A Probability-based Approach to Modeling the Risk of Unauthorized Propagation of Information in On-line Social Networks

Barbara Carminati, Elena Ferrari, Sandro Morasca, Davide Taibi
Dipartimento di Informatica e Comunicazione
Università degli Studi dell'Insubria
Via Mazzini 5, I-21100, Varese, Italy
{barbara.carminati, elena.ferrari, sandro.morasca, davide.taibi}@uninsubria.it

ABSTRACT

The unauthorized propagation of information is an important problem in the Internet, especially because of the increasing popularity of On-line Social Networks. To address this issue, many access control mechanisms have been proposed so far, but there is still a lack of techniques to evaluate the risk of unauthorized flow of information within social networks. This paper introduces a probability-based approach to modeling the likelihood that information propagates from one social network user to users who are not authorized to access it. The approach is demonstrated via an example, to show how it can be applied in practical cases.

Categories and Subject Descriptors

H.2.0 [**Database Management**]: General—*Security, integrity and protection*; D.2.8 [**Software Engineering**]: Metrics—*complexity measures, performance measures*

General Terms

Security, Measurement

Keywords

Social Networks, Privacy, Access control, Information leakage

1. INTRODUCTION

The Web is no longer just a simple tool for publishing textual data or images, but it has now evolved into a complex collaborative knowledge management system. This evolution is mainly due to the rapid spread of social computing services, such as blogs, wikis, social bookmarking, collaborative filtering, and social networks [15]. On-line Social Networks (OSNs) represent one of the most relevant phenomena related to Web 2.0. OSNs are online communities that allow users to publish resources and record and/or establish relationships with other users, possibly of different type ("friend of," "colleague of," etc.), for purposes that may concern business, entertainment, religion, dating, etc. To have an idea of the relevance of the social networking phenomena, just think that Facebook counts more than 500 million users.[1]

Additionally, social networking services are today more and more used not only by single users, but at the enterprise level to communicate, share information, make decisions, and, in general, do business. This is in line with the emerging trend known as Enterprise 2.0 [14] — the use of Web 2.0 technologies within the Intranet, to allow for more spontaneous, knowledge-based collaboration. However, despite all the benefits of social network facilities in terms of knowledge-based collaboration and information sharing, there still exist important problems in the further diffusion of such technologies. One of the most serious obstacles is related to security, in terms of ensuring users that their privacy and access control requirements are preserved when sharing information within social networks. These needs have resulted in the development of several privacy preserving techniques and access control models (see, for example [5] for a survey) for OSNs. Almost all the defined access control mechanisms implement *topology-based* access control, which basically identifies authorized users by specifying constraints on the user social graph. As such, access control rules regulating information sharing are defined by specifying the relationships that users must have in order to have the right to access resources. For instance, by means of topology-based access control, it is possible to easily define rules to authorize "only direct friends," "only friends of friends," etc. Some of the access control models proposed so far also use trust and/or reputation as a further parameter on which access control is based. Additionally, a basic form of topology-based access control is also provided by existing commercial social networks. For example, in addition to allowing a user to mark a given resource as public, private, or accessible by direct contacts, Bebo (http://bebo.com), Facebook (http://facebook.com), and Multiply (http://multiply.com) support the option "selected friends" (selected contacts); Last.fm (http://last.fm) supports the option "profile neighbors" (i.e., the set of OSN members having musical preferences and tastes similar to mine); Facebook, Friendster (http://friendster.com), and Orkut (http://www.orkut.com) support the option "friends of friends"; Xing (http://xing.com) supports the options

CODASPY'11, February 21–23, 2011, San Antonio, Texas, USA.

[1]http://www.facebook.com/press/info.php?statistics.

"contacts of my contacts" (2nd degree contacts), and "3rd" and "4th degree contacts"; LinkedIn (http://www.linkedin.com) and Multiply support the option "my network" (n-th degree contacts, i.e., all the OSN members to whom a user is either directly or indirectly connected, independent of how distant they are).

The main benefit of topology-based access control is its flexibility in terms of policy specification, since authorized users can be simply specified by stating conditions on relationships, their depth, and trust levels. This flexibility, however, may potentially lead users to losing control of their data. Since access rules specify authorized users at an intensional level, i.e., as constraints on relationships in the OSN, the user specifying the rule might not be able to precisely identify who is authorized to access his/her resources. Even in small social networks, one can hardly understand which users are actually authorized even with simple access rules such as "friends of friends of my friends," due to the many relationships that users can establish. This possible loss of control generates serious potential *risks of unauthorized information flow*. A user does not directly know the set of users authorized by his/her policies, so he or she may not actually be aware of potentially malicious behaviors of these users in releasing accessed data to unauthorized users.

Therefore, there is a need for quantifying the *potential risks* that may result from the access rules specified in OSNs, so the users are fully aware of the possible effects of their decisions in specifying access rules. In this paper, we introduce a probability-based approach for quantifying the probability that user resources may become accessible to another user of the OSN. This probability is computed based on the probability of propagation of information associated with each direct relationship present in the OSN. Specifically, we show how to exactly compute the probability that a resource propagates from one user to another on the set of paths that link the two users. Also, because the exact computation of this probability may be computationally intensive, we show how an upper bound for this probability can be derived. Then, we quantify the *Unauthorized Access Risk* (UAR) as an upper bound to the probability that sensitive resources reach any unauthorized user in an OSN that enforces topology-based access control. The approach is demonstrated via an example having as target the Enterprise 2.0 domain, to show how it can be applied in practical cases. It is relevant to note that the probability-based approach for UAR estimation presented in this paper is just the core component of a more comprehensive framework for information flow management and prevention in OSNs. As it will be discussed in Section 5, the framework needs to be complemented with other important functionalities (e.g., automatic computation of probability of information propagation associated with a relationship, tailored GUI helping users to set up access control rules based on the UAR metric).

Assessing the implications of access control policies traditionally lies in the domain of safety/security analysis, which has been addressed for several different domains (e.g., operating systems [10], role-based access control [13], trust management [16]) but to the best of our knowledge not for OSNs. In contrast, in the field of OSN, literature offers several topology-based access control models and mechanisms for social networks (e.g., [1, 4, 6, 7, 8, 12]). However, to the best of our knowledge, this is the first work proposing a measure for the risk of information leakage due to unauthorized propagation. Inference problems in OSNs have been addressed by other work, but from a totally different perspective, mainly related to sensitive attribute inference. For instance, Zheleva and Ghetoor in [17] address the problem of inferences of private user attributes from public profile attributes, links, and group memberships in OSNs, whereas [11] investigates the effect of social relations on sensitive attribute inference. The work that is most related to the proposal in this paper is [2], where a privacy-preserving tool is proposed to enable a user to visualize the view that other users have of his or her Facebook profile, on the basis of the specified privacy policies. This means that a user should explicitly select one of his or her neighbors n in the OSN to see what n can see of his or her profile. However, due to the huge number of users in an OSN, it may be almost impossible by using this tool to understand the effect of a policy in terms of unauthorized information disclosure, which is the focus of our work.

The remainder of this paper is organized as follows. Section 2 introduces basic concepts on OSNs and topology-based access control. Section 3 presents the probability-based approach, whereas Section 4 shows some examples of its application. Finally, Section 5 concludes the paper and outlines future work.

2. BASIC CONCEPTS

In this section, we introduce the modeling approach we use to represent an OSN (Section 2.1), then, we illustrate the reference access control model we adopt to identify authorized users (Section 2.2).

2.1 The Underlying Model of OSNs

An OSN may be modeled as a directed labeled graph, where nodes correspond to users and arcs denote relationships between users. Given a relationship, the initial node of an arc denotes the user that has established the relationship and the terminal node the user that has accepted that relationship. For notational convenience, we use letters from the Greek alphabet to denote nodes.

The OSN model also supports different types of relationship (e.g., "friend of," "colleague of"), which are modeled as labels of the arcs. We say that two users α and β are in a *direct* relationship of a given type rt if there is an arc connecting α and β that bears the label rt. Also, two users α and β are in an *indirect* relationship of a given type rt if there is a directed path of more than one arc connecting α and β such that all of the arcs on the path bear the label rt.

A relationship of type rt from user α to user β may be characterized by a trust level, representing how trustworthy α considers β, as far as a relationship of kind rt is concerned. Thus, each arc is annotated with a value $t \in [0, 1]$ that quantifies the trust level associated with the relationship represented by the arc.

