

# Cryptography 4 People

where crypto & security should be heading

Jan Camenisch

IBM Research – Zurich

[jca@zurich.ibm.com](mailto:jca@zurich.ibm.com), [@JanCamenisch](https://twitter.com/JanCamenisch), [ibm.biz/jancamenisch](https://ibm.biz/jancamenisch)

## Facts

33% of cyber crimes, including identity theft, take less time than to make a cup of tea.



## Facts

10 Years ago, your identity information on the black market was worth \$150. Today....



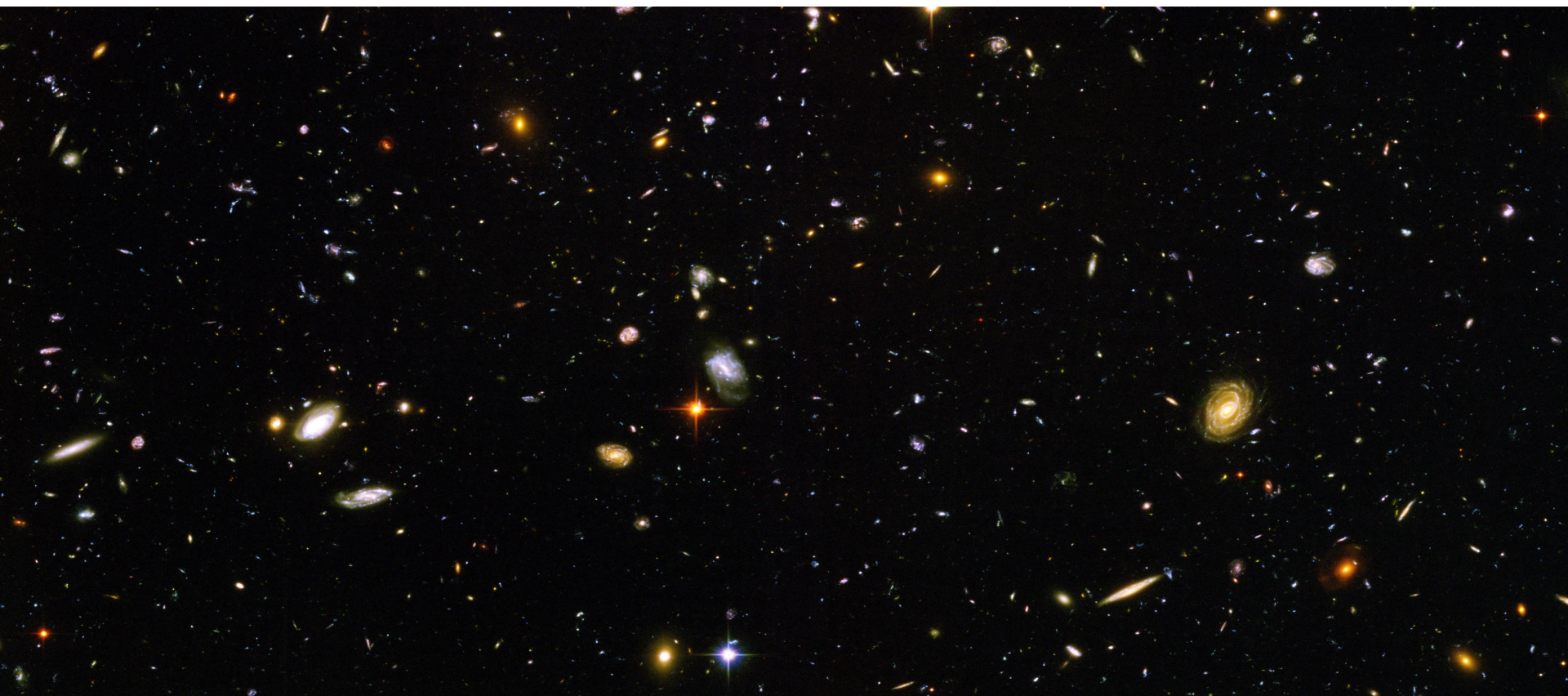
## Facts

\$15'000'000'000 cost of identity theft worldwide (2015)





Attackers hide easily in the vast of cyberspace







Houston, we  
have a problem!





...computers never forget



- Data is stored by default
- Data mining gets ever better
- Apps built to use & generate (too much) data
- New (ways of) businesses using personal data



- Humans forget most things too quickly
- Paper collects dust in drawers

*But that's how we design and build applications!*



# Learnings from Snowden – Very Short Summary

## *Massive* scale mass surveillance

- Meta data vs plain texts
- Google's data from companies (e.g., Google), Industrial “collaborations”, industrial espionage
- But also from underwater cables

Weak access control to (the large amount of) collected data (security clearance)

## Technical sophistication (hardly a surprise)

- Rigged equipment, chips, etc
- Redundancy of access to corp. data
- Subverted standards (PRG)
- Control of CAs → control of network



# Learnings from Snowden – Take Aways

*Not* by breaking encryption schemes! But using insecurity of systems, etc.

- However, Snowden had limited access to docs (no crypt-analysis reports)

Many things doable by ordinary hackers or somewhat sophisticated crooks

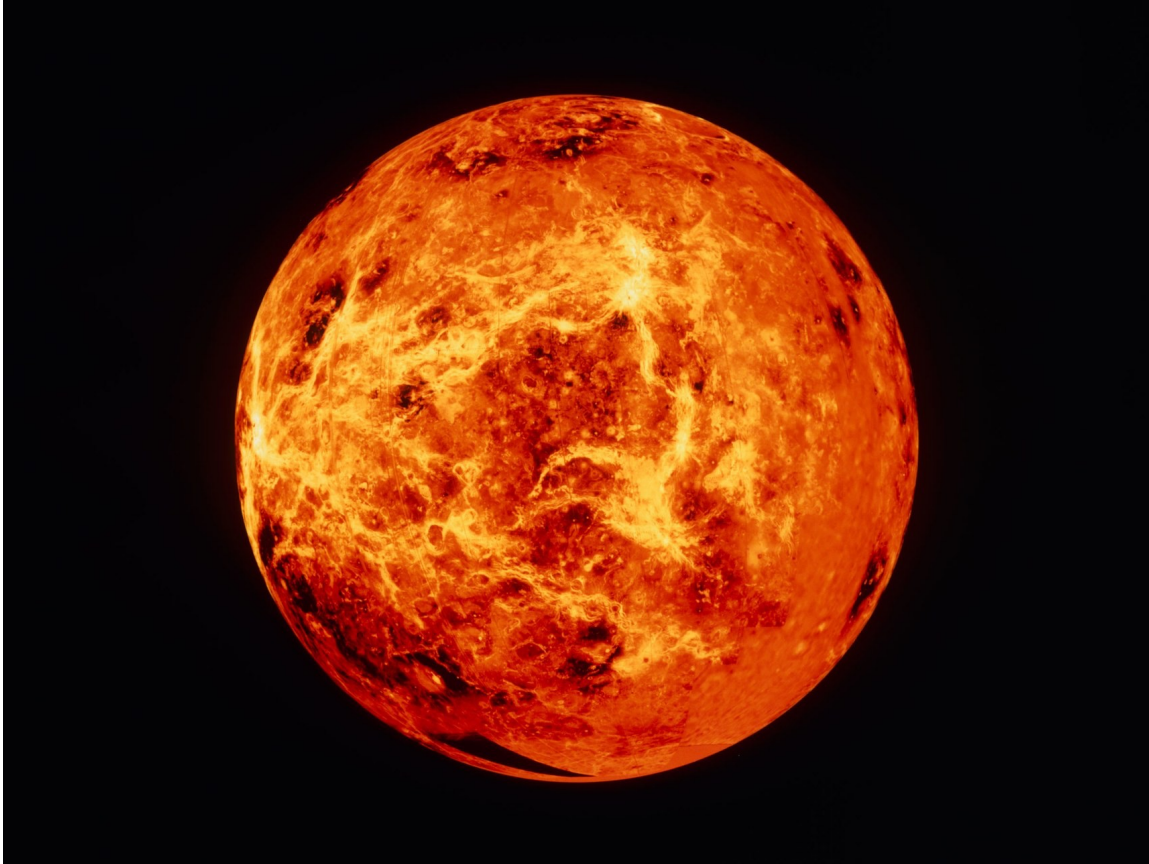
- Some of CA infiltration
- Stealing data at rest

Other things require large budget and organization

- FPGA, ASICS
- Deliberate weakening of infrastructure (PRG standards, etc) - *very* bad idea



So it seems our environment is even nastier...

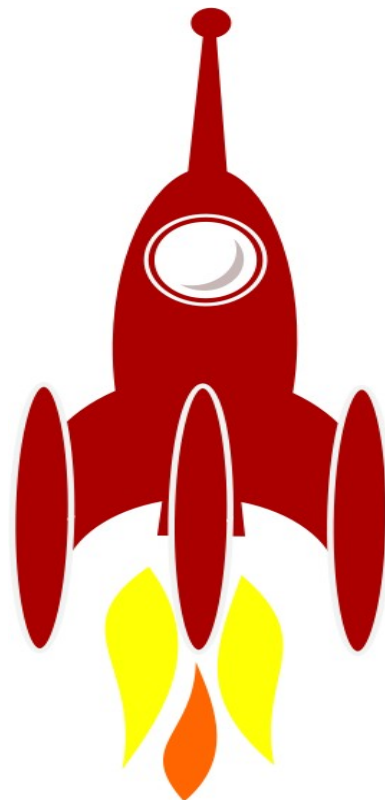




Security & Privacy is not a lost cause!

We need paradigm shift:

*build stuff for the moon  
rather than the sandy beach!*



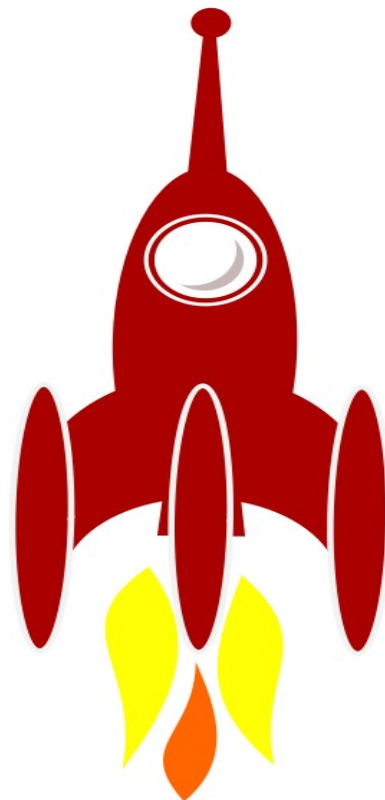


Security & Privacy is not a lost cause!

