Key Management in Distributed Online Social Networks

WoWMoM 2011 – D-SPAN, Lucca

TECHNISCHE UNIVERSITÄT DARMSTADT

Felix Günther, Mark Manulis, Thorsten Strufe
TU Darmstadt & CASED

Users and the Web

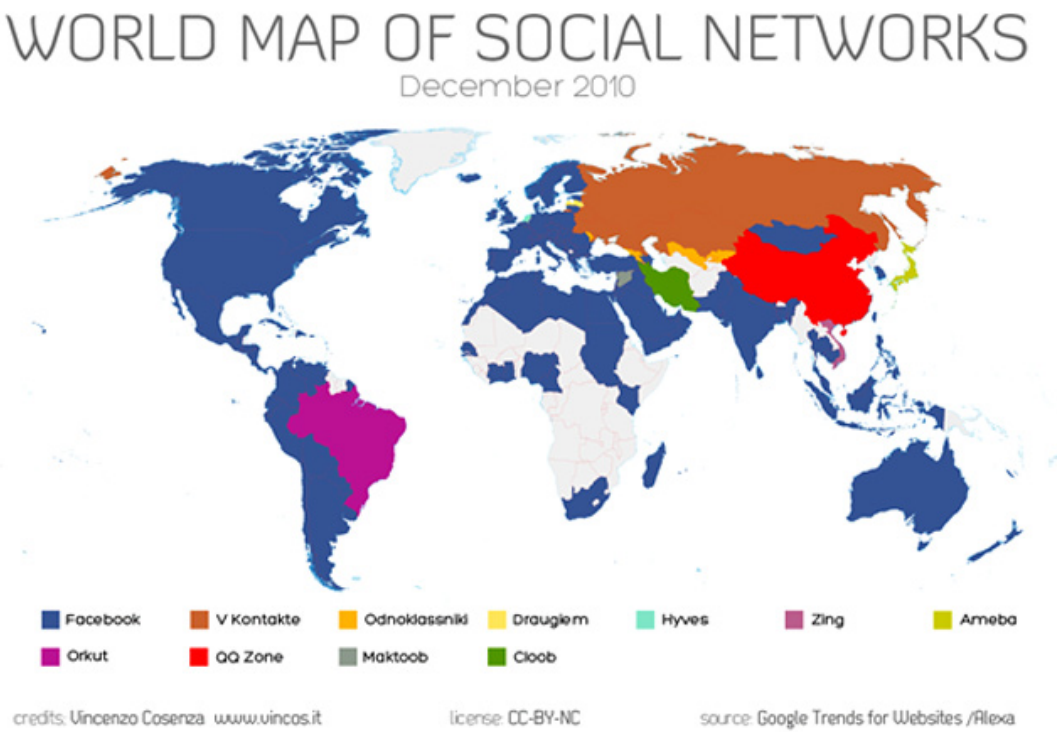
TECHNISCHE UNIVERSITÄT DARMSTADT

Modern Web is dominated by social interaction, networking, online communities.
Services are offered by users for users – Users are at the heart of the Web.

Social Networks as Global Players

TECHNISCHE UNIVERSITÄT DARMSTADT

Social networking on the Web enjoys popularity all over the world.

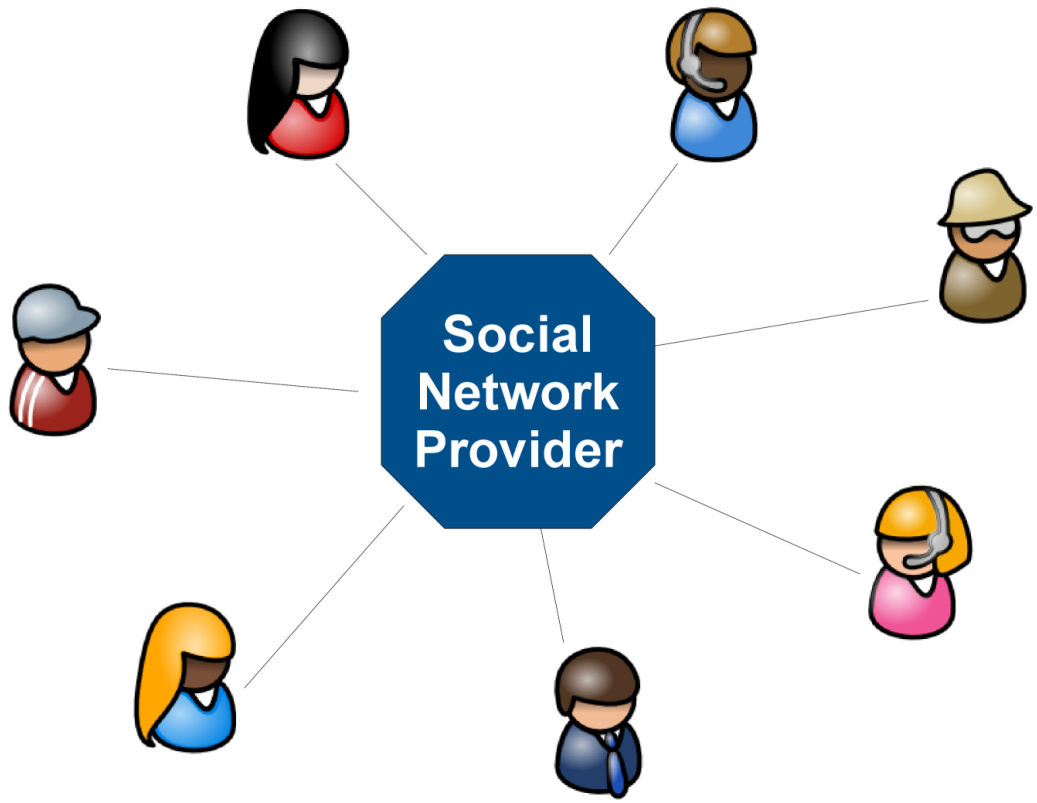


Facebook alone has over 500 Mio. active users, 50% logging on every day.

