PERSONAL ATTRIBUTES AND PRIVACY

How to ensure that private attribute management is not subverted by datamining.

Howard Chivers
Department of Computer Science, Univeristy of York, Heslington, York, YO10 5DD
chive@cs.york.ac.uk

Abstract: The aggregation of personal attributes into user profiles is a significant privacy concern. Existing attribute management systems support the controlled release of attributes and unlinkability between session protocols, but do not address the problem that attributes distributed in this way may be data-mined for features that allow user profiles to be reconstructed.

This paper identifies the aggregation problem as the missing element in the protection of personal attributes, and introduces attribute management principles that are sufficient to provide an overall framework to protect users from profile aggregation. The principles are clarified by formalizing them as constraints on primitive operations in a service-based architecture, and this analysis is the basis for a proof that they support system wide privacy. These results are of particular value in the decomposition of business processes into services, and a location-privacy problem is used to show how they can be applied in practice.

Key words: privacy protection, distributed-system security, web-services security, privacy-enhancing technology, electronic commerce, pseudonymity

1. ATTRIBUTE MANAGEMENT PRINCIPLES

1.1 Introduction

Privacy is a fundamental right in the UN Declaration of Human Rights, and in the European Union a Data Protection Directive [1] is enforced by legislation in member nations [2]. These data protection requirements

embody the concepts of *necessity, purpose* and *consent*: personal data can be processed only if it is necessary for the application, and only for the purpose for which it was provided. Although the US has not adopted the same legislative approach, similar principles are described in the influential Code of Fair Information Practices [3].

The subject of privacy is of growing importance because of the ease with which information can be assembled and processed, and because of public sensitivity about commercial trading in personal profiles [4]. Technical trends toward highly distributed systems also exacerbate privacy management issues, including the balance between privacy and accountability [5].

These social, statutory and technical pressures have motivated researchers to develop protocols that limit access to personal data. The general approach is to regard any fact about an individual as a *private attribute* whose release should be subject to a policy that takes into account the subject's consent and the needs of the application. For example, Shibboleth [6] manages user authentication in this way, and researchers have proposed pseudonomous PKI attribute certificates [7]; other approaches do not provide enforcement, but specify contracts [8] or policies [9].

These systems use temporary pseudonyms to ensure that different user sessions cannot be linked by a primary identity. The type and degree of unlinkability depends on the protocol, but the pseudonymous user must still present private attributes in order to access the service. Unfortunately, little attention has been given to the threat that these attributes may be directly consolidated by data mining, bypassing the unlinkability of the access protocol and enabling the profile aggregation that is precisely the issue of public concern.

This paper addresses the problem of attribute aggregation; it shows how private attributes can be partitioned and their distribution constrained in order to ensure that attribute release policies set by a pseudonymous user cannot be subverted by aggregation in the distributed system.

The remainder of this paper is in three parts; the first is a discussion of the problem, and proposes *attribute management principles* as criteria for system design. The second part clarifies these principles by formalizing them as constraints on the services and security tokens of a service-based distributed system. Proof that these constraints are effective is given in Appendix A. Finally, a location-privacy problem is used to show how these results can be applied during the system design process.

1.2 Privacy and Aggregation Management

User concern about profile aggregation suggests that privacy is fundamentally related to the linkage or correlation of private data [4], and although data protection legislation is not framed directly in this way, it supports this view: the UK legislation [2] defines a relevant filing system as *structured*, and the linkage of census data to individual population records led to the landmark constitutional challenge of the 1983 German census [10].

The defining nature of linked data is also argued by Wallace [11] who defines anonymity as *noncoordinatability of traits in a given respect.* She provides a powerful example: a criminal (the Unabomber) was completely specified by the single trait of his crimes, but remained anonymous until another trait (writing style) became known. This suggests that linkability of personal traits is a more fundamental issue than the number of individuals determined by any given attribute.

Managing the aggregation of private data is therefore a defining feature of privacy-supporting technologies, implying that *the extent that personal attributes remain independent constitutes the degree to which privacy is maintained.* This definition has the benefit that it naturally includes threats such as aggregation or data mining. Previous work has addressed unlinkable privacy mechanisms, but has not made the case for the fundamental importance of unlinkability as a criterion for technical privacy.

