A Statistical Matching Approach to Detect Privacy Violation for Trust-Based Collaborations

Mohamed Ahmed, Daniele Quercia and Stephen Hailes
Department of Computer Science, University College London, London, WC1E 6BT, UK.
{M.Ahmed, D.Quercia, S.Hailes}@cs.ucl.ac.uk

Abstract

Distributed trust and reputation management mechanisms are often proposed as a means of providing assurance in dynamic and open environments by enabling principals to building up knowledge of the entities with which they interact. However, there is a tension between the preservation of privacy (which would suggest a refusal to release information) and the controlled release of information that is necessary both in order to accomplish tasks and to provide a foundation for the assessment of trustworthiness. However, if reputation-based systems are to be used in assessing the risks of privacy violation, it is necessary both to discover when sensitive information has been released, and then to be able to evaluate the likelihood that each of the set of principals that knew that information was involved in its release.

In this paper, we argue that statistical traceability can act as a basis for reaching a proper balance between privacy and trust. To enable this, we assume that interacting principals negotiate service level agreements that are intended to constrain the ways in which personal information may be used, and then monitor violations, ascribing likelihoods of involvement in release using an approach based on statistical disclosure control. Even though our approach cannot guarantee perfect privacy protection for personal information, it provides a framework using which detected privacy violation can be mapped onto a measure of accountability, which is useful in deterring such violation.

1 Introduction

It is a tautology to argue that pervasive systems, if and when they become a commercial reality, will necessarily invade many aspects of our lives. One of the real questions at issue is, given that they will monitor our environment, to whom is should that information be available? The answer to that question is inevitably policy based and is concerned with the identification and release of information relative to its subjective value. But the question of how violations of policy should be determined is still open [3].

Factors such as environment size, functional and behavioural heterogeneity, mobility and unpredictability of interaction are often cited as complicating the challenges for effective management and enforcement of privacy constraints. For this reason, reputation and trust management systems are proposed to support the provision of the required levels of assurance in a flexible and scalable manner by *locally* discriminating between the entities with which a principal should interact, based on their trustworthiness in some given context.

Trust and reputation mechanisms function on the premise that reliable and useful information concerning the entities they judge is easily accumulated and disseminated. Thus, amongst their minimum requirements is to be able to collect and identify the attributes to associate with principals in support of the formation and evolution of reputation and trust profiles. However, this premise differs somewhat from the traditional role of privacy management, which aims to reduce the linkability of information to principals. Therefore, in trust based collaborative environments, perfect privacy protection will undoubtedly hinder the formation and evolution of trust because less information will be available for correlation.

Traditionally, privacy concerns are addressed either through: (i) *Point of interaction* discrimination or, (ii) *Point of use* enforcement. Discrimination mechanisms typically employ policy defined incremental information release schemes, to determine what is appropriate for disclosure in a given interaction [14], or manage numerous pseudonyms so as to disperse the information that could be attributed to a given identity [4, 11]. However, for information providers, incremental information release can be reduced to the most maximally valued attribute (i.e. highest value credentials) being disclosed more frequently than would otherwise be desirable. In time, this property can easily lead to homogenising a user's credential set and, consequently, uniquely identifying them, thereby rendering the countermeasures null.

Enforcement mechanisms apply policy constraints to the transformation of data throughout its entire lifecycle. This approach relies on tagging data with its policy constraints, such that operations cannot be applied without referring to them. Typically, trusted computing platforms are employed, whether through hardware control as with the TCPA initiative [13] or software control as in the case of the Tagged OS [10]. However, the envisioned openness and heterogeneity of emerging networking environments with few universally trusted third parties complicates the task of policy enforceability. Therefore, while information hiding techniques can effectively obfuscate data, they cannot control its propagation. Though Trusted Computing Platforms can control the propagation of data, the organisational, and functional heterogeneity in an open environment means that their existence cannot be relied upon.

In this paper, we propose an audit mechanism that attempts to address the first issue raised in tackling the control of information propagation in open, trust-based environments, namely identifying the perpetrators. In our framework, information collecting principals sign service level agreements similar to P3P specifications [5] for which they are then accountable. Information providers support accountability by actively seeking and then analysing the patterns of leakage using an approach similar to risk assessment in statistical disclosure control [6]. Based on their assessments, principals may then select appropriate enforcement mechanisms including, for example, a withdrawal of patronage or a reduction in the trust value associated with that principal, which is then propagated to others.

