Privacy for Distributed Databases via (Un)linkable & Auditable Pseudonyms

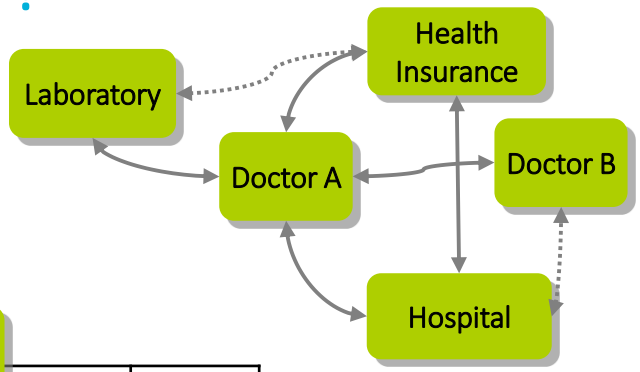
Anja Lehmann
IBM Research – Zurich

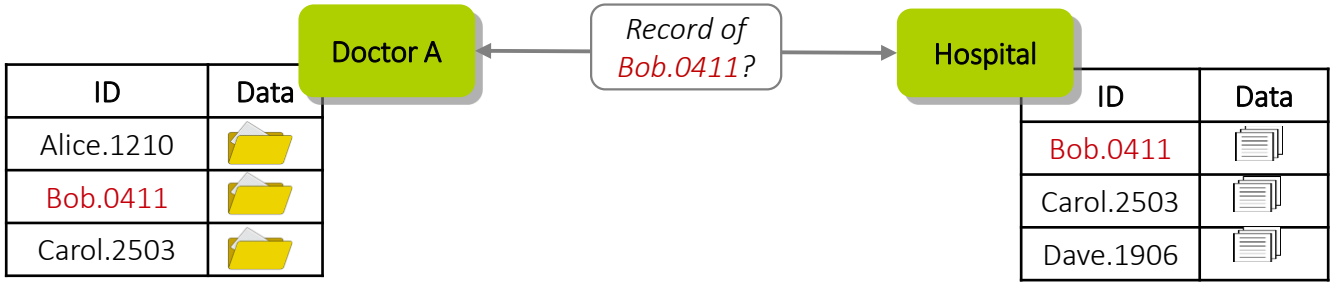
based on joint work with Jan Camenisch

IBM

How to maintain related yet distributed data ?

- examples: social security system, ehealth
  - different entities maintain data of citizens
  - eventually data needs to be exchanged or correlated

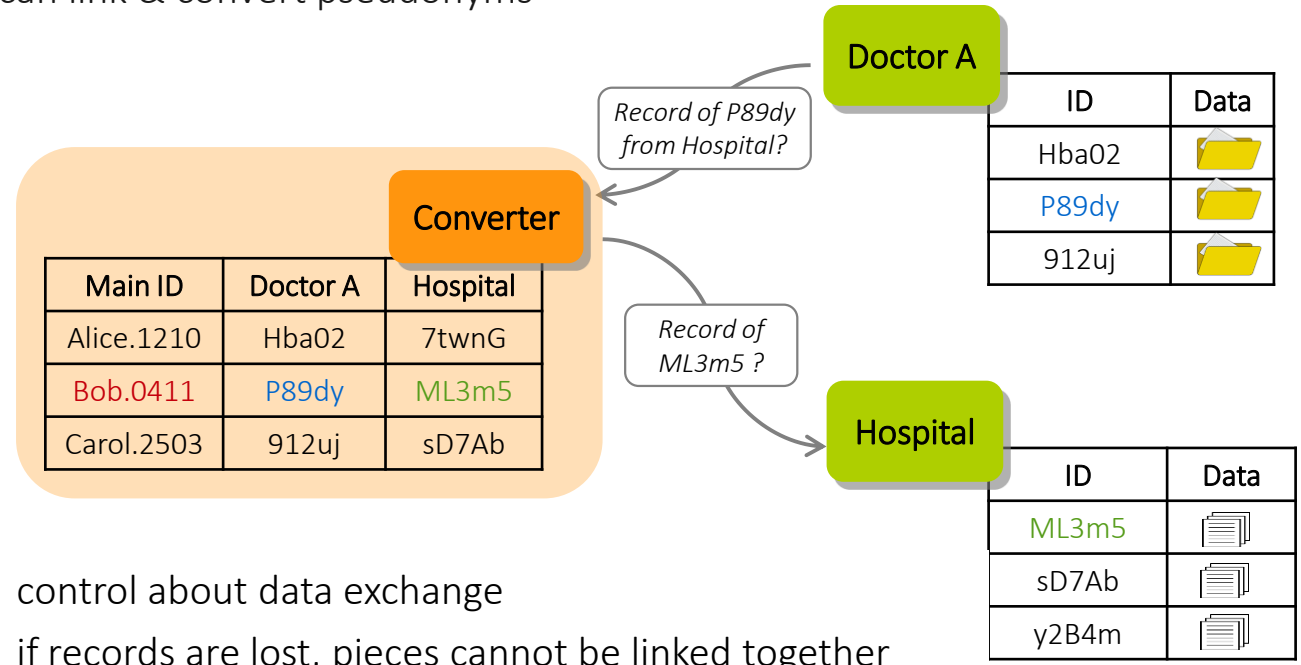




- simple solution: data gets associated with globally unique identifiers (e.g., US, Belgium, Sweden, ...)
- unique identifiers are security & privacy risk
  - no control about data exchange & usage
  - if associated data is lost, all pieces can be linked together
  - linkability of data allows re-identification of “anonymized” data (e.g. Netflix challenge)

Local Pseudonyms & *Trusted* Converter

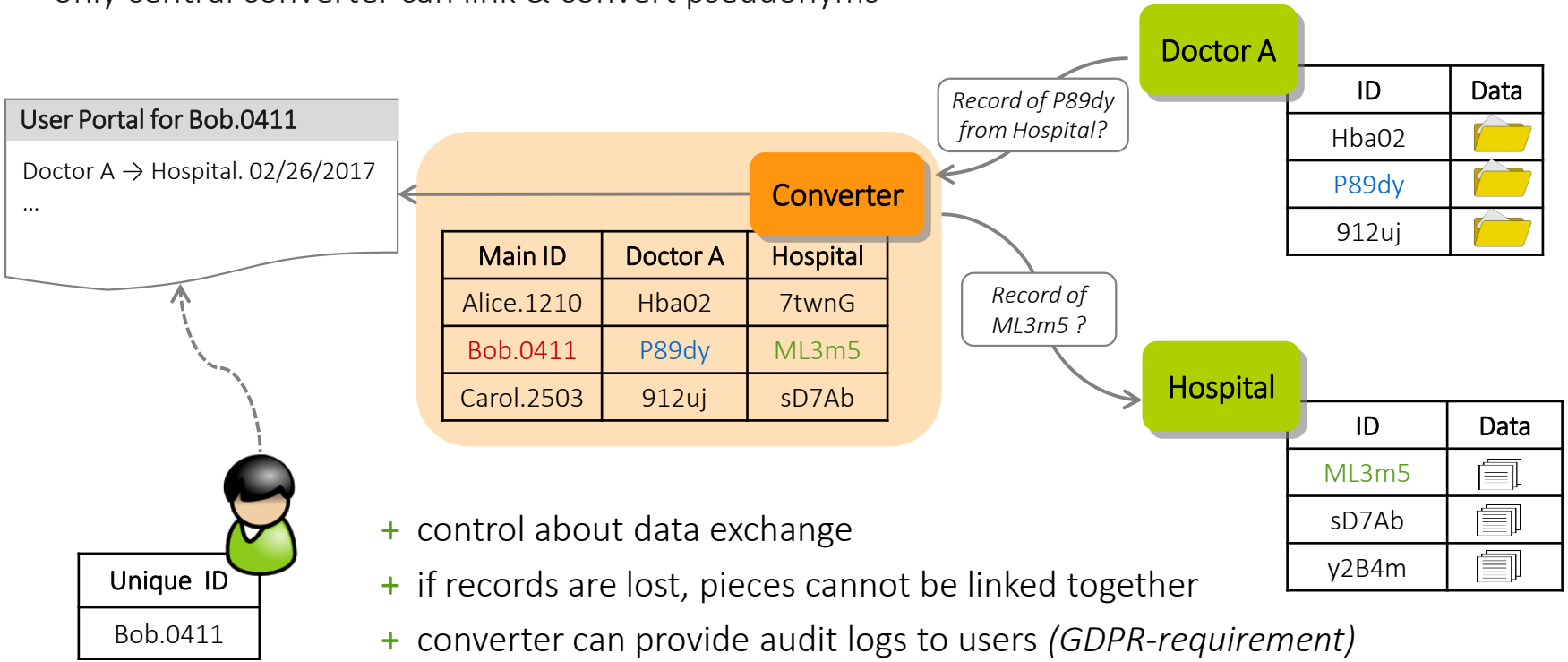
- user data is associated with (unlinkable) server-local identifiers aka “pseudonyms”
- only central converter can link & convert pseudonyms