That means:

- Use only minimal data necessary
- Encrypt every bit – and keep it like that
- Attach usage policies to each bit

Good news: Cryptography allows for that!



# Cryptography to the Aid!



Mix Networks      Oblivious Transfer

Searchable Encryption

Onion Routing

Confirmer signatures

Anonymous Credentials

Group signatures

Pseudonym Systems

OT with Access Control

e-voting

Priced OT

Blind signatures

Private information retrieval

Secret Handshakes

Homomorphic Encryption



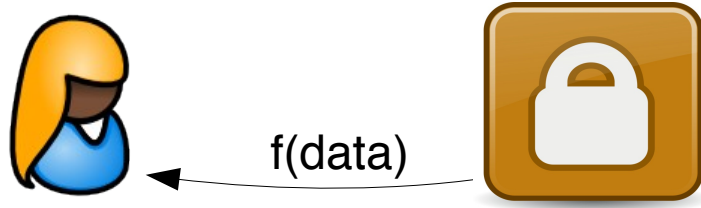
A stack of several books is positioned in the center of the frame, resting on a rough, textured surface that resembles stone or concrete. The books are of various thicknesses and colors, with some spines visible. The background is a mottled, greyish-blue texture. The text is overlaid on this background.

# Cryptography to the Aid

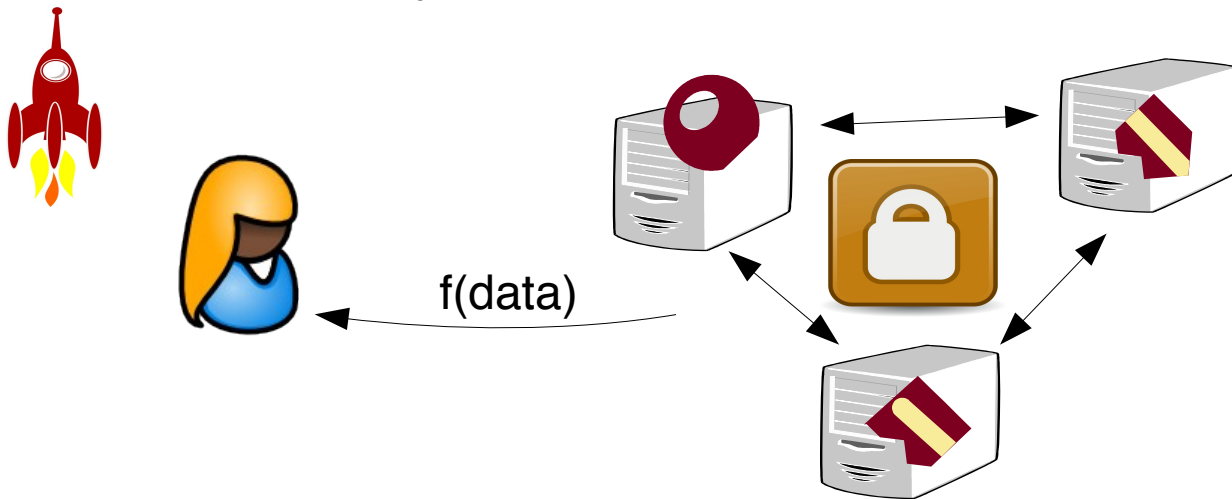
*a few examples of rocket science*

## Multi Party Computation

# Data Protection

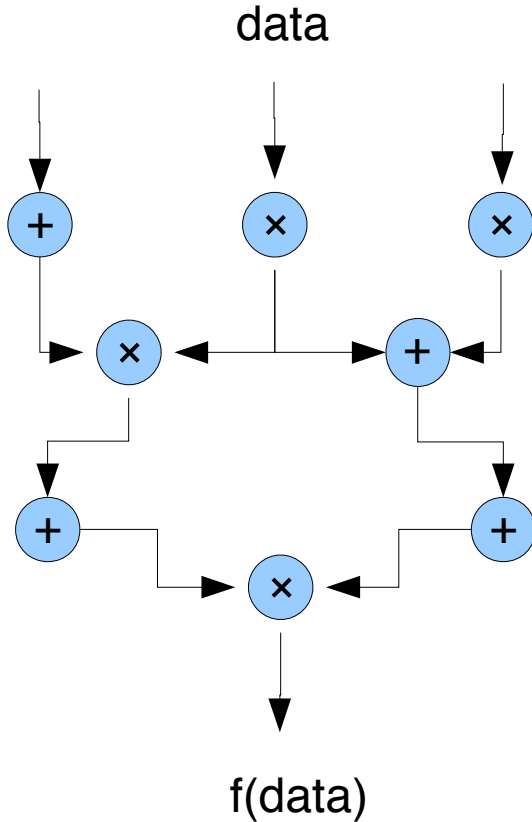
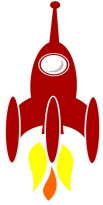


# Secure Multi Party Computation for Data Protection

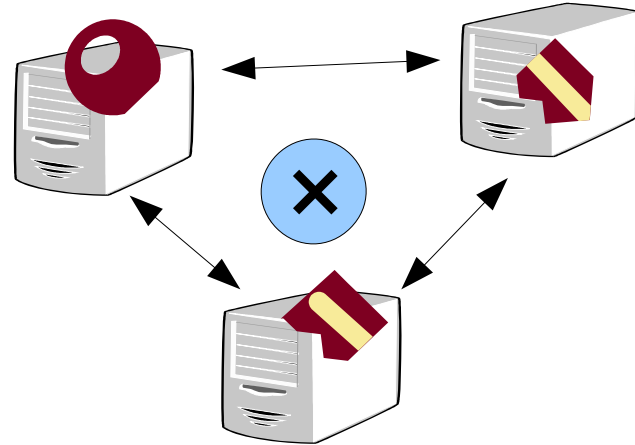


- Can be done for any function – typically not considered efficient.
- Two or three parties protocols today can be very efficient
  - e.g., computing AES in 100ms 3PC with one party corrupt

# Multi Party Computation – Basic Principles



Evaluate Circuit gate per gate  
with distributed protocol

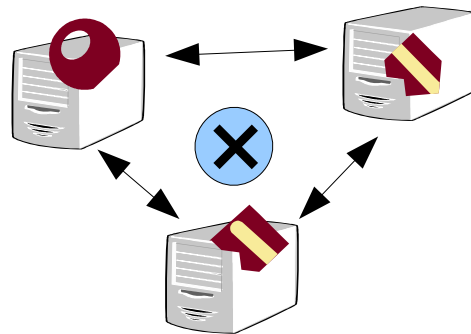


# Multi Party Computation – Basic Principles

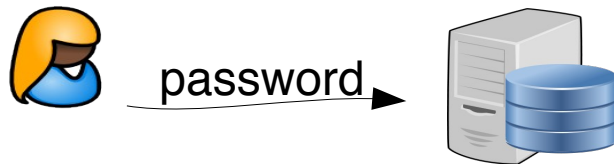


## Main approaches:

- Computation of gates
  - $+$  typically for free
  - $\times$  requires protocols
- Encrypted data under shared key
  - (Fully) homomorphic encryption...
- Secret-share data, compute with shares



# Dedicated MPC Protocol – Today: Getting Rid of Password Databases

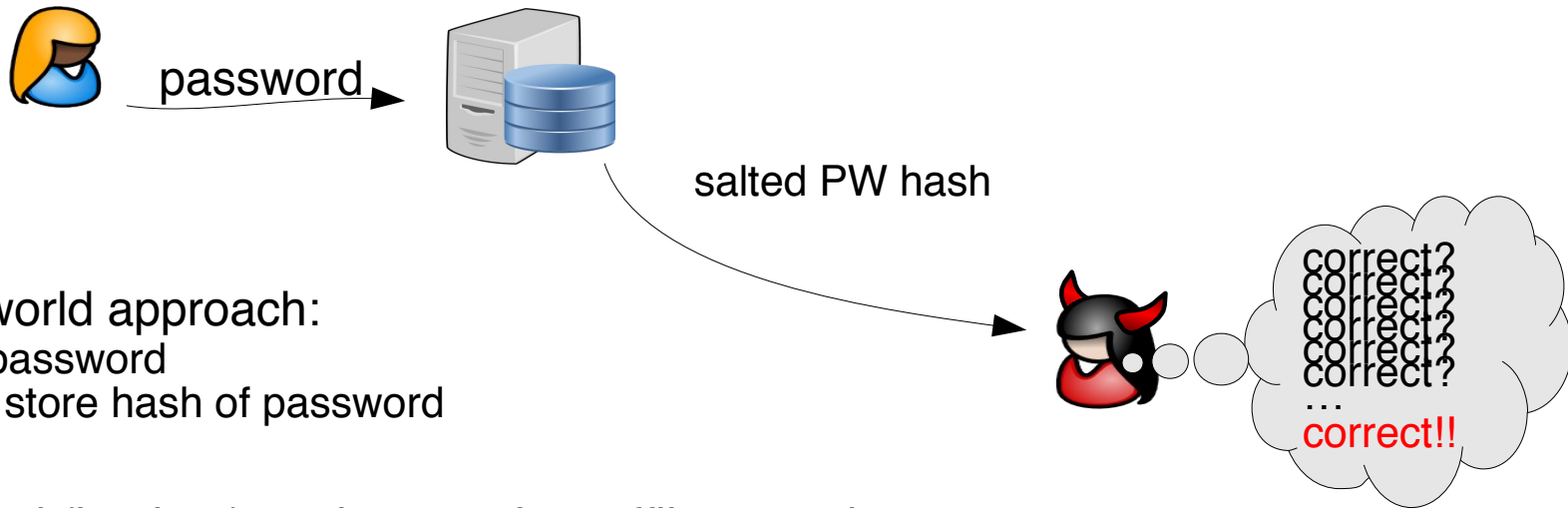


Paper-world approach:

- store password
- better, store hash of password



# Dedicated MPC Protocol – Today: Getting Rid of Password Databases



## Paper-world approach:

- store password
- better, store hash of password

## Password (hashes) useless against offline attacks

- Human-memorizable passwords are inherently weak
- NIST: 16-character passwords have 30 bits of entropy  $\approx$  1 billion possibilities
- Rig of 25 GPUs tests 350 billion possibilities / second, so  $\approx$  3ms for 16 chars
- 60% of LinkedIn passwords cracked within 24h

# Homomorphic Encryption

## Encryption scheme

$$KGen(l) \rightarrow (PK, SK)$$

$$C = Enc_{PK}(m)$$

$$m = Dec_{SK}(c)$$



## Plaintext homomorphism

$$Enc_{PK}(m1) \diamond Enc_{PK}(m2) \quad \Leftrightarrow \quad Enc_{PK}(m1 * m2)$$

$$Enc_{PK}(m) \diamond Enc_{PK}(m) = Enc_{PK}(m)^2 \quad \Leftrightarrow \quad Enc_{PK}(m^2)$$

## Secret key homomorphism

$$SK = SK1 * SK2 \quad \Rightarrow \quad Dec_{SK1}(Dec_{SK2}(Enc_{PK}(m))) = m = Dec_{SK1}(Dec_{SK2}(Enc_{PK}(m)))$$