Information may be passed along the relationships of the OSN, and there is a risk that a confidential resource is illegally released to unauthorized users. As shown in Section 3.5, we introduce the *Unauthorized Access Risk* as an upper bound to the probability that a confidential resource reaches unauthorized users directly or via a path of relationships in the OSN. To this end, in our model, each arc is associated with the probability that information is propagated by means of the relationship represented by the arc. More precisely, given two users α and β, directly connected by an arc, $p(\alpha,\beta)$ quantifies the conditional probability that, if α knows a given resource rsc, then he or she propagates rsc to β, i.e.,

$$p(\alpha,\beta) = p(\alpha\_makes\_rsc\_known\_to\_\beta|\alpha\_knows\_rsc) \quad (1)$$

Thus, we assume that the probability of propagation of rsc from one node to another does not depend on the previous propagation history of rsc. So, even if α may receive rsc from multiple nodes, $p(\alpha,\beta)$ does not depend on the specific nodes that have propagated rsc to α, nor on the fact that α may have created rsc. Note that, in addition, $p(\alpha,\beta)$ is defined regardless of the fact that a resource rsc is legally or illegally propagated on the arc connecting α and β according to the access rules associated with rsc (see Section 2.2).

Summarizing, a social network OSN can be formally modeled as a tuple $OSN =< N, A, RT, TL, lab >$, where

- N represents the set of nodes (i.e., the users) of the social network;
- RT represents the set of relationship types existing in the social network;
- $A \subseteq N \times N \times RT$ is the set of arcs (i.e., the set of relationships between users in the social network) of the social network OSN;
- TL is the set of supported trust levels, which we assume to be the closed interval $[0, 1]$ in this paper;
- $lab : A \to TL \times [0, 1]$ is a labeling function that assigns to each relationship $r \in A$ a trust level $t \in TL$, and a probability $p \in [0, 1]$ that information propagates along the arc.

Note that in what follows, for simplicity and notational convenience, we use graphs and not multigraphs, i.e., given any two nodes α and β, there is at most one arc connecting α to β. For instance, this means that it is not possible that α and β are connected by a "friend of" and "colleague of" direct relationship at the same time. So, the pair $< \alpha, \beta >$ uniquely denotes an arc connecting two nodes, where for simplicity we omit the relationship type.[2] Therefore, we can safely write $p(\alpha,\beta)$ to denote the probability associated with it. This will not affect the computation of the resource propagation probabilities of Section 3.

There may be several ways to compute probability $p(\alpha,\beta)$. Indeed, based on the social network context, it is easy to figure out different factors that impact this probability, like users' reputation, relationships semantics, etc. However, since this probability value is just a parameter of the proposed Unauthorized Access Risk measure, we do not address the issue of its computation in the current paper, but we plan to address this in our future work.

Figure 1 contains an example of a portion of an OSN for a financial domain. For instance, the arc from α to β shows that α is in relationship "MOf" (i.e., "manager of") with β, that this relationship has a 0.8 trust level, and that it has a 0.5 probability that information is propagated from α to β. This example is explained in more details in Section 2.2 and used in Section 4 to show how our approach can be applied in practice.

[2]Note that this assumption is in line with proposals in the social network analysis literature, where arcs are no labelled. Moreover, some of existing online social networks fit into a simply graph representation. As an example, in Facebook two users can establish only a unique friendship relationship.

2.2 Access Control

The access control mechanism allows us to identify the users that are authorized to access a confidential resource and those that are not authorized. As the reference access control model for OSNs, we now summarize the one proposed by us in [6]. The use of this access control model is motivated by the fact that it supports all properties of other access control models for OSNs proposed so far, i.e., constraints on type, depth and trust level of the relationships identifying authorized users. According to this model, each resource to be shared in the network is protected by a set of *access rules*, denoting the users authorized to access the resource in terms of the type, depth, and trust level of existing relationships in the network. Each access rule ar has the form $ar =< rsc, AC >$, where AC is a set of *access conditions*, all of which need to be satisfied in order to get access to resource rsc. Formally, an access condition is a tuple $ac =< v, rt, d\_max, t\_min >$, where v is the network user with whom the requestor of a given resource must have a direct or indirect relationship to obtain the access, whereas rt, d_max, and t_min are, respectively, the type, maximum depth, and minimum trust level that the relationship must have in order to get the access. The trust level of a direct relationship is provided by the annotation on the corresponding arc. The trust level of an indirect relationship, which is represented by a path linking two nodes of the graph, needs to be computed based on the trust levels associated with the arcs composing the path. The literature offers several algorithms to compute the trust of indirect relationships in OSNs [9]. At any rate, the specific algorithm for trust computation is not the focus of our paper, as it is used only to find the set of users that are or are not authorized to receive a confidential resource. So, the algorithm for trust computation does not impact the proposed probability-based approach shown in Section 3. In this paper, for simplicity, we suppose that the trust level of a path is obtained by multiplying the trust levels of all the arcs in the path. In addition, if users α and β are linked by a set of paths, we take the maximum value of trust along all these paths as the value for the trust that α has in β.

We exemplify the considered access control model by means of the OSN in Figure 1. The social network is designed to support agents working for a given financial company. By using the social networking functionalities, agents are able to find updated information on the company products and share a variety of information (e.g., opinions on new products, marketing strategies, data about the sales). Moreover, agents are able to establish relationships of different types.

Relationship types are defined according to the FOAF vocabulary [3], which has been extended to model the roles agents may play in the company. Thus, for instance, agent α has a relationship of type *ManagerOf* (MOf for short) with β and a relationship of type *ColleagueOf* (COf) with γ. In the example, social network relationships can be established also based on agents' personal relationships. As an example, β has established a *FriendOf* relationship (FOf) with γ.

Moreover, according to the company business strategies, agents can also form smaller networks or groups (for instance related to products of a particular type, or denot-

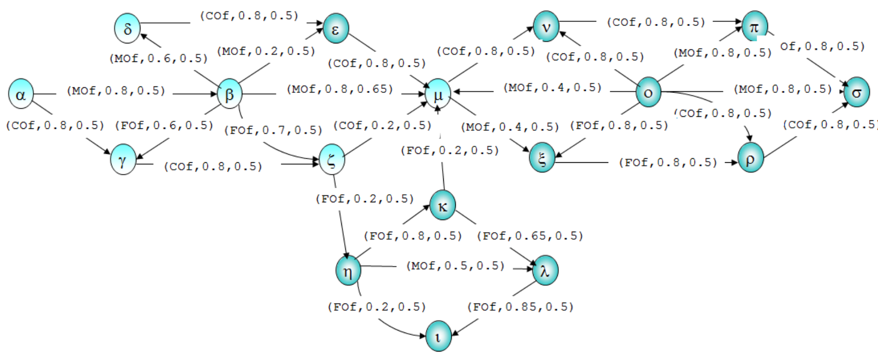

Figure 1: A portion of an OSN for a financial company

ing a partnership among some of the agents). As such, agents could have different requirements about resources sharing. For example, we can assume that agent α would like to share his/her opinions about a product (contained in the report $ProdX\_\alpha\_opinions$) with: *(1)* his/her colleagues and colleagues of his/her colleagues; *(2)* agents managed by him/her as well as agents managed by agents he/she manages. Moreover, α would like to share the report only with those nodes with whom the required relationship has a minimum trust value of 0.5. To enforce these requirements, α can specify the following access rule: $ar =< ProdX\_\alpha\_opinions, \{< \alpha, MOf, 2, 0.5 >, < \alpha, COf, 2, 0.5 >\} >$. Referring to Figure 1, the nodes that can access $ProdX\_\alpha\_opinions$ are β, γ, ζ, and μ. According to the specified access rule, ϵ is not allowed to access the report even if he or she satisfies the requirements on the relationship type, in that the trust level is 0.16, less than the 0.5 threshold required by the access rule in both access conditions.

3. PROPOSED APPROACH

We first show in Section 3.1 how we can compute the probability that a resource rsc propagates along a specific path from a specified node α to another node β. Then, in Section 3.2, we we discuss, through a few representative examples, how we can compute the probability that rsc propagates from α to β, *regardless of the specific path followed*. This leads to the explanation of the general formula and algorithm for computing this probability (3.3). Due to the computational complexity of the algorithm described in Section 3.3, we provide an upper bound to this probability in Section 3.4. Building on these concepts, Section 3.5 introduces the *Unauthorized Access Risk*, i.e., an upper bound to the probability that a resource is accessed by any unauthorized user.

