Access-Control Policy Administration in XACML

by Erik Rissanen and Babak Sadighi Firozabadi

In recent years, researchers at SICS have been looking at managing large numbers of access permissions in a dynamic and decentralized network. The main results of our work are a framework and a calculus, called privilege calculus, for access permissions and their administration.

The eXtensible Access Control Markup Language, XACML, is a very effective and now widely adopted standard language for expressing access control policies. The specification of XACML includes the language, its semantics and a framework for making access control decisions based on XACML policies. However, XACML is currently lacking an access control model for the policy itself.

The current XACML model of policy administration puts the access control of policy administration outside the policy model. To control who may edit the policy, mechanisms such as access control at the operating system level must be used. In large distributed systems, such mechanisms may prove difficult to manage. There may be a need to manage the policies in parts of the system not under the control and within the trust of a specific Policy Decision Point, for instance from a mobile device. The rights to change the policy may themselves be highly dynamic. Consequently, there is a need for the policy itself to have an access-control policy model. Our research has been focused on these issues.

The Policy-Based Reasoning group at SICS has for several years been performing research on how best to manage large numbers of access permissions in a dynamic and decentralized network. The main results of our research are a framework and a calculus, called privilege calculus. In the framework, we distinguish between access permissions and administrative permissions, both referred to as privileges. Privilege calculus allows us to reason about privileges and their administration. The core mechanism of privilege calculus is constrained delegation, which allows constraints to be put on the creation of privileges, access permissions or administrative permissions.

Recently, a number of XACML Technical Committee (TC) members have discussed the need for adding administrative support to XACML. The discussed ideas are very similar to the delegation mechanism of privilege calculus. We are now looking into the possibility of extending the current XACML specification and implementing our delegation model in SUN's open-source XACML implementation. Our work will be part of two projects – the TrustCom EU FP6 project and Decentralized Authorization Management in Network-Based Defence – in which we investigate the use of XACML as a policy language for distributed services in highly dynamic and decentralized networks.

Adding delegation to XACML involves defining new forms of policy that can express administrative rights, and a new processing model that can verify that delegations have been performed in an authorized manner. The new features of XACML help users to implement flexible decentralized access control management, for instance in the setting up of a large organization or joint business venture. This will reduce the administration costs of the organizations and make them more flexible. Having these features available in a standard access control language will make their use simpler and more widely adopted.

Please contact:
Babak Sadighi Firozabadi or Erik Rissanen,
SICS, Sweden
Tel: +46 8 633 1500
E-mail: babak@sics.se, mirty@sics.se

Contributions of Team Automata in Security

by Maurice ter Beek, Gabriele Lenzini, and Marinella Petrocchi

Researchers from two CNR Institutes in Pisa are studying ways in which a formal model of team automata can be exploited to specify and analyse security-related issues.

The Information Security group at the Institute for Informatics and Telematics (IIT-CNR) has both practical and theoretical experience with many aspects of security. The Formal Methods and Tools (FM&&T) group at the Institute of Information Science and Technologies (ISTI-CNR) has experience in research on formal methods for the specification, design and verification of computer systems. Recently, researchers from these two groups teamed up to investigate how a formal model of team automata can contribute to the specification and analysis of security issues. This cooperation will be continued in the context of an EU-funded project on Software Engineering for Service-Oriented Overlay Computers (SENSORIA).

Team automata form a mathematical framework introduced in 1997 by C.A. Ellis to model components of groupware systems and their interconnections. Their

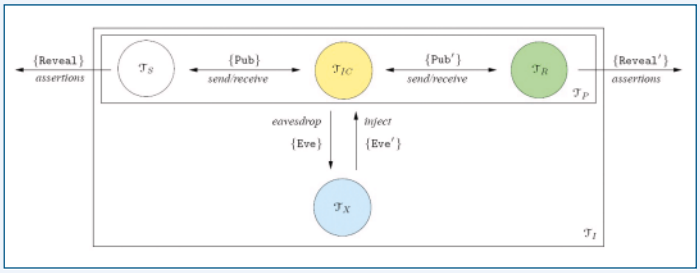
usefulness however extends to modelling collaborations between system components in general (for an overview, see [TA]).

