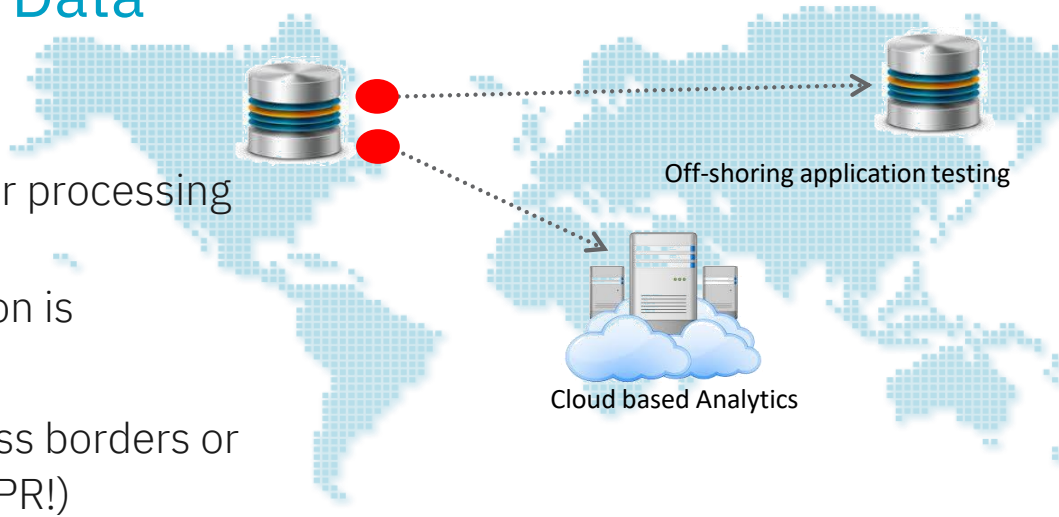
ScrambleDB: Oblivious (Chameleon) Pseudonymization-as-a-Service

Anja Lehmann

IBM Research – Zurich

IBM

Motivation | De-Sensitizing Data



- Outsourcing of data for remote storage or processing
- Shipping of sensitive/personal information is often not desired or allowed
  - Personal data cannot be moved across borders or used outside of original purpose (GDPR!)
  - Cloud environment / external service provider are not fully trusted

- How to protect sensitive data & preserve utility ?

Pseudonymization (aka Tokenization)

- Pseudonymization = replacing a primary identifier with random-looking substitute
  - Mandated e.g., by Payment Card Industry Data Security Standard (PCI DSS)

Name	Post Code	Date of Birth	Balance	IBAN
Alice Doe	64289	21.08.1978	52.650,77	CH56 0483 5084 1385 0100 0

- Requirements: pseudonymization must be
  - Consistent, i.e., referential integrity must be preserved
  - Dynamic, i.e., data can be pseudonymized on the fly
  - Secure, i.e., infeasible to determine **uid** from **nym**

UserID	Credit
Alice	€ 8.000
Bob	€ 23.500

UserID	Credit
Alice	€ 599

Nym	Credit
xH2ban6	€ 8.000
P3b0Ws	€ 23.500

Nym	Credit
xH2ban6	€ 599

Pseudonymization | Current Solution

- Deterministic one-way functions, e.g. Hash function $H(uid) \rightarrow nym$

  - Unkeyed functions vulnerable to offline attacks:

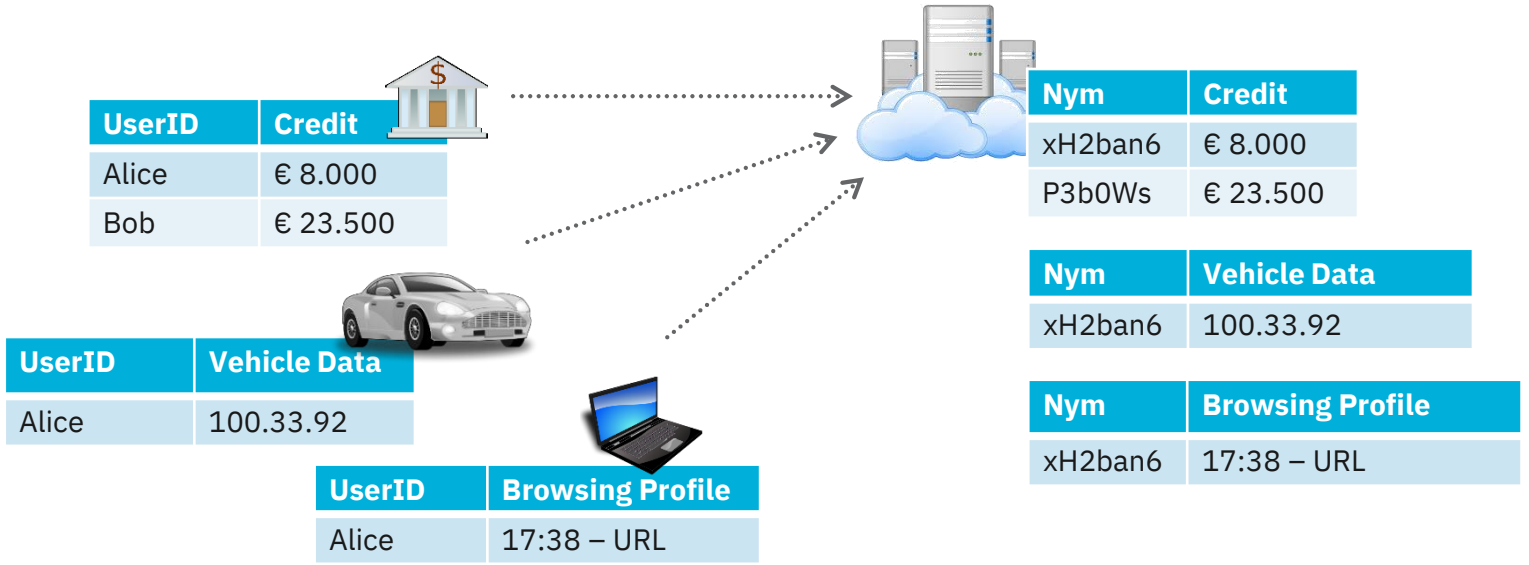
$$H(Bob) = nym\,?, H(Eve) = nym\,?, \ldots$$

  - Pseudonymization must be based on strong secret key, e.g., $PRF(key, uid) \rightarrow nym$

In theory: „Let K be a key“ In practice: Well, its not that easy ....

Challenge 1: Distributed Pseudonymization & Consistency

- Pseudonymization from multiple sources while preserving consistency

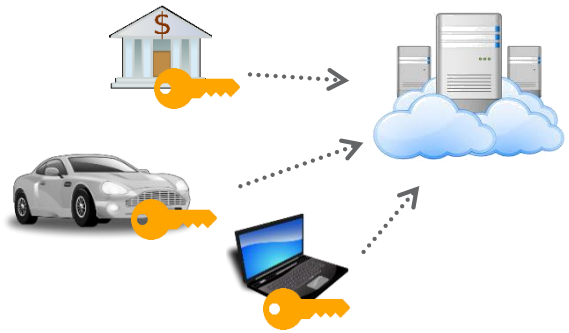


How to ensure consistent and secure pseudonymization when data is pushed by many, diverse entities ?

Distributed Pseudonymization | Existing Solutions

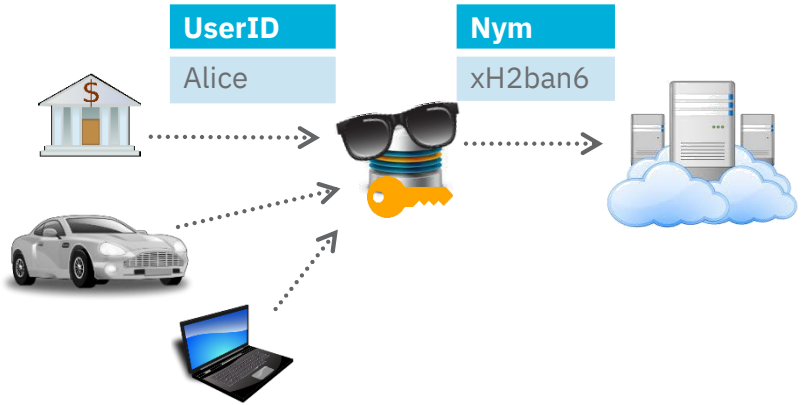
- Secure pseudonymization requires strong cryptographic keys

