2008 IEEE International Conference on Technologies for Homeland Security

SFINKS: Secure Focused Information, News, and Knowledge Sharing

Tatyana Ryutov, tryutov@isi.edu, (310) 448-9111
Tatiana Kichkaylo, tatiana@isi.edu, (310) 448-8435
Robert Neches, rneches@isi.edu, (310) 448-8481
Michael Orosz, mdorosz@isi.edu, (310) 448-8266

USC Information Sciences Institute
4676 Admiralty Way
Marina del Rey, California 90292-6601
FAX: (310) 822-6593

Abstract *-- Cross-agency collaboration and sharing of digital data is critical to respond to or prevent threats to U.S. interests. While traditional hierarchical information sharing approaches ensure that only relevant information is delivered to authorized nodes, the resulting organizational overhead severely impedes timely sharing of critical information. Although alternative approaches to secure data release have previously been proposed, they all have had severe practical limitations.*

We are developing SFINKS – a flexible collaboration platform that enables secure and focused information sharing across organizations. SFINKS uses two key technologies developed at ISI to support a new concept of fine-grained semantically controlled information visibility. The iLands infrastructure provides a semantic network-based data model, search and filtering capabilities, distributed systems support and fine-grained control of resource visibility. The Adaptive Trust Negotiation and Access Control (ATNAC) provides flexible access control and trust management.

SFINKS solves many typical problems plaguing information sharing in coalitions:

1. *Supporting security requirements of multiple autonomous peer authorities without a single "root".*
2. *Integrating on-the-fly dynamically generated data, temporary/dynamic users, and interaction context.*
3. *Allowing on-demand coalition formation by negotiating security requirements of participants.*

Initial SFINKS implementation is deployed within Risk Analysis Workbench – a collaborative information sharing environment.

1. INTRODUCTION

The ability to collaborate by sharing digital data among federated organizations is critical for supporting homeland security applications. Information needs to flow between government agencies, contractors, national labs, and DHS Centers of Excellence like USC's Center for Risk and Economic Analysis of Terrorism Events (CREATE) and National Center on Foreign Animal and Zoonotic Diseases (FAZD). Direct communication between organizations, when authorized, enables faster and more accurate access to information critical in detecting or responding to events. Unfortunately, existing security models force a choice between two bad alternatives: either impose hierarchical controls that severely hamper rapid information sharing when urgent needs arise, or leave information accessible beyond prudent confidentiality and sensitivity bounds. Although many alternative approaches to secure data release have been developed over last thirty years, they all have had severe practical limitations. What is needed is a way to expedite information flows with the speed of direct communication without losing fine grained hierarchical control. Emerging technologies and new types of intelligence data make it necessary to revise existing approaches to the information release problem.

SFINKS addresses the following information sharing problems:

1. Interaction of multiple peer authorities requires horizontal integration of independently defined security requirements and evaluation and enforcement methods.

2. Dynamic environments require support for both dynamic entities, such as data generated on-the-fly, and situation-dependent access control policies.

3. Coalitions should be created and disbanded on demand by negotiating participants' (possibly sensitive) security requirements. Dissolution of coalitions should involve revocation of access rights and removal of shared resources.

4. The dynamic nature of the information sharing environment requires efficient storage and processing of security requirements.

5. A comprehensive user interface is needed to allow end-users to understand existing access policies and specify the desired security requirements.

