  - Signatures on committed messages  $cred = \text{Sign}(isk, upk) = \text{"Sign}(isk, usk)\text{"}$
  - Efficient proofs of knowledge of a signature – ideal: re-randomizable signature
  - (Practical) Instantiations:  
[CL'01] (strong RSA), [CL'04] (LRSW), [BBS'04] (q-SDH), [PS'16/18] (q-MSDH-1)

# Group Signatures | Standard Approach for Opening

- Most common approach: Sign & Encrypt & Prove [BMW'03]



$$\pi = \text{NIZK} \left\{ \begin{array}{l} usk, upk, cred: Vf(ipk, cred, upk) = 1 \wedge \\ upk = f(usk) \wedge C = \text{Enc}(opk, upk) \end{array} \right\} (m)$$

$$C = \text{Enc}(opk, upk)$$

$$gsk[i] = (usk, cred)$$

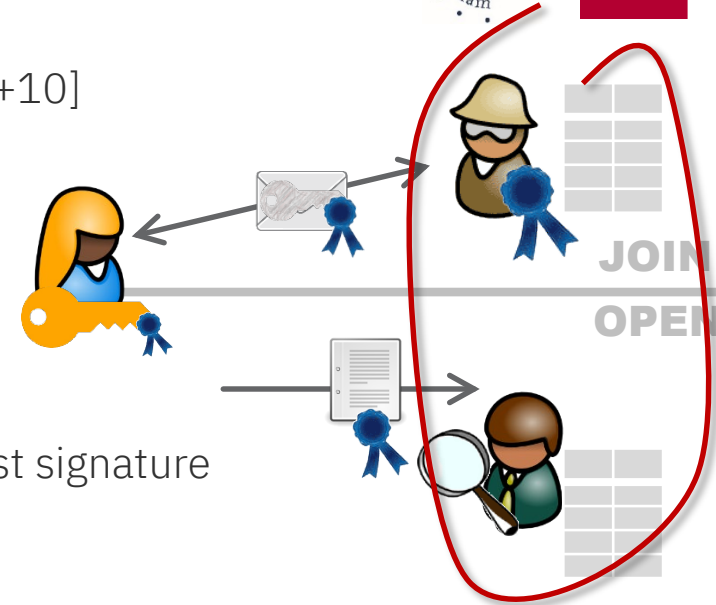


- Advantages:
  - Simple and generic design
  - Easy to separate Issuer & Opener
- Disadvantages:
  - Large signatures
  - ...significant part for opening that is hardly used!
- Problem when short signatures are needed
  - e.g., V2V communication (300 byte per message, 10 sigs per vehicle/sec)
  - Opening in case of accident/dispute

# Group Signatures | Short Signatures w/o Encryption

“Get Shorty via Group Signatures without Encryption” [BCN+10]

- Join creates user-specific opening secret  $\tau_i$  at Issuer
- Group signature  $\sigma$  does not contain encryption of  $upk$  but allows for  $Test(\tau_i, \sigma) = 0/1$
- To open a certain signature  $\sigma$ :  
Opener iterates through all opening secrets & test against signature
- Advantage: short signatures
- Disadvantage:
  - Opening gets more expensive – scales linearly in #users (feature or bug?)
  - Issuer = Opener



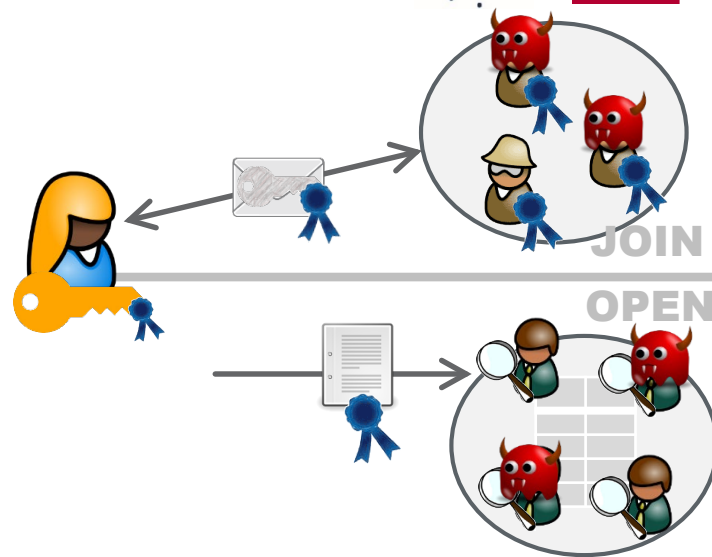
Inherently weaker security guarantees!