Current Social Networks are Centralized

TECHNISCHE UNIVERSITÄT DARMSTADT

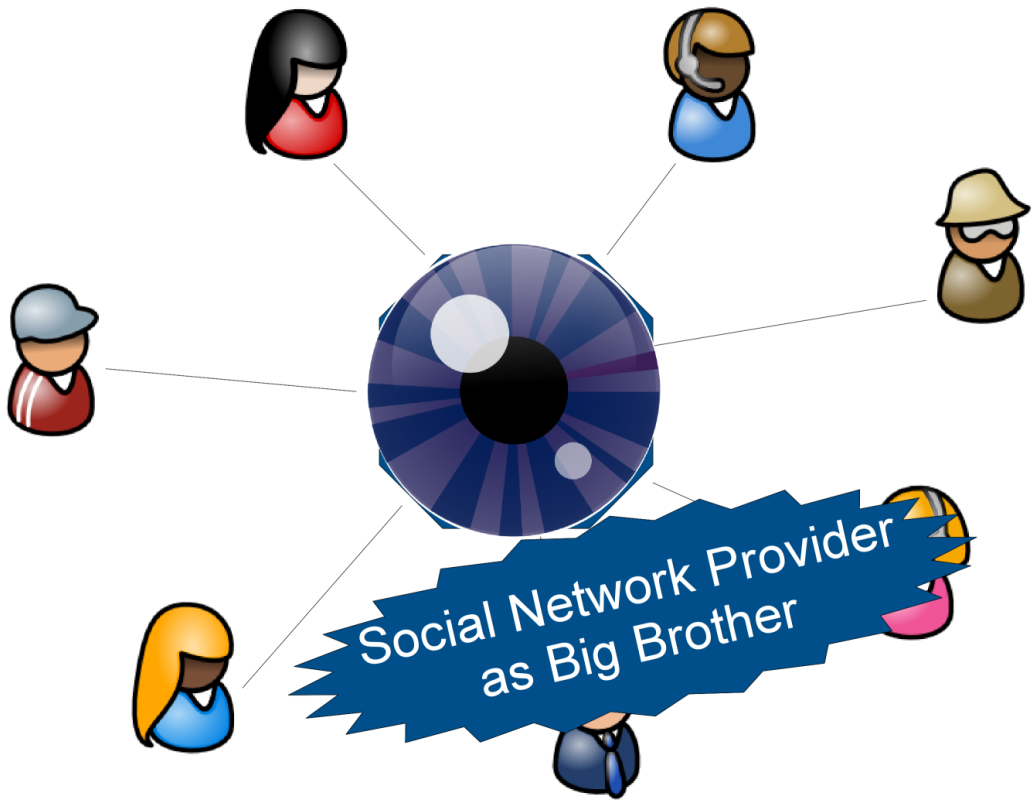
All big online social networks today are centralized.



Current Social Networks are Centralized

TECHNISCHE UNIVERSITÄT DARMSTADT

All big online social networks today are centralized.
Many trust has to be put into the Provider, which can act as a Big Brother.



Decentralized Approach and Requirements

TECHNISCHE
UNIVERSITÄT
DARMSTADT

One approach to avoid a Big Brother is to decentralize storage and control. This however introduces some challenges:

Decentralized Approach and Requirements

TECHNISCHE UNIVERSITÄT DARMSTADT

One approach to avoid a Big Brother is to decentralize storage and control. This however introduces some challenges:

- availability of profiles

Decentralized Approach and Requirements

TECHNISCHE UNIVERSITÄT DARMSTADT

One approach to avoid a Big Brother is to decentralize storage and control. This however introduces some challenges:

- availability of profiles $\rightarrow$ replication on friends' systems

Decentralized Approach and Requirements

TECHNISCHE UNIVERSITÄT DARMSTADT

One approach to avoid a Big Brother is to decentralize storage and control. This however introduces some challenges:

- availability of profiles $\rightarrow$ replication on friends' systems
- privacy/access control

Decentralized Approach and Requirements

TECHNISCHE UNIVERSITÄT DARMSTADT

One approach to avoid a Big Brother is to decentralize storage and control. This however introduces some challenges:

- availability of profiles $\rightarrow$ replication on friends' systems
- privacy/access control $\rightarrow$ encryption of profile data

Decentralized Approach and Requirements

TECHNISCHE UNIVERSITÄT DARMSTADT

One approach to avoid a Big Brother is to decentralize storage and control. This however introduces some challenges:

- availability of profiles $\rightarrow$ replication on friends' systems
- privacy/access control $\rightarrow$ encryption of profile data

Qualitative Requirements for Encryption Schemes

- confidentiality: hide data from unauthorized users
- privacy: hide identities of authorized users

Decentralized Approach and Requirements

TECHNISCHE UNIVERSITÄT DARMSTADT

One approach to avoid a Big Brother is to decentralize storage and control. This however introduces some challenges:

- availability of profiles $\rightarrow$ replication on friends' systems
- privacy/access control $\rightarrow$ encryption of profile data

Qualitative Requirements for Encryption Schemes

- confidentiality: hide data from unauthorized users
- privacy: hide identities of authorized users

Quantitative Requirements for Encryption Schemes

- low storage overhead
- little interaction with authorized users
- low ressources requirements for computations

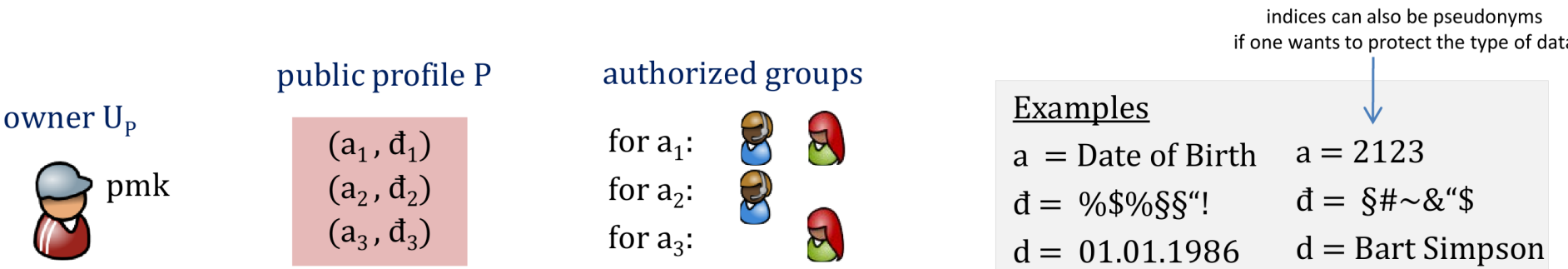
Formal Model for User Profiles and Profile Management

TECHNISCHE UNIVERSITÄT DARMSTADT

- A profile P is modeled as a set of pairs

$$P \stackrel{def}{=} \left\{(a, \bar{d}) | a \in \mathcal{I}, \bar{d} \in \{0,1\}^*\right\}$$

