Modeling Data Flow in Socio-Information Networks: A Risk Estimation Approach

Ting Wang[†] Mudhakar Srivatsa[‡] Dakshi Agrawal[‡] Ling Liu[†]
[†]College of Computing, Georgia Institute of Technology, Atlanta, GA
[‡]IBM T.J. Watson Research Center, Hawthorne, NY
[†]{twang, lingliu}@cc.gatech.edu [‡]{msrivats, agrawal}@us.ibm.com

ABSTRACT

Information leakage via the networks formed by subjects (e.g., Facebook, Twitter) and objects (e.g., blogosphere) – some of whom may be controlled by malicious insiders – often leads to unpredicted access control risks. While it may be impossible to precisely quantify information flows between two entities (e.g., two friends in a social network), this paper presents a first attempt towards leveraging recent advances in modeling socio-information networks to develop a statistical risk estimation paradigm for quantifying such insider threats. In the context of socio-information networks, our models estimate the following likelihoods: *prior flow* – has a subject s acquired covert access to object o via the networks? *posterior flow* – if s is granted access to o, what is its impact on information flows between subject s' and object o'? *network evolution* – how will a newly created social relationship between s and s' influence current risk estimates? Our goal is not to prescribe a one-size-fits-all solution; instead we develop a set of composable network-centric risk estimation operators, with implementations configurable to concrete socio-information networks. The efficacy of our solutions is empirically evaluated using real-life datasets collected from the IBM SmallBlue project and Twitter.

Categories and Subject Descriptors

K.6.5 [**Computing Milieux**]: Management of Computing and Information Systems—*Security and Protection*

General Terms

Security, Management

Keywords

Risk estimation, social network, access control

1. INTRODUCTION

The ever-increasing complexity and dynamics of information sharing infrastructures have presented grand challenges for today's access control mechanisms. One key issue is the information leakage via the complex and dynamic networks formed by subjects and objects (e.g., social network and blogosphere): subject s who has access to object o may (un)intentionally leak it to a socially linked subject s', or object o' derived from o may reveal information about o.

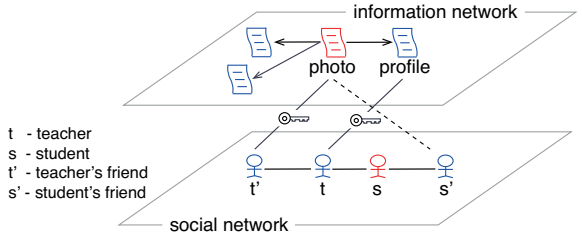

Figure 1: Networks of subjects and objects.

Example 1. In [2], a high school teacher had to resign since she was tagged in a friend's photo with her holding a wine glass, which was then visible via her Facebook profile to her students. As illustrated in Figure 1, this incident is essentially caused by an access channel in the socio-information network: "student → teacher → teacher's profile → photo".

Indeed information leakage via such socio-information networks often leads to unpredicted access control risks – more so in the presence of malicious insiders within such networks. While it may be impossible to precisely quantify information flows between two entities (e.g., two friends in a social network), this paper leverages statistical models of information flows in such networks (such as those proposed in [4, 16, 25, 8]) to develop a risk estimation paradigm for quantifying such insider threats. In the context of socio-information networks, our models estimate the following likelihoods:

- *prior flow* estimation: how likely is it for s to have acquired covert access to o (or a *fraction* of o) via the channels formed by relevant subjects N_s and objects N_o?
- *posterior flow* estimation: how would granting s access to o (denoted by $s \to o$) potentially affect the information flows with respect to N_s and N_o?

Loosely speaking, the risk of granting access ($s \to o$) is high if it would significantly increase (= *posterior flow* – *prior flow*) the information flow between certain subject $s' \in N_s$ and object $o' \in N_o$. Our goal is to develop risk estimation operators that capture such *network effects*, namely, that one access granting decision ($s \to o$) has on several other related subjects and objects ($s' \to o'$) in a socio-information network. To our best knowledge this work also represents the first attempt to study the impact of network effects (in socio-information networks) in Risk-based Access Control, an emerging security paradigm [10, 20].

SACMAT'11, June 15–17, 2011, Innsbruck, Austria.

The key contributions of our paper are as follows. First, we develop a set of composable network-centric risk estimation operators, with implementations configurable to concrete socio-information networks. To make the estimation practical, we further refine our estimates by taking account of factors including the evolution of networks and the incompleteness or uncertainty in network information. Second, we show that a range of state-of-the-art access control models can be enhanced by our risk estimation paradigm, typically by encoding the policy-enforced information flows as weighted links in the socio-information network. We note that our approach does not replace classical access control models; instead it addresses an orthogonal problem and augments classical access control models with risk assessments in the context of socio-information networks. Third, we develop a suite of scalable algorithms for implementing the operators on large networks.

The efficacy of our solution is empirically evaluated using real socio-informatics network datasets of varied network scale, collected from Twitter[1] (200K subjects) and the IBM SmallBlue project [18] (40K subjects). Specifically, on Twitter an user s_i that *follows* another user s_j may *re-tweet* (re-post) the messages sent by s_j to its (s_i's) followers, which can be considered as one type of leakage (though not necessarily unwanted information flow). We show that our model is able to effectively capture such leakage behavior. Further, using the multiple snapshots (separated by six month time interval) of socio-information network in the Smallblue dataset, we show how our solution is able to incorporate the inherent dynamic aspects of social-information network, such as evolution, granted/revoked accesses, etc.

The remainder of this paper is organized as follows. Section 2 formalizes the mathematical model of our risk estimation paradigm; the library of composable risk estimation operators is introduced in Section 3; Section 4 details scalable implementation of these operators followed by experimental evaluation in Section 5; Section 6 concludes the paper.

2. NETWORK-CENTRIC RISK ESTIMATION

2.1 Basic Model

In network-centric risk estimation, a multi-layer network model is used to capture subject-subject, object-object, and subject-object relationships, wherein relationships are encoded as intra-network or inter-network links (see Figure 1). Specifically, the interconnected objects form an *object network* (or information network), $G_O = (O, L_O)$, with nodes O and links L_O representing the set of objects and their relationships. Analogously, a *subject network* (or social network), $G_S = (S, L_S)$, captures the relationships between subjects, where S and L_S denote the set of subjects and their connections. Further, a collection of inter-network links L_I between the subject and object networks encode their interactions (e.g., access and leakage history).

We apply a generic information flow model to quantify the qualification of a subject to access an object, wherein information is viewed as fluid that flows along links in socio-information networks. The dynamics of fluid flow may be different (e.g., gossips in subject-subject links, database triggers in object-object links, access history in subject-object links, etc.). The weight on a (directed) network link is indicative of the propensity of information flow along the link. Our goal is to regulate the flow of information between subjects and objects by controlling subject-object links, namely, subject-object links (L_I) created as a consequence of granting access requests. More specifically, in our information flow model, each intra-network or inter-network link is associated with two attributes:

- *Enforced flow capacity*, $\texttt{enf}(\cdot)$, that specifies flows permitted by an underlying access control model. It is typically encoded as either "1" or "0", with "1" indicating that information is free to flow (e.g., from low security to high security, or from high integrity to low integrity), and "0" indicating that no information flow is permitted. For instance, two subjects s and s' have $\texttt{enf}(\overline{ss'}) = 1$ if the role of s' dominates s (e.g., *employee* $\prec$ *manager*); two objects o and o' have $\texttt{enf}(\overline{oo'}) = \texttt{enf}(\overline{o'o}) = 1$ if they are not "mutually exclusive" in Chinese-wall policy. In Appendix A we show that a wide range of conventional access control models may be emulated by our model by suitable parameterization (e.g., setting link weights).
- *Leakage flow capacity*, $\texttt{leak}(\cdot)$, that specifies flows introduced by potential information leakage via networked subjects and objects. It is typically a real number within the interval $[0, 1]$. For object network, $\texttt{leak}(\overline{oo'})$ may specify the fraction of information object o that is inferable from o' (called *residual information*), which can usually be measured using information-theoretic metrics, e.g., Kullback-Leibler divergence; while for subject network, $\texttt{leak}(\overline{ss'})$ may be interpreted as the likelihood that s leaks (intentionally or unawarely) information to s'. In general, such leakage likelihood may be discriminative with respect to specific objects and subjects under consideration, i.e., a function of object metadata [23] and the corresponding social relationship.

