PRIVACY UNDER CONDITIONS OF CONCURRENT INTERACTION WITH MULTIPLE PARTIES

Martin S Olivier
Department of Computer Science, University of Pretoria, South Africa
http://mo.co.za

Abstract This paper explores the possibility to represent the privacy policies of an individual, as well as the processing steps of those who (concurrently) process the data, using a simple process algebra, FSP. The approach leads to the identification of two major classes of privacy policies: aggregation policies and quantitative policies. Automated analysis (with the LTSA tool) of such policies, in combination with the actions of parties that process personal information allows the automated discovery of possible breaches of privacy.

It is demonstrated that addressing the breaches often involves tradeoffs, such discontinuing interaction with some parties, so that policies are no longer violated.

1. Introduction

Personal privacy has, in some respects, been an evasive topic in Computer Science. While it has become entrenched in. phrases such as *Security and Privacy,* it is often difficult to establish what *privacy* adds to the notion that is not already covered under the rubric of *security.* The one clear exception is the association between *privacy* and *anonymity:* If privacy is a form of security for the individual, then anonymity guarantees privacy. Anonymity, however, is not a panacea: Not only are anonymity and accountability incompatible, but a tradeoff between anonymity and privacy implies a tradeoff between privacy and freedom — freedom of speech [23] and freedom to act; the degree to which actions and utterings can be observed, stored and matched in a digital environment would constrain such actions to a significant extent if one's only option to perform them with a reasonable amount of privacy were to perform them anonymously.

Having said this, it is important to realise that sharing of personal information with others always carries some risk. There are very few guarantees that

personal information, once shared, will be treated in confidence and used in the manner in which it was intended. This risk is not without benefit, however: society depends on information about individuals being available. Society's need for accountability justifies individuals carrying identity cards and passports. This need justifies the requirement that cars are fitted with licence plates. This need also justifies (responsible) use of blacklists for services where individuals moved outside society's reasonable expectations of how individuals should behave under certain circumstances.

If total anonymity (for total privacy) is not the only option, the question becomes one of how much privacy can be achieved by how much reserve? Unfortunately, privacy and reserve (with many other such notions) are notoriously difficult to quantify. It seems that the best approximation to this is some (formal) model that enables one to reason about specific tradeoffs.

P3P [29] provides a first step in this direction. P3P allows both the user and the service provider to represent (aspects of) their privacy policy. These policies are then compared and, only if the policies are compatible, is the individual's information shared. No attempt, however, is made to correlate information shared with a number of parties. Arguably, when multiple parties are involved, privacy risks for the combined case are significantly higher than the sum of the individual risks. This is amongst others due to the possibility of correlating ('matching') and aggregation of information by the various parties.

This paper explores the possibility to achieve some indication of the tradeoffs that occur when an individual interacts with multiple parties and wishes to maintain privacy according to some personal policy. The activities of parties with whom an individual deals are formally modelled. Next, the activities that have privacy implications are highlighted using the same formal modelling technique. Finally, the individual's privacy policy is represented using the same technique. In this privacy policy undesirable states are indicated. Reachability analysis is then used to check the possibility of reaching such undesirable privacy states. The individual can now decide how to react to such a possible breach of privacy.

A simple process algebra, FSP [22], is used to model the activities of those parties with whom the individual interacts and the individual's privacy policies. The LTSA tool [22] is used to perform the reachability analysis.

The approach followed in this paper still offers many challenges, not the least of which is the complexity of appropriately modelling something as intangible as privacy. This complexity will specifically impact on the possibility of practical application of the proposed approach. This paper has a more modest goal than attempting to give a comprehensive solution to the stated problem; its goal is rather to explore the notions involved in addressing the problem and thereby help set the stage for a solution (if a practical solution does exist).

The paper is structured as follows. Section 2 gives some background about privacy. Section 3 introduces modelling the actions of parties with whom the individual deals as well as the individual's privacy policy. Next, section 4 uses reachability analysis to identify privacy policies and considers courses of action the individual may pursue to address such problems. Section 5 ties up some loose ends after which section 6 concludes the paper.