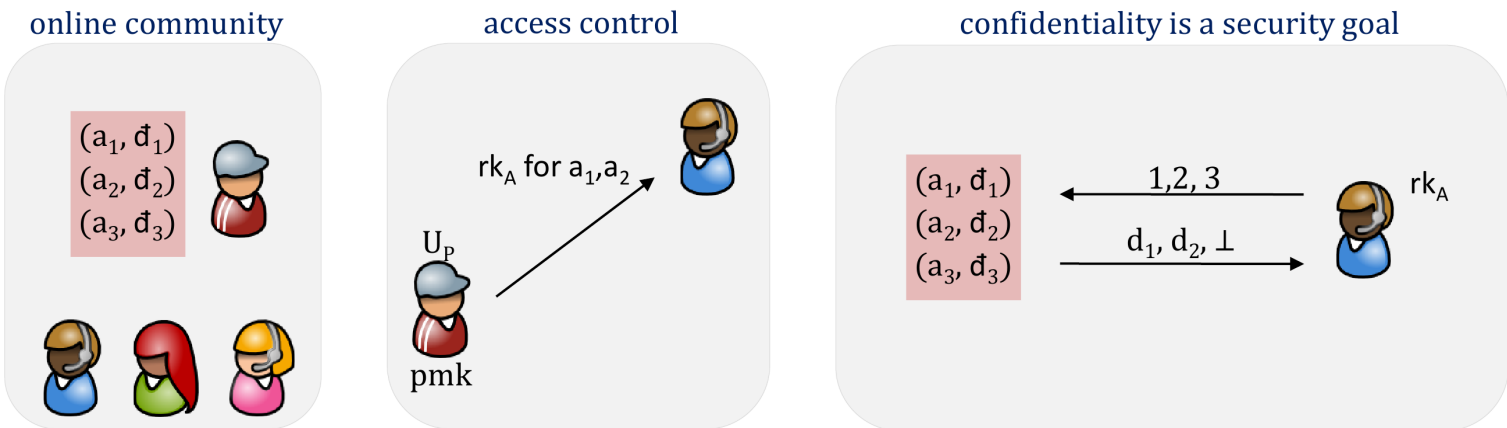
- $\mathcal{I}$ is set of unique attribute indices a; $\bar{d}$ is the corresponding value stored in P.
- P is public + authenticated by owner U_P, having profile management key pmk.
- U_P given $(a, \bar{d}) \in P$ knows attribute d and group G of authorized users.
- Profile Management Scheme offers:
  $\texttt{Init}(\kappa)$, $\texttt{Publish}(pmk, P, (a, d), G)$, $\texttt{Retrieve}(rk_U, P, a)$,
  $\texttt{Delete}(pmk, P, a)$, $\texttt{ModifyAccess}(pmk, P, a, U)$



Security Goal: Confidentiality

TECHNISCHE UNIVERSITÄT DARMSTADT

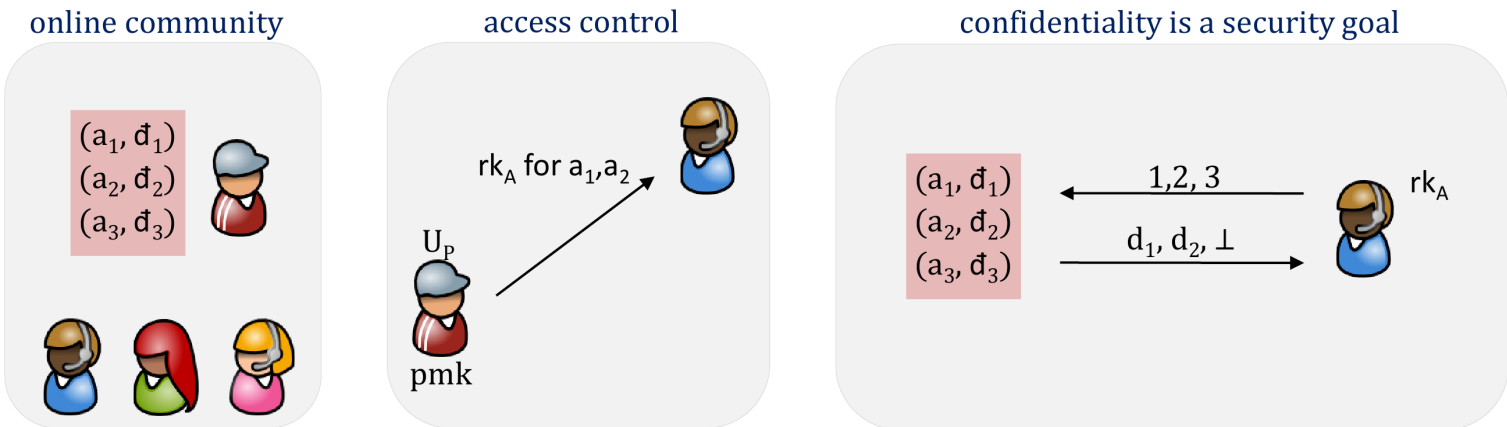
U_P publishes pairs $(a, \bar{d})$ in P and gives U retrieval key rk_U for some indices.
Confidentiality: Attributes d should remain hidden from unauthorized users.



Security Goal: Confidentiality

TECHNISCHE UNIVERSITÄT DARMSTADT

U_P publishes pairs $(a, \bar{d})$ in P and gives U retrieval key rk_U for some indices.
Confidentiality: Attributes d should remain hidden from unauthorized users.