2. RELATED WORK

Qin and Atluri [9] introduced concept-level semantic access control model which considers some semantic relationships supported by the ontology. However, the authors do not define how the instances are associated with the ontological concepts and how to enforce access control requirements on these concepts. SFINKS supports grouping of resources into DAGs and allows attaching policies to the DAG nodes. Ellison and Dohrmann [3] presented Next Generation Collaboration – security architecture with limited flexibility: new members can only be added by invitation issued by authorized members. This means that in general only parties known to the core users could participate. SFINKS supports on-the-fly creation of groups based on dynamic semantic properties of users. Nita-Rotaru and Li [8] presented a framework for role-based access control in group communication systems. This approach combines role-based access controls with environment parameters (e.g., time and IP address). SFINKS provides an additional flexibility by supporting semantic properties of users. Li and Mitchell [6] presented a role-based trust management framework which provides policy language and deduction engine suitable for attribute based access control. The framework is defined at a high level and lacks implementation details. Cross domain SFINKS operation is based on automated trust negotiation. Some of the trust negotiation systems and languages developed in recent years include PeerTrust [7] , SD3 [4] , Cassandra [1] , and χ-TNL [2] .

3. SFINKS OVERVIEW

SFINKS is based on technologies developed at ISI. It utilizes iLands infrastructure originally developed for the Risk Analysis Workbench (RAW)[1] project. In particular, iLands includes a semantic network-based data model, search and filtering capabilities, distributed systems support (multiple clients and servers), and fine-grained control of resource visibility. iLands enables dynamic, adaptive data navigation, subscription and distribution. iLands serves as a platform for large-scale collaboration among dynamic coalitions.

We also use the Adaptive Trust Negotiation and Access Control (ATNAC) framework [10] [11] based on two well established systems: TrustBuilder [15] and Generic Authorization and Access control API[2] [12] . ATNAC provides trust negotiation and adaptive access control capturing dynamically changing security and situational requirements. ATNAC's flexible rule composition modes (hierarchical and horizontal) allow several authorities to have fine-grained control over information. By building upon these technologies, SFINKS provides flexible and secure collaboration platform for sensitive environments.

Figure 1 shows a high level view of SFINKS. *Visibility of data* in SFINKS is controlled at fine-grained level due to data representation based on semantic network [13] . Sensitive resources and policies can be dynamically made visible to different users depending on the situation without any changes in the structure of the data or organizational hierarchy, thus supporting dynamic coalition formation.

[1] RAW is used by DHS Centers of Excellence such as CREATE and FAZD - National Center for Foreign Animal and Zoonotic Diseases.

[2] Developed for DARPA's Dynamic Policy Evaluation for Containing Network Attacks (DEFCN) program.

