Using properties of physical phenomena and information flow control to manage privacy

David Evans and David M. Eyers
Computer Laboratory
University of Cambridge
Cambridge, UK
Email: {firstname.lastname}@cl.cam.ac.uk

***Abstract*—Information Flow Control allows mandatory access restrictions to be tightly coupled with data, usually in order to enforce confidentiality and/or to track data provenance. We are interested in the interplay between physical phenomena and their detection by sensors and the use of resulting data by distributed applications that may use the Internet as infrastructure. Within this context, IFC is the perfect tool to translate the privacy properties of the physical phenomena into control of the software system and allow formal reasoning about the system's behaviour.**

I. Introduction

Contemporary distributed applications, including those that use the Internet as infrastructure, transport personal data. These data are captured from the world, stored, processed, displayed, and used to compute other data. It is accepted by privacy advocates that greater controls on this process are needed; it is up to the technical community to provide tools that make this easier. In particular, designers of infrastructure must provide mechanisms so that "privacy by design" by those that build distributed applications is much more easy and natural to achieve than it is today. It is instructive to look at the mechanisms supporting type safety in programming languages and distributed middleware. It has evolved to the point that designers can usually have it if they want it, and its presence can allow drawing strong conclusions through formal analysis about system behaviour.

Information Flow Control (IFC) [1] is a technique for attaching attributes to data such that they cannot be modified or removed without authorisation. Common examples include recording who may see data and how the data were generated. These assertions of use and provenance are rolled into "labels" that travel with the data as it moves through the system. Access to data is via IFC-aware infrastructure that ensures only appropriately anointed software components access data and that the provenance portion of the label reflects any modifications that are made to it. Often the infrastructure in question is implemented at a language level, such as in Jif [2], or at an operating system level [3]. Making effective use of IFC mechanisms in distributed sensor networks requires definition of a labelling regime, understanding how data sources (particularly low-power sensors) can do labelling efficiently, and having an effective means for performing formal reasoning about the resulting system.

Here we concentrate on systems (or the subset of any given system) where data that are private come from the sensing of a physical phenomenon. We are concerned with things such as detecting the movement of individuals, recording car number-plates, and noting the time, place, and participants of a mobile phone call. We are not addressing users placing photographs of themselves or others online or them entering personal information into web forms. We do not intend to diminish the importance of regulating such applications—none of what we suggest impedes doing a good job of this—but have narrowed our scope because of our interest, as explored in our previous work, in the boundary between physical phenomena and subsequent data processing [4].

II. Summary of IFC

Information Flow Control uses data labelling to enforce where data may go, and thus can guarantee strong protection of data confidentiality and integrity [1]. In IFC, all data are tagged with *security labels*. Labels usually consist of a set of *confidentiality* tags, describing the "secrecy" of the data, and a set of *integrity* tags that attest to the data's provenance. Data can only flow to components with compatible labels and data released by a component must be compatible with the component's label. Special *declassification* and *endorsement* privileges are used to allow information to decrease in confidentiality and increase in integrity as it is processed. For example, information labelled "secret" can be handled by components with "top-secret" clearance but not vice versa. "Top-secret" data is therefore confined but for the use of declassification privileges. A static set of labels is unwieldy, so *decentralised information flow control* [2] permits creation of tags on the fly and allows privileges over these tags to be assigned and shared dynamically.

III. The strategy

A simple abstraction of the systems of interest to us is shown in figure 1. Sensors are the source of data; these data are created by the sensors responding to physical phenomena and are then passed to the rest of the system. Crucially, each phenomenon carries with it privacy implications. For example, a sensor detecting a temperature change might not lead to private data, but the detection of a vehicle moving past a certain position might. The privacy properties of

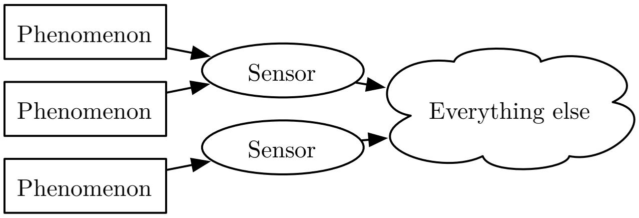

Figure 1. A conceptual view of distributed systems that capture personal data and process, store, and disseminate it.

this reading first of all depend on what is being sensed. One can imagine a reading able to determine increasing detail about a vehicle, perhaps spanning a spectrum from the vehicle's size, to its colour, to its make and model, and thence to the number of occupants and its registration number. Furthermore, the privacy properties may depend on the value sensed: it may not be very revealing to note that yet another grey car has passed compared to seeing a car that is electric purple.