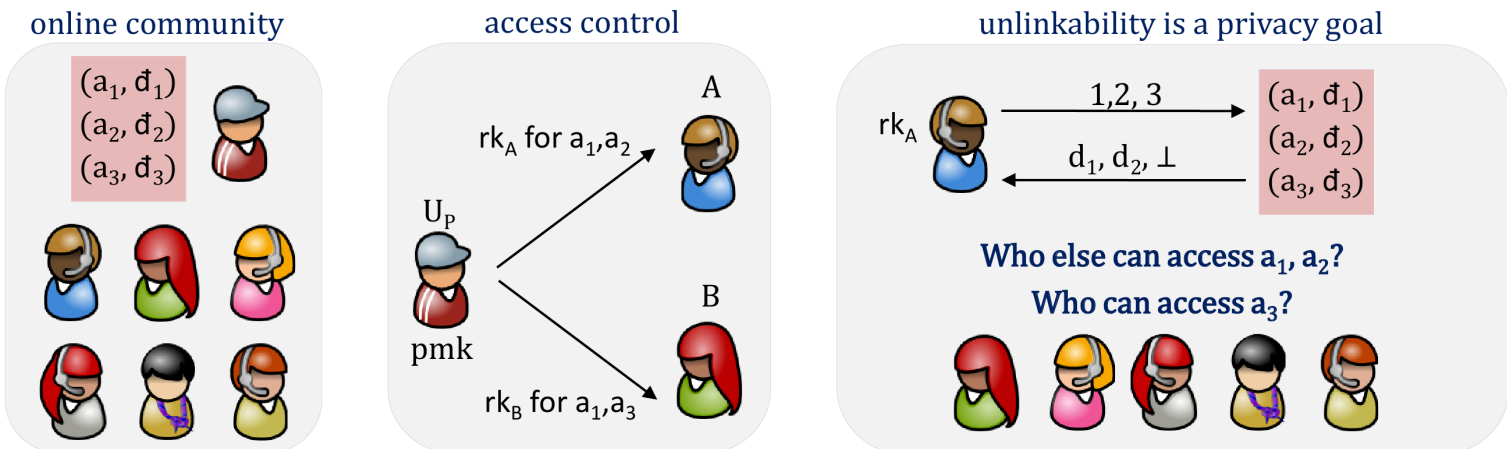
Indistinguishability approach:
$\mathcal{A}$ without access rights to $(a, \bar{d})$ should not be able to distinguish which attribute d is encrypted in $\bar{d}$.

. . . even if $\mathcal{A}$ can access other attributes in the same profile.

Privacy Goal: Unlinkability

TECHNISCHE UNIVERSITÄT DARMSTADT

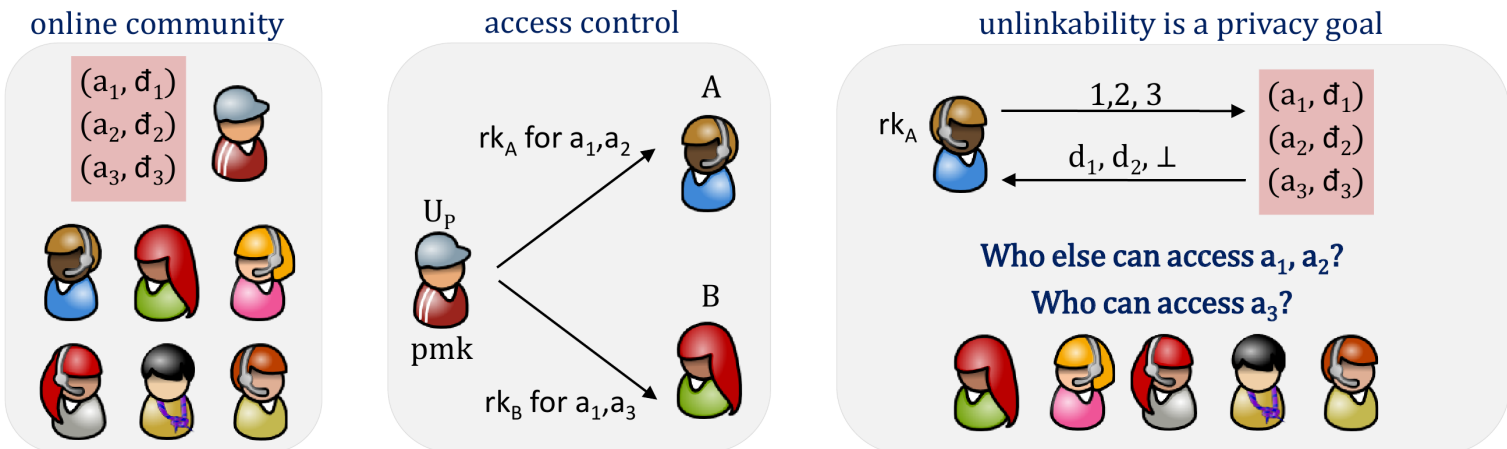
Owner U_P knows which users were granted access to which pairs (a, d) in P.
Unlinkability: Profiles should hide which users can access which attributes.



Privacy Goal: Unlinkability

TECHNISCHE UNIVERSITÄT DARMSTADT

Owner U_P knows which users were granted access to which pairs (a, d) in P.
Unlinkability: Profiles should hide which users can access which attributes.

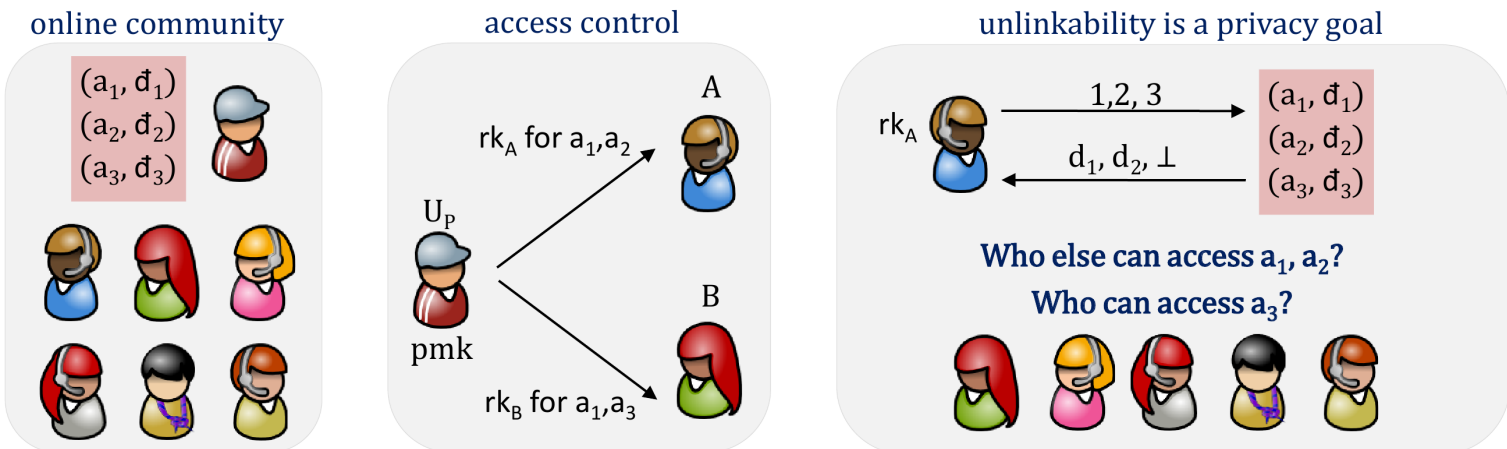


Indistinguishability approach:
$\mathcal{A}$ without access rights to $(a, \bar{d})$ should not be able to distinguish whether user A or user B was granted to access a.