2. Background

Privacy is a complex issue. This is already evident if one only focusses on technical issues — the main concern of this paper. It seems that technical work has been done in five areas regarding privacy, namely anonymity, private communication, inference control, organisational safeguards and personal control. Each of these is briefly considered below. For an introduction to non-technical aspects of privacy see [4, 8, 9, 18, 23, 27, 31,40].

Anonymity work in privacy is based on the fact that observation of one's actions in Cyberspace is easier, and records tend to persist for longer. Work done to achieve anonymity (or pseudonymity) include onion routing [13] and Crowds [30] — both of which hide the origin of messages flowing through a network — and anonymous proxies (such as web anonymizers and anonymous remailers [5,11,21]) — that hide the IP-address and other identifying details of the client from the server. In this regard, it is worth noting Chadwick's point [6] that anonymity is not only a bad idea on the Internet (due to its incompatibility with accountability), but also that most of these solutions actually provide a pseudonymous solution.

The second area — privacy of personal communications — has primarily been addressed by encryption. This is evidenced by, for example, the name of the popular encryption product, PGP: Pretty Good *Privacy* [12].

The third aspect of (technical) privacy that has received some attention is that of statistical inference. This has been achieved by modifying data in ways that are statistically insignificant [35] and by limiting statistical queries over such data to queries that include a sufficient number of entries so that inferences cannot be made about individual cases [7] and removing identifying information from such entries [33, 36]. In the last case, it is interesting to note that removing obviously identifying information (such as names, identity numbers, etc) is not sufficient and a more extensive sanitising of data is required [33, 36].

In the case of organisational safeguards, most work done on security is relevant, but not specific to privacy. Specific privacy safeguards will focus on protection of the individual's information against inadvertent (or even intentional) misuse by the organisation or those associated with the organisation. It may be approached by implementing checks and balances to ensure that personal information is not misused. This can, amongst others, be ensured by recording

the purpose for which personal data has been collected, recording the privacy policy applicable at the time of data collection, recording personal preferences, and other similar privacy-related information. Additionally, an access monitor then controls all accesses to such data by ensuring that access attempts are compatible with all the security information recorded earlier, as well as other privacy considerations at the time of collection and/or at the time of use. A number of solutions to achieve this have been proposed [3, 10, 19, 1, 2, 20, 26]. A common reservation about such solutions is the doubt that businesses will voluntarily implement them. The business case for such solutions have been argued in a number of places [2, 14, 15, 16, 17, 20, 23, 28, 39]. The essence of these arguments is that these technologies make business sense because they will help to attract customers and to build a relationship of trust with new and existing customers, yielding a (mutually) beneficial associasion. Such technologies can also help to prevent costly privacy-related accidents. Products that have been introduced in this area include IBM's Tivoli Privacy Manager [38] and PrivacyRight's TrustFilter [28]. Note should also be taken of IBM's Enterprise Privacy Architecture [15, 19].

The importance of organisational safeguards for the current paper is the fact that they provide a degree of assurance that organisations will indeed honour their privacy commitments as expressed in their privacy policies. When such technologies are properly installed and regularly audited, personal control becomes more dependable.

The final privacy aspect to be addressed here is indeed that of personal control. The best-known example here is P3P [29] where the individual controls whether information should be divulged to an organisation, given the organisation's privacy policy. Also in this category is the work done by Teepe et al [37] that establishes what information an individual will need to divulge during the course of a workflow process before the individual initiates the workflow. This enables the individual to know whether participation in a workflow will become too sensitive, before divulging some private information as part of a process. The current paper also addresses this aspect (personal control) of the privacy problem.

The relationships between the technologies considered above has been investigated elsewhere [24], where it has been shown that a fully ordered relationship exists between them, that was then formalised as the Layered Privacy Architecture (LaPA).

The use of a process algebra to model aspects of security is not new [32]. Also privacy (specifically anonymity) has been modelled using a process algebra [32, 34]. In these cases, the goal is to model security protocols and determine whether specific security properties (secrecy, authentication, non-repudiation and anonymity) hold for these protocols as required. In our case it is not security protocols that will be modelled, but business processes. We will also not be

primarily concerned about whether specific properties hold, but whether an arbitrary (user-specifiable) privacy policy can be violated.

