DatShA :A *Data Sharing Algebra* for access control plans

Luc Bouganim
InriaSaclay
luc.bouganim@inria.fr

AthanasiaKatsouraki
InriaSaclay
athanasia.katsouraki@inria.fr

Benjamin Nguyen
INSA Centre Val de Loire
benjamin.nguyen@insa-cvl.fr

1. INTRODUCTION

Online social networks (OSN) are one of the most successful applications that have been created this last decade. Central to these applications is the problem of sharing data, such as texts, photos, geolocation, etc. In most cases, this data is private, and thus is only shared with "friends", a loose concept. Some OSN, such as Google+ let you define *circles* in order to categorize your friends: friends, close friends, acquaintances, etc. Data can then be shared on finer grain using these circles. However, there is no automatic way to control the simultaneous sharing of data to several circles, with different data precision granularities, such as in the following scenario: *Alice wants to share a set of photos with her family, photos with no metadata with her close friends, photos without faces (and without metadata) in a reduced definition with her acquaintances, and does not want to share anything with anyone else.*

In this article, we will show how the use of a data sharing algebra to write a variety of *access control plans* (ACP) can overcome these current limitations of OSN access control. Moreover, by using an algebra, it becomes simple to modify, compose, and share these ACPs. Thus less advanced users can easily reuse ACPs shared on a marketplace by more experienced users.A prototype of the DatShA system has been implemented using XQuery 3.0 and is briefly described.

2. OVERVIEW OF DatShA

In current OSNs, users have on one side vast quantities of personal data, and on the other side numerous "friends" with whom they wish to share (or sometimes hide) this data. In the current systems, it is not obvious how to share a specific piece of data while modifying it (e.g. changing its precision or removing information) depending on the target with whom it is shared.

Consider the examples mentioned in the introduction. The ACP related to Alice's close friends should transform a set of photos to another set where metadata is removed. This could be done by simply specifying a regular expression to identify images files to be shared (FileSearchoperator – see Figure 1.e), "type"of this file to images (PathToImage operator – see Figure 1.e), then remove metadata (RemoveMeta operator). For Alice's acquaintances, other operators could be invoked: ExtractFaces, ExtractMeta, Select and ReduceDefinition operators (not detailed here).

Thus the objective of DatShA is to provide the infrastructure and an extensible set of generic operators to describe how users would prefer to process their data before sharing it. The operators must be able to be combined on any sort of (semi-structured) data to form analgebra. Finally, ACP may include user-dependent data (e.g., contact files) such that it can also compute the set of users with whom the data isshared, thus linking a *plan* with its*grantee*.

3. BACKGROUND AND RELATED WORKS

Access Control. Many different access control models exist, such as DAC, MAC, or RBAC. Many works exist on enforcing such models in OSN [1]. We adopt a complementary approach: the goal of DatShA can be seen as helping the user to write complex views of her data, on which she can then apply any existing AC model (most often, DAC or RBAC).

Data Sharing on OSN. Current works on secure data sharing in OSNs consider various problems such as securing communications, i.e. how to securely share data, once access control has been checked [2], or how to write access control policies over data concerning several users [3].

XQuery 3.0. XQuery 3.0. is not only a declarative query language, it is also Turing complete. Rather than using a traditional language such as Java or C, we have chosen to use XQuery and XQuery Update Facility 3.0. Indeed, evaluating an ACP is done through modifications to a structured document (that we chose to code in XML). Generic operators can be completed by snippets of XPath or XQuery code referring to this data structure, which are directly evaluated by the DatShA system.

4. THE DATA SHARING ALGEBRA

4.1 General principle

An ACP is seen as a set of sequences of (polymorphic) operators, serialized as an XML file (see Figure 1.a). It takes as input an XML file containing or referencing private sensitive data and produces an XML file containing or referencing data that can be shared or published (See Figure 1.c). Users or sets of users (such as G+ circles) can be given access rights both on atomic data, and on ACPs. As with traditional access control through views, when access rights are given on an ACP, the data accessed during the process is done with the rights of the *grantor*. For example, if Alice grants Bob the right to view the country where she is in, which is computed using her precise GPS coordinates, the execution of the ACP will use Alice's rights, but only return to Bob the final result.

4.2 Sharing ACPs through a marketplace

Operators and ACPs can be published on a "marketplace", and described by a short text explaining their goal.They can be downloaded by users in order to fine tune their data sharing policies. Thus, it is possible, even for non-expert users to apply complex access control policies, by combining existing operators or using existing policies. Search, recommendation, or ranking of ACP or operators based on their level of intrusiveness or their usability is possible within the marketplace. The only complexity is to link groups of users to their ACPs, but as the data shared is defined *intentionally* rather than *extensionally*, we believe this is much easier to do than with current privacy settings in OSN.

4.3 ACP Example

We propose the following example which illustrates well DatShA potential : *Alice wants to participate in a survey to determine the most photographed place on Earth, which can be done by computing a "fuzzy" location of all her photos, where the "fuzzy" location is defined by GPS coordinate and an error bar e.g. X=45.23+/-0.01 Y=27.67+/-0.01.* Note that this error bar could also be a function of the density of photos in a given area.

We present, in Figure 1, the corresponding ACP.