Given access request $(s \rightarrow o)$, our network-centric risk estimation gauges the overall enforced flow $f_e(o \rightarrow s)$ (details in Appendix A) and leakage flow $f_c(o \rightarrow s)$ (details in Section 4) from object o to subject s. More specifically, $f_e(o \rightarrow s) = 1$ if the flow from o to s is allowed by the access control model, and 0 otherwise; while $f_c(o \rightarrow s)$ comprises two estimates: f_c^{frac}, the fraction of information of o leaked to s, and f_c^{like}, the likelihood of such leakage.

2.2 Access Request Evaluation

In our risk estimation paradigm, the risk of an access request is no longer solely based on the requesting subject and the concerned object; rather, within the context of socio-information network, it depends on (i) relevant subjects and objects, and (ii) their profound network influence before (and after) approving this access. Towards this end, we propose two fundamental operations, *prior*-flow and *posterior*-flow estimation, as the foundation of access risk evaluation.

Prior-Flow Estimation

Given a new request $(s \rightarrow o)$, we first evaluate the existing enforced and leakage flows $f_e(o \rightarrow s)$ and $f_c(o \rightarrow s)$. Conceivably, if $f_e(o \rightarrow s) = 1$, i.e., the flow is allowed by the access control policies, this access should be granted. Meanwhile, when the flow is disabled by the policies (i.e., $f_e(o \rightarrow s) = 0$), but the leakage flow is significant, i.e., $f_c^{frac}(o \rightarrow s) \geq \epsilon$ and $f_c^{like}(o \rightarrow s) \geq \delta$ (ϵ and δ are threshold

[1] http://www.twitter.com

parameters)[2], it may make limited sense to impose strict control over s to access o. Intuitively, in addition to evaluating the qualification of s with respect to o, prior-flow estimation provides an important risk-based exception handling mechanism. Formally, request $(s \to o)$ is granted only if the following condition is met (necessary condition):

$$(i) f_e(o \to s) = 1 \vee f_c(o \to s) \geq \epsilon$$

Example 2. In Figure 1, on evaluating request (student → photo), one may notice that teacher has possession of profile (as indicated by the inter-network link $\overline{\text{profile} - \text{teacher}}$), which semantically refers to photo (as indicated by the reference relationship $\overline{\text{photo} - \text{profile}}$). The leakage flow photo → profile → teacher → student may carry sufficient information for student to completely infer photo, which makes simple comparison of classification level class(photo) and clearance level clear(student) non-informative.

Posterior-Flow Estimation

While prior-flow captures existing information flows before an access $(s \to o)$ is granted, posterior-flow estimation evaluates how the access once granted would impact the information flows for relevant subjects and objects. Essentially, posterior-flow estimation measures the potential risk of approving an access request. Let $f'_e(\cdot)$ and $f'_c(\cdot)$ be the enforced and leakage flows[3], after an inter-network link $\overline{so}$ is created. If the posterior enforced flow invalidates a previously approved access, or the posterior leakage flow enables a previously disabled access, a violation is raised. Formally,

$$(ii) \nexists\, o' \in O, s' \in S, \text{ s.t. } f_e(o' \to s') = 1 \wedge f'_e(o' \to s') = 0$$

$$(iii) \nexists\, o' \in O, s' \in S, \text{ s.t. } \begin{cases} f_e(o' \to s') = 0 \\ f_c(o' \to s') < \epsilon \\ f'_c(o' \to s') \geq \epsilon \end{cases}$$

Here condition (ii) dictates that the requested access should not invalidate any previously (potentially) approved access $(s' \to o')$; while condition (iii) states that it should not increase leakage flow capacity beyond the threshold ϵ.

Example 3. Recall the example in Figure 1. The approval of access (student's friend → photo) may significantly change the leakage flow f'_c(photo → student), given the close relationship $\overline{\text{student} - \text{student's friend}}$. If f_e(photo → student) = 0 and f_c(photo → student) < ϵ, this increased leakage may result in a violation of existing access control policy.

To summarize, conditions (i), (ii), and (iii) together form the sufficient condition for granting an access request: if significant prior leakage flow exists, request $(s \to o)$ may be granted even if the security level of s is inadequate to access o; meanwhile, if access $(s \to o)$ incurs the risk of information leakage that violates access control policies, $(s \to o)$ may not be approved even if the security level of s is sufficient to access o. It is worth noting that as more accesses are granted, the average leakage flow (i.e., risk) in the networks tends to increase; after the overall risk reaches certain "frozen point", no more accesses would be granted. Fortunately, the value of tactical information typically decays over time [22]; it is thus possible to incorporate such time sensitivity to maintain the system operability, which we consider as one ongoing research direction.

[2]In following, we assume that the leakage likelihood threshold δ is fixed, and use $f_c(\cdot)$ to denote $f_c^{frac}(\cdot)$ if $f_c^{like}(\cdot) \geq \delta$, and 0 otherwise.

[3]For dynamic models (e.g., Chinese-wall, History-based Access Control models (HBAC)), granted accesses may change enforced flows.

op.	input	output
D	object o	objects depending on o
U	object o	objects depended by o
J	object o, objects N_o	overall information of o at N_o
S	subject s, threshold δ	subjects with leakage to s of likelihood above δ
T	subject s, threshold δ	subjects with leakage from s of likelihood above δ
A	subjects s	subjects connected with s
X	subjects N_s	objects N_o accessed by N_s

Table 1: List of atom operators.

operation	algebra
estimate prior flow	$\mathtt{J}(o, \mathtt{X} \cdot \mathtt{S}(s, \kappa))$
estimate posterior flow	$\triangleright s' \in \mathtt{T}(s, \kappa), \triangleright o' \in \mathtt{U}(o) : \mathtt{J}(o', \mathtt{X} \cdot \mathtt{S}(s', \kappa))$
add new object	$\triangleright o' \in \mathtt{U}(o) : \mathtt{J}(o', \{o\})$
update existing object	$\triangleright o' \in \mathtt{U}(o), \triangleright o'' \in \mathtt{D}(o) : \mathtt{J}(o', \{o''\})$
add new subject link	$\triangleright s' \in \mathtt{A}(s), \triangleright o \in \mathtt{X} \cdot \mathtt{A}(s) : \mathtt{J}(o, \mathtt{X} \cdot \mathtt{S}(s', \kappa))$

Table 2: Risk estimation algebra.

3. RISK ESTIMATION ALGEBRA

In this section we describe the realization of our risk estimation paradigm. Conceptually, we construct an expressive algebra framework (e.g., estimating leakage flow, updating existing network, and predicting network evolution) by composing a library of fundamental *atom operators*. Following we briefly introduce the set of atom operators. The summaries of atom operators and risk estimation algebra are listed in Table 1 and 2.

