Microsoft Research

An introduction to
Privacy Technologies

George Danezis
Microsoft Research Cambridge
gdane@microsoft.com

University of Cambridge Computer Lab – Security II – 06 Feb 2012

Purpose of this lecture

- Key take home messages:
  - Processing personal data requires special protections.
  - Some of these protections are technical.
  - What privacy technologies can do to help.
    - Some illustrative technical constructions.

WARNING: DO NOT DO THIS AT HOME / WORK
This lecture gives only a flavour of privacy technologies. Specifying, designing and building security and privacy systems is a highly specialized task requiring years of study and experience. If you are asked to do any of the above without such preparation just say "no".

What is Privacy?

- The need for user confidentiality
  - Users do not wish their data or data about themselves to be collected, used or shared with others without their consent.
  - Threats: communications, storage
- The need for user control
  - User want to have control of their informational environments.
  - This includes the ability to share or not share.
  - It also includes the ability to use their data for their own purposes, on devices of their choice, be free from monopolies & lock-in, and have visibility on how others use their data.
  - Threats: services misusing data, loss of data, abuse
- The need to comply to regulations
  - EU, Canada & others: 8 Data protection principles
  - US: Sectorial regulations (HIPPA = Health, Video, libraries, …)
  - Disclosure requirements.
  - Threats: company reputation, non-compliance / legal, compulsion

Revision: data protection principles

Personal Data: any data item that could be linked to an individual.

Personal Data must be:

1. Processed fairly and lawfully (i.e. with **consent** / respect for sensitive data).
2. Obtained for specific and lawful purposes.
3. Adequate, relevant and not excessive. (**Minimization**)
4. Accurate and up-to-date. (**Modify**)
5. Not kept longer than necessary. (**Deletion**)
6. Process according to subject's rights. (**Access**)
7. Securely kept.
8. Not transferred where it would be less protected.

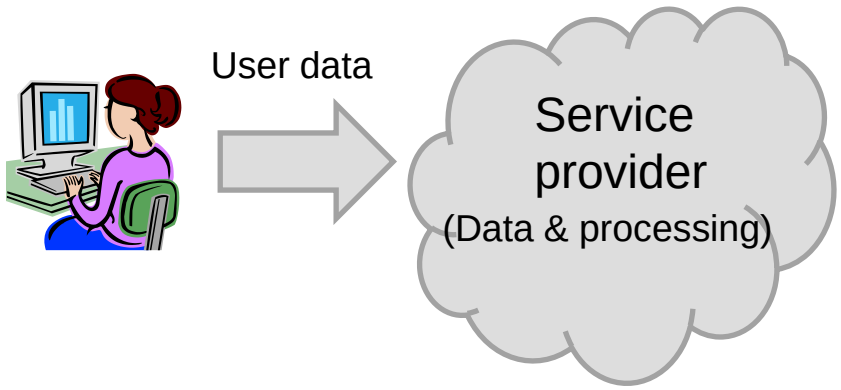
Soft & Hard Privacy Technologies

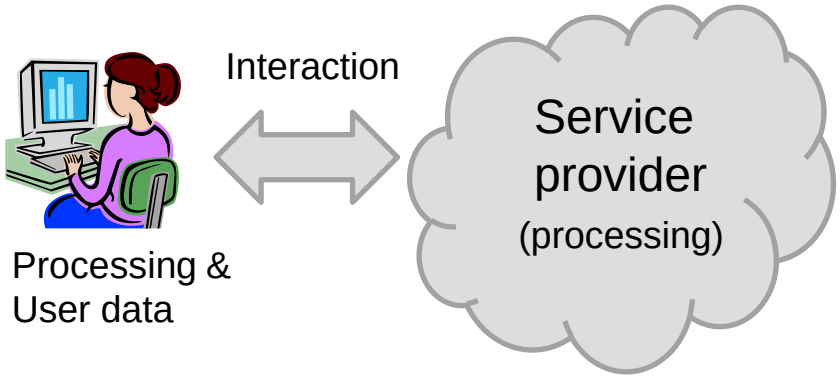
Soft – Trust 3rd Party

- User gives personal information to 3rd party, and trusts it will be processed according to DP.
- Example: Search engine

Hard – Data Minimization

- User only provides the absolutely minimum amount of information to receive a service or complete an interaction.
- Example: encrypted cloud storage.