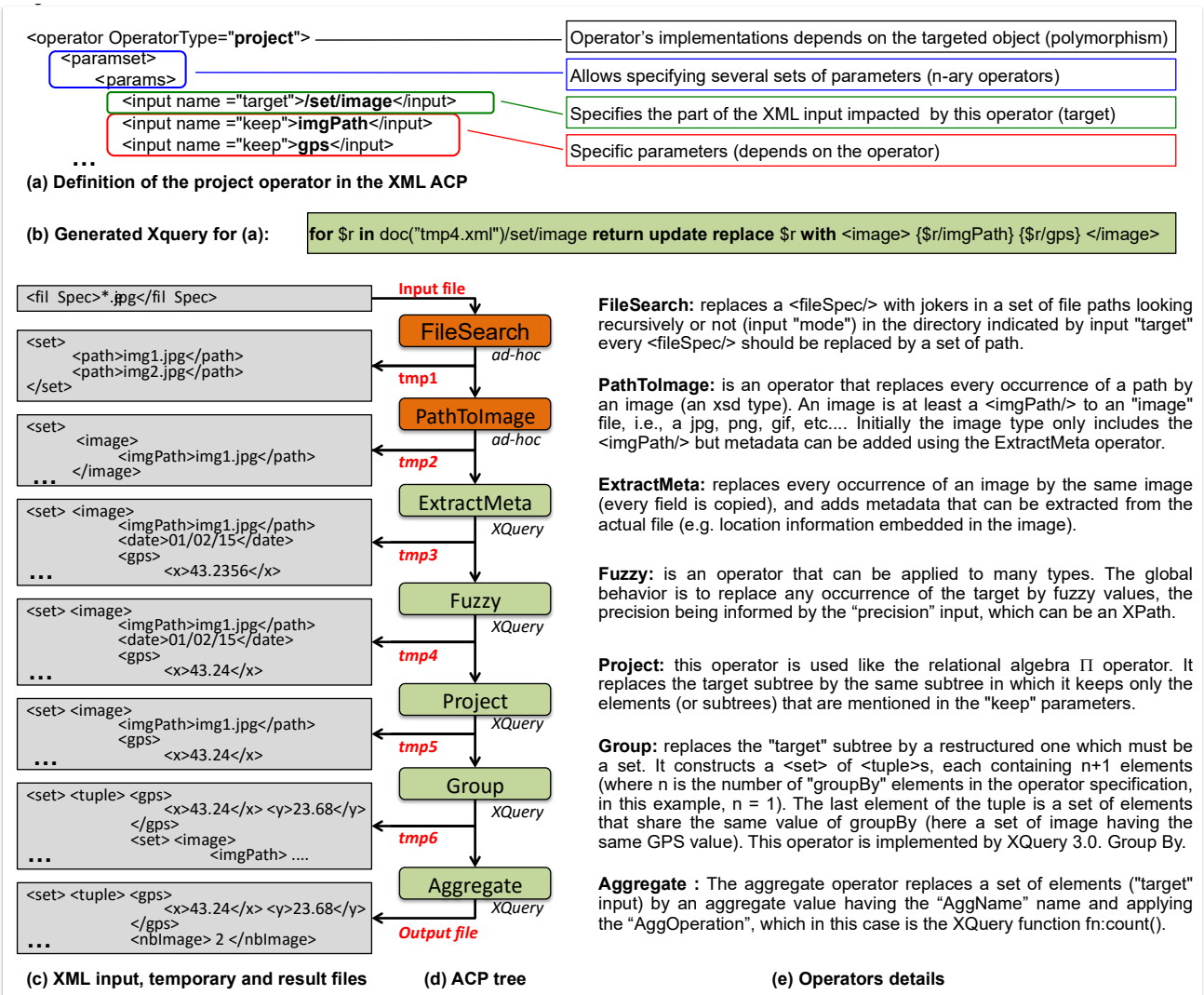

Figure 1: A detailed example of an ACP delivering statistics (number of photos by fuzzy location)

This ACP can be written as a sequence of operators (Due to space limitations, the actual XML ACP file is available online at: http://www.benjamin-nguyen.fr/data/ACP.xml). Each operator takes as input an XML file, and produces as output an XML file.It is possible to type-check the ACP at compile time, given that the operators are typed, but this discussion is beyond our scope here.The sequence is the following: for every image in the file path given in the input file, meta-data of the image is extracted, and an operator to reduce the precision of the GPS coordinates is executed. All meta-data apart from the blurred GPS coordinates is removed, pictures are grouped together by fuzzy GPS location, then counted.

Operators can in most cases be implemented in XQuery 3.0 but they can also be *ad-hoc* operators. Note that DatShA also defines many other operators, including binary (or even n-ary) operators such as the join operator which can be used to join two different sequences, thus needing two input files, and producing a single output file.All these operators have been implemented using XQuery. The framework executing a DatShA ACP has been written in Java, using eXistDB to execute the XQuery fragments. We believe that the use of operators to build ACPs drastically simplifies the creation, reuse, combination and correctness checking of ACPs.

5. CONCLUSION AND FUTURE WORK

DatShA can be used to create ACPs to manage access control to one's data. We believe that the full power of DatShA appears when users start sharing ACPs between each other, either by simply reusing an ACP written by another user, or by integrating such an ACP into a more complex one. Indeed, any ACP can be encapsulated as a DatShA operator. Creating an online market-place, and testing its usability and adoptability by real users is the next step of our work.

6. REFERENCES

[1] B. Carminati, E., and A. Perego. 2009. Enforcing access control in Web-based social networks. *ACM Trans. Inf. Syst. Secur.* 13, 1, Article 6 (November 2009).

[2] H. Qinlong, M. Zhaofeng, Y.Yixian, N.Xinxin, F. Jingyi, Improving security and efficiency for encrypted data sharing in online social networks in IEEE China Communications, 11(3):104-117, 2014

[3] H. Hu, G-J. Ahn, J. Jorgensen: Multiparty Access Control for Online Social Networks: Model and Mechanisms. IEEE Trans. Knowl. Data Eng. 25(7): 1614-1627 (2013)