+ control about data exchange
+ if records are lost, pieces cannot be linked together

Local Pseudonyms & *Trusted* Converter

- user data is associated with (unlinkable) server-local identifiers aka “pseudonyms”
- only central converter can link & convert pseudonyms



\+ control about data exchange

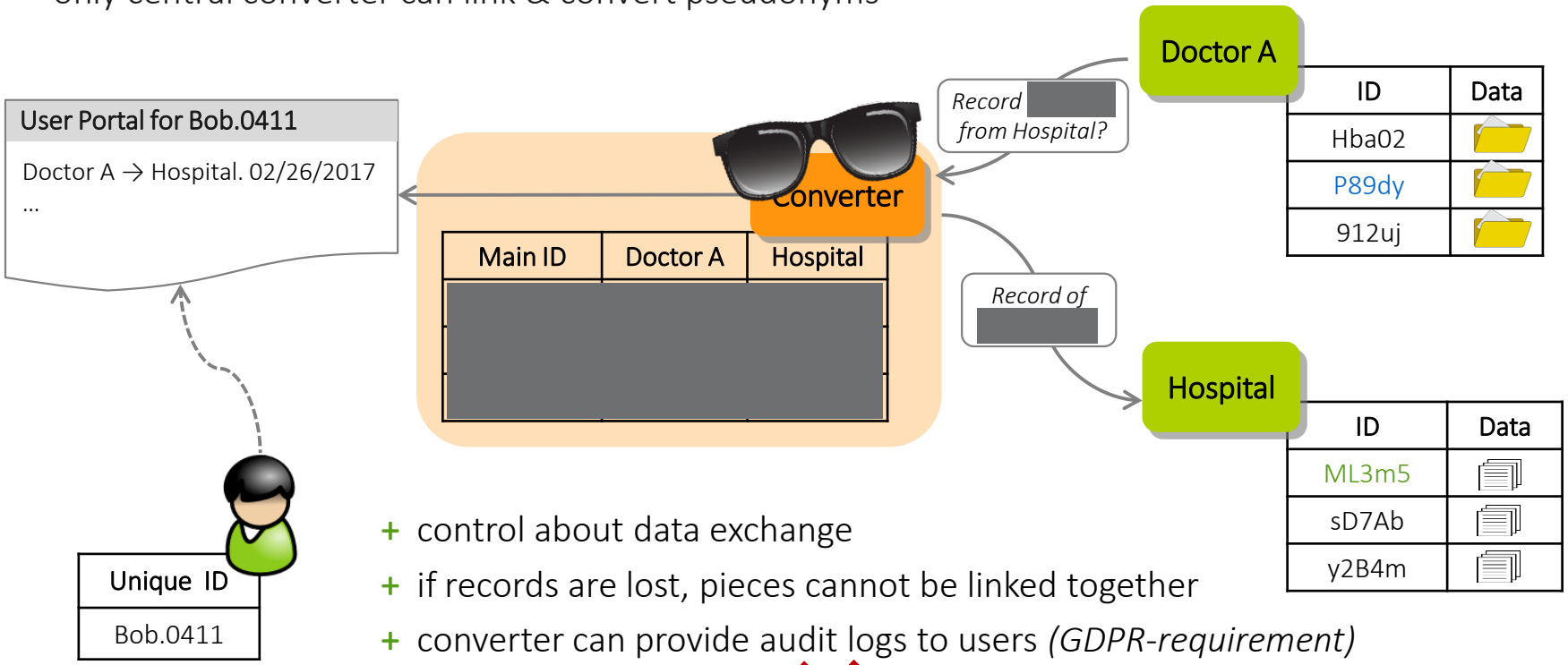
\+ if records are lost, pieces cannot be linked together

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

– converter learns all request & knows all correlations

Our Work: Local Pseudonyms & Oblivious Converter

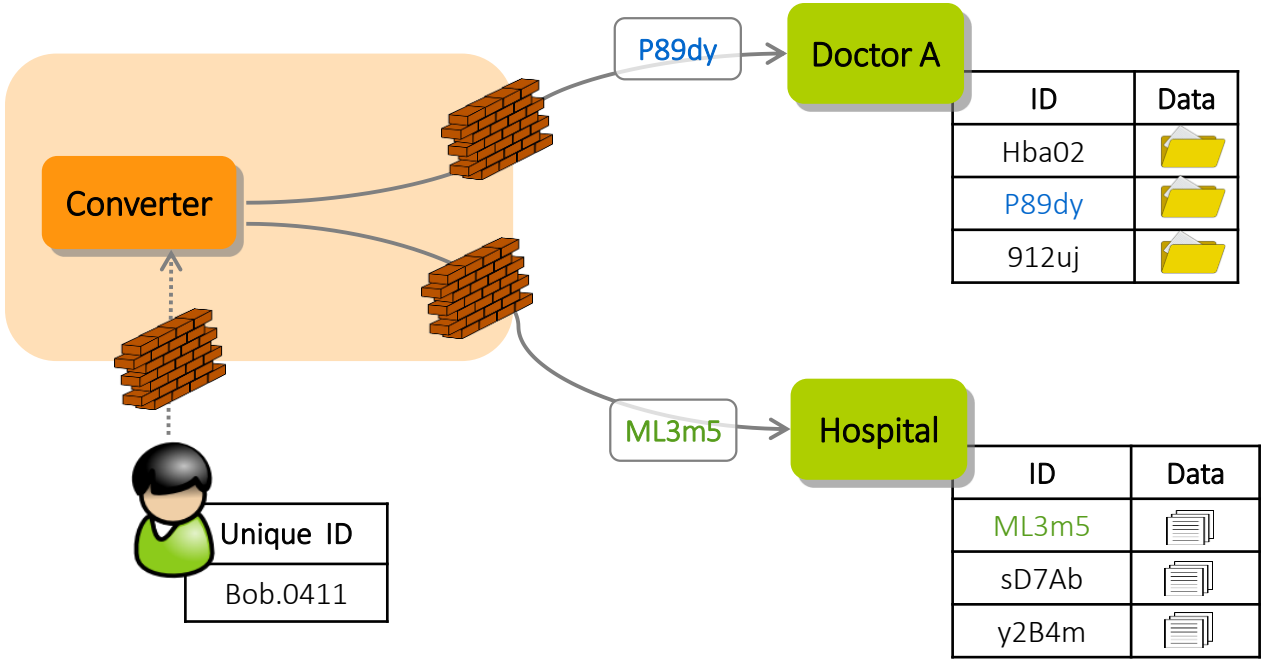
- user data is associated with (unlinkable) server-local identifiers aka “pseudonyms”
- only central converter can link & convert pseudonyms



+ 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 requests & knows all correlations