Challenges (1)

- Challenges for soft privacy approach:
  - How can users be sure that data is processed in accordance to a privacy policy?
  - How can such processing be supported by technologies (think access / backup)?
  - How to make sure that a single rogue employee cannot get access to all data?
  - Limits: “lawful” access and “legal” compulsion.

Challenges (2)

- Challenges for hard privacy approach:
  - If service provider cannot see all data how can they provide good service?
  - How to prevent users from lying and cheating?
  - Do users have to be on-line / understand technology?
  - How to deal with abuses of anonymity privacy?

- Technology plays a key role in supporting both approaches!

Soft Privacy tools

- What architecture & privacy policy?
  - Example architectures & requirements
- How to enforce the privacy policy, know it is doing the right thing?
  - Policy language example: SecPAL.
- How to anonymize query results securely?
  - Differential privacy

Microsoft Research

Soft privacy insecure architecture

Data is stored and copied across whole organization.

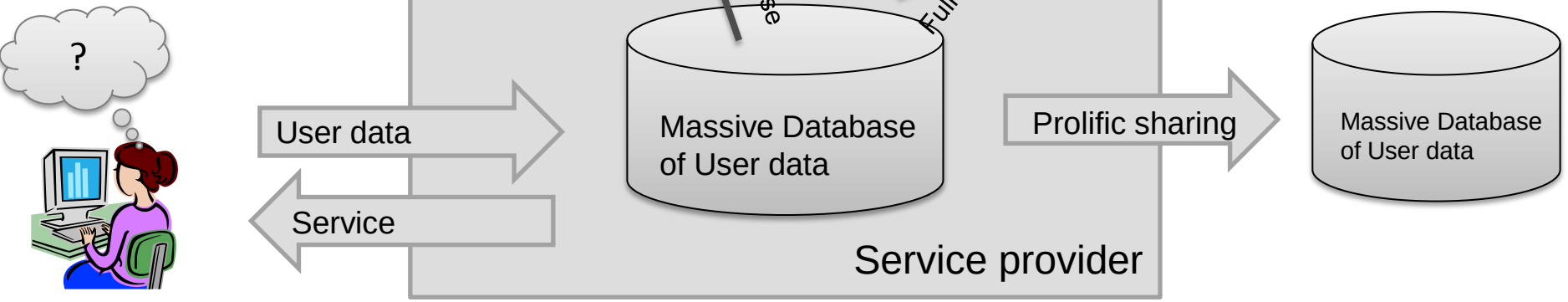
Everyone has access to everything for any reason.

No policy just convention.

Give data wholesale to outside entities.

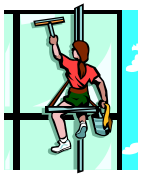
No logging.

Poor user control.



