Access Control Policies for XML:
Verification, Enforcement and Collaborative Edition

ACCESS

Proposition d'Action de Recherche Collaborative INRIA, appel 2010-2011

Summary

This project is concerned with the security and access control for Web data exchange, in the context of Web applications and Web services. We aim at defining automatic verification methods for checking properties of access control policies (ACP) for XML, like consistency or secrecy, and for the comparison ACPs. One of our goals is to apply formal tools from tree automata theory for this purpose.
A second important goal is to design efficient methods for ACP enforcement for secure query evaluation. We will study several scenarios for solving different variants of this problem, based on the notion of secure user views. As a case study, we will apply our methods to an XML-based collaborative editing system.

Keywords: Access Control, Internet Data Security, Distributed Databases, XML, Formal Verification, Query Rewriting, Security Views, Collaborative Edition Systems, Tree Automata, Term Rewriting.

Responsible: Florent Jacquemard, Dahu team, Research Center INRIA Saclay – Île-de-France.

1 Participants

Cassis and Pareo team (INRIA Nancy – Grand-Est).

The expertise of the Cassis team in symbolic verification methods will permit to contribute to aspects concerned with formal verification of ACP. The Pareo team studies the theoretical foundations and the practical applications of strategic rewriting. The theoretical and practical tools developed in the team have been used for the specification and analysis of security policies. The following members of the teams Cassis and Pareo will be involved in the project:
Horatiu Cirstea is permanent member of the INRIA Grand Est Pareo team. His main research interests include the theoretical foundations of rewriting and logic and their practical applications to the security and safety of information systems.
Abdessamad Imine is permanent member of the INRIA Grand Est Cassis team. He is working on access control policies for distributed editors, safe updating strategies for firewall policies and consistency verification of collaborative systems.
Michael Rusinowitch is the founder and current leader of the INRIA Grand Est Cassis team. He is working on automated deduction and application to software verification. He is particularly interested to the development of decision procedure for security properties.
Asma Cherif is a 2nd year PhD student at University of Nancy. She is a member of EPI Cassis and she works on access control models for collaborative editors.

Dahu team (INRIA Saclay – Île-de-France).

The Dahu team is part of the INRIA Research Center of Saclay-Ile de France, and aim at providing solid foundations for data centric specification and verification in an Internet environment, in order to impact on the new generation of systems manipulating data over the Internet, making them safer and more reliable. In particular this encompass modeling and verifying ACP. This team has solid expertise in many of the facets of this proposal including specification, distributed databases and XML, tree automata and verification. The Dahu team is located at the Laboratoire Spécification and Vérification (LSV) which is a research unit combining resources from ENS Cachan (Ecole Normale Supérieure de Cachan), CNRS and INRIA-Saclay. The following members of the team Dahu will be involved in the project.
Luc Segoufin is the founder and current leader of the INRIA Saclay Dahu team. His main interests are database theory, finite model theory, and verification. His recent contributions concern the expressive power of logics over trees, query rewriting using views and verification of data-driven systems.

Florent Jacquemard is the vice-responsible of the INRIA Saclay Dahu team. His research topics include automated verification of systems and software, tree automata and logics, automated deduction and term rewriting. He will be responsible of this project.
Camille Vacher is a third year PhD student at France Telecom R&D and the Laboratoire Specification and Verification (LSV), UMR CNRS and ENS Cachan. He is also member of the INRIA Saclay Dahu team. His thesis focuses on extended tree automata models and their application to security protocol verification.

Mostrare team (INRIA Lille – Nord Europe).

The main topic of the Mostrare team is information extraction on tree structures (XML, HTML, ...), organised in two research tracks : modelling of tree structures and tree query languages, and developing machine learning techniques. The group is recognized for its work on tree automata, which these days are fundamental to XML querying and transformation. It combines various competences on logic, constraints, automata, and XML database theory. The members of Mostrare involved in this project are :
Sophie Tison, full professor in Computer Science, Univ. Lille 1 and member of the Mostrare team. She is currently director of the computer science department (LIFL). Her research topics include automata, logics, term rewriting, currently applied to XML.
Iovka Boneva, assistant professor, Univ. Lille1, member of the Mostrare team. She is currently interested in confidentiality in XML databases. Her previous research topics include tree automata and logic, and graph transformations.
Anne-Cécile Caron, assistant professor, Univ. Lille 1, member of the Mostrare team. She applies tree automata techniques to XML querying or updating.
Yves Roos assistant professor, Univ. Lille 1, member of the Mostrare team. His research topics are automata and formal methods and their applications to XML querying. He is also concerned in learning algorithms for regular languages.
Slawek Staworko assistant professor, Univ. Lille 3, member of the Mostrare team. He is interested in machine learning of XML transformations, security of XML databases, and consistency management and database repairing.
Benoît Groz 2nd year PhD student at Univ. Lille1, member of the Mostrare team. He works on access control in XML databases in presence of multiple users.