Authentication by attribute, rather than identity, requires an authority such as the user's organization to vouch for the attribute, and a temporary binding between the user and the target service that avoids providing further information. The user expects that this arrangement will preserve privacy by preventing the aggregation of attributes released in different sessions.

Systems that pseudonymously manage attribute release have already been mentioned [6, 7]. There is also an important body of research on unlinkable protocols; much of which is based on blind signatures [12] and variations [13] that provide accountability, or on MIX networks [14]. However, these still require a user to provide authentication information to obtain a service.

Existing work therefore provides only part of the solution. It is also necessary to ensure that private attributes released to services cannot be aggregated directly, for example, by using statistical techniques to identify and exploit overlapping values. This paper proposes attribute management principles to address this problem; unlinkable protocols are important supporting mechanisms, but are not discussed further.

1.3 Indirect Transactions and Attribute Partitioning

Consider the flow of private data in an electronic purchase (for the sake of discussion, of some CDs). The identity of the CDs is not a privacy concern unless they can be associated with the user making the purchase, which requires an identifier such as a bank account number. Privacy concerns multiply as more attributes are collected, for example, adding the user's address provides both a marketing target and information about social and economic group.

The user reveals information about buying preferences by selecting the product and may also provide credit card information to pay for the purchase. The association of these two attributes of private data allows the long term tracking of the user's buying profile. This violates the data protection *purpose* principle, because it is not necessary.

The supplier might argue that credit information is necessary to perform their contract with the subject, but many forms of payment, including cash, do not identify individual accounts. Even complex financial transactions can be carried out indirectly, for example via escrow accounts or international letters of credit. It is arguable, therefore, that this pattern of electronic business has not been established because of a financial or business precedent, but because it is technically convenient.

An alternative pattern for the transaction is that the user obtains an opaque 'electronic check' from their bank that is presented to the supplier. This provides separation of purpose between the private attributes held by the bank and those available to the supplier. There are a number of possible implementations of electronic cheques[1]; this paper uses the generic term 'authorization token', implying that such a token does not carry private data.

This example motivates general principles for the design of systems that manage attribute aggregation: private data provided to any service provider should be partitioned by *trait* (or type); and where a service provider needs to invoke the services of another, then the data subject's authority should be conveyed by an opaque authorization token, rather than by the provision of additional personal information.

1.4 Attribute Management Principles

These ideas can be expressed as attribute management principles that are sufficient to ensure that *no part of a system can aggregate an individual's private attributes:*

[1] Such as electronic cash [12] [15], but recall that this paper is concerned with the management of personal attributes themselves, not the underlying protocols.

1. An individual's private attributes should be grouped into a number of sets (named by 'trait' or 'type').
2. Private attributes provided to any single service provider should be limited to a single trait.
3. Where a service provider needs to invoke the services of another, and the second requires attributes of a different trait, delegation of the user's authority should be via opaque tickets, rather than by the provision of additional private data.

These principles do not specify how private attributes should be grouped into *Traits*, a privacy policy for a specific system needs to balance privacy and feasibility when making this choice. A trait should represent a class of information about a user, for example: buying preferences for particular products; address and location; employment; bank and credit information.

The next section formalizes these principles, to clarify the constraints that they represent, and this provides the basis to prove that they are sufficient to maintain privacy.

2. ANALYSIS

2.1 Introduction

The attribute management principles are intuitively appealing, but their usefulness must be demonstrated in the context of the rich range of services that are supported by practical distributed systems. From the privacy perspective these services can all be viewed as moving data of various types and origins between *principals*[2]. This section models the movement of data in a service-based system in order to clarify the constraints implied by the attribute management principles, and shows that they result in the desired privacy properties for the system as whole.

Both the operation of services, and the distribution of security management information are important. The primitive operations modeled are therefore:

- The creation of authorization tokens.
- The distribution of authorization tokens.
- The use of an authorization token to invoke a service.
- The transfer of an authentication token, or other privacy sensitive data, between principals.
- The transfer of non-privacy sensitive data between principals.

