On Frameworks for the Visualization of Privacy Policy Implications

(Extended Abstract)

Rafael Accorsi and Thomas Stocker

Department of Telematics
Albert-Ludwigs-Universität Freiburg, Germany
{accorsi,stocker}@iig.uni-freiburg.de

Abstract. Privacy policies provide a way to automate the control of data access and usage across different systems and enterprise domains. Due to the ever growing complexity and number of policies, users are often unaware of the amount of information they implicitly release as a function of a given (possibly negotiated or combined) privacy policy. This is substantiated by a several experiments demonstrating that users fail to capture their privacy preferences when specifying their policies. Below, we motivate the need for frameworks to compute and visualize the implications of a policy, i.e. to make *implicit* access and usage decisions *explicit* to users. In enhancing the usability of policy specification and negotiation, users are eventually able to define more precise policies, which is an essential feature for current computing models based on social networks and cloud and ubiquitous computing.

Policies provide a way to automate the control of data access and usage across different systems and enterprise domains. The specification of such non-functional requirements demands expressive policy languages. Today, academia and industry have (often together) proposed a number of policy languages, such as XACML [7], ExPDT [9] and EPAL [2] deployed over a variety of enforcement architectures, such as [3, 4, 6]. These policy languages are on the one hand syntactically expressive enough to represent complex policy rules, and offer on the other hand a formal semantics for operators to reason about policies, e.g. their conjunction [8] and recently difference [5]. These operators considerably improve the usability of policy languages, facilitating the detection and resolution of conflicts and hence the deployment of large policies.

However, with the need for compliance automation and the massive service virtualization and delegation, as well as its distribution through ubiquitous and cloud applications, the number of policies and their complexity increased enormously, posing a challenge for correct policy specification [10]. Experiments and recent privacy breaches substantiate this,

showing that users (including experienced policy engineers) easily oversee the full implications of their privacy policies [11]. This leads to a situation in which users may *implicitly* allow the collection and usage of data they *explicitly* did not want to. This clearly has a negative impact, both for users, whose privacy perishes, and enterprises, which may eventually fail to be compliant.[1]

We motivate the need for frameworks to compute and visualize the implications of a policy, i.e. to make *implicit* access and usage decisions *explicit* to users. Given an arbitrary privacy policy $\mathcal{P}$ and an environment domain $\mathcal{C}$, implications would be statements of the kind "given domain $\mathcal{C}$, $\mathcal{P}$ implicitly discloses data items $d_1, \ldots, d_n$ which are used as $u_1, \ldots, u_m$". Despite the extensive number of different policy operators available today, the derivation of implications is in existing policy languages impossible.

With such a framework at hand, users could detect unwanted information leaks before committing to the policy, thereby improving the quality and precision of the resultant rules. This has a great societal relevance as well. The visualization of policy implications helps into solving the asymmetry between "data providers" and "data consumers": users obtain more control at making privacy decisions and can hence reduce risk of privacy threats, leading to more acceptance of technology [10].

Technical Challenges At a technical level, the realization of frameworks for the visualization of privacy policy implications encompasses two challenges. First, the design of algorithms and methods for the derivation of the implications from (a set of) policies and/or dynamic domain information; second, the presentation of these results in an intelligible manner, so that users can employ this information to strengthen or relax their privacy policies accordingly. Below, we report on some proposals to addressing these challenges.

With regard to the derivation of implications, some sort of calculi would be needed to derive implications of a given policy. Modern policy languages, such as ExPDT and XACML, build on top of an ontology. In the case of ExPDT, this is the smallest decidable set of W3C OWL-DL. Using the capabilities of the underlying semantic web, algorithms could implement an "inference engine" that takes as input data hierarchies and domain descriptions, outputting data items that implicitly follow from the original policy.

---

[1] See `http://www.pgp.com/insight/newsroom/press_releases/2008_annual_study_germany_cost_of_data_breach.html` for a recent study on the costs of data breaches in Germany.

Another, admittedly more speculative approach to addressing these challenge could include testing how data items are generally collected and processed by a data consumer and then relating this information to the original policy. To do so, one could analyze the workflow models used in Service-Oriented Architectures. Such workflows denote how data is processed within an enterprise and an analysis elucidates the information flows happening between the different subjects [1]. This could be represented as a graph or (equivalently) XML dialect, against which the original policy could be examined.

With regard to the visualization, the implications of a privacy policy need to be represented in a way that helps users to formulate strong privacy policies. A preliminary experiment we carried out in the context of policy (re)presentation without the automated derivation of implication consisted in dividing the data hierarchies in green (allowed by the policy) and red (prohibited by the policy) areas. The vast majority of subjects found this representation intuitive and "enlightening" – despite the inherent loss of expressivity caused by hiding additional information, such as the role and the purpose of an access. Advancing this area is clearly *not* the role of a standardization body, but we firmly believe that intensive discussion is needed to raise awareness of the importance and intricacies of (industrial) policy representation.

References

1. R. Accorsi and C. Wonnemann. Detective information flow analysis for business processes (extended abstract). In W. Abramowicz, L. Macaszek, R. Kowalczyk, and A. Speck, editors, *Business Processes, Services Computing and Intelligent Service Management*, volume 147 of *Lecture Notes in Informatics*, pages 223–224. Springer-Verlag, 2009.
2. P. Ashley, S. Hada, G. Karjoth, C. Powers, and M. Schunter. Enterprise privacy authorization language. Technical report, IBM Research, March 2003.
3. D. W. Chadwick and S. F. Lievens. Enforcing "sticky" security policies throughout a distributed application. In R. Scandariato and G. Russello, editors, *Proceedings of the Workshop on Middleware Security*, pages 1–6. ACM, 2008.
4. D. K. W. Chiu, S. C. Cheung, and S. Till. A three-layer architecture for e-contract enforcement in an e-service environment. In *Proceedings of the Annual Hawaii International Conference on System Sciences*, volume 3, page 74a, 2003.
5. M. Kähmer. Difference in security policies for dynamic systems. In R. Accorsi and J. Peters, editors, *Proceedings of the Workshop on Security in Autonomous Systems*, volume 183 of *CEUR Workshop Proceedings*. RWTH-Aachen, 2006.
6. M. C. Mont, S. Pearson, and R. Thyne. A systematic approach to privacy enforcement and policy compliance checking in enterprises. In S. Fischer-Hübner, S. Furnell, and C. Lambrinoudakis, editors, *Proceedings of the International Conference on Trust and Privacy in Digital Business*, volume 4083 of *Lecture Notes in Computer Science*, pages 91–102. Springer, 2006.

7. OASIS. Extensible access control markup language. `http://www.oasis-open.org/committees/xacml/`, 2008.

8. P. Rao, D. Lin, and E. Bertino. XACML function annotations. In *International Workshop on Policies for Distributed Systems and Networks*, pages 178–182. IEEE, 2007.

9. S. Sackmann and M. Kähmer. ExPDT: A policy-based approach for automating compliance. *Wirtschaftsinformatik*, 50(5):366–374, October 2008.

10. S. Sackmann, J. Strüker, and R. Accorsi. Personalization in privacy-aware highly dynamic systems. *Communications of the ACM*, 49(9):32–38, September 2006.

11. S. Trudeau, S. Sinclair, and S. Smith. The effects of introspection on creating privacy policy. In *To appear in the Proceedings of the Workshop on Privacy in the Electronic Society*. IEEE, 2009.