2 Objectives of the collaboration

2.1 Context and State of the Art

XML has developed into the de facto standard for the exchange and manipulation of data on the Web [1]. An increasing number of applications use XML as their data model or as a format to export other data. That includes in particular user content exchange applications (social networks, mashups, blogs, photo sharing sites, wikis...) for which privacy is a central issue. It is therefore critical to investigate the problem of access control for XML documents.

The specification, enforcement and verification of **Access Control Policies** (ACP) for XML differs from approaches existing in other domains. An XML document is a textual representation of data stored in a tree structure, commonly presented as a finite labeled tree. In this representation, the data is not only stored in the labels of the nodes of the tree but also in the structure of the tree itself. The tree structure may in particular induce dependencies between a node and its ancestors in the tree for access authorizations. In contrast, there are no such dependencies between the entities of relational tables. Moreover, conditional access is possible for XML. For instance, it is possible to grant the access to the nodes of a subtree containing some personal user information only if the subtree has a node labeled with an appropriate identity. This situation does not occur in the case of the access control for the UNIX file system, even though it also deals with a tree structure.

Several approaches have been proposed in the literature for the specification of XML access control policies. Most of them use declarative languages [12, 3, 22, 21], based on sets of rules. Each rule typically specifies

- a *requester*, i.e. the user or group concerned by the authorization,
- a *resource*, which is the part of a XML document the requester is authorized (or forbidden) to access. It is given by a node or a set of nodes, defined in general by an Xpath expression,
- an *action* (read, write, delete, rename...),
- an *authorization* ("granted" or "denied") and,
- the scope of the rule, i.e. how the authorizations are propagated of from parent to child nodes.

In the category of rule based ACP, XACML [29] has emerged as a standard. The XML documents under access control are sometimes assumed to comply to type restrictions, defined by a **schema**, typically a DTD (i.e. they must be valid for a given DTD). In this case, it is convenient to specify the ACP by adding some annotations to this schema [16].

The notion of **security views** [33] is central in many approaches to XML ACP enforcement, e.g. [16, 17, 31, 23]. These approachs can be roughly summarized by the following scenario. Given an XML document D, a schema τ and an ACP P (based on this schema), a virtual view D_v is defined, which comprises all the nodes of D accessible wrt the ACP P. This view D_v is not materialized (it would be too space consuming and inefficient). Instead, its schema τ_v is computed (from τ) and presented to the users (the complete document D and its schema τ remain unknown to unauthorized users).

A user can formulate a query q_v on the view schema τ_v, and, instead of being evaluated on D_v (which was not materialized), the query q_v is rewritten to a query q on τ, which is then evaluated on D. The queries q and q_v are equivalent in the sense that the evaluation of q on D returns the same result as the evaluation of q_v on D_v.
There are several parameters in the above approach.

- the class of schemas considered (DTD, restricted DTDs, tree automata...),
- the language for the definition of resources in the ACP P (fragment of Xpath),
- the kind of scope allowed for ACP rules (single node, propagation to descendant...),
- the language for user queries...

Another important question is whether we consider ACP for read-only access, like in the works cited above, or read/write ACPs, e.g. for XML update operations like node renaming, deletion, insertion... see e.g. [20]. The case of read/write ACP can be complicated by the presence of **multiple users** with concurrent access to a document, like in the case of collaborative editors. Indeed, in this context it may be the case that several individual write access are allowed but the combination of them (i.e. the global change to the document) is not allowed by the ACP. A related interesting problem is the case of **dynamic ACP**, which can be updated by users.

Formal language theory, and more particularly **tree automata** theory (see e.g. [10]) is used intensively in the definition of XML standards and XML processing techniques, e.g. document validation and querying, see [32]. Indeed most of the typing formalisms currently used for XML are based on finite tree automata. In the context of XML ACP and user view definition, tree automata techniques have been used for instance for static analysis of queries [28], or for verifying security properties of XML views [27].