Key is distributed to all entities



– Replication of keys is security issue
– HSMs everywhere too expensive

Central service (TTP)

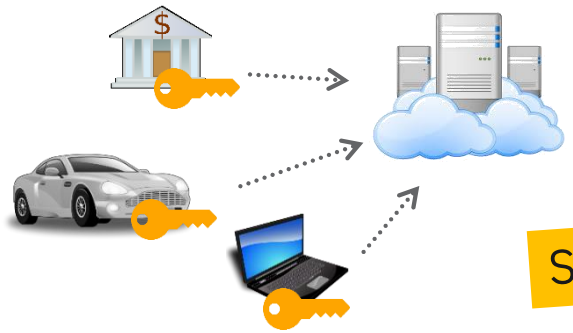


+ Simple deployment (no keys at data source)
– TTP knows the relation between UID & Nym
– TTP learns all metadata
– TTP must be fully trusted → privacy risk itself!

Distributed Pseudonymization | Existing Solutions

- Secure pseudonymization requires strong cryptographic keys

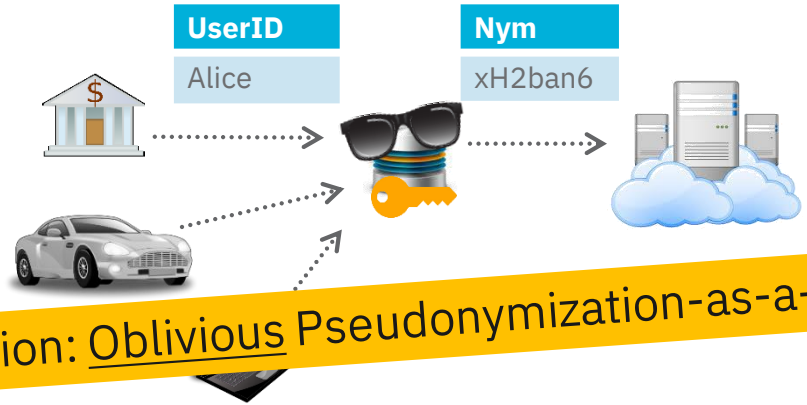
Key is distributed to all entities



– Replication of keys is security issue

– HSMs everywhere too expensive

Central service (TTP)



Solution: Oblivious Pseudonymization-as-a-Service

+ Simple deployment (no keys at data source)

– ~~TTP knows the relation between UID & Nym~~

– ~~TTP learns all metadata~~

– ~~TTP must be fully trusted → privacy risk itself!~~

Challenge 2: Privacy vs. Utility

- ~~Consistency in pseudonymization needed to preserve data utility~~ **NO!**
- **Linkability is a privacy risk** – inference attacks allow re-identification

Nym	Work
xH2ban6	IBM Research

Nym	Education
xH2ban6	PhD Cryptography

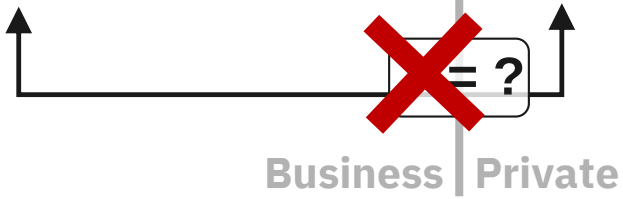
Nym	Nationality
xH2ban6	Germany

- Using context-specific pseudonyms better for privacy – but restricts usability
  - Decision which data is linkable upon pseudonymization
  - Unlinkable data cannot be linked afterwards → risk of losing too much information