A team automaton is composed of component automata that distinguish input, output and internal actions. Input actions are not under the automaton's control, but are triggered by the environment, which can include other automata. Output and internal actions are under its control, but only the output actions are observable by other automata. Input and output actions together constitute the external actions and form the interaction interface between the automaton and its environment; internal actions do not participate in any interactions.

In composing a team automaton, the crux is to define the way in which those originally independent components interact. Their interactions are formulated in terms of synchronizations of shared actions, a method for modelling collaboration among system components that is well known in the literature. A component automaton does not necessarily participate in every synchronization of an action it shares. Hence there is no such thing as the unique team automaton over a set of component automata. Rather, a whole range of team automata, distinguishable only by their transition relation, can be constructed from a given set of components. It is this freedom to choose a transition relation that sets the team automata framework apart from most other automata-based models, most of which use a single and very strict method for choosing the transition relation of an automaton composed over a set of automata - in effect resulting in composite automata that are uniquely defined by their constituents.

In a series of papers (see [TA]) we have shown how team automata can adequately be used to model (and sometimes verify) various access control policies, multicast/broadcast communication protocols and general (cryptographic) communication protocols.

To begin with, we have demonstrated the model usage and utility for capturing information security and protection structures, as well as critical coordinations between these structures. On the basis of a spatial access metaphor, various known access-control strategies have been given a rigorous formal description in terms of synchronizations in team automata. Moreover, we have initiated to validate some of the resulting specifications with the model checker Spin.



An insecure communication scenario for team automata.

Later we have initiated the use of team automata for the security analysis of multicast and broadcast communication. For this purpose, we have performed a case study in which team automata were used to model an instance of a particular stream signature protocol. The one-to-many and one-to-all communications, which are so typical of multicast and broadcast communications, were captured by team automata in a native way as synchronizations between the set of component automata constituting a team automaton. We have also developed a framework for security analysis with team automata, which has required three basic formal steps.

First, we defined an insecure communication scenario based on the addition of a so-called 'most general intruder' to a team automaton model of a secure communication protocol. The intruder was modelled as an active agent able to influence communication among honest agents. This insecure scenario can be used to analyse some security properties of cryptographic communication protocols involving two roles – an initiator and a responder. Rather than occurring directly, all communication is assumed to flow through an insecure channel. This insecure channel may release some messages to an intruder, which in its turn can either listen to or modify (fake) the messages passing through this channel. When verifying security properties for cryptographic communication protocols, it is indeed quite common to include an additional Dolev-Yao-style intruder that is supposed to be malicious and whose aim is to subvert the protocol's correct behaviour. A protocol specification is consequently considered secure with reference to a security property if it satisfies this property despite the presence of the intruder. Abstracting from the cryptographic details concerning the operations according to which messages can be encrypted, decrypted, etc, the insecure scenario is informally described by the team automata interactions sketched in the figure.

Second, a well-established theory for defining and verifying a variety of security properties was reformulated in terms of team automata and subsequently, a compositional analysis strategy was described for it. Under appropriate assumptions, this can be used to verify some security properties in the communication protocol modelled by the scenario.

Third, this framework was applied to show that integrity is guaranteed for the particular setting of the case study. This shows the effectiveness of our approach for a realistic stream signature protocol, thus facilitating an easy comparison for those familiar with other approaches. In fact, an approach that uses an automata-based formalism for the specification and verification of properties in the field of security is not unique, but has become very popular in recent years.

Finally, very recently, team automata have been used to model and verify a protocol aiming at privacy in communication among mobile agents. This was the first attempt to use team automata for the analysis of privacy properties.

Links:
[TA]: http://fmt.isti.cnr.it/~mtbeek/TA.html
FM&&T: http://fmt.isti.cnr.it/
IIT: http://www.iit.cnr.it/

Please contact:
Maurice H. ter Beek, ISTI-CNR, Italy
Tel: +39 050 315 3471
E-mail: maurice.terbeek@isti.cnr.it