(Un)linkable Pseudonyms | Pseudonym Generation

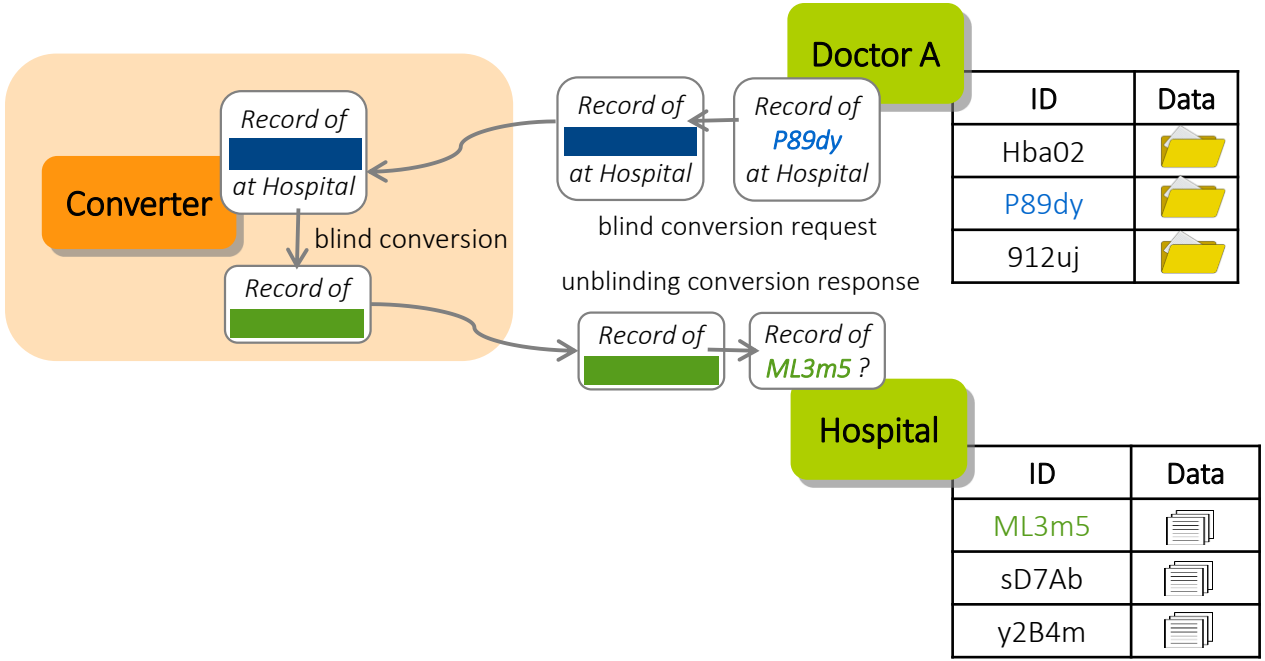
- user, converter & server jointly derive pseudonyms from unique identifiers



- [CL15] generation triggered by converter, knows unique IDs
- [CL17] oblivious pseudonym generation triggered by user

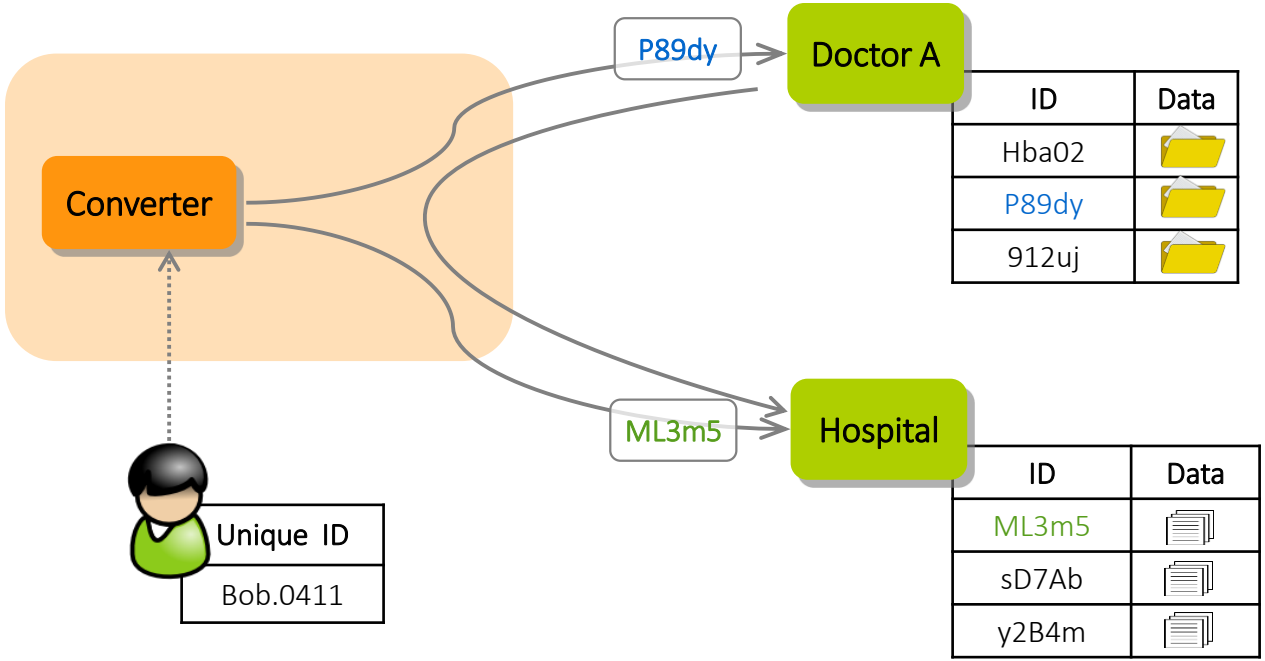
(Un)linkable Pseudonyms | Pseudonym Conversion