Nym	Work
xH2ban6	IBM Research

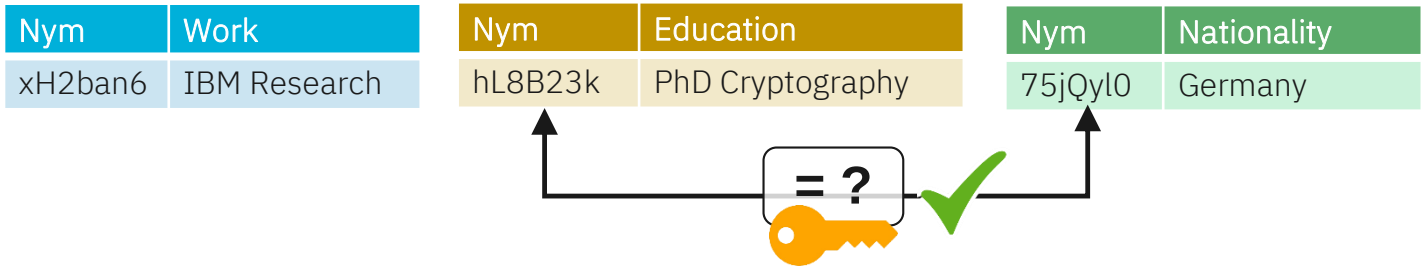
Nym	Education
xH2ban6	PhD Cryptography

Nym	Nationality
75jQyl0	Germany



Chameleon Pseudonyms

- Our approach: Chameleon = flexible pseudonyms
  - Correlating data without enforcing linkability during pseudonymization
  - Setting: large data collections (data lake), small subsets used for analytics
  - Full unlinkability when data is stored



  - Selective linkability when data is used

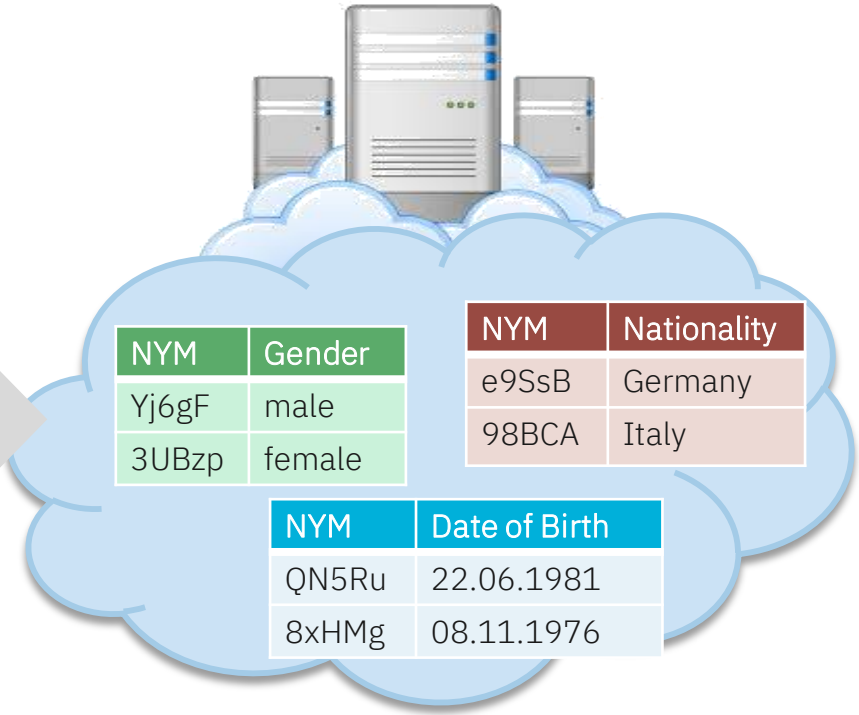
Chameleon Pseudonyms | Unlinkable Data Storage

UserID	Date of Birth	Gender	Nationality
Alice	22.06.1981	female	Germany
Bob	08.11.1976	male	Italy

NYM	Date of Birth
QN5Ru	22.06.1981

NYM	Gender
3UBzp	female

NYM	Nationality
e9SsB	Germany



NYM	Gender
Yj6gF	male
3UBzp	female

NYM	Nationality
e9SsB	Germany
98BCA	Italy

NYM	Date of Birth
QN5Ru	22.06.1981
8xHMg	08.11.1976

Converter generates **attribute-specific pseudonyms**

Pseudonym generation is done **blindly**

Data is stored in unlinkable data snippets

Chameleon Pseudonyms | Controlled Linkability

Only required sub-sets of the data are made linkable w.r.t. to join-specific pseudonym