- *Downstream* - D. It returns the set of objects that refer to (or are derived from) a source object o (directly or indirectly), i.e., they contain the information of o.
- *Upstream* - U. It is the reverse operator of D. It returns a set of objects referred by a target object o.
- *Join* - J. It measures the overall fraction of information of a source object o in a set of target objects N_o.

The next two operators are designed for subject network.

- *Source* - S. It takes as input a subject s and returns the set of subjects that have high leakage likelihood (above threshold δ) to s.
- *Target* - T. It is the reverse operator of S. For a given subject s, it finds the set of subjects that feature high leakage likelihood from s.
- *All* - A. This operator identifies all subjects N_s connected (transitively) to a given subject s.

The final operator extracts subject–object relationships.

- *Cross* - X. For a given set of subjects N_s, it identifies the set of objects N_o that have been accessed by N_s.

Further, we use $\cdot$ to denote the composition of two operators, and $\triangleright$ to denote an iterator which iterates over the set of elements (for each). Next we describe how these atom operators may be combined to estimate prior and posterior leakage flow capacities between subjects and objects.

Operation 1: Prior Leakage Flow Estimation

$$\mathtt{J}(o, \mathtt{X} \cdot \mathtt{S}(s, \delta))$$

For given access request $(s \to o)$, the operation of *prior-flow* estimation determines the leakage flow $f_c(o \to s)$ from object o to subject s in the current network before the request is granted. It may be implemented by composing J-,

X- and S-operator: (i) We first use S to determine the set of source subjects N_s featuring high leakage likelihood (above δ) from s. (ii) We then apply X over N_s to find the set of objects N_o accessible to N_s. (iii) Taking N_o and o as input, we use J-operator to estimate the residual information of o at N_o. If the residual information is above certain threshold ϵ, the leakage flow is considered as informative enough for s to learn o via the socio-information networks.

Operation 2: Posterior Leakage Flow Estimation

$$\triangleright s' \in \mathtt{T}(s, \delta), \triangleright o' \in \mathtt{U}(o) : \mathtt{J}(o', \mathtt{X} \cdot \mathtt{S}(s', \delta))$$

For given request $(s \rightarrow o)$, the operation of *posterior-flow* estimation identifies subject-object pairs whose flows change significantly because of granting $(s \rightarrow o)$. It may be implemented in the following steps. (i) We first apply T-operator on s to identify the set of subjects N_s featuring high leakage likelihood from s. (ii) We then identify the set of objects N_o referred by o. (iii) For each subject s' of N_s, and object o' of N_o, we follow the procedure of prior-flow estimation (with and without link $\overline{so}$) to measure the flows from o' to s'. (iv) A pair (s', o') is identified if its flow changes significantly (i.e., $f_c(o' \rightarrow s') < \epsilon \wedge f'_c(o' \rightarrow s) \geq \epsilon$) due to $(s \rightarrow o)$.

Further, since we intend to evaluate if granting an access may result in violations to access control policies, the search space can be reduced by focusing on subject-object pairs (s', o') that carries zero enforced flow, i.e., $f_e(o' \rightarrow s') = 0$.

The next set of operations are designed to support changes to security policy and personnel, i.e., administrative model. A bulk of work is available on administrating enforced flow (conventional access control models) (e.g., [11]). We therefore focus on the leakage flow part; particularly, we are interested in *incremental* update, e.g., new subjects or objects are added, new links are created, etc., and similar discussion applies to *decremental* update.

Operation 3: Adding A New Object

$$\triangleright o' \in \mathtt{U}(o) : \mathtt{J}(o', \{o\})$$

It is noted that each object may depend on (refer to) multiple other objects, e.g., one blog refers to multiple blogs; hence, on inserting a new object, we need to consider all these referring objects. We assume that the objects are inserted according to their orders of dependency (or creation time); that is, an object can be inserted only after all its dependent objects have been inserted.

Let o be the object to insert. (i) We first apply U-operator over o to identify all objects K_o directly or indirectly depended by o (closure). (ii) For each object $o' \in K_o$, we apply J-operator to estimate the residual information of o' at o. Note that this operation affects the existing information flow between objects and subjects only through new access of o, which is implemented mainly by Operation 2.

Operation 4: Updating An Existing Object

$$\triangleright o' \in \mathtt{U}(o), \triangleright o'' \in \mathtt{D}(o) : \mathtt{J}(o', \{o''\})$$

This operation updates the content of an existing object o. We assume that the acyclic directed structure of the object network is preserved after the update. Clearly, the update will affect the residual information of objects referred by o. Following the dependency relationships, we incrementally apply J-operator to update the estimation of residual information at affected objects. Note that the update influences leakage information flow only after new accesses have been executed upon o. We place a "red" flag on o to indicate that existing access privileges on o need to be re-evaluated.

Operation 5: Adding A Link in Subject Network

$$\triangleright s' \in \mathtt{A}(s), \triangleright o \in \mathtt{X} \cdot \mathtt{A}(s) : \mathtt{J}(o, \mathtt{X} \cdot \mathtt{S}(s', \kappa))$$

This operation evaluates the risk of adding a new link (let s be either one of the involved subject) to the existing subject network. Essentially, it could influence all the covert information flows for all subjects K_s relevant to s.

The implementation is as follows. (i) We first find the set of subjects K_s relevant to s using A-operator. (ii) We further apply X-operator to collect all objects K_o accessed by K_s. (iii) For each pair (s', o) $(s' \in K_s,\ o \in K_o)$, if it carries zero enforced flow, i.e., $f_e(o \rightarrow s') = 0$, we evaluate the leakage flow $f_e(o \rightarrow s')$. The asymptotic complexity of this operation is $O(|K_s| \times |K_o|)$; while in our scenarios, we focus on the subject-object pairs with zero enforced flow, which significantly reduces the search space. Further, scalable evaluation methods exist for a range of leakage measures [21].

If the inserted link results in any violation of access control policies, further actions may be taken, e.g., revoking existing access privileges.

Operation 6: Adding A New Subject

This operation inserts a new subject into the network. For a new subject whose leakage behavior is not clear (with unknown leakage flows), one may simply apply the enforced flow estimation only. As new observations are collected, one can creates its links with relevant subjects, following the procedure of Operation 5. Details are omitted here.

4. ATOM OPERATORS

We have thus far described an expressive algebra for supporting network-centric risk estimation. In this section we present one possible implementation of the atom operators. We note that while this algebra is general-purpose (applicable across a wide range of information and social networks), the concrete realizations of these operators (and their space-time complexity) are inherently tied to the complexity of information flow models in these networks. In this section we first highlight the complexity of information flow models by comparing it against the classical network flow problem [13] and then delve into concrete realizations of risk operators under specific assumptions on information flow models in the networks.

In contrast to a classical network flow problem, studying information flow in social and information networks features unique challenges. (i) Information flows violate *flow conservation*: outbound flow may exceed inbound flow at a network node (e.g., when one creates an identical (or partial) copy of an object). (ii) Information flows are *non-additive*: a set of information flows may merge into a new flow; however, this merge may be non-additive (e.g., merge of flows f_1 and f_2 results in f_1 if f_2 is derived from f_1, even though the residual information in f_2 is non-zero). (iii) Information flows in social networks are *stochastic*; it may be possible to derive statistical properties of such flows (e.g., using contact time distribution between subjects – email frequency). (iv) Network evolution must be taken account in order to refine information flow estimation over a period of time. (v) The solution should be able to support large-scale networks

(up to hundreds of thousands of nodes), and handle stream-manner updates. To the best of our knowledge, our work presents a first of a kind solution to address this information flow problem.

