Privacy protocols

Jason Castiglione* Dusko Pavlovic†
Peter-Michael Seidel
Email: {jcastig, dusko, pseidel}@hawaii.edu
University of Hawaii, Honolulu HI, USA

Abstract

Security protocols enable secure communication over insecure channels. Privacy protocols enable private interactions over secure channels. Security protocols set up secure channels using cryptographic primitives. Privacy protocols set up private channels using secure channels. But just like some security protocols can be broken without breaking the underlying cryptography, some privacy protocols can be broken without breaking the underlying security. Such privacy attacks have been used to leverage e-commerce against targeted advertising from the outset; but their depth and scope became apparent only with the overwhelming advent of influence campaigns in politics. The blurred boundaries between privacy protocols and privacy attacks present a new challenge for protocol analysis. Covert channels turn out to be concealed not only below overt channels, but also above: subversions, and the *level-below* attacks are supplemented by sublimations and the *level-above* attacks.

1 Introduction: What is privacy?

The concept of privacy has been a source of much controversy and confusion, not only in social and political discourse, but also in research.

The *controversy* arises from the fact that privacy is not a security requirement, like secrecy or authenticity, but a fundamental *social right*. In the US jurisprudence, justices Warren and Brandeis [57] defined it as the *right to be left alone*. Privacy is thus not just a technical task of controlling some assets, but first of all a *political* task, requiring that some policies should be specified and implemented. Privacy policies are generally designed to balance public and

*Supported by NSF.
†Partially supported by NSF and AFOSR.

private interests, assets, and resources. This balancing sometimes comes down to playing out the political forces against one another.

The *confusion* around privacy also arises from the fact that it is both a technical and a political problem. The problems of privacy are discussed in many different research communities, often in different terms, or with the same terms denoting different concepts. Privacy provides an opportunity for security researchers to contribute to political discourse [47]. It also provides politicians an incentive to get involved in the technical discourse about security. The overarching source of confusion are the political narratives constructed the process of shifting and blurring the boundaries between the public and the private, which has been the driving force behind social transformations for centuries [6, 26].

History of privacy. Social history is first of all the history of shifting demarcation lines between the public sphere and the private sphere [7, 39, 50]. Communist revolutions usually start by abolishing not only private property, but also private rights. Tyrannies and oligarchies, on the other hand, erode public rights and ownership, and privatize resources and social life. The distinction between the realm of public (city, market, warfare...) and the realm of private (family, household, childbirth...) was established and discussed in antiquity [4, 13]. It was a frequent topic in Greek tragedies: e.g. Sophocles' *Antigone* is torn between her private commitment to her brothers and her public duty to the king. The English word *politics* comes from the Greek word πόλις, denoting the public sphere; the English word *economy* comes from the Greek word οἶκος, denoting the private sphere.

Distinguishing privacy. There are many aspects of privacy, conceptualized in different research communities, and studied by different methods; and perhaps even more aspects that are not conceptualized in research, but arise in practice, and in informal discourse. We carve out a small part of the concept, and attempt to model it formally.

As an abstract requirement, privacy is a negative constraint, in the form *"bad things should not happen"*. Note that secrecy and confidentiality are also such negative constraints, whereas authenticity and integrity are positive constraints, in the form *"good things should happen"*. More precisely, authenticity and integrity require that some desirable information flows happen. E.g., a message *"I am Alice"* is authentic if it originates from Alice. On the other hand, confidentiality, secrecy and privacy require that some undesirable information flows are prevented: Alice's password should be secret, her address should be confidential, and her health record should be private.

But what is the difference between privacy, secrecy and confidentiality? Let us first move out of the way the difference between the latter two. In the present paper, we ignore that difference. In the colloquial usage, the terms confidentiality and secrecy allow subtle distinctions: e.g., when a report is confidential, we don't know its contents; but when it is secret, we don't even know that it exists. Secrecy and confidentiality are both security properties. We bundle them into one and use them interchangeably. (Restricting to just one of them gets awkward.)

As for the difference between privacy and secrecy (or confidentiality), it comes in two flavors:

1) while secrecy is a property of *information*, privacy is a property of any *asset* or *resource* that can be secured[1]; and

2) while secrecy is a *local* requirement, usually imposed on information flowing through a given channel, privacy is a *global* requirement, usually imposed on all resource requests and provisions, along any channel of a given network.

Let us have a look at some examples.

Ad (1), Alice's password is secret, whereas her bank account is private. Bob's health record is private, and it remains private after he shares some of it with Alice. It consists of his health information, but it may also contain some of his tissue samples for later analysis. On the other hand, Bob's criminal record is in principle not private, as criminal records often need to be shared, to protect the public. Bob may try to keep his criminal record secret, but even if he succeeds, it will not become private. Any resource can be made private if the access to it can be secured. E.g., we speak of a private water well, private funds, a private party if the public access is restricted. On the other hand, when we speak of a secret water well, secret funds, or a secret party, we mean that the public does not have any information about them. A water well can be secret but not private, or it can be private, but not secret; or it can be both, or neither.

Ad (2), to attack Alice's secret password, Bob eavesdrops at the channel where she uses it; to attack her bank account, he can, of course, impersonate Alice using her password; but he can also initiate a request through any of the channels of the banking network that he can access; best of all, he can coordinate an attack through many channels. To attack a secret, a cryptanalyst analyzes a given cipher. To attack privacy, a data analyst gathers and analyzes data from as many surveillance points as he can access. To protect secrecy, the cryptographer must assure that the plaintexts cannot be derived from the ciphertexts without the key, for a given cipher. To protect privacy, the network operator must assure that there are no covert channels anywhere in the network.

Defining privacy. Secrecy is formally defined in cryptography. The earliest definition, due to Shannon [51], says that it is a property of a channel where the outputs are statistically independent of the inputs. It is tempting, and seems natural, to define privacy in a similar way. This was proposed by Dalenius back in the 1970s [15]. A database is private if the public data that it discloses publicly say nothing about the private data that it does not disclose. This *desideratum*, as Dalenius called it, persisted in research for a number of years, before it became clear that it was generally impossible as a requirement. E.g., if everybody knows that Alice eats a lot of chocolate, but there is an anonymized database that shows a statistical correlation between eating a lot of chocolate and heart attacks, then this database discloses

[1]Information is, of course, a resource, so it can be private.

that Alice may be at a risk of heart attack, which should be Alice's private information, and thus breaches Dalenius' desideratum. Notably, this database breaches Alice's privacy *even if* Alice's record does not come about in it. Indeed, it is not necessary that Alice occurs in the database either for establishing the correlation between chocolate and heart attack, or for the public knowledge that Alice eats lots of chocolate; the two pieces of information can arise independently. Alice's privacy can be breached by linking two completely independent pieces of information, one about Alice and chocolate, the other one about chocolate and heart attacks. But since Alice's record does not come about in the database, it cannot be removed from it, or anonymized in it. Making sure that neither Alice nor any other record can be identified closer than up to a set of k other records with the same attributes, as required by the popular k-anonymity approach to privacy [56, 55], would not make any difference for Alice's privacy in this case either, since Alice cannot be identified at all in a database where she does not come about. Since Alice's privacy is not breached by identifying her, but by linking her public attribute (chocolate) with the public statistic correlating that attribute with a private attribute (heart disease), it follows that anonymity cannot assure privacy.

Notation.

It is convenient to view each natural number as the set of the numbers preceding it, i.e. $n = \{0, 1, 2, \ldots, n-1\}$.

2 Resources and their fusion

2.1 Concepts

Definition 2.1. *A* source element *of a set $\mathcal{Y}$ is a function $\beta : \mathcal{Y} \longrightarrow [0, 1]$ such that*

- *the set $\beta^{\#} = \{y \in \mathcal{Y} \mid \beta(y) \neq 0\}$ is finite, and*
- *the sum $\Sigma\beta = \sum_{y \in \mathcal{Y}} \beta(y)$ is not greater than 1.*

The set of all source elements of $\mathcal{Y}$ is denoted by $\mathcal{D}\mathcal{Y}$. The set $\beta^{\#}$ is called the support *of β, and its number of elements $\#\beta$ is called the* size *of β. The number $\Sigma\beta$ is the* total weight *of β. The values $\beta(y)$ are* weights *(or* probabilities*) of $y \in \mathcal{Y}$.*

Ordinary elements $y \in \mathcal{Y}$ correspond to the source elements $\chi_y : \mathcal{Y} \longrightarrow [0, 1]$ where $\chi_y(y) = 1$ and $\chi_y(z) = 0$ for all $z \neq y$. A source element β that happens to be *total*, in the sense that $\sum_{y \in \mathcal{Y}} \beta(y) = 1$, corresponds to what would normally be called a finitely supported probability distribution over $\mathcal{Y}$. The set of all total source elements of $\mathcal{Y}$ is $\Delta\mathcal{Y}$. They are often viewed geometrically, as points of the convex polytope spanned by $y \in \mathcal{Y}$ as vertices. If we adjoin to $\mathcal{Y}$ a fresh element $*$, and thus form the set $\mathcal{Y}' = \mathcal{Y} \cup \{*\}$, then each source element β of $\mathcal{Y}$ (not necessarily total) can be mapped into a total source element β' of $\mathcal{Y}'$,

defined

$$\beta'(y) \;=\; \begin{cases} \beta(y) & \text{if } y \in \mathcal{Y} \\ 1 - \Sigma\beta & \text{if } y = * \end{cases}$$

It is easy to see that this gives a bijection

$$\mathcal{D}\mathcal{Y} \;\cong\; \Delta\mathcal{Y}'$$

so that the source elements of $\mathcal{Y}$ can be viewed as the points of the convex polytope over $\mathcal{Y}'$. Extended along this bijection, the inclusion $\Delta\mathcal{Y} \subseteq \mathcal{D}\mathcal{Y}$ becomes the retraction $\Delta\mathcal{Y} \leftleftarrows \Delta\mathcal{Y}'$, which projects the polytope $\Delta\mathcal{Y}'$ to the face where the weight of $*$ is 0.

This geometric view of source elements of $\mathcal{Y}$, as convex combinations from $\mathcal{Y}'$, supports the intuition that they are *'incomplete'* probability distributions, which don't add up to 1. The probability deficiency $1 - \Sigma\beta$ can be thought of as the chance that sampling the source element β does not yield any output.

Definition 2.2. *A* resource *is a function $\varphi : \mathcal{X} \longrightarrow \mathcal{D}\mathcal{Y}$ whose* support $\varphi^{\#} = \{x \in \mathcal{X} \mid \varphi_x \neq 0\}$ *is finite. The number of elements of $\varphi^{\#}$ is called the* size *of φ again, and its* total size *is the number*

$$\#\#\varphi \;=\; \sum_{x \in \varphi^{\#}} \#\varphi_x$$

A resource that takes as its values the ordinary elements, i.e. the source elements which take a value 1, is just a partial function from $\mathcal{X}$ to $\mathcal{Y}$. A resource that takes only total source elements as its values can be viewed as a stochastic matrix, i.e. a matrix of numbers from the interval $[0, 1]$ where columns add up to 1. Any resource $\varphi : \mathcal{X} \longrightarrow \mathcal{D}\mathcal{Y}$ can be viewed as a $\mathcal{X} \times \mathcal{Y}$-matrix of numbers $\varphi_x(y) \in [0, 1]$. Although the sets $\mathcal{X}$ and $\mathcal{Y}$ can be infinite, the finiteness of $\#\varphi$ and of all $\#\varphi_x$ implies that only $\#\#\varphi$ many entries of this infinite matrix are different from 0. Any given resources $\varphi : \mathcal{X} \longrightarrow \mathcal{D}\mathcal{Y}$ and $\psi : \mathcal{Y} \longrightarrow \mathcal{D}\mathcal{Z}$ can thus be composed into a resource $\psi\varphi : \mathcal{X} \longrightarrow \mathcal{D}\mathcal{Z}$, obtained by matrix composition

$$(\psi\varphi)_x(z) \;=\; \sum_{y \in \psi^{\#}} \varphi_x(y) \cdot \psi_y(z)$$

Notation. Since we will most of the time look at the functions in the form $\mathcal{X} \longrightarrow \mathcal{D}\mathcal{Y}$, let us omit the $\mathcal{D}$, and write such functions as $\mathcal{X} \longrightarrow\!\!\!\!\!\rightarrow \mathcal{Y}$, denoting by the arrow with full head $\longrightarrow\!\!\!\!\!\rightarrow$ a function whose outputs are source elements. Such functions are our resources.

2.2 Examples

While the notion of resource is very general, we begin with some very special cases, to ground the intuitions.

Example 0: Simple communication channels are the most basic examples of resources. More precisely, a *memoryless channel*, as defined in any information theory textbook, is a resource $\varphi : \mathcal{X} \longrightarrow \mathcal{Y}$, where $\mathcal{X}$ and $\mathcal{Y}$ are finite alphabets, and all source elements φ_x are total. E.g., the binary symmetric channel is such a resource over the alphabets $\mathcal{X} = \mathcal{Y} = \{0, 1\}$, and with the elements distributed by

$$\varphi_x(y) \;=\; \Pr(y|x) \;=\; \begin{cases} 1-p & \text{if } x = y \\ p & \text{otherwise} \end{cases}$$

A channel that depends on state, but with no feedback, can also be viewed as a resource, at least at each finite step, since its outputs always depend on finite histories, and are thus finitely supported. Asynchronous channels can be modeled using source elements that may not be total. However, our path in this paper leads in a different direction.

Example 1: A **database** is a function $\varphi : \mathcal{R} \times \mathcal{C} \longrightarrow \mathcal{Y}$, where

- $\mathcal{R}$ is a set of *rows*, or *records*, or *items*,
- $\mathcal{C}$ is a set of *columns*, or *attributes*,
- $\mathcal{Y}$ is a set of *values* or *outputs*.

The "row-column" terminology suggests that we think of a database as an $\mathcal{R} \times \mathcal{C}$-matrix of entries from $\mathcal{D}\mathcal{Y}$. When the attributes $c \in \mathcal{C}$ are expressed using different sets of values $\mathcal{Y}_c$, we take $\mathcal{Y} = \bigcup_{c \in \mathcal{C}} \mathcal{Y}_c$. For ordinary databases, the entries are ordinary elements $\varphi_{rc} \in \mathcal{Y}$, or they may be empty, and the resource φ is an ordinary map, or a partial map. For online databases, it can be uncertain what will be returned in response to a query, since the data are updated dynamically, often concurrently; so the entries are stochastic, and we model them as source elements $\varphi_{rc} \in \mathcal{D}\mathcal{Y}$.

