Fine-Grained User Privacy from Avenance Tags*

Eleanor Birrell Fred B. Schneider

Department of Computer Science
Cornell University
Ithaca, NY 14853
{eleanor, fbs}@cs.cornell.edu

April 20, 2014

Abstract

In the Internet, users interact with service providers; these interactions exchange information that might be considered private by the user. Existing schemes for expressing and enforcing user privacy on the Internet—notably notice and consent—are inadequate to address privacy needs of Internet users. This paper suggests a new, practical, and expressive policy tag scheme that would enable users to express both control-based and secrecy-based restrictions. We identify key design goals, explore various design choices that impact these goals, and outline a proposed implementation—called *avenance tags*—that realizes these goals.

*Supported in part by AFOSR grants F9550-06-0019 and FA9550-11-1-0137, National Science Foundation grants 0430161, 0964409, and CCF-0424422 (TRUST), ONR grants N00014-01-1-0968 and N00014-09-1-0652, and grants from Microsoft. Birrell is also supported in part by a National Science Foundation Graduate Research Fellowship.

1 Introduction

In the Internet, *users* interact with *service providers* (e.g., Amazon, Facebook, Bank of America). These interactions involve *attributes* that are explicitly disclosed by the user and *behaviors* that can be observed during the interaction. Some or all of this information might be considered private by the user.

Definitions of privacy typically incorporate elements of secrecy [18, 39, 46, 1] and of control [16, 17, 24, 1]. Early privacy protection standards—including the OECD guidelines [37] and the FTC fair information practice principles (FIPPs) [15]—exemplify this dichotomy. Secrecy is inherent in the OECD *Collection Limitation* principle and the FTC *Notice/Awareness* principle, which give users the right to make informed decisions about what data are collected or shared. Control underlies the OECD *Use Limitation* principle and the FTC *Choice/Consent* principle, which each give users the right to limit uses or choose how personal information should be used.

The current regime for expressing and enforcing user privacy on the Internet—called *notice and consent*—is intended to implement FIPPs. Under notice and consent, each service provider publishes a list of data practices (what information will be collected and what will or will not be done with it). This *site privacy policy* is typically a long document, available somewhere on the service provider's website. The act of using a service or sharing data with it is interpreted as consenting to the data uses published in the service provider's site privacy policy. The presumption is that notice and consent enables free, informed consent, resulting in a satisfactory trade-off between privacy and economic benefits. But does it?

The answer is no. Fifteen years of critiques [3, 4, 7, 8, 9, 10, 26, 35, 36, 38, 42, 40, 34] suggest that notice and consent is inadequate to address privacy needs of Internet users. One problem is that notice and consent gives users a single all-or-nothing choice: to decline the service or to consent to the published data practices and share the requested information. In many cases, providers of equivalent services publish similar site privacy policies—and many such services are socially or economically necessary—so users, left with no real choice, are coerced to consent. Notice and consent thus compresses the space of user choices to a single option: whether or not to reveal certain data (secrecy). It fails to adequately implement FIPPs by failing to give practical means for users to express and control how their data are used.

In response to all this criticism, there is a resurgent focus on viewing privacy in terms of limitations on data use. In some cases, the emphasis is placed on preventing harmful uses without explicit user control [8, 9]; in others, the emphasis is on enabling user control over data uses [28, 34]. Since different users are likely to have different opinions regarding what constitutes a privacy violation and since there is no consensus on how to define "harmful," we focus on options for enabling user control.

Data use can occur at any point after data is collected, so control over data use naturally aligns with the idea of *policy tags*. Policy tags are digital artifacts that travel with a value and express limitations on how that value may be used. The ideas behind policy tags can be traced to early work on Owner-Retained Access Control [29]. Forms of policy tags have been previously used in database systems [47, 33] and information flow control systems [12, 27]. Several recent proposals address problems with notice and consent by employing some form of policy tag [38, 25, 2, 13, 5, 6, 28, 34]. However, this prior work has failed to adequately address all the shortcomings of notice and consent.[1]

This paper therefore suggests a new, practical, and expressive policy tag scheme that would enable user control over data at all times and in all contexts. We identify key design goals, explore

[1] We identify and discuss five shortcomings of notice and consent in Section 2.

various design choices that impact these goals, and outline a proposed implementation—called *avenance tags*—that realizes these goals.

2 Shortcomings of Notice and Consent

Each user has a *user privacy policy* that defines which interactions would constitute privacy violations involving that user's data. User privacy policies might include conditions like "my privacy is violated if any service provider knows my age" or "my privacy is violated if I am displayed an ad that was targeted based on my gender." Privacy-enhancing schemes attempt to enable users to express these user privacy policies (and enable service providers to comply with user privacy policies) without imposing restrictions that might disproportionately inhibit utility or economic benefits.

Under notice and consent, users in effect communicate their user privacy policy by selectively granting or withholding consent to data uses published in a site privacy policy. As long as consent is freely given by a fully-informed user, notice and consent allows the user to decide whether the trade-off between privacy and utility is equitable. However, coerced or uninformed consent introduces opportunities for privacy violations.