3.1 Resource Propagation along a Path

We can define the probability that rsc propagates along a path in the graph denoting an OSN based on the probabilities associated with each arc. Given $path =< \alpha_1, \alpha_2, \ldots, \alpha_n >$, the probability $P(path)$ that rsc propagates from α_1 to α_n along $path$ is the conditional probability:

$$P(path) = P(\alpha_1\_makes\_rsc\_known\_to\_\alpha_n\_along\_path | \alpha_1\_knows\_rsc) \quad (2)$$

That is also computed as

$$\begin{aligned} P(path) = P(&\alpha_1\_makes\_rsc\_known\_to\_\alpha_2 \\ &\wedge \alpha_2\_makes\_rsc\_known\_to\_\alpha_n\_along\_path | \\ &\alpha_1\_knows\_rsc \wedge \alpha_1\_makes\_rsc\_directly\_known\_to\_\alpha_2) \\ &P(\alpha_1\_makes\_rsc\_directly\_known\_to\_\alpha_2) \quad (3) \end{aligned}$$

The first probability in the expression in the right-hand side of Formula (2) can be simplified as follows. The fact that $\alpha_1\_makes\_rsc\_directly\_known\_to\_\alpha_2$ is implied by the conditioning event $\alpha_1\_knows\_rsc \wedge \alpha_1\_makes\_rsc\_directly\_known\_to\_\alpha_2$, so we can remove $\alpha_1\_makes\_rsc\_known\_to\_\alpha_2$ from the conditional event and we have $P(\alpha_2\_makes\_rsc\_known\_to\_\alpha_n\_along\_path | \alpha_1\_knows\_rsc \wedge \alpha_1\_makes\_rsc\_directly\_known\_to\_\alpha_2)$. The conditioning event can be rewritten as $\alpha_1\_knows\_rsc \wedge \alpha_1\_makes\_rsc\_directly\_known\_to\_\alpha_2 \wedge \alpha_2\_knows\_rsc$. As the probability of propagation of rsc from α_2 does not depend on the previous history of rsc, $\alpha_1\_knows\_rsc \wedge \alpha_1\_makes\_rsc\_directly\_known\_to\_\alpha_2$ can be removed from the conditioning event. Also, by definition, $p(\alpha_1, \alpha_2) = p(\alpha_1\_makes\_rsc\_directly\_known\_to\_\alpha_2)$, where $p(\alpha_1, \alpha_2)$ is the probability given as an annotation of the arc from α_1 to α_2, as described in Section 2, so Formula (3) can be rewritten as

$$\begin{aligned} P(path) = P(&\alpha_2\_makes\_rsc\_known\_to\_\alpha_n\_along\_path | \\ &\alpha_2\_knows\_rsc) \\ &P(\alpha_1\_makes\_rsc\_directly\_known\_to\_\alpha_2) \quad (4) \end{aligned}$$

We can now recursively apply the same reasoning on this probability and we stop the recursion when $P(\alpha_{n-1}\_makes\_rsc\_known\_to\_\alpha_n\_along\_path | \alpha_{n-1}\_knows\_rsc)$, which is by definition equal to $P(\alpha_{n-1}, \alpha_n)$. So, $P(path)$ is actually the product of the individual probabilities of the arcs encountered along $path$, that is:

$$P(path) = \prod_{i \in 1..n-1} p(\alpha_i, \alpha_{i+1}) \quad (5)$$

3.2 Resource Propagation along a Set of Paths

Several different paths may connect two nodes α and β in an OSN. In what follows, we denote the *set of paths* that connect α to β as $\alpha \to \beta$. In this section, we show how we compute the probability $P(\alpha \to \beta)$ that rsc propagates from α to β along any path in $\alpha \to \beta$.

To this end, we use a few examples for illustration purposes. We start with the case of two paths that do not have any arc in common, even though they have the same start and end node. We then illustrate the more general case of two paths that have the same start and end node and that share at least one arc. Finally, we also discuss how to deal with paths with loops.

3.2.1 Two Paths with No Arcs in Common

In Figure 2, nodes α and γ are connected by means of two paths:[3] $path_1 =< \alpha, \beta, \gamma >$ and $path_2 =< \alpha, \gamma >$. So, we have $\alpha \to \beta = \{< \alpha, \beta, \gamma >, < \alpha, \gamma >\}$. Information may propagate from α to γ along both paths of even along one path and not the other. We assume that propagation of information along one arc is independent from propagation along any other arc. So, for instance, the propagation of rsc along $< \alpha, \gamma >$ is independent from the propagation of rsc along $< \alpha, \beta >$, and, therefore, along $< \alpha, \beta, \gamma >$.

Probability $P(\alpha \to \gamma) = P(path_1 \vee path_2)$, where $path_1 \vee path_2$ is the event that rsc propagates along $path_1$ or $path_2$, i.e., the event obtained as the disjunction of events $path_1$ and $path_2$. We can apply a general property of probabilities in the case of events built via disjunctions of events, which we rephrase for our case as follows:

$$\begin{aligned} P(path_1 \vee path_2) = \\ P(path_1) + P(path_2) - P(path_1 \wedge path_2) = \\ P(path_1) + P(path_2)(1 - P(path_1|path_2)) \end{aligned} \quad (6)$$

This general property will be later applied to the more general example of two paths with arcs in common and used in the derivation of the general formula for the computation of the propagation probability (Formula (14)).

In Figure 2, we have $P(path_1) = p(\alpha,\beta)p(\beta,\gamma)$ and $P(path_2) = p(\alpha,\gamma)$. The two paths are independent, i.e., rsc's propagation along $path_1$ is independent of rsc's propagation along $path_2$, so we also have $P(path_1|path_2) = P(path_1)$ and

$$P(\alpha \to \gamma) = p(\alpha,\beta)p(\beta,\gamma) + p(\alpha,\gamma)(1 - p(\alpha,\beta)p(\beta,\gamma)) \quad (7)$$

As a further proof, we can also compute $P(\alpha \to \gamma)$ in a different way, which we use as the basis for computing the upper bound of $P(\alpha \to \gamma)$ in Section 3.4. $P(\alpha \to \gamma)$ can be computed as the complement of probability $Q(\alpha \to \gamma) = 1 - P(\alpha \to \gamma)$ that rsc does *not* propagate from α to γ *on either path*. Since the two paths are independent, $Q(\alpha \to \gamma)$ is the product of probability $1-P(path_1) = 1-p(\alpha,\beta)p(\beta,\gamma)$ and probability $1 - P(path_1) = 1 - p(\alpha,\gamma)$, i.e.,

$$P(\alpha \to \gamma) = 1 - (1 - p(\alpha,\beta)p(\beta,\gamma))(1 - p(\alpha,\gamma)) = \quad (8)$$

$$p(\alpha,\beta)p(\beta,\gamma) + p(\alpha,\gamma)(1 - p(\alpha,\beta)p(\beta,\gamma)) \quad (9)$$

3.2.2 Two Paths with Arcs in Common

However, it is not always the case that paths are independent. In the general case, two paths connecting α and β may very well have arcs in common, so they are not independent.