4.1 Operators for Information Network

We assume a class of information networks wherein information flows are restricted to a directed acyclic graph (DAG) [9], i.e., links in the network capture directed dependency among objects. Such dependency prevails in real life, in the form of *reference*, *derivation*, and *inheritance* (e.g., links between blogs, tweets, etc.). Assuming that the information network is a DAG, the implementations of *upstream* (U) and *downstream* (D) operators are fairly straightforward. Our following discussion focuses on *join* (J) operator.

For given source o_i, J-operator estimates the (fractional) *residual information* of o_i existing in a set of (target) objects $\{o_j\}$. We start with the case of a single target object o_j.

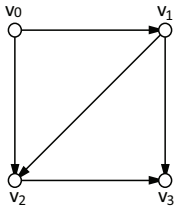

Figure 2: Diffusion and fusion in network.

If o_i and o_j are adjacent, e.g., o_j directly refers to (or inherits) certain parts of o_i, it is typically feasible to quantify the residual information r_{ij}, using information theoretic metrics [4, 19]. We thus specify the leakage flow capacity w_{ij} of direct link $\overline{o_i o_j}$ as $w_{ij} = r_{ij}$. The difficulty lies in estimating the residual information r_j[4] where o_i and o_j are not neighbors, particularly, when there exist multiple directed paths from o_i to o_j.

Example 4. Figure 2 illustrates this scenario: the information of v_0 flows through several overlapping paths to v_3. Before discussing in detail our method, we first introduce a set of fundamental concepts.

For given object o_i, the *flow* of a directed link $\overline{o_k o_j}$, f_{kj}, is the residual information of o_i passed through it. It is estimated by $f_{kj} = r_k \cdot w_{kj}$. Another key concept is the *union* of residual information (regarding o_i) over a set of flows $F = \{f_{k_1 j_1}, \ldots, f_{k_n j_n}\}$, denoted by $\oplus F = f_{k_1 j_1} \oplus \ldots \oplus f_{k_n j_n}$.

The problem of estimating the residual information r_j can then be formulated in an iterative manner:

$$\begin{cases} r_j = \bigoplus_{o_k \in P_j} f_{kj} \\ f_{kj} = r_k \cdot w_{kj} \end{cases} \quad (1)$$

where P_j represents the set of parents (direct ancestors) of o_j in the network.

Clearly, the union operation $\oplus$ is the key to estimating r_j. Following, we detail its implementation. Consider a set of flows F. We can establish the following bounds.

$$\max_F f \le \oplus F \le \min\{\sum_F f, 1\} \quad (2)$$

We focus on establishing a tighter upper bound, based on the following observation. It is observed that for two flows f_{kj_1} and f_{kj_2} going out of o_k, the residual information of their union cannot exceed that of o_k, i.e., $f_{kj_1} \oplus f_{kj_2} \le \min\{r_k, f_{kj_1} + f_{kj_2}\}$. Next we first introduce the concept of *cut*. Let all the flows of F (no inheritance relationship) inject into a virtual sink o_{sink}. A set of links are called a cut of F if they separate o_i and o_{sink}. Intuitively, all the flows of F must go through every cut of F.

For given flow f, we attempt to bound its effective part responsible for generating F (called *effective flow*, denoted by $\underline{f}$) using the observation above. Estimating the upper bound of $\oplus F$ is equivalent to finding a cut that carries the minimum (upperbound) effective flow with respect to F. We now show how to estimate the upper bound of effective flow for involved links.

[4]Following, without ambiguity, we omit the referred source object o_i in the notations.

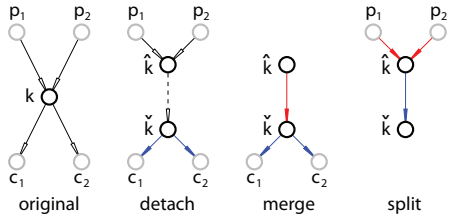

Figure 3: Primitives of constructing effective-flow graph. The effective flows in blue are materialized, and those in red are being updated.

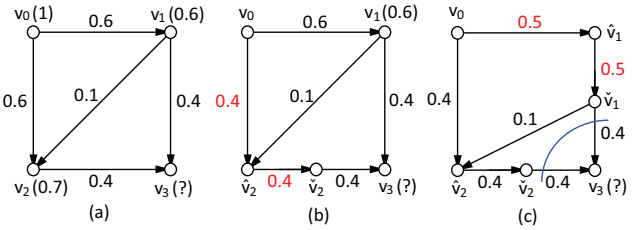

Figure 4: Construction of effective-flow graph.

We achieve this in a bottom-up manner: starting from F, following the reverse topological order, we trace back to the root o_i. At each step, we apply the following three primitives, as illustrated in Figure 3:

- *detach*. For each object o_k encountered in the process, we detach o_k into two nodes $o_{\hat{k}}$ and $o_{\check{k}}$, and connect them with a link $\overline{o_{\hat{k}} o_{\check{k}}}$; $o_{\hat{k}}$ is connected to the parent flows of o_k, while $o_{\check{k}}$ is connected to the child flows of o_k. The flow on $\overline{o_{\hat{k}} o_{\check{k}}}$, $f_{\hat{k}\check{k}}$, is set as r_k.
- *merge*. For the set of outbound flows of o_k (now $o_{\hat{k}}$), $\{f_{kc_1}, \ldots, f_{kc_n}\}$, whose effective parts have been estimated, we update the effective flow of $\overline{o_{\hat{k}} o_{\check{k}}}$ as follows (i.e., the effective flow on $\overline{o_{\hat{k}} o_{\check{k}}}$ cannot exceed the sum of the effective parts in all its child flows):

$$\underline{f}_{\hat{k}\check{k}} = \min\{f_{\hat{k}\check{k}}, \sum_{l=1}^{n} \underline{f}_{kc_l}\}$$

- *split*. Given the set of inbound flows of o_k (now $o_{\check{k}}$), $\{f_{p_1 k}, \ldots, f_{p_m k}\}$, we update the estimation regarding their maximum effective flows using the rule:

$$\underline{f}_{p_l k} = \min\{f_{p_l k}, \underline{f}_{\hat{k}\check{k}}\} \quad (1 \le l \le m)$$

Intuitively, the effective flow on each incoming link can not exceed that of $\overline{o_{\hat{k}} o_{\check{k}}}$.

If the effective flows on all the links have been estimated, finding the upper bound of $\oplus F$ is equivalent to finding a minimum cut of this *effective-flow* graph, where the capacity of each link is defined as its effective flow. Now, J-operator can be implemented as follows. For each target object o_j, we create a link $\overline{o_j o_{sink}}$ to a virtual sink o_{sink} with $w_{j,sink} = 1$, $F = \{f_{j,sink}\}$, then construct the effective-flow graph using the three primitives above, and find the minimum cut of the graph, with detailed algorithm in Algorithm 1.

Example 5. This operation over the network in Figure 3 is illustrated in Figure 4. In (a) the maximum residual information of objects v_1 and v_2 (regarding v_0) has been esti-

Input: object network $G_O = (O, L_O)$, a set of flows F
Output: upper bound of $\oplus F$
create a virtual object o_j in G_O collecting all flows in F;
$S \leftarrow$ sources of F;
// topological sorting
sort S in decreasing order;
while $S \neq \emptyset$ **do**
  $o_k \leftarrow$ pop the head of S;
  // detach + merge + split operation
  detach o_k;
  $O \leftarrow O \setminus \{o_k\} \cup \{o_{\bar{k}}, o_{\bar{\bar{k}}}\}$;
  // CF_k: child flows of o_k
  merge $CF_k \cap F$;
  // PF_k: parent flows of o_k
  split PF_k;
  // update S and F
  $F \leftarrow F \setminus CF_k \cup PF_k$;
  $S \leftarrow S \cup$ sources of PF_k;
end
// remove irrelevant objects
remove from G_O all unvisited objects;
// min-cut process
find the minimum $o_i - o_j$ cut w;
output w as the upper bound of $\oplus F$;

Algorithm 1: Sketch of J-operator.

mated (shown in parenthesis); in (b) and (c), we apply the primitives over objects v_2 and v_1, respectively. The maximum residual information of v_3 is estimated using the cut of the effective-flow graph in (c).