# Dedicated MPC Protocol – Today: Getting Rid of Password Databases



Account Setup



$p$



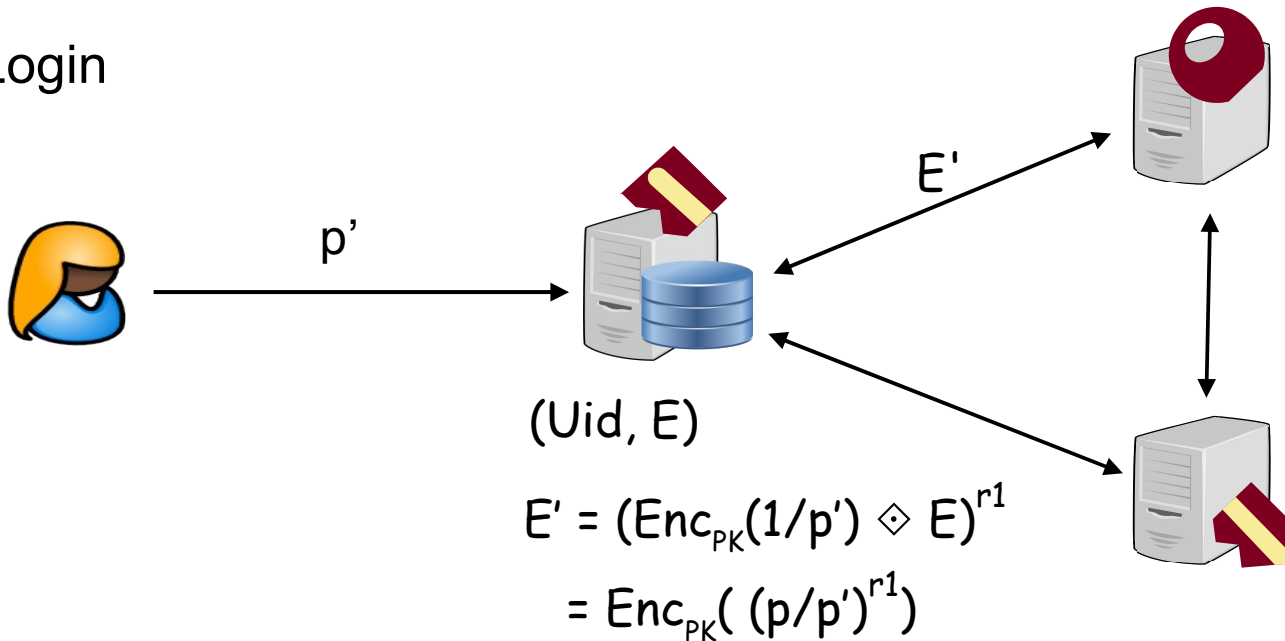
$Uid, E = \text{Enc}_{PK}(p)$



# Dedicated MPC Protocol – Today: Getting Rid of Password Databases



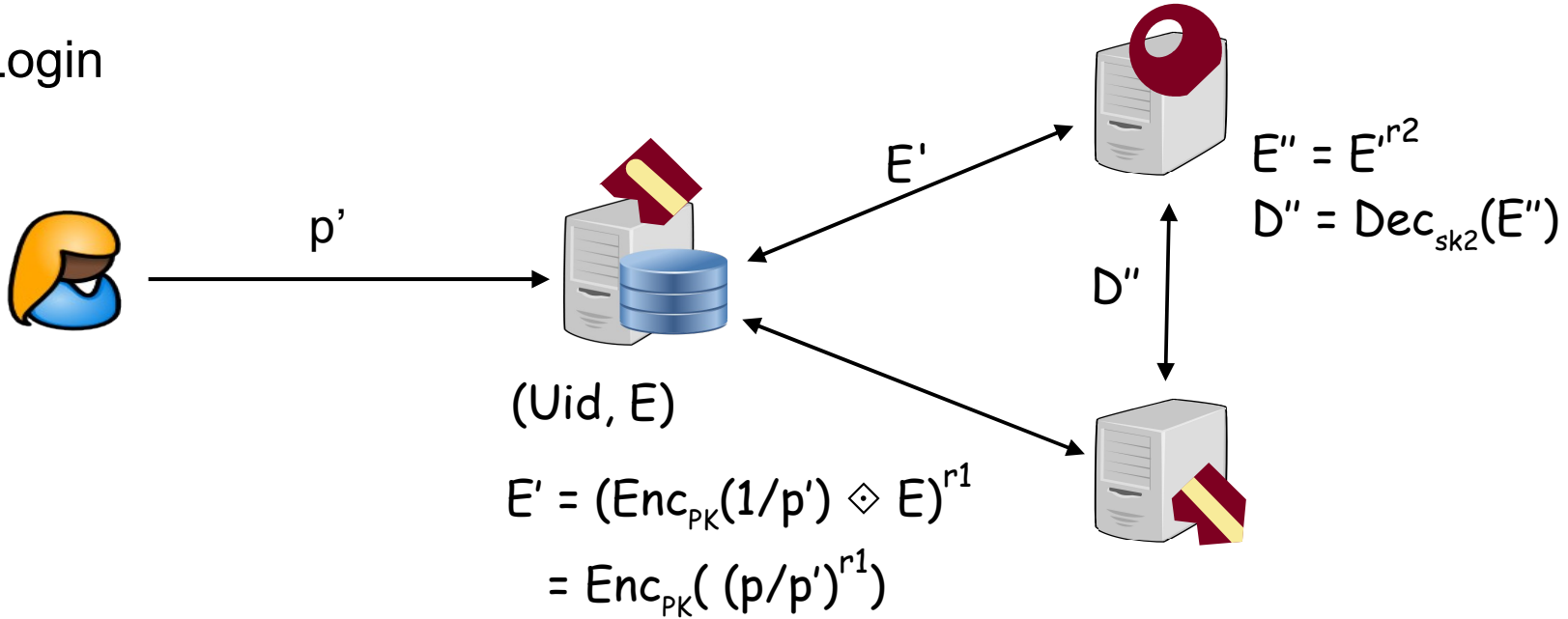
Login



# Dedicated MPC Protocol – Today: Getting Rid of Password Databases



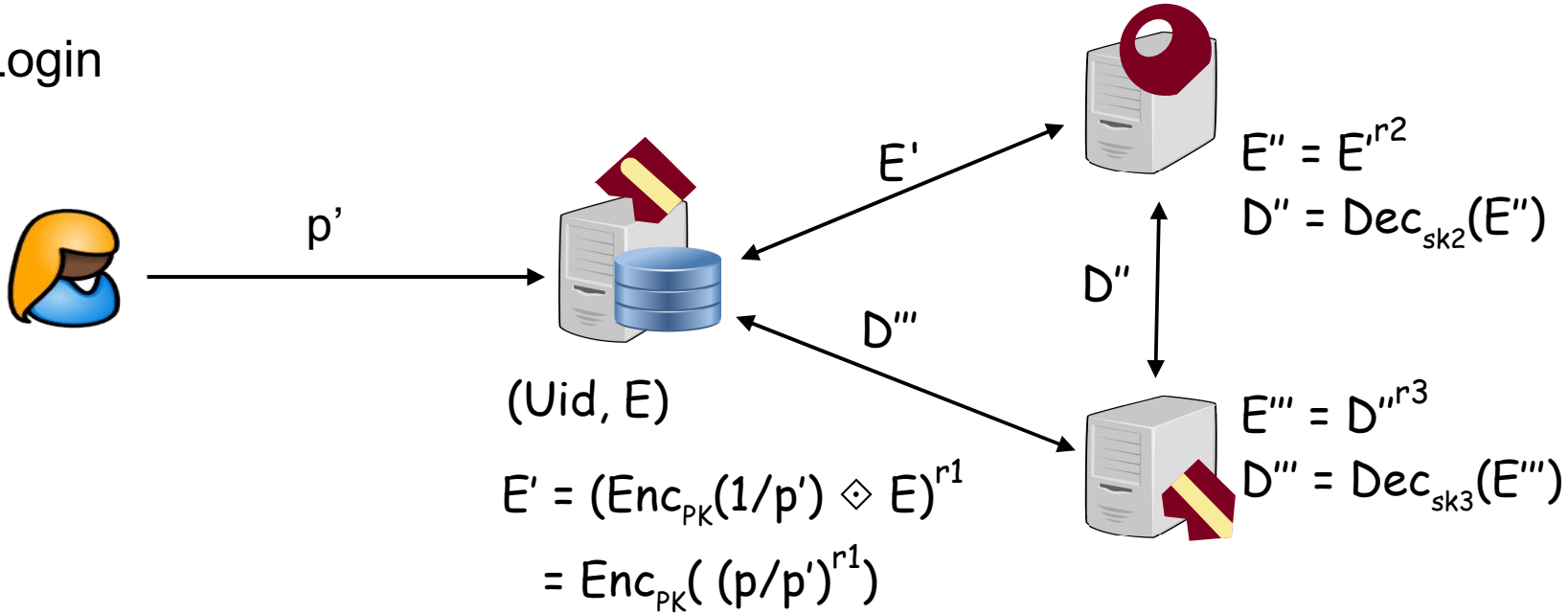
Login



# Dedicated MPC Protocol – Today: Getting Rid of Password Databases



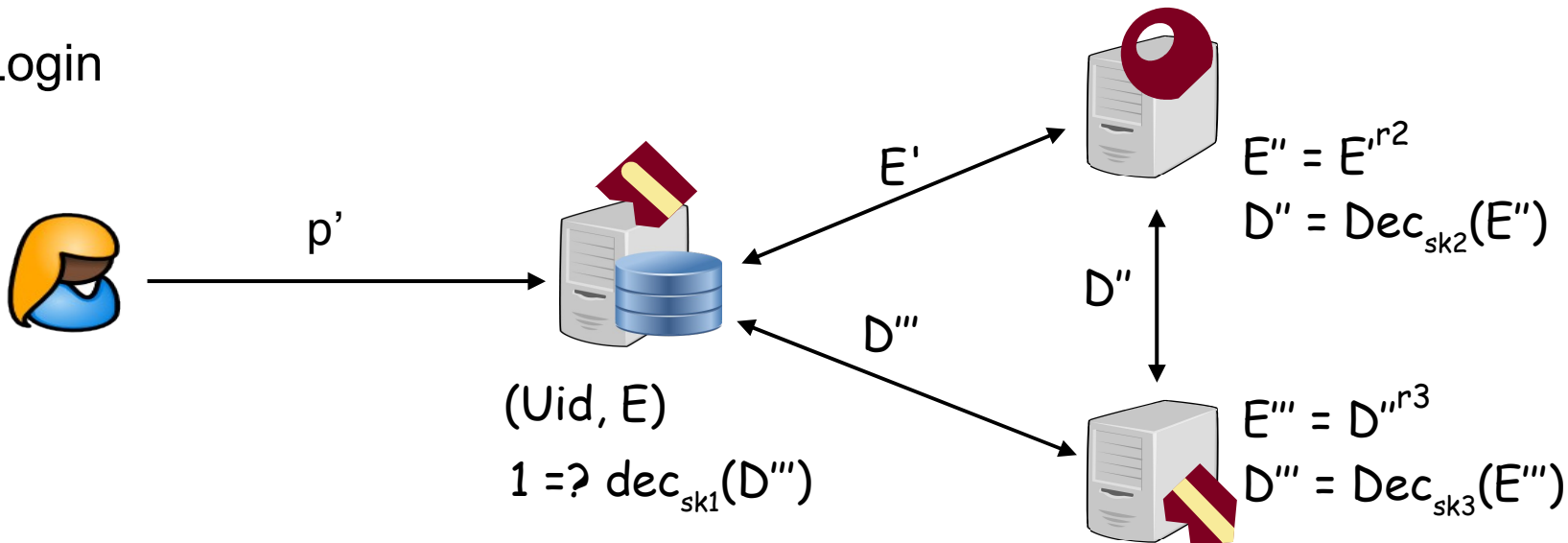
Login



# Dedicated MPC Protocol – Today: Getting Rid of Password Databases



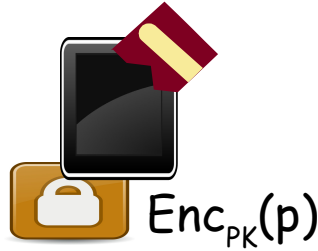
Login



- Result 1 if password match, random otherwise
  - With ElGamal, each server makes two exponentiations only
- Passwords safe as long as not *all* servers are hacked
  - off-line attacks no longer possible
  - on-line attacks can be throttled

# From password to cryptographic keys

[CLN12,CLLN14,CEN15]

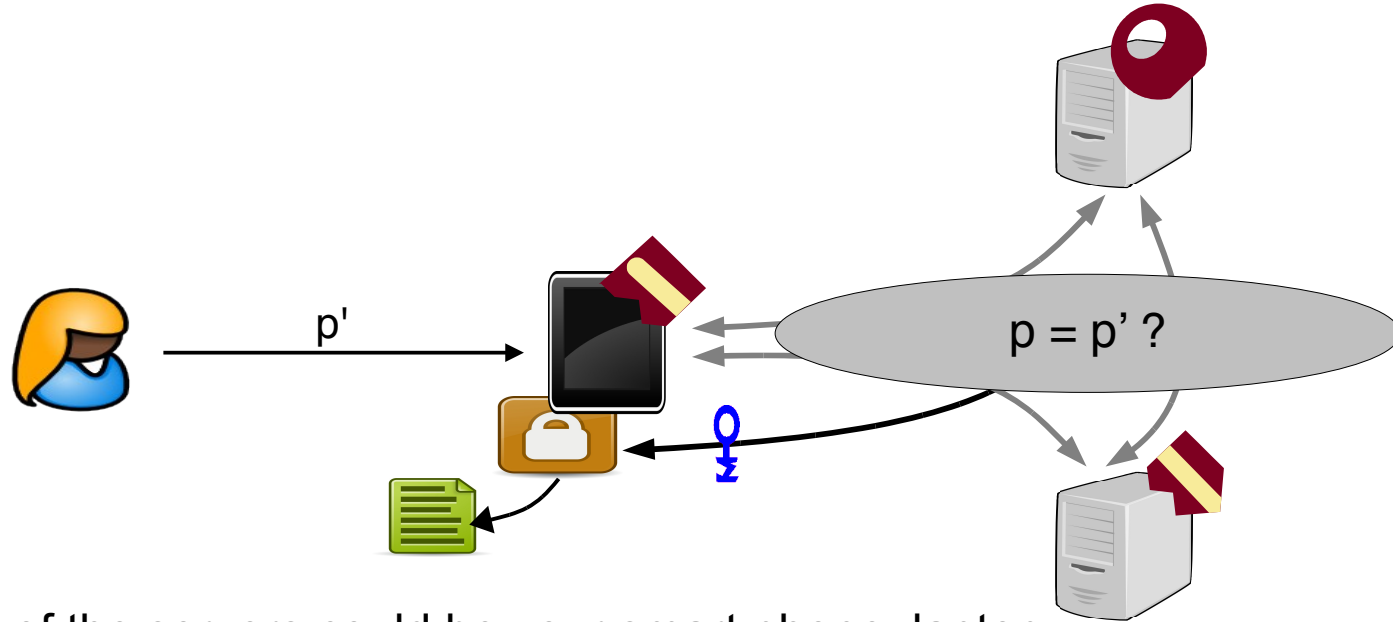


- One of the servers could be your smart phone, laptop, ...
- Get key share from if password check succeeded
- Decrypt all your files on phone (or stored in the cloud, etc)



# From password to cryptographic keys

[CLN12,CLLN14,CEN15]



- One of the servers could be your smart phone, laptop, ...
- Get key share from if password check succeeded
- Decrypt all your files on phone (or stored in the cloud, etc)



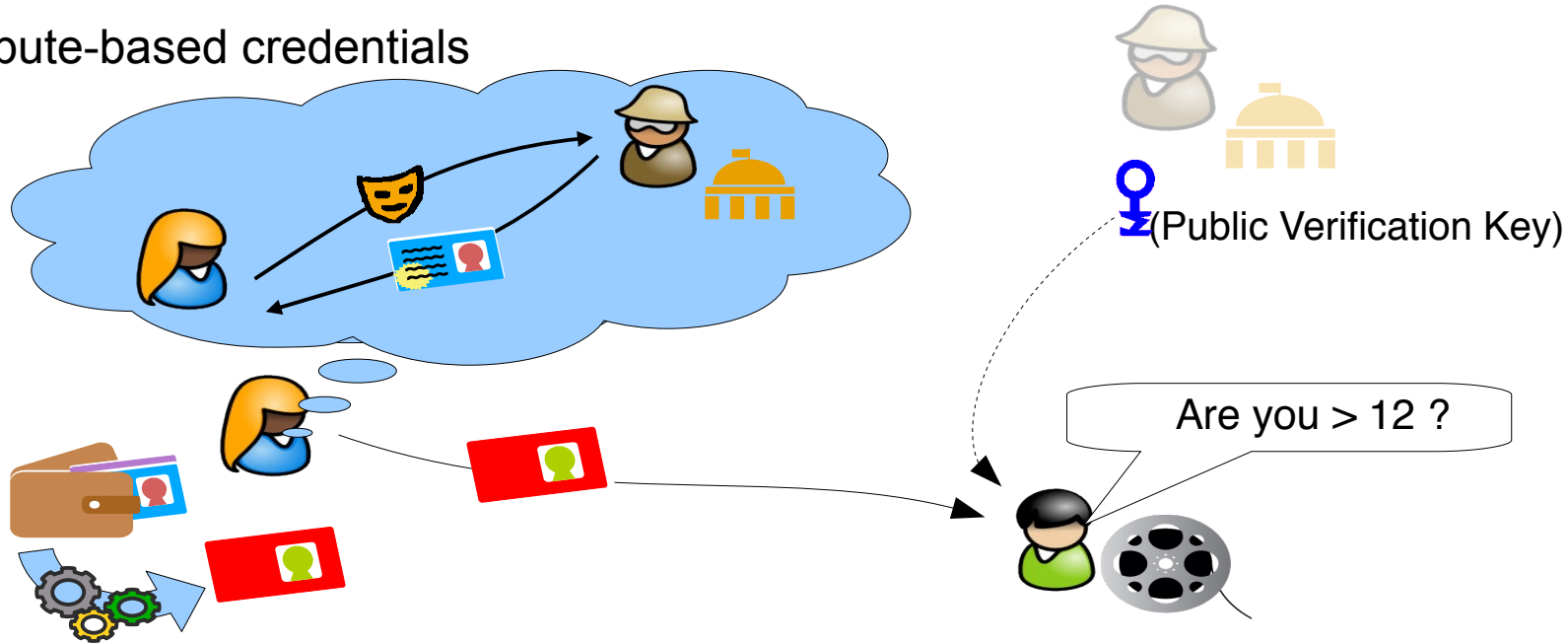
# Cryptography to the Aid

*an example of rocket science*

## Authentication without Identification

# Data Minimizing Authorization w/ ABCs

## Privacy Attribute-based credentials



- Service provider tells user what attribute are required
- User transforms credentials into a token with just these attributes
- Service provider can verify token w.r.t. issuers' verification keys

A stack of several books is positioned in the center of the frame, resting on a rough, textured surface that resembles stone or concrete. The books are of various thicknesses and colors, with some spines visible. The background is a mottled, greyish-blue texture. The text is overlaid on the image in a dark blue color.