The two paths $path_1 =< \delta, \alpha, \beta, \gamma >$ and $path_2 =< \delta, \alpha, \gamma >$ from δ to γ share arc $< \delta, \alpha >$, so they are not independent. We can apply Formula (6), where $P(path_1) = p(\delta,\alpha)p(\alpha,\beta)p(\beta,\gamma)$ and $P(path_2) = p(\delta,\alpha)p(\alpha,\gamma)$. We now need to compute $P(path_1|path_2)$ to complete the formula. $P(path_1|path_2)$ is the probability that rsc propagates along $path_1$, once it is already known that rsc propagates along $path_2$. Thus, it is the probability that rsc propagates along

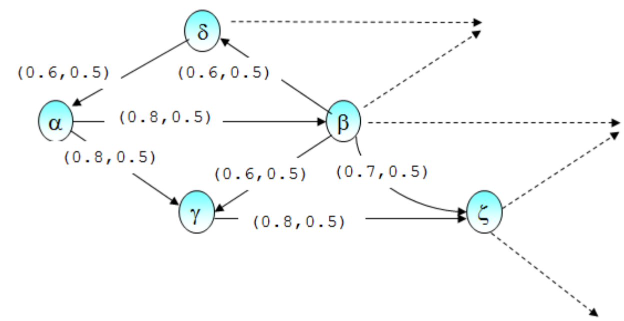

Figure 2: A fragment of an OSN

$< \delta, \alpha >$, $< \alpha, \beta >$, and $< \beta, \gamma >$, once it is known that it propagates along $< \delta, \alpha >$ and $< \alpha, \gamma >$. So it is the probability that rsc propagates along $< \alpha, \beta >$ and $< \beta, \gamma >$, since we already know rsc propagates along $< \delta, \alpha >$. Summarizing, we have:

$$\begin{aligned} P(\delta \to \gamma) = \\ p(\delta,\alpha)p(\alpha,\beta)p(\beta,\gamma) + p(\delta,\alpha)p(\alpha,\gamma)(1 - p(\alpha,\beta)p(\beta,\gamma)) = \\ p(\delta,\alpha)(p(\alpha,\beta)p(\beta,\gamma) + p(\alpha,\gamma)(1 - p(\alpha,\beta)p(\beta,\gamma))) = \\ p(\delta,\alpha)P(\alpha \to \gamma) \end{aligned} \quad (10)$$

The right-hand part of the last equality in Formula (10) shows that Formula (6) gives results that are consistent with what one may already expect. $P(\delta \to \gamma)$ is the product of the probability $p(\delta,\alpha)$ that rsc propagates from δ to α and the probability $P(\alpha \to \gamma)$ that rsc propagates from α to γ.

3.2.3 Dealing with Loops

Some care needs to be exercised when cycles are present in the graph, but, as we now show with an example, the result will actually be a simplification of the graph. Suppose we have the graph in Figure 3 i.e., a graph with an "entry" node α, a loop $< \beta, \gamma, \delta, \beta >$, and an "exit" node ϵ. The computation of $P(\alpha \to \epsilon)$ can be broken down as the product of three probabilities:

$$P(\alpha \to \epsilon) = P(\alpha \to \beta)P(\beta \to \gamma)P(\gamma \to \epsilon) \quad (11)$$

Set $\beta \to \gamma$ contains an infinite number of paths, because of the presence of loop $< \beta, \gamma, \delta, \beta >$. However, no paths that contain a loop need to be taken into account for our goals. Suppose that rsc has reached node γ along path $< \alpha, \beta, \gamma >$. The probability that rsc reaches γ along that path is $P(< \alpha, \beta, \gamma >)$. The probability that rsc is known by γ after one iteration of the loop is:

$$\begin{aligned} P(\alpha, \beta, \gamma, \delta, \beta, \gamma) = P(\gamma, \delta, \beta, \gamma|\alpha, \beta, \gamma)P(\alpha, \beta, \gamma) = \\ P(\alpha, \beta, \gamma)P(\gamma, \delta, \beta, \gamma) \end{aligned} \quad (12)$$

However, according to the meaning of our probabilities, $P(\alpha \to \beta)$ is the probability that, if rsc is known at node α, it also gets known at node β. So, $P(\alpha \to \alpha) = 1$. As a consequence, $P(< \gamma, \delta, \beta, \gamma >) = 1$, and:

$$P(< \alpha, \beta, \gamma, \delta, \beta, \gamma >) = P(< \alpha, \beta, \gamma >) \quad (13)$$

Thus, when computing $P(\alpha \to \beta)$, we can ignore all loops in $\alpha \to \beta$, and $\alpha \to \beta$ can be reduced to the paths in the hierarchy (i.e., the directed acyclic graph) in which α is not preceded by any other node and β is not followed by any

[3]Here and in the following figures, for simplicity we omitt the relationship type informationa associated with an arc.

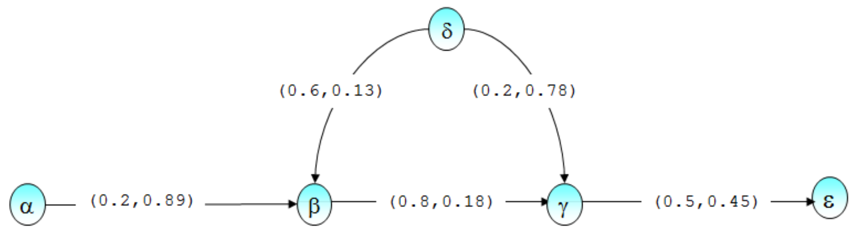

Figure 3: An example of graph with a loop

other node. Thus, we deal with a finite set of paths. Once the hierarchy from α to β is known, we can build $P(\alpha \to \beta)$ by starting from α and proceeding down the levels of the hierarchy.

3.3 Exact Computation of the Probability of Propagation along a Set of Paths

We can now show what happens in the general case, and how the probability of information propagating from one node to another node can be computed in a recursive manner. Given two nodes α and β, let us suppose that $\alpha \to \beta$ is composed of n paths $path_1, path_2, \ldots, path_n$. We compute $P(path_1 \vee path_2 \vee \ldots \vee path_n)$, where, from a logical point of view, $path_1 \vee path_2 \vee \ldots \vee path_n$ is a formula in Disjunctive Normal Form containing $n-1$ disjunction operators and each term $path_i$ is a conjunction of k_i predicates, each denoting the fact that rsc propagates along a specific arc of $path_i$. For instance, in Figure 1, $path_1$ can be also represented as the conjunction of the two predicates $prop_{<\alpha,\beta>}$, which denotes the fact that rsc propagates on arc $<\alpha,\beta>$, and $prop_{<\beta,\gamma>}$, which denotes the fact that rsc propagates on arc $<\beta,\gamma>$. So, we can write $path_1 = prop_{<\alpha,\beta>} \wedge prop_{<\beta,\gamma>}$. Likewise, $path_2 = prop_{<\alpha,\gamma>}$, and $path_1 \vee path_2 = prop_{<\alpha,\beta>} \wedge prop_{<\beta,\gamma>} \vee prop_{<\alpha,\gamma>}$. Because of the general property of probabilities of Formula (6), we can write:

$$P(path_1 \vee path_2 \vee \ldots \vee path_n) = P(path_1 \vee path_2 \vee \ldots \vee path_{n-1}) + P(path_n)(1 - P(path_1 \vee path_2 \vee \ldots \vee path_{n-1} | path_n)) \quad (14)$$

Let us examine the terms appearing in the formula.

- $P(path_n)$ can be computed directly as shown in Section 3.1.
- $P(path_1 \vee path_2 \vee \ldots \vee path_{n-1})$ can be computed recursively, by applying Formula (14) to the set of paths $\{path_1, path_2, \ldots, path_{n-1}\}$, which contains one less path than the initial set of paths, so recursion is guaranteed to end when the set of paths contains only one path.
- $P(path_1 \vee path_2 \vee \ldots \vee path_{n-1} | path_n)$ can be first simplified and then computed recursively. As for the simplification part, $P(path_1 \vee path_2 \vee \ldots \vee path_{n-1} | path_n)$ is the probability that rsc propagates along at least one path in $\{path_1, path_2, \ldots, path_{n-1}\}$ once it is known that rsc propagates along $path_n$. For instance, let us take the example in Figure 2 and let us show with logical arguments that $P(path_1 | path_2) = p(\alpha,\beta)\, p(\beta,\gamma)$ in that case, as we have already shown when we discussed Formula (10). The two paths from δ to γ can be rephrased in logical terms as $path_1 = prop_{\delta,\alpha} \wedge prop_{\alpha,\beta} \wedge prop_{\beta,\gamma}$ and $path_2 = prop_{\delta,\alpha} \wedge prop_{\alpha,\gamma}$. So, $P(path_1 | path_2) = P(prop_{\delta,\alpha} \wedge prop_{\alpha,\beta} \wedge prop_{\beta,\gamma} | prop_{\delta,\alpha} \wedge prop_{\alpha,\gamma})$. The conditioning event $prop_{\delta,\alpha} \wedge prop_{\alpha,\gamma}$ is assumed to occur, so both $prop_{\delta,\alpha}$ and $prop_{\alpha,\gamma}$ are true. So, we can set $prop_{\delta,\alpha}$ and $prop_{\alpha,\gamma}$ to true in the conditional event (i.e., since only $prop_{\delta,\alpha}$ appears in the conditional event, we removed it from the conditional event) and the conditioning event, so $P(path_1 | path_2) = P(prop_{\alpha,\beta} \wedge prop_{\beta,\gamma}) = p(\alpha,\beta)p(\beta,\gamma)$. From a logical point of view, we can replace $P(path_1 \vee path_2 \vee \ldots \vee path_{n-1} | path_n)$ with $P(path'_1 \vee path'_2 \vee \ldots \vee path'_{n-1})$, in which each single conjunction $path'_i$ is obtained by eliminating from the corresponding conjunction $path_i$ all those predicates that also appear in $path_n$, because it is assumed that those predicates are true, so they need not be evaluated when evaluating the truth value of $path_i$. As a consequence, the new probability $P(path'_1 \vee path'_2 \vee \ldots \vee path'_{n-1})$ is based on a predicate which is built as
  - a formula in Disjunctive Normal Form containing $n-2$ disjunction operators, one less than the original formula
  - and each term $path'_i$ is a conjunction of k'_i predicates, with $k'_i \leq k_i$, where k_i denotes the number of predicates in $path_n$ and k'_i the number of predicates in $path'_n$.