In fact, the whole web can be viewed as a large, dynamic, stochastic database.

Example 2: The **web** is a resource $\omega : \mathcal{X} \longrightarrow \mathcal{Y}$ where

- $\mathcal{X}$ is the set of URLs (or more precisely of all possible HTTP requests),
- $\mathcal{Y}$ is the set of HTML documents (embedded in the HTTP responses, often extended with JavaScript executables, JSON or XML object references, etc.)

The web is, of course, a very complex, very dynamic environment, and when you input a URL $x \in \mathcal{X}$ into your browser, the process that determines what output will the random variable ω_x return to your browser is ongoing nonlocally, and there is a lot of uncertainty and chaos in it, like in any complex natural process, such as the weather. The web is a typical source showing why privacy is so hard: it combines politics and thermodynamics.

Example 3: A **search engine** $\gamma : \mathcal{X} \longrightarrow \mathcal{Y}$ (e.g. Google) is an attempt at a map of the web:

- $\mathcal{X}$ is the index of keywords and search terms built from them,
- $\mathcal{Y}$ is the set of indexed web pages.

The distribution of $\gamma_x \in \mathcal{D}\mathcal{Y}$ over the set of the hits for the search term $x \in \mathcal{X}$ captures the web page *ranking* [40, 41]: higher ranked pages are assigned higher probabilities in γ_x.

Example 4: A **social engine** $\zeta : \mathcal{X} \longrightarrow\!\!\!\!\!\bullet\ \mathcal{Y}$ (e.g. Facebook) is a shared resource for social networking through posting messages, media, and gestures, and distributing them according to some specified privacy policies along the social channels provided by the platform. In the most basic model,

- $\mathcal{X}$ is the set of identifiers of all users' postings,
- $\mathcal{Y}$ are the contents of the postings, i.e. the set of the posted messages, media, and gestures.

For simplicity, we assume that the identifiers $x \in \mathcal{X}$ contain all needed references to their sources, and that the contents of the postings are either equal (i.e., they can be repeated) or different, but have no intrinsic structure or correlations. A social engine can thus be viewed as an ordinary mapping $\zeta : \mathcal{X} \longrightarrow \mathcal{Y}$ of identifiers as ordinary elements $x \in \mathcal{X}$ into the postings as ordinary elements $\zeta_x \in \mathcal{Y}$. The randomness will emerge from sharing: who sees whose postings. And sharing is determined by running privacy protocols.

2.3 Resource fusion

Data sources are naturally ordered by the amount of information that they provide: e.g., a database φ may provide Alice's IP address, say 98.151.86.153; a database ψ may provide just the first block: 98. The amount of information is usually quantified by the entropy of its source: e.g., Alice's record φ_A would contain 32 bits of information, whereas Alice's record ψ_A would contain only 8 bits. However, if the IP prefix given by ψ_A is different from the IP prefix given by φ_A, and if we are interested in the actual address, then counting the bits does not suffice. Data analysts do not just *quantify* the amounts of information in the available data, they also *qualify* them, and compare their information contents. To model that practice, we need an ordering $\prec$ where $\psi_A \prec \varphi_A$ only if the information provided by ψ_A is really contained in φ_A, not just more uncertain; and moreover, we need an operation $\psi_A \widehat{\Upsilon} \varphi_A$ for *information fusion*, which will join together the parts of ψ_A and φ_A where they are consistent with each other, and discard the parts where they contradict each other. This may sound like a tall order in theory, but that is what data analysts do in practice.

The problems of ordering information sources turn out to have been studied, albeit implicitly, in the theory of *majorization* [2, 30]. The basic techniques go back to [27], where a host of inequalities from different parts of mathematics were derived using majorization, as if by magic. In the meantime, it has been well understood that the power of magic was due to ordering information sources, but the problem of conjoining and reconciling information sources has not yet been been directly addressed, although some technical results and conceptual expositions came close to it [3, 38].

2.3.1 Preferences and consistency

Definition 2.3. *Every source element $\beta \in \mathcal{D}\mathcal{Y}$ induces the following binary relations:*

- strict preference*:* $u \overset{\beta}{\lhd} v \iff \beta(u) < \beta(v)$*;*
- preference*:* $u \overset{\beta}{\trianglelefteq} v \iff \beta(u) \leq \beta(v)$*;*
- indifference*:* $u \overset{\beta}{\diamond} v \iff \beta(u) = \beta(v)$.

Clearly, $\overset{\beta}{\lhd}$ is transitive, $\overset{\beta}{\trianglelefteq}$ is also reflexive; and $\overset{\beta}{\diamond}$ is moreover symmetric, and thus an equivalence relation. The preference relation $\overset{\beta}{\trianglelefteq}$ is thus a preorder on $\mathcal{Y}$ and a partial order on the set of $\overset{\beta}{\diamond}$-equivalence classes $\mathcal{Y}/\overset{\beta}{\diamond}$. For $U, V \subseteq \mathcal{Y}$ we thus write

$$U \overset{\beta}{\lhd} V \iff \forall uu' \in U\ \forall v'v \in V.\ u \overset{\beta}{\diamond} u' \wedge u' \overset{\beta}{\lhd} v' \wedge v' \overset{\beta}{\diamond} v$$

and ditto for $U \overset{\beta}{\trianglelefteq} V$.

Definition 2.4. *For a finite set of source elements $B \subset \mathcal{D}\mathcal{Y}$, we define*

$$\begin{aligned} u \overset{B}{\trianglelefteq} v \iff\ & u \overset{\beta_0}{\trianglelefteq} w_1 \overset{\beta_1}{\trianglelefteq} w_2 \overset{\beta_2}{\trianglelefteq} \cdots w_n \overset{\beta_n}{\trianglelefteq} v \\ & \text{for some } \beta_0, \beta_1, \ldots, \beta_n \in B \text{ and } w_1, \ldots, w_n \in \mathcal{Y} \\ u \overset{B}{\diamond} v \iff\ & u \overset{B}{\trianglelefteq} v \text{ and } v \overset{B}{\trianglelefteq} u \end{aligned}$$

Then we say that the set B is consistent *if*

$$u \overset{\beta}{\lhd} v \text{ for some } \beta \in B \implies u \overset{\delta}{\trianglelefteq} v \text{ for all } \delta \in B$$

In addition, we define a $\lhd$-cycle *in B as $u_0 \overset{\delta_1}{\lhd} u_1 \overset{\delta_2}{\lhd} \ldots u_{n-1} \overset{\delta_n}{\lhd} u_n$ where $u_i \in \mathcal{Y}$ and $\delta_i \in B$.*

Comment. Inconsistency means that taking the transitive closure of the relations $\overset{\beta}{\trianglelefteq}$ for $\beta \in B$ creates new $\lhd$-cycles of preferences. For instance, if $B = \{\beta, \delta\}$ are such that

- $a \overset{\beta}{\lhd} b$ and $c \overset{\beta}{\lhd} d$, but
- $b \overset{\delta}{\lhd} c$ and $d \overset{\delta}{\lhd} a$,

then $\overset{B}{\trianglelefteq}$ contains the $\lhd$-cycle $a \overset{\beta}{\lhd} b \overset{\delta}{\lhd} c \overset{\beta}{\lhd} d \overset{\delta}{\lhd} a$. This $\lhd$-cycle makes a, b, c and d all $\overset{\beta,\delta}{\diamond}$-equivalent, while it is easy to assign the weights in β and δ in such a way that no pair of elements is either $\overset{\beta}{\diamond}$-equivalent or $\overset{\delta}{\diamond}$-equivalent.

Example. Let $\beta, \gamma, \delta \in \mathcal{D}\mathcal{Y}$ be sources with support $\{w, x, y, z\} \subset \mathcal{Y}$. Let their values on the support be defined in the table below.

	w	x	y	z
β	0.1	0.2	0.2	0.3
γ	0.1	0.1	0.3	0.5
δ	0.6	0.1	0.1	0.2

To illustrate consistency, we display the strict preference order on equivalence classes below.

$$\begin{array}{ccccc} \{w\} & \overset{\beta}{\lhd} & \{x,y\} & \overset{\beta}{\lhd} & \{z\} \\ \{w,x\} & \overset{\gamma}{\lhd} & \{y\} & \overset{\gamma}{\lhd} & \{z\} \\ \{x,y\} & \overset{\delta}{\lhd} & \{z\} & \overset{\delta}{\lhd} & \{w\} \end{array}$$

Let $B = \{\beta, \gamma\}$, then the transitive closure results in the total order

$$\{w, x, y\} \overset{B}{\lhd} \{z\}$$

Observe $z \overset{\delta}{\lhd} w$, yet $w \overset{\beta}{\lhd} z$ and $w \overset{\gamma}{\lhd} z$. Thus δ is not consistent with neither β nor γ, and therefore not B.

Definition 2.5. *An* ordering ϑ *of a source element* $\beta : \mathcal{Y} \longrightarrow [0,1]$ *is a pair of functions* $N \underset{\vartheta}{\overset{\widetilde{\vartheta}}{\leftrightarrows}} \mathcal{Y}$ *such that*

- $N > \#\beta$*;*
- *for all* $i \in N = \{0, 1, \ldots, N-1\}$ *holds* $\widetilde{\vartheta}\vartheta(i) = i$ *;*
- *for all* $y \in \beta^{\#}$ *holds* $\vartheta\widetilde{\vartheta}(y) = y$*,*
- $\beta\vartheta(0) \geq \beta\vartheta(1) \geq \cdots \geq \beta\vartheta(\#\beta - 1) > \beta\vartheta(\#\beta) = \cdots = \beta\vartheta(N-1) = 0.$

The set of all orderings of $\beta \in \mathcal{D}\mathcal{Y}$ *is denoted by* $\Theta(\beta)$*.*

We hope that this formal definition does not conceal the idea of ordering a source element, which is as simple as it sounds: an ordering ϑ of $\beta : \mathcal{Y} \longrightarrow [0,1]$ enumerates the support $\beta^{\#}$ by the indices from the set $N = \{0, 1, \ldots, \#\beta - 1, \#\beta, \ldots, N-1\}$ in such a way that $u \in \beta^{\#}$ are ordered by weight, with those with the highest weights $\beta(u)$ coming first. All $u \in \mathcal{Y}$ for which $\beta(u) = 0$, and thus $u \notin \beta^{\#}$, are mapped to $\vartheta(u) \geq \#\beta$. Any indifferent pair $u, v \in \mathcal{Y}$, i.e. such that $\beta(u) = \beta(v)$ and thus $u \overset{\beta}{\diamond} v$, allows 2 different orderings. Any indifference class with k elements allows $k!$ different orderings. On the other hand, a source element β where for any pair $u \neq v \in \beta^{\#}$ holds $\beta(u) \neq \beta(v)$ induces a unique ordering of its support $\beta^{\#}$. Indeed, it is easy to show that in that case, all orderings $\vartheta, \xi \in \Theta(\beta)$ must satisfy $\beta\vartheta(i) = \beta\xi(i)$ for all $i < \#\beta$, and thus induce the same ordered sequence $\beta\vartheta = \beta\xi$.

This unique ordered version of β is usually written $\beta^{\downarrow}$, but we will write $\beta\vartheta$ when an explicit ordering is needed, as it will be the case in Prop. 2.10. Identifying the set $\beta^{\#} \subseteq \mathcal{Y}$ with the set of numbers $\#\beta = \{0, 1, \ldots, \#\beta - 1\} \subset N$ allows writing the source element $\beta : \mathcal{Y} \longrightarrow [0, 1]$ as the descending sequence

$$\beta^{\downarrow} \;=\; \beta\vartheta \quad = \quad \big\langle \beta\vartheta(0), \beta\vartheta(1), \ldots, \beta\vartheta(\#\beta - 1), 0, 0, 0, \ldots \big\rangle$$

Note that "un-ordering" $\beta\vartheta$ by $\widetilde{\vartheta}$ restores β, because for all $y \in \beta^{\#}$ holds $\beta\vartheta\widetilde{\vartheta}(y) = \beta(y)$. We can also consider $\beta^{\downarrow}$ as a source over the natural numbers, $\beta^{\downarrow} : \mathbb{N} \to [0, 1]$. Given a set $B \subset \mathcal{D}\mathcal{Y}$ and the unique ordered version of the source elements, we may construct a set $B^{\downarrow} = \{\beta^{\downarrow} | \beta \in B\} \subset \mathcal{D}\mathbb{N}$. Observe, given any finite set of source elements the above set will always have a common ordering. Let $N > \#B$. Then an example of a common ordering is $\vartheta(i) = i$ for all $i \in N$, $\widetilde{\vartheta}(i) = i$ for $i < N$, and $\widetilde{\vartheta}(i) = N - 1$ for $i > N$.

Example. Let $\mathcal{Y}$ be the set of URLs and $\beta : \mathcal{Y} \longrightarrow [0, 1]$, a source element. Let β represent the probability of visiting a URL first for a certain user when they open a web browser.

URL	β
http://www.google.com	1/4
https://www.amazon.com	1/4
https://en.wikipedia.org	1/4
https://stackoverflow.com/	1/8

Let β be zero for all other URLs. Observe that the probabilities do not add up to one. The intent is to capture what happens if they immediately close the browser, or say the computer crashes and the user is unable to visit a URL. Given the above ordering we may define $\vartheta(0), \vartheta(1), \vartheta(2), \vartheta(3), \vartheta(4)$ accordingly. Thus we have $\beta^{\downarrow} = \big\langle \frac{1}{4}, \frac{1}{4}, \frac{1}{4}, \frac{1}{8}, 0 \big\rangle$. The resulting total order is

$$\{\vartheta(3)\} \overset{\beta}{\lhd} \{\vartheta(0), \vartheta(1), \vartheta(2)\}$$

The orderings of β that are extended to $N > \beta^{\#}$ just add more indices than there are in $\#\beta$, and thus cover by the enumeration ϑ not only the support $\beta^{\#}$, but also some $y \in \mathcal{Y}$ for which $\beta(y) = 0$. Such extensions are needed when we look for common orderings of $\mathcal{Y}$ induced by different source elements $\beta, \gamma : \mathcal{Y} \longrightarrow [0, 1]$. Two such source elements may have different supports $\beta^{\#}$ and $\gamma^{\#}$, but their orderings may be consistent, in the sense that they both may extend to the same ordering of $\beta^{\#} \cup \gamma^{\#}$. Such an extended ordering ϑ with N enumerating $\beta^{\#} \cup \gamma^{\#}$ would belong both to $\Theta(\beta)$ and to $\Theta(\gamma)$, and would thus be an element of the intersection $\Theta(\beta) \cap \Theta(\gamma)$. It will turn out that this intersection characterizes the situation when β and γ are consistent, and can be conjoined into a single source: see Prop. 2.10 below.