	Anonymity	Traceability
Issuer	Honest	Honest
Opener	Honest	Honest

# Group Signatures | Threshold Short Signatures

[CDL+20]: GetShorty signatures with separate threshold issuance and signing

- $(t_I, n_I)$  – Issuance and  $(t_O, n_O)$  – Opening
- Signature size independent of #issuers/openers
- Uses PS-signatures with threshold issuance & verifiable threshold encryption of user secret under Opener keys during Join-protocol
- Cocks–Pinch pairing curve with ~128-bit security: signature length = 232 Bytes



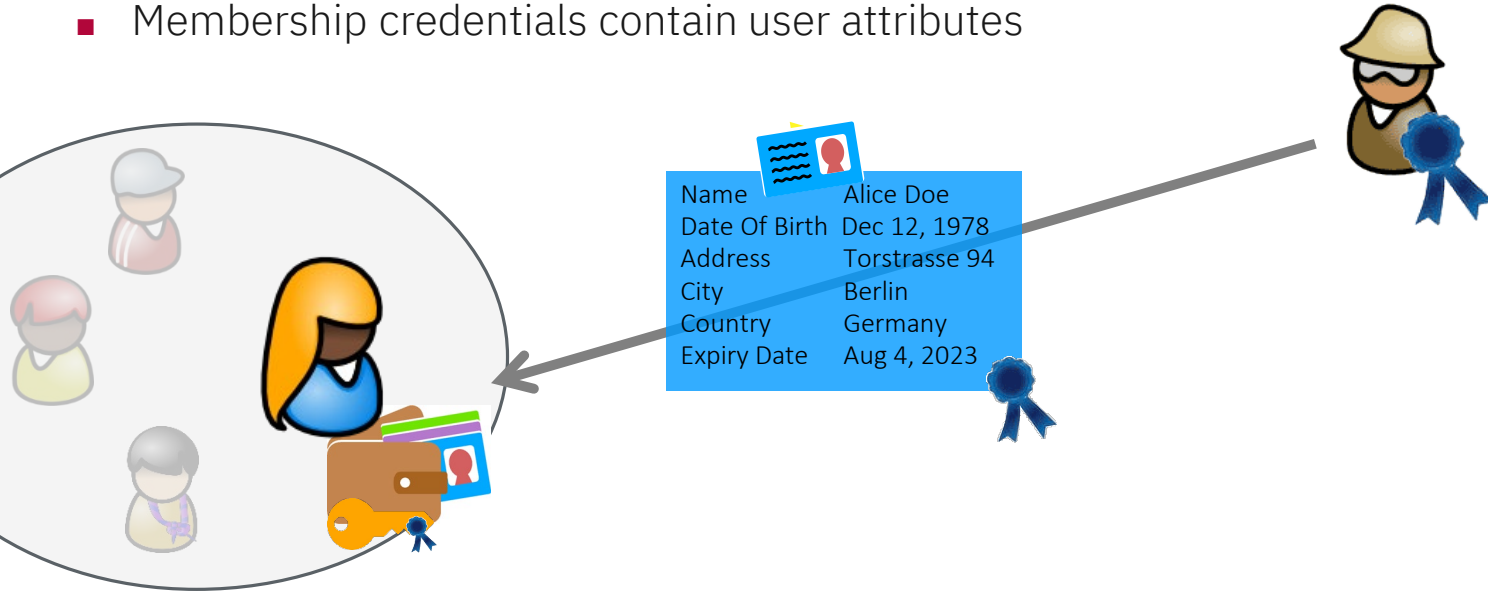
\* Still not all though

	Anonymity	Traceability
IssuerS	Corrupt*	$> t_I$ Honest
OpenerS	$> t_O$ Honest	Corrupt*



# Adding more Context & Pseudonyms: Anonymous Credentials

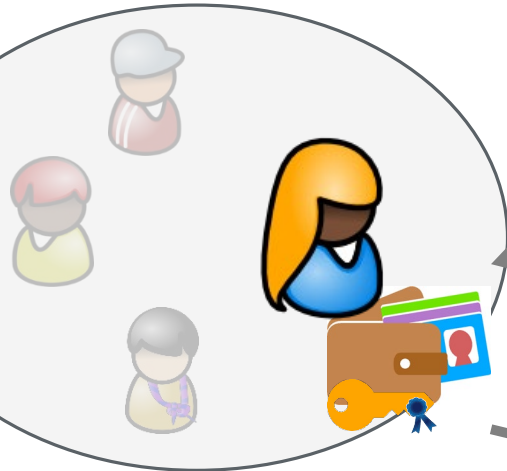
- Membership credentials contain user attributes



# Adding more Context & Pseudonyms: Anonymous Credentials

- Membership credentials contain user attributes

- User can **selectively disclose** each attribute
- User can prove **predicates over the attributes**, e.g., “I’m over 18”
- **Multi-show unlinkability** (between original & derived credentials)



