PSpec: A Formal Specification Language for Fine-Grained Control on Distributed Data Analytics

Chen Luo*†, Fei He*, Dong Yan‡, Dan Zhang‡, Xin Zhou‡ and Bow-Yaw Wang§
*KLiss, MoE; TNList; School of Software, Tsinghua University
Email: cluo8@uci.edu, hefei@tsinghua.edu.cn
†University of California, Irvine
‡Intel Labs China
§ Academia Sinica, Taiwan

***Abstract*—Organizations often share business data with third-parties to perform data analytics. However, the business data may contain a lot of customers' private information. One major concern of these organizations is thus to ensure such private information is properly used. In this paper, we present PSpec, a formal language for specifying data usage restrictions in distributed data analytics. Compared with previous works, PSpec specializes in data analytics and provides explicit support for data desensitization and association to balance data privacy and utility. We moreover present redundancy and conflict analysis algorithms to help data owners write PSpec privacy policies. To evaluate PSpec we carry out a case study on TPC-DS benchmark. The results demonstrate applicability and practicality of the PSpec language.**

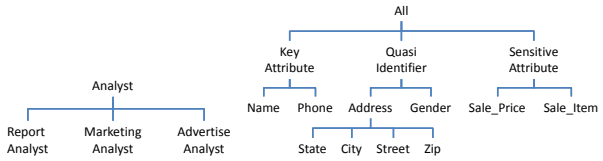
I. Introduction

It is highly desirable for organizations to discover values from business data with data analysis techniques. For example, a retail company may discover the sales trend to make future business decisions, and a hospital may share some medical records to researchers to promote the study of certain disease. However, these data often contain a lot of customers' personal information, such as the financial and health information. Improper use of these data can cause severe privacy breaches, which in turn will significantly degrade the organization's reputation and even incur charges or penalties from the government. Ensuring the privacy-related data are properly used is definitely a major concern of these organizations.

Lots of techniques have been developed to deal with the privacy issue, such as anonymization [1], differential privacy [2] or privacy-aware access control [3]. However, these techniques often suffer from several drawbacks in practice. For example, adopting differential privacy requires tedious analysis and mathematical insights, which can be too challenging at present.

A more practical approach is to let the data owner specify data usage restrictions with a specification. In this paper, we present PSpec, a formal language for specifying data usage restrictions for data analytics. PSpec provides explicit support for data association and desensitization. It moreover abstracts away details of underling data models and desensitization operations. To ensure the written specifications are sound, we further develop two analysis algorithms, namely redundancy analysis and consistency analysis. A rule is *redundant* if it has no effect on the entire specification; a set of rules are *inconsistent* if they may issue restrictions that cannot be satisfied simultaneously.

The work was done while the first author was with Tsinghua University. He is now with University of California, Irvine, USA.



(a) Example User Hierarchy (b) Example Data Hierarchy

Fig. 1: Example Vocabulary

II. PSpec Language

The scenario considered in this paper involves the following three participants: the data *owner*, who shares data with third-party analysts under agreements; the data *analyst*, who performs data analysis over the shared data; and the data analytics systems, which manages the data and allows the analyst to submit queries for data analysis.

The primary goal of PSpec is to allow the data owner to specify data usage restrictions such that the data can only be accessed by the privacy-compliant queries. To achieve usability, PSpec is designed as a high-level language to leave out details of the underling data models and queries. To be suitable for data analytics, PSpec provides explicit support for data association and desensitization.

PSpec comprises two parts, namely vocabulary and policy. In the vocabulary part, the data owner defines a set of hierarchical user categories and data categories. A user category represents a role. A data category represents a privacy-related data concept. Fig. 1 shows category hierarchies for a retail company as an example.

In the policy part, the data owner defines a set of PSpec rules to regulate data access. Informally, each rule states under what *restrictions* can a *user category* perform a *data association*. A data association means to access certain *data categories* together. A restriction specifies a set of admissible

978-1-5386-1589-8/17 $31.00 © 2017 IEEE
DOI 10.1109/ICSE-C.2017.120

desensitization operations for each data access in the data association.