Rewriting systems have been also used for the study of a broad range of security issues and in particular for the specification, implementation, and validation of security policies. For instance, policies for control of information leakage [14] and policies that are used to protect resources in centralised computer systems [2], have been specified as rewrite systems. When using

such an approach we can apply rewriting techniques to study their properties [13]; for example, we can check the confluence and termination of the reduction relation induced by the rewrite rules and thus the consistency of the subsequent policy. Moreover, the analysed rewriting systems can be straightforwardly implemented using rewrite based languages such as TOM (`http://tom.loria.fr/`), a language extension which adds new matching primitives to existing imperative languages. One of the important features of Tom is the support for equational matching and in particular, for list matching. This characteristic is essential to manipulate and analyze XML documents in an abstract way: an XML document can be seen an algebraic term where the list of elements are considered to be associative. In this context, TOM provides a standard XML syntax to retrieve information and transform an XML document [8]. This system has been already used to describe and analyse communication protocols and security policies [7, 6].

2.2 Objectives

We will define automatic verification methods for checking properties of access control policies (ACP) for XML, like consistency or secrecy and for the comparison ACPs. One of our goals is to apply formal tools from tree automata theory for this purpose (§ 2.3.1 and § 2.3.2 below).

Another important problem is to design efficient methods for ACP enforcement for secure query evaluation. We will study several scenarios for solving this problem, for different variants of the parameters mentioned above, based on the notion of secure user views (§ 2.3.3 and § 2.3.4).

As a case study, we will apply our methods to an XML-based collaborative editing system. (§ 2.3.5).

2.3 Scientific Program

2.3.1 Verification of ACP Properties

We aim at developing formal methods for the automated verification of some properties of access control policies, like

- *consistency*: absence of conflicts, such as the possibility to obtain two conflicting access rights for the same pair of requester and resource.
- *completeness*: the ACP defines authorizations for all the considered resource and requesters.
- *emptiness*: each resource can be accessed by at least one requester.

Besides these properties, the problem of *secrecy* (wether given sensitive piece of data can never be accessed without an explicit authorization) is also mentioned below. These problems could also be generalized to more complex safety properties.

We are planning to tackle the problem of formal ACP verification following a language theoretic approach, based on tree automata. This formalism is already widely used in fields related to the subject of this project, like XML processing or security protocol verification. The idea is to do a static analysis by reduction of the above properties to decision problems for tree automata. **Regular model checking** techniques should be a promising method for solving the complex cases of read/write ACPs (for instance for XML updates) or dynamic ACPs. In this approach (which was already successfully applied to the static analysis of several kind of programs [4]), infinite set of reachable states are represented as finite word or tree automata in order to prove safety properties of systems.

The Dahu team will be the coordinator of this task, which completion will benefit from expertise of the Mostrare, Cassis and Pareo teams in tree automata, formal verification, and term rewriting techniques.

2.3.2 ACP Comparison, Secret and Public Information

We are planning to devise formal tools for answering the following questions:

1. *comparison*: which one of two given ACPs is more restrictive?
2. *secrecy*: does a given ACP protect the secret it was designed for?
3. *publicity*: does it make available the information which was intended to be public?

The simplest comparison of ACPs consists in comparing the parts of the document which are hidden by each of the ACPs. However, this naive method is not satisfactory, as it may be the case that an ACP which hides more, also reveals more information. In [23], we proposed another criterion for measuring the restrictiveness of an ACP, bases on the queries that can be answered through the ACP. We propose to study other information-oriented criteria for comparing ACPs.

Secrecy and publicity of information are related to ACP comparison. Before answering questions 2. and 3. above, a first step is to give a formal definition for secret and public information. As for comparison of ACPs, query-based definitions make sense : i.e. information is a (possibly non-monadic) query. We will also be interested in the problems raised by the presence of multiple users or roles that leads to multiple ACPs. In particular, if a user has two different roles in the system (i.e. two different ACPs), the administrator should be able to check whether the information hidden by both ACPs taken separately remains secret. One may also want to compute a single ACP which combines all the roles of a given user. This leads us to a last and more difficult problem : Is it possible to infer an ACP from a set of queries that represent public and secret information ?