2.3.2 Majorization preorder

Sequence differentials and integrals. To define joins and meets of source elements, we borrow the following operations from [45]. Let $\mathbb{R}^*$ be the set of sequences of reals. The operations $\int, \partial : \mathbb{R}^* \longrightarrow \mathbb{R}^*$ map any sequence $\langle \alpha(0), \alpha(1), \ldots, \alpha(n-1) \rangle$ to its *integral version* $\langle \int\alpha(0), \ldots, \int\alpha(n-1) \rangle$ and its *differential version* $\langle \partial\alpha(0), \ldots, \partial\alpha(n-1) \rangle$ defined

$$\int\alpha(k) \;=\; \sum_{i=0}^{k} \alpha(i) \qquad\qquad \partial\alpha(k) \;=\; \alpha(k) - \alpha(k-1)$$

for $0 \leq k < n$. We assume $\alpha(-1) = 0$. Note that $\int\partial(\alpha) = \alpha = \partial\int(\alpha)$.

Proposition 2.6. *For any $\beta, \gamma \in \mathcal{D}\mathcal{Y}$, and $n = \max(\#\beta, \#\gamma)$ the following conditions are equivalent:*

a) $\beta = D\gamma$, where D is a doubly substochastic matrix[2];

b) $\beta = \sum_{i=0}^{n-1} \lambda_i P_i \gamma$ where $\lambda_i \in [0,1]$, $\sum_{i=0}^{n-1} \lambda_i \leq 1$, and P_i are partial permutations, i.e., a submatrix of a permutation matrix;

c) $\int\beta^{\downarrow}(k) \leq \int\gamma^{\downarrow}(k)$ for all $k < n$.

Definition 2.7. *When source elements β and γ satisfy any of the equivalent conditions of Thm. 2.6, we say that β is* majorized[3] *by γ and write $\beta \prec \gamma$.*

2.3.3 Meets, joins, and fusions

Since all $\overset{\beta}{\trianglelefteq}$ are *total* preorders, in the sense that any two elements of $\mathcal{Y}$ are comparable, and since each of them thus induces a strict total order on its indifference equivalence classes, it follows that the total preorder $\overset{B}{\diamond}$ is the least common refinement of all partitions of $\mathcal{Y}$ induced by $\beta \in B$, in the sense that it induces a strict total order on its own of equivalence classes, which are the least common refinement of $\overset{\beta}{\diamond}$ for $\beta \in B$. The task is now to lift the least common refinement $\overset{B}{\trianglelefteq}$ of the preference preorders $\overset{\beta}{\trianglelefteq}$ induced by source elements $\beta \in B \subset \mathcal{D}\mathcal{Y}$ to a least lower bound $\curlyvee B$ with respect to a suitable information preorder $\prec$.

[2] A $\mathcal{Y} \times \mathcal{Y}$-matrix with finitely many nonzero, nonnegative entries is doubly stochastic if the sums of the entries in each nonzero row and in each nonzero column are 1. Already Garrett Birkhoff considered infinite doubly stochastic matrices, asking for the infinitary generalization of his doubly stochastic decomposition in the problem 111 of his *Lattice Theory*.

[3] This is also referred to as weak majorization, see for example [31].

Proposition 2.8. *Let $B \subset \mathcal{D}\mathcal{Y}$ be a finite set of source elements such that $\bigcap_{\beta\in B} \Theta(\beta)$ is not empty. Let $M \overset{\widetilde{\vartheta}}{\underset{\vartheta}{\rightleftarrows}} \mathcal{Y}$ be any one of the common orderings in the intersection. The meet and join of B with respect to the majorization preorder are respectively*

$$\curlywedge B \;=\; \Big(\partial \bigwedge_{\beta\in B} \textstyle\int \beta\vartheta\Big)\widetilde{\vartheta} \qquad\qquad \curlyvee B \;=\; \Big(\partial \bigvee_{\beta\in B} \textstyle\int \beta\vartheta\Big)\widetilde{\vartheta}$$

where $\bigwedge$ and $\bigvee$ are pointwise.

Corollary 2.9. *Let $B \subset \mathcal{D}\mathcal{Y}$ be a finite set of source elements. There exists at least one common ordering $M \overset{\widetilde{\vartheta}}{\underset{\vartheta}{\rightleftarrows}} \mathcal{Y}$ in $\bigcap_{\beta^{\downarrow}\in B^{\downarrow}} \Theta(\beta^{\downarrow})$. Their meet and join with respect to the majorization preorder are respectively*

$$\curlywedge B^{\downarrow} \;=\; \partial \bigwedge_{\beta^{\downarrow}\in B^{\downarrow}} \textstyle\int \beta^{\downarrow} \qquad\qquad \curlyvee B^{\downarrow} \;=\; \partial \bigvee_{\beta^{\downarrow}\in B^{\downarrow}} \textstyle\int \beta^{\downarrow}$$

where $\bigwedge$ and $\bigvee$ are pointwise and $\curlywedge B^{\downarrow}, \curlyvee B^{\downarrow} \in \mathcal{D}\mathbb{N}$

Corollary 2.9 states that given any finite set of source elements, the meet and join always exist for the unique order. Furthermore, this meet and join will help quantify the inconsistency of a finite set of source elements.

Relating majorization to fusion. The fact that majorization is a reflexive and transitive relation, i.e. a *preorder* on $\mathcal{D}\mathcal{Y}$, is just a slight refinement of the classical results of [27] and [11]. In the meantime, the role of majorization as *information* preorder has also been well established in several research communities [38]. Our main interest here is to understand and model the operation of *fusion* of resources, used in the practices of data analysis, and in the attacks on private resources. The usual scenario is that an analyst acquires two or more resources, and fuses them together, in order to extract as much information or value as possible. Intuitively, this corresponds to finding a least upper bound (usually called *supremum*, or *join*) with respect to the majorization preorder of the acquired resources. But a pair of sources may not have any upper bounds with respect to majorization, and therefore cannot always be conjoined together. The following proposition characterizes that situation as inconsistency. Prop. 2.12, coming right after, describes what can be done to make a set of sources consistent.

Proposition 2.10. *The following conditions on a finite set $B \subset \mathcal{D}\mathcal{Y}$ are equivalent:*

a) B is consistent (in the sense of Def. 2.4);

b) there is a common ordering $\vartheta \in \bigcap_{\beta\in B} \Theta(\beta)$ of B (in the sense of Def. 2.5);

c) $\exists\ \theta : \mathbb{N} \to \mathcal{Y}$, *such that,* $\int \bigwedge B^{\downarrow} = \bigwedge_{\beta \in B} \int \beta\theta$;

d) $\exists\ \theta : \mathbb{N} \to \mathcal{Y}$, *such that,* $\int \bigvee B^{\downarrow} = \bigvee_{\beta \in B} \int \beta\theta$.

Given a finite set of sources, what can we do if they are inconsistent? Suppose we are predicting a hurricane path, and several models each predict fundamentally different paths. How do we extract the information that is consistent between all models? Given $B \subset \mathcal{DY}$ is finite, our first step is to partition the source $\mathcal{Y}$, so that we may find the common information on the partition elements. Each set in the partition is defined in a minimal way so that sources in B must be made constant on them to fuse the consistent information. Given $y \in \mathcal{Y}$ we define the *consistency class* of B at y as the set

$$B_y = \{y\} \cup \{u \in \mathcal{Y} \mid u \text{ is in a } \lhd\text{-cycle with } y \text{ in } B\}$$

Lemma 2.11. *Given $B \subset \mathcal{DY}$ is finite, the consistency classes of B, $\{B_y\}_{y \in \mathcal{Y}}$, are a partition of $\mathcal{Y}$. Furthermore, if $B_u \cap B_v = \emptyset$, then either $B_u \overset{\delta}{\trianglelefteq} B_v$ or $B_v \overset{\delta}{\trianglelefteq} B_u$ for all $\delta \in B$.*

The following proposition captures how to make a set of sources consistent while not increasing the individual probabilities of any element. Furthermore, the fixed points will correspond to consistent sets.

Proposition 2.12. *Let $B \subset \mathcal{DY}$ be a finite set. Given $\beta \in B$ define the function $\widehat{\beta} : \mathcal{Y} \longrightarrow [0, 1]$ by*

$$\widehat{\beta}(y) \;=\; \bigwedge_{u \in B_y} \beta(u)$$

Then $\widehat{\beta}$ is the greatest among all source elements $\alpha \prec \beta$ which are consistent with the elements of B. Moreover, the set $\widehat{B} = \left\{\widehat{\beta} \mid \beta \in B\right\}$ is consistent.

Corollary 2.13. *Any finite set of source elements $B \subset \mathcal{DY}$ has a majorization meet $\bigwedge B = \bigwedge \widehat{B}$, whereas its meet is constructed as in Prop. 2.8.*

Definition 2.14. *The* source fusion *of a finite set $B \subset \mathcal{DY}$ is the source element $\widehat{\bigvee} B \in \mathcal{DY}$ defined by*

$$\widehat{\bigvee} B \;=\; \bigvee \widehat{B}$$

where $\widehat{B}$ is the consistent set constructed as in Prop. 2.12.

A concrete and familiar **example** of this operation will be described in Sec. 5.2.

Definition 2.15. *The* resource fusion $\widehat{\Upsilon}\Phi : \mathcal{X} \longrightarrow \mathcal{Y}$ *of a finite set of resources* $\Phi \subset \mathcal{X} \longrightarrow \mathcal{Y}$ *is defined pointwise along* $x \in \mathcal{X}$ *by*

$$\left(\widehat{\Upsilon}\Phi\right)_x \;=\; \widehat{\Upsilon}\Phi_x$$

where the source fusion on the right-hand side is from Def. 2.14.

Inconsistent sources generate a higher-order source. Another view of the above definition is that the set of resources Φ is the pointwise join of the set of resources $\widehat{\Phi}$, where for each $x \in \mathcal{X}$, the set of sources $\widehat{\Phi}_x$ is the greatest consistent set under Φ_x. The notion of consistency of sources, as imposed in Def. 2.4 requires a consensus of all sources, wherever they declare their preferences by assigning different weights. This is a very restrictive notion of consistency. Different domains require different notions of fusion. Through centuries, many different forms of preference aggregation have been proposed and are nowadays systematized and analyzed in theories of voting and social choice [49, 53, 12]. An important aspect not analyzed within that tradition are the *higher-order* source aggregations. While with the preference aggregations and the source fusion operations like the one presented above, inconsistent sources are conjoined into a less informative source, any inconsistencies observed in hypothesis testing are the source of new information within a higher-order source [46]. See Sec. 5.2 for an example.

The only point of even mentioning this vast conceptual area in this constrained space is in support of our claim that the rapidly evolving practices of information gathering and analysis require a theory of information that takes into account the information content and quality, and not just the transmission rate measures, and quantity.

3 Privacy protocols for private channels

3.1 Concepts

Interactions, communication, and resource sharing are usually modeled using networks. A *network* structure is based on a graph, here consisting of

- a set $\mathcal{S} = \{A, B, C, D, \ldots, S, \ldots\}$ of *nodes*, representing subjects or users, often called Alice, Bob, Carol, Dave…; and
- a set of links $A \to B$, representing the channels between Alice and Bob.

In most examples, there will be at most one channel between any pair of subjects, so the network structure boils down to a binary relation on the set $\mathcal{S}$. In any case, channels allow us to model *local* interactions: Alice can interact with Bob only if there is a channel $A \to B$, in which case we say that Bob is Alice's *neighbor*. A network is often assumed to provide routing services, whereby Alice can send a message or object to Dave, who is not her neighbor, from

neighbor to neighbor: $A \to B \to C \to D$. The Internet is, of course, a network with routing services.

A network can be viewed as infrastructure for sharing some private resources. The node Alice has a private resource $[\![A]\!] : \mathcal{X}_A \longrightarrow\!\!\!\!\!\bullet\ \mathcal{Y}_A$, but she can achieve more if she cooperates with Bob, who has a resource $[\![B]\!] : \mathcal{X}_B \longrightarrow\!\!\!\!\!\bullet\ \mathcal{Y}_B$, and can share some of it with Alice, often in exchange for some of hers. Privacy protocols are abstract models of such transactions. Optimizing utility of their private resources, all rational agents engage in such transactions at all network levels. The obvious examples are market transactions, where Alice and Bob trade goods, money, labor; also health care, insurance, rescue missions, regulatory control, search, education. At lower levels of interaction, many basic social phenomena arise from privacy protocols. But in a data-driven society, certain data privacy protocols take a life of their own, while other privacy protocols become invisible, and private.

The basic building block of private interactions is a 2-message protocol pattern, depicted in Fig. 1, where Alice submits to Bob a *request* $\boldsymbol{r}^{AB}$ for some of his private resource $[\![B]\!]$, and Bob responds according to his *policy* $\boldsymbol{p}^{AB}$, and shares with Alice the part of his resource that results from composing her request and his policy, i.e. $[\![AB]\!] = \boldsymbol{p}^{AB}[\![B]\!]\boldsymbol{r}^{AB}$. Our claim is that all privacy protocols are obtained by composing suitable instances of this pattern. The idea is that these *Request-Policy (RP)*-components are atoms of privacy, just like the *Challenge-Response (CR)*-components are atoms of authentication [19, 14, 42, 44, 43]. The incremental approach to protocol design, analysis, taxonomy, and to security proofs [17, 22, 35] seems to extend naturally from authentication and key establishment protocols to privacy protocols.

3.2 Basic privacy protocols

Bob's private resource $[\![B]\!] : \mathcal{X}_B \longrightarrow\!\!\!\!\!\bullet\ \mathcal{Y}_B$ accepts Bob's private inputs from $\mathcal{X}_B$ and produces Bob's private outputs in $\mathcal{Y}_B$. In order to be able to request access to some of Bob's private resource, Alice must be able to reference some of his private inputs, and to observe and utilize some of his private outputs. To allow subjects to refer to each other's private resources, we distinguish

- sets $\mathcal{X}$ and $\mathcal{Y}$ of *global* identifiers; and
- sets $\mathcal{X}_S$ and $\mathcal{Y}_S$ of *local* identifiers, owned by each $S \in \mathcal{S}$, and accessed by the owner using the functions

$$\mathcal{X}_S \underset{\underline{\pi}_S}{\overset{\overline{\pi}_S}{\rightleftarrows}} \mathcal{X} \qquad\qquad \mathcal{Y}_S \underset{\underline{\rho}_S}{\overset{\overline{\rho}_S}{\rightleftarrows}} \mathcal{Y} \tag{1}$$

  such that $\overline{\pi}_S\underline{\pi}_S(x) = x$ and $\overline{\rho}_S\underline{\rho}_S(y) = y$.