Dog

Audit



Contractors

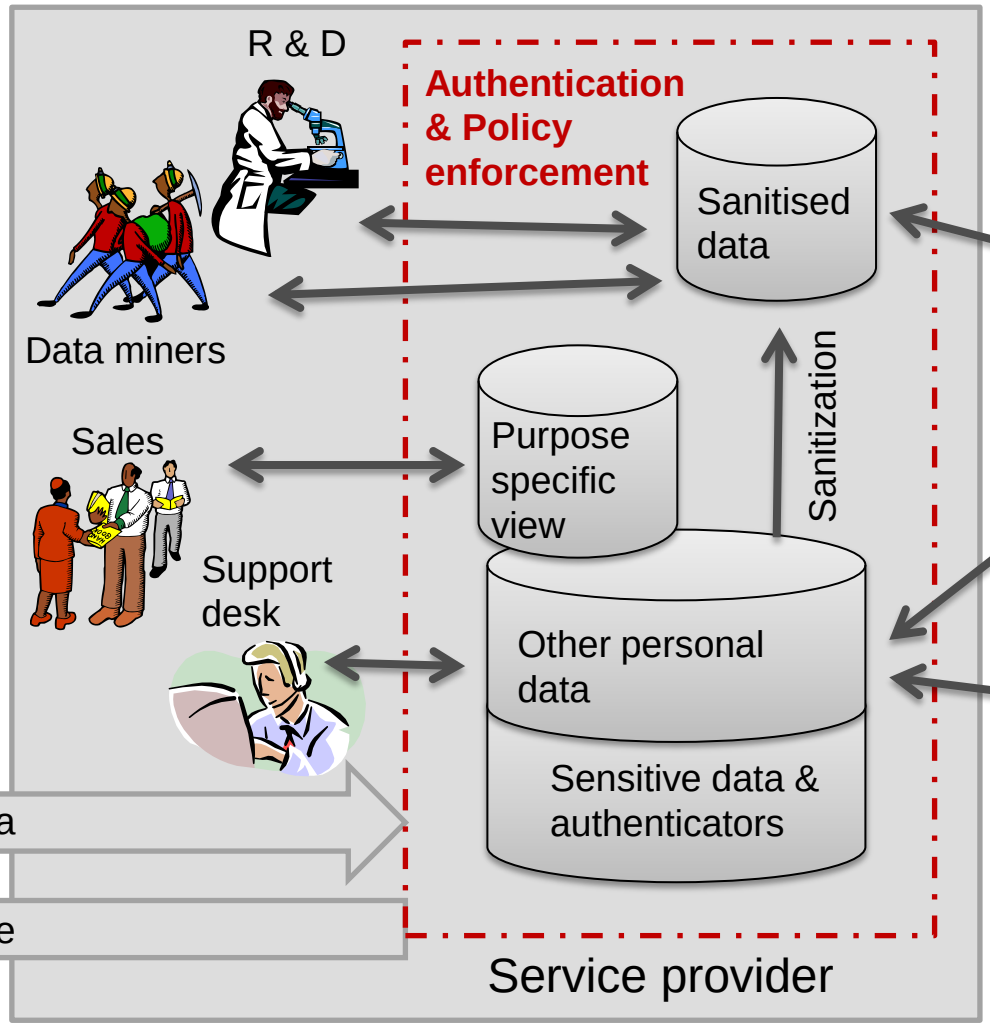
Microsoft Research

Soft privacy architecture

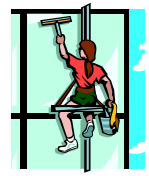
Store data in different compartments according to sensitivity.

Protect all data access through authentication, policy application & logging.

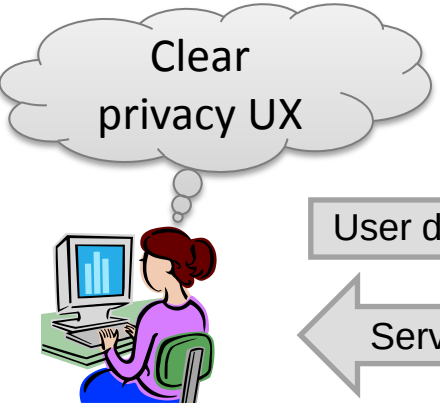
Sanitise & restrict views when possible.



Audit



Contractors



Formal Policy Languages

- Engineering Problem:
  - How do you express & implement the authorization / privacy policy?
- Option 1: Embed in the code
  - Paper document & series of checks when data is accessed
  - Eg. If (user.role == doctor and patient.doctor == user) { … process patient data … }
  - Error prone, difficult to verify compliance, unmaintainable.

Formal Policy Languages

- Option 2: Separate policy from code
  - Call a formal policy engine to perform high level checks
  - Eg. If(access(user, patient)) { … process …}
  - Language support: tools can automatically detect whether a resource is used without appropriate checks (guard methods).
  - Flexibility: policy can be changed later with no code modification.
  - Deal with delegation!