  Thus, we can apply Formula (14) to $P(path'_1 \vee path'_2 \vee \ldots \vee path'_{n-1})$, and recursion is guaranteed to end.

Thus, we have found out a recursive algorithm for computing $P(\alpha \to \beta)$, regardless of the path along which rsc propagates from α to β. However, the computational complexity of the algorithm may be too high, as we now show. The number of recursions clearly depends on the number of paths. Suppose we have a hierarchy with $n+2$ nodes, i.e., with one initial node, one terminal node, and n intermediate nodes. Suppose that this hierarchy has $l+2$ levels and that each level has the same number of nodes, i.e., $n = a \cdot l$, except for the initial and the terminal levels. Suppose also that there is an arc from each node at level j to each node at level $j+1$. Then, it can be shown that the number of paths for this graph is $(\frac{n}{l})^l = a^{\frac{n}{a}}$. So, the number of paths grows exponentially with n, in this case.

3.4 An Upper Bound to the Probability of Propagation along a Set of Paths

Since the computational complexity for the exact computation of $P(\alpha \to \beta)$ may be too high, we here derive an upper bound for it. To this end, let us take $\alpha \rightarrow \beta = \{path_1, path_2, \ldots, path_n\}$ like we did in Section 3.2, so we can use $P(path_1 \vee path_2 \vee \ldots \vee path_n)$ in our derivation.

For notational convenience, let $Pre(\beta)$ be the set of nodes in the "preset" of β, i.e., the set of those nodes ζ that have a direct arc to β, i.e., $<\zeta, \beta> \in A$. We first show that:

$$P(\alpha \rightarrow \beta) \leq 1 - \prod_{\zeta \in Pre(\beta)} (1 - P(\alpha \rightarrow \zeta) p(\zeta,\beta)) \qquad (15)$$

We can write $P(\alpha \rightarrow \beta)$ as follows:

$$P(\alpha \rightarrow \beta) = P(\alpha \rightarrow \zeta_1 \rightarrow \beta \vee \ldots \vee \alpha \rightarrow \zeta_n \rightarrow \beta) \qquad (16)$$

Formula (16) shows that the probability that rsc propagates from α to β is the probability that it propagates on at least one path that goes from α to β through one of the $\zeta_i \in Pre(\beta)$. Based on the probability properties of disjunctions (that we already used in Formula (6)), we can also write that:

$$P(\alpha \rightarrow \beta) = P(\alpha \rightarrow \zeta_1 \rightarrow \beta \vee \ldots \vee \alpha \rightarrow \zeta_{n-1} \rightarrow \beta) + P(\alpha \rightarrow \zeta_n \rightarrow \beta) (1 - P(\alpha \rightarrow \zeta_1 \rightarrow \beta \vee \ldots \vee \alpha \rightarrow \zeta_{n-1} \rightarrow \beta | \alpha \rightarrow \zeta_n \rightarrow \beta)) \qquad (17)$$

Now, we have:

$$P(\alpha \rightarrow \zeta_1 \rightarrow \beta \vee \ldots \vee \alpha \rightarrow \zeta_{n-1} \rightarrow \beta | \alpha \rightarrow \zeta_n \rightarrow \beta) \geq P(\alpha \rightarrow \zeta_1 \rightarrow \beta \vee \ldots \vee \alpha \rightarrow \zeta_{n-1} \rightarrow \beta) \qquad (18)$$

because knowing that rsc propagates from α to β via ζ_n will never decrease the probability of it propagating along any other paths. As we already discussed, knowing that rsc propagates from α to β via ζ_n means that some of the predicates in the paths in $\alpha \rightarrow \zeta_n \rightarrow \beta$ are true, so they can be removed from the conditional event $\alpha \rightarrow \zeta_1 \rightarrow \beta \vee \ldots \vee \alpha \rightarrow \zeta_{n-1} \rightarrow \beta$. This implies that the probability of propagation of rsc along the paths in $\alpha \rightarrow \zeta_1 \rightarrow \beta \bigcup \ldots \bigcup \alpha \rightarrow \zeta_{n-1} \rightarrow \beta$ may increase, but never decrease.

As a consequence, we can write:

$$\begin{aligned} P(\alpha \rightarrow \beta) &\leq P(\alpha \rightarrow \zeta_1 \rightarrow \beta \vee \ldots \vee \alpha \rightarrow \zeta_{n-1} \rightarrow \beta) + P(\alpha \rightarrow \zeta_n \rightarrow \beta) \\ &\quad (1 - P(\alpha \rightarrow \zeta_1 \rightarrow \beta \vee \ldots \vee \alpha \rightarrow \zeta_{n-1} \rightarrow \beta)) = P(\alpha \rightarrow \zeta_n \rightarrow \beta) + \\ &\quad Q(\alpha \rightarrow \zeta_n \rightarrow \beta) P(\alpha \rightarrow \zeta_1 \rightarrow \beta \vee \ldots \vee \alpha \rightarrow \zeta_{n-1} \rightarrow \beta) = \\ &\quad 1 - Q(\alpha \rightarrow \zeta_n \rightarrow \beta) Q(\alpha \rightarrow \zeta_1 \rightarrow \beta \vee \ldots \vee \alpha \rightarrow \zeta_{n-1} \rightarrow \beta) \end{aligned} \qquad (19)$$

which can be rewritten as:

$$Q(\alpha \rightarrow \beta) \geq Q(\alpha \rightarrow \zeta_n \rightarrow \beta) Q(\alpha \rightarrow \zeta_1 \rightarrow \beta \vee \ldots \vee \alpha \rightarrow \zeta_{n-1} \rightarrow \beta) \qquad (20)$$

We can now apply the same reasoning to $Q(\alpha \rightarrow \zeta_1 \rightarrow \beta \vee \ldots \vee \alpha \rightarrow \zeta_{n-1} \rightarrow \beta)$, so we obtain:

$$Q(\alpha \rightarrow \beta) \geq \prod_{\zeta \in Pre(\beta)} Q(\alpha \rightarrow \zeta \rightarrow \beta) \qquad (21)$$

which can be rewritten as:

$$P(\alpha \rightarrow \beta) \leq 1 - \prod_{\zeta \in Pre(\beta)} (1 - P(\alpha \rightarrow \zeta) p(\zeta,\beta)) \qquad (22)$$

since $Q(\alpha \rightarrow \zeta \rightarrow \beta) = 1 - P(\alpha \rightarrow \zeta) p(\zeta,\beta)$.