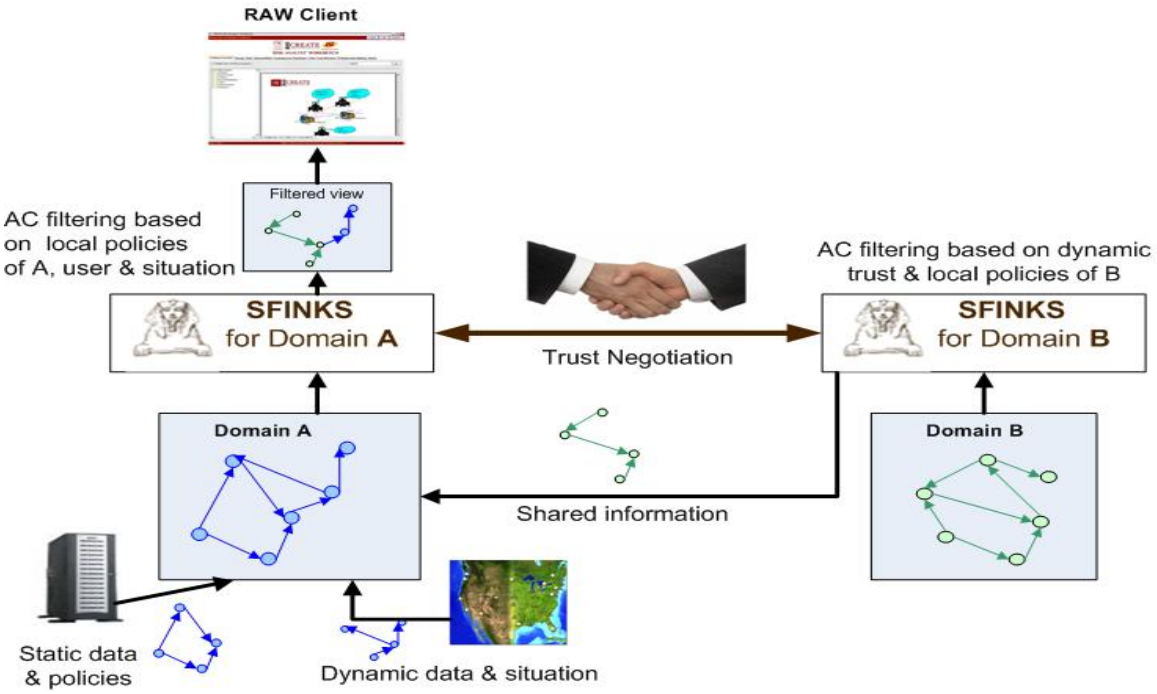

Figure 1. Overview of SFINKS system

For example, the fact that a certain expert works on a given project may be visible to some users but not others.

While it is possible to use existing technologies to assign permissions to each data element, this approach is hard to manage for large data sets or apply to dynamically generated information. Instead, SFINKS computes permissions based on dynamic semantic properties of the user, the data, and the situation. SFINKS supports highly dynamic *Access Control (AC) filtering*. Not only can permissions be computed for dynamic data and users, but also the effective resource visibility can be changed depending on the situation (e.g., threat of a terrorist attack) or current dynamic level of trust between cooperating organizations. SFINKS employs *Trust Negotiation* to enable negotiation of mutually acceptable security requirements.

Complexity of policy specification languages often hinders adoption of tools and causes users to make mistakes that hamper security. To address this problem, SFINKS provides intuitive *User Interfaces* for visualization of security policies that govern resources, allowing the user to readily assess effect of a policy on visibility of resources to various groups of users.

4. SFINKS: CURRENT STATUS

Initial SFINKS implementation is deployed within RAW, an iLands-based collaborative information sharing tool developed by CREATE - USC's Center for Risk and Economic Analysis of Terrorism Events. RAW users can share resources (documents, models) with other users and locate information using semantic tags and links. SFINKS serves as the access control system of RAW. The initial implementation of SFINKS is limited to single domain and does not support dynamic resources. The rest of this section describes architecture of the SFINKS access control module, policy model, and user interface.

4.1 SFINKS Access Control (AC)

Figure 2 illustrates SFINKS components involved in making access control decisions and answering access permission queries. Currently implemented components are represented by objects outlined with solid lines.

Access Control Object (ACO) provides a single, uniform interface to other services that constitute SFINKS AC system. The RAW application interacts with the AC system through ACO.

Policy Retrieval Service (PRS) collects all per resource policies and forwards them to *Policy*