3. Modelling actions and privacy

3.1. Behaviour of parties

If the actions of some party with whom an individual wants to interact are known, it is relatively easy to model the actions of that party in FSP. To illustrate, the actions of a shop may be

```
SHOP = (address.request -> pack -> ship -> address.release ->
   SHOP).
```

Here the `pack` and `ship` actions are assumed to be primitives that have no real impact on the privacy of the individual. In contrast `address.request` and `address.release` are actions that were chosen to indicate that the shop acquires the customer's address somehow and deletes it afterwards.

Even the simple definition of a `SHOP` holds a number of challenges. Firstly, the question arises about who should define the actions of the `SHOP`. The end-user may not have access to the internal operation of the `SHOP` and may therefore not be able to model the `SHOP` accurately. A better solution would therefore be for the `SHOP` to model its own actions and publish that as part of its privacy policy. This, however, would require a standardised nomenclature (or ontology) — a problem that we ignore given the explorative nature of this paper. A second problem introduced by the `SHOP` publishing its own actions as part of a privacy policy would be knowing which of its actions are relevant for a diverse population of individuals, each with his or her own privacy policy. It is possible to address this by expecting the `SHOP` to publish a relatively detailed account of its actions, and where the individual then selects only those that are appropriate to his or her privacy policy. The following FSP statement expresses that only `address.request` and `address.release` are relevant to the individual:

```
SHOP = (address.request -> pack -> ship -> address.release ->
   SHOP)
   @{address.{request,release}}.
```

It is also possible to model temporal retention of data for audit purposes, as follows

```
SHOP = (address.request -> pack -> ship -> waitFiveYears ->
   address.release -> SHOP).
```

However, this paper does not consider such temporal constraints. What the paper does consider are cases where information is retained until some action leads to its deletion. This is illustrated by the following definition of `INSURER`:

```
INSURER = (insurancedata.request -> INSURED),
```

```
INSURED = (claim -> process -> payout -> INSURED |
   cancel -> insurancedata.release -> INSURER).
```

Here the insurance data is retained until the policy is cancelled by an explicit `cancel` action.

For the sake of illustration below, one further class of organisations is introduced here:

```
MARKETER = (address.request -> sendmail -> address.release ->
   MARKETER).
```

Finally, the instances of these categories of organisations are defined and specified as processes that execute concurrently. For our purposes `a` and `d` are `INSURER`s, `b` is a `SHOP` and `c` is a `MARKETER`:

```
||ORGANISATIONS = (a:INSURER || b:SHOP || c:MARKETER ||
   d:INSURER).
```

3.2. Categories of parties

Now that the behaviour of parties has been specified, and some instances of those parties have been defined, it becomes possible to identify categories of organisations so that it will be possible to specify privacy policies in terms of such categories, rather than specific parties (although specification of privacy policies in terms of such specific parties will not be precluded).

The first category is one that inherently holds a negative connotation:

```
SPAMMER = (address.request -> spam -> SPAMMER).
```

The intention here is that, whenever a party identified as a `SPAMMER`, performs an `address.request` action, it becomes possible for the `SPAMMER` process to perform the `spam` action, that will be used below in the policies section to specify a privacy policy to limit the actions of spammers.

The second category to be used in this paper is not inherently value laden; it will be used to identify financial institutions and specifically identify the points at which they request and release information. `FINANCIAL` accomplishes this:

```
FINANCIAL = ({address,insurancedata}.request -> finreq ->
   {address,insurancedata}.release -> finrel -> FINANCIAL).
```

This definition incorporates a rudimentary form of categorising data: `address` is assumed to be part of `insurancedata`, and when either is requested, the `finreq` *(financial request)* action becomes enabled. Similarly, when either is released, the `finrel` *(financial release)* action becomes enabled.