The Mostrare team (coordinator of this task) and the Dahu team will bring together their competences in formal language theory, tree automata, logic and database theory for completing these goals.

2.3.3 Query Rewriting and Update Propagation

A secure user view (see § 2.1) is the portion of a document which is accessible for the user according to an ACP. For space constraints, it is preferable not to materialize the user views. In this case, user queries expressed on the view have to be translated into queries to be evaluated on the original XML document. This technique is known as query rewriting (read-only queries) or update propagation (update queries).

Query rewriting was studied in [18, 23]. We are planning to extend these results to n-ary queries, to richer query languages and to more complex ACPs, by using tree-automata and language theory techniques.

Update propagation is much harder than query rewriting. Several issues have to be tackled since the update propagation may change sensitive hidden parts of the document, e.g. the secure view of the updated document is not as expected (side effects) or the updated document does not match the (hidden) document schema. Several of these problems arise also in collaborative edition frameworks, but for less general settings. It is impossible to avoid all of these issues for reasonable ACP and schema definitions. Depending on the particular application, one may want to give priority to some of the requirements. We aim at:

- defining one or several update propagation mechanisms;
- identifying conditions (on ACP definitions, schema definitions and update languages) which guarantee desired properties.

We are planning to apply techniques and tools such as tree automata, tree transducers, logic and tree repairing.

The Mostrare team will be coordinator for this task, and will collaborate with the Dahu team on the query rewriting problem. The Mostrare team is particularly interested in update propagation, and welcomes the experience of the Cassis team on collaborative editors for attacking the problem.

2.3.4 Computing Secure User Views

Most of the work on XML access control policies verification and enforcement are concerned with schema which are regular tree languages (languages of finite tree automata) or DTDs (which are a strict restriction of tree automata). However, in some cases, the view schema τ_v associated to a schema τ and an ACP (see § 2.1) cannot be characterized by a regular tree automaton, and that some strict extension is needed. We aim at studying some

strict **extensions of tree automata** for unranked trees similar to the known extensions of their counterpart for ranked trees: extensions with an auxiliary memory stored in a stack [24, 5] or a tree [9, 11], or local or global tests of isomorphisms between subterms [26, 19, 25]. The extended classes of automata, though they are not standard type schemas, will be useful for static analysis of XML ACP. The study of the extended automata include the development of a number of desired constructions algorithms like combination of languages under union, intersection and difference and decision algorithms for problems like emptiness or inclusion.

A complementary topic of interest is the study of **approximations** of the set τ_v of user views by a regular tree language, or even by a DTD-definable tree language. This is useful for e.g. providing to the user a document type in some well-known formalism, thus guiding him for formulating queries. In [23] we have proposed some simple approximations which preserve desired properties, such as indistinguishability w.r.t. a class of queries. We plan to extend this work by optimizing other criteria, such the as size of the representation of the approximation.

The Dahu team will be coordinator for this task; it will collaborate with the Cassis team on the study of classes of extended tree automata and their applications, and with the Mostrare team on the study of approximations.

2.3.5 ACP in XML-based Collaborative Editing Systems

Distributed Collaborative Editors (DCE) belong to a class of distributed systems which enables several and dispersed users to form a group for editing documents such as XML documents (e.g. Google Wave). To ensure data availability, the shared documents are *replicated* on the site of each participating user. Each user modifies locally his copy and then sends this update to other users. DCE are characterized by human interactions. So, they should be as *higher responsive* as single-user editors [15, 34]. One of the most challenging problem in DCE is balancing the computing goals of collaboration and access control to shared information [35]. Indeed interaction in collaborative editors is aimed at making a shared document available to all users who need it, whereas access control seeks to ensure this availability only to users with proper authorization. However, when adding an access control layer, high responsiveness is lost because every update must be granted by some authorization coming from a distant server.