# Cryptography to the Aid

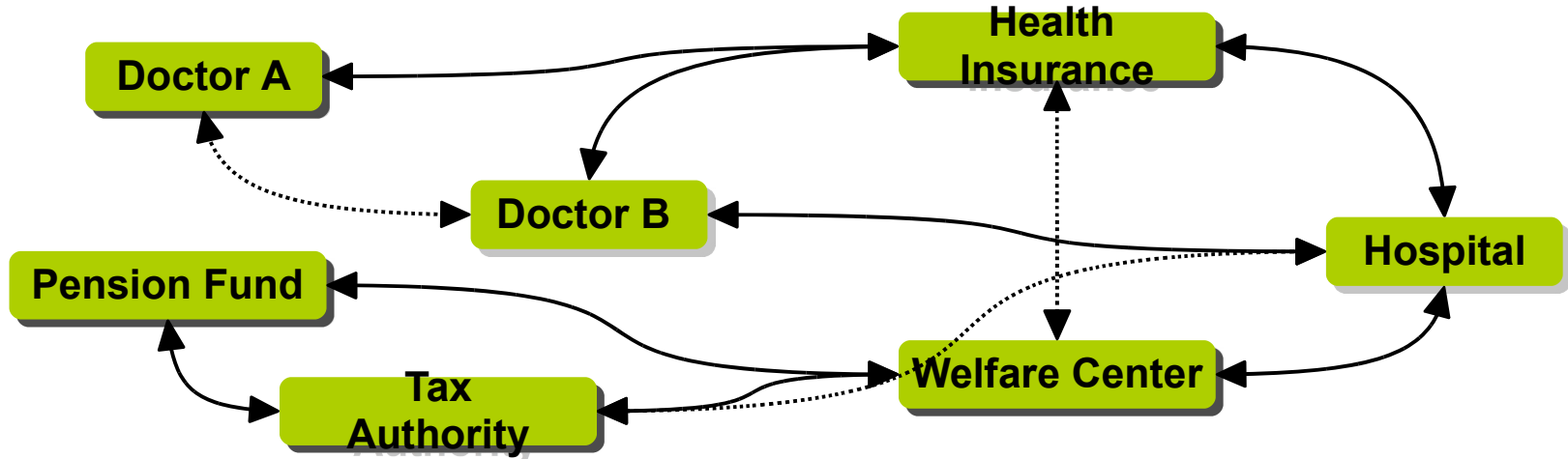
*an example of rocket science*

## Convertible Pseudonyms

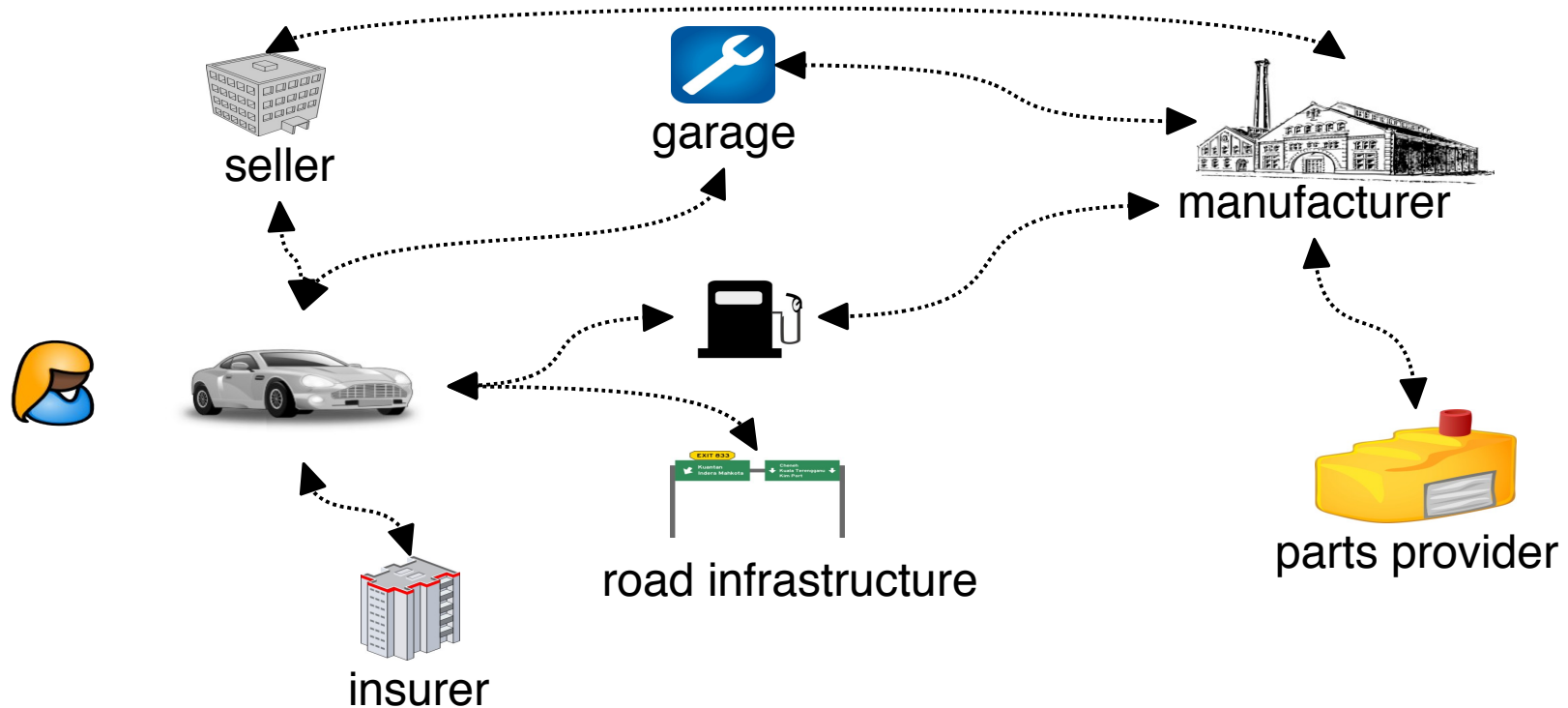
# How to maintain related yet distributed data?

## Example use case: social security system

- Different entities maintain data of citizens
- Eventually data needs to be exchanged or correlated



# IoT Use case – Car Example



Many other different use case: IoT, Industry 4.0, Home Appliances, Metering, ...

# Requirements

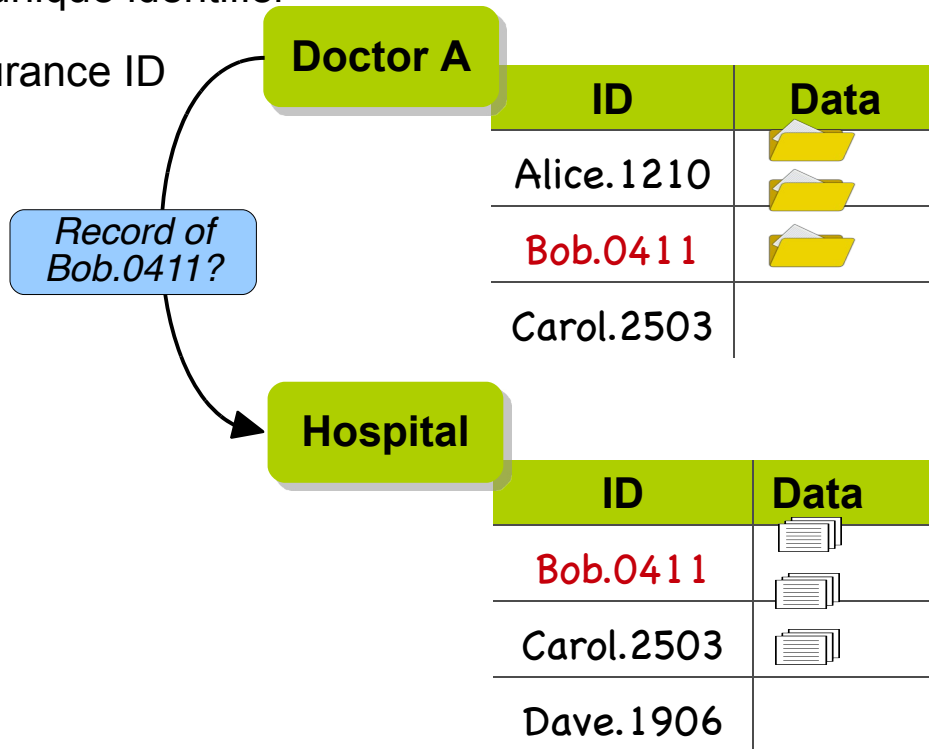
- Data originating from (or being related to) an individual
- Interactions with many different parties who share, exchange, and store data
- Data needs to be protected
  - Stored in encrypted form
  - Anonymized
  - Stored distributedly (different data base, different data controller)
  - User needs to be informed where data resides, how it is processed etc
- Still different parties want to use data
  - No too much anonymized, otherwise not usable anymore
  - If somewhat anonymized, how can user still keep track?

How can we do this?



