Position Paper on the Future of P3P

Jorge R Cuellar
Siemens AG

October 21, 2002

1 Language for classes of policies

For several purposes a language is needed to specify *classes* of policies (and/or preferences)[1], not a single policy. It should be possible then to check if a policy is within a policy class.

This could be interesting for validating or enforcing a policy and it will be important for delegating the right of writing policies.

There are situations where the user wants to delegate the right of making the policies. For instance a user could delegate the details of his presence or location information polices to her secretary. Or certain kind of policies, where he is not aware of the full details or trust relationships, may be delegated to an NGO offering privacy services. To do this, the must be able to describe the class of policies that are to be delegated.

2 Ontology of private data their order relationship (and later, the abstraction functions)

Many personal data have different degrees of "accuracy" or "granularity": my location may be given within 100 km accuracy or within 10 m; a report of my blood pressure may contain the exact numbers or just an indication of it is high or low; a medical information may report "heart problems" without going into more detail, etc. At least for location data and for a collection of e-health data it would be important to define the accuracy levels and the order relationships between them. Then a user (or the hospital if he is the rule maker) may say: "this data may be used in this particular way only with in accuracy level, or lower", etc.
An example is the following: a user may say that his location may only be given within an accuracy of 20 km, or lower. But location information is not only given in "latitude/longitude ± x km", but also as state/city, time zone, country, location within an airport, ellipsoids, etc. Suppose for some reason, the city or county of the user is requested. Depending on the city or county the user is located, the information "the user is in city/county X" may violate the policy or not. Therefore a simple ordering of the different "data types with accuracies" is needed, to automatically check the policy.

[1] We use here the word "policy" with a more general meaning, including both policies and preferences. A policy is a set of rules that stipulate how a set of data should be used.

Later it could be perhaps possible to define "standard" abstraction functions that translate an information item to one with less accuracy.

3 User choice: Data item policy tagging

The user should be in position to say: "this information is subject to this policy, while this more sensitive information is subject to a more restrictive policy". This may be combined with the "ontology, orders and abstractions" point mentioned before: the user could write: this one piece of data is subject to the following policy, but in this abstraction level (or accuracy), it is subject to a less restrictive policy. This is particularly interesting when the policies are communicated and also stored together with the date (sticky policies). If an application requests the use of the data in certain accuracy, the server may check then first which policy applies. Also if an application requests a data for a certain purpose, the server may check first with which accuracy this information may be used.