The remainder of the paper is organised as follows: section 2 presents some related work that looks at privacy concerns, focusing on ubiquitous networking environments. This is followed a presentation of a simple scenario in Section 3. Section 4 explains the concept of protected service level agreements and is followed by Section 5 which proposes algorithms for detecting privacy leakage. Section 6 presents a brief discussion of the algorithms, and Section 7 concludes the paper.

2 Related Work

To support "accountability in a world of invisible services", Langheinrich [8] adapts the P3P specification [5] to enable implicit and active announcement of a system's privacy policies at the point of service discovery. The architecture enables devices and users to inspect the information handling policies of service providers before engaging in transactions and to monitor the life-cycle of their information through requests of audit trails of its release. Lederer et al. [9] propose a similar model that takes into account legal, economic, contextual, and the subjective perception on end-user privacy as factors. The common thread behind the approaches is that it is possible to rely upon social and legal norms to enforce privacy constraints.

The Sticky Policies approach of Casassa Mont et al. [10], which is aimed at B2B scenarios, enables users to attach enforceable policy constraints to their information, and focuses on giving enforcement powers to users. However, the requirement of trusted third parties to provide IBE encryption services and trusted computing platforms to enforce policy constraints may be too strong an assumption for massively heterogeneous environments.

Amongst others, Winslett [14] proposes a method for iteratively disclosing personal credentials between communicating entities based on 'need to know'. However, the incomplete definition of incremental credential release, and inequality of information value to users make it difficult to specify policies.

Seigneur et al. [11] propose an architecture in which principals can use pseudonyms to establish unlinked social profiles in several communities. In this approach, each principal selects a pseudonyms according to its current community - with the aim of making identity and attribute association more costly for an attacker, thereby alleviating some of the privacy concerns. However, besides the added complexity of multiple pseudonym management, this approach suffers from two drawbacks. Firstly, it is required that different community pseudonyms be linked in order to satisfy high assurance requests. The result is that a principal may be disposed to give up some of its privacy and link parts of its pseudonym chain in order to provide necessary assurances to be trusted. In doing this, the principal exposes its interactions in separate communities leading to a partial loss of privacy. Secondly, the actions taken under different pseudonyms in several communities can be correlated through community collusion. Though cryptographic mechanisms such IDEmix [4] can be used to complement this approach and enable anonymous credentials, the need for linkability to support trust reduces their effectiveness.

Mechanisms for providing privacy protection using statistical disclosure control [6, 1] are already popular in several fields. However, the term 'statistical disclosure control' typically refers to the ability to prevent the identification of individual population units from previously unknown confidential data, e.g. census data or medical records. In this work, we have been motivated by disclosure risk assessment methods [7, 15, 12], which look at possible methods of minimising the proportions of population units that can be identified by their unique combinations of attributes. However, we have reversed the role of statistical disclosure control by examining how to use potentially unique combinations of attributes to identify population units to evaluate their likelihood of participating in the violation of privacy agreements.

3 Scenario

As an example, Alice interacts with the services offered in a Ubicomp environment such as location tracking services, micro-payments providers, network resource providers and so on. As such, Alice necessarily reveals and generates potentially private or confidential information with each principal but may not want to have this information disclosed publicly.

Let us assume that Alice wishes to find a service provider that will accept her micro-payments, but also give her credit to pay for her service usage in instalments. Alice decides to use a reputation-enabled mechanism to find a service provider and she does not mind service providers sharing their evaluations of her. Alice does not, however, want them to distribute the raw evidence she generates during the lifecycles of her transactions and some of the evidence that she provides to gain access.

Though this scenario is slightly over-simplified, it illustrates a real problem; specifically how does Alice know whether her privacy agreements are respected? Accordingly, Alice may wish to find out whether the private information she has generated in previous interactions has been disclosed.

4 Service Level Agreement

Flexible and privacy orientated policy specification languages such as the Enterprise Privacy Authorization Language (EPAL) [2] which extends the P3P framework [5], enable policy writers to express complex constraints on data at different levels such as the context of interaction, the network state and so on. However, they do not typically provide enforcement or accountability mechanisms for detecting the occurrence of violations. Our proposed mechanism relies heavily on the construction of protected service level agreements (PSLA). It is assumed that, prior to an interaction, principals will negotiate privacy policies, for example using P3P to determine how the information attributed to their transaction is managed after the transaction has completed. This information is translated into PSLA between principals and is used to determine for which attributes to look when evaluating breaches of confidence.