Consider the privacy policy of a retail company. First, the company forbids any access to customer names by *Analyst*:

```
r1: Analyst,[access Name]=>forbid
```

Second, the company further requires the sales data should be aggregated when outputted with personal information:

```
r2: Analyst,[output All exclude Sensitive Attribute,
output Price]=>[{},{avg,max,min,sum}]
```

In r_2, we define a data association of length two. The first data access refers to the output of all but sensitive data categories. The second data access refers to the output on sales price. The rule r_2 requires *Price* to be aggregated when outputted with personal information together. Finally, the company forbids any access to birth, gender, and zip together:

```
r3: Analyst,[access Birth,access Gender,access Zip]=>forbid
```

Note that r_3 only forbids access to them together. One can access any one or two of them freely.

III. Policy Analysis

A data usage specification must be semantically sound. We discuss in this section redundancy analysis and consistency analysis for PSpec.

A rule r' is said *redundant* with respect to another rule r if for any query q, if q satisfies r, q also satisfies r'. The redundant rules not only cost unnecessary time for policy enforcement, but also may represent potential errors or unintended side-effects in a policy [4]. For example, consider the following rule r_4 and the previous rule r_2:

```
r4:Analyst,[access Price]=>[{},{avg}]
```

Obviously, r_2 becomes redundant since whenever r_4 is satisfied, i.e., *Price* is averaged, r_2 must also be satisfied.

To check whether rule r' is redundant w.r.t. rule r, we need to check whether r is applicable to more queries, i.e., scope checking, and is more restrictive, i.e., restriction checking. All PSpec rules are encoded into logical formulas. SMT solvers, e.g., Z3 [5], are employed to perform redundancy analysis.

Moreover, some rules may issue conflict restrictions, which lead to inconsistency. Note that data association makes the consistency analysis much more subtle since data associations with different lengths often have different privacy implications and should be treated separately. To this end, we first fix a seed rule r_s. A set of rules R is then said to be *inconsistent* w.r.t. r_s if there exists a query q such that q triggers r_s, and rules in $R \cup r_s$ can not be satisfied together.

For example, consider the following seed rule and the previous rule r_4.

```
rs:Analyst,[output Zip, access Price]=>{{},{min,max}}
```

Obviously, whenever one analyst outputs both *Zip* and *Price*, r_s is triggered. However, r_s and r_4 cannot be satisfied together, since r_s requires *Price* to be aggregated using *min* or *max*, while r_4 requires *Price* to be averaged.

In practice, we are only interested in minimal inconsistent rule sets. We apply the *levelwise search* technique [6] to find all minimal inconsistent rule sets. For each rule set, we check whether there exists a user category and a data association in r_s leading to inconsistency, again using SMT solvers.

IV. Implementation and Evaluation

We have implemented a PSpec parser with XML, and policy analysis algorithms with Java and Z3 [5].

We use a case study on TPC-DS [7] to evaluate how easily can PSpec be grasped by a non-expert user. One junior undergraduate student (who have received *one-hour* training in PSpec, and is familiar with the TPC-DS) was demanded to write PSpec rules to ensure the personal data in TPC-DS is properly used. This student succeeded in writing a vocabulary and all 13 PSpec rules in *three hours*.

We performed extensive experiments on synthetic rules to evaluate the performance of the policy analysis algorithms. Up to 50 user categories, 100 data categories, and 1000 PSpec rules are randomly generated for evaluation. Both algorithms finished in reasonable time and thus show their fitness for practical use.

V. Related Works

Several privacy languages have been proposed to formalize text-based policies, including P3P [8], EPAL [3], XACML [9], and P-RBAC models [10]. These languages, however, have weak support for data association and desensitization, and cannot specify fine-grained privacy specifications. Besides, In order to automatically enforce privacy and security requirements on programs, several extensions for programming languages have been proposed[11], [12], [13], [14], [15], [16]. However, PSpec is a new privacy specification language, not an extension to existing programming languages.

Policy analysis has been extensively studied in past decades. These works range from system configuration policies [4], firewall policies [17], [18], to access control policies [19], [20]. Although the intuition is similar, the policy analysis algorithms in this paper are pertaining to PSpec. They are further complicated by data association.

VI. Conclusion

In this paper, we present the language PSpec for specifying data usage restrictions in data analytics. To facilitate the user reason and analyze the PSpec policy, we also introduce two policy analysis algorithms, namely redundancy analysis and conflict analysis.

In the future, we plan to perform more real-world case studies to further evaluate the applicability of PSpec. Moreover, we also plan to explore other policy analysis algorithms to help users write and analyze their policies.

VII. Acknowledgment

This work was supported in part by the Chinese National 973 Plan (2010CB328003), the NSF of China (61672310, 61272001, 91218302) and the Chinese National Key Technology R&D Program (SQ2012BAJY4052).