SecPAL example

Mary (Doctor)

Delegation:

Mary says access(Jack, Pete)

Now: access(Jack, Pete)

Jack (Nurse)

Jack says access(Dog, Pete)

Pete (Patient)

Dog

Policy:

Rules:

DB says access(X, P) if doctor(X) and care (X,P)

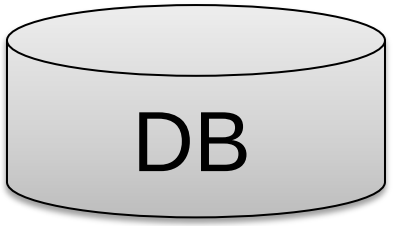
DB says X cansay access(Y, P) if doctor(X) and access(X, P)

Facts (no free variables):

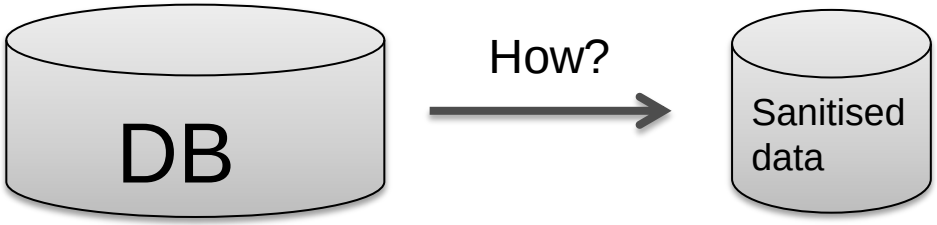
DB says doctor(Mary)

DB says care (Mary, Pete)

Authenticate U &
Require "DB says access(U,P)"



Sanitisation of personal data



- Can you in general anonymize a dataset of personal data?
  - Remove identifiers, generalize or add noise to entries.
  - So that records cannot be linked back to individuals?
  - Even by entities with arbitrary side / background information?
  - Answer: in general NO!
  - Therefore: even sanitised datasets subject to the privacy / access policy.

- What about answering aggregate queries on personal data?
  - Answer queries without leaking (too much) information about an individual.
  - Eg. What is the average height of people in this room? What is the median salary of a lecturer?
  - Naïve approaches are insecure: e.g. ensure at least K people in the population.
  - Answer: YES! (Differential Privacy)

Differential Privacy Intuition

- A (too) strong definition of privacy:
  - A statistic leaks no information about an individual record.
  - Impossible: A study on the average height of humans reveals information about my height, even if my height is not included in the study.

- Differential Privacy definition:
  - The probability of a result should be “close” to the probability of getting the same result from a database not containing an arbitrary record.
  - I.e. Learning results of queries should not leak “much” information about an individual record.
  - Strong: No matter what side information is available!

Differential Privacy Formalism

- Formal definition:
  - Consider 2 databases D and D', where D' and D are different in at most one element.
  - Consider a randomized algorithm f(.) that takes a database and calculates a result.
  - We say that f is "differentially private" if for any D, D' it holds that:

$$\Pr[\, f(D) \,] \leq e^{\varepsilon} \Pr[\, f(D') \,]$$

  - Where $\varepsilon << 1$ is a security parameter. (If ε is small, then $e^{\varepsilon} \approx 1 + \varepsilon$)
  - Composition: 2 DP queries with ε = DP with 2ε
    - I.e. there is a "Privacy budget".

Distribution of f(D) is similar to distribution of f(D')

Concrete example: Laplace mechanism

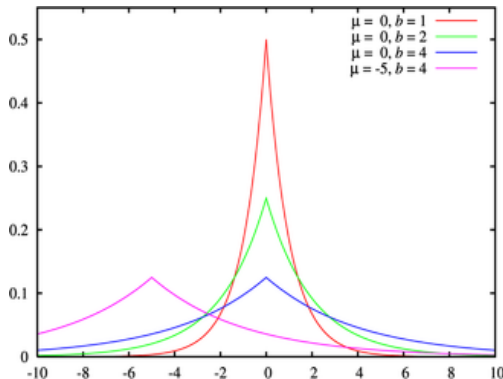
Concrete voting problem:

How many people like chocolate on average?
(Each answer with 0 / 1).

- Privacy concern: What if we all / none do?