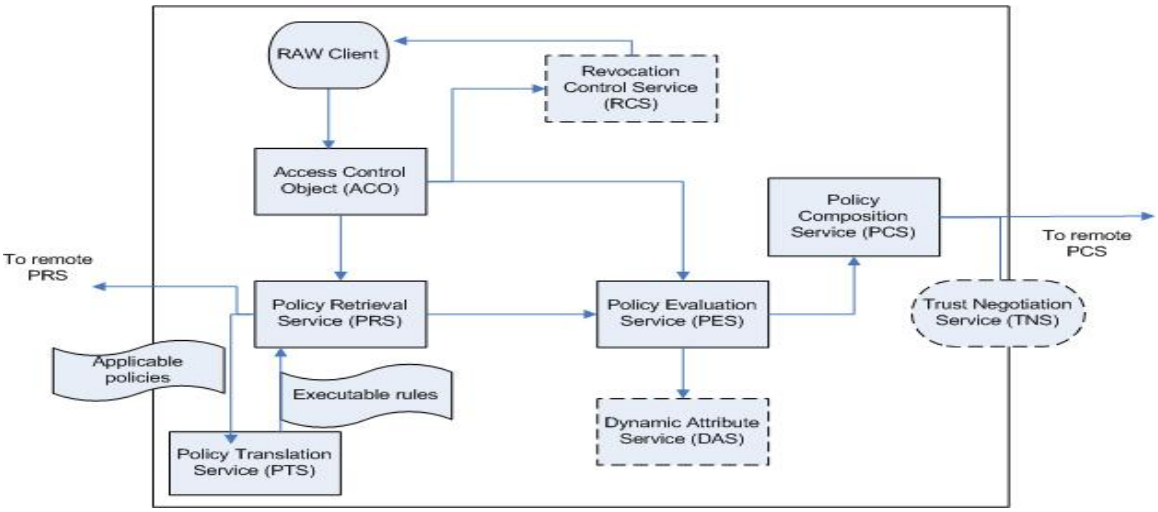

Figure 2. SFINKS Access Control deployed within a local domain

Translation Service. In later versions of the system, some policies can be stored remotely; therefore, interaction between PRSs located in different domains can be required.

Policy Translation Service (PTS) performs authorization propagation and translation of user-level policies to low level rules. Policy separation and automatic authorization propagation enables improved performance while minimizing storage and policy specification overhead. Flexible user-level policy specification in the limited resource visibility situations requires expressive representations, which are hard to compute. SFINKS addresses this expressiveness-performance tradeoff by dynamically compiling user-level security requirements into low-level executable rules using PTS.

Policy Evaluation Service (PES) evaluates translated access policies for a given object. However, policies associated with an object are not necessarily evaluated by a single PES object.

Policy Composition Service (PCS) relates policies defined by different authorities and resolves conflicts, thus supporting coalition environment.

4.2 Policies

SFINKS supports grouping of users (*groups*) and resources (*bundles*); and organizing them into hierarchical structures – DAGs (shown in Figure 3). A policy can be attached to an object (user, group, bundle or resource). Policy specifies groups of users and individual users, a set of positive and/or negative permissions of different types (e.g., read, write, view, execute, etc.) and optional context conditions. In addition a policy may specify a "non propagation" rule and policy composition rules.

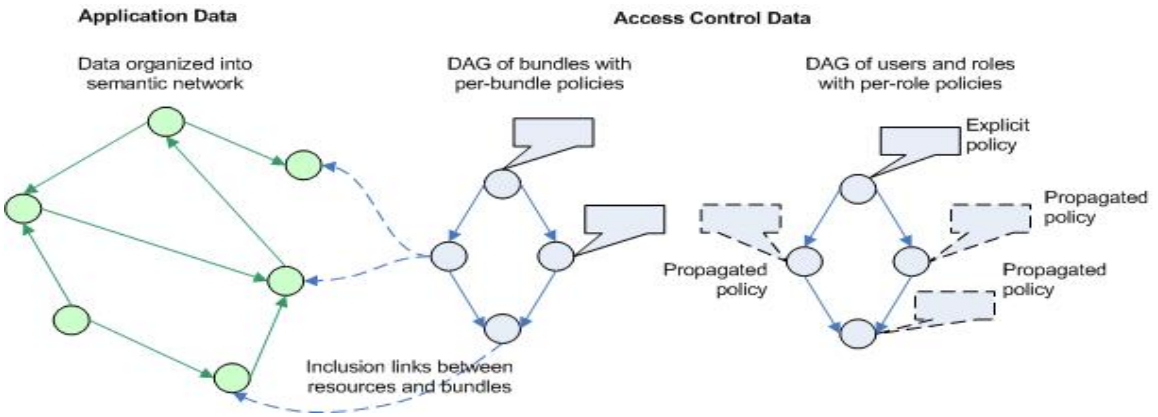

Figure 3. DAGs of users and bundles with attached explicit and propagated policies

Policies can be *explicit* or *propagated*. Explicit policies are created using a graphical user interface and are directly attached to an object. Figure 4 shows a policy editor which allows one to add users, specify positive and negative permissions, enable/disable policy propagation. SFINKS supports permission propagation along the DAGs. Propagation simplifies the access control management since less explicit authorizations are required (no need to attach a policy to each object in the system). Ability to express a single access control policy that applies to the entire set of nodes in a DAG, rather than having to specify a separate policy for each node, increases both ease of use and the likelihood that the policy will correctly reflect the desired permission structure.