In this section, we identify five shortcomings of notice and consent that result in coerced or uninformed consent: the problem of *expressiveness*, the problem of *scalability*, the problem of *transparency*, the problem of *user policy revision*, and the problem of *enforcement*.

Expressiveness. Under notice and consent, users are presented with an all-or-nothing choice for each interaction: use the service (and accept the site privacy policy for all data) or decline to use the service. Consequently, user privacy degenerates to a question of secrecy (the user decides whether to reveal certain data by interacting with the service provider), and there is no mechanism for expressing limitations on data use. Users must consent to all of the disclosed data uses in order to receive the service. This coerced consent is exacerbated in the case of services that are socially or economically necessary (e.g., email, employer software, etc.) [8].

Moreover, current practices allow service providers to share data with third parties (as long as sharing is a disclosed use) without requiring that the site privacy policy either disclose or constrain data use by third parties. Since notice and consent allows users to express their privacy policy only by granting or withholding consent, the absence of information about third party uses results in uninformed consent that might not be consistent with the user privacy policy [32, 25].

Scalability. Notice and consent presumes that a decision is made (whether to consent to the site privacy policy) for each service provider with which a user interacts. For a user who interacts with many service providers, reading each site privacy policy is likely to be costly, if not infeasible [11, 26, 7, 42]. By one estimate [31], it would take the average American Internet user 244 hours to read the privacy policies for all service providers with which they interact in a single year. Not surprisingly, users typically don't read the site privacy policy and simply give uninformed consent to whatever data uses the site privacy policy contains.

Transparency. In order to grant informed consent, a user must actually understand the site privacy policy. There is evidence that this is not the case [30, 21, 23, 22]. The problem is not syntax. A high-level policy—one that specifies data uses intensionally and in human-readable language—will necessarily elide details that might be relevant; a low-level policy—one that describes exactly

Tag Scheme	Expressive.	Scalabil.	Transpar.	Pol. Revision	Enforce.	Impl.
P3P [38, 25]	✓	✓				✓
PRIME [6]	✓	✓			✓	✓
Bandhakavi et al. [2]			✓		✓	✓
Bussard et al. [5]	✓					
DNT [13]		✓	✓			✓
SDC [28]			✓		✓	
Nguyen et al. [34]				✓		

Table 3.1: Extent to which Prior Work Addresses the Shortcomings of Notice and Consent

how all data will be used in all cases—might well present too many details for a user to understand.[2]

The problem of transparency is compounded by *derived values* [2, 42]. Big data techniques give service providers means to infer new, often sensitive or valuable information from apparently innocuous data. If a site privacy policy only includes data uses for disclosed attributes (currently a common practice), then the user is not being informed about the use of values derived from observed behaviors. For example, a user would not be informed if a service provider is never sent the user's gender but still targeted ads based on gender inferred from observed HTTP requests.

User Policy Revision. User privacy policies are not necessarily static. If a user's attribute values change or new information emerges, that user might reconsider which interactions are acceptable. Consent to the site privacy policy is now not being freely given after a user has revised their privacy policy; consent is assumed by a site because notice and consent provides no mechanism for retracting previously granted consent [3, 40].

Enforcement. Finally, notice and consent relies on service providers voluntarily adhering to the data uses disclosed in their site privacy policies. There is no comprehensive, effective mechanism for enforcing privacy policies. If actual data uses deviate (either intentionally or unintentionally) from uses disclosed in a published site privacy policy, then it is impossible for users to grant informed consent to the actual data use.

These five shortcomings mean that notice and consent fails to ensure that users are giving free, informed consent to data sharing and data use. And this, in turn, motivates the need for a new regime that will solve these problems and enhance the privacy of Internet users.

3 Policy Tags for Internet Privacy

Policy tags—tags that are permanently attached to particular data—appear to be a promising solution to the problem of expressiveness. This is because tag contents can be accessed whenever data are used by any service provider (including third parties). Previous efforts have employed some form of policy tags to enhance Internet privacy [38, 6, 2, 5, 13, 28, 34]. The extent to which each of these efforts addresses the problems identified in Section 2 is summarized in Table 3.1. None addresses all of the shortcomings of notice and consent, and several have no complete specification or current implementation.

[2]This tradeoff between clarity and detail is what Nissenbaum [36] calls the transparency paradox.

Platform for Privacy Preferences (P3P) [38, 25] is a framework for enabling users to express and exercise preferences over service provider privacy practices. Users can communicate preferences to a client-side software agent that subsequently would negotiate data usage on that user's behalf. Data released to a service provider is associated with a tag that references the agreed data-use policy. An extension for the enterprise setting, E-P3P, adopted the *sticky policy* paradigm, where the association between data and policy tags persists even after data are disclosed to another enterprise. However, P3P policies fail to address the problem of transparency, since data derived from observations are not tagged, and policy tags are not propagated to derived data. P3P and E-P3P also do not address the problems of user policy revision and enforcement.

PRIME [6] is an identity management system developed for the European Union. It introduced *obligation* tags—a type of policy tag that express event-triggered mandatory actions, like notification and information deletion. Obligation tags are enforced by an obligation manager. PRIME also addressed the problem of scalability by delegating negotiations about data use policies to a trusted identity provider. Like P3P, PRIME does not solve the problem of transparency—in particular, it does not attach tags to observed behaviors or propagate policy tags to derived values. It also doesn't solve the problem of user policy revision.