- Function f = mean(D) + n (noise)
- How much noise?
  - Sensitivity of f: $\Delta f = \max(f(D) - f(D'))$ for any D, D'
  - I.e. for our example f, $\Delta f = 1$
  - Noise from Laplace distribution:

$$Pr[n] = \exp(-|n| \, \varepsilon / \Delta f)$$



Hard Privacy

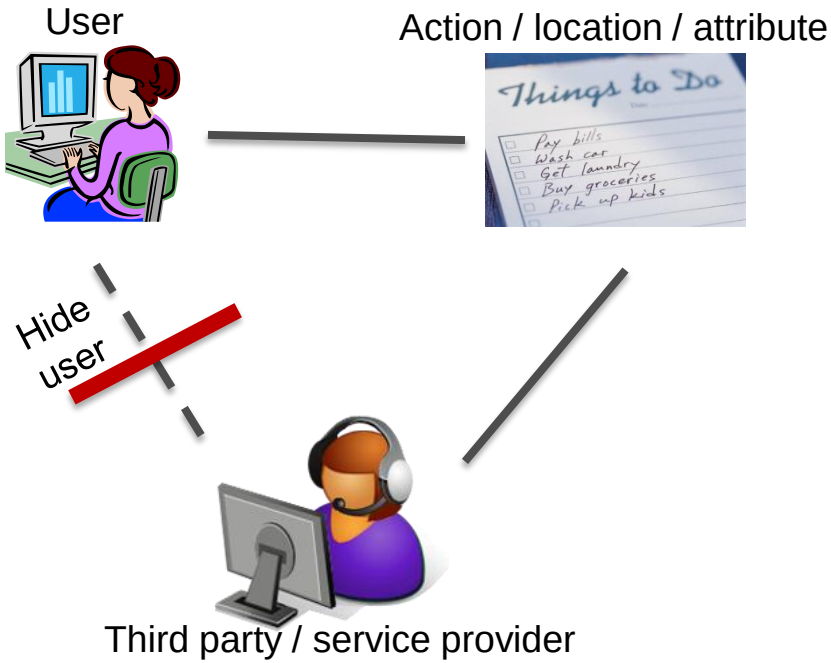
- No single entity can violate the privacy / security policy of the user.

- How?
  - Apply modern cryptography & data security techniques
  - Involve user devices