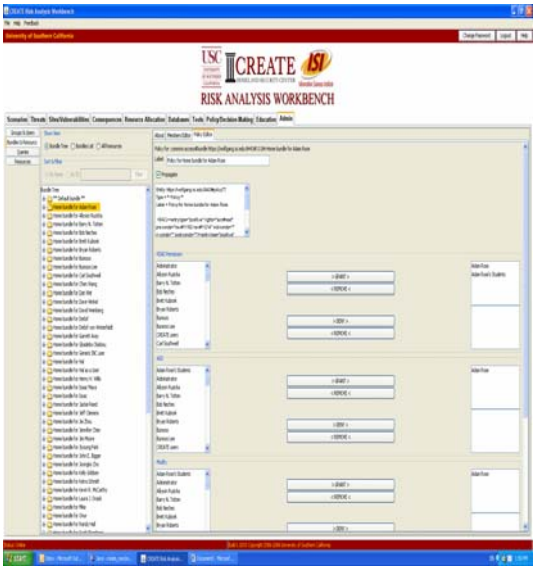

Figure 4. Policy editor interface

4.3. Permission Queries

To enable horizontal integration of multiple authorities and to support dynamic resources and users, SFINKS represents access control policies in terms of groups and bundles, each of which can be specified using semantic expressions (as opposed to explicit sets of static resources). While addressing important challenges, this approach requires innovative user interfaces to help users assess effective permissions and help them design good policies that would have the intended effect.

Figures 4 and 5 show our initial implementation of access control editor and viewer for the RAW application. The viewer allows a user X to quickly determine which part of the resources visible to X will be accessible by user Y, assuming user X is allowed to know about existence of user Y.

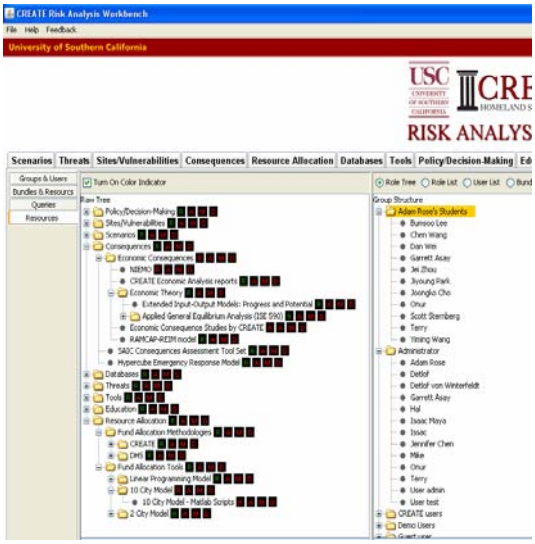

Figure 5. Permission query interface

5. SFINKS: FUTURE WORK

Our future work includes extending SFINKS to support authorizations based on dynamic groups and resources. This will enable mapping dynamic/temporary users and generated resources to locally defined groups and bundles (with statically attached permissions) based on the verified user and resource attributes and the context of the interaction. We will implement SFINKS' *Dynamic Attribute Service (DAS)* which is responsible for run-time computation of required attributes. SFINKS' extendable rule base will allow data providers to describe rules that map users to groups in terms of task (need to know), security clearance, system threat status, communication parameters, location, etc. Similarly, on-the-fly generated resources will be mapped to a particular bundle with attached policies based on the source, sensitivity, semantic content, ownership, integrity, etc.