The data that a sensor and its attendant software creates from each phenomenon should reflect the phenomenon's privacy properties. Within the context of IFC, this should be reflected in the data's integrity tags. The rest of the system can use these tags' encoding of the data's privacy properties to decide whether the data should receive special treatment (such as "data that identify an individual will not be stored") or to draw formal conclusions about the data handled by certain software components ("there has been evidence that this subsystem was sent data that identify an individual"). In this way the physical phenomena can be thought of as tainting the sensors and software, resulting in the data that they produce bearing tags that reflect the phenomena's privacy properties.

We have not said much about how confidentiality tags should be set. First and foremost, they should reflect the sensor owner's data flow policies (elsewhere we have examined how this can be done [5]). However, these policies should include, for example, statutory privacy responsibilities. This means that the confidentiality tags will, to a certain extent, mirror the integrity tags' use, limiting access to software modules that are authorised to deal with data having particular privacy properties.

IV. Research questions

What we have suggested is merely a sketch and building a useful large-scale implementation would require answering some important research questions. These include:

1) Do we need to enumerate the privacy properties of phenomena corresponding to each sensor and, if so, is the overall problem tractable? In other words, given that we want to assign integrity tags to the data resulting from the phenomena, do we need to know in advance every possible thing those tags could represent? Decentralised IFC means that we can manage a dynamic set of tags, but it is not clear that the necessary remapping can be done on-demand.
2) Can a sensor efficiently determine the privacy properties of a phenomenon? These must be specified sufficiently well that correct tags can be assigned and, further, it has to be done rapidly on devices that may have meagre resources.
3) We have said that confidentiality tags, because they express an organisation's policy at a level that is perhaps legal, are analogues of but not identical to integrity tags. It is not clear how to map from the mechanically-applied integrity tags, which represent objective measures of privacy by the sensor machinery. The confidentiality tags might require a more subjective notion of policy that groups data into defined security classes on the basis of numerous aspects of the actual sensor reading.
4) How should the IFC infrastructure alter integrity tags in response to data processing? At the most basic level, integrity tags will be destroyed when data are modified, but for many classes of sensor processing functionality (such as those implementing differential privacy-aware operations [6]) a formal framework might be able to add new, appropriately cross-referenced integrity tags automatically.

References

[1] D. E. Bell and L. J. La Padula, "Secure computer systems: Mathematical foundations and model," Tech. Rep. M74-244, The MITRE Corp., Bedford MA, May 1973.

[2] A. C. Myers and B. Liskov, "Protecting privacy using the decentralized label model," *ACM Trans. Softw. Eng. Methodol.*, vol. 9, pp. 410–442, October 2000.

[3] M. Krohn, A. Yip, M. Brodsky, *et al.*, "Information flow control for standard OS abstractions," in *SOSP '07*, (New York, NY, USA), pp. 321–334, ACM, 2007.

[4] D. Evans, D. M. Eyers, and J. Bacon, "Linking policies to the spatial environment," in *Proceedings of the IEEE International Symposium on Policies for Distributed Systems and Networks*, July 2010.

[5] D. Evans and D. M. Eyers, "Efficient policy checking across administrative domains," in *Proceedings of the IEEE International Symposium on Policies for Distributed Systems and Networks*, July 2010.

[6] J. Reed and B. C. Pierce, "Distance makes the types grow stronger: a calculus for differential privacy," in *Proceedings of the 15th ACM SIGPLAN international conference on Functional programming*, ICFP '10, (New York, NY, USA), pp. 157–168, ACM, 2010.