- only converter can link & convert pseudonyms, but does so in a blind way



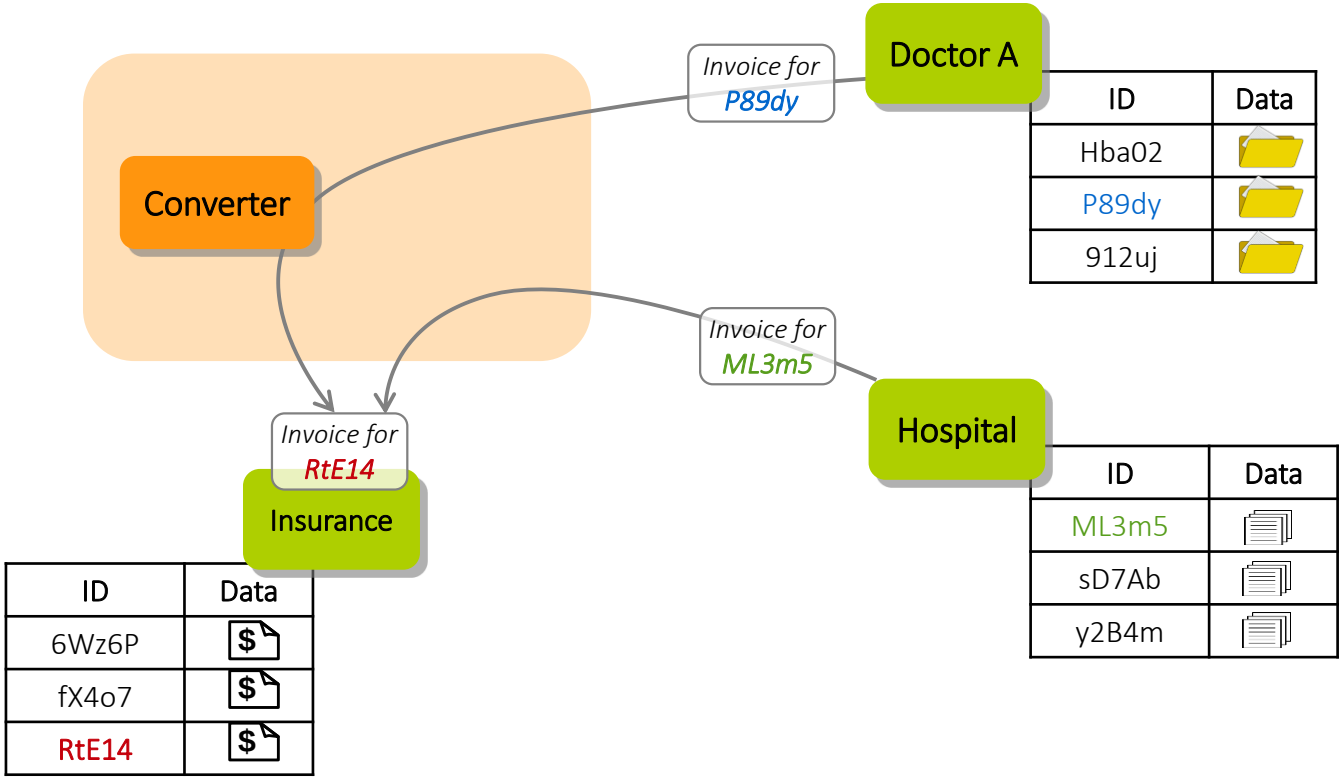
(Un)linkable Pseudonyms | Consistency

- pseudonym generation is deterministic & consistent with blind conversion



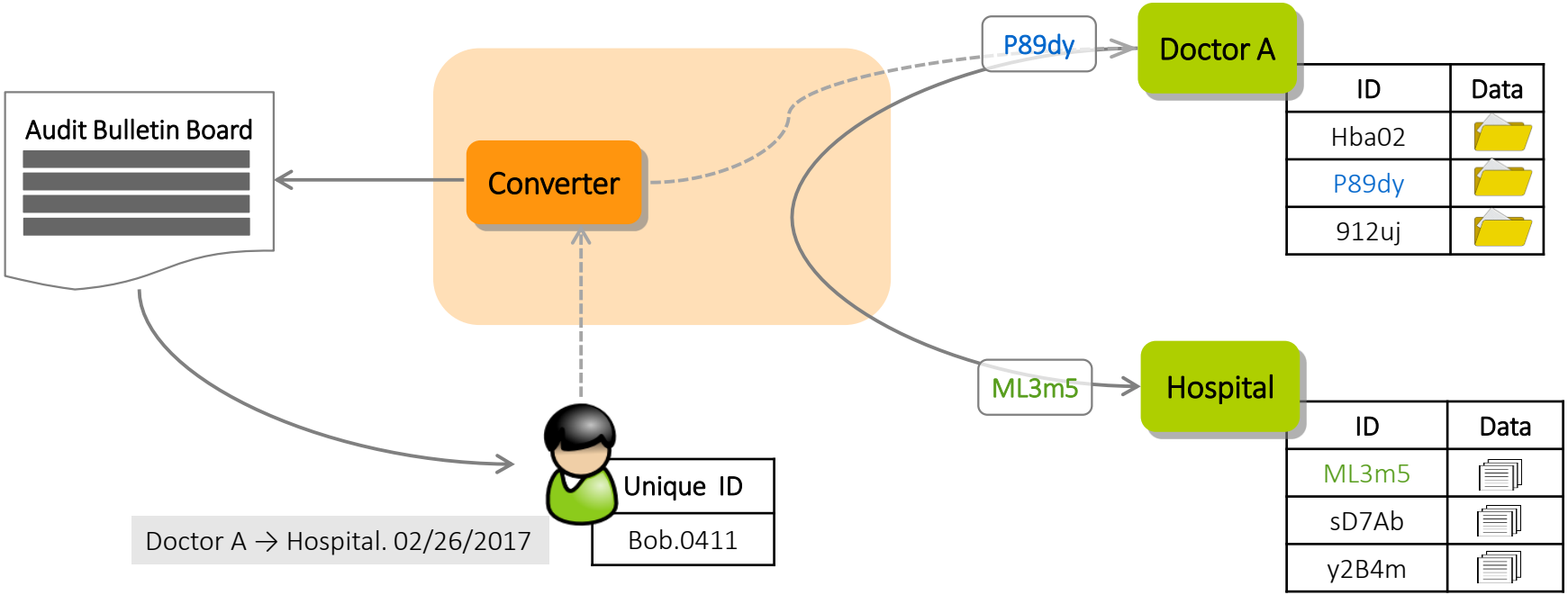
(Un)linkable Pseudonyms | Consistency

- pseudonym conversions are transitive, unlinkable data can be aggregated

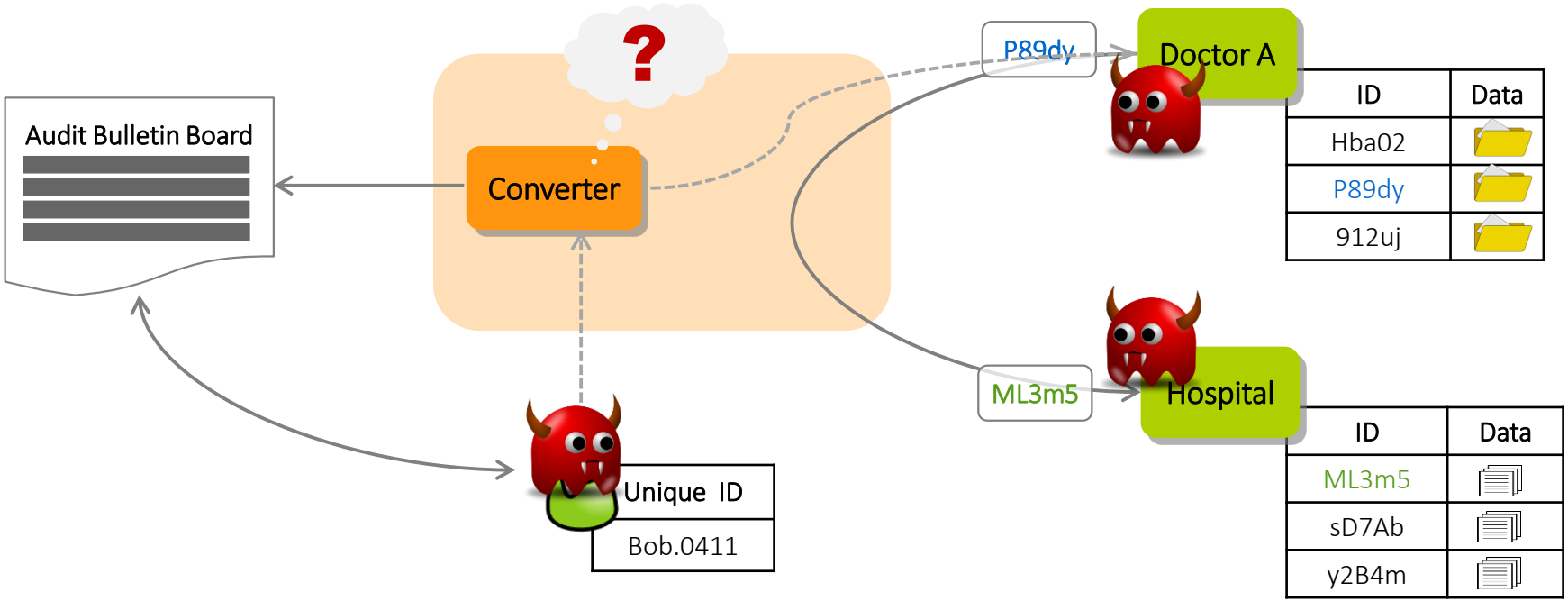


(Un)linkable Pseudonyms | User Audits

- [CL17] every pseudonym conversion triggers blind generation of audit log entry



(Un)linkable Pseudonyms | Corruption Model



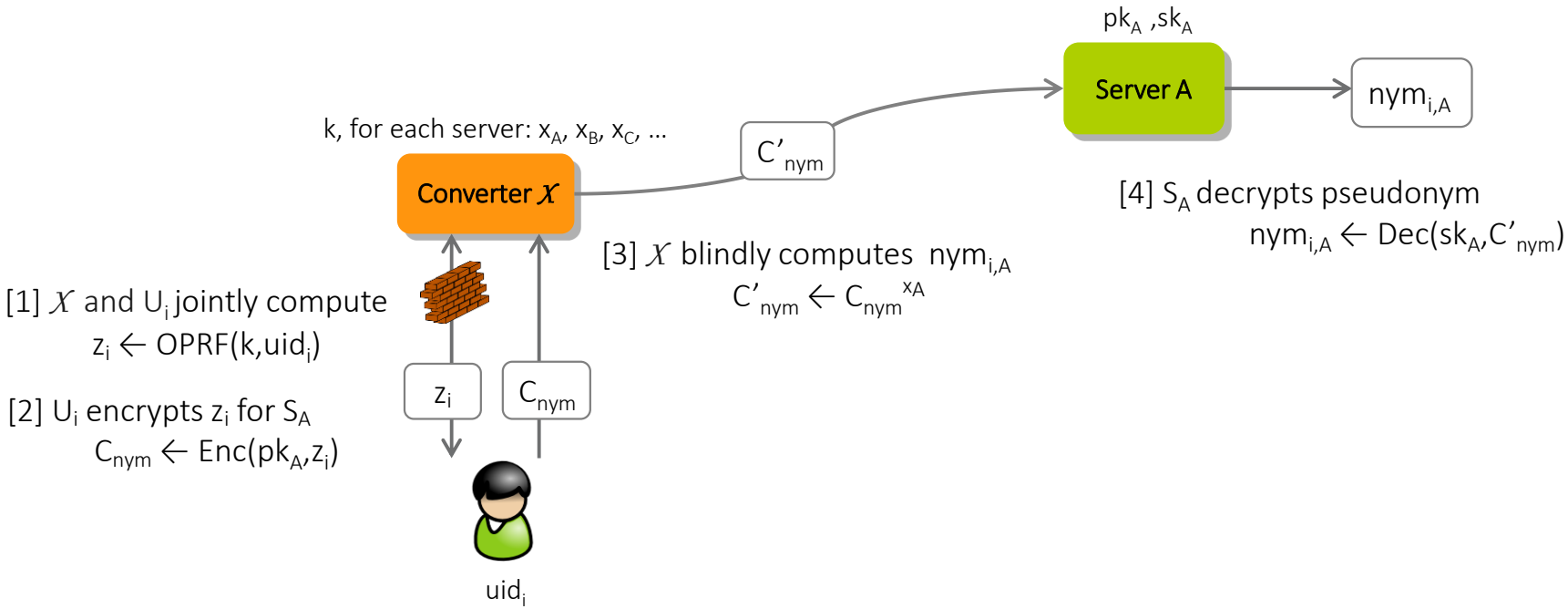