The last two equations make $\underline{\pi}_S$ and $\underline{\rho}_S$ total, but $\overline{\pi}_S$ and $\overline{\rho}_S$ can be genuinely partial, in which case the composites $\pi_S = \underline{\pi}_S\overline{\pi}_S$ and $\rho_S = \underline{\rho}_S\overline{\rho}_S$ are also partial. It is easy to see that they are also idempotent, i.e. satisfy $\pi_S\pi_S = \pi_S$ and $\rho_S\rho_S = \rho_S$. The following definition puts together all of the above.

Definition 3.1. *A* resource network *consists of*

- *a* network $\mathcal{S} = \{A, B, C, \ldots, S, \ldots\}$*;*
- global identifiers $\mathcal{X} = \{x, x', \ldots\}$ *for inputs and* $\mathcal{Y} = \{y, y', \ldots\}$ *for outputs;*
- *for each network node* $S \in \mathcal{S}$*, the local castings like in* (1)*, which induce*
  - *the projector* $\pi_S : \mathcal{X} \rightharpoonup \mathcal{X}$ *determining the* local input identifiers $\mathcal{X}_S = \{x \in \mathcal{X} \mid \pi_S(x) = x\}$*, and*
  - *the projector* $\rho_S : \mathcal{Y} \rightharpoonup \mathcal{Y}$ *determining the* local output identifiers $\mathcal{Y}_S = \{y \in \mathcal{Y} \mid \rho_S(y) = y\}$*,*
  - *a* resource $[\![S]\!] : \mathcal{X} \longrightarrow \mathcal{Y}$*.*

Remark. Alice's resource $[\![A]\!]$ provides her with the output $\rho_A[\![A]\!]x$ on input $x \in \mathcal{X}_A$, and in many cases it satisfies $[\![S]\!] = \rho_S[\![S]\!]\pi_S$, or equivalently $\rho_S[\![S]\!] = [\![S]\!] = [\![S]\!]\pi_S$. However, it may happen that $x, x' \in \mathcal{X}$ are indistinguishable for Alice as inputs, i.e. $\pi_A(x) = \pi_A(x')$, but that they give her different outputs i.e. $\rho_A[\![A]\!](x) \neq \rho_A[\![A]\!](x')$. Intuitively, this means that the global identifiers may *interfere* with the environment in ways that are for Alice not directly observable, but that she may be able to observe such interferences indirectly, from the outputs of her resource. This will be discussed in Sec. 4.

Definition 3.2. *A* Request-Policy (RP) *protocol between Alice and Bob in a resource network* $\mathcal{S}$ *is a pair of partial functions* $\Phi = \langle \boldsymbol{r}_\Phi^{AB}, \boldsymbol{p}_\Phi^{AB} \rangle$*, where*

- $\boldsymbol{r}_\Phi^{AB} : \mathcal{X} \rightharpoonup \mathcal{X}$ *is Alice's* request *for Bob's private resource, and*
- $\boldsymbol{p}_\Phi^{AB} : \mathcal{Y} \rightharpoonup \mathcal{Y}$ *is Bob's privacy* policy *towards Alice.*

The outcome *of a run of the privacy protocol is that the part of Bob's private resource that is approved by Bob's policy and referenced by Alice's request is released to her, providing her with the resource:*

$$[\![AB]\!] \;=\; \big(\mathcal{X} \xrightarrow{\boldsymbol{r}_\Phi^{AB}} \mathcal{X} \xrightarrow{[\![B]\!]} \mathcal{Y} \xrightarrow{\boldsymbol{p}_\Phi^{AB}} \mathcal{Y}\big) \tag{2}$$

which is then conjoined with Alice's own resource $\mathcal{X} \xrightarrow{[\![A]\!]} \mathcal{Y}$*, thus providing Alice with the total resource* $[\![A]\!]\widehat{\Upsilon}[\![AB]\!]$*, as displayed in Fig. 1.*

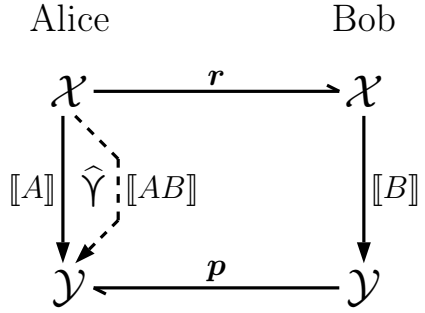


Figure 1: Basic privacy protocol: Request-Policy (RP)

Privacy protocols can be driven by demand, or by supply. The two interactions displayed in Fig. 1, corresponding to the request $\boldsymbol{r}$ and the policy $\boldsymbol{p}$ may happen in time in either order: some RP-protocols are initiated by the requester Alice, whereas other are initiated by the provider Bob. Fig. 2 displays the two directions of in which the protocols can be executed as the hollow arrows. E.g., a seller Bob may advertise his goods, provide

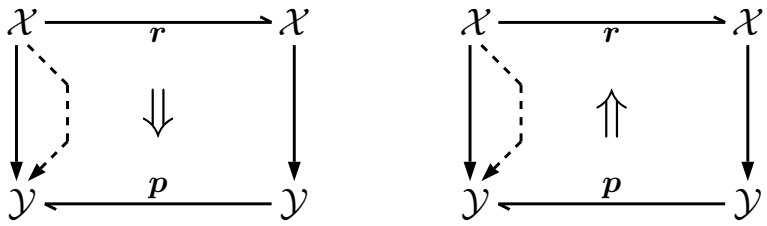


Figure 2: Demand-driven protocols vs supply-driven protocols

samples, and influence the buyer Alice to request to buy more. In a demand-based economy, the buyers stroll the market place and request to buy the good that the sellers display, in a supply-based economy, the buyers are passive, and the sellers supply their goods, and try to create the demand, e.g. by presenting to the buyers previously unknown goods. But even in the traditional market place, neither side is passive, and the demand and supply are actively balanced to maximize profits on each side: the sellers look for buyers, the buyers shop around, they all haggle, they try to outsmart each other, which gives rise to deception.

3.3 Examples of basic protocols

Example 1: Let $\varphi : \mathcal{R}\times\mathcal{C} \longrightarrow \mathcal{Y}$ be the database of a credit rating agency Exavier, presented as a resource in the following format:

- $\mathcal{R} = \mathcal{S}$, i.e. the database rows correspond to the network nodes;
- $\mathcal{C} = \{T_0, T_1, \ldots, T_n\}$ are the types of financial and other relevant transactions;
- $\mathcal{X} = \mathcal{S}\times\mathcal{C}$, i.e. the public identifiers $x \in \mathcal{X}$ are in the form $x = \langle A, T_i\rangle$, or T_i^A, denoting a type of Alice's transactions;
- $\mathcal{X}_S = \{T_0^S, T_1^S, \ldots, T_n^S\}$, i.e. $\pi_S(T_i^A) = T_i^A$ if $S = A$ or $S = E$, otherwise it is undefined, meaning that by $[\![A]\!] = [\![A]\!]\pi_A$ Alice can see her own record, and Exavier can see all records;
- $\mathcal{Y} = \coprod_{i=0}^n T_i^*$ are the transaction history listings, unencrypted, and
- $\mathcal{Y}_S = \mathcal{Y}$, i.e. $\rho_S(y) = y$ for all $y \in \mathcal{Y}$ makes any released transaction listing readable to anyone.

If Alice is a lender and Bob has requested a loan, then Alice may request access to some of his credit history by $\boldsymbol{r}^{AB} : \mathcal{X} \rightharpoonup \mathcal{X}$ with $\boldsymbol{r}^{AB}(x) = x$ if $x \in \mathcal{X}_B$, otherwise undefined. Bob's privacy policy $\boldsymbol{p}^{AB} : \mathcal{Y} \rightharpoonup \mathcal{Y}$ will then determine which transaction types T_i will be released to Alice by setting $\boldsymbol{p}^{AB}(t) = t$ when $t \in T_i^*$ for some chosen values of i, otherwise

undefined; or he may set his policy to release just some partial information about some of the transaction types. To prevent that Bob tampers with this information, this privacy protocol for obtaining loan applicants' credit rating information is in practice combined with an authentication protocol, where Bob's transaction record $t \in T_i^*$ is released to Alice not by Bob, but by the credit rating agency Exavier himself, upon Bob's approval. Exavier then uses the opportunity to record Alice's request in Bob's credit rating, and thus expands his shared database.

In any case, Alice the lender is provided the resource $[\![AB]\!] = \rho^{AB}[\![B]\!]\pi^{AB}$, controlled by Bob and Exavier. If Bob has joint accounts with Carol, Alice may request and Bob and Exavier may provide some information about Carol. Alice will then conjoin the obtained information with her own information and resource $[\![A]\!]$ and use the compiled information $[\![A]\!]\widehat{\Upsilon}[\![AB]\!]$, or maybe $[\![A]\!]\widehat{\Upsilon}[\![AB]\!]\widehat{\Upsilon}[\![ABC]\!]$, to make her lending decision. Note that the described Request-Policy protocol is thus embedded within another RP-protocol, where Bob requests from Alice a loan, and Alice requests from Bob some private data, and uses the obtained information resources $[\![A]\!]\widehat{\Upsilon}[\![AB]\!]\widehat{\Upsilon}[\![ABC]\!]$ as the input into her loan policy, which outputs the loan decision. The privacy protocol for credit rating is thus composed with an authentication protocol for data release, and embedded into a privacy protocol for loan provision.

Example 2: The web as a resource $\omega : \mathcal{X} \longrightarrow \mathcal{Y}$ is controlled by Bob through $\pi_B : \mathcal{X} \rightharpoonup \mathcal{X}$, which filters the set of URLs $\mathcal{X}_B$ under Bob's control, locating his web site. The projector $\rho_B : \mathcal{Y} \rightharpoonup \mathcal{Y}$ determines what is published on his web site $[\![B]\!] = \rho_B\omega\pi_B$. If Alice surfs or navigates to Bob's web site, she submits a request $\boldsymbol{r}^{AB} : \mathcal{X} \rightharpoonup \mathcal{X}$ for some of Bob's URLs. Bob may then request some of her identifiers, and maybe some of her private data to authenticate her, and input the obtained information into his policy $\boldsymbol{p}^{AB} : \mathcal{Y} \rightharpoonup \mathcal{Y}$ which generates and outputs Bob's web content $[\![AB]\!] = \boldsymbol{p}^{AB}[\![B]\!]\boldsymbol{r}^{AB}$ in response to Alice's request. Alice can now add what she got from Bob to her own resource $[\![A]\!] = \rho_A\omega\pi_A$, and use $[\![A]\!]\widehat{\Upsilon}[\![AB]\!]$ to serve instances of her own web site, when requested in other runs of the web protocol.

Example 3: If Bob owns the search engine $\gamma : \mathcal{X} \longrightarrow \mathcal{Y}$, then his private resource is simply $[\![B]\!] = \gamma$. If anyone can submit any search term from $\mathcal{X}$, and if all pages indexed in $\mathcal{Y}$ are publicly accessible, then $\mathcal{X}_S = \mathcal{X}$ and $\mathcal{Y}_S = \mathcal{Y}$ for all subects S on the network $\mathcal{S}$. A session of the RP-protocol thus consists of a query, represented as the partial function $\boldsymbol{r}^{AB} : \mathcal{X} \rightharpoonup \mathcal{X}$, undefined everywhere except on some search term $x = \boldsymbol{r}^{AB}(x)$; and a reply according to a policy $\boldsymbol{p}^{AB} : \mathcal{Y} \rightharpoonup \mathcal{Y}$. In modern search engines, the search results are *personalized*, in the sense that our search engine Bob tailors his response especially for Alice. This may mean that Bob's policy $\boldsymbol{p}^{AB}$ is to display just what is of interest for Alice, or what she wants to hear. To achieve this, $\boldsymbol{p}^{AB}$ skews the sample of the source element $\omega_x \in \mathcal{D}\mathcal{Y}$, output by the search engine $\omega : \mathcal{X} \longrightarrow \mathcal{Y}$ in response to the query x; *or* it modifies the induced preference ordering $\overset{\omega_x}{\lhd}$ of the web pages in $\mathcal{Y}$, according to which the search results are displayed to Alice. This preference ordering is based on the ranking of the web content according to

relevance and quality of the provided information [40, 41].

3.4 Composing privacy protocols

Security protocols are often composed [36, 43, 44]. The goal of security protocol composition is to preserve and conjoin the security properties of the protocol components [14, 16, 18]. The problem is that security properties are usually not compositional [17, 22]. The problem with privacy protocols is even more subtle, and in fact intentionally mind-boggling. The goal of privacy protocol composition is often to combine the low-level privacy protocol components in such a way that their privacy properties are subverted, and sublimated into different privacy properties. Alice's privacy goals of the low-level protocol components are replaced by Eve's privacy goals of high-level composite protocols. We spell out some examples, and make a first couple of steps towards analyzing this protocol knot.

Example 3 continued: In general, Bob does not own the search engine, but only his own web pages: his resource $[\![B]\!] : \mathcal{X} \longrightarrow \mathcal{Y}$ is just a map from $\mathcal{X}_B \subseteq \mathcal{X}$ (which can be thought of as the content menu of his web site, filtered by $\pi_B : \mathcal{X} \rightharpoonup \mathcal{X}$ from the universe of keywords $\mathcal{X}$) to $\mathcal{Y}$, containing the actual content of the pages that he runs. When Alice visits Bob's web page and submits request $\boldsymbol{r}^{AB}$ for some content from his site, he responds according to his policy $\boldsymbol{p}^{AB}$, and displays $[\![AB]\!] = \boldsymbol{p}^{AB}[\![B]\!]\boldsymbol{r}^{AB}$. Moreover, in addition to providing the content that Alice explicitly requested, Bob can also try to sell a fragment of Alice's expressed interest to advertisers, and provide her with some content that she did not explicitly request, but may be related. The search for the advertisers willing to buy a fragment of Alice's interest is a second use that the owner of the search engine, whom we call Gogol[4], will find for his web index $\gamma : \mathcal{X} \longrightarrow \mathcal{Y}$. He will thus extract from γ two different resources:

- the search index $[\![G_{se}]\!] : \mathcal{X} \longrightarrow \mathcal{Y}$, assigning to each search term from $\mathcal{X}$ a source element of web contents from $\mathcal{Y}$, but this time not informative but advertising contents; and on the other hand
- the advertising index $[\![G_{ad}]\!] : \mathcal{Z} \longrightarrow \mathcal{X}$, assigning to each advertising opportunity[5] a source element of related search terms.