Privacy Goal: Unlinkability

TECHNISCHE UNIVERSITÄT DARMSTADT

Owner U_P knows which users were granted access to which pairs (a, d) in P.
Unlinkability: Profiles should hide which users can access which attributes.



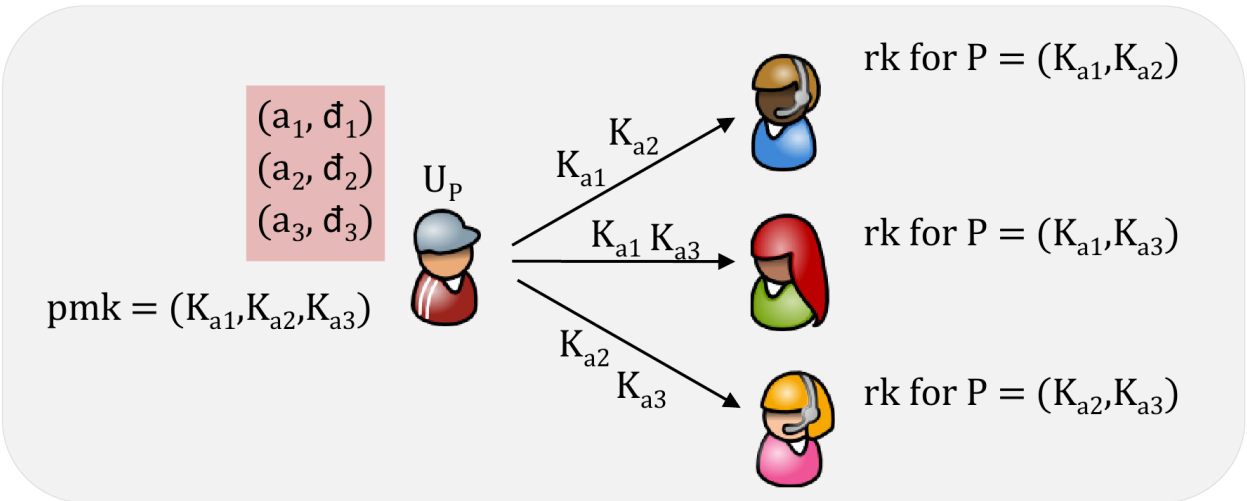
Indistinguishability approach:
$\mathcal{A}$ without access rights to $(a, \bar{d})$ should not be able to distinguish whether user A or user B was granted to access a.

- Formal model + Conf./Unlink. games in Günther et al. FC/RLCPS 2011.

Shared Key (SK) Approach

TECHNISCHE UNIVERSITÄT DARMSTADT

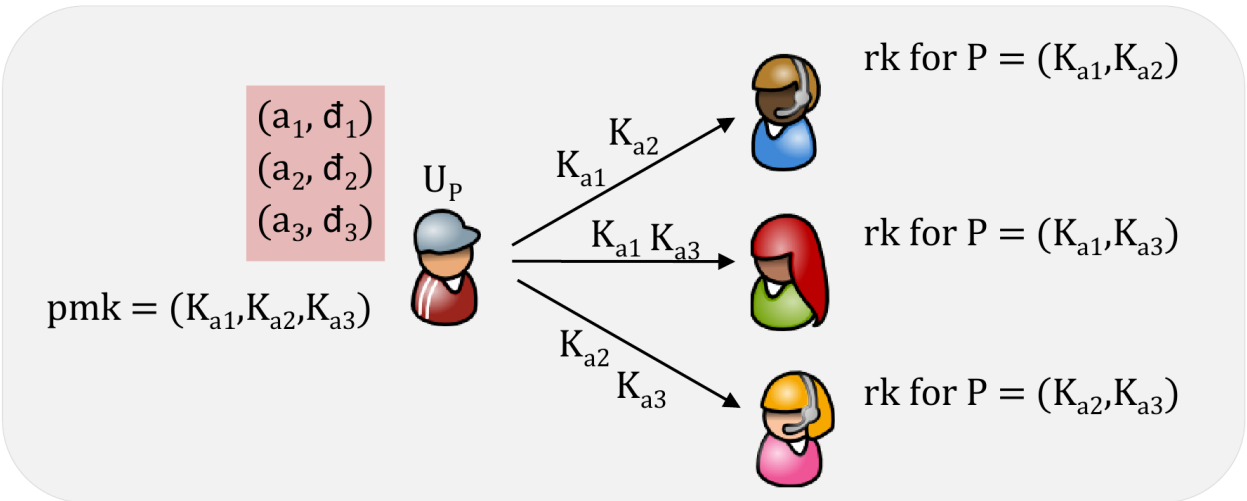
- intuitive approach: shared secret key for each attribute
- separate keys $K_a \leftarrow SE.KGen(\kappa)$ for each pair $(a, \bar{d})$: $\bar{d} = SE.Enc(K_a, d)$
- revocation: re-encryption with new K_a



Shared Key (SK) Approach

TECHNISCHE UNIVERSITÄT DARMSTADT

- intuitive approach: shared secret key for each attribute
- separate keys $K_a \leftarrow SE.KGen(\kappa)$ for each pair $(a, \bar{d})$: $\bar{d} = SE.Enc(K_a, d)$
- revocation: re-encryption with new K_a