However, computing $P(\alpha \rightarrow \zeta)$ would imply enumerating all the paths in $\alpha \rightarrow \zeta$, whose computational complexity

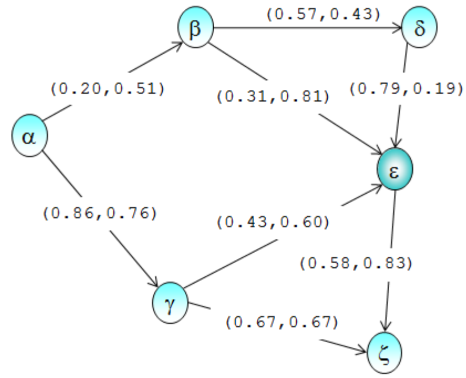

Figure 4: An example hierarchy

Table 1: Probability upper bounds and trust for the nodes in Figure 4

Node	UB	UB'	$Trust$
α	1.00	1.00	1.00
β	0.51	0.51	0.20
γ	0.76	0.76	0.86
δ	0.2193	N/A	0.114
ϵ	0.7435	0.456	0.3698
ζ	0.8121	0.5092	0.5762

may be too high. So, we introduce another approximation, based on the fact that, if we select $UB(\alpha \rightarrow \zeta) \geq P(\alpha \rightarrow \zeta)$ (UB as in Upper Bound) we have:

$$\begin{aligned} P(\alpha \rightarrow \beta) &\leq 1 - \prod_{\zeta \in Pre(\beta)} (1 - P(\alpha \rightarrow \zeta) p(\zeta,\beta)) \\ &\leq 1 - \prod_{\zeta \in Pre(\beta)} (1 - UB(\alpha \rightarrow \zeta) p(\zeta,\beta)) \end{aligned} \qquad (23)$$

So, we need to build UB for all nodes. Here is one possibility:

$$UB(\alpha \rightarrow \beta) = 1 - \prod_{\zeta \in Pre(\beta)} (1 - UB(\alpha \rightarrow \zeta) p(\zeta,\beta)) \qquad (24)$$

with $UB(\alpha \rightarrow \beta) = p(\alpha,\beta)$ for all those nodes β such that $Pre(\beta) = \{\alpha\}$, i.e., whose only node in the preset is α. Thus, we start from α and its successor nodes, and proceeding level by level in the hierarchy we can build function UB for all nodes. For instance, for the hierarchy in Figure 4, we obtain the values for UB reported in Table 1. (In this section, we only deal with column UB. The meaning of the other two columns will be illustrated in Section 3.5.)

Let us show how the computations of the values of UB were carried out for the nodes in Figure 1. Obviously, $UB(\alpha) = 1$, as α is the original owner of the resource. The values $UB(\beta) = 0.51 = p(\alpha,\beta)$ and $UB(\gamma) = 0.76 = p(\alpha,\gamma)$ can be computed directly as α is directly linked to β and to γ. At any rate, by using Formula (24), we also obtain $UB(\beta) = 1 - (1 - UB(\alpha) \cdot p(\alpha,\beta)) = p(\alpha,\beta)$ and $UB(\gamma) = 1 - (1 - UB(\alpha) \cdot p(\alpha,\gamma)) = p(\alpha,\gamma)$. Again, $UB(\delta) = 0.76 = p(\alpha,\beta) \cdot p(\beta,\delta)$ can be computed based on the probabilities associated with the arcs, because there is only one path from α to δ. Alternatively, via Formula (24), we also obtain $UB(\delta) = 1 - (1 - UB(\beta) \cdot p(\beta,\delta)) = p(\alpha,\beta) \cdot p(\beta,\delta)$. Let us now focus on $UB(\epsilon)$, which we compute based on

Formula (24), i.e., $UB(\epsilon) = 1 - (1 - UB(\beta)p(\beta,\epsilon))(1 - UB(\delta)p(\delta,\epsilon))(1 - UB(\gamma)p(\gamma,\epsilon))$. Likewise, $UB(\zeta) = 1 - (1 - UB(\gamma)p(\gamma,\zeta))(1 - UB(\epsilon)p(\epsilon,\zeta))$.

The value of $UB(\alpha \to \beta)$ obtained is a sharp approximation, as it does coincide with the real value of $P(\alpha \to \beta)$ whenever $\alpha \to \beta$ contains only independent paths, like the ones of the example of Section 3.2.1.

The computation of $UB(\alpha \to \beta)$ according to Formula (24) involves a number of multiplications that is quadratic with the number of nodes, as we now show. The computation of $UB(\alpha \to \beta)$ involves a number of multiplications equal to the number of incoming arcs of β, once the values of $UB(\alpha \to \zeta)$ are known for all ζ in $Pre(\beta)$. Likewise, the number of multiplications needed to compute the values of $UB(\alpha \to \zeta)$ for all of these ζ's is equal to the number of the incoming arcs of all of the ζ's, once the values of $UB(\alpha \to \tau)$ are known for all τ in their presets. By proceeding backwards from β) to α, we obtain that the total number of multiplications needed to compute $UB(\alpha \to \beta)$ is equal to the sum of the number of the incoming arcs of all the nodes in $\alpha \to \beta$. Since the sets of incoming arcs of two different nodes are obviously disjoint, we have that the number of of multiplications needed to compute $UB(\alpha \to \beta)$ is equal to the number of arcs in $\alpha \to \beta$, which grows quadratically with the number of nodes.

3.5 Unauthorized Access Risk

We here introduce the *Unauthorized Access Risk* ($UAR(ar)$) as the probability that, given an access rule ar, a resource is passed to any unauthorized user. $UAR(ar)$ depends on the probability of propagation of the resource across the OSN, as defined in Section 3 and on the considered access rule (see Section 2). The intuition behind the definition of UAR is the following. An access rule identifies a set of authorized users and, consequently, a set of unauthorized users. An unauthorized release of a resource happens when a user not authorized by any access rules receives the resource. From that moment on, the resource can be always illegally propagated. Clearly, if an unauthorized user receives a resource, then there is at least an authorized user that passes the resource to him or her. This may happen only if there is a relationship in the OSN linking the authorized user to the unauthorized one. Therefore, we can quantify the UAR as the probability that any unauthorized user linked to at least one authorized user receives the resource from the latter.

Let $Auth(ar) \subseteq N$ be the set of authorized nodes and $UnAuth(ar) \subseteq N$ be the set of nodes not authorized by an access rule ar, given the set of nodes N and a resource rsc. Also, let $BorderUnAuth(ar) \subseteq UnAuth(ar)$ be the set of unauthorized nodes on the border with the authorized nodes, more precisely, $BorderUnAuth(ar)$ is the set of unauthorized nodes in whose preset there is at least one authorized node, i.e.,

$$BorderUnAuth(ar) = \{\alpha \in N | Pre(\alpha) \cap Auth(ar) \neq \emptyset\} \quad (25)$$

where, $Pre(\alpha) = \{\beta | <\beta, \alpha> \in A\}$. We define UAR as the probability that any node in $BorderUnAuth(ar)$ receives rsc. Once rsc is known to any of these nodes, it can be always propagated in an unauthorized way.

Based on these definitions, $UAR(ar)$ is defined as in Formula (26):

$$UAR(ar) = P(\bigvee_{\beta \in BorderUnAuth(ar)} \alpha \to \beta) \quad (26)$$

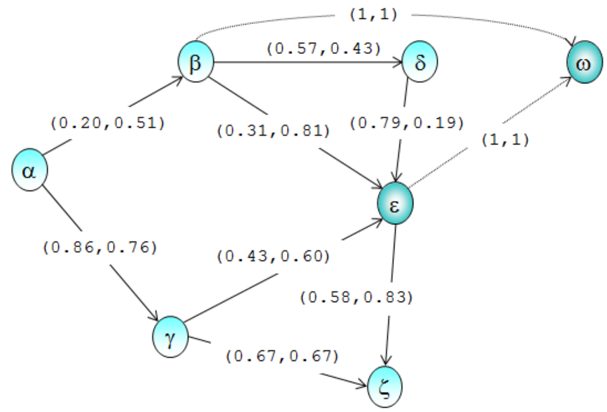

Figure 5: An example hierarchy with node ω

A first upper bound for $UAR(ar)$ can be computed as follows, by directly using the upper bound approximation derived in Section 3.4. Once the nodes in $BorderUnAuth(ar)$ have been identified, suppose we introduce an additional node ω and an arc from each node in $BorderUnAuth(ar)$ to node ω associated with a probability 1 of information propagation. We can compute UAR as the probability that rsc propagates from α to ω, according to the formulas in Section 3.1 and we can find an upper bound for it according to the procedure shown in Section 3.4.