These two resources are used as follows. Gogol initiates pooling of his and Bob's resources by offering in $\boldsymbol{r}^{GB}$ some advertising opportunities from $\mathcal{Z}$. Bob creates some advertising space $[\![B_{ad}]\!] : \mathcal{Z} \longrightarrow \mathcal{X}$, ready to be inserted into his web site $[\![B]\!] : \mathcal{X} \longrightarrow \mathcal{Y}$, and accepts Gogol's collaboration proposal by $\boldsymbol{p}^{GB}$. Gogol processes the provided part $[\![GB]\!] = \boldsymbol{p}^{GB}[\![B_{ad}]\!]\boldsymbol{r}^{GB}$ of Bob's resource, and uses $[\![GB]\!]\widehat{\Upsilon}[\![G_{ad}]\!]$ to determine which of the advertising opportunities from $\mathcal{Z}$ will best suit Bob's site. When Bob receives Alice's request $\boldsymbol{r}^{AB}$, he forwards it to

[4]Nikolai Vasilievich Gogol was a XIX century Russian writer. Gogols are also the ape-like enemies in the video game *Xenoblade Chronicles*.

[5]Gogol receives advertising requests in a separate privacy protocol. It will be briefly discussed in the next section.

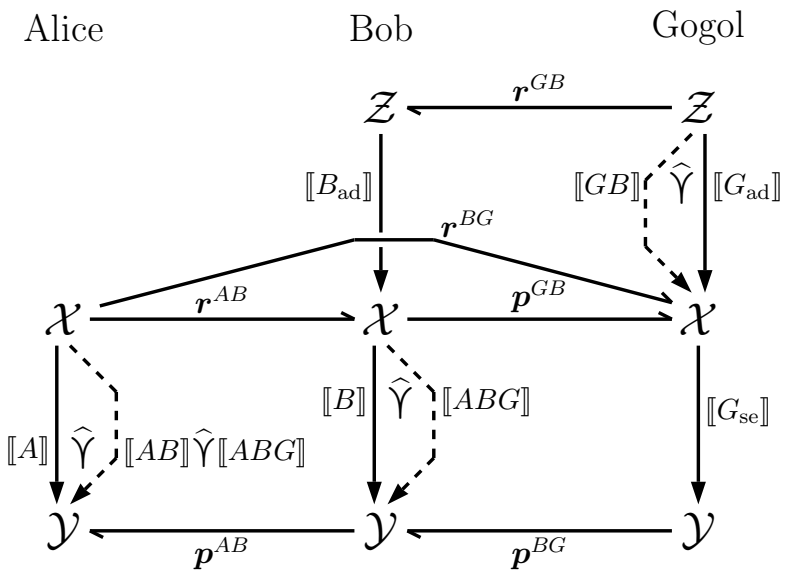


Figure 3: Composite privacy protocol: Targeted Advertising (TAd)

Gogol as his request $\boldsymbol{r}^{GB} = \boldsymbol{r}^{BG}\boldsymbol{r}^{AB}$ for the ad content. Using the current search index $[\![G_{se}]\!]$ of advertising contents and the index $[\![GB]\!]\widehat{\Upsilon}[\![G_{ad}]\!]$ of advertising topics suitable for Bob's site, Gogol provides Bob with the targeted web ad $[\![ABG]\!] = \boldsymbol{p}^{AB}\boldsymbol{p}^{BG}[\![G]\!]\boldsymbol{r}^{BG}\boldsymbol{r}^{AB}$, deemed to be of interest for Alice because of her interest in Bob's web content. Bob then displays $[\![B]\!]\widehat{\Upsilon}[\![ABG]\!]$, and Alice receives $[\![AB]\!]\widehat{\Upsilon}[\![ABG]\!]$. This composite protocol is displayed in the bottom row of Fig. 3. The protocol component where Gogol pays Bob for displaying the ad is omitted, as is the component where the advertiser pays Gogol, and the one where Alice requests to purchase the advertised goods, and the one where she remits the payment to the merchant, etc. We shall see some such protocols in the next section, but the mosaic of protocols for sharing and trading private resources always spreads beyond the horizon.

Example 4: While Gogol the search engine builds ranked indices of the content existing on the web, and shares these resources with Alice, Bob, and Carol, Zuck the social engine elicits the content from Alice, Bob, and Carol, builds an index of that, and shares their content with them as their social interactions. A bird's eye view of a fragment of this process is displayed in Fig. 4. Zuck's requests $\boldsymbol{r}^{ZS}$ to all $S \in \mathcal{S}$ offer to index and distribute everyone's media and contents. In response he receives $[\![ZS]\!] = \boldsymbol{p}^{ZS}[\![S]\!]\boldsymbol{r}^{ZS}$ from all $S \in \mathcal{S}$, which are some of their private media and content that they want to share with their social network. To better distribute them, Zuck conjoins the shared resources into his main resource, the social engine index

$$\zeta \;=\; [\![Z_{se}]\!] \;=\; \widehat{\Upsilon}_{S\in\mathcal{S}}[\![ZS]\!] \;=\; \widehat{\Upsilon}_{S\in\mathcal{S}}\boldsymbol{p}^{ZS}[\![S]\!]\boldsymbol{r}^{ZS}$$

and shares a part of it with Alice

$$[\![AZ]\!] \;=\; \boldsymbol{p}^{AZ}[\![Z_{se}]\!]\boldsymbol{r}^{AZ} \;=\; \boldsymbol{p}^{AZ}\left(\widehat{\Upsilon}_{S\in\mathcal{S}}\boldsymbol{p}^{ZS}[\![S]\!]\boldsymbol{r}^{ZS}\right)\boldsymbol{r}^{AZ}$$

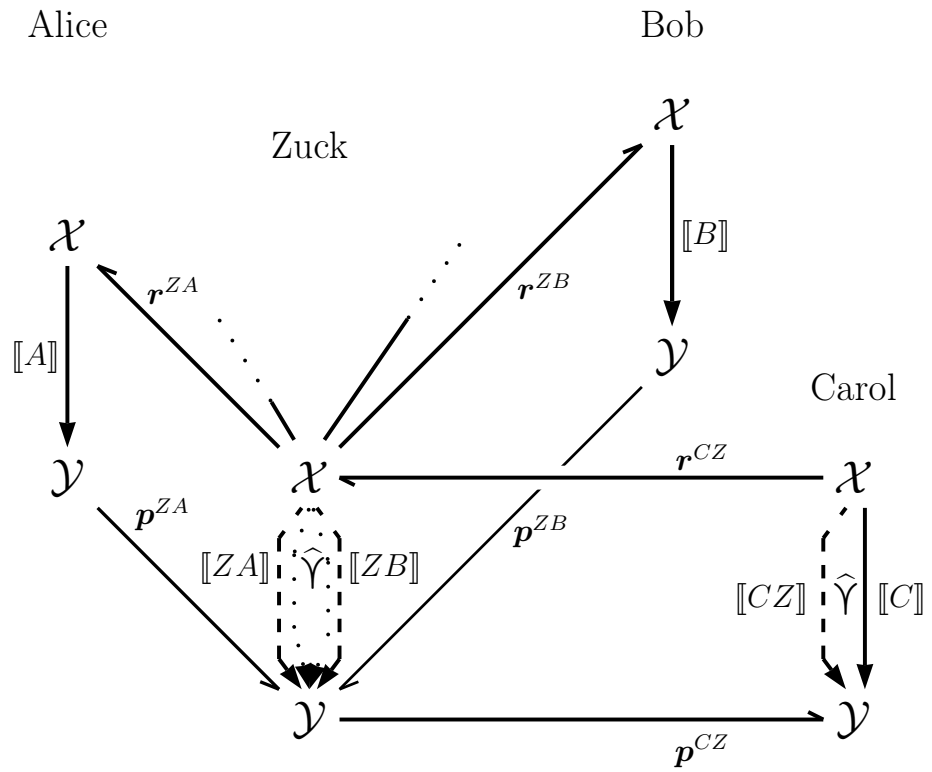


Figure 4: Composite privacy protocol: Social Networking (SNet)

The idea is that Zuck's resource $[\![AZ]\!]$ shared with Alice consists of Alice's friends' private contents processed by Zuck. As a participant who only relays messages, Zuck plays the role of a legitimate Man-in-the-Middle in this privacy protocol. The Man-in-the-Middle pattern is usually an attack strategy, and not a protocol role. Protocols sometimes use a Trusted Third Party for a particular functionality. However, Zuck is here more than a Trusted Third Party (TTP), because he does not provide a particular functionality, but processes *all* protocol content; he is also less than a TTP, because he does not originate any content. Most importantly, he fuses all protocol content into his private resource $\zeta = [\![Z_{se}]\!] : \mathcal{X} \longrightarrow \mathcal{Y}$. Like in a search engine, the weights of the source elements $\zeta_x \in \mathcal{D}\mathcal{Y}$ determine a preference ranking of the content.

Zuck's privacy policy for Carol $\boldsymbol{p}^{CZ}$ determines which of Alice's, Bob's and Dave's postings Carol will see, and in which order. Carol's request $\boldsymbol{r}^{CZ}$ can filter out what she does not want to see; but Carol cannot use $\boldsymbol{r}^{CZ}$ to request what she will see, because Zuck's index is not available to her.

Zuck's resource $[\![Z_{se}]\!]$ and all of his policies $\boldsymbol{p}^{SZ}$ also take into account Alice's and Bob's privacy policies $\boldsymbol{p}^{ZA}$ and $\boldsymbol{p}^{ZB}$, which may specify whether Carol should have access to their content. However, the existing social engines provide policy languages for $\boldsymbol{p}^{ZS}$ at the expressiveness level of 1960s operating systems, before the concepts of access control were introduced, where only a small number of fixed access control policies are available. The

users are usually offered to either make their postings available to general public, or to all of their private contacts ("friends"), and sometimes also to all of their contacts' contacts ("friends' friends"). The option of establishing subgroups and hierarchies of private contacts is not offered. The option of sharing content with specific subgroups is offered as a separate functionality, but the recipients must be reentered with each message, or kept in an address book, which is separate from privacy policies.

While more expressive access control systems, allowing refined privacy policies, would surely vastly enhance usability of the social engine for Alice and Bob, allowing them to manage their content using the access control mechanisms that they got used to while using their computers, this would leave Zuck with narrower choices of his policies. And Zuck, of course, monetizes his services by manipulating the weight biases in his social engine index $[\![Z_{se}]\!] : \mathcal{X} \longrightarrow\!\!\!\!\!\rightarrow \mathcal{Y}$, and by selling the choices what to display to each user. These choices are expressed by his policy $\boldsymbol{p}^{SZ} : \mathcal{Y} \longrightarrow\!\!\!\!\!\rightarrow \mathcal{Y}$ for each user $S \in \mathcal{S}$. The more privacy choices are made by Alice and Bob as content originators, the fewer privacy choices are left to Zuck as the content aggregator and monetizer.

4 Local privacy failures: Covert channels

4.1 Privacy protocol failures

A privacy protocol establishes an *overt* private channel to distribute private resources. In the basic RP-protocol in Fig. 1, Bob releases some of his private resource to Alice. In the composite privacy protocols, multiple resources can be shared or exchanged between multiple subjects along multiple overt channels. Privacy protocol failures occur when, in addition to the overt channels, some *covert* channels are also established. In the simplest case, a covert channel from Bob to Alice arises when Bob intends to release to Alice the part $[\![AB]\!] = \boldsymbol{p}^{AB}[\![B]\!]\boldsymbol{r}^{AB}$ of his resource $[\![B]\!]$, but Alice manages to extract in $[\![A]\!]\widehat{\curlyvee}[\![AB]\!]$ a bigger part of $[\![B]\!]$. Bob's privacy problem with respect to Alice is thus to constrain his policy $\boldsymbol{p}^{AB}$ to assure that $[\![AB]\!] = \boldsymbol{p}^{AB}[\![B]\!]\boldsymbol{r}^{AB}$ satisfies the requirement

$$\Big([\![A]\!]\widehat{\curlyvee}[\![AB]\!]\Big) \curlywedge [\![B]\!] \;\prec\; [\![AB]\!] \tag{3}$$

Note that (3) does not require that Alice is prevented from extracting some of Carol's resource from the interaction with Bob, which she could achieve by cross-referencing $[\![AB]\!]$ with some information about $[\![C]\!]$ previously stored in $[\![A]\!]$. Requirement (3) also does not require that Dave is prevented from extracting some of Bob's resource from a later interaction with Alice, who may make some of $[\![AB]\!]$ available to him. Requirement (3) just expresses Bob's *local* privacy goal, concerning the present channel between him and Alice. The stronger requirement

$$[\![A]\!]\widehat{\curlyvee}[\![AB]\!] \;\prec\; [\![AB]\!] \tag{4}$$

would, of course, be more socially responsible; but Bob only has access to $[\![B]\!]$, no direct access to $[\![A]\!]$, and he has therefore no way to observe or preclude any crossreferences that Alice might be able establish between $[\![A]\!]$ and $[\![AB]\!]$, except those that he observes in $[\![B]\!]$. If Bob could enforce (4), then he could realizing the Dalenius' desideratum, discussed in Sec. 1. The non-local privacy protocol failures, resulting from *non-local covert channels*, will be discussed in the next section. In this section, our considerations are limited to *local covert channels* that may arise from Bob to Alice, in parallel with the overt channel set up by an RP-protocol like in Fig. 1.

But even for this, there is more to the story than (3). A covert channel may arise even if there is no overt channel at all. Take the trivial case, where Bob's policy is to not share anything with Alice, which is expressed by the everywhere undefined function $\boldsymbol{p}^{AB} = \emptyset$ giving $[\![AB]\!] = \emptyset$. Requirement (3) thus boils down to

$$[\![A]\!] \curlywedge [\![B]\!] \;\prec\; \emptyset$$

This is easily assured by $\mathcal{Y}_A \cap \mathcal{Y}_B = \emptyset$ and $\mathcal{X}_A \cap \mathcal{X}_B = \emptyset$. But if a resource is shared, then the users may submit their inputs in turns, get their outputs in turns, yet there will be *interferences*.