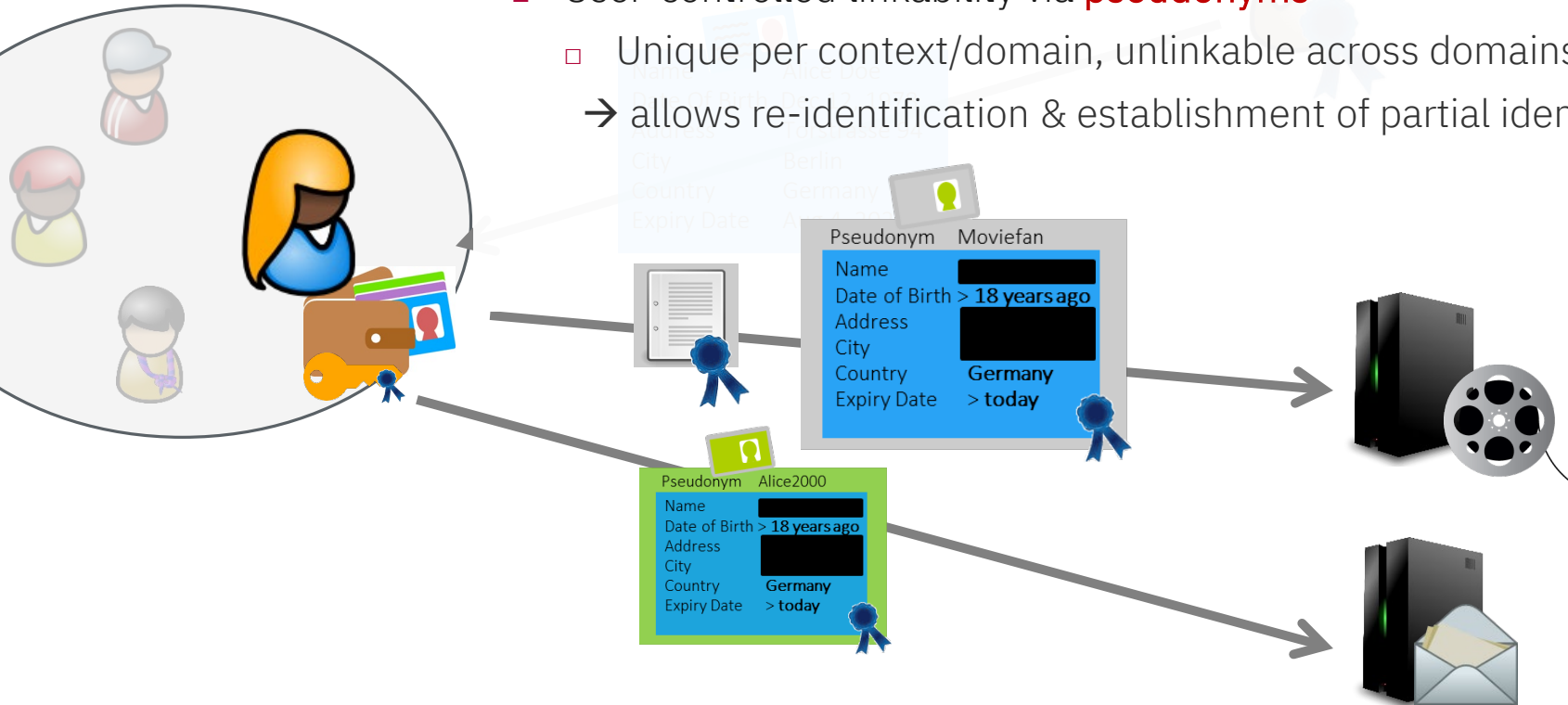
Name	Alice Doe
Date Of Birth	Dec 12, 1978
Address	Forsstraße 3
City	Berlin
Country	Germany
Expiry Date	Aug 4, 2023