Bandhakavi et al. [2] introduces *super-sticky release policies*, policy tags that are propagated to derived data. The intention was to handle cases where data aggregates are more or less sensitive than the original input values; outputs of such operations are associated with modified policy tags that are generated automatically. Bandhakavi et al. extends the PeerAccess authorization framework [48] to include these tags. In the extended framework, policies are enforced by peers in a distributed manner (receivers check that data is tagged with a signed proof that the policy is satisfied). However, expressiveness is limited, because policy tags are not propagated to third party service providers. This work also does not address the problems of scalability or user policy revision.

Bussard et al. [5] investigates a policy language that can express *downstream usage policies*, which specify conditions (agreed usage control restrictions) under which information can be released. They propose XML-based languages with which service providers specify their data-use practice and the user express—for each service provider in the chain of downstream recipients—how data can be used. Mutual agreement concerning usage of a piece of information is described in a policy tag associated (and propagated) with data. This mechanism does not, however, address the problems of scalability, transparency, user policy revision, or enforcement.

Do Not Track [13] is an effort to combat tracking of users across different websites. It treats HTTP requests as data, and it associates that data with a particular policy tag (a flag encoding the policy "do not collect, retain, or derive ads this data"). Do Not Track has been incorporated into all major web browsers, but it is neither enforced nor widely honored. Do Not Track tags also do not enable users to express policies about other data or other uses.

Secure Data Capsules [28] propose that sensitive data objects have a policy tag giving provenance and a usage policy. These policy tags are cryptographically bound to the associated data, and derived data are tagged with a derived policy tag. Untrusted applications can operate on tagged objects only while inside a secure execution environment that enforces and propagates policy tags. The proposed system was designed to allow incremental deployment. However, the scheme leaves many questions unresolved, including how to express policies governing data use and how to attain acceptable performance overhead. It also does not address the problem of user policy revision.

Nguyen et al. [34] proposes a system that describes policy tags using interoperable metadata. Under the proposed system, policy tags are integrity-protected and cryptographically bound to the data. Policy tags would be implemented as references to a single stored copy, enabling user policy revision (including the often discussed "right to be forgotten"). This approach was never

implemented, and it leaves unresolved the language for specifying user policies, the scalability and deployment of such a metadata-based architecture, the propagation of policy tags to derived data, and the enforcement of such policies.

So the potential for policy tags to enhance user privacy on the Internet is present, but prior work has not successfully solved all of the existing problems with notice and consent.

4 Avenance Tags

Five goals—derived from the shortcomings identified in Section 2—motivate the design of our scheme.

(1) **Expressiveness:** Users should be able to control how their data are used as well as what data become known (both by a service provider with which they interact and by third parties).

(2) **Scalability:** The burden placed on users should be reasonable, even if users interact with many service providers.

(3) **Transparency:** Privacy policies should be easily understood and transparent. They should clearly specify how observed data and derived values are used.

(4) **User Policy Revision:** Users should be allowed to revise privacy policies and, thereafter, should enforce the revision.

(5) **Enforcement:** Some enforcement mechanism ensures policy compliances.

In order to realize these goals, we propose a scheme called *avenance tags*.[3] Avenance tags use a new language for expressing privacy policies (which addresses goals (1) and (3)) and are handled in the context of an *avenance ecosystem* (which addresses goals (2),(3),(4), and (5)).

4.1 Policy Language

Differences among existing policy languages are symptomatic of the debate about whether privacy is about control or secrecy. A policy language might be designed to express limitations on data use (control), limitations on what predicates about data can be known (secrecy), or both. Each view comes with advantages and disadvantages for expressiveness and complexity.

The *control-based* view can be instantiated at various levels of abstraction. At one extreme, permitted uses (or operations) can be specified using the very programming language that implements the operations to be controlled. Uses defined in terms of implementation details can be easily evaluated but yield opaque privacy policies. At the other extreme, whole classes of allowed or disallowed operations can be characterized intensionally by specifying properties that operations to be controlled must satisfy. Characterizing uses intensionally is difficult but results in more transparent privacy policies.

At all levels of abstraction, adopting a control-based view of privacy facilitates expressing policies that depend on future or past uses, but the only way to express a policy about secrecy of particular values requires knowledge about all operations that generate those values. Such knowledge is rarely available. Moreover, synthesizing tags for derived values requires a complicated

[3]The term avenance is derived from the French word *avenir*, meaning future or yet to occur; the etymology is analogous to that of provenance (from *provenir*). Avenance tags—which contain privacy policies that describe how the associated data can be used in the future—are a natural complement to provenance tags—which describe how the associated data has been used in the past.

mechanism, because the permitted uses of a derived value might not simply be the intersection of the permitted uses of the data on which it depends. For example, if a user expresses the policy "Don't share this data unless it is encrypted under a secure key," then the allowed uses for the output of an encryption function would differ from the allowed uses for the input. Here, the privacy policy for the output depends on (1) whether the data was used as the plaintext or the key (i.e., an encryption takes two inputs and their policies should be treated differently), (2) how the key was generated (e.g., did it use a good source of randomness), and (3) other operations previously applied to the key (e.g., a key that previously has been sent in the clear is not secure).