Example of interference. Suppose that Alice, Bob and Carol live in the same highrise building and share an elevator. Each of them inputs an elevator call, and then each of them receives the elevator service. During the service, the elevator is not available. After each service, the elevator cabin may be left at a different floor. By observing the state of the elevator (its location, whether it is currently moving, etc.), Alice can obtain information about Bob's and Carol's interactions with the elevator; by interfering with its state, she can make it unavailable for them. The elevator can thus provide a covert channel between Alice, Bob and Carol just by the virtue of being shared, without providing an overt channel between them.

Covert channels. Even if Alice's, Bob's and Carol's private inputs come from disjoint sets $\mathcal{X}_A$, $\mathcal{X}_B$ and $\mathcal{X}_C$ (which is the case if the elevator authenticates calls, say by requiring room cards in a hotel, or biometric checks in an appartment building), the state of the elevator in general depends on a whole history of private inputs. Suppose that each user $S \in \mathcal{S}$ has a devoted set of actions Σ_S, so that only Alice can call the elevator to her penthouse, because only Σ_A contains that action. If $\Sigma_A \cap \Sigma_B = \emptyset$, then it follows that there is no overt channel between Alice and Bob, because all sequences of inputs that Alice alone or Bob alone may enter are also disjoint, i.e. $\mathcal{X}_A \cap \mathcal{X}_B = \emptyset$ also holds for

$$\mathcal{X}_S \;=\; \Sigma_S^*$$

But the actual inputs that the elevator receives are usually not from any user alone, but they are shuffled in

$$\mathcal{X} \;=\; \left(\coprod_{S \in \mathcal{S}} \Sigma_S \right)^*$$

The projections are thus in the form

$$\mathcal{X}_A = \Sigma_A^* \underset{\underline{\pi}_A}{\overset{\overline{\pi}_A}{\rightleftarrows}} \left(\coprod_{S\in\mathcal{S}} \Sigma_S\right)^* = \mathcal{X}$$

where

- $\overline{\pi}_A(x) \;=\; \begin{cases} \langle\rangle & \text{if } x = \langle\rangle \\ \sigma :: \overline{\pi}_A(x') & \text{if } x = \sigma :: x' \text{ and } \sigma \in \Sigma_A \\ \overline{\pi}_A(x') & \text{if } x = \sigma :: x' \text{ and } \sigma \notin \Sigma_A \end{cases}$
- $\underline{\pi}_A = \iota_A^*$ lifts the inclusion $\iota_A : \Sigma_A \hookrightarrow \coprod_{S\in\mathcal{S}} \Sigma_S$ to the sequences.

An interference of Alice's use of the elevator with Bob's and Carol's use occurs if Alice observes two situations $x, x' \in \mathcal{X}$ such that

- $\pi_A(x) = \pi_A(x')$ — her own inputs are the same, but
- $\rho_A[\![A]\!](x) \neq \rho_A[\![A]\!](x')$ — the elevator produces different outputs,

where $\rho_A = \underline{\rho}_A \overline{\rho}_A : \mathcal{Y} \rightharpoonup \mathcal{Y}$ is the projector from (1). The difference must be caused by Bob's or Carol's different inputs within the strings x and x'. Their inputs are not directly observable for Alice, and she may not know what they are, or who has entered them; but the fact that they are different gives Alice a single bit of information about Bob's and Carol's actions. Hence the covert channel.

Noninterference. The task of recognizing such covert channels and the methods to eliminate them have been extensively studied in system security, and led to the requirement of *noninterference*, and a whole range of related properties [25, 32, 33, 34, 48, 52]. A simple way to state the noninterference requirement is in terms of the equivalence relations

$$\begin{array}{rcl} x \lfloor A \rfloor x' & \iff & \pi_A(x) = \pi_A(x') \text{ and} \\ x \lceil A \rceil x' & \iff & \rho_A[\![A]\!](x) = \rho_A[\![A]\!](x') \end{array}$$

The noninterference requirement is then simply

$$x \lfloor A \rfloor x' \quad \Longrightarrow \quad x \lceil A \rceil x' \tag{5}$$

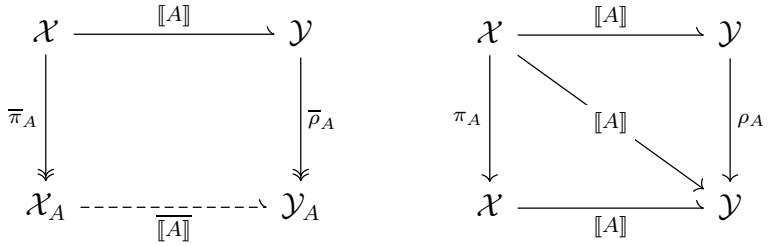
This is a very strong requirement: that $[\![A]\!]$ does not interfere with anyone else's inputs or outputs, except Alice's. The version localized to Alice and Bob is obtained by defining

$$\begin{array}{rcl} x \lfloor AB \rfloor x' & \iff & \rho_A[\![AB]\!](x) = \rho_A[\![AB]\!](x') \\ x \lceil AB \rceil x' & \iff & \rho_A\Big(\left([\![A]\!] \mathbin{\widehat{\Upsilon}} [\![AB]\!]\right) \curlywedge [\![B]\!]\Big)(x) \;=\; \rho_A\Big(\left([\![A]\!] \mathbin{\widehat{\Upsilon}} [\![AB]\!]\right) \curlywedge [\![B]\!]\Big)(x') \end{array}$$

and requiring

$$x \lfloor AB \rfloor x' \quad \Longrightarrow \quad x \lceil AB \rceil x'$$

For deterministic resources, i.e. when $[\![A]\!]$, $[\![AB]\!]$ etc. are partial functions $\mathcal{X} \rightharpoonup \mathcal{Y}$, it is easy to see that e.g. requirement (5) is equivalent to the existence of a function $\overline{[\![A]\!]}$ that makes the diagram on the left-hand side of (1) commute.



(6)

Recalling the projections from (1), it is clear that the commutativity of the diagram on the right-hand side of (6) also provides an equivalent form of the noninterference requirement. Yet another equivalent form is that $[\![A]\!] = \rho_A [\![A]\!] \pi_A$.

For general resources from Def. 2.2, the requirements from (6) can still be imposed, but validating that the probability distributions of source elements are equal on the nose is almost never feasible in practice. Requiring that diagrams (6) commute up to ε takes us in the direction of *differential privacy* [23], which is a popular and practical tool for dealing with covert channels. In the context of privacy protocols, its effectiveness, at least in the original form, seems to be limited to *local* covert channels. — But what are non-local covert channels?

5 Nonlocal privacy failures: Covert protocols

The rapid transformation of the socio-technical context of privacy is an acute socio-political problem, which has been described from many angles [1, 5, 8, 20, 59]. The idea of this research is to shed some light on the problem of privacy by analyzing privacy protocol problems. In this section, we attempt to outline the unusual shape of some of the typical privacy protocol problems. They are unusual in that their vulnerabilities are not design flaws that open some attack vectors for outside attackers, as they do in security protocols. Typical privacy protocol vulnerabilities seem inherent to protocols themselves. Not that the protocols always have back doors; but they have transparent roofs. They are not vulnerable to subversions, or level-below attacks, but to sublimations, or level-above applications.

5.1 Man-in-the-Middle protocols

We have seen that Zuck was an MitM in the SNet protocol on Fig. 4, where his private resource $[\![Z_{se}]\!]$ is inserted in-between everybody else's private resources, as their fusion; and that Bob is an MitM in the TAd-protocol on Fig. 3, where he is inserted in-between Gogol and Alice, and serves Gogol's targeted ads to Alice. A level above SNet and TAd, both Zuck

and Gogol are MitMs in the privacy protocols where they monetize their services (social networking and web search, respectively) by inserting themselves in-between the advertisers and their targets (usually voters or consumers). Zuck's protocol for monetizing his social engine through Social Influencing (SInf) is displayed in Fig. 5. This is a level-above protocol in the sense that it is built on top of the SNet protocol, which it uses as an encapsulated procedure. In Fig. 5, we only display a single RP-component of SNet: the rectangle on the left, where Zuck provides Carol her friends' postings. We represent SNet by its single component only because we do not have a good diagrammatic notation to encapsulate a low-level protocol as an atom of a high-level protocol. SInf uses SNet on the left to leverage its

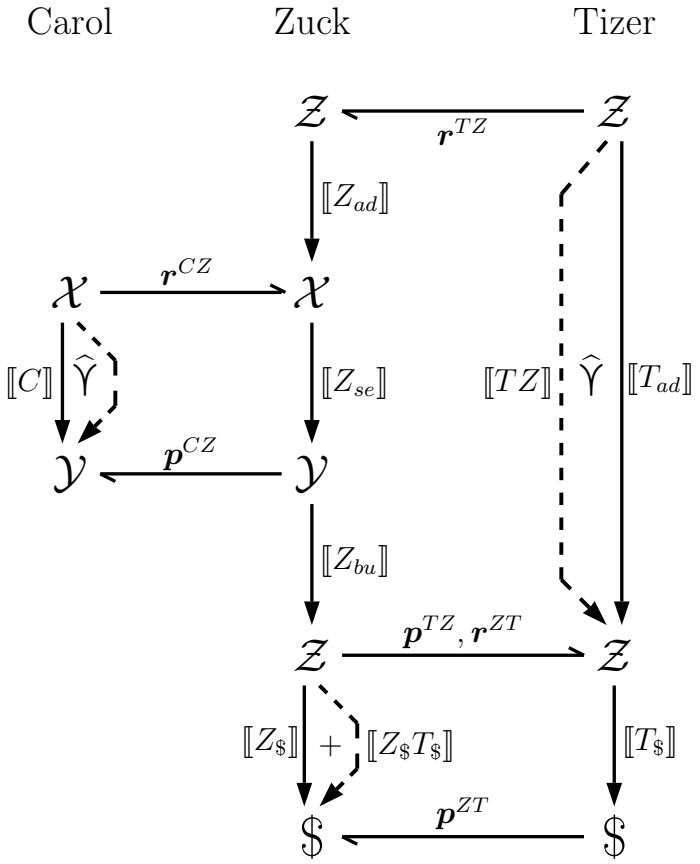


Figure 5: Level-above privacy protocol: Social Influencing (SInf)

business process on the right. Tizer is an advertising campaign manager, and his resource $[\![T_{ad}]\!] : \mathcal{Z} \longrightarrow \mathcal{Z}$ is a campaign strategy, presented as a Markov chain over the a set of campaign messages $\mathcal{Z}$. Tizer initiates an SInf protocol run with $\boldsymbol{r}^{TZ}$, requesting that Zuck places the campaign messages of interest for Tizer at that moment. Zuck runs the request through his advertising index $[\![Z_{ad}]\!] : \mathcal{Z} \longrightarrow \mathcal{X}$, assigning to each campaign message from $\mathcal{Z}$ a suitable context of recipients and their postings in $\mathcal{X}$, and his social engine $[\![Z_{se}]\!] : \mathcal{X} \longrightarrow \mathcal{Y}$ then inserts the campaign messages among Carol's friends' postings. Zuck's business engine $[\![Z_{bu}]\!] : \mathcal{Y} \longrightarrow \mathcal{Z}$ then generates the service report $\boldsymbol{p}^{TZ}$ and the invoice $\boldsymbol{r}^{ZT}$ for Tizer, who

responds with the payment $\boldsymbol{p}^{ZT}$ from his advertising budget $[\![T_\$]\!]$. The payment $[\![ZT]\!]$ is added to Zuck's budget $[\![Z_\$]\!]$.

5.2 Resource inflation, privacy deflation

Dave would like to make pancakes, but he does not have either milk or eggs, so he needs to borrow from Alice, who is his neighbor. But he is shy by nature, and he already borrowed many things. So to avoid intruding into Alice's privacy even more, Dave asks Carol to ask Alice for milk, and he asks Bob to ask her for eggs. If Dave is also short of flour and oil, he can also ask Elizabeth and Frank to knock on Alice's door and ask for that. Then he collects it all, and makes pancakes. This *cross-sharing* privacy protocol is displayed in Fig. 6. Dave

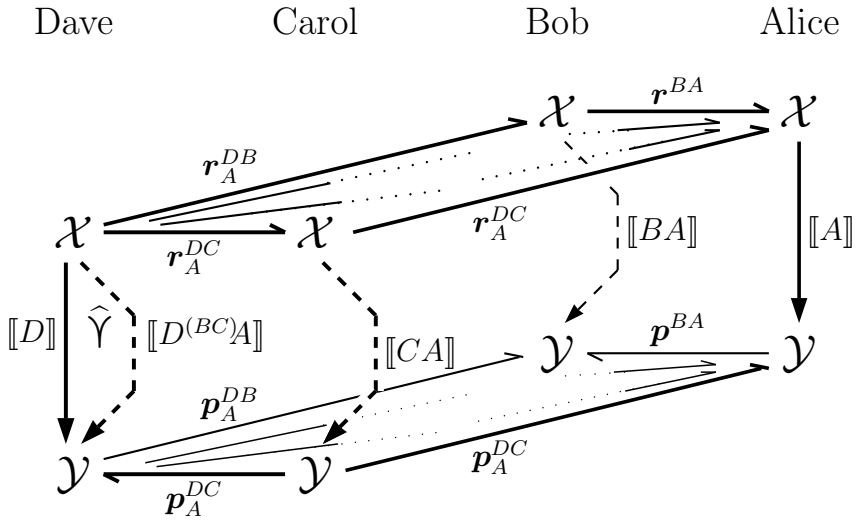


Figure 6: Resource inflation: Cross-referencing and cross-sharing

developed this protocol when he worked as a police detective. Alice was often a suspect in his investigations, so whenever he needed to check the details of Alice's alibi, it turned out to be better to ask Alice's friends, than to ask her directly. By conjoining the information obtained from others, he would not only get more details, without drawing attention with too many requests at once, but he could also *cross-reference* the different details that Alice provided to different people. If she tells one person one thing, and another person another thing, then Dave would detect an inconsistency in Alice's alibi. While a fusion of inconsistent sources eliminates the inconsistent parts, and thus contains less information than either of the original sources, the inconsistency itself becomes a *higher-order information*, as mentioned at the end of Sec. 2. Level-above attacks and protocols sublimate entire low-level protocol sessions as higher-order information, and manipulate it towards some privacy goals. The loss of a private resource for Alice is a gain of a private resource for Dave. Having detected an inconsistency in Alice's alibi by asking around in a cross-referencing protocol, like the alibi check interpretation of Fig. 6, Dave confronts Alice himself in a level-above protocol, and requests more higher-order information, to eliminate the inconsistencies from the acquired

resources. An analogous level-above sublimation of the cross-sharing protocol in the pancake sources interpretation of Fig. 6 would be that Dave asks Bob, Carol, Elizabeth and Frank to leave the milk, egg, flour and oil that they got for him from Alice — to leave if all for him with Alice. Then he could collect it all simply by asking Alice if the neighbors by any chance left anything for him.