REFERENCES

[1] L. Sweeney, "k-anonymity: A model for protecting privacy," *International Journal of Uncertainty, Fuzziness and Knowledge-Based Systems*, vol. 10, no. 05, pp. 557–570, 2002.

[2] C. Dwork, "Differential privacy," in *Automata, languages and programming*. Springer, 2006, pp. 1–12.

[3] P. Ashley, S. Hada, G. Karjoth, C. Powers, and M. Schunter, "Enterprise privacy authorization language (epal 1.2)," *Submission to W3C*, 2003.

[4] D. Agrawal, J. Giles, K.-W. Lee, and J. Lobo, "Policy ratification," in *Policies for Distributed Systems and Networks, 2005. Sixth IEEE International Workshop on*. IEEE, 2005, pp. 223–232.

[5] L. De Moura and N. Bjørner, "Z3: An efficient smt solver," in *Tools and Algorithms for the Construction and Analysis of Systems*. Springer, 2008, pp. 337–340.

[6] H. Mannila and H. Toivonen, "Levelwise search and borders of theories in knowledge discovery," *Data mining and knowledge discovery*, vol. 1, no. 3, pp. 241–258, 1997.

[7] R. O. Nambiar and M. Poess, "The making of TPC-DS," in *Proceedings of the 32nd international conference on Very large data bases*. VLDB Endowment, 2006, pp. 1049–1058.

[8] L. Cranor, M. Langheinrich, M. Marchiori, M. Presler-Marshall, and J. Reagle, "The platform for privacy preferences 1.0 (p3p1. 0) specification," *W3C recommendation*, vol. 16, 2002.

[9] T. Moses *et al.*, "Extensible access control markup language (xacml) version 2.0," *Oasis Standard*, vol. 200502, 2005.

[10] Q. Ni, E. Bertino, J. Lobo, C. Brodie, C.-M. Karat, J. Karat, and A. Trombeta, "Privacy-aware role-based access control," *ACM Transactions on Information and System Security (TISSEC)*, vol. 13, no. 3, p. 24, 2010.

[11] J. Yang, K. Yessenov, and A. Solar-Lezama, "A language for automatically enforcing privacy policies," in *ACM SIGPLAN Notices*, vol. 47, no. 1. ACM, 2012, pp. 85–96.

[12] J. Reed and B. C. Pierce, "Distance makes the types grow stronger: a calculus for differential privacy," *ACM Sigplan Notices*, vol. 45, no. 9, pp. 157–168, 2010.

[13] M. Gaboardi, A. Haeberlen, J. Hsu, A. Narayan, and B. C. Pierce, "Linear dependent types for differential privacy," in *ACM SIGPLAN Notices*, vol. 48, no. 1. ACM, 2013, pp. 357–370.

[14] A. C. Myers, "Jflow: Practical mostly-static information flow control," in *Proceedings of the 26th ACM SIGPLAN-SIGACT symposium on Principles of programming languages*. ACM, 1999, pp. 228–241.

[15] C. Hammer and G. Snelting, "Flow-sensitive, context-sensitive, and object-sensitive information flow control based on program dependence graphs," *International Journal of Information Security*, vol. 8, no. 6, pp. 399–422, 2009.

[16] J. Chen, R. Chugh, and N. Swamy, "Type-preserving compilation of end-to-end verification of security enforcement," in *ACM Sigplan Notices*, vol. 45, no. 6. ACM, 2010, pp. 412–423.

[17] E. S. Al-Shaer and H. H. Hamed, "Discovery of policy anomalies in distributed firewalls," in *INFOCOM 2004. Twenty-third AnnualJoint Conference of the IEEE Computer and Communications Societies*, vol. 4. IEEE, 2004, pp. 2605–2616.

[18] H. Hu, G.-J. Ahn, and K. Kulkarni, "Detecting and resolving firewall policy anomalies," *Dependable and Secure Computing, IEEE Transactions on*, vol. 9, no. 3, pp. 318–331, 2012.

[19] D. Lin, P. Rao, E. Bertino, N. Li, and J. Lobo, "Exam: a comprehensive environment for the analysis of access control policies," *International Journal of Information Security*, vol. 9, no. 4, pp. 253–273, 2010.

[20] H. Hu, G.-J. Ahn, and K. Kulkarni, "Anomaly discovery and resolution in web access control policies," in *Proceedings of the 16th ACM symposium on Access control models and technologies*. ACM, 2011, pp. 165–174.