For instance, take the example OSN in Figure 4 and suppose that the access rule specifies that authorized nodes need to have at least a level of trust of 0.5 and may have a maximum distance from α of 4. As the maximum distance between nodes in this hierarchy is 4, the nodes in $BorderUnAuth(ar)$ are those with a trust level lower than 0.5. As column *trust* in Table 1 shows, we have $BorderUnAuth(ar) = \{\beta,\epsilon\}$. Note that δ does not belong to $BorderUnAuth(ar)$ because none of the arcs in its preset is an authorized node, i.e., δ can only receive rsc from β, which is already an unauthorized node. Figure 5 is a modification of Figure 4, in which ω and the arcs that lead to it are represented with dashed lines, to pictorially denote the fact that they do not belong to the original graph. At any rate, if rsc is propagated to any node in $BorderUnAuth(ar)$, it is also propagated to ω with certainty, and, *vice versa*, if rsc is propagated to ω, then it must have been propagated to at least one node in $BorderUnAuth(ar)$.

We now show how we can compute an even stricter upper bound for the value of $UAR(ar)$, which, however, may require some additional computations. This is the upper bound we will use in the application example of Section 4. Since we are dealing with hierarchies, we may suppose that the nodes in the entire hierarchy are ordered, and we can index them in such a way that, given two values i and j, with $i < j$, then there may be a direct or indirect relationship from $node_i$ to node $node_j$, but no relationship from $node_j$ to node $node_i$. Therefore, we can extract the sub-ordering of the nodes in $BorderUnAuth(ar)$ from the general ordering of the nodes in the hierarchy and use a specific indexing from 1 to $bua = |BorderUnAuth(ar)|$ when dealing only with the nodes in $BorderUnAuth(ar)$. Thus, we can rewrite Formula

26 as follows:

$$UAR(ar) = P(\bigvee_{i \in \{1..bua\}} (\alpha \rightarrow \beta_i)) \quad (27)$$

$UAR(ar)$ can also be computed as the complement of the probability that rsc does not propagate to any of the nodes in $BorderUnAuth(ar)$, i.e.,

$$UAR(ar) = 1 - P(\bigwedge_{i \in \{1..bua\}} \neg(\alpha \rightarrow \beta_i)) \quad (28)$$

where $\neg(\alpha \rightarrow \beta_i)$ denotes the fact that the resource does not propagate from α to β_i. Based on the properties of conditional probabilities, we can also write:

$$UAR(ar) = 1 - P(\bigwedge_{i \in \{2..bua\}} \neg(\alpha \rightarrow \beta_i) | \neg(\alpha \rightarrow \beta_1)) \\ P(\neg(\alpha \rightarrow \beta_1)) \quad (29)$$

As for the right-hand side of Formula 29, note that we can compute an upper bound for $P(\alpha \rightarrow \beta_1)$ based on the results of Section 3.4. So, we can compute a lower bound for $P(\neg(\alpha \rightarrow \beta_1)) = 1 - P(\alpha \rightarrow \beta_1)$, which leads to this first majorization of $UAR(ar)$:

$$UAR(ar) \leq 1 - P(\bigwedge_{i \in \{2..bua\}} \neg(\alpha \rightarrow \beta_i) | \neg(\alpha \rightarrow \beta_1)) \\ (1 - UB(\alpha \rightarrow \beta_1)) \quad (30)$$

We now show how a lower bound approximation can be found for $P(\bigwedge_{i \in \{2..bua\}} \neg(\alpha \rightarrow \beta_i) | \neg(\alpha \rightarrow \beta_1))$. This is the probability that none of the nodes β_i in $BorderUnAuth(ar)$ except β_1 receives rsc, conditioned on the fact that β_1 has not received rsc. Again, based on probability properties:

$$P(\bigwedge_{i \in \{2..bua\}} \neg(\alpha \rightarrow \beta_i | \neg(\alpha \rightarrow \beta_1)) = \\ 1 - P(\bigvee_{i \in \{2..bua\}} (\alpha \rightarrow \beta_i) | \neg(\alpha \rightarrow \beta_1)) \quad (31)$$

i.e., it is the complement of the probability that at least one of the nodes β_i in $BorderUnAuth(ar)$ (except β_1) receives rsc, conditioned by the fact that β_1 has not received rsc. The problem now becomes finding an upper bound approximation for $P(\bigvee_{i \in \{2..bua\}} (\alpha \rightarrow \beta_i) | \neg(\alpha \rightarrow \beta_1))$. This upper bound approximation can be found by "ignoring" the presence of β_1 in the graph, i.e., by computing $P(\bigvee_{i \in \{2..bua\}} (\alpha \rightarrow \beta_i))$ in a new graph obtained from the original one by removing β_1. To provide an intuitive justification for the fact that we obtain an upper bound, take the two sets of paths $\alpha \rightarrow \beta_1$ and $\alpha \rightarrow \beta_2$ and suppose that some paths in $\alpha \rightarrow \beta_1$ have arcs in common with at least one path $path_{\beta_2}$ in $\alpha \rightarrow \beta_2$. We know that rsc has not reached β_1 and that may have happened because rsc did not follow one of the arcs in $path_{\beta_2}$ in common with the paths in $\alpha \rightarrow \beta_1$. So the probability of $path_{\beta_2}$, once it is known that rsc has not reached β_1, is lower than the probability of $path_{\beta_2}$ if β_1 simply did not exist. This line of reasoning can be extended to all other paths in $\alpha \rightarrow \beta_2$ and to all other β_is, with $i \in 3..bua$. So, we can compute an upper bound approximation for $UAR(ar)$ as follows:

$$UAR(ar) \leq 1 - P'(\bigwedge_{i \in \{2..bua\}} \neg(\alpha \rightarrow \beta_i))(1 - UB(\alpha \rightarrow \beta_1) \\ = 1 - (1 - P'(\bigvee_{i \in \{2..bua\}} (\alpha \rightarrow \beta_i)))(1 - UB(\alpha \rightarrow \beta_1) \quad (32)$$

where $P'(\bigwedge_{i \in \{2..bua\}} \neg(\alpha \rightarrow \beta_i))$ and $P'(\bigvee_{i \in \{2..bua\}} (\alpha \rightarrow \beta_i))$ are computed as if β_1 did not exist. Note that $P'(\bigvee_{i \in \{2..bua\}} (\alpha \rightarrow \beta_i)) \leq P(\bigvee_{i \in \{2..bua\}} (\alpha \rightarrow \beta_i))$, which would be the probability computed by taking into account β_1 as well. As Formula (32) shows, we now need to find an upper bound approximation for $P'(\bigvee_{i \in \{2..bua\}} (\alpha \rightarrow \beta_i))$, which we can obtain by recursively applying the same technique until all the nodes in $BorderUnAuth(ar)$ have been taken into account. We use $UB'(\alpha \rightarrow \beta_i)$ to denote the value of the upper bound obtained in this way.

For instance, take the OSN in Figure 4, for which we have $BorderUnAuth(ar) = \{\beta, \epsilon\}$. The resulting UB's are reported in Table 1. Note that, none of the values of UB' is greater than the corresponding UB value, and the difference appears to be significant in some cases. Also recall that node δ does not belong to $BorderUnAuth(ar)$, so we do not compute UB' for it. In this example, node β precedes node ϵ. We first compute the value of $UB'(\alpha \rightarrow \beta)$, which actually coincides with $UB(\alpha \rightarrow \beta)$, as β is not preceded by any node in $BorderUnAuth(ar)$. To compute the value of $UB'(\alpha \rightarrow \epsilon)$, we just need to remove β and all of its incoming and outgoing arcs from the graph.

We can also interpret this in a different way. Suppose we are computing an upper bound to the probability that rsc reaches β or ϵ and we are looking for an upper bound of the probability of rsc reaching ϵ. We should discard the possibility that ϵ receives rsc from β, because β is already an unauthorized node, so rsc would have already reached the unauthorized region of the graph.

4. A SIMULATION EXAMPLE

We have conducted several experiments in order to evaluate the effectiveness of UAR, and specifically its upper bound UB' that we derived in Section 3.4. As a dataset, we have considered a synthetic social network which has been generated by randomly creating relationships of 34 different types, among about 200 nodes.[4] The obtained OSN has the following features: 200 nodes, an average outdregee of 200 (note that this outdegree is for all the 34 relationship types), and 24.800 relationships. We limit the OSN at 200 nodes as we do not need a big graph to show the effectiveness of UAR, as this mainly depends on nodes authorized by the considered access control policies. We believe 200 nodes are enough to include such a set. Moreover, since the key reference scenario for our measures is Enterprise 2.0, we do not expect huge graphs as the ones of general purpose social networks, like Facebook. In the synthetic OSN, each arch has randomly associated a relationship type and a trust value. In contrast, the probabilities of propagation along the arcs have been set up on the basis of the experiments.