The problem of cross-referencing is a well-studied privacy problem, because it arises in statistical databases [29, 37, 54]. Statistical databases anonymize their records, and make them publicly available for statistical analyses. The problem is that Alice's data may be recorded in several statistical databases, one owned by Bob, another one by Carol, another one by Elizabeth, and Frank; and Alice's data may be anonymized differently in each case: Bob may omit Alice's address, Carol her date of birth, Elizabeth her phone number. Now Dave may link Alice's records in Bob's and Carol's databases by the phone number, in Carol's and Elizabeth's databases by the address, etc. He may then rearrange data according to a common ordering, and align the anonymized records so that the gap on each of them is filled by the date in another one. This is a special case of the fusion operation operation described in Sec. 2.3. On the other hand, the attack is clearly an instance of the privacy protocol in Fig. 6. Re-identifying someone from statistical databases is similar to borrowing the pancake ingredients from a neighbor, and to checking an alibi.

Large-scale versions of the cross-sharing protocol are nowadays routinely launched as level-above attacks over many services and applications. E.g. the SNet protocol from Fig. 4 is often sublimated into an instance of a cross-sharing attack from Fig. 6: providers leverage requests for ongoing access to private channels against users' requests for a single service. In a much publicized incident, a psychology lecturer from Cambridge University, let us call him Dave, designed a Facebook app to collect user data for research purposes [28, 24], which offered a simple personality quiz, in exchange for full access to users' contact lists. The quiz was taken by 270,000 users, some of them called Bob or Carol, who unwittingly delivered the profiles of 87,000,000 of their contacts, some of them called Alice. The harvested profiles were sold to a political consultancy, Cambridge Analytica [58], and were used to a great effect in a level-above instance of the SInf protocol from Fig. 5.

6 From security protocol analysis to privacy protocol analysis

Security is a useful property. Security requirements are usually publicly declared, with a clear utility. Security protocols implement explicit security requirements. Security protocol failures are unintended protocol runs missed by the protocol designers, usually because of the complexities of the underlying distributed algorithms. The goal of security protocol analysis is to outrun the attackers in detecting and eliminating the unintended runs.

Privacy is a right. But my right to privacy may counter your right to privacy. Private rights

need to be balanced against one another, and the privacy requirements are not always agreed upon, or publicly declared. Privacy protocols often implement implicit privacy requirements, sublimated to their level-above deployments. In the realm of privacy, the task of protocol analysis is not any more just to detect the unintended failures with respect to declared requirements. In the realm of protocols that implement private utility, the task of protocol analysis is also to detect the intended sublimated protocol runs that arise from undeclared protocol requirements. Privacy protocol analysis thus provides a technical underpinning for the process of balancing information and value distributions in network society [9, 21, 59].

Given a security protocol, we strive to prove that its declared security requirements are enforced, or to uncover any attacks that may exist. Given a privacy protocol, the task is to establish whose privacy requirements it implements. Alice's and Bob's privacy requirements often clash, and the boundary between protocols and attacks is blurred. At the level above, though, instead of attacks, there are now deceptions to be uncovered and analyzed.

References

[1] Alessandro Acquisti, Stefanos Gritzalis, Costas Lambrinoudakis, and Sabrina di Vimercati. *Digital Privacy: Theory, Technologies, and Practices*. CRC Press, 2007.

[2] Peter M. Alberti and Armin Uhlmann. *Stochasticity and partial order: double stochastic maps and unitary mixing*. Mathematics and its applications. Deutscher Verlag der Wissenschaften, 1981.

[3] Tsuyoshi Ando. Majorization, doubly stochastic matrices, and comparison of eigenvalues. *Linear Algebra and its Applications*, 118:163 – 248, 1989.

[4] A. Angela and G. Conti. *A Day in the Life of Ancient Rome*. Europa Editions, 2009.

[5] Julia Angwin. *Dragnet Nation: A Quest for Privacy, Security, and Freedom in a World of Relentless Surveillance*. Henry Holt and Company, 2014.

[6] Hannah Arendt. *The Human Condition*. Charles R. Walgreen Foundation lectures. University of Chicago Press, second edition, 1998.

[7] Joe Bailey. From public to private: The development of the concept of "private". *Social Research*, 69(1):15–31, 2002.

[8] Kirstie Ball, Kevin Haggerty, and David Lyon. *Routledge Handbook of Surveillance Studies*. Routledge International Handbooks. Taylor & Francis, 2012.

[9] Yochai Benkler. *The Wealth of Networks: How Social Production Transforms Markets and Freedom*. Yale University Press, 2006.

[10] G. Birkhoff. Tres observaciones sobre el algebra lineal. *Univ. Nac. Tucumán Rev. Ser. A*, 5:147–151, 1946.

[11] Garrett Birkhoff. Tres observaciones sobre el algebra lineal. *Univ. Nac. Tucuma?n Rev. Ser. A*, 5:147–151, 1946.

[12] Felix Brandt, Vincent Conitzer, Ulle Endriss, Jérôme Lang, and Ariel D. Procaccia. *Handbook of Computational Social Choice*. Cambridge University Press, 2016.

[13] S. Burke. *Delos: Investigating the Notion of Privacy Within the Ancient Greek House*. PhD thesis, University of Leicester, 2000.

[14] Iliano Cervesato, Catherine Meadows, and Dusko Pavlovic. An encapsulated authentication logic for reasoning about key distribution protocols. In Joshua Guttman, editor, *Proceedings of CSFW 2005*, pages 48–61. IEEE, 2005.

[15] Tore Dalenius. Towards a methodology for statistical disclosure control. *Statistik Tidskrift*, 15:429–444, 1977.

[16] Anupam Datta, Ante Derek, John Mitchell, and Dusko Pavlovic. Secure protocol composition. *E. Notes in Theor. Comp. Sci.*, pages 87–114, 2003.

[17] Anupam Datta, Ante Derek, John Mitchell, and Dusko Pavlovic. A derivation system and compositional logic for security protocols. *J. of Comp. Security*, 13:423–482, 2005.

[18] Anupam Datta, Ante Derek, John C. Mitchell, and Dusko Pavlovic. A derivation system for security protocols and its logical formalization. In Dennis Volpano, editor, *Proceedings of CSFW 2003*, pages 109–125. IEEE, 2003.

[19] Anupam Datta, Ante Derek, John C. Mitchell, and Dusko Pavlovic. Abstraction and refinement in protocol derivation. In Riccardo Focardi, editor, *Proceedings of CSFW 2004*, pages 30–47. IEEE, 2004.

[20] Whitfield Diffie and Susan Landau. *Privacy on the Line: The Politics of Wiretapping and Encryption*. MIT Press, 2010.

[21] Jan van Dijk. *The Network Society*. SAGE Publications, 2012.

[22] Nancy Durgin, John Mitchell, and Dusko Pavlovic. A compositional logic for proving security properties of protocols. *J. of Comp. Security*, 11(4):677–721, 2004.

[23] Cynthia Dwork and Aaron Roth. The algorithmic foundations of differential privacy. *Foundations and Trends in Theoretical Computer Science*, 9(3-4):211–407, 2014.

[24] Maurice H. Yearwood *et al.* On wealth and the diversity of friendships: High social class people around the world have fewer international friends. *Personality and Individual Differences*, 87:224 – 229, 2015.

[25] Joseph A. Goguen and Jose Meseguer. Security policies and security models. In *Proc. of IEEE Symposium on Security and Privacy, Oakland, CA*, pages 11–20. IEEE, 1982.

[26] Jürgen Habermas. *The Structural Transformation of the Public Sphere: An Inquiry Into a Category of Bourgeois Society*. Studies in contemporary German social thought. MIT Press, 1991.

[27] Godfrey H. Hardy, John E. Littlewood, and George Pólya. *Inequalities*. The University Press, 1934.

[28] Michal Kosinski, David Stillwell, and Thore Graepel. Private traits and attributes are predictable from digital records of human behavior. *Proceedings of the National Academy of Sciences*, 110(15):5802–5805, 2013.

[29] Bradley Malin and Latanya Sweeney. Re-identification of DNA through an automated linkage process. In *AMIA 2001, American Medical Informatics Association Annual Symposium, Washington, DC, USA, November 3-7, 2001*. AMIA, 2001.

[30] Albert W. Marshall and Ingram Olkin. *Inequalities: Theory of Majorization and Its Applications*, volume 143 of *Mathematics in Science and Engineering*. Academic Press, 1979.

[31] Albert W Marshall, Ingram Olkin, and Barry C Arnold. *Inequalities: Theory of Majorization and Its Applications*. Springer-Verlag New York, 2011.

[32] Daryl McCullough. Covert channels and degrees of insecurity. In *First IEEE Computer Security Foundations Workshop - CSFW'88, Franconia, New Hampshire, USA, June 12-15, 1988, Proceedings*, pages 1–33. MITRE Corporation Press, 1988.

[33] John McLean. Security models and information flow. In *IEEE Symposium on Security and Privacy*, pages 180–189, 1990.

[34] John McLean. Security models. In *Encyclopedia of Software Engineering*, page 85. Wiley, 1994.

[35] Catherine Meadows and Dusko Pavlovic. Deriving, attacking and defending the GDOI protocol. In Peter Ryan, Pierangela Samarati, Dieter Gollmann, and Refik Molva, editors, *Proceedings of ESORICS 2004*, volume 3193 of *Lecture Notes in Computer Science*, pages 53–72. Springer Verlag, 2004.

[36] Catherine Meadows, Radha Poovendran, Dusko Pavlovic, LiWu Chang, and Paul Syverson. Distance bounding protocols: authentication logic analysis and collusion attacks. In R. Poovendran, C. Wang, and S. Roy, editors, *Secure Localization and Time Synchronization in Wireless Ad Hoc and Sensor Networks*. Springer Verlag, 2006.

[37] Arvind Narayanan and Vitaly Shmatikov. Robust de-anonymization of large sparse datasets. In *Proceedings of the 2008 IEEE Symposium on Security and Privacy*, SP '08, pages 111–125, Washington, DC, USA, 2008. IEEE Computer Society.

[38] Michael A. Nielsen. Characterizing mixing and measurement in quantum mechanics. *Physical Review A*, 63(2):022114, February 2001.

[39] Lena C. Orlin. *Locating Privacy in Tudor London*. Oxford University Press, 2009.

[40] Lawrence Page, Sergey Brin, Rajeev Motwani, and Terry Winograd. The PageRank citation ranking: Bringing order to the Web. Technical report, Stanford Digital Library Technologies Project, 1998.

[41] Dusko Pavlovic. Network as a computer: ranking paths to find flows. In Alexander Razborov and Anatol Slissenko, editors, *Proceedings of CSR 2008*, volume 5010 of *Lecture Notes in Computer Science*, pages 384–397. Springer Verlag, 2008. arxiv.org:0802.1306.

[42] Dusko Pavlovic and Catherine Meadows. Deriving authentication for pervasive security. In John McLean, editor, *Proceedings of the ISTPS 2008*. ACM, 2008. 15 pp.

[43] Dusko Pavlovic and Catherine Meadows. Actor Network Procedures. In Ram Ramanujam and Srini Ramaswamy, editors, *Proceedings of International Conference on Distributed Computing and Internet Technologies 2012*, volume 7154 of *Lecture Notes in Computer Science*, pages 7–26. Springer Verlag, 2012. arxiv.org:1106.0706.

[44] Dusko Pavlovic and Catherine Meadows. Deriving ephemeral authentication using channel axioms. In Bruce Christianson et al, editor, *Security Protocols XVII*, volume 7028 of *Lecture Notes in Computer Science*, pages 240–268. Springer Verlag, 2013.

[45] Duško Pavlović and Martín Escardó. Calculus in coinductive form. In Vaughan Pratt, editor, *Proceedings. Thirteenth Annual IEEE Symposium on Logic in Computer Science*, pages 408–417. IEEE Computer Society, 1998.

[46] Karl R. Popper. *Conjectures and Refutations: The Growth of Scientific Knowledge*. Classics Series. Routledge, 2002.

[47] Phillip Rogaway. The moral character of cryptographic work. *IACR Cryptology ePrint Archive*, 2015:1162, 2015.

[48] John Rushby. Noninterference, transitivity, and channel-control security policies, 1992. Technical report.

[49] Donald G. Saari. *Basic Geometry of Voting*. Basic Geometry of Voting Series. Springer Berlin Heidelberg, 1995.

[50] Ferdinand D. Schoeman. *Philosophical Dimensions of Privacy: An Anthology*. Cambridge University Press, 1984.

[51] Claude E. Shannon. Communication theory of secrecy systems. *Bell System Technical Journal*, 28(4):656–715, 1949.

[52] David Sutherland. A model of information. In *Proc. of the 9th National Computer Security Conference*, pages 175–183, 1986.

[53] Kotaro Suzumura. *Rational Choice, Collective Decisions, and Social Welfare*. Cambridge University Press, 2009.

[54] Latanya Sweeney. Weaving technology and policy together to maintain confidentiality. *Journal of Law, Medicine and Ethics*, 25:98–110, 1997.

[55] Latanya Sweeney. Achieving k-anonymity privacy protection using generalization and suppression. *International Journal of Uncertainty, Fuzziness and Knowledge-Based Systems*, 10(5):571–588, 2002.

[56] Latanya Sweeney. k-anonymity: A model for protecting privacy. *International Journal of Uncertainty, Fuzziness and Knowledge-Based Systems*, 10(5):557–570, 2002.

[57] Samuel D. Warren and Louis D. Brandeis. The right to privacy. *Harvard Law Review*, 4(5):193–220, 1890.

[58] Wikipedia. Cambridge Analytica. wikipedia.org/wiki/Cambridge_Analytica.

[59] Shoshana Zuboff. *The Age of Surveillance Capitalism: The Fight for a Human Future at the New Frontier of Power*. PublicAffairs, 2019.