In this project, we propose a new access control model for editing collaboratively XML documents where we replicate the ACP on every user site. Thus, a user will own two copies: the shared XML document and its ACP. Note that this ACP may contains both confidentiality (read access) and integrity (update access) rules. It is clear that this replication enable users to gain performance since when they want to manipulate (read or update) the

shared document, this manipulation will be granted or denied by controlling only the local copy of the ACP. This model raises interesting issues.
Using the existing XML access models is not well-suited for DCE. For example, consider two users working concurrently on a shared XML document controlled by an ACP that restricts to a node n to have only one child. If n have initially no child and each user creates one child, this change is locally granted by the ACP. But after exchanging their write operations, the integration of theses changes is not allowed at any user site. What action should be taken to merge the contribution of all collaborators according to the local ACP? It is possible to cancel (or undo) some write access to enforce the local ACP but at the expense of losing the work done by other collaborators; this is often considered as a undesirable situation in collaborative applications. On the other hand, we can update the local ACP to preserve some operations after integration. This requires us to deal with **dynamic ACP** and the consistency problem – make all ACP copies converge to the same state.

As DCE have to allow for dynamic groups, they require dynamic change of access rights. It is possible to achieve this goal when duplicating access rights. The ACP can be edited by one or many users called administrators. Thus, updates locally generated by the administrators are then broadcasted to other users. The shared XML document's updates and the ACP's updates may be applied in different orders at different user sites. The absence of safe coordination between these different updates may cause security holes (*i.e.* permitting illegal updates or rejecting legal updates on the shared document). Inspired by the *optimistic security* concept introduced in [30], we propose to use an **optimistic approach** that tolerates momentary violation of access rights but then ensures the copies to be restored in valid states with respect to the stabilized ACP.

The Cassis team will coordinate this task. All the four teams will collaborate to the study of the consistency problem of dynamic local ACPs.

3 Activities Planed

Meetings. The main exchange forum between participants will be coming from regular meetings with scientific presentations, software demonstrations and discussions. Working meetings between participants of two or more sites will be held, involving the members working on common tasks and publications. In addition, we are planning to hold one **global meeting** each year of the project, with all project members, in order to summarize our advances and discuss how to overcome technical difficulties.

Workshops. Every yearly global meeting will also be the opportunity to organize a workshop involving the project members and some international

experts, possibly amongst the ones listed below.

Invitations. We are willing to organize short term visits of international experts for consulting and discussion purposes. Some of these visits will be held at the occasion of the workshop of the global meetings or of working groups. We are planning one invitation for each of the 3 sites and for each year of the project; most of such invitations will be for about one week, and require sufficient travel budgets. Some researchers we might invite include (non exhaustive list): Wenfei Fan, Irini Fundulaki, Leonid Libkin, Maarten Marx, Benjamin Pierce, Maarten Rits, Helmut Seidl, Thomas Schwentick.

Interaction with other projects. We are involved in the following related European and ANR projects and are expecting interactions with their members on the topics of this proposition.

- The **AVANTSSAR** STREP project aims to propose a rigorous technology for the formal specification and "Automated VAlidatioN of Trust and Security of Service-oriented ARchitectures". It involves the team Cassis and 9 other laboratories and companies in Europe.
- The **FoX** STREP project on foundations of XML, leaded by Luc Segoufin, involves the Dahu team and six other teams in Europe.
- The ERC project **Webdam** on the development of develop a formal model for Web data management, leaded by Serge Abiteboul, involves in particular the teams Dahu and GEMO at INRIA Saclay Île-de-France and LRI - Univ. Paris Sud.
- The ANR project **CODEX**, Efficiency, Dynamicity and Composition for XML: Models, Algorithms and Systems, involves members the Mostrare, five other INRIA and French University research labs partners, and one company.
- The ANR project **Enumerations** aims at studying algorithms and complexity of enumerating all solutions of a given problem. It involves members of the Mostrare team and Luc Segoufin from the Dahu team.

Internships. We are asking for support for some internships (one internship per year for each of the 3 sites), typically for master or engineer students. The purpose of the internship may vary according to the case: in some cases, they will consist in some research work on a particular problem, in other cases, they will aim at conducting some experiments on the methods developed by the project.

Web. We will install and maintain a public web site presenting the activities of the project. We will also be using an internal page in order to help the coordination needs of the project; this gForge will run services like

collaborative access to the project files, concurrent version systems, a wiki, task lists, management of mailing lists etc.

4 Results Expected

Publications. All the results obtained in the scope of the project must be made publicly available as soon as possible and in any case at most six months after the end of the project. Dissemination is recognized as the central activity within the project and all partners are contributing to the achievement of this task. It is our intend to present our results publicly in international symposiums and to publish them by any means.

Preparation of Project Proposals. Besides publications, we aim at preparing during the project a proposition of collaboration on the same topics, based on the results obtained during the two years, to be submitted to the EC or the french national research agency (ANR).