NYM	Gender
Yj6gF	male
3UBzp	female

NYM	Nationality
e9SsB	Germany
98BCA	Italy

NYM	Date of Birth
QN5Ru	22.06.1981
8xHMg	08.11.1976

- Related pseudonyms will be converted into the same join-pseudonym („join-id“)

query: join gender & nationality

NYM	Gender
kOLc6	male
4T3gq	female

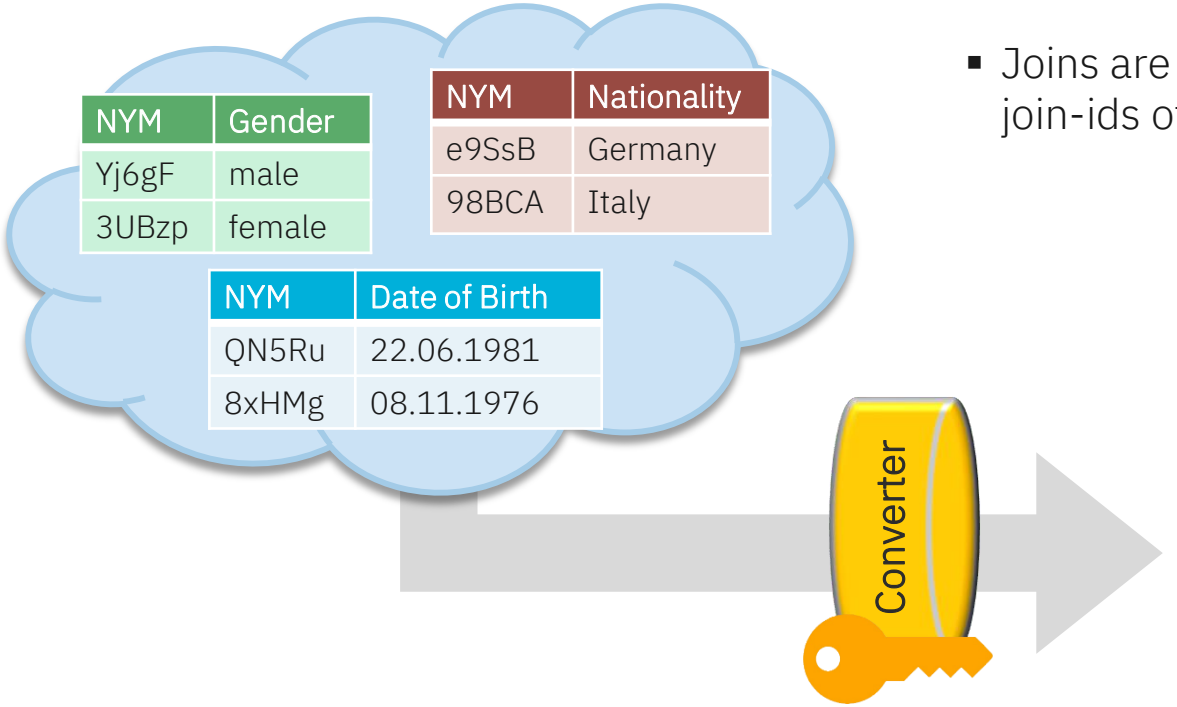
NYM	Nationality
4T3gq	Germany
kOLc6	Italy

NYM	Gender	Nationality
4T3gq	female	Germany
kOLc6	male	Italy

- Join conversion is done **blindly**

Chameleon Pseudonyms | Controlled Linkability

Only required sub-sets of the data are made linkable w.r.t. to join-specific pseudonym



- Joins are **non-transitive**, i.e., join-ids of different queries are unlinkable

query: age & EU nationality

NYM	Date of Birth	Nationality
GDA12	08.11.1976	Italy
0tU5r	22.06.1981	Germany

query: gender & nationality

NYM	Gender	Nationality
4T3gq	female	Germany
kOLc6	male	Italy

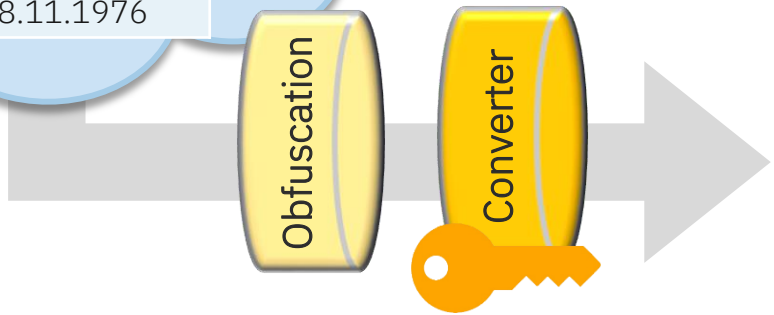
Chameleon Pseudonyms | Controlled Linkability