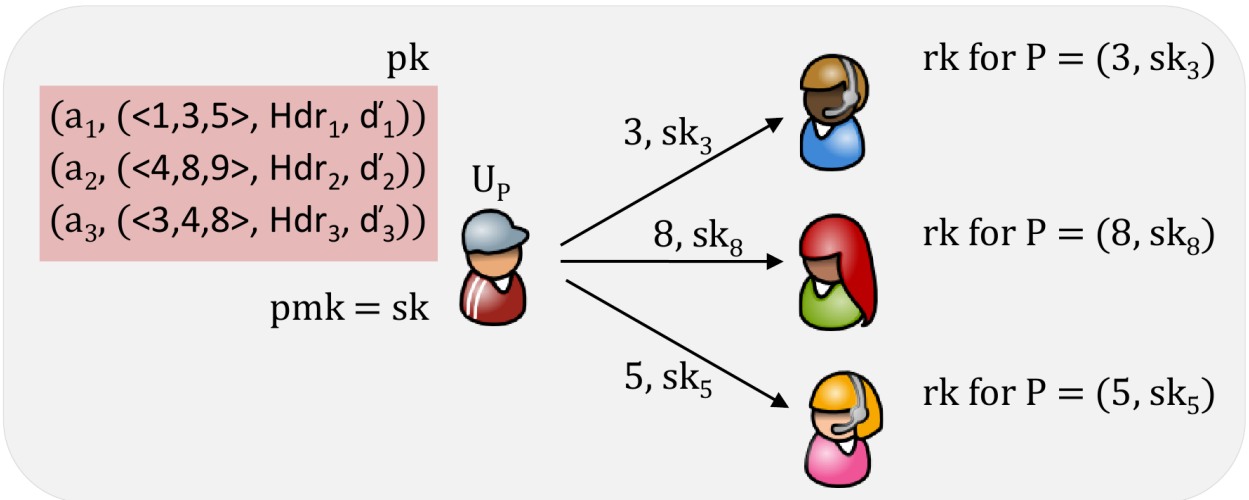


- provides confidentiality and perfect unlinkability
- each user has to store one key per attribute per profile
- two storage variants: at the users or (encrypted) in the profile
- key updates can be optimized with group key management for K_a (LKH, OFT)

Broadcast Encryption (BE) Approach

TECHNISCHE UNIVERSITÄT DARMSTADT

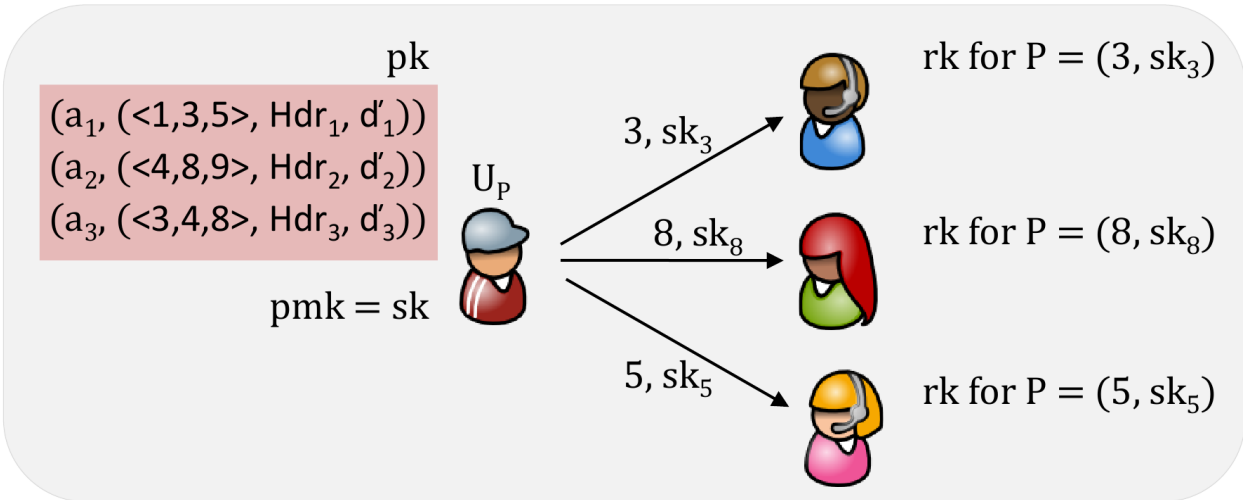
- each user manages own broadcast group using $(pk, sk) \leftarrow BE.Setup(\kappa, n)$
- each authorized user i receives a single key sk_i per profile
- for each $(a, d) : (Hdr, K_a) \leftarrow BE.Enc(S, pk)$, authorized users S, $\bar{d} = SE.Enc(K_a, d)$ and finally $\bar{d} = (Hdr, S, \bar{d})$.
- revocation: re-encryption with new (Hdr, K_a) for the modified set S



Broadcast Encryption (BE) Approach

TECHNISCHE UNIVERSITÄT DARMSTADT

- each user manages own broadcast group using $(pk, sk) \leftarrow BE.Setup(\kappa, n)$
- each authorized user i receives a single key sk_i per profile
- for each (a, d) : $(Hdr, K_a) \leftarrow BE.Enc(S, pk)$, authorized users S, $\bar{d} = SE.Enc(K_a, d)$ and finally $\bar{d} = (Hdr, S, \bar{d})$.
- revocation: re-encryption with new (Hdr, K_a) for the modified set S



- confidentiality and perfect anonymity (strictly weaker than unlinkability)
- each user has to store one key per profile

Quantitative Evaluation

TECHNISCHE UNIVERSITÄT DARMSTADT

Quantitative Evaluation was done regarding three properties:

Quantitative Evaluation

TECHNISCHE UNIVERSITÄT DARMSTADT

Quantitative Evaluation was done regarding three properties:

- storage requirements
  - at the profile owner (outside the profile)
  - in the profile
  - at the authorized users

Quantitative Evaluation

TECHNISCHE UNIVERSITÄT DARMSTADT

Quantitative Evaluation was done regarding three properties:

- storage requirements
  - at the profile owner (outside the profile)
  - in the profile
  - at the authorized users
- number of encryptions
  - on initialization
  - on user addition
  - on user removal

Quantitative Evaluation

TECHNISCHE UNIVERSITÄT DARMSTADT

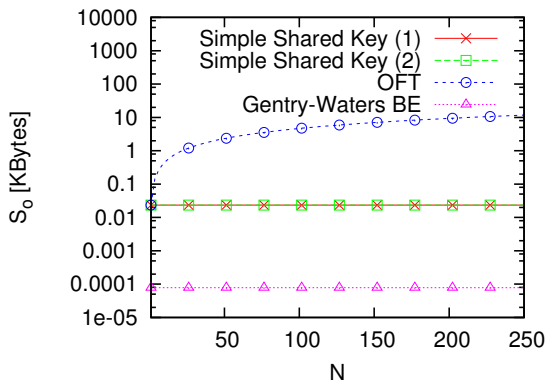
Quantitative Evaluation was done regarding three properties:

- storage requirements
  - at the profile owner (outside the profile)
  - in the profile
  - at the authorized users
- number of encryptions
  - on initialization
  - on user addition
  - on user removal
- number of messages
  - on initialization
  - on user addition
  - on user removal