The *secrecy-based* view of privacy gives rise to policy languages that explicitly express what predicates about data can become known. Such languages facilitate transparent privacy policies of the form "no one should know whether I am over 50." However, the secrecy-based view lacks a convenient mechanism for expressing privacy policies that inherently depend on data derivation (e.g., "Do not share information derived from this data unless it has been hashed"). Enforcing policy compliance is also more difficult; a reference monitor that checks each invoked operation against the list of permitted uses no longer suffices.

To avoid the debate between these two views, Helen Nissenbaum introduced *contextual integrity* [35, 36]. Here, privacy is neither about control nor secrecy but about appropriate flow of personal information. "Appropriate" is defined by social norms associated with the roles and contexts in which information is disclosed. (A version of contextual integrity has been formalized by Barth et al. [4] using Linear Temporal Logic.) Nissenbaum's approach, however, can handle only those cases for which there is an agreed social norm. For example, some users consider behavioral advertising[4] to be a privacy violation while others consider it benign or even useful. So there is no agreed social norm regarding behavioral advertising; contextual integrity, therefore, cannot be applied. Unlike Nissenbaum, our underlying philosophy is that the "appropriate" uses and knowledge that characterize a privacy policy are best defined by the user (who may, in turn, adopt social norms in applicable cases).

In order to enable users to express both control and secrecy, avenance tags generalize the control-based and secrecy-based views. We model service providers as principals that execute interactive programs in an environment $\Gamma : \mathit{Attr} \to \mathit{Val}$ that defines a value for each user attribute. Let Envs be the set of all possible environments. Service providers interact with each other and with users by sending messages. We can model these messages as containing sets of attribute-value pairs, consistent with the current environment.

Avenance policies express limitations on the dissemination of particular information. An avenance policy is formalized as a pair $\langle \mathcal{C}, \Theta \rangle$ comprising:

- A set of *sensitive classes* $\mathcal{C} \subseteq 2^{\mathit{Envs}}$. Each sensitive class $C \in \mathcal{C}$ is a set of environments.
- A *selection predicate* $\Theta : M \to \mathit{Bool}$ evaluated over messages

Policy $\langle \mathcal{C}, \Theta \rangle$ is interpreted as saying that for all principals P, the subset of P's outgoing messages out_P satisfying Θ—denoted $\Theta(\mathit{out}_P)$—must not reveal that any particular sensitive class $C \in \mathcal{C}$ contains the current environment. For instance, the policy "Service providers should not know my age" could be expressed by defining a sensitive class for each possible value of age (that is, $\mathcal{C} = \{\{\Gamma : \Gamma(\mathit{age}) = v\} : v \in \mathit{Range}(\mathit{age})\}$) and by selecting all outgoing messages (that is, $\Theta = \mathit{true}$). Observe that sensitive classes are not necessarily disjoint. So the privacy policy

[4]Behavioral advertising is the practice of tracking user behaviors (e.g., websites visited or products viewed), then inferring user attributes (e.g., gender, political affiliation, income, or hobbies) based on statistical models, and ultimately displaying ads targeted to those populations (e.g., male car enthusiasts).

"Service providers should not know my age within 5 years" can be expressed by sensitive classes containing five-year age ranges (which overlap).

4.2 Auxiliary Information

Some policies (e.g., "service providers can't know $e(\Gamma)$") can be expressed using selection predicates that are easy to evaluate (e.g., $\Theta = true$). But not all predicates can be evaluated with the attribute-value pairs in a message. In order to handle a more expressive class of predicates, and thus support a more expressive class of policies, avenance tags henceforth are triples $\langle \mathcal{C}, \Theta, a \rangle$, where a is auxiliary information on which Θ can depend. To motivate this design decision, consider some examples.

We might be interested in a policy like "Don't store my credit card for more than 30 days." A rule like this requires formalizing what it means to "store" data; we might interpret this rule as "Don't demonstrate knowledge of credit card after 30 days." However, "occurs after more than thirty days" cannot be evaluated given only a credit card value; compliance requires access to some notion of time. So avenance tags support privacy policies of this type by using auxiliary information. Here, the auxiliary information would encode an initialization time and the time at which the message is sent.

Auxiliary information in avenance tags is useful for expressing other types of policies, too, like "don't forward data unless authorized by a federal judge" and "don't display ads derived from age." In these cases, auxiliary information would include a signed authorization from a federal judge or labels indicating that this message contains an ad. A selection predicate then can use this auxiliary information to identify appropriate outgoing messages.

Provenance is a particularly useful kind of auxiliary information for expressing privacy policies. A simple provenance-dependent policy is "Don't send data derived from age." Moreover, by incorporating labels on values and operations in the provenance for a value, we can also express additional policies. For example, "don't combine data from NSA and FBI" or "don't release information derived from my password unless it has been hashed." The former depends on value labels to indicate the data collector while the latter depends on operation labels to identify computationally-noninvertable hash functions.