In addition to the upper and lower bounds of the union of a flow set F, we show that computing the expected value, $\mathbb{E}(\oplus F)$ is equivalent to a maximization problem over a graph (which is NP-hard) and present a Monte Carlo sampling approach to estimate $\mathbb{E}(\oplus F)$ (details can be found in [1]).

4.2 Operators for Social Network

In contrast with object network, subject network demonstrates dynamic aspects: the leakage behavior of each subject may only stochastically follow certain pre-defined formation. Hence, we are particularly interested in estimating the likelihood that one subject leaks (or shares) its information to another subject in the network (leakage likelihood).

We assume complete information regarding the social network. Later we will lift this assumption and take account of possible information incompleteness/uncertainty. We assume that each link $\overline{s_i s_j}$ (from subject s_i to s_j) in the social network is associated with a leakage flow capacity (or *leakage likelihood*) w_{ij}, indicating the likelihood that s_i leaks (shares) received information to s_j. This quantity has a variety of instantiations in realistic networks [12]. For instance, in online social networks, it can be estimated as the amount of information in the blogs re-posted by s_i, relative to the total information of the blogs viewed by s_i; in enterprise social networks, it can be estimated as the quantity of information in outbound emails from s_i to s_j relative to the overall information in s_i's incoming emails.

We capture the behavior of network-wise information leaking using *random walk with restart* (RWR) model [3, 24]: the leaked information is modeled as a random particle that originates at the source subject s_i. It iteratively transmits to a neighboring subject with probability proportional to the corresponding leakage flow capacity. Further, at each step, it has certain probability of stopping propagation (i.e., it is kept confidential). The likelihood that s_i leaks information to s_j, denoted by r_{ij}, can be measured by the steady-state probability that the particle is observed at s_j. This model features several desirable properties for our purpose: (i) it embeds all the likelihoods of leaking information or keeping confidentiality; (ii) it considers the multi-facet relationships between two subjects; (iii) it captures the global structure of the subject network. Next, we formalize this model, and construct Source (S) and Target (T) operators based on it.

Consider a subject s_i as the source subject. Let N_i denote the set of outgoing neighbors of s_i. For each subject $s_j \in N_i$, we specify the probability that the information (particle) transmits through $\overline{o_i o_j}$ as: $p_{ij} = w_{ij} / \sum_{j' \in N_i} w_{ij'}$. At each step, the probability c_i that the information stops propagation is specified as: $c_i = w_{ii} / (w_{ii} + \sum_{j \in N_i} w_{ij})$. If we stack the steady-state probability that the information (particle) is observed at each subject of the network into a column vector r_i, the definition of RWR gives us:

$$r_i = (1 - c_i) \cdot A \cdot r_i + c_i \cdot e_i \tag{3}$$

where A is the column normalized adjacent matrix of the subject network, such that $A_{ji} = p_{ij}$ if $o_j \in N_i$ and 0 otherwise, and e_i is the starting vector for s_i with the i-th entry set as 1 and 0 otherwise. Furthermore, we have the following matrix formation: $R = A \cdot R \cdot (I - C) + C$. Here R denotes the stack of the leakage probabilities with respect to all subjects, $R = [r_1, r_2, \ldots]$, I is an identity matrix, and C represents the diagonal matrix with the i-th diagonal element as c_i. Scalable algorithms are available to compute R (e.g., [15]).

Based on this formulation, the implementations of S and T operators are as follows. For given target s_j, S operator identifies the set of subjects that feature high leakage likelihood (above a threshold δ_i) to s_j, which correspond to the elements s_i in the j-th row of R with $R_{ji} \geq \delta_i$[5], denoted by $\mathtt{S}(s_j) = \{s_i | R_{ij} \geq \kappa_i\}$. Meanwhile, for given source subject s_i, T operator returns the set of subjects that feature high leakage likelihood from s_i, i.e., $\mathtt{T}(s_i) = \{s_j | R_{ji} \geq \delta_i\}$.

5. EMPIRICAL EVALUATION

This section presents an empirical study of our network-centric access control paradigm. The experiments are specifically designed to center around the following metrics: (i) its validity in terms of capturing leakage flow, (ii) its efficacy in quantifying unpredicted risk incurred by ignoring the network effects among subjects and objects, (iii) its effectiveness in incorporating the impact of network evolution over risk estimation, and (iv) its execution efficiency. We start with describing the setup of the experiments.

5.1 Experimental Setting

Our experiments used three datasets collected from real-life social and information networks (attributes of interest to us are listed in Table 3).

The Twitter dataset contains 18,617,827 tweet messages, involving 203,222 users, over three weeks of 2009. The social network is constructed according to the following/followed relationships among users: one user s_i follows another user s_j if s_i wishes to receives tweets from s_j; also, s_i can *re-tweet* (re-post) the viewed tweets to its followers.

The SmallBlue dataset describes the social network of IBM employees who participated in the SmallBlue project. It consists of two snapshots of as of January 2009 and July 2009, involving 41,702 and 43,041 individuals, respectively,

[5]The parameter δ for each subject needs to be normalized to accommodate the difference of subjects' influence in the network. In implementation, we set $\delta_i = \delta / (\sum_{j \neq i} \mathbf{1}_{R_{ji} > 0})$.

Attribute	Description
Twitter dataset	
createdat	timestamp of message post
tweetid	unique message id
text	plain message content
userid	id of the message author
rtstatus	re-tweeted message id (by this one)
rtuser	id of the re-tweeted message author
Smallblue dataset	
location	working location
position	managerial position
division	working department
tie strength	volume of exchanged emails
Dogear dataset	
email	email address of user s (identifier of subject)
url	url o bookmarked by s (identifier of object)
tags	bookmark tags made by s regarding o
time	time-stamp that s accesses o

Table 3: Attributes of datasets.

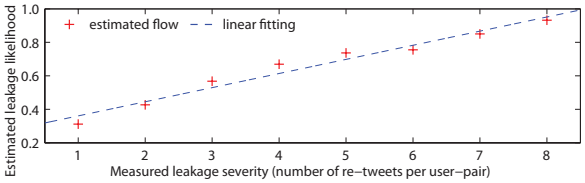

Figure 5: Estimated leakage likelihood with respect to actually observed leakage severity (number of re-tweets per user-pair).

We construct the social network according to the social connections among individuals.

The Dogear dataset consists of 20,870 bookmark records, relevant to 7,819 urls. The *email* and *url* attributes uniquely identify a user (subject) and a webpage (object), respectively, and *tags* encodes the semantics of the object. We construct the object network as follows. Let t_i be the collection of tags suggested by users regarding object o_i. We consider both potential temporal and semantic dependency among objects. For object o_j, we assume that it is directly dependent on the minimum set of (temporally) most recent objects O_j with the union of tags maximally possibly covering t_j, i.e., $\max \cup_{o_i \in O_j} t_i \bigcap t_j$. The weight of link $\overline{o_i o_j}$ is defined as $w_{ij} = |t_i \cap t_j|/|t_i|$.

All the core algorithms (the library of operators) are implemented in Java. The experiments are conducted on a workstation with 3.20GHz Intel Celeron CPU and 2GB RAM, running Windows XP.