# Globally Unique Identifier

- user data is associated with globally unique identifier
  - e.g., social security number, insurance ID
- different entities can easily share & link related data records

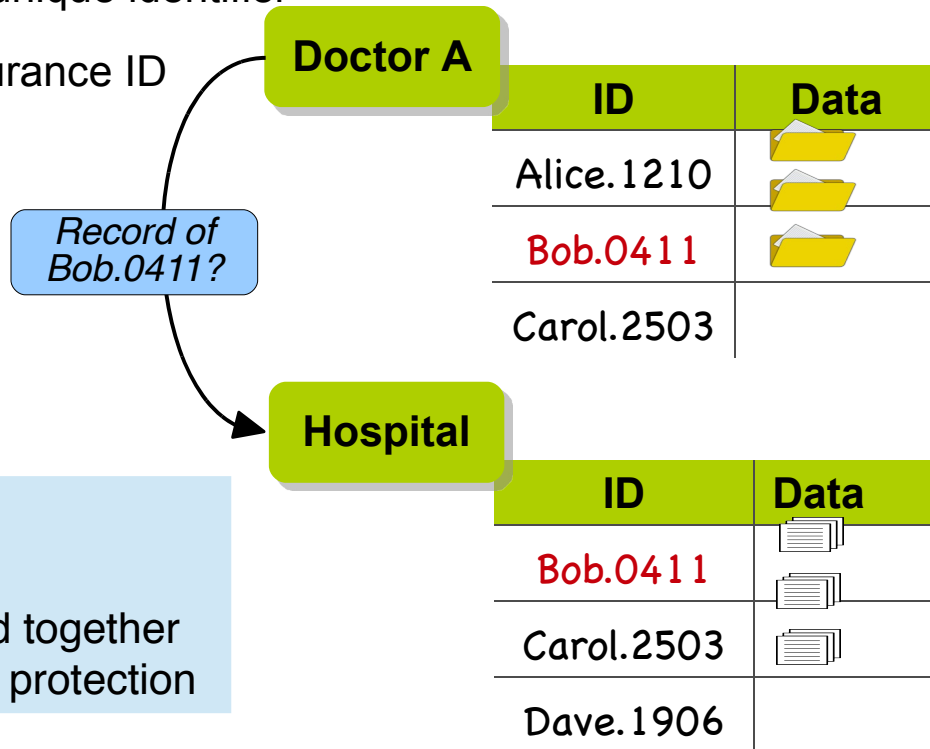




# Globally Unique Identifier

- user data is associated with globally unique identifier
  - e.g., social security number, insurance ID
- different entities can easily share & link related data records

- + simple data exchange
- no control about data exchange
- if records are lost, pieces can be linked together
- data has high-value → requires strong protection



# Using Privacy-ABCs to derive Identifiers



Doctor A

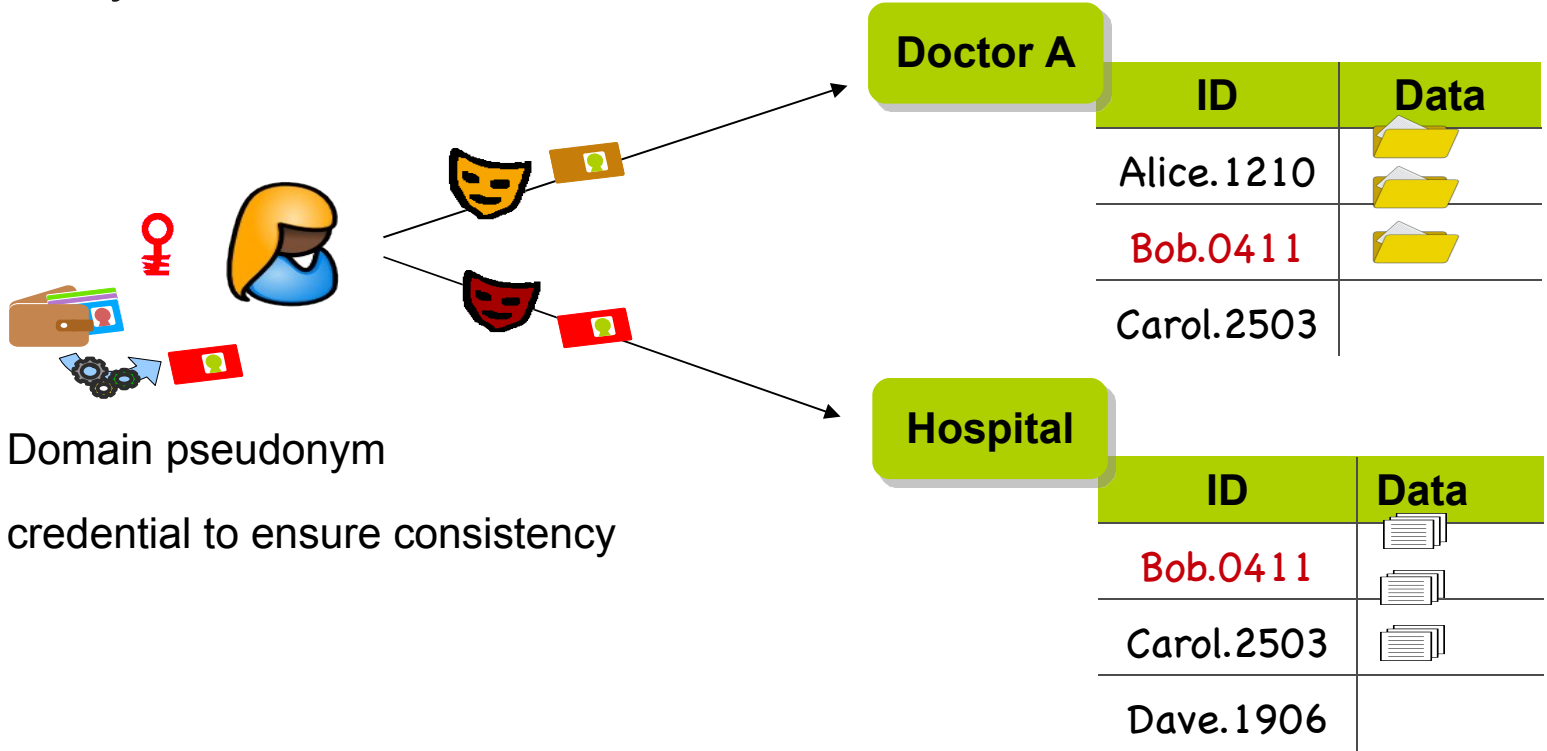
ID	Data
Alice.1210	
Bob.0411	
Carol.2503	

Hospital

ID	Data
Bob.0411	
Carol.2503	
Dave.1906	

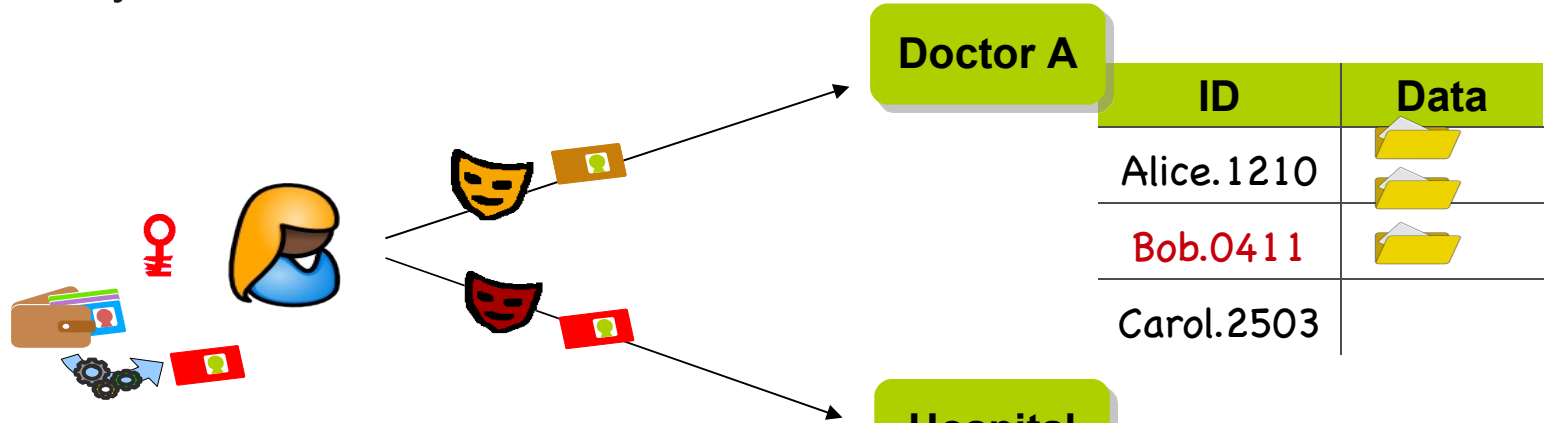
- Use Domain pseudonym

# Using Privacy-ABCs to derive Identifiers



- Use Domain pseudonym
- Use credential to ensure consistency

# Using Privacy-ABCs to derive Identifiers



- Use Domain pseudonym
- Use credential to ensure consistency
- Exchanging records via user and credentials

- data exchange needs to involve user
- + control about data exchange
- + lost records are cannot be linked together

## Local Pseudonyms & *Trusted* “Converter”




- central converter derives independent server-local identifiers from unique identifier
- user data is associated with (unlinkable) server-local identifiers *aka* “pseudonyms”
- only converter can link & convert pseudonyms

→ central hub for data exchange




### Converter

Main ID	ID-A	ID-H
Alice.1210	Hba02	7twnG
Bob.0411	P89dy	ML3m5
Carol.2503	912uj	sD7Ab
Dave.1906	5G3wx	y2B4m

### Doctor A

ID	Data
Hba02	
P89dy	
912uj	

### Hospital

ID	Data
ML3m5	
sD7Ab	
y2B4m	



# Local Pseudonyms & *Trusted* “Converter”

- central converter derives independent server-local identifiers from unique identifier
- user data is associated with (unlinkable) server-local identifiers *aka* “pseudonyms”
- only converter can link & convert pseudonyms

→ central hub for data exchange




Main ID	ID-A	ID-H
Alice.1210	Hba02	7twnG
Bob.0411	P89dy	ML3m5
Carol.2503	912uj	sD7Ab
Dave.1906	5G3wx	y2B4m

**Converter**




Record of P89dy  
from Hospital?

Record of  
ML3m5 ?

**Doctor A**

ID	Data
Hba02	
P89dy	
912uj	

**Hospital**

ID	Data
ML3m5	
sD7Ab	
y2B4m	

# Local Pseudonyms & *Trusted* “Converter”

- central converter derives independent server-local identifiers from unique identifier
- user data is associated with (unlinkable) server-local identifiers *aka* “pseudonyms”
- only converter can link & convert pseudonyms

→ central hub for data exchange




Main ID	ID-A	ID-H
Alice.1210	Hba02	7+wnG
Bob.0411	P89dy	ML3m5
Carol.2503	912uj	sD7Ab

**Converter**




Record of P89dy from Hospital?

Record of ML3m5 ?