Unfortunately, given the nature of the process algebra used, these trigger actions (`spam`, `finreq` and `finrel`) will automatically be enabled for those parties who are not placed in the respective categories. (For example, if `a` is not identified as a `SPAMMER` in this section, `a.spam` will automatically be enabled

in the policies section below, because there is no `a.spam` in the category section here (yet) for which it will have to 'wait'.) In order to avoid this, it is necessary to introduce 'opposites' for the categories already introduced:

```
NOTSPAMMER = NOTSPAMMER +{spam}.
NOTFINANCIAL = NOTFINANCIAL +{finreq, finrel}.
```

In both of these cases the trigger actions do not form part of the process descriptions but the alphabets of the processes are extended with these actions; they therefore inherently cannot occur in these processes and will therefore not be enabled in the policies section below.

All that remains in this section is to combine the various instances of parties with their respective categories. A possible combination looks as follows:

```
||CATEGORIES =
   ({c}:SPAMMER || {a,b,d}:NOTSPAMMER ||
   {a,d,b}:FINANCIAL || {c}:NOTFINANCIAL)
```

3.3. Privacy policies

Once the parties and categories have been modelled, the policies can be modelled.

In some cases (possible) occurrence of the trigger action will violate privacy and therefore needs to be indicated as such:

```
property CHECKSPAM = (spam -> ERROR).
```

Another possible policy is to limit the number of occurrences of one's information in different databases. The following policy limits one's information to two financial institutions:

```
const MaxFin = 2
property CHECKFIN = CHECKFIN[0],
CHECKFIN[i:0..MaxFin] = (
   when (i<MaxFin) finreq -> CHECKFIN[i+1] |
   when (i==MaxFin) finreq -> ERROR |
   when (i>0) finrel -> CHECKFIN[i-1] |
   when (i==0) finrel -> ERROR).
```

These examples suffice for the initial discussion and now it is possible to combine the various policies:

```
set AllProc = {a,b,c,d}
||POLICIES = (AllProc::CHECKSPAM || AllProc::CHECKFIN).
```

Finally the parties, categories and policies can be combined:

```
||SYSTEM = (ORGANISATIONS || CATEGORIES || POLICIES).
```

4. Discussion

A reachability analysis of the system described above can now be performed to see if the ERROR state can be reached. If it can, a trace to this state will illustrate how it is possible to violate the individual's privacy policy. As is clear from the simple examples given above, it is indeed possible to violate the policy. A safety analysis with LTSA produces the following trace:

```
Depth 2 -- States: 4 Transitions: 27 Memory used: 7356K
Trace to property violation in POLICIES.{a,b,c,d}::CHECKSPAM:
        c.address.request
        c.spam
```

This indicates the (obvious) problem that the individual's CHECKSPAM policy can be violated if c requests the individual's address leading to the possible execution of the trigger action spam by c.

This simple example illustrates one of the major points of this paper: Once a possible breach of privacy has been identified, it is, in general, up to the individual to deal with the possible breach. The individual may, for example, avoid the breach by no longer dealing with c. The individual may also investigate the reasons why he or she has identified c as a spammer. If it possible to accept the amount of junk mail c sends, it may warrant reclassifying c as a NOTSPAMMER. To further elaborate on this last point, suppose that b is classified as a SPAMMER, but in our agreed upon interaction (as specified by the FSP definition of b's behaviour) b does not send any mail. If we accept the specified descriptions as true of actual behaviour (a point that we already noted earlier in the paper), the individual may decide to phrase the SPAMMER category in terms of the mail action rather than (or in addition to) the address.request action, as follows:

```
SPAMMER = (mail -> spam -> SPAMMER).
```

or

```
SPAMMER = (address.request -> mail -> spam -> SPAMMER).
```

Given such a trigger action definition, b would not violate the policy even if b were identified as a SPAMMER. In other words, thinking about the semantics of actions may lead to a redefinition of a trigger action or a privacy policy that would avoid the conflict. These options are in addition to the options of no longer dealing with the party involved in the breach of a privacy policy, or renegotiating the interaction with that party.