We will implement the *Trust Negotiation Service (TNS)* to support cross domain policy retrieval & evaluation. We will use Trustbuilder2 (http://dais.cs.uiuc.edu/dais/security/tb2/) system.

We plan to implement the *Revocation Control Service (RCS)* that will maintain information

about sensitive released resources (who accessed which resources and when) and forces revocation of the resources when authorizations expire.

6. CONCLUSIONS

SFINKS delivers a reference implementation of a controlled sharing and collaboration framework, incrementally adding functionality and testing it within the RAW system. As SFINKS becomes increasingly proven in that setting, it can quickly transition to other sensitive coalition environments within and across government and industry. Wider adoption of the technology will provide federal agencies the tools necessary to expedite information flows with the speed of direct communication without losing fine grained hierarchical control. We will also continue our work on user interfaces for viewing and editing access control policies

REFERENCES

[1] M. Y. Becker and P. Sewell, Cassandra: Distributed access control policies with tunable expressiveness, in *Proc. IEEE 5th International Workshop on Policies for Distributed Systems and Networks*, pp. 159–168, 2004.

[2] E. Bertino, E. Ferrari, and A. Squicciarini, χ-TNL: An XML-based language for trust negotiation, in *Fourth IEEE International Workshop on Policies for Distributed Systems and Networks*, pp. 81–84, June 2003.

[3] C. Ellison and S. Dohrmann, Public-key Support for Group Collaboration. *ACM Trans. Information System Security, 6,* 547–565, 2003.

[4] T. Jim, Sd3: A trust management system with certified evaluation, in *Proc. IEEE Symposium on Security and Privacy*, pp. 106–115, 2001.

[5] F. Lehmann, ed. Semantic Networks in Artificial Intelligence, *Computers and Mathematics with Applications* 23:6-9. 1992.

[6] N. Li and J. C. Mitchell, RT: A Role-based Trust-management Framework. In *Proceedings of the Third DARPA Information Survivability Conference and Exposition* IEEE Computer Society Press, pp. 201–212, 2003.

[7] W. Nejdl, D. Olmedilla, and M. Winslett, Peertrust: Automated trust negotiation for peers on the semantic web, in *Proc. of the Workshop* on Secure Data Management in a Connected World , 2004.

[8] C. Nita-Rotaru and N. A. Li, Framework for Role-Based Access Control in Group Communication Systems. In *Proc. of International Workshop on Security in Parallel and Distributed Systems*, 2004.

[9] L. Qin and V. Atluri. Concept-level access control for the semantic web. In *Proc. Workshop on XML Security*, 2003.

[10] T. Ryutov, L. Zhou, L., C. Neuman, T. Leithead, K. and Seamons, Adaptive trust negotiation and access control, in *Proc. of ACM Symposium on Access Control Models and Technologies*, 2005.

[11] T. Ryutov, L. Zhou, N Foukia, C Neuman, T. Leithead, K. Seamons. Adaptive Trust Negotiation and Access Control for Grids, In *Proc. of the 6th IEEE/ACM International Workshop on Grid Computing*, 2005.

[12] T. Ryutov, C. Neuman, and D. Kim, Dynamic authorization and intrusion response in distributed systems, in *Proc. 3rd DARPA Information Survivability Conf. and Exposition (DISCEX III)*, pp. 50–61, 2003.

[13] J. F. Sowa, ed. Principles of Semantic Networks: Explorations in the Representation of Knowledge, *Morgan Kaufmann Publishers*, 1991.

[14] W. Winsborough, K. Seamons, and V. Jones, Automated trust negotiation, in *Proc. 1st DARPA Info. Survivability Conf. and Exposition (DISCEX I)*, pp. 88–102, 2000.

[15] M. Winslett, T. Yu, K. E. Seamons, A. Hess, J. Jacobson, R. Jarvis, B. Smith, and L. Yu, Negotiating trust on the web, *IEEE Internet Computing*, vol. 6, no. 6, pp. 30–37, 2002