5.2 Experimental Results

In the first set of experiments, we use the Twitter dataset and Enron archive to validate the leakage flow model on real social and information network platform.

Validity of Leakage Model

We set up the experiments as follows. On Twitter, the social network is constructed according to the following/followed relationships: one user (subject) s_j opts to follow another user s_i if s_j wishes to receive messages (*tweets*) from s_i; also, s_j can "leak" (*re-tweet*) the tweets from s_i, which may be further leaked by followers of s_j. Clearly, such leakage can happen between two remotely connected users due to the network effects.

We intend to apply the leakage flow model to quantify the likelihood that the information (tweets) possessed by one subject leaks to another subject in the network, and compare the estimated leakage flow with actually measured leakage (number of re-tweets). We use the data corresponding to

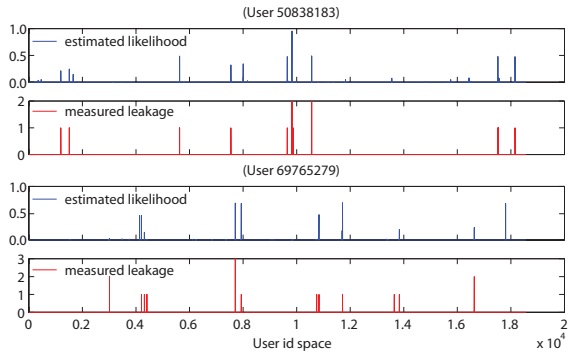

Figure 6: Individual level comparison of estimated leakage likelihood and observed leakage severity.

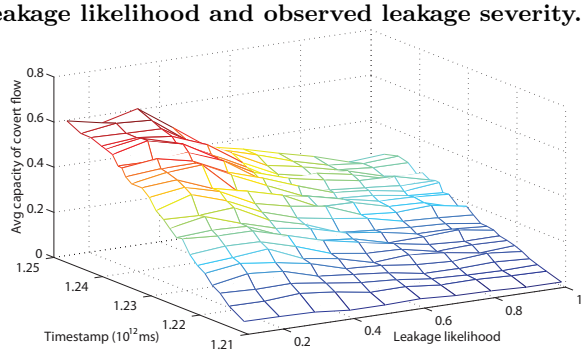

Figure 7: Average covert information flow measured at different time-stamps.

October 2009 to collect the overall statistics regarding each user, particularly the number of received tweets and among them the number of re-tweets, which we use to set up the parameters $\{w\}$ and $\{c\}$ as in Appendix 4.2. We apply the model to predict the leakage likelihood for the period of November 2009.

For a specific pair of users s_i and s_j, we consider the severity of leakage as the number of re-tweets s_j posts with original tweets from s_i during the considered time period. For each specific level of observed leakage severity, Figure 5 shows the corresponding estimated leakage flow (leakage likelihood) averaged over pairs of users demonstrating such severity. One can notice the high correlation between the estimated flow and the actual leakage severity, indicating that the leakage flow model captures the essence of leakage patterns. We further perform individual level comparison of estimated leakage flow and observed leakage severity. We randomly pick two (sources) users, measure their leakage severity to the rest users, and compare the results with the estimated likelihood by our model. As shown in Figure 6, it is noticed that the predicted "peaks" match well with the actually measured results.

Impact of Leakage Flow

Next we intend to evaluate the impact of leakage flow existing in the social and information networks over the risks associated with access control decisions, specifically, the risk of information leakage that would be under-estimated if ignoring the network effects among subjects and objects.

We use the SmallBlue and Dogear datasets to construct the socio-information network. We consider an bookmarking action as an access; hence, each access request q is associated with a time-stamp t_q. At each specific time-stamp t^*, we assume that the set of requests before t^*, $\{q|t_q \leq t^*\}$, have

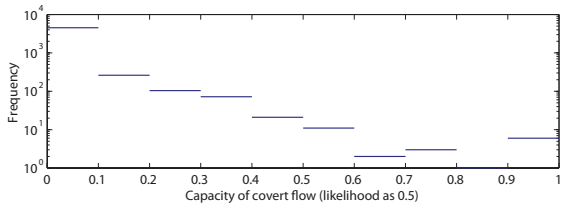

Figure 8: Distribution of leakage flows of 5K randomly generated requests (likelihood fixed as 0.5).

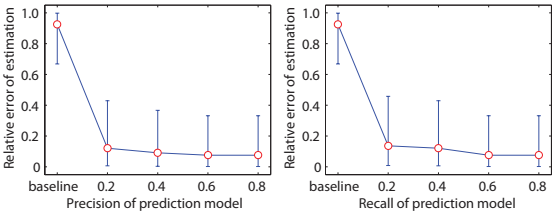

Figure 9: Estimation error with respect to the accuracy of 2-hop link prediction model, where no prediction is employed in the baseline approach.

been granted; we then randomly generate a set of access requests, and evaluate their corresponding leakage flows.

More concretely, we consider the history from 12/01/2005 to 07/20/2009, 20,870 access requests in total. At a step of 1,000 requests, we evaluate the leakage flows for 5K randomly generated requests. The average leakage flow with respect to time-stamp and leakage likelihood is plotted in Figure 7. It is noticed that as more requests are granted, the average flux increases significantly. This is explained by that the newly-created inter-network links between social and information networks generally increase the leakage flow capacity between the two networks, which also implies the non-negligible impact of the network effects among subjects and objects over access control risks.

We further look into the distribution of leakage flows of subject-object pairs. For the time-stamp of 07/20/2009, we measure the leakage flows for 5K randomly generated requests. Figure 8 shows the result. The distribution demonstrates a long tail, which is mainly attributed to the heterogeneity of social and information networks; that is, there exist "hot" spots in both networks, which feature large leakage flows; the existence of "hot" spots necessitates careful risk estimation before making access control decisions.

Incorporation of Network Evolution

One critical feature that makes our paradigm useful is its capability of incorporating predicted network evolution in current risk estimation procedure. In our experiments, we focus on on the impact of social network evolution. We consider the two snapshots of the social network, and focus on the set of individuals appearing in both snapshots, which contains 32,028 users. From January to July 2009, 81,592 new relationships were created among these subjects. We assume the prediction model [17] that predicts new links that spans two hops.

We intend to study the robustness of our estimation model against the prediction error incurred by the prediction model. For 0.5K randomly generated access requests, we measure the leakage flow over the network snapshot as of July 2009 (measured flow), and compare the result with that estimated based on the snapshot as of January 2009, in conjunction of the prediction model (estimated flow). We evaluate the relative estimation error as a function of the accuracy of prediction model, with default recall and precision set as 0.5. The result is shown in Figure 9. In both cases, the estimation made by the baseline approach (without prediction) considerably deviates from the actually measured result, with relative error around 0.9. Employing the prediction model significantly improves the estimation accuracy; even with recall fixed as 0.2, the average error is reduced around 0.2. However, as the precision (or recall) increase, the further accuracy improvement is flat; this is explained by that the model only considers 2-hop links, while other types of links account for 49% of new relationships.

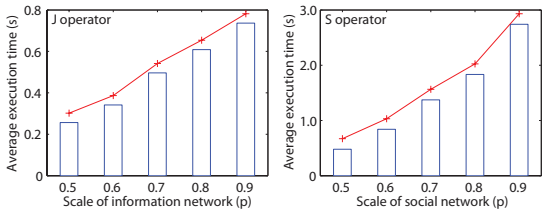

Figure 10: Average execution time of J-operator and S-operator as a function of network scale.