Once the SPAMMER violation has been addressed in any of the above manners, a reachability analysis with LTSA leads to the following trace:

```
Depth 6 -- States: 570 Transitions: 3756 Memory used: 8463K
Trace to property violation in POLICIES.{a,b,c,d}::CHECKFIN:
        a.insurancedata.request
        b.address.request
```

```
d.insurancedata.request
a.finreq
d.finreq
b.finreq
```

What this shows is the (again obvious) result that the maximum number of financial institutions that are allowed to simultaneously hold the individual's information can be exceeded (as illustrated here, when `a`, `b` and `d` simultaneously hold some of the individual's information).

Resolving the violation may again be considered by stopping interaction with any of the parties involved (`a`, `b` or `d`), changing the categorisation (for example deciding that `b` is not a financial institution after all) or changing the policy (by, for example, setting MaxFin equal to three). For example, changing classification of `b` to `NOTFINANCIAL` yields the following safety results from LTSA:

```
Depth 25 -- States: 16384 Transitions: 93184 Memory used: 8082K
No deadlocks/errors
```

The question that needs to be addressed now, is one on the nature of privacy violations that can be modelled (and detected) using the approach described in this paper. While it is clearly possible to check privacy violations from a single party (such as that represented by the `CHECKSPAM` policy above) the clear power of the technique lies in detecting (and addressing) privacy violations that arise from the interaction of many parties that are concurrently active and processing one's data. Before turning our attention to that, it is worth noting that single-party violations need not be as simple as the `CHECKSPAM` policy above: it is also possible to have a policy that monitor's the party's acquisition of personal information over time and flags the point at which the amount of collected information exceeds some threshold. The details of how this can be done will be clear after the multi-party scenario has been discussed below.

One type of privacy policy for which the described approach holds potential, is that of computerised matching. Suppose that some party `m` has access to the individual's medical records, while another, `t`, has access to the individual's tax records. If the individual is concerned that `m` and `t` might use their access to match the data to learn more about the individual, the individual can set up a privacy policy that, while permitting the two parties to access his or her information, does not allow them to do it simultaneously. To do this, suppose that appropriate categories have been created to cause suitable `medicalreq`, `medicalrel`, `taxreq` and `taxrel` trigger actions. Then the following policy will specify the non-matching requirement:

```
property CHECKMATCH = (taxreq -> HASTAX | medicalreq ->
   HASMED),
HASTAX = (taxrel -> CHECKMATCH | medicalreq -> ERROR),
HASMED = (medicalrel -> CHECKMATCH | taxreq -> ERROR).
```

with POLICIES now defined as

```
||POLICIES = (AllProc::CHECKSPAM || AllProc::CHECKFIN ||
   t,m::CHECKMATCH).
```

Prevention of computerised matching is clearly a special case of what may be termed an *aggregation* policy — where the individual wants to prevent simultaneous processing of a set of personal data items that, when aggregated, may form a more comprehensive image of the individual (whether it is true or false) than the individual data items would have. Definition 1 formalises this notion.

Definition 1 (Aggregation policies) *Let $\mathbb{A}$ be the set of all personal attributes of an individual and $\mathbb{P}$ the set of all parties with whom the individual interacts. Let $A \subseteq \mathbb{A}$ be a set of personal attributes about some individual. Let $P \subseteq \mathbb{P}$ be a set of those parties that occasionally process the attributes in A. Let $proc_i(p) \subseteq A$, with $p \in P$ indicate the set of attributes being processed by some party p at some instant i. An aggregation policy for A over P specifies that no instant i exists for which*

$$\bigcup_{p \in P} proc_i(p) = A$$

Note that the case where $|P| = 1$ is the special case alluded to above when single-party policies were briefly considered.