5 Amount requested and destination of funds

Missions. We are planing one global meeting and two visits between sites for bilateral interaction on some specific topics for every member of the project, each year. That make a total of 3 missions every year (250 € per mission) for each of the 13 members of the project.

total cost for missions: 19 500 €

Workshops and invitations. We will organize two workshops during the yearly global meetings (2000 € for the organization per workshop) and invite three personalities for around one week every year (1500 € per invitation).

total cost for invitations and workshops: 13 000 €

Internships. We are willing to host one internship per site (3 in total) every year, for a duration of 4 months on average. We are planing, for the whole duration of the project, 3 internships for master students (400 €/month), 1 internship for an engineer student (2000 €/month), and 2 internships held through the INRIA International Internship program (1100 €/month).

total cost for internships: 21 600 €

Overall cost: 54 100 €

References

[1] S. Abiteboul, P. Buneman, and D. Suciu. *Data on the Web: From Relations to Semistructured Data and XML*. Morgan Kaufmann, 1999.

[2] S. Barker and M. Fernández. Term rewriting for access control. In *DBSec*, pages 179–193, 2006.

[3] E. Bertino and E. Ferrari. Secure and selective dissemination of xml documents. *ACM Trans. Inf. Syst. Secur.*, 5(3):290–331, 2002.

[4] A. Bouajjani, B. Jonsson, M. Nilsson, and T. Touili. Regular model checking. In *Proc. of the 12th Int. Conf. on Computer Aided Verification*, volume 1855 of *LNCS*, pages 403–418, 2000.

[5] J. Chabin and P. Réty. Visibly pushdown languages and term rewriting. In *Proceedings 6th International Symposium on Frontiers of Combining Systems (FroCos 2007)*, volume 4720 of *Lecture Notes in Computer Science*, pages 252–266. Springer, 2007.

[6] H. Cirstea. Specifying authentication protocols using rewriting and strategies. In *PADL*, pages 138–152, 2001.

[7] H. Cirstea, P.-E. Moreau, and A. S. de Oliveira. Rewrite based specification of access control policies. *Electr. Notes Theor. Comput. Sci.*, 234:37–54, 2009.

[8] H. Cirstea, P.-E. Moreau, and A. Reilles. Tomml: A rule language for structured data. In *RuleML*, pages 262–271, 2009.

[9] H. Comon and V. Cortier. Tree automata with one memory, set constraints and cryptographic protocols. *Theoretical Computer Science*, 331(1):143–214, Feb. 2005.

[10] H. Comon, M. Dauchet, R. Gilleron, C. Löding, F. Jacquemard, D. Lugiez, S. Tison, and M. Tommasi. Tree automata techniques and applications. Available on: `http://www.grappa.univ-lille3.fr/tata`, 2007. release October, 12th 2007.

[11] H. Comon-Lundh, F. Jacquemard, and N. Perrin. Tree automata with memory, visibility and structural constraints. In H. Seidl, editor, *Proceedings of the 10th International Conference on Foundations of Software Science and Computation Structures (FoSSaCS'07)*, volume 4423 of *Lecture Notes in Computer Science*, pages 168–182. Springer, Mar. 2007.

[12] E. Damiani, S. D. C. di Vimercati, S. Paraboschi, and P. Samarati. A fine-grained access control system for xml documents. *ACM Trans. Inf. Syst. Secur.*, 5(2):169–202, 2002.

[13] D. J. Dougherty, C. Kirchner, H. Kirchner, and A. S. de Oliveira. Modular access control via strategic rewriting. In *ESORICS*, pages 578–593, 2007.

[14] R. Echahed and F. Prost. Security policy in a declarative style. In *PPDP*, pages 153–163, 2005.

[15] C. A. Ellis and S. J. Gibbs. Concurrency Control in Groupware Systems. In *SIGMOD Conference*, volume 18, pages 399–407, 1989.

[16] W. Fan, C.-Y. Chan, and M. Garofalakis. Secure xml querying with security views. In *Proceedings of the 2004 ACM SIGMOD international conference on Management of data (SIGMOD'04)*, pages 587–598, New York, NY, USA, 2004. ACM.

[17] W. Fan, F. Geerts, X. Jia, and A. Kementsietsidis. Smoqe: A system for providing secure access to xml. In *Proceedings of the 32nd International Conference on Very Large Data Bases*, pages 1227–1230. ACM, 2006.