Only required sub-sets of the data are made linkable w.r.t. to join-specific pseudonym

NYM	Gender
Yj6gF	male
3UBzp	female

NYM	Nationality
e9SsB	Germany
98BCA	Italy

NYM	Date of Birth
QN5Ru	22.06.1981
8xHMg	08.11.1976



- Data can be **obfuscated before join** (outside of this work, though)

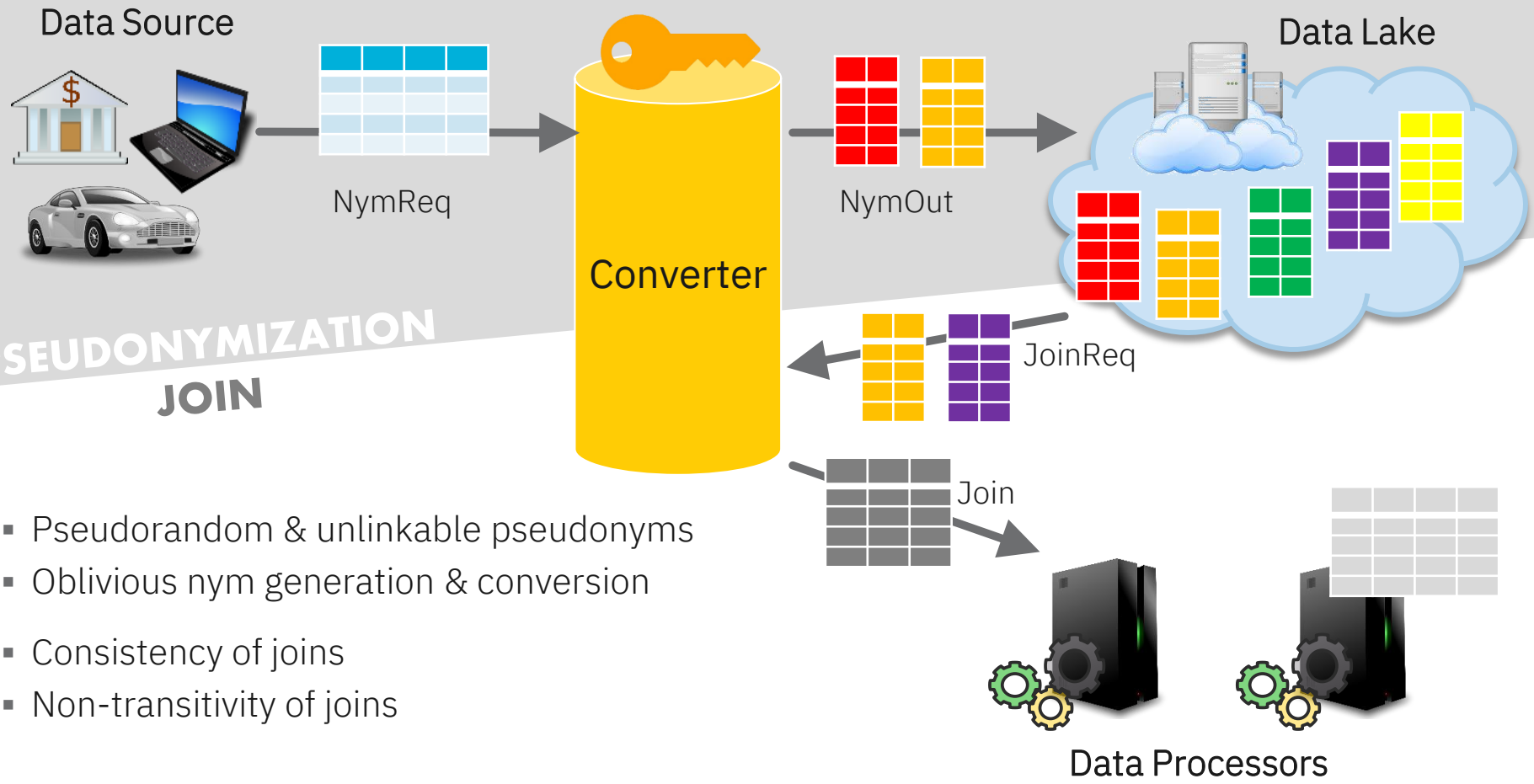
query: age & EU nationality

NYM	Date of Birth	Nationality
GDA12	**.**.1976	EU
0tU5r	**.**.1981	EU

query: gender & nationality

NYM	Gender	Nationality
4T3gq	female	Germany
kOLc6	male	Italy

Chameleon Pseudonyms | High-Level Overview



- Pseudorandom & unlinkable pseudonyms
- Oblivious nym generation & conversion
- Consistency of joins
- Non-transitivity of joins

Chameleon Pseudonyms | Cryptographic Scheme

- Core-building block: 3-Party Convertible & Oblivious Pseudorandom Function (coPRF)
  - Convertibility (& Multi-key setting):

$$y_i = \mathrm{Eval}(k_i, x) \qquad y_j = \mathrm{Convert}(k_i, k_j, y_i) \qquad s.t.\ y_j = \mathrm{Eval}(k_j, x)$$

$$y_i = H(x)^{k_i} \qquad y_j = y_i^{k_j / k_i} = H(x)^{k_j}$$

  - Oblivious 3-party evaluation & conversion

$$\bar{x} = \mathrm{Blind}(bpk, x) \qquad \bar{y_i} = \mathrm{Eval}(k_i, \bar{x}) \qquad y_i = \mathrm{Unblind}(bsk, \bar{y_i}) \qquad s.t.\ y_i = \mathrm{Eval}(k_i, x)$$

$$\bar{y_i'} = \mathrm{Blind}(bpk', y_i) \qquad \bar{y_j} = \mathrm{Convert}(k_i, k_j, \bar{y_i'}) \qquad y_j = \mathrm{Unblind}(bsk', \bar{y_j}) \qquad s.t.\ y_j = \mathrm{Eval}(k_j, x)$$

Blind = ElGamal.Encrypt

Unblind = ElGamal.Decrypt