- servers and users can be fully corrupt
- converter at most honest-but-curious (w/o audits even fully corrupt [CL15])

Our Protocol

- high-level idea of convertible pseudonyms
- adding (efficient) auditability
- security against active adversaries

High-level Idea | Pseudonym Generation

Core Idea
Generation: $\mathcal{X}$ blindly computes $nym_{i,A} \leftarrow PRF(k,uid_i)^{x_A}$



High-level Idea | Pseudonym Conversion

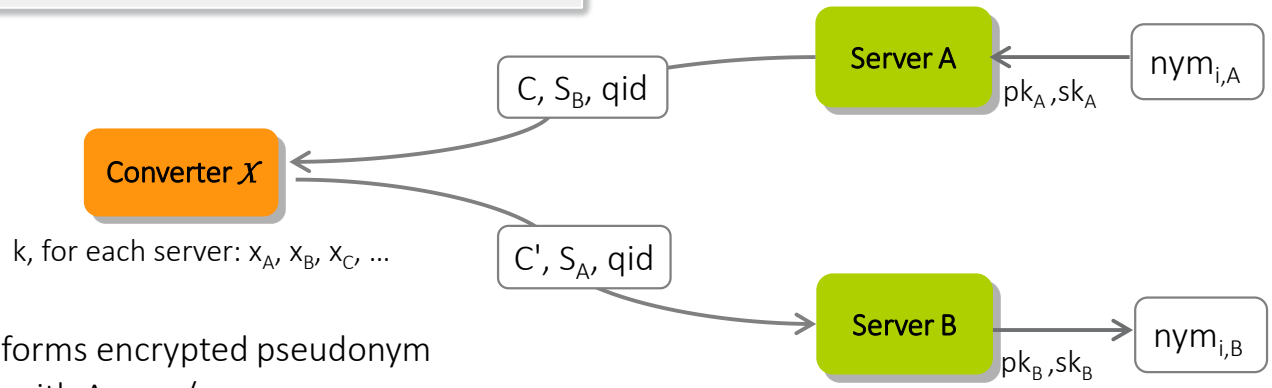
Core Idea

Generation: $\mathcal{X}$ blindly computes $nym_{i,A} \leftarrow PRF(k,uid_i)^{xA}$

Conversion: $\mathcal{X}$ blindly computes $nym_{i,B} \leftarrow nym_{i,A}^{xB\,/\,xA}$