[2] The term *principal* is used to denote an application that can both invoke and provide a service, and is administered by a single authority

2.2 Formal Model

The following model captures the flow of information between principals as a directed graph. The main data types are modeled as vertices and the possible movement of information as directed edges. The information flow implied by each service is expressed as a relation between vertices, and the edge set is the union of these relations. This form of modeling is conservative, because of the implicit assumption that all information flows are transitive; however, it provides a compact and direct representation of the service primitives and allows straightforward reasoning about the overall system. The graph is specified using set theory, in Z syntax [16].

2.2.1 Static types and relations

The base types in this model are the disjoint sets introduced in table 1.

Table 1. Base types

Type	Name	Description
P	Principal	Service providers and users
A	Private Attribute	Private Data, including authentication data
D	Public Data	Data that is not private
K	Authorization Token	An opaque token
T	Transaction	An atomic business transaction
Y	Attribute Trait	The trait, or type, of a private attribute

Principals are active system entities that can provide or invoke services; *Private Attributes, Public Data* and *Authorization Tokens* are types of data that can be accessed by Principles, or transferred between them.

Transactions are data types that model system state. Distributed business transactions are often a sequence of operations with intermediate state held by the service provider. For example, an on-line purchase may involve the selection of goods, followed by setting delivery and payment options. Atomic transactions are an instance of this more general case.

Traits (Y) are used to constrain the model, rather than represent information sources or sinks. This has the effect of neglecting information flow through the type system, which is justifiable because Traits are expected to be static and knowledge of the Traits in the system is not a privacy concern.

The set of vertices in the model is therefore the union of the base types, except **Y**:

$$V = P \cup A \cup D \cup K \cup T \tag{1}$$

Both Private Attributes and Principals have identifiable Traits. It is also convenient to define a trait matching relation between Private Attributes and Principals:

$$YP = P \rightarrow Y \text{ [3]} \tag{2}$$

$$YA = A \rightarrow Y \tag{3}$$

$$MATCH = \{(a, p) \mid \exists k((a, k) \in YA \wedge (p, k) \in YP)\} \tag{4}$$

Both Public Data and Private Attributes are owned by specific Principals:

$$PUB = D \rightarrow P \tag{5}$$

$$PRIV = A \rightarrow P \tag{6}$$

The information flows present in the initial system are therefore:

$$IS = PUB \cup PRIV \tag{7}$$

2.2.2 Transactions

As noted above, Transactions record state. Since we are not concerned with functional behaviour it is sufficient to record data items that have contributed to state as a vertex from that data item to a Transaction. The primitive operations are directly modeled in this way, together with any constraints required to uphold the attribute management principles.

A Transaction is owned by particular Principal, so any data accumulated by that Transaction is also available to the Principal:

$$TA = T \rightarrow P \tag{8}$$

A Transaction may make use of local Public Data:

$$TB = \{(d, t) \mid \exists p((t, p) \in TA \wedge (d, p) \in PUB)\} \tag{9}$$

[3] When relations are introduced, $Z = A \times B$ represents $\{(a, b) \mid a \in A \wedge b \in B\}$ as usual, but additionally where A and B are sets of vertices, the pair (a,b) is a directed edge from a to b. Functions such as $Z = A \rightarrow B$ similarly have their usual meaning with the additional connotation of a directed edge.

Creating an Authorization Token. The purpose of the token is to identify one or more Transactions. The Token must not carry information about the state of any Transaction.

$$CA = K \leftarrow T \tag{10}$$

Distributing an Authorization Token. The Token can be distributed to any Principal.[4]

$$DA = K \times P \tag{11}$$

Distributing an Authentication Token. This binds Private Attributes from one Principal to a Transaction owned by a second Principal. It models the provision of user attributes for the purpose of authentication, or more generally any operation that transfers private data between Principals.

This operation is constrained by attribute management principle 2, limiting the distribution of private data to Principals of the correct Trait.

$$BA = \{(a,t) \mid \exists p((t,p) \in TA \wedge (a,p) \in MATCH)\} \tag{12}$$

Using an Authorization token to access a service. This models the use of an authorization token to invoke a service on an existing Transaction. Of course, data may be returned to the Principal that invokes this service. This requires an additional constraint to ensure privacy: the only bindings to the Transaction must be from Private Attributes that are either private to the Principal that invoked the service or have the same Trait as the Principal that invoked the service.

$$SA = \{(t,p) \mid \forall a((a,t) \in BA \Rightarrow [(a,p) \in PRIV) \vee (a,p) \in MATCH])\} \tag{13}$$

Public Data. To complete the graph it is necessary to record data flows that involve public data. Any item of public data can influence a transaction:

$$PA = D \times T \tag{14}$$

and data flows between public data items are not constrained:

$$PB = D \times D \tag{15}$$

[4] In a practical system, possession, or first use, of a token may confer access to a service, or there may be constraints on which Principals could make use of a token, either statically encoded in the access policy of the service or dynamically encoded in the state of the Transaction. These additional constraints are beyond the scope of this paper.