While the class of aggregation policies can indeed include a range of specific policies, it does not include the case that was used to introduce the modelling of properties above, where some financial information was limited to a specific number of instances or occurrences amongst a set of financial institutions. That case is formally defined in definition 2

Definition 2 (Quantitative policies) *Let $\mathbb{A}$ again be the set of all personal attributes of an individual and $\mathbb{P}$ the set of all parties with whom the individual interacts. Let $A \subseteq \mathbb{A}$ be some set of personal attributes and $P \subseteq \mathbb{P}$ some set of parties with whom the individual interacts. Let $n \geq 0$ be an integer. A quantitative policy for A over P specifies that*

$$|\{p \in P | proc_i(p) \cap A \neq \emptyset\}| \leq n$$

where $proc_i$ has the same meaning it had in definition 1.

Again note that the 'spamming policy' used above is a special case of a quantitative policy for which $n = 0$. When $n = 1$ the privacy policy for a document implies that the document should be treated like a paper-based document: it could be at different parties, but not simultaneously. For $n > 1$ the choice of n in general seems to be arbitrary, with a lower value of n simply specifying a higher degree of privacy (with its concomitant costs).

5. Tying up loose ends

One of the fundamental assumptions of this paper is that is is possible to specify the behaviour of parties with whom one interacts and then 'know' that they will stick to that behaviour. The notion that it is possible to specify real-world behaviour in a compact manner seems plausible but needs to be tested with real-world examples. This is, however, left to future research.

The requirement that parties will stick to a given behaviour, once that behaviour has been formalised in a policy, is also a complex issue, with psychological, social, ethical, technical and (possibly) other dimensions. This paper accepts that policies are routinely used to specify behaviour and that mechanisms exist in society to audit adherence to a policy and legal, societal and other sanctions can be used guarantee a reasonable degree of compliance. However, apart from the remarks made earlier about organisational safeguards, it is outside the scope of the current work to consider compliance with policies in practice.

A more pressing issue for the current work is the nature of interaction specified in the policies. Information was typically requested by some action (for example, `address.request`) and released afterwards (for example, `address.release`). The assumption was that data was to be provided when the request occurs and deleted later when the release occurs. This, however, does not accurately reflect practices in the real world. Since a customer database is an asset owned by a company, it is unlikely to destroy parts of it the moment it is no longer required for purposes of the transaction conducted. With the emergence of data mining the value of such data increases and the chances of voluntary deletion decreases. One possible route to explore is social pressure to enforce deletion, which is a viable option if society is indeed better served by such a practice than by permanent archiving — an aspect that is not explored in the current paper.

A more viable approach seems to be the following: If parties who process personal information are willing to change their behaviour such that they send out a request to process the information about an individual that they already have on record, it becomes possible to specify privacy policies in terms of not only data acquisition and release, but also in terms of processing. This would require the parties responsible to specify such points in their behaviour where they start and stop processing data; in the case of a `claim` in our earlier example, behaviour might be specified as follows:

```
INSURED = (claim -> insurancedata.beginprocessing -> process ->
   payout -> insurancedata.endprocessing -> INSURED |
   cancel -> insurancedata.release -> INSURER).
```

While this approach is again against the spirit of the current paper, in that it expects a behaviour change from parties who process information rather than only focussing on the individual's ability to control his or her private infor-

mation, this behaviour change clearly enhances the individual's possibilities of taking control: If this is implemented, the individual will be able to specify privacy policies much more precisely; computerised matching, for example, can be controlled even if a party holds two pieces of information that should not be matched — simultaneous possession is less of a concern if simultaneous processing can be excluded. While consideration of such policies is well within the scope of the current paper, a lack of space precludes a detailed discussion here.

It is also worth noting that expecting a party to indicate its intention to begin processing a personal data attribute (and possibly to be blocked at that point by a privacy policy) need not be a practical major concern: It is possible to establish trusted third parties that store individuals' privacy policies and grant or deny semaphores based on the appropriate policy. (See the version of this paper published in the preproceedings for more information on the use of such semaphores in this context.) Again, availability and fault-tolerance issues for such third parties will not be considered in the current paper.

6. Conclusion

This paper considered the possibility of representing personal privacy policies using a process algebra. Simple examples were used to illustrate two major classes of privacy policies: aggregation policies and quantitative policies. Automated analysis of the examples were used to identify the tradeoffs an individual has to make when enforcing such privacy policies.