So far, we have discussed avenance tags for specifying user-provided privacy policies. However, restrictions on data use can come from company policy and local, national, or international law. Such restrictions could be explicitly encoded in individual avenance tags or (for efficiency) could be omitted but implicitly imposed on all principals operating within some given jurisdiction. In some cases (e.g., when a user decides to disclose data only because of a company use policy precludes certain uses) jurisdictional restrictions might have to persist for data that is derived and/or further forwarded. Here, the original jurisdiction or the jurisdictionally-imposed privacy policy can be encoded in the auxiliary information prior to leaving the jurisdiction.

4.3 Expressiveness

To appreciate the expressiveness of our policy language, observe that the control-based view and the secrecy-based view are both captured.

- To express a control-based policy such as "operation *op* cannot be applied to attribute a," we define $\mathcal{C} = \{Envs\}$ and define Θ to select those messages generated by applying *op* to a. The derivation history of a value would be encoded in its auxiliary information.

- To express a secrecy-based policy such as "service providers can't know $e(\Gamma)$" for some expression e, identify sensitive classes with those environments in which $e(\Gamma)$ has the same value

and define $\Theta = true$, so all outgoing messages are selected.

Compliance with $\langle \mathcal{C}, \Theta, a \rangle$ can be reduced to an information flow problem. *Possibilistic noninterference* [41] states that a change in secret inputs does not affect the set of possible public outputs.[5] For our context, $\mathcal{C}$ is identified with high inputs (a separate bit indicates membership in each sensitive class $C \in \mathcal{C}$) and Θ determines which subset of outgoing messages (outputs) are low (or public). Note, our definition of compliance is weaker than possibilistic noninterference, because outputs are allowed to reveal non-membership in a class C.

Compliance with $\langle \mathcal{C}, \Theta, a \rangle$ can alternatively be considered through the lens of nondeducibility [44]. Nondeducibility defines information flow between functions f_1 and f_2 to occur when, for a given argument, knowing the value of f_1 precludes some value for the output of f_2. Policy compliance then can be interpreted as limiting the information flow from the function f_1 that returns a service provider P's view of an interaction to a function f_2 that encodes (non)membership in the sensitive classes $C \in \mathcal{C}$.

There are various other relaxations of noninterference (e.g., [45]). Any could be selectd as a basis for compliance with $\langle \mathcal{C}, \Theta, a \rangle$, and the result will be a different privacy standard. For example, a probabilistic policy language might allow avenance tags that are 4-tuples $\langle \mathcal{C}, \Theta, a, p \rangle$ that would require that outgoing messages satisfying Θ not reveal membership in any sensitive classes $C \in \mathcal{C}$ with probability above p. This could express policies like "service providers shouldn't be able to guess my age with probability above .5"

4.4 Local Checking of Compliance

To comply with a user's privacy policy, a service provider must ensure that its outgoing messages do not violate that policy. If a service provider is not aware of a policy, compliance is difficult for even well-intentioned service providers. Two design choices mitigate this burden:

- **Policy Dependency:** A message is tagged with a policy only if it is possible for a service provider to use the message contents in a manner that violates that policy.
- **Policy Propagation:** Policies stored in an avenance tag associated with the input to an operation are attached to the output of that operation. In the case of multi-input operations, the output (or outputs) of that operation are associated with the union of the policies associated with input values.[6] Auxiliary information for derived tags is synthesized according to operation-specific procedures that must be available to all principles that handle the tag.

By adopting Policy Dependency and Policy Propagation, privacy compliance can be locally enforced. That is, if no *local policy violation* occurs—that is, no principal P sends a message that violates any of the policies associated with that message—then no policy violations occur. Compliance with avenance policies is, therefore, a reasonable expectation to place on service providers.

[5] Possibilistic noninterference is motivated by nondeterministic programs, in which a single input value can yield multiple possible output values. It relaxes standard noninterference [19, 20], which stipulates that changes in secret input do not change the (unique) public outputs. Since executions that distinguish between different environments in a single class are not privacy violations (unlike the traditional information flow scenario), possibilistic noninterference is the more appropriate.

[6] While sufficient, this can lead to avenance tags that include unnecessary policies. So, it seems useful to consider a declassification mechanism, whereby a principal may replace some of the policies by a signed assertion that those policies have become irrelevant (that is, no subsequent messages or actions can violate that policy, either due to independence from the input value or temporal restrictions).

4.5 Policy Enforcement

There are two general approaches to thwarting harms from policy violations: prevention and detection. Prevention ensures that violations do not occur, but it can be expensive. Detection merely ensures that policy violations are noticed; either recovery or deterrence through accountability is employed to prevent harms. Of course, not all violations can be recovered from after detection nor deterred by accountability. So a regime is typically adopted that lies somewhere along the spectrum between prevention and detection, depending on the threat model and harms associated with a particular deployment.

Policy tags are usually associated with prevention. For example, information flow control labels [12, 27] are amenable to static enforcement measures. Programs are checked for policy compliance by a compiler's type checker, and only programs that successfully type check are compiled and run. Provenance tags [33] and other information flow control systems [43, 14] can also be conservatively enforced at run-time—either by an operating system or by a application-level reference monitor. Such dynamic prevention mechanisms often add performance overhead, and assurance that dynamic reference monitors are consistently and correctly deployed can be difficult to achieve.