Completing the model. The edge set of information flow paths in the system can now be constructed:

$$E = IS \cup TA \cup TB \cup CA \cup DA \cup BA \cup SA \cup PA \cup PB \quad (16)$$

2.3 The Privacy Proposition

Informally, the privacy proposition is:

Any flow of information from a Private Attribute to a Principal is either from the Principal's own Private Attributes, or from a Private Attribute of a Trait that the Principal is allowed to process.

A formal account of this proposition and its proof is given in Appendix A. This demonstrates that the system has the property that no service provider is able to reconstruct the private data associated with another Principal by invoking any sequence of system services.

2.4 Summary of Constraints

This analysis clarifies the constraints that are required in a system to prevent aggregation of personal data. The constraints embodied in the model are:

1. A Principal must be assigned a single Trait (Eq. 2).
2. Each Private Attribute is a member of a single Trait (Eq. 3).
3. A Principal has an identified set of Private Attributes (Eq. 6).
4. Authorization tokens must be opaque identifiers that do not include private information (Eq. 10).
5. Private Attributes provided to a Principal as part of an authorization token, or otherwise, must match the Principal's Trait (Eq. 12).
6. Any data returned to a Principal from a Transaction must originate from either that Principal's Private Attributes, from Private Attributes that match the trait of the Principal, or from non-private data (Eq. 13).

These constraints mirror the privacy principles exactly; the last two (5,6) are important because they provide a more detailed formulation of the second principle. The first of these (5) is a simple statement of principle 2 (only provide attributes to principals with appropriate trait); the second (6) proves to be subtler:

- The 'Transaction' is a record of information flow into service state, and so this places a requirement on service providers to know when state has originated from private data.
- A data item that originates from private data may be returned to the original owner of that private data, or to any Principal of the correct trait.

3. USING THE PRINCPLES

The foregoing demonstrates that comprehensive attribute protection requires unlinkable protocols, information flow trust in services, system-wide knowledge of service traits, and agreements about how attributes are grouped into traits. However, if attribute management principles are followed when business processes are decomposed into services, then some of these constraints (such as knowledge of the trait of a service) can be encoded in the design, rather than requiring operational mechanisms. Space precludes a full discussion of implementation issues, but a further example will illustrate this process.

A common concern in mobile computing is location privacy – how users are able to obtain services based on their location, while avoiding personal tracking. The attribute-management solution is to query services by providing the user's location, but no further information. The issue of unlinkable temporary pseudonyms has also been considered in this context [17] but researchers have not dealt with the problem that further personal attributes are needed to utilize services after they have been located.

Consider the case of a roaming user who wishes to print a document. The user requires a print service to locate a nearby printer and manage printing. The document resides on a workgroup server, to which the user must provide authentication information before access to the file is granted.

In this case it is straightforward to partition the user's personal attributes (location and workgroup information) into two separate traits, and assign one to the print system and the other to the file server. The primitive protocol elements used in the analysis (see 2.1) are sufficient to outline the process:

- The user presents a workgroup authentication attribute to the file server and obtains an opaque authorization token that confers access to the specified file.
- The user provides location co-ordinates to the print service, and obtains a reference to the nearest printer.
- The user presents the authorization token to the printer, which is able to retrieve the file and print it.

This outline description avoids the protocol details: how a user establishes temporary unlinkable pseudonyms with the services in order to carry out the transactions, suitably opaque forms of authorization tokens, and the use of an attribute authority to authenticate the workgroup attribute. However, it does demonstrate how the attribute management principles can be used to influence process design, and the importance focusing on the whole process chain, not just a single service interaction.