A protected service level agreement in this context lists an arbitrary set of *attributes*, defining how a principal expects its information to be managed. For example Alice may define a PSLA that lists she is interested in keeping her {`salary, location, address`} private. Attributes list an arbitrary *category* and its *value*, for example {`salary, 10K, private`}. Attributes are associated with a scope that states whether they are public, private or otherwise, and may be of type *evidence* or *evaluation*.

Evidence may be locally generated, e.g. the movement of Alice in her neighbourhood, or it may be collected from the principal, e.g. Alice's salary. Evaluations state a context of interaction and arbitrary value of satisfaction, generated by the evaluating party. For example, in the scenario above, an evaluation may be made of Alice's ability to make her payments reliably, based on the regularity of her payments.

5 Detection of private attribute leakage

A malicious entity may not respect one or more of its service level agreements and may disclose private information within its community. For a privacy-concerned principal, it is in its interest to investigate whether its confidential attributes have been disclosed. The aim of the measures we propose is to enable a principal to evaluate whether some of its private attributes have been released and to give some indication of who may have released the information. The process for accomplishing this is as follows: (i) query for the attribute in question within a community; (ii) given the responses, determine whether the attribute value has been leaked; (iii) if so, given the principals to whom the attribute has been released legitimately, determine the probability that each principal is responsible.

5.1 Determining leakage

In our model, the principal about whose privacy we are concerned, Alice, is assumed to possess a set of data items or attributes ($A_i : (a_1, a_2, \ldots, a_{|A|})$). A given transaction can be represented as the release of a subset of A. Thus each attribute can be associated with a frequency of appearance ($f(a_i)$) representing the number of times it is disclosed:

$$f(a_i) = \sum_{j \in E} I(a_i \in A_j), \tag{1}$$

where E is the set of principals in the community, A_j is the set of attributes released to a given principal j and $I()$ is the indicator function: $I(B) = 1$ if B is true, otherwise $I(B) = 0$.

To determine whether an attribute has been leaked, Alice has two choices: wait until there is activity that could have come from the leak of the attribute (e.g. particular types of SPAM, junk mail, etc.), or proactively query the environment to determine whether it contains personal information that can be linked to attributes she has released. It is clearly the case that if the people to whom the information is leaked do not respond to queries and do not use the information in a way that can be obviously linked to its release, then there is no way to determine that a leak has occurred. However, we assume that those to whom the information is leaked make use of that information to which

they have access for the purposes of gain. If this is the case, then there is a good chance that intelligence gathering will lead to a view on which attributes were leaked.

One very simple mechanism for determining whether release has occurred would be to count the frequency of its appearance. Because the frequency of appearance for an attribute expresses how often it is disclosed, Alice may define an arbitrary disclosure threshold (dt) detailing the frequency of appearance she expects for a given attribute, given the PSLAs into which she has entered. If Alice has queried the environment she may evaluate a disclosure ratio (dr) for the attribute in which she is interested relating the appearance of the attribute in the environment to the number of times it was legitimately released. If Alice repeatedly queries the environment and receives in response information that she can link to the release of certain attributes then, in effect, her queries return a series of sets of attributes $\mathcal{T}_i$. Now, she can define the disclosure ratio for attribute a_i:

$$dr(a_i) = \frac{\sum_j I(a_i \in \mathcal{T}_j)}{f(a_i)}. \quad (2)$$

Alice can define, for each attribute, the upper limit of how often she would expect to see a given attribute (a_i) in the environment. Therefore a querying entity can identify the breach of a PSLA for an attribute by examining whether its disclosure ratio is greater than its disclosure threshold ($dr(a_i) > dt(a_i)$). For example, for a confidential attribute, we may define its disclosure threshold $dt(a_i) = 0$, i.e. it should never be disclosed. Alternatively, for information that is sensitive but not secret, Alice might be concerned to see it appear several thousand times in the environment if we released it to a few tens of people, whereas a slightly greater number of responses than releases would not concern her.