[1] S_A encrypts $nym_{i,A}$ under S_B's key

$C \leftarrow Enc(pk_B, nym_{i,A})$



[2] $\mathcal{X}$ blindly transforms encrypted pseudonym

$C' \leftarrow C^{\Delta}$ with $\Delta = x_B / x_A$

$C' = Enc(pk_B, nym_{i,A})^{xB\,/\,xA}$

$= Enc(pk_B, PRF(k,uid_i)^{xA})^{xB\,/\,xA}$

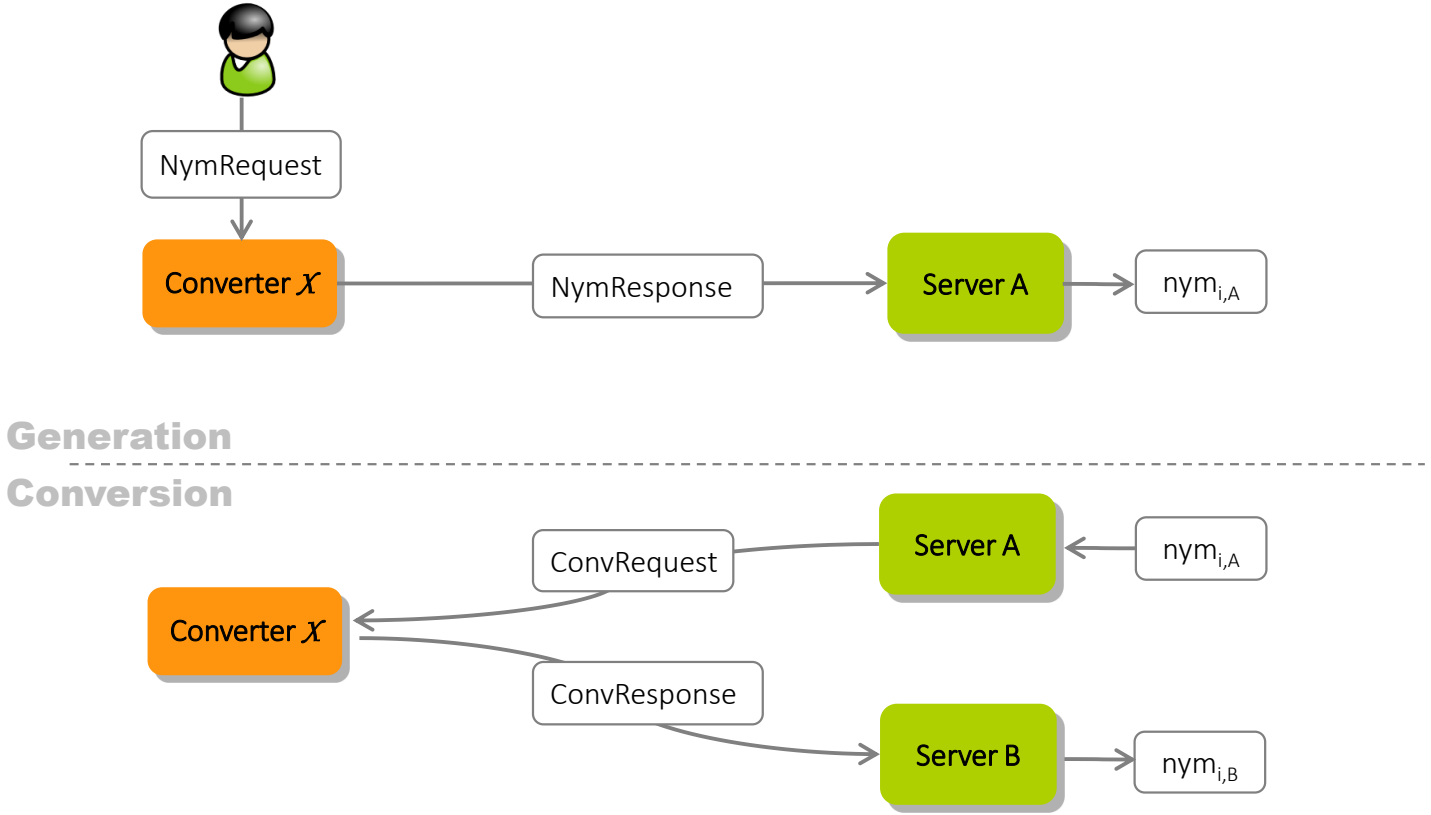
$= Enc(pk_B, PRF(k,uid_i)^{xB})$

$= Enc(pk_B, nym_{i,B})$