4. CONCLUSIONS

The extent that personal attributes can be linked determines the extent that private data can be profiled; this is of fundamental importance to privacy, and this viewpoint is consistent with the principles of purpose and consent contained in the Data Protection standards.

This paper has investigated the threat that users' private attributes may be directly aggregated into personal profiles, and shows that it is possible to avoid this problem if services are designed to meet *attribute management principles* (1.4). The principles group private attributes into *traits* and ensure that no service needs attributes from more than one trait.

The analysis of these principles shows that they can be applied in service-based systems that support the distribution of authentication and authorization tokens [18], and a worked example demonstrates their practical use in the decomposition of a business process into services.

The constraints derived in the analysis provide a single framework for system level privacy that motivates the need for established mechanisms, such as unlinkable protocols, as well as additional concerns arising from direct attribute aggregation.

The analysis and proof shows that if the management principles are observed then the desired property of unlinkability is upheld in the system as a whole. The relative robustness or fragility of different attribute distribution policies in the face of a collusion attack is still an open question, but the principles described here are believed to be robust, because they would force many services to collude before a user profile could be reconstructed.

REFERENCES

[1] *On the protection of individuals with regard to the processing of personal data and on the free movement of such data*, in *European Parliament and of the Council*. 1995.

[2] *The Data Protection Act 1988*, in *United Kingdom*. 1998.

[3] *Secretary's Advisory Committee on Automated Personal Data Systems, Records, Computers, and the Rights of Citizens viii*, U.S. Dep't. of Health, Education and Welfare, July 1973. http://www.epic.org/privacy/hew1973report/foreword.htm

[4] D. G. Johnson and K. W. Miller, The Ties that Bind: Connections, Comet Cursors, and Consent. *ACM SIGCAS Computers and Society*. **31**(1): p. 12 - 16 (2001)

[5] H. Chivers, J. A. Clark, and S. Stepney. Smart Devices and Software Agents: the Basic of Good Behaviour, in Proceedings of *The first International Conference on Security in Pervasive Computing*, Boppard, Germany. LNCS vol 2802. Springer-Verlag (2003)

[6] M. Erdos and S. Cantor, *Shibboleth Architecture*, Internet2, 8 October, 2001. http://middleware.internet2.edu/shibboleth/

[7] V. Benjumea, J. Lopez, J. A. Montenegro, and J. M. Troya. A First Approach to Provide

Anonymity in Attribute Certificates, in Proceedings of *Public Key Cryptography – PKC 2004: 7th International Workshop on Theory and Practice in Public Key Cryptography*, Singapore. LNCS vol 2047. Springer Verlag (2004)

[8] S. Holtmanns. Privacy in a Mobile Environment, in Proceedings of *13th International Workshop on Database and Expert Systems Applications (DEXA'02)*, Aix-en-Provence, France. IEEE Computer Society (2002)

[9] *Platform for Privacy Preferences (P3P) Project*, W3C, http://www.w3.org/P3P/

[10] H. M. Choldin, Government Statistics: The Conflict Between Research and Privacy. *Demography*. **25**(1): p. 145-154 (1988)

[11] K. A. Wallace, Anonymity. *Ethics and Information technology*. **1**(1): p. 21-31 (1998)

[12] D. Chaum, Security without identification: Transaction Systems to Make big brother obsolete. *Communications of the ACM*. **28**(10): p. 1030-1044 (1985)

[13] J. Camenisch, J. M. Piveteau, and M. Stadler. Fair blind signatures, in Proceedings of *Advances in Cryptology - EUROCRYPT '95, International Conference on the Theory and Application of Cryptographic Techniques*, Saint-Malo, France. Lecture Notes in Computer Science vol 921. Springer Verlag (1995)

[14] D. Chaum, Untraceable Electronic Mail, Return Address, and Digital Pseudonyms. *Communications of the ACM*. **24**(2): p. 1030-1044 (1981)

[15] T. Okamoto and K. Ohta. Universal Electronic Cash, in Proceedings of *Advances in Cryptology - CRYPTO '91*. LNCS vol 576. Springer Verlag (1991)

[16] J. M. Spivey, *The Z notation: A Reference Manual*. Prentice Hall International Series in Computer Science, ed. C.A.R.Hoare. 1989: Prentice Hall.