5.2 Determining the culprit

Once it has been determined that some attribute (a_i) has been leaked, i.e. $dr(a_i) > dt(a_i)$, we attempt to identify the culprit. If an attribute that is deemed secret has been released only to one person, and then appears elsewhere in the environment, we have very good grounds to suspect that individual of having leaked it[1]. The process of marking information in this way is, in effect, a form of steganography - at a simple level, Alice might misspell her address when giving it to different people giving sufficient difference to ensure that each individual receives a unique address, but not making the differences so great that her mail is misdelivered. Unfortunately, in general, we cannot guarantee the uniqueness of individual attributes released to each principal. However, we note that attributes are rarely released singly, and the *set* of attributes known to an individual may, if they are found in the environment, indicate a greater likelihood of them having been involved in the release than others. Consider table 1: Alice has released attributes a_1 to a_3 to hosts h_1 to h_4 as shown. Assuming that Alice has queried the environment and discovered set $\mathcal{T} = \bigcup_i \mathcal{T}_i = \{a_1, a_2, a_3\}$. In order to see this set, either h_1 must have been involved in the release of some or all of the information it knew, or the set $\{h_2, h_3, h_4\}$ must have released the information. If we know nothing about the dispositions of the hosts towards information release, then we could conclude that it is less probable for three hosts to release information than for a single host to be somehow involved. In this section, we analyse the probabilities obtainable from this observation; however, in section 6 we discuss how reputation information may be used to weight the probabilities.

In the absence of information to bias the calculation, we assume that it is equiprobable that each host knowing an attribute was involved in its release. Now, the probability that the host h_n revealed a single attribute a_i may be expressed as:

$$p^{\{a_m\}}_{\{h_n\}} \equiv p^m_n \quad = \quad \begin{cases} \frac{1}{|\mathcal{H}(a_m)|}, & \text{if } h_n \in \mathcal{H}(a_m) \\ 0, & \text{if } h_n \notin \mathcal{H}(a_m) \end{cases} \quad (3)$$

where $\mathcal{H}(a_m)$ is the set of hosts that know the attribute a_m.

From Equation 3, the probability that the host h_n revealed an attribute set $A = \{a_1, \ldots a_{|A|}\}$ can be derived:

$$p^A_{\{h_n\}} \equiv p^A_n = \prod_{a_m \in A} p^m_n \quad (4)$$

And the probability that a set of hosts, H, revealed an attribute set A is given by:

$$p^A_H = \prod_{a_m \in A} \sum_{h_n \in H} p^{a_m}_{h_n} \quad (5)$$

From these, we can calculate the probability that a set of hosts H_α is involved in some way in disclosing an attribute set $\mathcal{T}$:

$$P(H_\alpha \text{ is involved in disclosing } \mathcal{T}) = 1 - (p^{\mathcal{T}}_{H-H_\alpha}) \quad (6)$$

Table 1 illustrates the results of evaluation, for the case where $\mathcal{T} = \{a_1, a_2, a_3\}$. Note that the probabilities do *not* sum to one, since this is the probability of involvement in the leakage of $\mathcal{T}$, not the probability that a host leaked $\mathcal{T}$ precisely.

6 Discussion

The calculations we present give the likelihood of an entity revealing an attribute set known to them. Since we

[1] However, we cannot be certain of this, because there is a difference between information and data - we know we have released certain data to that individual, but we may also have released the same information through a different channel.

	a_1	a_2	a_3	$P(h_n)$
h_1	*	*	*	$\frac{7}{8}$
h_2	*			$\frac{1}{2}$
h_3		*		$\frac{1}{2}$
h_4			*	$\frac{1}{2}$

Table 1. Domain knowledge and evaluation

are concerned with privacy in trust and reputation management systems, we may refine the proposition in Equation 3 by taking into account the disposition of an entity to cheat. This disposition may emerge over time, but one way of assessing it simply in terms of a local value of trust since this is precisely intended to reflect observed evidence of an entity's behaviour. Thus, an alternative to the simple initial probability calculation would be one that uses this weighting factor:

$$p_n^m = \begin{cases} \frac{pd_{h_n}}{\sum_{i \in \mathcal{H}(a_m)} pd_i}, & \text{if } h_n \in \mathcal{H}(a_m) \\ 0, & \text{if } h_n \notin \mathcal{H}(a_m) \end{cases} \quad (7)$$

Where pd_n is a trust value related to the probability that h_n defects. This view suggests a co-dependency between the trust value for a host and the probability that that host defected. We are in the process of exploring a scheme in which likelihood of defection is used in the calculation of reputation which is itself iteratively used to assess the likelihood of whether a node has defected or not. This work is intended both to ensure convergence and to explore the extent to which such schemes can be made strategy proof.