group signature wrt attribute-based membership-credential

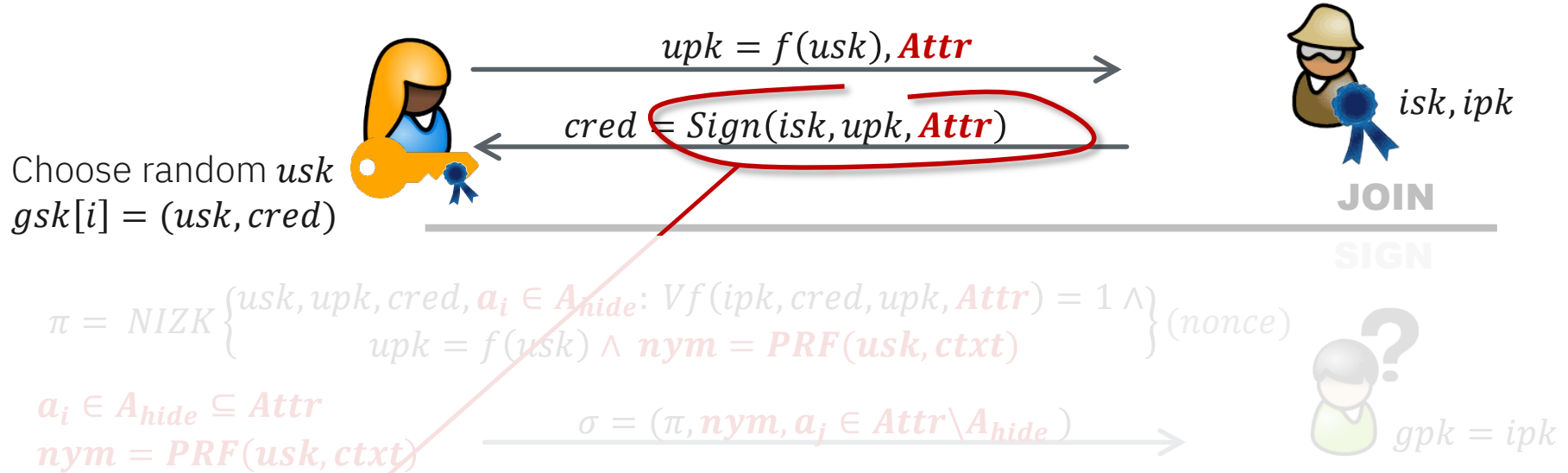
Name	[REDACTED]
Date of Birth	> 18 years ago
Address	[REDACTED]
City	[REDACTED]
Country	Germany
Expiry Date	> today



- Different/additional approach to privacy vs. accountability:
  - User-controlled linkability via **pseudonyms**
    - Unique per context/domain, unlinkable across domains
    - allows re-identification & establishment of partial identities



# Anonymous Credentials | Constructions



Same core building-block as in group signatures: CL/BBS/PS-signature

# Group Signatures vs. Anonymous Credentials

- Anonymous credentials  
= pseudonymous group signatures with attributes used for authentication?

	Group Signatures	Anonymous Credentials
Opener	Light Green	Light Green
Pseudonyms	Light Green	Dark Green
Attributes	Light Green	Dark Green

.. Well, it's a blurry line

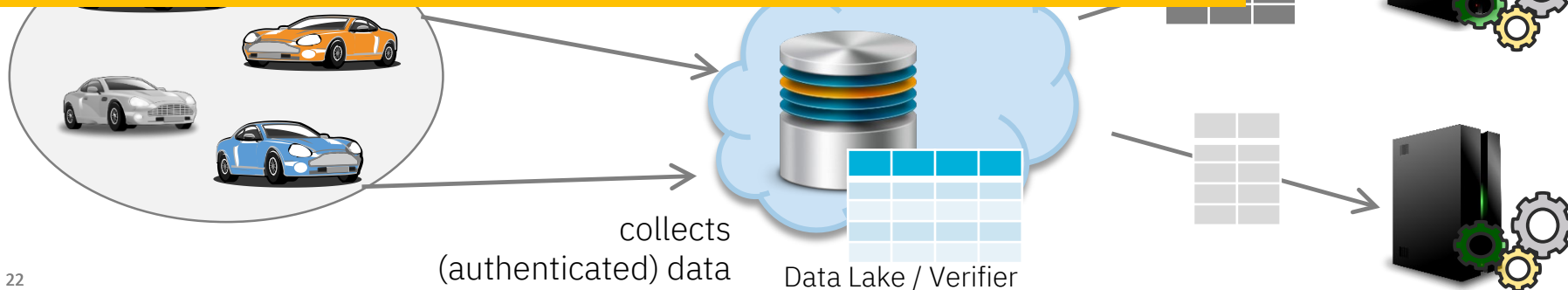
- Main differentiator: opener and/or pseudonyms  
→ steers privacy & determines unforgeability

Is that enough to balance privacy and utility/accountability?

# Authenticated & Privacy-Preserving Data Collection

- Setting:
  - Large data collections (data lake), from various sources (vehicles, sensors, IoT devices, ...)  
Data should be authenticated
  - Small subsets used for analytics  
Correlation among data items is important – but also privacy risk  
Data usage often not clear at time of collection

How can we use privacy-enhancing authentication to preserve utility yet have optimal privacy?