Work in computer security usually adopts a threat model in which a program and its inputs are provided by an anonymous adversary that cannot be identified or held accountable. In the context of Internet privacy, however, the adversaries are service providers. Service providers are assumed to be rational economic actors who violate policies only for economic benefit. They thus would be deterred by accountability. Moreover, since service providers are identifiable entities operating in a defined jurisdiction, society provides means—regulating authorities—for punishing non-compliant service providers through legal fines or reputation damage. Privacy policies are therefore particularly well-suited to enforcement based on deterrence.

With avenance tags, policies are formulated in terms of attribute names. Yet it is not realistic to assume that all users and service providers will agree on a single global lexicon. So different names might be used to identify the same characteristic of an individual. For example, different names might refer to different encodings of the same characteristic (e.g., age and birth date) or to different characteristics that just happen to be correlated (e.g., age and social security number or even age and taste in music). No computer program can know nor should the author of a privacy policy be expected to know all relationships among the different attribute names. So that raises a question about the feasibility of enforcing a policy described using avenance tags.

Our society, however, long ago evolved institutions and processes to deter harms that result from violations in the spirit if not the letter of a policy. These institutions employ people to judge whether a violation has occurred. The judges are provided with evidence of the circumstances associated with some action, and the judges invoke common sense as well as knowledge of norms when deciding whether some act violates a policy. Representatives from each side argue the case, so the judges have only to decide on the relative merits of what they have seen.

We propose to leverage this prosecutorial approach for implementing deterrence through accountability with avenance tags.

- The system is designed to furnish evidence to any user who claims harm and to any service provider that needs to defend against such a claim. This evidence would include logs of inputs and outputs, certified copies of programs used for processing requests, and certified copies of data used and stored.

- An authority—like a court—is used to resolve disputes about whether some service provider's action violates some given user's privacy policy. Judges are the arbiters, so they are able to

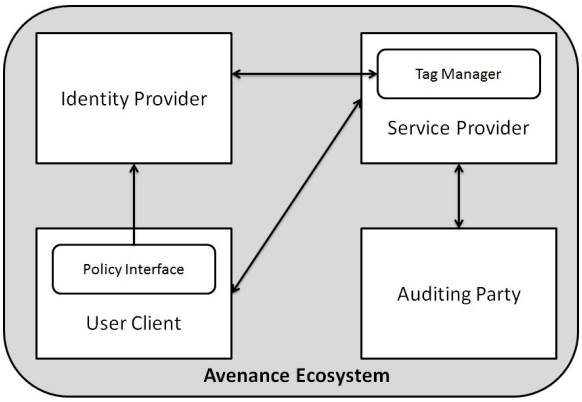

Figure 5.1: Relationship between Principals in the Avenance Ecosystem

resolve questions about whether a policy that names specific attributes applies to differently-named attributes.

5 The Avenance Ecosystem

Avenance tags alone are not sufficient to remedy all of the shortcomings of notice and consent. In order to achieve that goal, we propose that users and service providers operate within an *avenance ecosystem* that additionally includes *identity providers*, *policy interfaces*, and *auditing parties*. The high-level architecture of the avenance ecosystem is shown in Figure 5.1.

5.1 Identity Providers

Avenance tags, which include sets of policies and auxiliary information, might grow ever larger as a value travels through a system. So it could become infeasible to store all that information in a tag. We instead propose to store user privacy policies and auxiliary information at an *identity provider*—a user-trusted principal that acts as a proxy for the user.

We envision many identity providers being managed by different parties, so a user can select which identity provider (or providers) to trust with each policy. Identity providers are assumed to be available to users and service providers at all times.

Avenance tags then can be implemented as references to policies stored at an identity provider. In theory, a service provider is expected to consult the policy associated with a value each time an operation is invoked on that value or each time a message containing that value is sent. In practice, the identity provider would grant time-limited leases on policies to service providers, allowing service providers to use locally-cached copies of a policy until the lease expires. And also, messages would contain policies with expiration times instead of references to policies.

Implementing avenance tags as references to policies stored at an identity provider enables the avenance ecosystem to support user policy updates, because now users (or principals acting on behalf of a user) can modify the single copy of the user privacy policy (stored at the identity provider) and the updated policy will be automatically propagated to all values derived from the associated data, including remotely stored values. However, this raises a question: What updates are allowed?

Updated policies are problematic if the new policy must enforced retroactively. We therefore

require that policies be *ex ante facto*. That is, an updated policy applies only to actions that occur after the update is made. In order to support this requirement, operation executions are timestamped (with the time defined by the identity provider when the tag is dereferenced for the associated operation). And privacy policies are defined with respect to classes of time-stamped operation executions.

5.2 Policy Interfaces

Policy interfaces are client-side code responsible for (1) collecting user privacy policies from a user and forwarding them to an identity provider and (2) associating the correct avenance tags with all information, including data that is generated based on observed behaviors. Users need only make privacy decisions once for each attribute or behavior; the policy interface subsequently ensures that all outgoing message are tagged. A message containing particular data is tagged with a policy if and only if something a service provider can do with that data would cause the policy to be violated. This design ensures that the actions required for a user to express their privacy do not become excessive, even if a user interacts with many service providers.