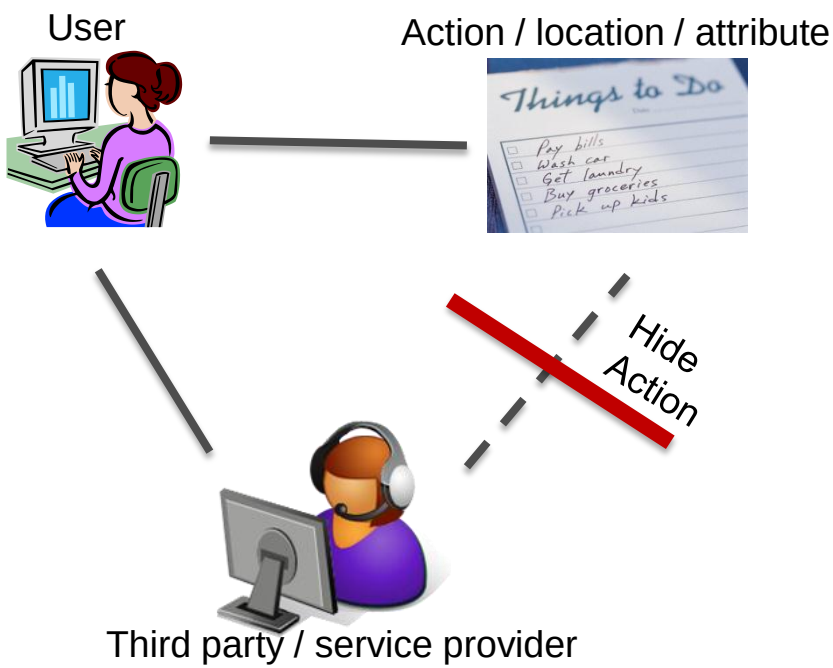
Microsoft Research

Two Hard Privacy architectures

Hide Identity



Hide Action



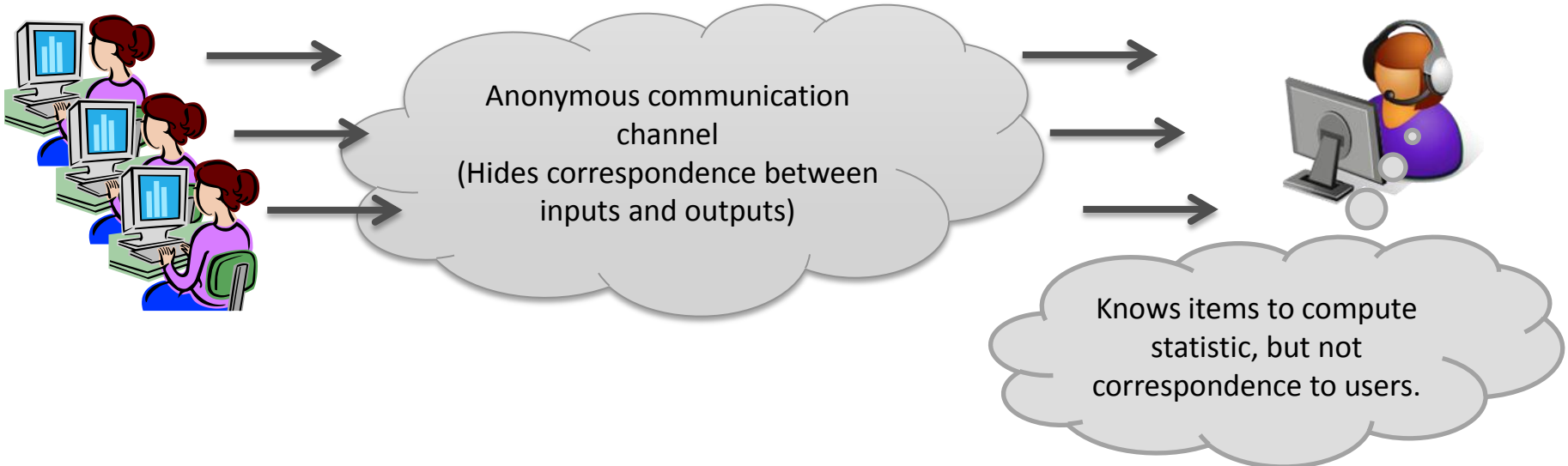
Two Example Protocols

- Aim of protocol:
  - Need to collect statistics from a population of k users (eg. mean, histogram)
  - Each user has a data item di.
  - BUT we no-one should know the link between user and their data item.

- 2 protocols:
  - Use an anonymous channel to hide identity.
  - Use a joint computation to hide di.