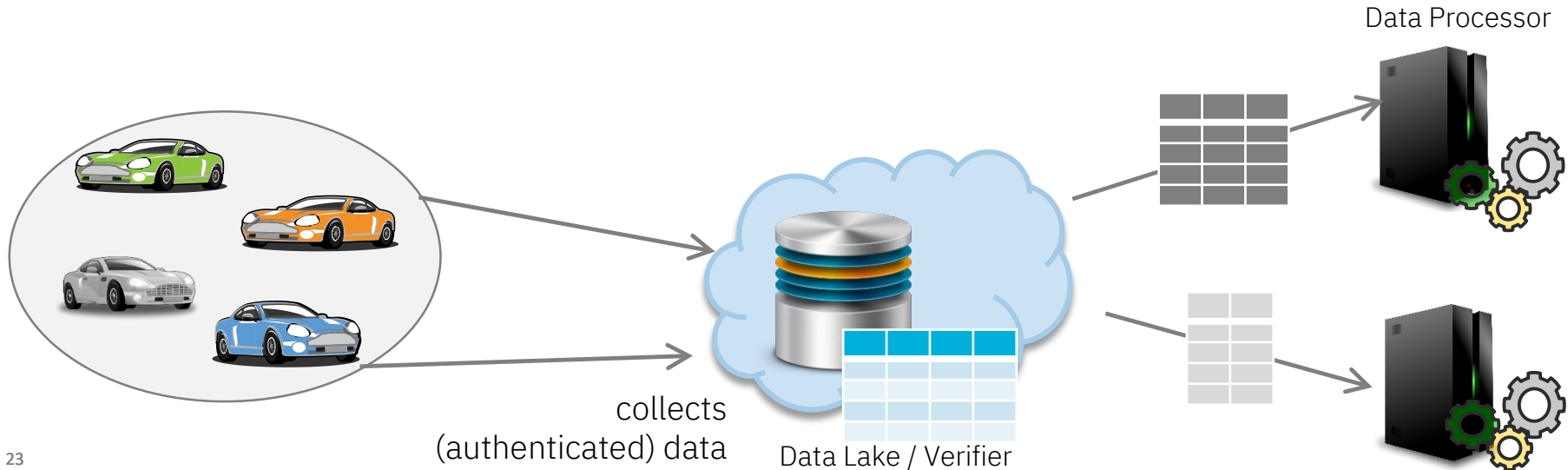


## Opening:

- Full privacy at data collection
  - No privacy when data is used
- very privacy-invasive and inefficient (either signatures or opening)

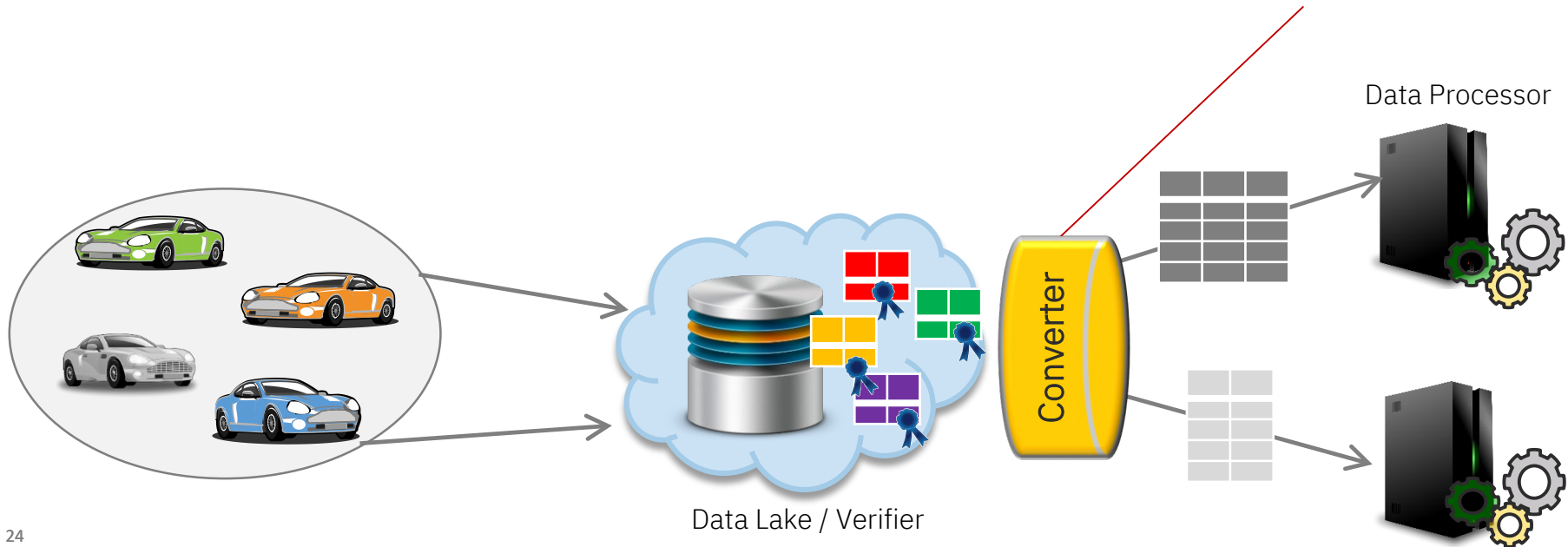
## Pseudonyms:

- Decision about linkability must be done at the moment the data is disclosed
  - No option to selectively correlate data later on
- too inflexible, bad tradeoff between privacy & utility



# Group Signatures with Selective Linkability [GL'19, DL'21]

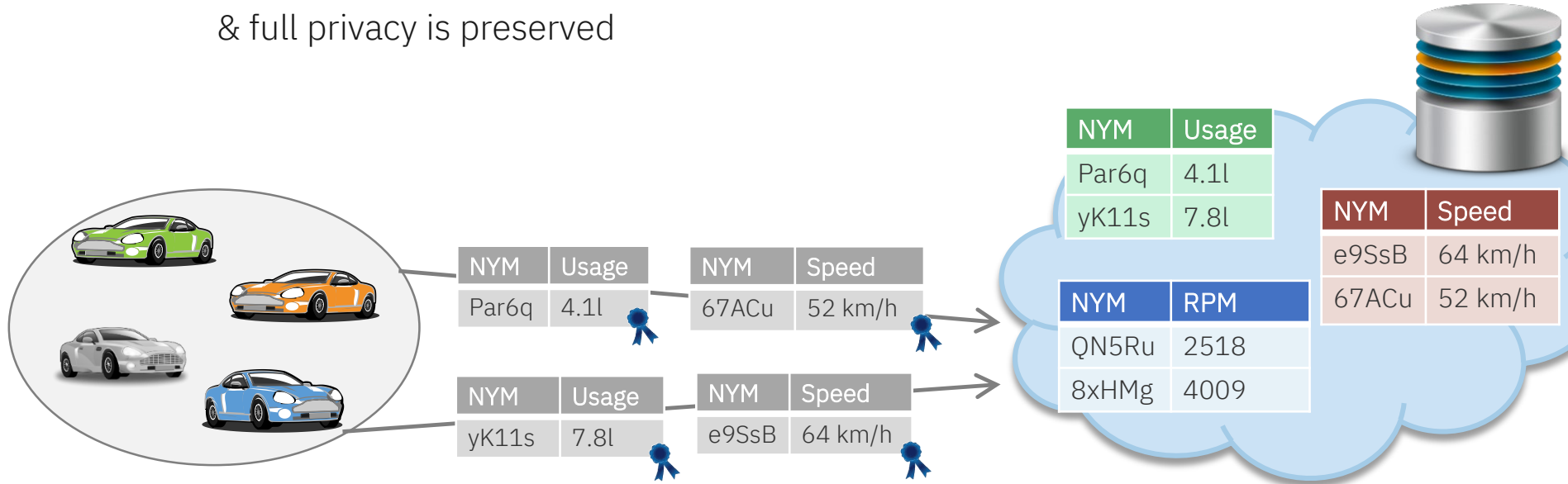
- Extends group signatures to allow for selective linkability after the data is collected
  - Data is (fully) unlinkable and anonymous when its collected
  - Selective subsets can be correlated in a consistent manner later on
  - Linkability is created either through user or a dedicated entity → the **converter**