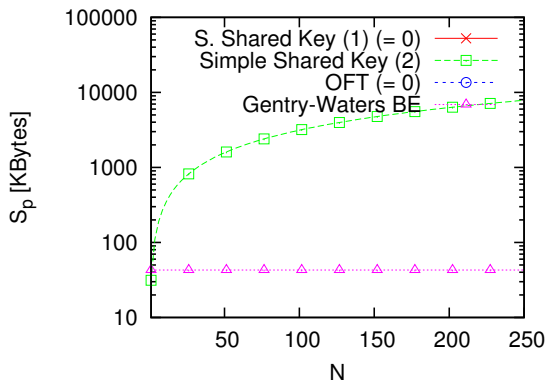
Quantitative Evaluation

TECHNISCHE UNIVERSITÄT DARMSTADT

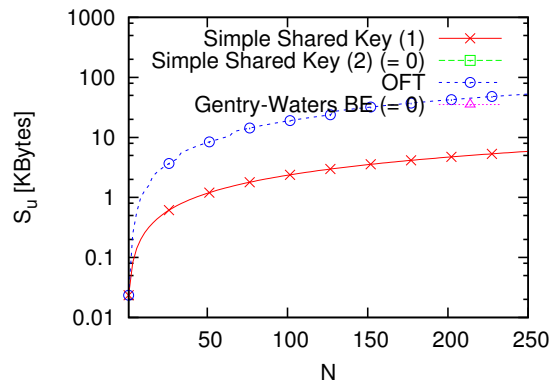
How much storage capacity is needed?



(a) at the owner



(b) in the profile



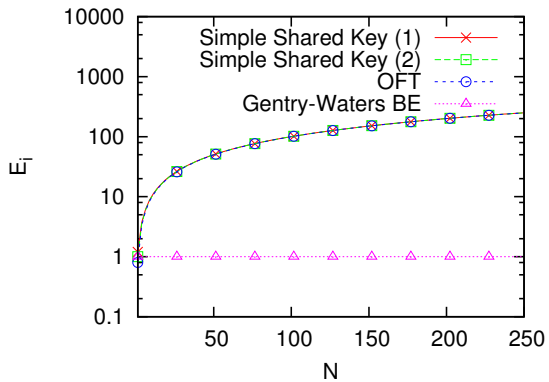
(c) at the users

- storage plots for a single attribute published for N users
- linear growth for some property for SK and OFT
- BE only needs constant storage

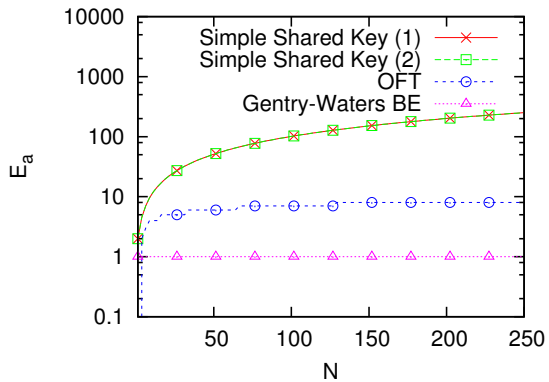
Quantitative Evaluation

TECHNISCHE UNIVERSITÄT DARMSTADT

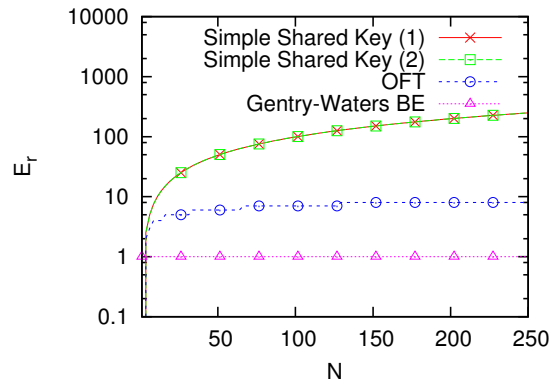
How many encryptions are needed?



(d) on initialization



(e) on user addition



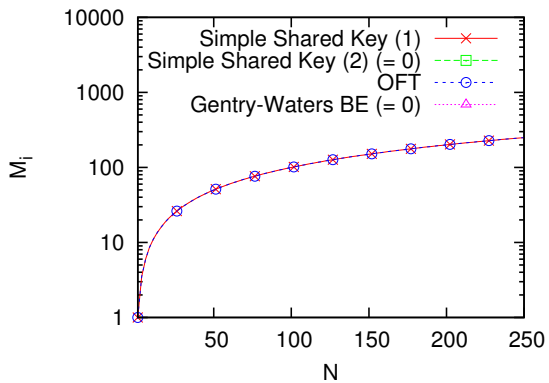
(f) on user removal

- encryptions needed for a single attribute published for *N* users
- linear or logarithmic number of encryptions for Shared Key and OFT
- only a single encryption needed for BE, however more expensive

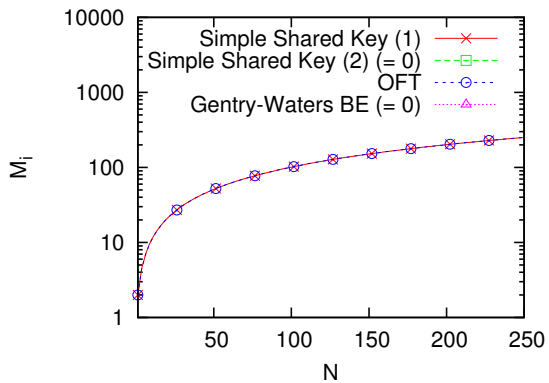
Quantitative Evaluation

TECHNISCHE UNIVERSITÄT DARMSTADT

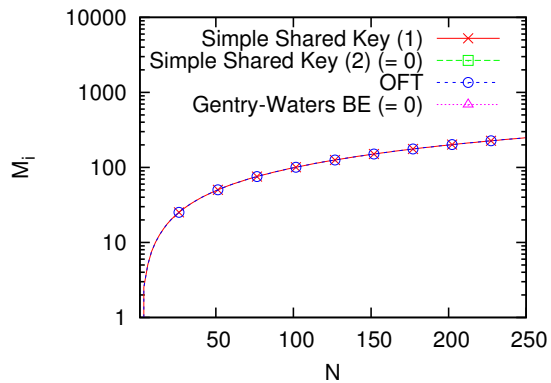
How many messages are sent?