[18] W. Fan, F. Geerts, X. Jia, and A. Kementsietsidis. Rewriting regular xpath queries on xml views. In *Data Engineering, 2007. ICDE 2007. IEEE 23rd International Conference on*, pages 666–675, April 2007.

[19] E. Filiot, J.-M. Talbot, and S. Tison. Tree automata with global constraints. In *12th International Conference in Developments in Language Theory (DLT 2008)*, volume 5257 of *Lecture Notes in Computer Science*, pages 314–326. Springer, 2008.

[20] I. Fundulaki and S. Maneth. Formalizing xml access control for update operations. In *SACMAT '07: Proceedings of the 12th ACM symposium on Access control models and technologies*, pages 169–174, New York, NY, USA, 2007. ACM.

[21] I. Fundulaki and M. Marx. Specifying access control policies for xml documents with xpath. In T. Jaeger and E. Ferrari, editors, *Proceedings 9th ACM Symposium on Access Control Models and Technologies (SACMAT 2004)*, pages 61–69. ACM Press, 2004.

[22] A. Gabillon and E. Bruno. Regulating access to xml documents. In *Proc. of the 15th Annual IFIP WG 11.3 Working Conference on Database Security*, 2001.

[23] B. Groz, S. Staworko, A.-C. Caron, Y. Roos, and S. Tison. Xml security views revisited. In *Database Programming Languages (DBPL)*, 2009.

[24] I. Guessarian. Pushdown tree automata. *Theory of Computing Systems*, 16(1):237–263, 1983.

[25] F. Jacquemard, F. Klay, and C. Vacher. Rigid tree automata. In C. Martín-Vide, editor, *Proceedings of the 3rd International Conference on Language and Automata Theory and Applications (LATA'09)*, Lecture Notes in Computer Science, Tarragona, Spain, Apr. 2009. Springer. To appear.

[26] F. Jacquemard, M. Rusinowitch, and L. Vigneron. Tree Automata with Equality Constraints Modulo Equational Theories. In U. Furbach and N. Shankar, editors, *Proceedings of 3rd International Joint Conference on Automated Reasoning, IJCAR*, volume 4130 of *Lecture Notes in Artificial Intelligence*, pages 557–571, Seattle (WA), August 2006. Springer.

[27] L. Libkin and C. Sirangelo. Reasoning about xml with temporal logics and automata. In *Proceedings of the 15th International Conference on Logic for Programming, Artificial Intelligence, and Reasoning (LPAR)*, volume 5330 of *Lecture Notes in Computer Science*, pages 97–112. Springer, 2008.

[28] M. Murata, A. Tozawa, M. Kudo, and S. Hada. Xml access control using static analysis. *ACM Trans. Inf. Syst. Secur.*, 9(3):292–324, 2006.

[29] OASIS. *OASIS eXtensible Access Control Markup Language (XACML)*, 2003. available at: http://www.oasis-open.org/xacml/docs/.

[30] D. Povey. Optimistic security: a new access control paradigm. In *NSPW '99: Proceedings of the 1999 workshop on New security paradigms*, pages 40–45. ACM, 2000.

[31] N. Rassadko. Policy classes and query rewriting algorithm for xml security views. In *Proceedings of the 20th Annual IFIP WG 11.3 Conference on Data and Applications Security*, volume 4127 of *Lecture Notes in Computer Science*, pages 104–118. Springer, 2006.

[32] T. Schwentick. Automata for xml - a survey. *J. Comput. Syst. Sci.*, 73(3):289–315, 2007.

[33] A. Stoica and C. Farkas. Secure xml views. In *Research Directions in Data and Applications Security, sixteenth IFIP WG 11.3 International Conference on Data and Applications Security*, volume 256 of *IFIP Conference Proceedings*, pages 133–146. Kluwer, 2002.

[34] C. Sun, S. Xia, D. Sun, D. Chen, H. Shen, and W. Cai. Transparent adaptation of single-user applications for multi-user real-time collaboration. *ACM Trans. Comput.-Hum. Interact.*, 13(4):531–582, 2006.

[35] W. Tolone, G.-J. Ahn, T. Pai, and S.-P. Hong. Access control in collaborative systems. *ACM Comput. Surv.*, 37(1):29–41, 2005.