**Doctor A**

ID	Data
Hba02	
P89dy	
912uj	

**Hospital**

ID	Data
ML3m5	
sD7Ab	
y2B4m	

Dev + control about data exchange

+ if records are lost, pieces cannot be linked together

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

# Local Pseudonyms & *Trusted* “Converter”

- central converter derives independent server-local identifiers from unique identifier
- user data is associated with (unlinkable) server-local identifiers *aka* “pseudonyms”
- only converter can link & convert pseudonyms

→ central hub for data exchange

Record of P89dy  
from Hospital?

Doctor A

ID

Data

How can we make the converter less trusted?

Main

Alice.12

Bob.04

Carol.2503

912uj

sD7Ab

David

+ control about data exchange

+ if records are lost, pieces cannot be linked together

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

Hospital

ID

Data

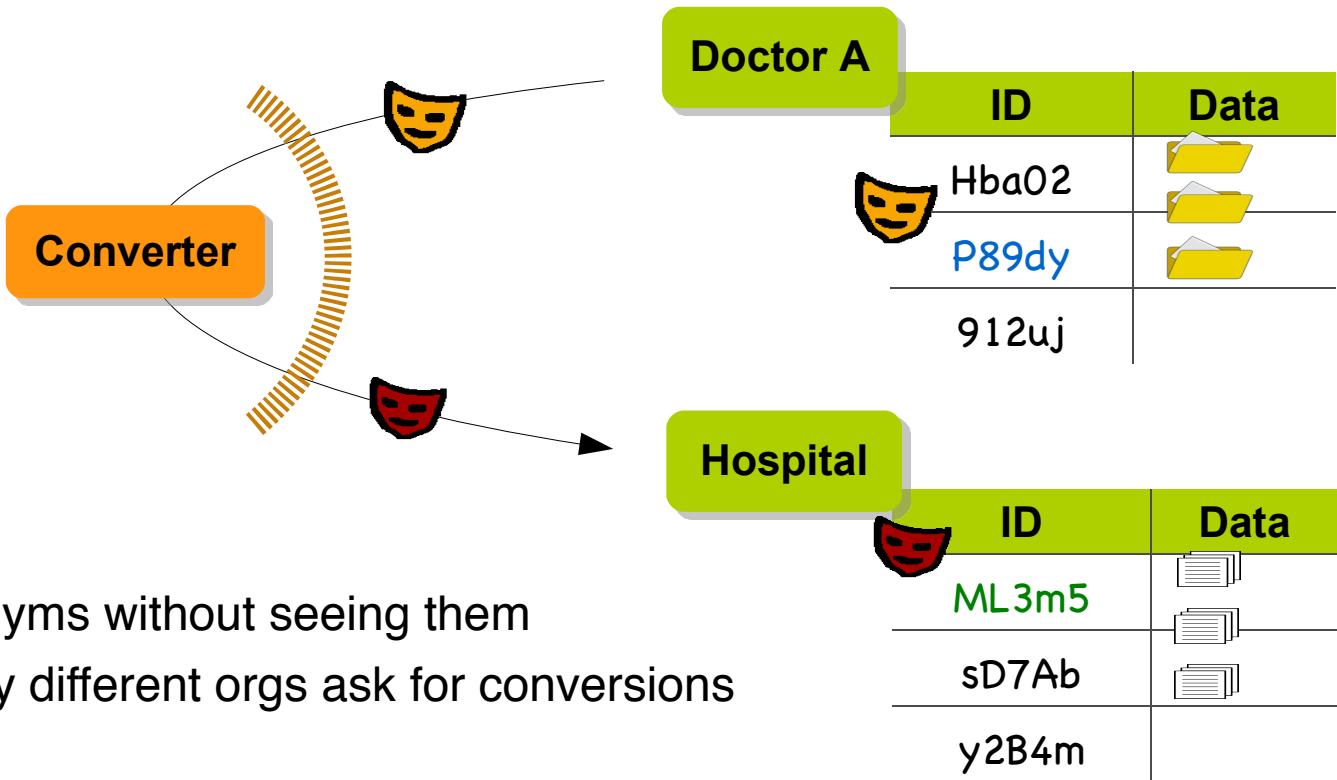
ML3m5

sD7Ab

y2B4m



# Blindly Translatable Pseudonyms



## Goal:

- Convert pseudonyms without seeing them
- Control frequency different orgs ask for conversions

# Blindly Translatable Pseudonyms [CL'15]

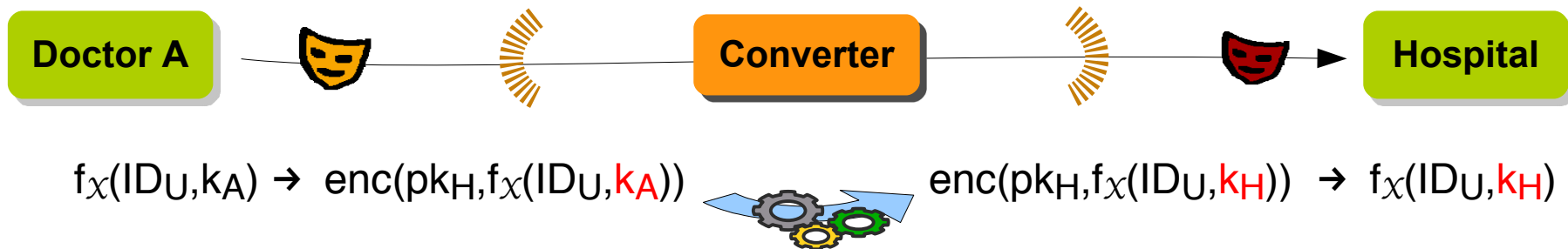
## Idea:

- Pseudonyms need to have mathematical relation

$$\text{nym}_{(U,A)} = f_{\chi}(\text{ID}_U, k_A)$$

- To convert:

- Doctor encrypts pseudonym under Hospital's encryption key
- Converter operates translation on encrypted pseudonyms  $\rightarrow$  homomorphic encryption



## Instantiation – Pseudonym Generation



converter  $\mathcal{X}$  and server  $S_A$  jointly to compute  $\text{nym}_{(U,A)} = f_{\mathcal{X}}(\text{ID}_U, k_A)$

1) compute global core identifier using secret key  $k$

$$z_U \leftarrow \text{PRF}(k, \text{ID}_U)$$

2) compute server-local pseudonym using server-specific secret key  $x_A$

$$\text{nym}_{U,A} \leftarrow z_U^{x_A}$$

$$\text{i.e., } f_{\mathcal{X}}(\text{ID}_U, k_A) = \text{PRF}(k, \text{ID}_U)^{x_A}$$

# Instantiation – Pseudonym Conversion

server  $S_A$  wishes to convert a pseudonym  $\text{nym}_{U,A}$  for server  $S_B$

$S_A$ 's input:  $\text{nym}_{U,A}$ ,  $\text{pk}_B$

**Converter  $\mathcal{X}$**

$k$ ,  $\text{sk}_{\mathcal{X}}$ , for each server:  $x_A$ ,  $x_B$ ,  $x_C$ , ...

**Server A**

$\text{sk}_A$

**Server B**

$\text{sk}_B$



# Instantiation – Pseudonym Conversion

server  $S_A$  wishes to convert a pseudonym  $\text{nym}_{U,A}$  for server  $S_B$

$S_A$ 's input:  $\text{nym}_{U,A}$ ,  $\text{pk}_B$

**Converter  $\chi$**

$k$ ,  $\text{sk}_\chi$ , for each server:  $x_A$ ,  $x_B$ ,  $x_C$ , ...

$$\text{nym}_{U,A} = z_U^{x_A}$$



$$\text{nym}_{U,B} = z_U^{x_B}$$

**Server A**

$\text{sk}_A$

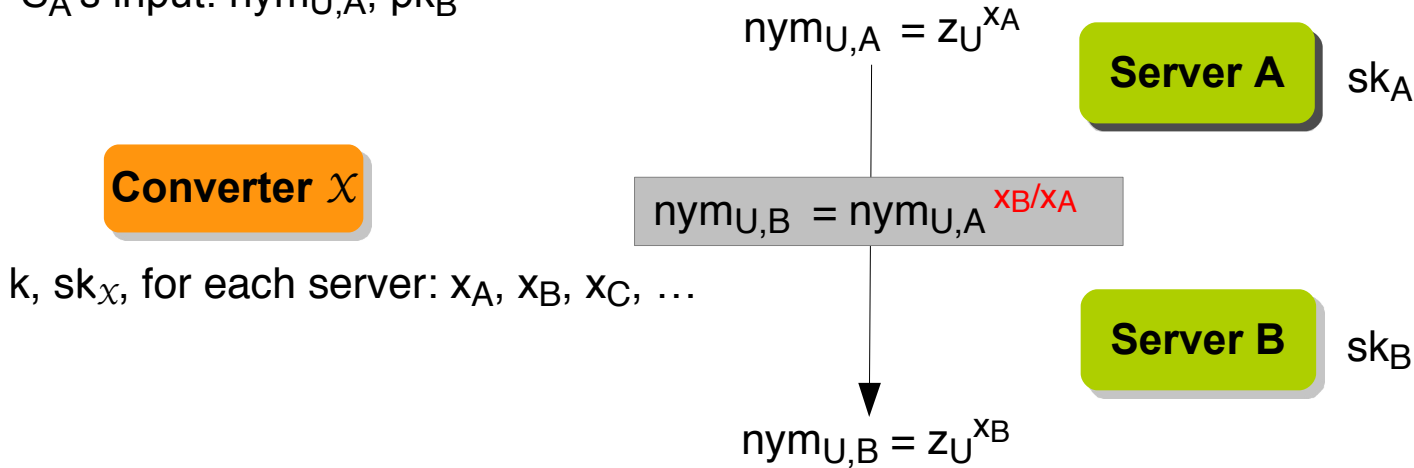
**Server B**

$\text{sk}_B$

# Instantiation – Pseudonym Conversion

server  $S_A$  wishes to convert a pseudonym  $\text{nym}_{U,A}$  for server  $S_B$