In what follows, we report the results of two experiments, in both of which we have considered an access control policy consisting of a single access condition of the form <v,

[4]We have exploited the RELATIONSHIP vocabulary available at http://purl.org/vocab/relationship.

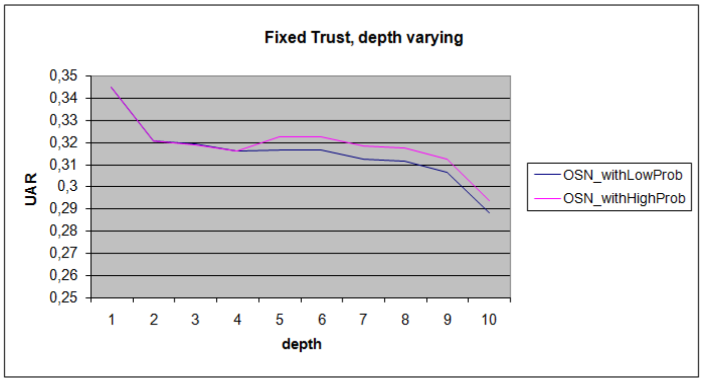

Figure 6: UAR values for ac=<v, Fof, d_max, 0.5>

Fof, d_max, t_min>,[5] where v and t_min are fixed, whereas d_max varies. More precisely, in the experiment reported in Figure 6 we have fixed t_min to 0.5, i.e., <v, Fof, d_max, 0.5>. The experiments have been conducted by considering two datasets. The first is a synthetic OSN, called $OSN\_withLowProb$, where relationships, nodes and trust values have been generated as described above and all the arcs have a low probability of propagation (i.e., less than 0.1), to simulate an OSN with a very low probability of passing information in an unauthorized way. Figure 6 confirms what we expect as the UAR general trend. In general, if a resource is publicly available, the obvious consequence is that no illegal propagation is enacted. As such, the corresponding UAR value is close to zero. Figure 6 gives us a proof of this. As the depth of the rule increases most of the 200 nodes of the OSN become authorized by the access condition, with always less users that are no authorized to access the resource. The decreasing of unauthorized users reflects in UAR, as this also reduces.

In the second dataset, called $OSN\_withHighProb$, we have set to an high level (i.e., a value greater than 0.9) the probability of propagation of about 10% of the nodes in the OSN. The aim of this experiment is to show how UAR detects this anomaly. As such, rather than randomly selecting the nodes whose probability have to be increased, we decided to select them in a particular area, to check if UAR shows this anomaly. In particular, we select 20 nodes among those with distance 5 to node v. As expected, the UAR measure detects these nodes, as confirmed by the jump between trends in Figure 6.

5. CONCLUSIONS AND FUTURE WORK

Access control for OSNs is becoming an urgent need and this has resulted in the definition of many access control models and mechanisms. Almost all of them exploit topology-based access control, according to which confidentiality requirements wrt resource release are defined in terms of the relationships in the network, their depth and trust level. Although topology-based access control is very powerful in terms of the access control requirements it can model, it is also true that, on the other hand, it may be difficult for the user specifying a policy to clearly understand its effects and the potential risks of unauthorized information leakage it may cause. To address this issue, in this paper, we have proposed a probabilistic-based approach to estimate illegal leakage of resources in an OSN where access control is regulated according to the topology-based paradigm.

We believe this represents just the core component of a more comprehensive framework to handle illegal information flow in OSNs. As such, we plan to extend this work along several directions. A first direction regards the investigation of several functions to compute the probability of resource propagation, taking into account different dimensions of the social network graph (e.g., user reputation, relationship semantics) as well as resource properties (e.g., content, history) . Moreover, we plan to extend the probability model such to consider also multigraph where indirect relationships can be represented with paths consisting of edges having different relationship types.

[5] Note that, as the relationship types have been uniformly distributed, there exist an average of 730 arcs of Fof type.

6. ACKNOWLEDGMENTS

The research presented in this article was partially funded by the IST project QualiPSo, sponsored by the EU in the 6th FP (IST-034763); the FIRB project ARTDECO, sponsored by the Italian Ministry of Education and University; the project "La qualità nello sviluppo software," sponsored by the Università degli Studi dell'Insubria; and the PRIN project ANONIMO, sponsored by the Italian Ministry of Education and University. The work by Elena Ferrari was partially supported by a Google Research Award.

7. REFERENCES

[1] B. Ali, W. Villegas, and M. Maheswaran. A trust based approach for protecting user data in social networks. In *Proceedings of the 2007 Conference of the Center for Advanced Studies on Collaborative research (CASCON'07)*, pages 288–293, 2007.

[2] M. M. Anwar, P. W. L. Fong, X.-D. Yang, and H. J. Hamilton. Visualizing privacy implications of access control policies in social network systems. In *4th International Workshop, DPM 2009 and Second International Workshop, SETOP 2009*, pages 106–120, 2009.

[3] D. Brickley and L. Miller. FOAF vocabulary specification. RDF Vocabulary Specification, July 2005.

[4] B. Carminati, E. Ferrari, R. Heatherly, M. Kantarcioglu, and B. Thuraisingham. A semantic web based framework for social network access control. In *SACMAT '09: Proceedings of the 14th ACM symposium on Access control models and technologies*, pages 177–186, New York, NY, USA, 2009. ACM.

[5] B. Carminati, E. Ferrari, M. Kantarcioglu, and B. Thuraisingham. *Privacy-Aware Knowledge Discovery: Novel Applications and New Techniques*, chapter Privacy protection of personal data in social networks. Chapman & Hall/CRC Data Mining and Knowledge Discovery Series, 2010.

[6] B. Carminati, E. Ferrari, and A. Perego. Enforcing access control in web-based social networks. *ACM Transactions on Information & System Security*, 13(1):1–38, 2009.

[7] J. Domingo-Ferrer, A. Viejo, F. Sebé, and I. González-Nicolás. Privacy homomorphisms for social networks with private relationships. *Comput. Netw.*, 52(15):3007–3016, 2008.

[8] N. Elahi, M. Chowdhury, and J. Noll. Semantic access control in web based communities. In *ICCGI '08: Proceedings of the 2008 The Third International Multi-Conference on Computing in the Global Information Technology (iccgi 2008)*, pages 131–136, Washington, DC, USA, 2008. IEEE Computer Society.

[9] J. Golbeck. *Computing and applying trust in Web-based social networks*. PhD thesis, Graduate School of the University of Maryland, College Park, 2005. Available at: http://trust.mindswap.org/papers/GolbeckDissertation.pdf.

[10] M. A. Harrison, W. L. Ruzzo, and J. D. Ullman. Protection in operating systems. *Commun. ACM*, 19(8):461–471, 1976.

[11] J. He, W. W. Chu, and Z. Liu. Inferring privacy information from social networks. In *IEEE International Conference on Intelligence and Security Informatics*, 2006.

[12] S. R. Kruk, S. Grzonkowski, A. Gzella, T. Woroniecki, and H. Choi. D-foaf: Distributed identity management with access rights delegation. In *Proc. of the 1st Asian Semantic Web Conference*, volume 4185 of *Lecture Notes in Computer Science*, pages 140–154. Springer, 2006.

[13] N. Li and M. V. Tripunitara. Security analysis in role-based access control. *ACM Trans. Inf. Syst. Secur.*, 9(4):391–420, 2006.

[14] A. McAfee. Enterprise 2.0: the dawn of emergent collaboration. *MITSloan Management Review*, 47(3), 2006.

[15] Y. Tim Oreill. What is web 2.0: Design patterns and business models for the next generation of software. *Social Science Research Network Working Paper Series*, 2003.

[16] W. H. Winsborough and N. Li. Safety in automated trust negotiation. *ACM Trans. Inf. Syst. Secur.*, 9(3):352–390, 2006.

[17] E. Zheleva and L. Getoor. To join or not to join: the illusion of privacy in social networks with mixed public and private user profiles. In *Proc. of the WWW Conference*, pages 531–540. ACM, 2009.