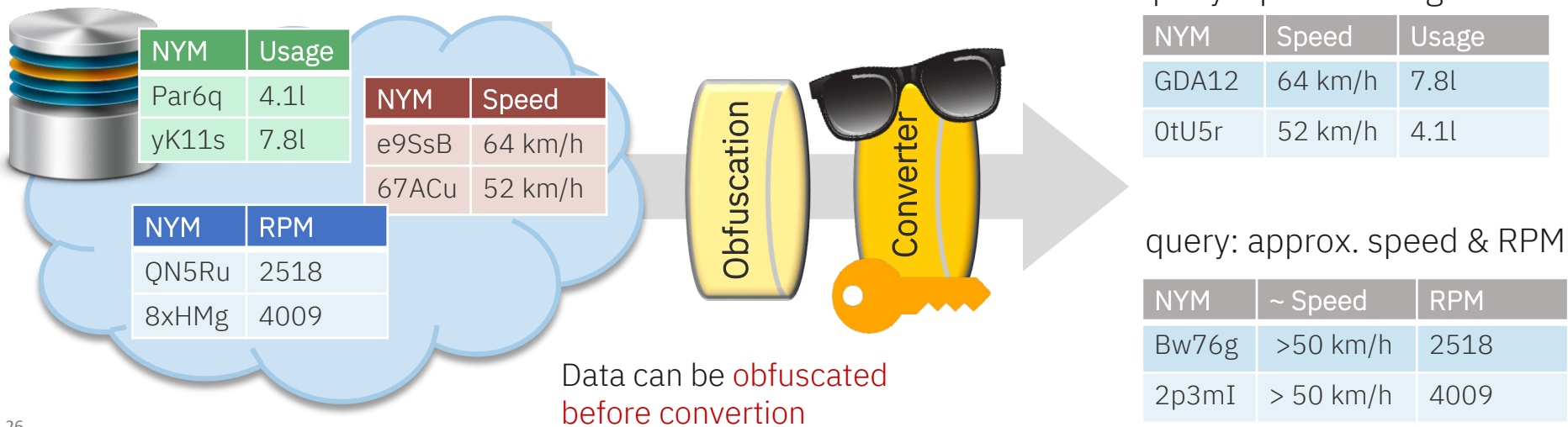


# Group Signatures with Selective Linkability | Sign

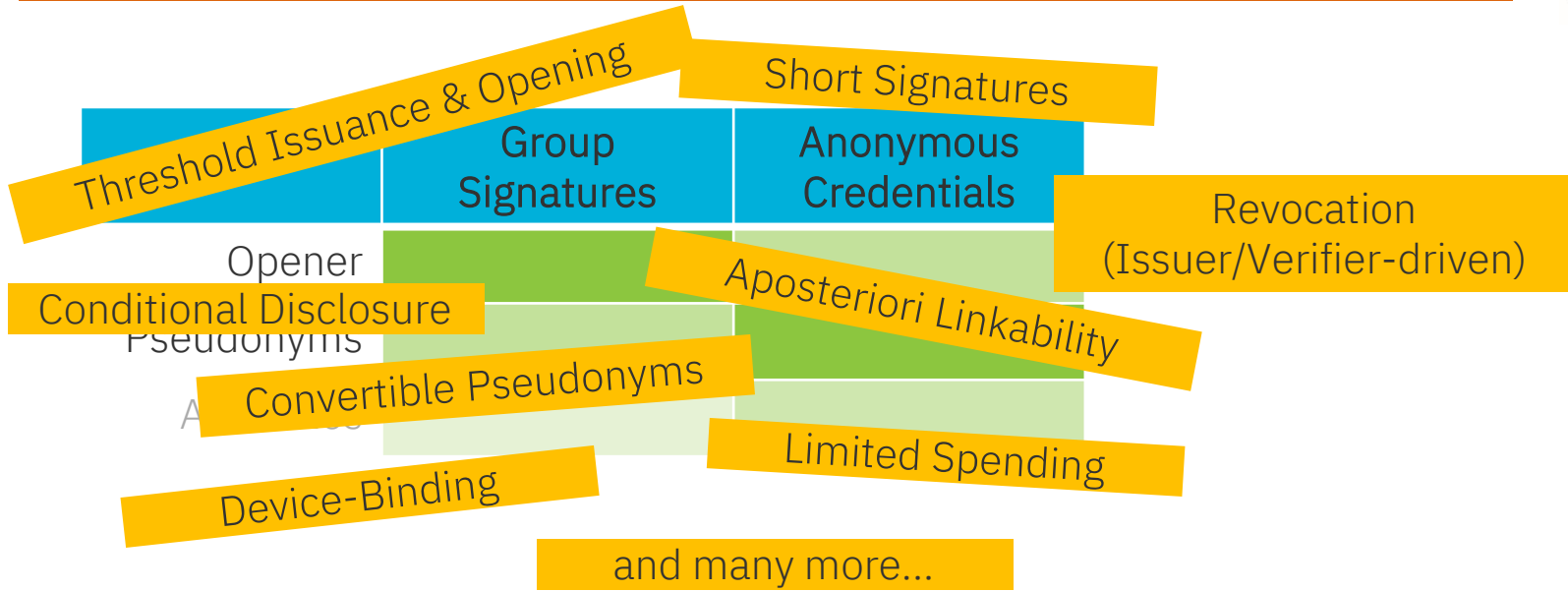
- Group signatures with fresh pseudonyms for every message
  - Data can be collected in unlinkable, authenticated snippets
- Data Lake is assured that only legitimate data gets uploaded & full privacy is preserved



- Only required sub-sets of the data are made linkable w.r.t. to join-specific pseudonym
- Converter transforms pseudonyms into consistent representation
  - **Obliviousness**: converter learns nothing about pseudonyms / messages it transforms
  - **Non-transitivity**: pseudonyms from different conversion requests cannot be linked