A number of open problems were introduced in the paper and will not be repeated here. One interesting implication of the approach has not yet been considered: If many individual privacy policies are modelled and publicly available, a party that processes private data can begin to measure the effects of its own processes in the light of known privacy constraints; it potentially becomes possible to measure how a certain action will constrain or enhance business processes. This, however, also needs to be left for future research.

References

[1] S. B. Adler, E. F. Bangerter, K. A. Bohrer, J. Brown, N. Howard, J. Camenisch, A. M. Gilbert, D. Kesdogan, M. P. Leonard, X. Liu, M. R. McCullough, A. C. Nelson, C. C. Palmer, C. S. Powers, M. Schnyder, E. Schonberg, M. Schunter, E. van Herreweghen, and M. Waidner. Using an object model to improve handling of personally identifiable information. United States Patent Application 20030004734, January 2003.

[2] R. Agrawal, J. Kiernan, R. Srikant, and Y. Xu. Hippocratic databases. In *28th Int'l Conf. on Very Large Databases (VLDB). Hong Kong,* 2002.

[3] P. Ashley, S. Hada, G. Karjoth, and M. Schunter. E-P3P privacy policies and privacy authorization. In *Proceedings of the ACM workshop on Privacy in the Electronic Society,* pages 103–109. ACM Press, 2003.

[4] D. Brin. *The Transparent Society — Will Technology Force us to Choose between Privacy and Freedom?* Perseus Books, Reading, MA, 1998.

[5] M. A. Caloyannides. Encryption wars: Shifting tactics. *IEEE Spectrum,* 37(5):46–51, 2000.

[6] D. Chadwick, M. S. Olivier, P. Samarati, E. Sharpston, and B. Thuraishingham. Privacy and civil liberties. In E. Gudes and S. Shenoi, editors, *Research Directions in Database and Application Security,* pages 331–346. Kluwer, 2003.

[7] G. T. Duncan and S. Mukherjee. Microdata disclosure limitation in statistical databases: Query size and random sample query control. In *IEEE Computer Society Symposium on Research in Security and Privacy,* pages 278–287, Oakland, CA, USA, May 1991.

[8] A. Etzioni. *The Limits of Privacy.* Basic Books, New York, NY, 1999.

[9] A. Etzioni. Medical records — enhancing privacy, preserving the common good. *Hastings Center Report,* 23(2): 14–23, 1999.

[10] S. Fischer-Hübner and A. Ott. From a formal privacy model to its implementation. In *21st National Information Systems Security Conference,* Arlington, VA, USA, October 1998.

[11] E. Gabber, P. B. Gibbons, D. M. Kristol, Y. Matias, and A. Mayer. Consistent, yet anonymous, web access with LPWA. *Communications of the ACM,* 42(2):42–47, February 1999.

[12] S. Garfinkel. *PGP: Pretty Good Privacy.* O'Reilly, 1995.

[13] D. M. Goldschlag, M. G. Reed, and P. F. Syverson. Onion routing. *Communications of the ACM,* 42(2):39–41, February 1999.

[14] S. Hunt. Market overview: Privacy management technologies. Giga Information Group, February 2003.

[15] IBM. Enterprise privacy architecture: Securing returns on e-business. Executive brief, IBM, 2001.

[16] IBM. Privacy in a connected world. White paper, IBM, May 2002.

[17] IDcide. IDcide introduces corporate privacy compliance software. Press release, February 2001. `http://www.idcide.com/pages/press_releas.htm#6`.

[18] D. G. Johnson. *Computer Ethics.* Prentice Hall, third edition, 2001.

[19] G. Karjoth, M. Schunter, and M. Waidner. Privacy-enabled services for enterprises. Research Report RZ 3391 (# 93437), IBM Research, 2002.

[20] G. Karjoth, M. Schunter, and M. Waidner. Platform for Enterprise Privacy Practices: Privacy-enabled management of customer data. In R. Dingledine and P. Syverson, editors, *Privacy Enhancing Technologies: Second International Workshop, PET 2002, San Francisco, CA, USA, April 14-15, 2002, Revised Papers.* Springer, 2003.