Policy interfaces also ensure that privacy policies relating to observed behaviors can be expressed. This includes policies like "Do not target ads based on the websites I visit" as well as "Do not infer my age from observed behaviors". A policy interface can automatically associate such tags with all HTTP requests, because all observed behaviors are derived from these requests.

5.3 Auditing Parties

Deterrence requires there to be a user-trusted *auditing party* having access to the appropriate records, the ability to detect violations, and the means to ensure that an appropriate party is punished. In order to achieve this:

1. All violations must be detectable. That is, for every violation there must be evidence available that irrefutably indicates that a violation occurred.

2. All violations must be unequivocally attributable to a guilty party.

In an avenance ecosystem there could be many auditing parties, some of which might also function as identity providers or in other roles. Auditing parties access append-only logs in order to detect and attribute policy violations. These logs can be maintained by any party and might record operations performed, messages sent or received, or proofs of policy compliance. Depending on the design of the logging scheme, an auditing party might (1) detect violations just from examining the logs, (2) detect violations by interacting with service providers using test inputs and by observing the outputs, or (3) inspect the source code to detect violations. In the event that a violation is detected, the evidence is communicated to an external legal system for prosecution.

6 Conclusion

Avenance tags are a new proposal for enhancing Internet privacy. They implement a privacy policy language that combines control with secrecy to solve the problem of expressiveness, and they are deployed within the context of a system designed to address the other shortcomings of notice and consent. While the described avenance ecosystem is a long way from practical deployment, we believe it offers an interesting, viable avenue for future work. And we are now attempting a prototype implementation.

References

[1] Anita Allen. *Uneasy access: Privacy for women in a free society.* Rowman & Littlefield, 1988.

[2] Sruthi Bandhakavi, Charles Zhang, and Marianne Winslett. Super-sticky and declassifiable release policies for flexible information dissemination control. In *Proceedings of the 5th ACM Workshop on Privacy in Electronic Society*, pages 51–58, 2006.

[3] Solon Barocas and Helen Nissenbaum. On notice: The trouble with notice and consent. In *Proceedings of the Engaging Data Forum*, pages 12–13, 2009.

[4] Adam Barth, Anupam Datta, John Mitchell, and Helen Nissenbaum. Privacy and contextual integrity: Framework and applications. In *IEEE Symposium on Security and Privacy*, pages 184–198, 2006.

[5] Laurent Bussard, Gregory Neven, and F.-S. Preiss. Downstream usage control. In *IEEE Internation Symposium on Policies for Distributed Systems and Networks (POLICY)*, pages 22–29, 2010.

[6] Jan Camenisch, Abhi Shelat, Dieter Sommer, Simone Fischer-Hübner, Marit Hansen, Henry Krasemann, Gérard Lacoste, Ronald Leenes, and Jimmy Tseng. Privacy and identity management for everyone. In *Proceedings of the ACM Workshop on Digital Identity Management*, pages 20–27, 2005.

[7] K. Cameron. The laws of identity. `http://www.identityblog.com/stories/2005/05/13/TheLawsOfIdentity.pdf`, 2005.

[8] Fred Cate. Principles for protecting privacy. *Cato Journal*, 22:33–57, 2002.

[9] Fred Cate. The failure of fair information practice principles. In Jane K. Winn, editor, *Consumer protection in the age of the 'information economy'*, pages 341–378. Ashgate, 2006.

[10] Fred Cate, Peter Cullen, and Viktor Mayer-Schönberger. Data protection priniples for the 21st century. Oxford Internet Institute, 2013.

[11] Lorrie Faith Cranor and Paul Resnick. Protocols for automated negotiations with buyer anonymity and seller reputations. *Netnomics*, 2(1):1–23, 2000.

[12] Dorothy Denning. A lattice model of secure information flow. *Communications of the ACM*, 19(5):236–243, 1976.

[13] Do not track. `http://donottrack.us`.

[14] William Enck, Peter Gilbert, Byung-Gon Chun, Landon Cox, Jaeyeon Jung, Patrick McDaniel, and Anmol Sheth. TaintDroid: An information-flow tracking system for realtime privacy monitoring on smartphones. In *USENIX Symposium on Operating Systems Design and Implementation*, pages 1–6, 2010.

[15] Federal Trade Commission. Privacy online: A report to congress. `http://www.ftc.gov/sites/default/files/documents/reports/privacy-online-report-congress/priv-23a.pdf`, 1998.

[16] C. Fried. Privacy. *Yale Law Journal*, 77(3):475–493, January 1968.

[17] A. Michael Froomkin. The death of privacy? *Stanford Law Review*, pages 1461–1543, 2000.

[18] Ruth Gavison. Privacy and the limits of law. *Yale Law Journal*, pages 421–471, 1980.

[19] Joseph Goguen and José Meseguer. Security policies and security models. In *IEEE Symposium on Security and Privacy*, pages 11–20, 1982.

[20] Joseph Goguen and José Meseguer. Unwinding and inference control. In *IEEE Symposium on Security and Privacy*, pages 75–86, 1984.