7 Conclusion

Information propagation is necessary to support emerging reputation and trust-aware architectures that are proposed for the management of security in ubiquitous networking environments. However, there is a tension between privacy protection and the availability of trust information on which to base future interaction. Thus, as many researchers have argued, a balance needs to be reached between privacy and trust. We argue that crucial to any balance mechanism is its enforceability. In the first steps to supporting this proposition, we have presented a mechanism for specifying how leakages of attributes may be detected and how perpetrators may be identified.

8 Acknowledgment

The authors would like to thank BT for funding under the MARS project and the EC for funding under SEINIT and RUNES.

References

[1] N. R. Adam and J. C. Wortmann. Security-Control Methods for Databases: A Comparative Study. volume 21, pages 515–556. ACM Press, 1989.

[2] P. Ashley, S. Hada, G. Karjoth, C. Powers, and M. Schunter. Enterprise Privacy Authorization Language (EPAL 1.2). Technical report, W3C Recommendation, 2003.

[3] Y. Beres, P. Bramhall, M. C. Mont, M. Gittler, and S. Pearson. On the Importance of Accountability and Enforceability of Enterprise Privacy Languages. In *W3C Workshop on the long term Future of P3P and Enterprise Privacy Languages*, June 2003.

[4] J. Camenisch and E. V. Herreweghen. Design and Implementation of the Idemix Anonymous Credential System. In *Proceedings of the 9th ACM conference on Computer and communications security (CCS '02)*, pages 21–30, 2002.

[5] L. Cranor, B. Dobbs, S. Egelman, G. Hogben, J. Humphrey, M. Langheinrich, M. Marchiori, M. Presler-Marshall, J. Reagle, M. Schunter, D. A. Stampley, and R. Wenning. The Platform for Privacy Preferences 1.1 (P3P1.1) Specification. Technical report, W3C Recommendation, 2005.

[6] M. Elliot. Statistical Disclosure Control. In *Encyclopedia of Social Measurement*. Elsevier, 2003.

[7] S. E. Fienberg, U. E. Makov, and A. P. Sanil. A Bayesian Approach to Data Disclosure: Optimal Intruder Behaviour for Continous Data. *Journal of Official Statistics*, 13:75–89, 1994.

[8] M. Langheinrich. A privacy awareness system for ubiquitous computing environments. In *UbiComp '02: Proceedings of the 4th international conference on Ubiquitous Computing*, pages 237–245. Springer-Verlag, 2002.

[9] S. Lederer. Everyday Privacy in Ubiquitous Computing Environments. In *Ubicomp 2002 Workshop on Socially-informed Design of Privacy-enhancing Solutions in Ubiquitous Computing*, 2002.

[10] M. C. Mont, S. Pearson, and P. Bramhall. Towards Accountable Management of Identity and Privacy: Sticky Policies and Enforceable Tracing Services. In *Database and Expert Systems Applications (DEXA) Workshop*, pages 377–382, Prague, Czech Republic, September 2003.

[11] J.-M. Signeur and C. D. Jensen. Trading Privacy for Trust. In *2nd International Conference on Trust Management*, volume 2995, pages 93–107, 2004.

[12] C. J. Skinner and M. J. Eliot. A measure of disclosure risk for microdata. *Journal of the Royal Statistical Society*, 64:855– 867, 2002.

[13] Trusted Computing Group. Trusted Computing Platform Alliance (TCPA), Main Specification Version 1.1b. Technical report, Trusted Computing Group, Incorporated, https://www.trustedcomputinggroup.org, 2003.

[14] M. Winslett. An Introduction to Trust Negotiation. In *Proceedings of the First International Conference on Trust Management (iTrust 2003)*, LNCS 2692, pages 275–283, Heraklion, Crete, Grece, May 28-30 2003.

[15] W. E. Yancey, W. E. Winkler, and R. H. Creecy. Disclosure Risk Assessment in Perturbative Microdata Protection. In *Inference Control in Statistical Databases, From Theory to Practice*, pages 135–152. Springer-Verlag, 2002.