(g) on initialization



(h) on user addition



(i) on user removal

- messages needed for a single attribute published for N users
- no messages needed for Shared Key with profile-side storage and BE
- client-side SK and OFT approaches need linear messaging

Impact on Real-Life Communities

TECHNISCHE UNIVERSITÄT DARMSTADT

Analysis for Facebook, Twitter, XING, Flickr (based on their own statistics)

community	# contacts	# attributes	# keys		storage (KB)*	
			SK	BE	SK	BE
facebook	150	180	~27000	332	650	8
twitter	50	180	~9000	232	220	6
XING	168	~36	~8350	220	200	5
flickr	12	200	2000	214	62	5

* 192bit keys (SE and BE)

- SK and BE costs differ by a factor of 10 to 80
  - SK profile-side storage adds factor "#contacts" ($\rightarrow$ quadratic, difference 10^2–10^4)
- SK and BE overhead remains below 1 MB which could be acceptable

Impact on Real-Life Communities

TECHNISCHE UNIVERSITÄT DARMSTADT

Analysis for Facebook, Twitter, XING, Flickr (based on their own statistics)

community	# contacts	# attributes	# keys		storage (KB)*	
			SK	BE	SK	BE
facebook	150	180	~27000	332	650	8
twitter	50	180	~9000	232	220	6
XING	168	~36	~8350	220	200	5
flickr	12	200	2000	214	62	5

* 192bit keys (SE and BE)

- SK and BE costs differ by a factor of 10 to 80
  - SK profile-side storage adds factor "#contacts" ($\rightarrow$ quadratic, difference 10^2–10^4)
- SK and BE overhead remains below 1 MB which could be acceptable
- generic approaches also applicable to secure established networks

Summary

TECHNISCHE UNIVERSITÄT DARMSTADT

- modern web dominated by social interaction
- decentralization to avoid omnipotent service provider
- introduces the need for encryption of profile data

Summary

TECHNISCHE UNIVERSITÄT DARMSTADT

- modern web dominated by social interaction
- decentralization to avoid omnipotent service provider
- introduces the need for encryption of profile data

- we introduced formal model for user profiles and profile management, security goal confidentiality, and privacy goal unlinkability
- two generic encryption approaches: shared key and broadcast encryption
- analysis in real-world settings shows low storage overhead
- practical trade-off between storage overhead and privacy

Summary

TECHNISCHE UNIVERSITÄT DARMSTADT

- modern web dominated by social interaction
- decentralization to avoid omnipotent service provider
- introduces the need for encryption of profile data

- we introduced formal model for user profiles and profile management, security goal confidentiality, and privacy goal unlinkability
- two generic encryption approaches: shared key and broadcast encryption
- analysis in real-world settings shows low storage overhead
- practical trade-off between storage overhead and privacy

- open question: possible to achieve unlinkability with sub-linear key overhead?
- implementation of plugin for existing OSN is ongoing

Security Goal: Confidentiality

Formal Definition

TECHNISCHE UNIVERSITÄT DARMSTADT

Confidentiality Game (high level):

1. Execute $\texttt{Init}(\kappa)$ for each user U.
2. $\mathcal{A}$ interacts with users through queries (incl. Corrupt) until it outputs
   - $(a, d_0), (a, d_1)$ two index-attribute pairs
   - G_t group of users
   - U_P profile owner who is not in G_t
3. Bit $b \in_R \{0,1\}^*$. Execute $\texttt{Publish}(pmk, P, (a, d_b), G_t)$.
4. $\mathcal{A}$ interacts with users through queries until it outputs some bit b^*.

$\mathcal{A}$ is successful if:

- $b = b^*$
- $\mathcal{A}$ did not corrupt U_P or any user who was ever authorized to access a
- $\mathcal{A}$ did not retrieve d_b trivially via some suitable $\texttt{Retrieve}$ query

Profile Management Scheme is confidential if for all $\mathcal{A}$:

$$|Pr[\text{successfull attack}] - 1/2| \text{ is negligible in } \kappa.$$

Security Goal: Unlinkability

Formal Definition

TECHNISCHE UNIVERSITÄT DARMSTADT

Unlinkability Game (high level):

1. Execute $\texttt{Init}(\kappa)$ for each user U.
2. $\mathcal{A}$ interacts with users through queries until it outputs
   - U_0, U_1 two users, (a, d) index-attribute pair, U_P profile owner
3. Bit $b \in_R \{0,1\}^*$.
   - If $(a, \cdot) \notin P$: execute $\texttt{Publish}(pmk, P, (a, d_b), \{U_b\})$.
   - If $(a, \cdot) \in P$: execute $\texttt{ModifyAccess}(pmk, P, a, U_b)$.
4. $\mathcal{A}$ interacts with users through queries until it outputs some bit b^*.

$\mathcal{A}$ is successful if:

- $b = b^*$
- U_P, U_0, and U_1 are uncorrupted
- $\mathcal{A}$ did not query Retrieve(P, a, U_0) or Retrieve(P, a, U_1)

Profile Management Scheme is unlinkable if for all $\mathcal{A}$:

$$|Pr[\text{successfull attack}] - 1/2| \text{ is negligible in } \kappa.$$

Security Goal: Anonymity

Formal Definition

TECHNISCHE UNIVERSITÄT DARMSTADT

Anonymity Game (high level):

1. Execute $\texttt{Init}(\kappa)$ for each user U.
2. $\mathcal{A}$ interacts with users through queries until it outputs
   - U_0, U_1 two users, (a, d) index-attribute pair, U_P profile owner
3. Bit $b \in_R \{0,1\}^*$.
   - If $(a, \cdot) \notin P$: execute $\texttt{Publish}(pmk, P, (a, d_b), \{U_b\})$.
   - If $(a, \cdot) \in P$: execute $\texttt{ModifyAccess}(pmk, P, a, U_b)$.
4. $\mathcal{A}$ interacts with users through queries until it outputs some bit b^*.

$\mathcal{A}$ is successful if:

- $b = b^*$
- U_P, U_0, and U_1 are uncorrupted
- $\mathcal{A}$ did not query Retrieve(P, a, U_0) or Retrieve(P, a, U_1)
- *U_0 authorized to access some attribute $\iff$ U_1 is also authorized*

Profile Management Scheme is unlinkable if for all $\mathcal{A}$:

$$\left|Pr[\text{successfull attack}] - 1/2\right| \text{ is negligible in } \kappa.$$