[21] Mark Graber, Donna D'Alessandro, and Jill Johnson-West. Reading level of privacy policies on internet health web sites. *Journal of Family Practice*, 51(7):642–642, 2002.

[22] Internet Society. Global internet user survey. `https://www.internetsociety.org/internet/global-internet-user-survey-2012`, 2012.

[23] Carlos Jensen, Colin Potts, and Christian Jensen. Privacy practices of Internet users: self-reports versus observed behavior. *International Journal of Human-Computer Studies*, 63(1):203–227, 2005.

[24] Jerry Kang. Information privacy in cyberspace transactions. *Stanford Law Review*, pages 1193–1294, 1998.

[25] Günter Karjoth, Matthias Schunter, and Michael Waidner. Platform for enterprise privacy practices: Privacy-enabled management of customer data. In *Privacy Enhancing Technologies*, pages 69–84, 2003.

[26] Lawrence Lessig. The architecture of privacy. *Vanderbilt Journal of Entertainment Law & Practice*, 1:56–65, 1999.

[27] Jed Liu, Michael George, Krishnaprasad Vikram, Xin Qi, Lucas Waye, and Andrew Myers. Fabric: A platform for secure distributed computation and storage. In *Proceedings of the ACM Symposium on Operating Systems Principles*, pages 321–334, 2009.

[28] Petros Maniatis, Devdatta Akhawe, Kevin Fall, Elaine Shi, Stephen McCamant, and Dawn Song. Do you know where your data are? Secure data capsules for deployable data protection. In *Proceedings of the 13th USENIX Conference on Hot Topics in Operating Systems*, 2011.

[29] Catherine Jensen McCollum, Judith Messing, and L. Notargiacomo. Beyond the pale of MAC and DAC—Defining new forms of access control. In *IEEE Computer Society Symposium on Research in Security and Privacy*, pages 190–200, 1990.

[30] Aleecia McDonald, Robert Reeder, Patrick Gage Kelley, and Lorrie Faith Cranor. A comparative study of online privacy policies and formats. In *Privacy Enhancing Technologies*, pages 37–55, 2009.

[31] Aleecia M McDonald and Lorrie Faith Cranor. The cost of reading privacy policies. *I/S: A Journal of Law and Policy for the Information Society*, 4:543–568, 2008.

[32] Marco Casassa Mont, Siani Pearson, and Pete Bramhall. Towards accountable management of identity and privacy: Sticky policies and enforceable tracing services. In *Proceedings of the 14th IEEE International Workshop on Database and Expert Systems Applications*, pages 377–382, 2003.

[33] Kiran-Kumar Muniswamy-Reddy, David Holland, Uri Braun, and Margo Seltzer. Provenance-aware storage systems. In *USENIX Annual Technical Conference*, pages 43–56, 2006.

[34] M.-H. Carolyn Nguyen, Peter Haynes, Sean Maguire, and Jeffrey Friedberg. A user-centered approach to the data dilemma: Context, architecture, and policy. In *Digital Enlightenment Forum*, September 2013.

[35] Helen Nissenbaum. *Privacy in Context: Technology, Policy, and the Integrity of Social Life*. Stanford University Press, 2009.

[36] Helen Nissenbaum. A contextual approach to privacy online. *Daedalus*, 140(4):32–48, 2011.

[37] Organization for Economic Co-operation and Development. Guidelines governing the protection of privacy and transborder flows of personal data, October 1980.

[38] Joseph Reagle and Lorrie Faith Cranor. The platform for privacy preferences. *Communications of the ACM*, 42(2):48–55, 1999.

[39] Jeffrey Reiman. Privacy, intimacy, and personhood. *Philosophy & Public Affairs*, pages 26–44, 1976.

[40] Robert Sloan and Richard Warner. Beyond notice and choice: Privacy, norms, and consent. Technical report, Chicago-Kent College of Law Research Paper, 2013.

[41] Geoffrey Smith and Dennis Volpano. Secure information flow in a multi-threaded imperative language. In *Proceedings of the 25th ACM Symposium on Principles of Programming Languages*, pages 355–364, 1998.

[42] Daniel Solove. Privacy self-management and the consent paradox. *Harvard Law Review*, 126(7):1880–1903, 2013.

[43] G. Edward Suh, Jae Lee, David Zhang, and Srinivas Devadas. Secure program execution via dynamic information flow tracking. *ACM SIGOPS Operating Systems Review*, 38(5):85–96, 2004.

[44] David Sutherland. A model of information. In *National Computer Security Conference*, pages 175–183, 1986.

[45] Dennis Volpano and Geoffrey Smith. Probabilistic noninterference in a concurrent language. *Journal of Computer Security*, 7(2):231–253, 1999.

[46] Alan Westin. Privacy and freedom. *Washington and Lee Law Review*, 25(1):166, 1968.

[47] Jennifer Widom. Trio: A system for integrated management of data, accuracy, and lineage. Technical report, Stanford InfoLab, 2004.

[48] Marianne Winslett, Charles Zhang, and Piero Bonatti. PeerAccess: A logic for distributed authorization. In *Proceedings of the 12th ACM Conference on Computer and Communications Security*, pages 168–179, 2005.