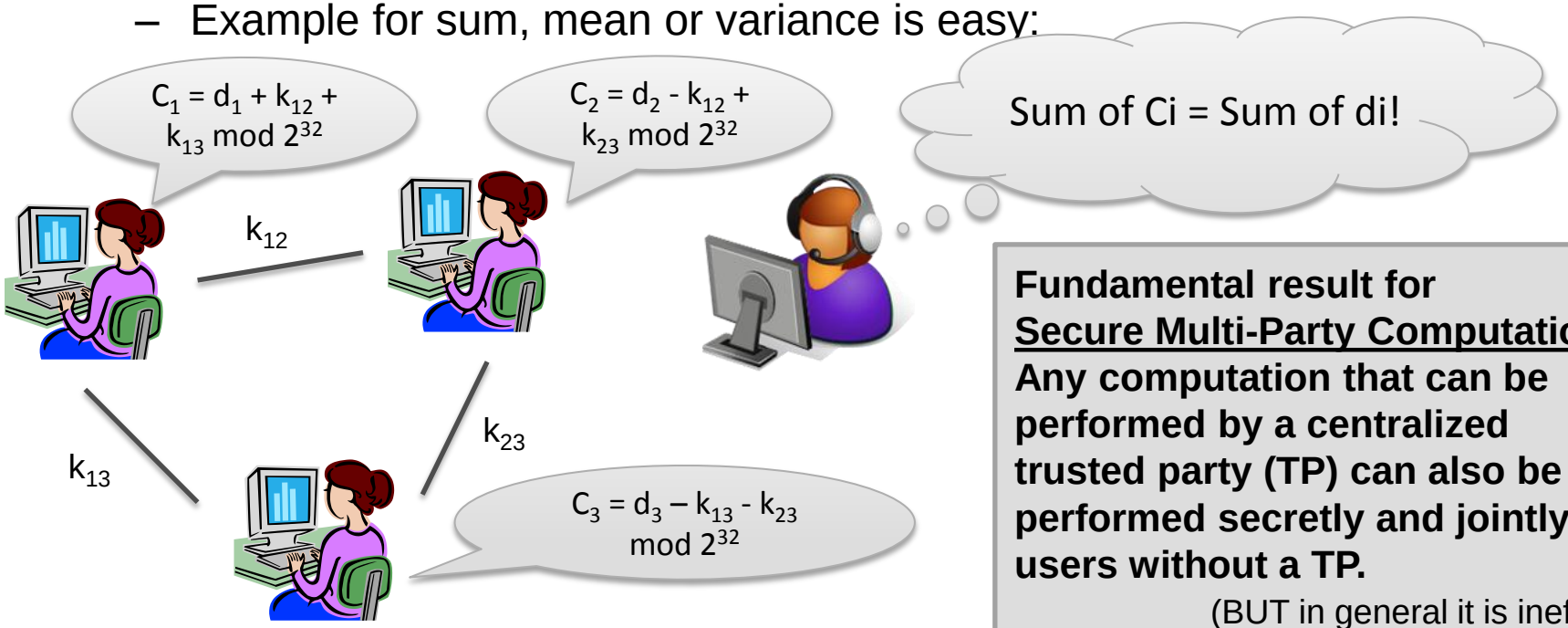
Using anonymous communications

- Protocol 1:
  - Each user sends their item di through an anonymous channel to an authority.
  - The authority collects the data items and computes the statistic.



Using multi-party computation

- Protocol 2:
  - Each user shares a secret key with all other users (using standard key exchange)
  - They jointly perform a multi-party computation to compute the statistic.
  - Example for sum, mean or variance is easy:



Fundamental result for <u>Secure Multi-Party Computation:</u>
Any computation that can be performed by a centralized trusted party (TP) can also be performed secretly and jointly by users without a TP.

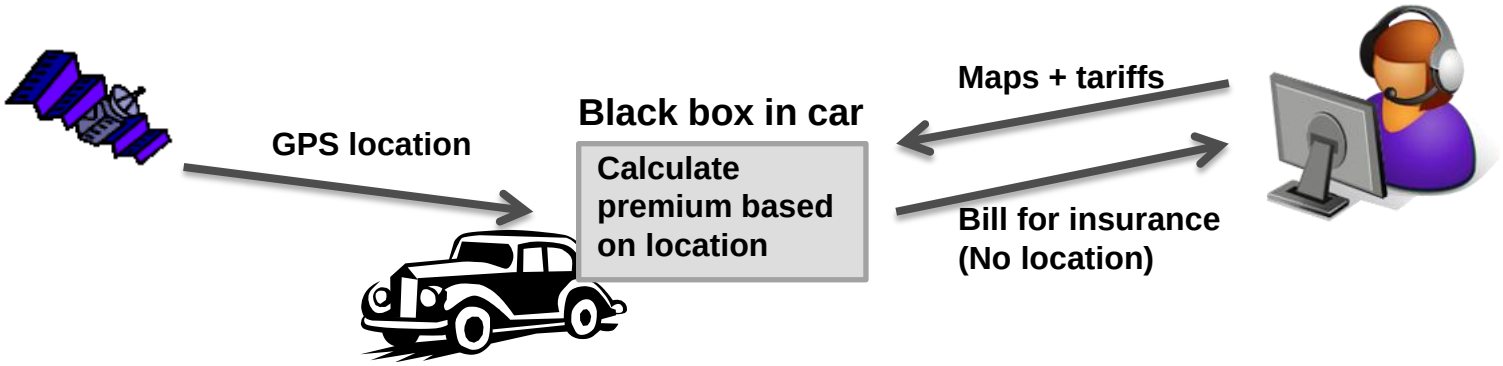
(BUT in general it is inefficient)

Preventing abuse

- Privacy alone is easy! – Privacy + integrity is hard!
- Examples:
  - **Publishing:** allow anonymous publishing to wikipedia BUT is a users received 10 negative votes they should be barred from contributing another article.
  - **Petitions:** allow people living in a specific region only to sign a petition only once.
  - **E-cash:** hide identity of payer, but make sure that they have a valid coin and spend it only once.
- Strategies:
  - Trusted parties that see everything (soft privacy).
  - Trusted hardware by all entities.
  - Cryptography: zero-knowledge proofs of knowledge.