[21] G. Lawton. Is technology meeting the privacy challenge? *IEEE Computer,* 34(9):16–18, 2001.

[22] J. Magee and J. Kramer. *Concurrency — State Models & Java Programs.* Wiley, 1999.

[23] M. S. Olivier. Database privacy. *SIGKDD Explorations,* 4(2):20–27, 2003.

[24] M. S. Olivier. A layered architecture for privacy-enhancing technologies. In J. H. P. Eloff, H. S. Venter, L. Labuschagne, and M. M. Eloff, editors, *Proceedings of the Third Annual Information Security South Africa Conference (ISSA2003),* pages 113–126, Sandton, South Africa, July 2003.

[25] M. S. Olivier. Privacy under conditions of concurrent interaction with multiple parties. In S. de Capitani di Vimercati, I. Ray, and I. Ray, editors, *Proceedings of the Seventeenth*

Anual IFIP WG11.3 Working Conference on Database and Application Security, pages 103–117, Estes Park, Colorado, USA, August 2003 (Preproceedings).

[26] M. S. Olivier. Using organisational safeguards to make justifiable decisions when processing personal data. In J. H. P. Eloff, P. Kotzé, A. P. Engelbrecht, and M. M. Eloff, editors, *IT Research in Developing Countries (SAICSIT 2003),* pages 275–284, Sandton, South Africa, September 2003.

[27] E. F. Paul, F. D. Miller, and J. Paul, editors. *The Right to Privacy.* Cambridge University Press, Cambridge, 2000.

[28] PrivacyRight. Control of personal information — the economic benefits of adopting an enterprise-wide permissions management platform. White Paper, 2001. `http://www.privacyright.com/info/economic.html`.

[29] J. Reagle and L. F. Cranor. The platform for privacy preferences. *Communications of the ACM,* 42(2):48–55, February 1999.

[30] M. K. Reiter and A. D. Rubin. Anonymous web transactions with Crowds. *Communications of the ACM,* 42(2):32–48, February 1999.

[31] A. Rosenberg. Privacy as a matter of taste and right. In E. F. Paul, F. D. Miller, and J. Paul, editors, *The Right to Privacy,* pages 68–90, Cambridge, 2000. Cambridge University Press.

[32] P. Ryan, S. Schneider, M. Goldsmith, G. Lowe, B. Roscoe, and G. Lower. *Modelling and Analysis of Security Protocols.* Addison Wesley, 2000.

[33] P. Samarati. Protecting respondents' identities in microdata release. *IEEE Transactions on Knowledge and Data Engineering,* 13(6): 1010–1027, 2001.

[34] S. Schneider and A. Sidiropoulos. CSP and anonymity. In *Proceedings of European Symposium on Research in Computer Security,* number 1146 in Lecture Notes in Computer Science, pages 198–218. Springer-Verlag, 1996.

[35] D. Stamate, H. Luchian, and B. Paechter. A general model for the answer-perturbation techniques. In *Seventh International Working Conference onScientific and Statistical Database Management,* pages 90–96, Charlottesville, VA, USA, Sep 1994. IEEE.

[36] L. Sweeney. Datafly: A system for providing anonymity in medical data. In T. Y. Lin and S. Qian, editors, *Database Security XI: Status and Prospects,* pages 356–381. Chapman & Hall, 1998.

[37] W. Teepe, R. P. van de Riet, and M. S. Olivier. Workflow analyzed for security and privacy in using databases. In B. Thuraisingham, R. P. van de Riet, K. R. Dittrich, and Z. Tari, editors, *Data and Applications Security — Developments and Directions,* pages 271–282. Kluwer, 2001.

[38] Tivoli Software. Enable your applications for privacy with IBM Tivoli Privacy Manager for e-business. Technical discussion, IBM, July 2002.

[39] Tivoli Software. IBM Tivoli Privacy Manager for e-business. Commercial brochure, IBM, 2002.

[40] R. Whitaker. *The End of Privacy — How Total Surveillance is Becoming a Reality.* New Press, New York, NY, 1999.