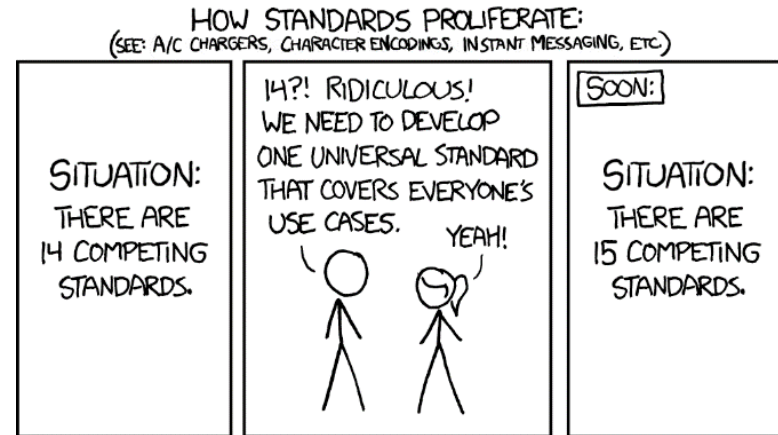


# Privacy-enhancing Authentication | Features



- Good news:
  - Lots of features needed to balance privacy and utility exist (in various forms)
  - Most can be combined easily (mostly DL-based)
- Bad news: probably not in the exact combination that is needed by your application

- Group signatures with opening are simple enough to be abstracted as generic building block
  - Simple APIs and (relatively) simple security properties
  - But rather limited functionality
- Better: privacy-enhancing authentication with pseudonyms, attributes, (revocation), ...  
But that makes the cryptographic primitive & security properties much more complex
  - Too complex to be captured by a single API / security model
  - Generic approach vs. efficiency
  - When used in a protocol: (re)build tailored scheme & redo security analysis
    - ... team up with cryptographers ☺



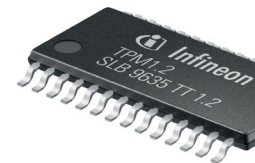
# Privacy-Enhancing Authentication in the Real-World?



## ■ Hardware-based Anonymous Attestation

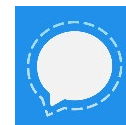
- Direct Anonymous Attestation (DAA) in 500 million TPM chips
- Intel's SGX: Enhanced Privacy ID protocol (EPID)

... include protocols – but are they actually used anywhere?



## ■ Related, simpler concepts for „symmetric setting“, i.e., Issuer = Verifier

- Keyed-Verified Anonymous Credentials [CMZ14'] from algebraic MACs  
Used (?) to manage group membership in private Signal groups [CPZ'20]



- Anonymous Tokens – Single use only (~blind MAC), e.g., Privacy Pass [DGST'18]  
Used by Cloudflare for anonymous IP Reputation



# What about User Identification?

- New attention through **Self-Sovereign Identity** (SSI) that uses decentralized identifier (DID), and verifiable credentials

Mostly conventional cryptography, but supports also privacy-enhancing credentials. e.g. Hyperledger Indy

- But... end users are a challenging target:

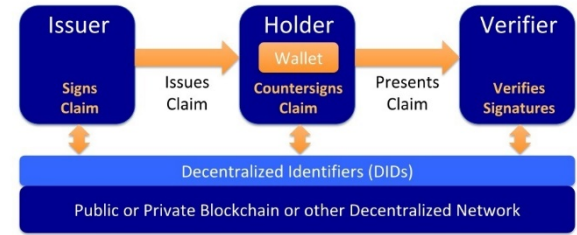
Anonymous credentials try to solve 2 problems at once: privacy & strong authentication

- Relies on users to manage cryptographic key material & certificates
- Usability and reliability challenges, e.g., different devices, backup

- Better alternative for end-users? → privacy-friendly Single Sign-On solutions?

- Privacy-enhancing authentication more suitable for „autonomous“ devices with “privacy needs” (close to users, e.g., vehicles, IoT, sensors, ...)

DIDs enable digitally signed **verifiable claims**



- Is this the right time to push this technology? Hint: quantum computers
  - All *practical* protocols rely on DL-related assumptions
    - Conditional privacy (either via opening or pseudonyms) requires DL assumption
  - Post-quantum solutions exist (but several magnitudes slower/larger than DL-based)
    - Lattice-based [dPLS'18, EZS+'19], symmetric/hash-based [DRS'17, KKW'18, BEF'19]
  - Hybrids: only privacy is based on quantum-safe assumptions [BCK+'14, BLLS'20]
- Summary:
  - Privacy-enhancing authentication allows to balance privacy & accountability
    - Main variants: opening vs pseudonyms, many extensions & flavors exist
  - Exact shape very use-case specific – not as easy to use as other „building blocks“ (yet)
    - Is there a common core functionality that is reusable?
  - Still lots of open problems for cryptographers

**Thanks! Questions?**