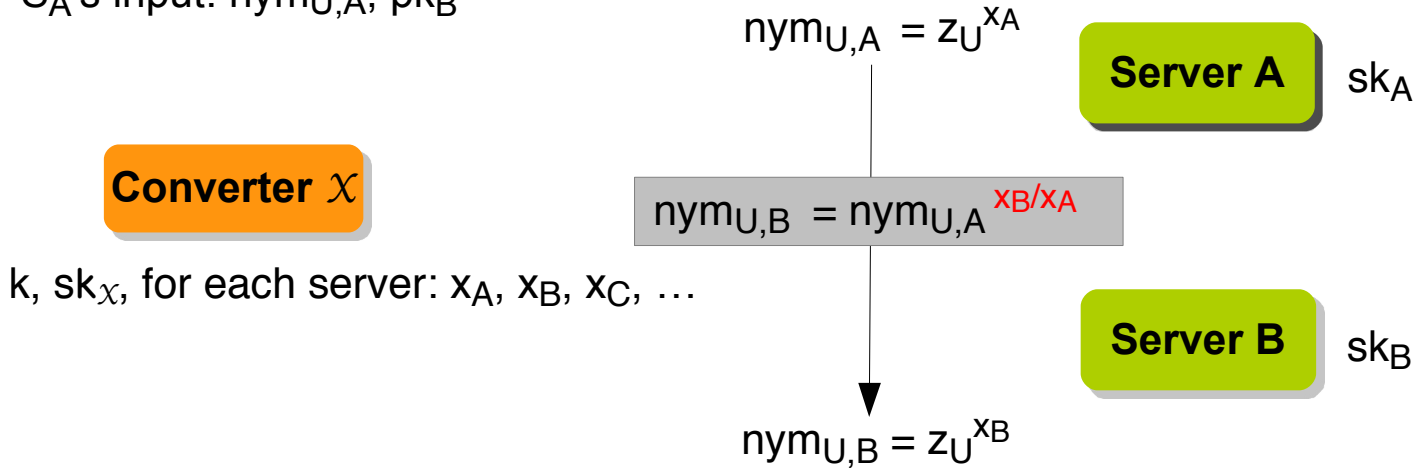
$S_A$ 's input:  $\text{nym}_{U,A}$ ,  $\text{pk}_B$



# Instantiation – Pseudonym Conversion

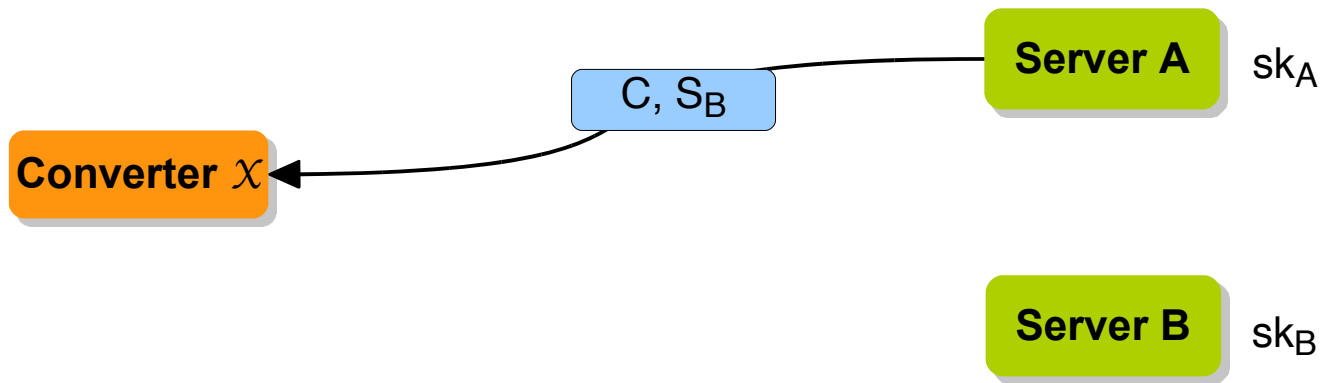
server  $S_A$  wishes to convert a pseudonym  $\text{nym}_{U,A}$  for server  $S_B$

$S_A$ 's input:  $\text{nym}_{U,A}$ ,  $\text{pk}_B$



## Instantiation – Pseudonym Conversion

$$C \leftarrow \text{Enc}(\text{pk}_\chi, (\text{Enc}(\text{pk}_B, \text{nym}_{U,A}))$$





# Instantiation – Pseudonym Conversion

**Converter  $\mathcal{X}$**

$$C' \leftarrow \text{Dec}(\text{sk}_{\mathcal{X}}, C)$$

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

$$\begin{aligned} C'' &= \text{Enc}(\text{pk}_B, \mathbf{nym}_{U,A})^{x_B/x_A} \\ &= \text{Enc}(\text{pk}_B, \mathbf{z}_U^{x_A})^{x_B/x_A} \\ &= \text{Enc}(\text{pk}_B, \mathbf{z}_U^{x_A \cdot x_B/x_A}) \\ &= \text{Enc}(\text{pk}_B, \mathbf{nym}_{iU,B}) \end{aligned}$$

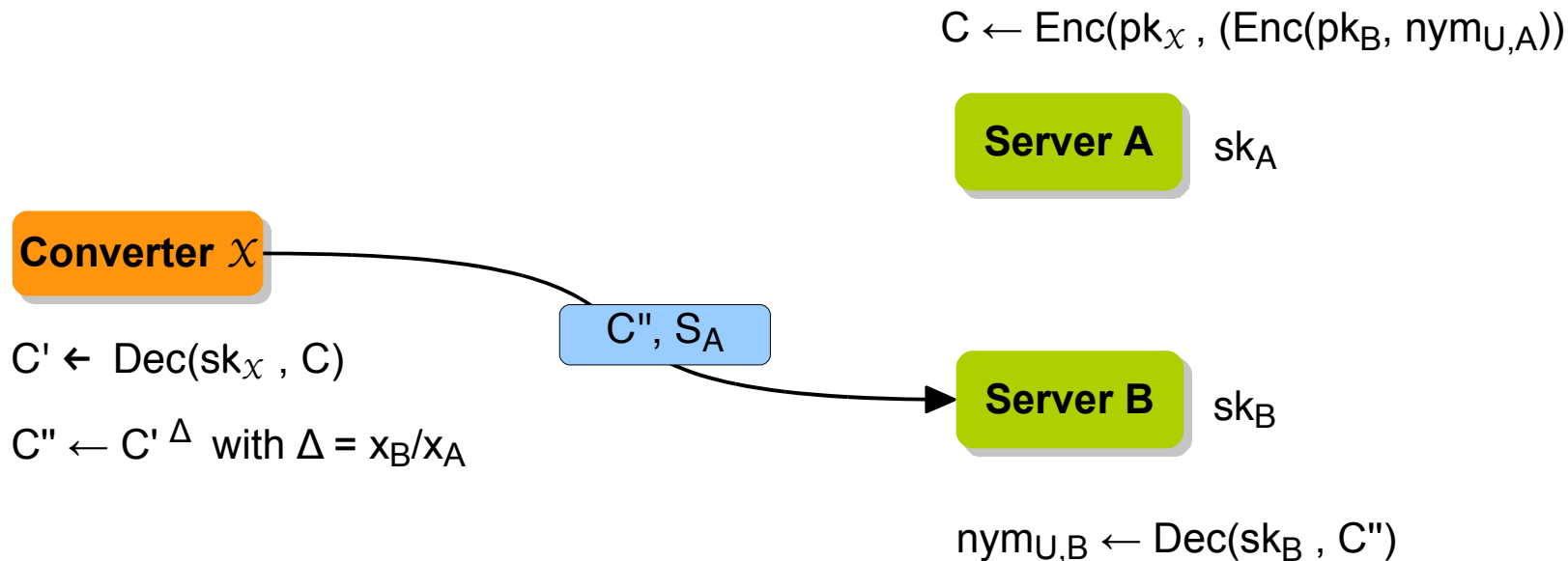
$$C \leftarrow \text{Enc}(\text{pk}_{\mathcal{X}}, (\text{Enc}(\text{pk}_B, \mathbf{nym}_{U,A})))$$

**Server A**  $\text{sk}_A$

**Server B**  $\text{sk}_B$



# Instantiation – Pseudonym Conversion



Still need to add proofs of correctness:

- 1) signatures on so that Server A can proof correct input
- 2) sign encrypted messages

Conclusion

# Further Research Needed!

## Provably secure protocols

- Properly modeling protocols (UC, realistic attacks models, ...)
- Verifiable security proofs
- Retaining efficiency

## Securing the infrastructure & IoT

- “ad-hoc” establishment of secure authentication and communication
- audit-ability & privacy (where is my information, crime traces)
- security services, e.g., better CA, oblivious TTPs, anon. routing, ...



# Further Work Needed!



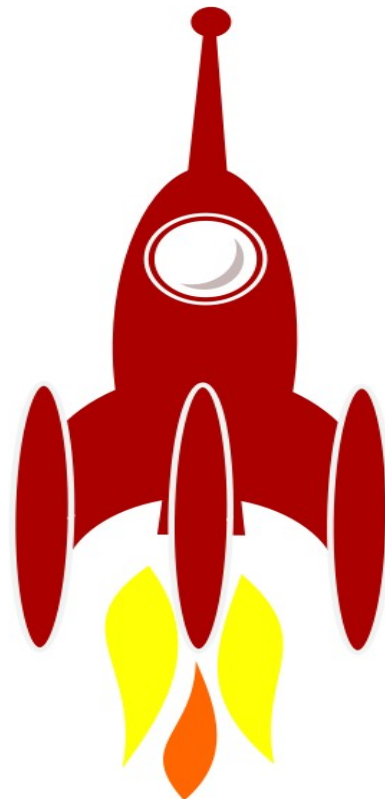
## Towards a secure information society

- Society gets shaped by quickly changing technology
- Consequences are hard to grasp yet
- We must inform and engage in a dialog

# Conclusion

- Much of the needed technology exists
- ... need to use them & build apps “for the moon”
- ... and make apps usable & secure for end users

Let engage in some rocket science!



# Thank you!

Joint work w/ Maria Dubovitskaya, Anja Lehmann,  
Anna Lysyanskaya, Gregory Neven, and many many more.

[jca@zurich.ibm.com](mailto:jca@zurich.ibm.com)    @JanCamenisch    [ibm.biz/jancamenisch](https://ibm.biz/jancamenisch)