[17] A. R. Beresford and F. Stajano, Location Privacy in Pervasive Computing. *IEEE Pervasive Computing*. **2**(1): p. 46-455 (2003)

[18] *Security Services Use Cases and Requirements, Consensus draft 1*, Organization for the Advancement of Structured Information Standards (OASIS), draft specification 30 May 2001. http://www.oasis-open.org/committees/security/

APPENDIX A: PROOF OF THE PRIVACY PROPOSITION

Section 2.3 defines the privacy proposition as: *Any flow of information from a Private Attribute to a Principal is either from the Principal's own Private Attributes, or from a Private Attribute of a Trait that the Principal is allowed to process.*

$$\text{PRIVATE\_FLOW} = \text{PRIV} \cup \text{MATCH} \tag{17}$$

Proof. The method is to enumerate all possible paths from A to P in the graph, and show that this set of paths is a subset of PRIVATE_FLOW.

The set of all possible paths is the set of possible relational compositions of the edge set:

$$\text{PATHS} = \text{E} \circ \text{E} \circ \text{E} \circ \text{E} \circ \text{E} \ldots\ldots \tag{18}$$

The possible paths in this graph can be enumerated straightforwardly by inspection of their types, assuming the base classes in the model are disjoint:

$$\text{PATHS} = \text{PRIV} \cup (\text{BA} \circ \text{TA}) \cup (\text{BA} \circ \text{SA}) \tag{19}$$

The three sets whose union is PATHS are considered separately and each is shown to be a subset of PRIVATE_FLOW:

PRIV

PRIV occurs in (17) as a subset of PRIVATE_FLOW.

BA∘TA

Expanding the definition of composition, then further expanding BA:

$$\begin{aligned} &BA \circ TA = \{(a, p) \mid \exists t((a, t) \in BA \wedge (t, p) \in TA)\} \\ &= \{(a, p) \mid \exists t(\exists x[(t, x) \in TA \wedge (a, x) \in MATCH] \wedge (t, p) \in TA)\} \end{aligned} \tag{20}$$

Moving the quantifier for x out; then since (t,x) and (t,p) are in TA, and TA is a function we can conclude that x=p, eliminate x by substitution and remove one of the conjoined TA membership predicates:

$$= \{(a, p) \mid \exists t((t, p) \in TA \wedge (a, p) \in MATCH)\} \tag{21}$$

Moving quantifiers in and re-arranging:

$$= \{(a, p) \mid \exists t((t, p) \in TA) \wedge ((a, p) \in MATCH)\} \tag{22}$$

Since an element of this conjunction is MATCH, we can conclude that *BA∘TA* is a subset of MATCH and hence, from (17) a subset of PRIVATE_FLOW.

BA∘SA

Expanding the definition of composition, then further expanding SA:

$$\begin{aligned} &BA \circ SA = \{(a, p) \mid \exists t((a, t) \in BA \wedge (t, p) \in SA)\} \\ &= \{(a, p) \mid \exists t((a, t) \in BA \wedge [(a, t) \in BA \Rightarrow ((a, p) \in PRIV) \vee (a, p) \in MATCH)])\} \end{aligned} \tag{23}$$

Expanding the implication [...] and distributing the conjunction across the resulting expression:

$$\begin{aligned} &= \{(a, p) \mid \exists t([(a, t) \in BA \wedge \neg(a, t) \in BA] \vee \\ &[(a, t) \in BA \wedge (a, t) \in BA \wedge ((a, p) \in PRIV \vee (a, p) \in MATCH)])\} \end{aligned} \tag{24}$$

The left hand side of the disjunction can be eliminated (false), moving the quantifier in and eliminating one of the conjoined BA membership predicates:

$$= \{(a, p) \mid [(a, p) \in PRIV \vee (a, p) \in MATCH] \wedge \exists t((a, t) \in BA)\} \tag{25}$$

Since an element of this conjunction is (PRIV v MATCH), we can conclude from (17) that *BA∘SA* is a subset of PRIVATE_FLOW.

Conclusion

Each of the three sets *PRIV*, *BA∘TA* and *BA∘SA* are subsets of PRIVATE_FLOW; their union PATHS (19) is therefore also a subset of PRIVATE_FLOW. ***QED***