Eval / Convert via homomorphic & re-randomization property of ElGamal

Chameleon Pseudonyms | Cryptographic Scheme

- Core-building block: 3-Party Convertible & Oblivious Pseudorandom Function (coPRF)
  - Convertibility (& Multi-key setting):

$y_i = \mathrm{Eval}(k_i, x)$ $\quad y_j = \mathrm{Convert}(k_i, k_j, y_i)$ $\quad s.t.\ y_j = \mathrm{Eval}(k_j, x)$

$y_i = H($

Pseudonym: PRF-value under attribute-specific key

Join: Pseudonym converted towards join-specific key

  - Oblivious 3-party eval

$\bar{x} = \mathrm{Blind}(bpk, x)$ $\quad \bar{y}_i = \mathrm{Eval}(k_i, \bar{x})$ $\quad y_i = \mathrm{Unblind}(bsk, \bar{y}_i)$ $\quad s.t.\ y_i = \mathrm{Eval}(k_i, x)$

$\bar{y}_i' = \mathrm{Blind}(bpk', y_i)$ $\quad \bar{y}_j = \mathrm{Convert}(k_i, k_j, \bar{y}_i')$ $\quad y_j = \mathrm{Unblind}(bsk', \bar{y}_j)$ $\quad s.t.\ y_j = \mathrm{Eval}(k_j, x)$

Blind = ElGamal.Encrypt

Unblind = ElGamal.Decrypt

Eval / Convert via homomorphic & re-randomization property of ElGamal

Summary

Data pseudonymization – crucial part for de-sensitization of data

- Schemes must use strong cryptographic keys → challenges for key management
  - **Oblivious** pseudonymization-as-a-service
- Linkability crucial for utility, but also weakens privacy
  - Chameleon pseudonyms: **unlinkability as default**, selective linkability when needed
- Strong security: our scheme is provably secure in Universal Composability Model
- BUT: Pseudonymization alone is not sufficient!
  - Needs to be complemented with proper attribute protection/obfuscation

Thanks! Questions?

anj@zurich.ibm.com