Appendix: Proofs

Proof of Prop. 2.6. Given a doubly substochastic matrix M, form the doubly stochastic matrix $\widetilde{M}$ as follows. For any $x = (x_1 \cdots x_n)$, let $\mathrm{diag}(x)$ be the square matrix with x along its main diagonal, and zeros elsewhere. Let $\bar{1}$ denote the all 1's vector. Then we may define a doubly stochastic matrix.

$$\widetilde{M} = \begin{bmatrix} M & I - \mathrm{diag}(M\bar{1}) \\ I - \mathrm{diag}(M^\intercal \bar{1}) & M^\intercal \end{bmatrix}$$

$(a) \Rightarrow (b)$ Suppose $\beta = D\gamma$, where $\beta, \gamma \in \mathcal{DY}$, and D is a $n \times n$ doubly substochastic matrix with $n = \max(\#\beta, \#\gamma)$. Then as above we form $\widetilde{D}$ which is a $2n \times 2n$ doubly stochastic matrix. We apply Birkhoff's Theorem [10], which states that the set of doubly stochastic matrices is the convex hull of the permutation matrices, to $\widetilde{D}$. Thus we write $\widetilde{D} = \sum_{i=0}^{m} \lambda_i P_i'$,

where $m < (2n)!$, $\lambda_i \in [0,1]$, $\sum_{i=0}^{m} \lambda_i \leq 1$, and P_i' are $2n \times 2n$ permutation matrices. We may extract the upper left $n \times n$ matrices from each P_i' and denote them by P_i. Then we have $\beta = D\gamma = \sum_{i=0}^{m} \lambda_i P_i \gamma$, and we are done.

$(b) \Rightarrow (a)$ In a similar manner as above, suppose $\beta = \sum_{i=0}^{m} \lambda_i P_i \gamma$, where the P_i are partial permutations and λ_i and m satisfy the conditions above. Then for each P_i we form $\widetilde{P}_i$ which are permutation matrices since they are doubly stochastic with only 1's and 0's as entries. By Birkhoff's Theorem, $\sum_{i=0}^{m} \lambda_i \widetilde{P}_i$ is a doubly stochastic matrix, say D', since it is a convex combination of permutation matrices. Extract the upper left $n \times n$ submatrix of D' and label it D. Then we have

$$\beta = \sum_{i=0}^{m} \lambda_i P_i \gamma = D\gamma.$$

Thus we have $(b) \Leftrightarrow (a)$.

$(a) \Rightarrow (c)$ As above, suppose $\beta = D\gamma$, and let $d_{i,j}$ denote the entry in row i, column j. We follow the proof of theorem A.4 in [31]. Let P, Q be any permutation matrices such that $P\beta = \beta^{\downarrow}$ and $Q\gamma = \gamma^{\downarrow}$, i.e., they are reordered in decreasing order. Since P and Q are invertible, $\beta = D\gamma$ if and only if $P\beta = PDQ^{-1}Q\gamma$. In addition, PDQ^{-1} is a doubly substochastic matrix since we may simply multiply on the right (left) by $\bar{1}$ ($\bar{1}^{\intercal}$) respectively and observe the entries of the resulting vector are in $[0,1]$. Thus we may assume for simplicity that β and γ are in decreasing order. Then we have

$$\int \beta(k) = \sum_{i=0}^{k} \beta(i) = \sum_{i=0}^{k} \sum_{j=0}^{n-1} \gamma(j) d_{i,j} = \sum_{j=0}^{n-1} \gamma(j) \sum_{i=0}^{k} d_{i,j}.$$

Let $t_j = \sum_{i=0}^{k} d_{i,j}$ and $\epsilon = k+1-\sum_{j=0}^{n-1} t_j$. Since D is substochastic, it follows that $0 \leq \epsilon \leq k$. Now we show $\beta \prec \gamma$.

$$\begin{aligned}
\sum_{i=0}^{k} \beta(i) - \sum_{i=0}^{k} \gamma(i) &= \sum_{j=0}^{n-1} \gamma(j) \sum_{i=0}^{k} d_{i,j} - \sum_{i=0}^{k} \gamma(i) \\
&= \sum_{j=0}^{n-1} \gamma(j) t_j - \sum_{i=0}^{k} \gamma(j) + \gamma(k) \left(k + 1 - \epsilon - \sum_{j=0}^{n-1} t_j \right) \\
&= \sum_{i=0}^{k} \left(\gamma(i) - \gamma(k) \right) \left(t_i - 1 \right) + \sum_{i=k+1}^{n-1} \left(\gamma(i) - \gamma(k) \right) t_i - \gamma(k)\epsilon \\
&\leq 0
\end{aligned}$$

The last inequality follows since $\gamma(i) - \gamma(k)$ is ≥ 0 for $i \leq k$ and ≤ 0 otherwise. In addition, $0 \leq t_i \leq 1$. Thus $\beta \prec \gamma$.

$(c) \Rightarrow (a)$ As before, we may assume β and γ are in descending order. We now will find where they differ and transform γ to β by a series of substochastic transformations and permutations.

We now perform the first step. Let j be the greatest index where $\beta(j) < \gamma(j)$, and if j does not exist then we are done since they are equal. Let $k > j$ be the smallest index such that $\beta(k) > \gamma(k)$. If k does not exist, i.e., β is less than γ component-wise, then form

$$D = \text{diag}(\frac{\beta(0)}{\gamma(0)}, \frac{\beta(1)}{\gamma(1)}, \ldots, \frac{\beta(m-1)}{\gamma(m-1)}, 0, \ldots, 0)$$

where $\gamma(m) = 0$ and $\gamma(i) > 0$ for $i < m$. Clearly D is substochastic and $\beta = D\gamma$. Let us assume k exists. We now define a T-transformation as follows. Let $\lambda \in [0, 1]$ and Q a permutation matrix. Then a T-transformation has the form

$$T = \lambda I + (1 - \lambda)Q.$$

Observe that

$$\gamma(j) > \beta(j) \geq \beta(k) > \gamma(k).$$

Now define

$$\lambda = 1 - \frac{\min(\gamma(j) - \beta(j), \beta(k) - \gamma(k))}{\gamma(j) - \gamma(k)}.$$

Let Q be the permutation matrix switching out the j and k coordinates, and let T as above. Then let $\gamma' = T\gamma$. Observe that $\beta \prec \gamma'$ since the total sum is preserved up to j since $\int \gamma'(i) = \int \gamma(i) \geq \int \beta(i)$ for $i < j$. At j, if

$$\gamma(j) - \beta(j) \leq \beta(k) - \gamma(k),$$

then

$$\gamma'(j) = \beta(j),$$

otherwise

$$\gamma'(j) = \gamma(k) + \gamma(j) - \beta(k).$$

In either case we have $\gamma'(j) \geq \beta(j)$. Thus $\int \gamma'(j) \geq \int \beta(j)$. For $j < i < k$, by choice of i, j we have $\gamma'(i) = \gamma(i) = \beta(i)$, and hence $\int \gamma'(i) \geq \int \beta(i)$. Finally for $i \geq k$, by definition of the transformation, T, $\int \gamma'(i) = \int \gamma(i) = \int \beta(i)$. Thus we have $\beta \prec \gamma'$.

Note that each time we perform this step, since either the $\gamma'(j) = \beta(j)$ or $\gamma'(k) = \beta(k)$, we convert one point a γ' to a point of β. Thus within n steps we will finish, and we are done.

□

Proof of Prop. 2.8. We show that there is a common ordering for $\curlywedge B$. It follows that $\curlywedge B$ is then majorized by the elements of B.

Let us first show $\curlywedge B$ is ordered by ϑ. The essential condition we must show is that $\curlywedge B\vartheta(j) \geq \curlywedge B\vartheta(j+1)$. For ease of notation let β_j denote the element of B that minimizes the sum $\sum_{i=0}^{j} \beta\vartheta(i)$.

$$\begin{aligned}
\curlywedge B\vartheta(j) &= \Big(\partial \bigwedge_{\beta\in B} \int \beta\vartheta\Big)(j) \\
&= \min_{\beta\in B} \sum_{i=0}^{j} \beta\vartheta(i) - \min_{\beta\in B} \sum_{i=0}^{j-1} \beta\vartheta(i) \\
&\geq \sum_{i=0}^{j} \beta_j\vartheta(i) - \sum_{i=0}^{j-1} \beta_j\vartheta(i) \\
&= \beta_j\vartheta(j) \\
&\geq \beta_j\vartheta(j+1) \\
&= \sum_{i=0}^{j+1} \beta_j\vartheta(i) - \sum_{i=0}^{j} \beta_j\vartheta(i) \\
&\geq \min_{\beta\in B} \sum_{i=0}^{j+1} \beta\vartheta(i) - \min_{\beta\in B} \sum_{i=0}^{j} \beta\vartheta(i) \\
&= \curlywedge B\vartheta(j+1)
\end{aligned}$$

Thus $\curlywedge B$ is ordered by ϑ, and we now show that it is majorized by all $\gamma \in B$.

$$\begin{aligned}
\int \curlywedge B^{\downarrow}(k) = \int \curlywedge B\vartheta(k) &= \int \Big(\partial \bigwedge_{\beta\in B} \int \beta\vartheta\Big)(k) \\
&= \min_{\beta\in B} \sum_{i=0}^{k} \beta\vartheta(i) \\
&\leq \sum_{i=0}^{k} \gamma\vartheta(i) \\
&= \int \gamma^{\downarrow}(k)
\end{aligned}$$

Observe by definition the bound is tight, since for every k there is a $\beta \in B$ such that $\int \curlywedge B^{\downarrow}(k) = \int \beta^{\downarrow}(k)$. □

Proof of Lemma 2.11. By definition, we have $\bigcup_{y\in\mathcal{Y}} B_y = \mathcal{Y}$. Now we show, given $u, v \in \mathcal{Y}$ that $B_u \cap B_v$ is either empty or $B_u = B_v$. If $u \neq v$ and $|B_v| = 1$ or $|B_u| = 1$ then $B_u \cap B_v = \emptyset$ since a consistency class with only one element implies that the element can

not be in any $\lhd$-cycles. Let us assume both B_u and B_v have more than one element and $\exists\ w \in B_u \cap B_v$. The previous statements imply $\exists\ \delta_0, \delta_1, \ldots, \delta_m, \gamma_0, \gamma_1, \ldots, \gamma_n \in B$ and $u_1, u_2, \ldots, u_m, v_1, v_2, \ldots, v_n \in \mathcal{Y}$ such that we have the two following $\lhd$-cycles

$$w \overset{\delta_0}{\lhd} u_1 \ldots u_m \overset{\delta_m}{\lhd} w$$
$$w \overset{\gamma_0}{\lhd} v_1 \ldots v_n \overset{\delta_n}{\lhd} w$$

where there are i, j so that $u_i = u$ and $v_j = v$. Without loss of generality, let $x \in B_u$. We will show $x \in B_v$. Then as above, x is in a $\lhd$-cycle

$$x \overset{\alpha_0}{\lhd} u'_1 \ldots u'_p \overset{\alpha_p}{\lhd} x$$

where we there is a k so that $u'_k = u$. Then we may construct the following $\lhd$-cycle

$$x \overset{\alpha_0}{\lhd} u'_1 \ldots \overset{\alpha_k}{\lhd} u'_k = u_i \ldots u_m \overset{\delta_m}{\lhd} w \overset{\gamma_0}{\lhd} v_1 \ldots v_n \overset{\delta_n}{\lhd} w \overset{\delta_0}{\lhd} u_1 \ldots u_i = u'_k \ldots u'_p \overset{\alpha_p}{\lhd} x$$

Thus we have shown that $x \in B_v$. For the second part of the lemma, we must show that if $B_u \cap B_v = \emptyset$ then $B_u \overset{\delta}{\trianglelefteq} B_v$ or $B_v \overset{\delta}{\trianglelefteq} B_u$ for all $\delta \in B$. If there exists $\delta, \gamma \in B$ and $x \in B_u$, $y \in B_v$ so that $x \overset{\delta}{\lhd} y$ and $y \overset{\gamma}{\lhd} x$, then it follows $x \in B_v$ and $y \in B_u$ resulting in a contradiction. $\square$

Proof of Prop. 2.12. We first show $\widehat{\beta}$ is consistent with all elements of B. Let $\gamma \in B$. Suppose $u \overset{\gamma}{\lhd} v$. Then either $B_u \cap B_v = \emptyset$ or $B_u = B_v$. If the intersection is empty, then by lemma 2.11 it follows $B_u \overset{\beta}{\trianglelefteq} B_v$. Hence $u \overset{\widehat{\beta}}{\trianglelefteq} v$. It the intersection is nonempty, it follows $B_u = B_v$ and trivially $u \overset{\widehat{\beta}}{\diamond} v$.

Conversely, suppose $u \overset{\widehat{\beta}}{\lhd} v$. Then $B_u \neq B_v$ and by lemma 2.11 the intersection is empty. In addition, since $w \overset{\beta}{\lhd} x$ for some $w \in B_u$ and $x \in B_v$ it follows $u \overset{\gamma}{\trianglelefteq} v$. Thus $\widehat{\beta}$ is consistent with the elements of B.

Now we show $\widehat{\beta}$ is maximal with regards to all sources consistent with the elements of B and majorized by β. Note trivially that $\widehat{\beta} \prec \beta$ since by definition $\widehat{\beta}(u) \leq \beta(u)$ for all $u \in \mathcal{Y}$. Suppose $\alpha \in \mathcal{D}\mathcal{Y}$ is consistent with the elements of B. We first make a quick observation, which is that any source consistent with each element of B, will be constant on the consistency classes of B, i.e., B_u for $u \in \mathcal{Y}$. Let $w, x \in B_u$ and suppose $w \overset{\alpha}{\lhd} x$. Then since α is consistent with B, we have $w \overset{\delta}{\trianglelefteq} x$ for all $\delta \in B$. This is a contradiction, since there must exist a γ such that $x \overset{\gamma}{\lhd} w$, otherwise $x, w \notin B_u$. From here it is immediate that $\widehat{\beta}$ was chosen maximally. This follows since both α and $\widehat{\beta}$ are constant on consistency classes of B. Furthermore since $\alpha \prec \beta$, it follows for any u that

$$\alpha(u) \leq \bigwedge_{y \in B_u} \beta(y) = \widehat{\beta}(u)$$

. Thus $\widehat{\beta}$ is maximal with regards to the majorization preorder.

The final part of the proposition is to show that the set $\widehat{B} = \left\{\widehat{\beta} \mid \beta \in B\right\}$ is consistent. It follows from lemma 2.11 and the previous remark, that any source consistent with each element of B will be constant on the consistency classes. In particular, the lemma implies that the consistency classes are ordered under $\overset{B}{\trianglelefteq}$. The previous remark implies that we may find a common ordering based upon the ordering of the consistency classes. Then since we have a common ordering for $\widehat{B}$, by prop. 2.10 it is consistent. $\square$