Trusted hardware approach

- The service provider gives user a piece of hardware that performs computations correctly, and does not leak private information.
- Example: Pay-as-you-drive insurance



- Other example: smart-card that holds tokens for payment.
- Problems:
  - Cost of deploying hardware.
  - How to make sure user will not open device and subvert it.
  - Versatility?

Cryptographic approach

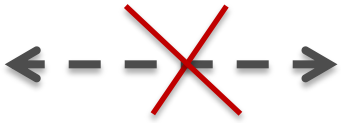
- Example: selective disclosure credentials
  - Alice needs to prove she is over 18 to get a drink at the bar, without revealing anything else about her identity.

Government

Issuing: get a credential that asserts Alice's age.

(How: Re-randomizable signatures)

Unlinkable!



Alice

"Showing": prove that Alice has a credential and that age > 18

Verify claim and statement

(How: Zero-knowledge proofs of knowledge)

Zero-knowledge proofs

- Key element in privacy friendly authentication, private computation and electronic cash.
  - Example: Alice needs to prove that (DATE – DOB > 18).
  - Prove a statement without leaking any other information.

- Schnorr proof :
  - Proof of knowledge of a discrete logarithm.
  - Identification scheme & signature scheme
  - Simplest zero-knowledge proof

- Key result concerning Zero-Knowledge proofs:
  - You can prove any computable relation between hidden / public quantities.
  - Of course some are more efficient than others.

Schnorr protocol

- Aim:
  - Alice wants to prove to Bob that she knows an x such that $C = g^x \bmod p$ for a given C, g, and prime p.
  - Note: finding such an x is hard given only C, g, p.

- Protocol:

Alice

1. Chose a random nonce w
2. Send $C_w = g^w \bmod p$ to Bob

3. Send a random challenge c to Alice

Bob (Bar)

4. Compute and send response $r = w - cx$

5. Verify: $C_w = g^r (g^x)^c \bmod p$

How does that work?

- Does the verification work?

$$C_w = g^r (g^x)^c \bmod p = g^{w-cx}\, g^{cx} = g^w$$

- Does it guarantee Alice knows x?
  - Alice cannot predict challenge c.
  - If she is able to answer with at least 2 different challenges she can extract the secret.
  - i.e. Solve $r_1 = w - c_1 x$ and $r_2 = w - c_2 x$ for x

- How do we know it leaks no other information?
  - Intuition: Bob can produce triplets (C_w, c, r) that pass the verification. How? Start with c, r then set C_w.
  - Thus only causation allows certainty – 1 bit leakage.
  - Simulation proof.

Key concepts

- Once you process data about individuals you need to afford them special protection.
- Key regulations on private data.
- Soft privacy vs. hard privacy.
- Policy languages can help manage access
- Differential privacy to publish results.
- Hiding identity vs. Hiding actions
- SMPC and ZK Proofs are powerful but costly.
- You can and must maintain privacy & prevent abuse.

Microsoft Research

Additional material

Example Technical Requirements

CRUD with privacy in mind:

- C1 – User consent for collection & processing of personal information (PII)
- R1 – User ability to view all PII we hold
- U1 – User ability to update PII we hold
- D1 – User ability to delete PII we hold

And some more:

- M1 – Minimize the data you collect, process or store
- I1 – Deep integration of privacy awareness and controls into UX.
- S1 – Secure storage and access to PII.
- A1 – Ability to anonymize / sanitise PII data views.
- D2 – Safely decommission service or app.

Reporting:

- R2/L1 – Logging of processing on PII.
- X1 – Lawful access provisions and logging.

Re-randomizable signatures

- Motivation:
  - An authority needs to sign some of your attributes (age, location, bank balance).
  - You want to show this signature to prove you have some credential (integrity)
  - BUT you do not want any bit-string to link the credential to the original signature (privacy).
- Special signature schemes (e.g. CL signatures):
  - Allow for signing, re-randomization, proving possession of a signature, and verifying possession of a signature.
  - (We will not go into details of how they are constructed – RSA and zero-knowledge proof under the hood)