Scalability of Risk Estimation

In this set of experiments, we investigate the overhead of risk estimation over the access control infrastructure. Each risk estimation operation (Section 3) is constructed by composing a set of atom operators, whose complex interactions make it fairly difficult to directly characterize their impact over the scalability of risk estimation. We therefore focus our discussion at the level of atom operators. Due to the space limitation, we particularly study J-operator and S-operator, given their high frequency of usage.

We evaluate their execution time with respect to the scale of social and information networks. To do so, we create a set of network copies of different scales by randomly removing certain number of nodes (and their associated links) from the original network; a p-scale network indicates that $100 * (1-p)\%$ nodes are removed. Figure 10 shows the result. It is clear that both operators scale approximately linearly with the size of the networks, which empirically substantiates the analysis in Section 4.

6. DISCUSSION AND CONCLUSION

This work advances the state-of-the-art in risk-based access control by presenting a novel network-centric access control paradigm that explicitly accounts for the network effects in information flows. We show that a broad range of traditional node-centric models can be enhanced in terms of risk estimation using this general framework. While our framework is rich and flexible, several key challenges need to be addressed before it can be readily adopted. First, our approach relies on measures of information flow in socio-information networks. We believe that recent advances in network science research make it feasible (in part) to estimate such information flows. Second, we believe that applications in the future will be risk-based, i.e., they will exploit risk estimates to guide their decision-making (e.g., using budget based policies or exception handling mechanisms). Third, we believe that incorporating the time dimension (e.g., due to network evolution or due to decay in sensitivity of information) into risk estimation is essential to make sound decisions; while our approach handles network evolution, it

does not explicitly address information items whose sensitivity decays over time. Despite these limitations, we believe that our proposed approach offers a new approach to modeling data flows in socio-information networks.

7. REFERENCES

[1] *Network-centric access control: models and techniques*. Georgia Tech Technical Report, GIT-CERCS-10-08, http://www.cercs.gatech.edu/tech-reports/.

[2] Teacher fired over Facebook sues district: http://www.cbsatlanta.com/news/21573759/detail.html.

[3] D. Aldous and J. A. Fill. Reversible markov chains, 1994.

[4] M. Backes, B. Kopf, and A. Rybalchenko. Automatic discovery and quantification of information leaks. In *SP*, 2009.

[5] D. E. Bell and L. J. LaPadula. Secure computer system: unified exposition and multics interpretation. In *MITRE Corporation*, 1976.

[6] E. Bertino, P. A. Bonatti, and E. Ferrari. Trbac: A temporal role-based access control model. *ACM Trans. Inf. Syst. Secur.*, 4(3):191–233, 2001.

[7] D. D. F. Brewer and D. M. J. Nash. The chinese wall security policy. *SP*, 1989.

[8] B. Carminati, E. Ferrari, S. Morasca, and D. Taibi. A probability-based approach to modeling the risk of unauthorized propagation of information in on-line social networks. In *CODASPY*, 2011.

[9] J. R. Challenger, P. Dantzig, A. Iyengar, M. S. Squillante, and L. Zhang. Efficiently serving dynamic data at highly accessed web sites. *IEEE/ACM Trans. Netw.*, 12(2):233–246, 2004.

[10] P.-C. Cheng, P. Rohatgi, C. Keser, P. A. Karger, G. M. Wagner, and A. S. Reninger. Fuzzy multi-level security: An experiment on quantified risk-adaptive access control. In *IEEE Security and Privacy Symposium*, 2007.

[11] J. Crampton. Understanding and developing role-based administrative models. In *CCS*, 2005.

[12] D. Easley and J. Kleinberg. *Networks, Crowds, and Markets: Reasoning About a Highly Connected World*. Cambridge University Press, 2010.

[13] J. Edmonds and R. M. Karp. Theoretical improvements in algorithmic efficiency for network flow problems. *J. ACM*, 19(2):248–264, 1972.

[14] D. Ferraiolo and R. Kuhn. Role-based access control. In *15th NIST-NCSC National Computer Security Conference*, 1992.

[15] D. Fogaras, B. Rácz, K. Csalogány, and T. Sarlós. Towards scaling fully personalized pagerank: Algorithms, lower bounds, and experiments. *Internet Mathematics*, 2(3), 2005.

[16] Y. Kanzaki, H. Igaki, M. Nakamura, A. Monden, and K.-i. Matsumoto. Characterizing dynamics of information leakage in security-sensitive software process. In *ACSW Frontiers*, 2005.

[17] J. Leskovec, L. Backstrom, R. Kumar, and A. Tomkins. Microscopic evolution of social networks. In *KDD*, 2008.

[18] C.-Y. Lin, N. Cao, S. X. Liu, S. Papadimitriou, J. Sun, and X. Yan. Smallblue: Social network analysis for expertise search and collective intelligence. In *ICDE*, 2009.

[19] S. McCamant and M. D. Ernst. Quantitative information flow as network flow capacity. In *PLDI*, 2008.

[20] I. Molloy, P.-C. Cheng, and P. Rohatgi. Trading in risk: using markets to improve access control. In *NSPW*, 2008.

[21] H. H. Song, T. W. Cho, V. Dave, Y. Zhang, and L. Qiu. Scalable proximity estimation and link prediction in online social networks. In *IMC*, 2009.

[22] M. Srivatsa, D. Agrawal, and S. Reidt. A metadata calculus for secure information sharing. In *CCS*, 2009.

[23] M. Srivatsa, P. Rohatgi, S. Balfe, and S. Reidt. Securing information flows: A metadata framework. In *QoISN*, 2008.

[24] H. Tong, C. Faloutsos, and J.-Y. Pan. Fast random walk with restart and its applications. In *ICDM*, 2006.

[25] N. Zeldovich, S. Boyd-Wickizer, and D. Mazières. Securing distributed systems with information flow control. In *NSDI*, 2008.

Acknowledgements

The research of this work was sponsored by the U.S. Army Research Laboratory and the U.K. Ministry of Defence and was accomplished under Agreement Number W911NF-06-3-0001. The views and conclusions contained in this document are those of the author(s) and should not be interpreted as representing the official policies, either expressed or implied, of the U.S. Army Research Laboratory, the U.S. Government, the U.K. Ministry of Defence or the U.K. Government. The U.S. and U.K. Governments are authorised to reproduce and distribute reprints for Government purposes notwithstanding any copyright notation hereon. The first and last authors are also sponsored by grants under NSF NetSE and NSF Cybertrust program, an IBM SUR grant and a grant from Intel Research Council.

APPENDIX

A. RECASTING CONVENTIONAL MODELS

Access control policies encode the rules used to regulate the qualification of $s \in S$ to access $o \in O$. In composing such policies, traditional models typically adopt a *node-centric* paradigm; they either treat each subject and object in isolation, or only partially account for the relationships between subjects, between objects and between subjects and objects.

Next we use three representative access control models as concrete examples to show the compatibility of our network-centric paradigm; how to extend to more complicated models (e.g., FuzzyMLS [10]) is also discussed; further we address how the dynamic aspects of access control models including administrative update of security setting and exception handling (e.g., the upgrade/downgrade of subjects' security levels in MLS model, the periodic role enabling and disabling in temporal RBAC model [6]) are implemented.

A.1 Multi-Level Security (MLS) Model

We use Bell-LaPadula (BLP) model, a classic MLS access control model, as the first concrete example. In its simplest form, the policies in BLP are described by two terms, the *security attributes* of the objects/subjects concerned and the rules for access. BLP attaches security labels to both objects (classification levels) and subjects (clearance levels) (more precisely, the combination of classification/clearance and a set of compartments); the classification/clearance scheme is described in terms of a lattice. Further, BLP has a *simple-security* and a $*$*-property* rule, which can be characterized as "no read up, no write down":

- Simple security. Read access is allowed only if the subject's clearance is above the object's classification.
- $*$-property. Write access is granted only if the subject's clearance is below the object's classification.