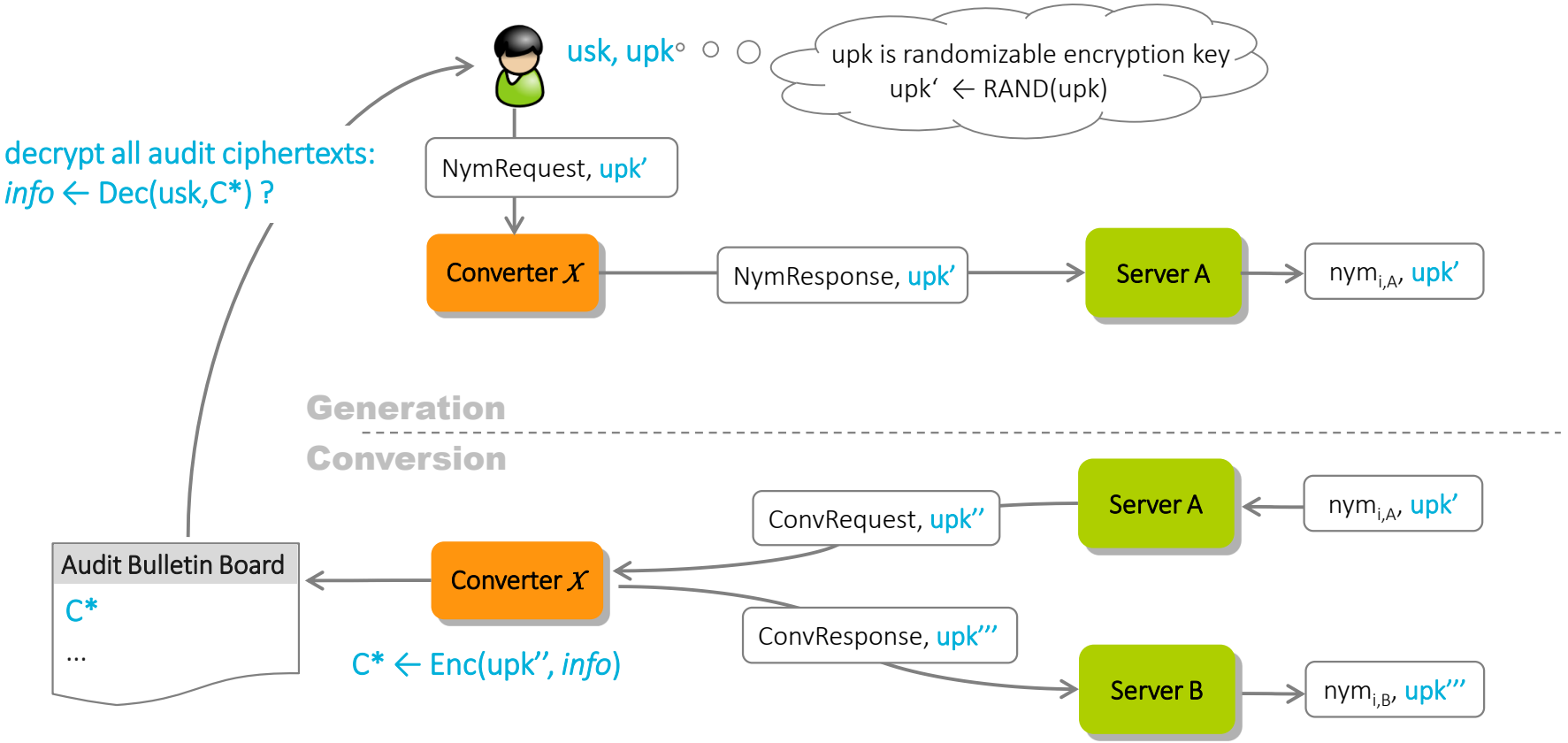
[3] S_B decrypts converted pseudonym

$nym_{i,B} \leftarrow Dec(sk_B, C')$

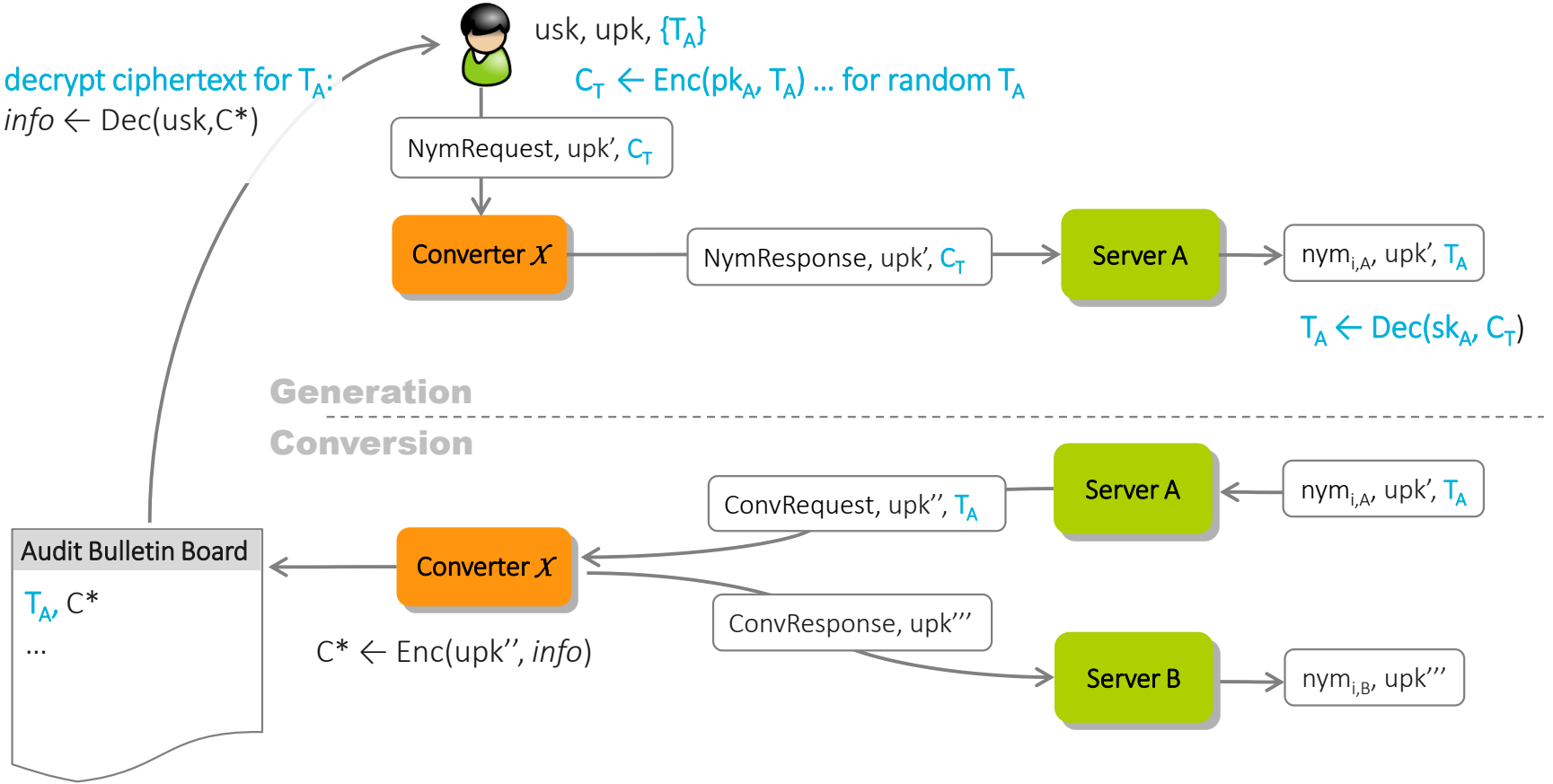
High-level Idea | Overview



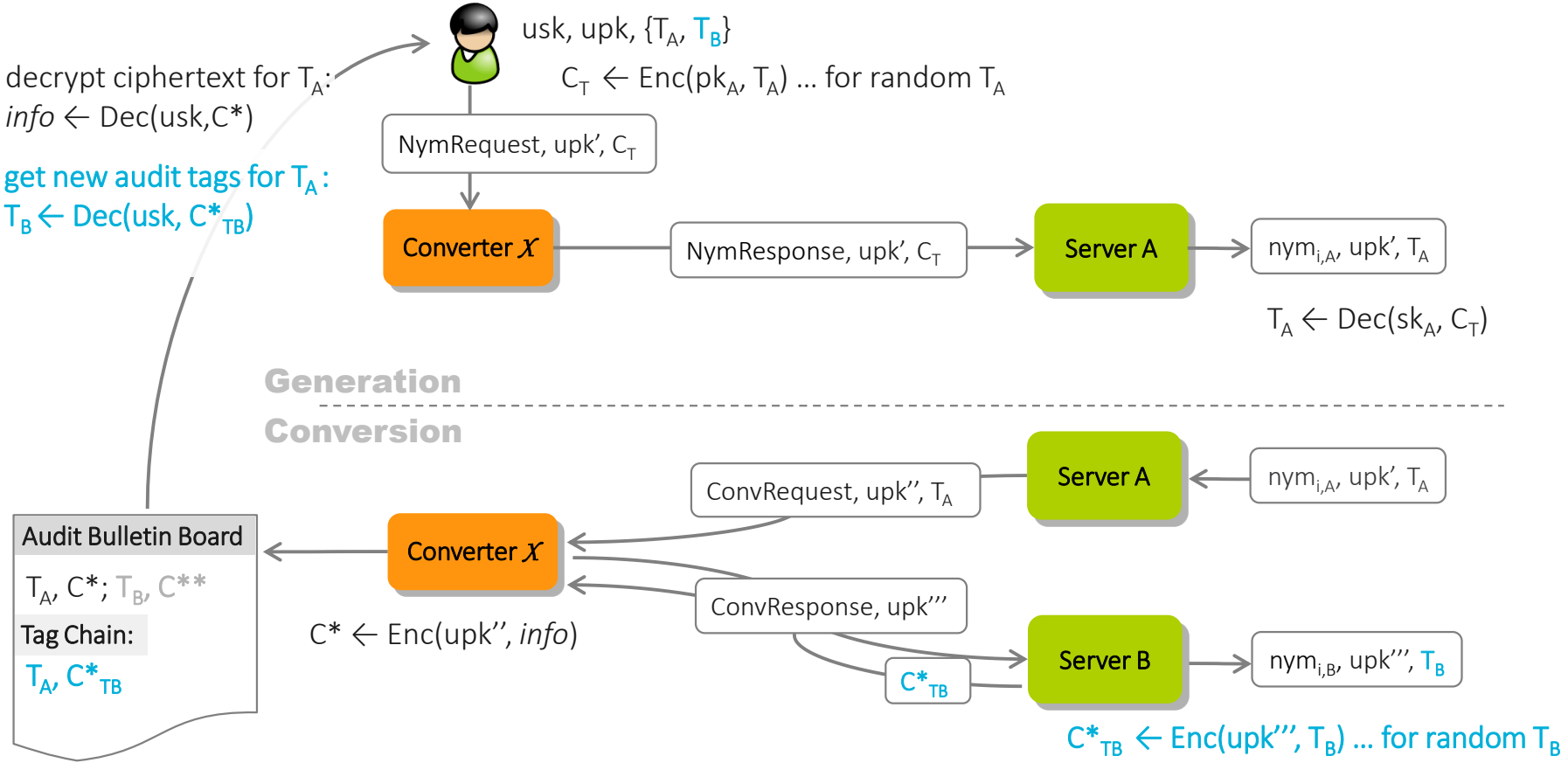
High-level Idea | Adding Auditability



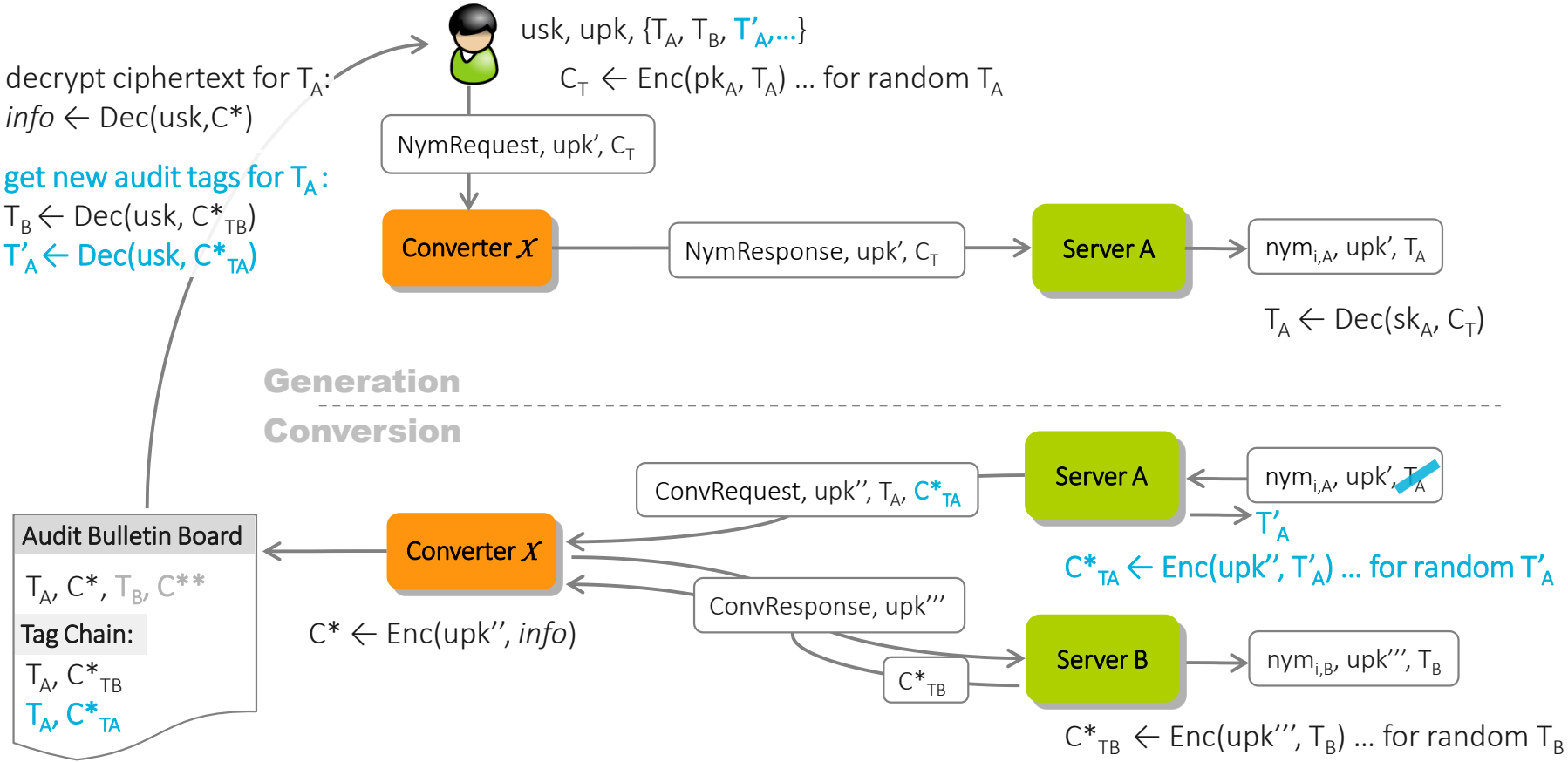
High-level Idea | Adding *Efficient* Auditability (via Audit Tags)