Under the network-centric framework, the implementation of BLP is fairly straightforward:

- The subject and object networks consist of the set of subjects S and objects O.
- The inter-network links encode the clearance/classification comparison of subjects and objects. Specifically, for subject s and object o, $\texttt{enf}(\overline{os}) = 1$ if $\texttt{clear}(s) > \texttt{class}(o)$, $\texttt{enf}(\overline{so}) = 1$ if $\texttt{class}(o) > \texttt{clear}(s)$, and 0 otherwise.
- An access request $(s \to o)$ is granted only when the enforced flow $f_e(s \to o) = 1$ for a read access, or $f_e(o \to s) = 1$ for a write access.

Variations in MLS can be accommodated by modifying this basic construction in different ways.

A.2 Fuzzy MLS

Unlike the simple dichotomic comparison of the classification $\texttt{class}(o)$ of object o and clearance $\texttt{clear}(s)$ of subject s

in conventional MLS [5], Fuzzy MLS computes a quantified *risk* for an access request $(s \to o)$ based on the gap between $\texttt{clear}(s)$ and $\texttt{class}(o)$, and specifies a region on the risk scale (which can be further divided into bands). It allows access with risk below the lower-bound of the region (soft boundary), denies access with risk above the upper-bound of the region (hard boundary), and charges access with risk laying between soft and hard boundaries by its difference to the soft boundary against subject's risk credit. Each subject is periodically (say, monthly) allotted a risk budget.

We can implement Fuzzy MLS using the network-centric paradigm as follows. Let $\texttt{risk}(s,o)$ represent the risk associated with the access $(s \to o)$, L and U denote the soft and hard boundaries, and $\texttt{budget}(s)$ be the budget allotted to s initially (all in the unit of risk credit). For given subject s, we create a link $\overline{os}$ if $\texttt{risk}(s,o) < \texttt{U}$, and specify its enforced flow capacity initially as $\texttt{enf}(\overline{os}) = \infty$ if $\texttt{risk}(s,o) \le \texttt{L}$, or $\texttt{enf}(\overline{os}) = \texttt{budget}(s)$ if $\texttt{risk}(s,o) > \texttt{L}$. An access $(s \to o)$ is granted only if $\texttt{enf}(\overline{os}) > 0$. After the access, all links of s update their enforced flow capacities by decreasing $(\texttt{risk}(s,o) - \texttt{L})$. After the periodic allocation of risk budget, the flow capacities of links $\overline{os}$ with $\texttt{L} < \texttt{risk}(s,o) < \texttt{U}$ are restored to $\texttt{budget}(s)$.

A.3 Role-based Access Control (RBAC) Model

RBAC models explicitly capture the relationships among subjects by organizing them according to their functional roles. A typical RBAC [14] model uses the following conventional notations: S, the set of subjects, R, the set of *roles*, which describe authorization levels, and P, the set of permissions, which represent the approval of access to concerned objects. Access control policy can be described by the following three mappings:

- Subject assignment, $SA \subseteq S \times R$, which is a many to many subject to role assignment relation;
- Permission assignment, $PA \subseteq P \times R$, which is a many to many permission to role assignment relation;
- Role hierarchy, $RH \subseteq R \times R$, which is a partially ordered role hierarchy. Two roles $r \succeq r'$ means r inherits the permissions of r'.

Under the network-centric paradigm, we intend to encode these three mappings via the subject and object networks and the inter-network relationships. One possible implementation could be as follows:

- In the subject network G_S, in addition to the set of subjects S, for each role $r \in R$, we create a corresponding node r. For each $s \in S$, a link $\overline{rs}$ $(\texttt{enf}(\overline{rs}) = 1)$ is created to indicate that s is assigned role r, i.e., SA mapping. The sub-network G_R over the set $\{r \in R\}$ encodes the role hierarchy RH: two nodes r and r' are adjacent over the link $\overline{rr'}$ $(\texttt{enf}(\overline{rr'}) = 1)$ if they are adjacent in RH and $r \succeq r'$.
- Due to role inheritance, if subject s is associated with any two roles r and r' with $r \succeq r'$, only the link $\overline{rs}$ is necessary, which implies the link $\overline{r's}$.
- The object network G_O is a set of nodes, each corresponding to one object $o \in O$.
- The inter-network relationships between G_O and the sub-network G_R encode the permission assignment PA. Each link $\overline{or}$ between object o and role r indicates that r has access to o, and the access mode is contained in the type information of $\overline{or}$.
- An access request is granted only if the enforced flow $f_e(o \to s) = 1$ (the type of inter-network link must be equivalent to the requested access mode) in this network.

The basic model can be further enriched to support features such as session-based role activation, constraints on subjects/objects/roles, and multiple security domains.

A.4 Chinese-wall Model

Beyond other conventional access control models, Chinese-wall model [7] and its variations further take into consideration the conflict of interest in objects. It also has a dynamic aspect that accounts for the the access history of subjects regarding the objects concerned. In a simplified Chinese-wall model, each object o is associated with two label x_o indicating the commercial database holding o, and y_o indicating its *conflict of interest class*. The basic Chinese-wall policy can be described as:

- Simple security. An access $(s \to o)$ is granted only if o has the same label x_o as an object o' already accessed by s, i.e., within the wall, or has an entirely different label y_o to all the objects already accessed by s.
- $*$-property. Write access is granted only if the simple security rule is honored, and no accessible object o' contains unsanitized information and has a different label $x_{o'}$ to the requested one o.

Under the network-centric framework, one implementation of Chinese-wall model could be as follows:

- The subject network is a set of nodes, each corresponding to a subject $s \in S$.
- In the object network, a pair of objects o and o' are bi-directionally adjacent if (i) $x_o = x_{o'}$ or (ii) $y_o \neq y_{o'}$ (both $\texttt{enf}(\overline{oo'}) = 1$ and $\texttt{enf}(\overline{o'o}) = 1$). Further, each object is labeled by either *sanitized* or *unsanitized*.
- Once subject s has accessed object o, an inter-network link $\overline{os}$ $(\texttt{enf}(\overline{os}) = 1)$ is added to the network.
- If subject s has not yet accessed any object, i.e., no inter-network link exists for s, an access request $(s \to o)$ is granted by default. Otherwise, the request is granted only if the enforced flow estimation $f_e(o \to s) = 1$ for a read access; and (i) $f_e(o \to s) = 1$, (ii) $\nexists\, o', f_e(o' \to s) = 1, x_o \neq x'_o$, o' is unsanitized for a write access.

A.5 Dynamic Aspects

To accommodate changing application environments, many access control models introduce dynamic aspects: e.g., downgrade/upgrade of subjects' sensitivity labels in MLS [5], dynamic role dependencies in temporal RBAC [6]. Here, we use the periodic role enabling/disabling in TRBAC as an example to show how to implement such dynamic aspects in network-centric paradigm.

We still follow the generic information flow model introduced in Section 2. In addition to labeling network links, we also assigns labels to network nodes. Now, nodes act like *switches*, which can block or unblock enforced flows. Particularly, in periodic role enabling/disabling, each role is associated with a *periodic expression* that indicates the activation period of the role, e.g., $all \cdot Year + \{1,4\} \cdot Months \triangleright 2$ represents the set of intervals starting at the first and fourth month of every year, and having a duration of two months [6]. We can attach such expression to the corresponding role node (see Appendix A.3), and activate the node only when the expression is true. This way, all permissions associated with a disabled role are detached from the subjects associated with the role (zero enforced flows).