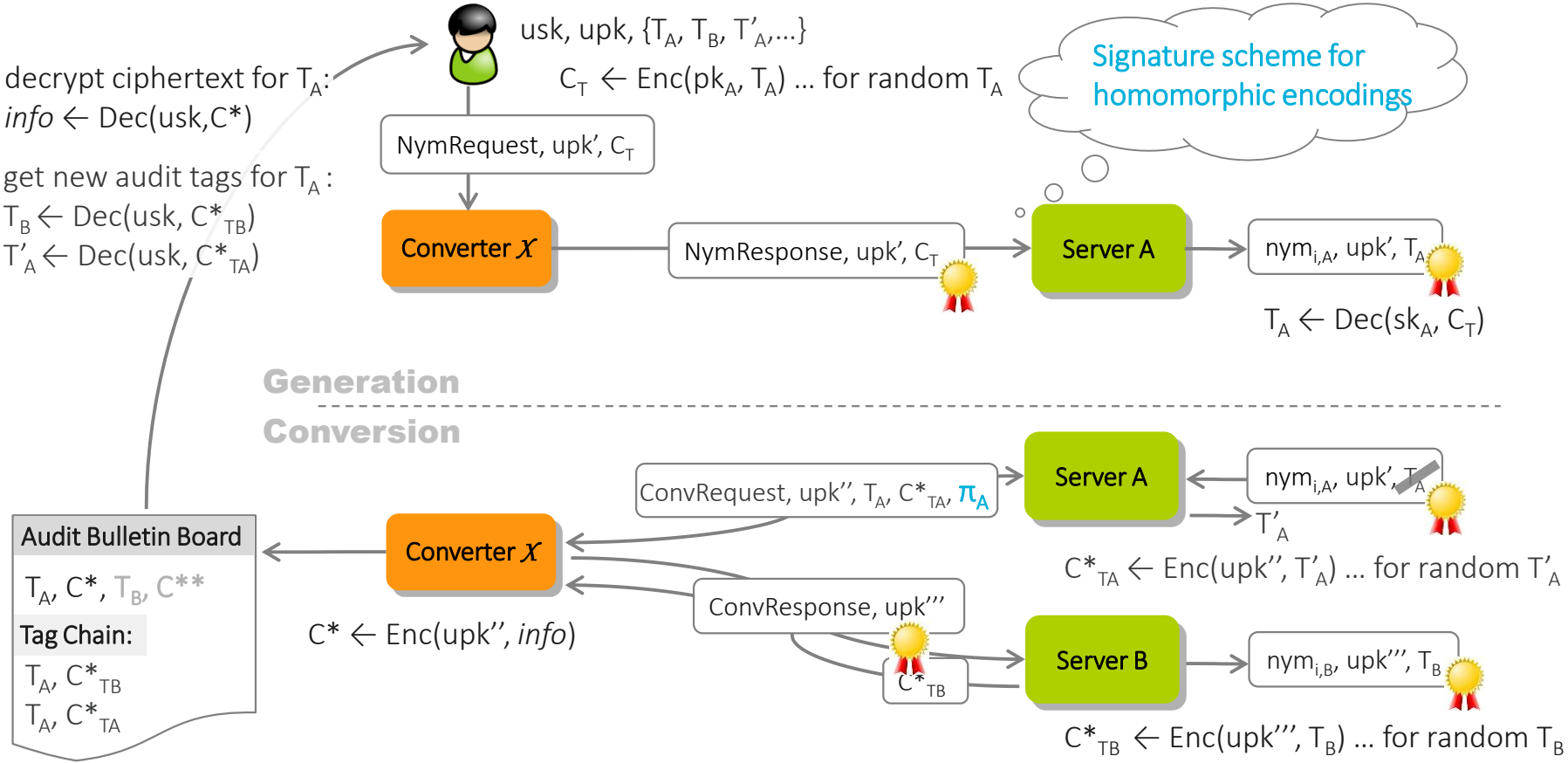
High-level Idea | Adding *Efficient* Auditability (via Audit Tags)



High-level Idea | Adding *Efficient* Auditability (via Audit Tags)



High-level Idea | Security against Active Adversaries



(Un)linkable & Auditable Pseudonyms | Summary

- provably secure in the Universal Composability (UC) framework
  - converter honest-but-curious, users & servers actively corrupt
- concrete instantiation ~50ms computational time per party for conversion
  - ElGamal-based encryption, Groth+ signature scheme for encoded messages, Dodis-Yampolskiy-based OPRF

Summary

- pseudonym scheme for (un)linkable data storage with controlled & auditable data exchange
- pseudonyms can only be linked via a central converter
- conversions & audit logs are done in a blind way → converter must not be a trusted entity

→ paradigm shift: unlinkability per default, linkability only when necessary

Thanks!

Questions?

(Un)linkable Pseudonyms for Governmental Databases. CCS15.
Privacy-Preserving User-Auditable Pseudonym Systems. IEEE EuroSP17.
anj@zurich.ibm.com

(Un)linkable & Auditable Pseudonyms | Security & Efficiency

- provably secure construction based on
  - homomorphic encryption scheme (ElGamal encryption)
  - homomorphic encryption scheme with re-randomizable public keys (ElGamal-based)
  - oblivious pseudorandom function (based on Dodis-Yampolskiy-PRF)
  - signature scheme for homomorphic encoding functions (Groth+)
  - zero-knowledge proofs (Fiat-Shamir NIZKs)
  - commitment scheme (ElGamal based)

Pseudonym Generation :	$\mathcal{U}_i$ $22\mathbb{G} + 6\mathbb{Z}^*_{n^2}$	$\mathcal{X}$ $22\mathbb{G} + \tilde{\mathbb{G}} + 6\mathbb{Z}^*_{n^2}$	$\mathcal{S}_A$ $4\mathbb{G}$
Pseudonym Conversion :	$\mathcal{S}_A$ $37\mathbb{G} + 4P$	$\mathcal{X}$ $47\mathbb{G} + 2\tilde{\mathbb{G}} + 4P$	$\mathcal{S}_B$ $13\mathbb{G}$
User Audit :	$\mathcal{U}_i$ $3c \cdot \mathbb{G}$	*with c denoting the amount of conversions for $\mathcal{U}_i$*	