Fighting Spam in Publish/Subscribe Networks Using Information Ranking

Nikos Fotiou, Giannis F. Marias and George C. Polyzos
Athens University of Economics and Business
Mobile Multimedia Laboratory
Patision 76, Athens 104 34, Greece
E-mail: {fotiou,marias,polyzos}@aueb.gr

***Abstract*—New paradigms for the Future Internet are receiving an increased attention by the research community. The publish/subscribe paradigm is one of these and of particular interest, as it turns the Internet into an information-centric rather than endpoint-centric place of communication. While significant work has been undertaken to secure publish/subscribe systems, little attention has been given to prevent spam. In this paper we propose a light-weight solution for fighting spam, based on information items ranking. We compare our solution to a users-ranking based solution and we show that our solution is more effective in terms of publication spam isolation.**

***Index Terms*—Information-Centric Architectures, Publish/Subscribe, Spam**

I. Introduction

Publish/Subscribe is regarded as a promising paradigm for future Internet applications or even as a candidate for a (clean slate) future Internet architecture. Publish/Subscribe is currently under investigation in a variety of research efforts–such as CCNx [1] PSIRP [2] and 4WARD [3]. Its information centrism, i.e., the routing of information from supplying to demanding parties, manifests a significant shift from the traditional endpoint-centric Internet paradigm, which merely routes pieces of data between dedicated endpoints. In addition, the interest-based decision on the information subscriber/recipient side, shifts the network balance towards the receiver, compared to the commonly used send-receive paradigm that empowers the sender. Publish/Subscribe as an overlay architecture, has been used in a variety of research projects and it has been found to be particularly effective when it comes to multicast [4], mobility [5], indirection [6] as well as caching [7].

In publish/subscribe architectures information providers, which are referred as publishers, advertise the pieces of information that they possess. On the other hand information consumers, subscribe to desired information items, therefore the term subscribers is used to describe them. A network of brokers–also known as the rendezvous network–is responsible for locating the publishers who provide the information items that satisfy the consumers' subscriptions and initiate a forwarding process from the information providers towards the information consumers. The broker in which publication-subscription matching takes place is known as the rendezvous point. Rendezvous networks are usually organized in a distribute hash table and every broker in the network is responsible for a set of publications. The publication process involves the advertisement of an information item to the proper broker, usually along with some metadata that describes this item. Similarly during the subscription process a message, that contains the criteria that an information item should fulfill in order to match subscribers interest, is sent to the proper broker. More details about the publish/subscribe architecture that is used as reference is this paper are given in Section III.

It is generally argued that by design it is difficult to achieve spam in publish/subscribe systems; in such systems no information flow occurs unless there exist explicit signaling denoting the demand as well as the availability of a specific information item. Nevertheless being mainly used for information dissemination, publish/subscribe is expected to become the target of spammers aiming at flooding these networks with bogus information items. Moreover publish/subscribe systems are not yet widely deployed, therefore their security properties have not been tested in real environment. Spam in publish/subscribe had not been studied until recently. Tarkoma [8] predicted that publish/subscribe spam will be similar in nature to email or usenet spam, nevertheless due to the nature of publish/subscribe systems it will not be possible to use solutions developed for fighting spam in email and usenet services. Lagutin et al. [9] identified unwanted traffic prevention as a primary goal in a publish/subscribe network architecture. Furthermore malicious publications had been proved to be the cause of many types of denial of service attacks in publish/subscribe networks [10].

This paper presents a light-weighted solution for fighting spam in publish/subscribe networks, based on inforanking [11]. This approach uses a two-step publication ranking; one based on the number of publishers that provide this publication and another based on the subscribers' feedback. Our suggested solution relies on the fact that malicious publishers will try to generate as many similar publications as possible in order to circumvent the publication blacklisting that is driven by the subscriber's feedback. By ranking publications and not publishers we ensure that malicious publishers will not have any gain by taking advantage of legitimate publishers, e.g., with usage of viruses and worms. Our solution can be easily deployed and presents significant advantages when compared to a publishers ranking based solution.

The rest of this paper is organized as follows. Section II presents related work in this area and gives an overview of

inforanking. In Section III we describe a publish/subscribe architecture that is used as reference architecture in this paper and in Section IV we present our solution which is evaluated in Section V. Finally Section VI presents our conclusions and future work.

II. Related Work

The only related work regarding publish/subscribe spam is–to our knowledge–the solution proposed by Tarkoma [8]. Tarkoma identifies three possible causes that may lead to spam in a publish/subscribe network; bogus brokers, event replication and users' interest prediction. He presents an infrastructure-based solution in which each entity digitally signs every message it sends or forwards. The messages are signed using a public key. Public keys can be either self-generated or provided by a third trusted party. Whenever a message is received, the message receiver checks whether this message was send/forwarded by an entity that is considered as a spammer. In order to do this it should extract the identity of all nodes that the message traversed and consult publicly available lookup service that contains all the spammers' ids.

Our solution differs from Takoma approach. Instead of trying to isolate the nodes that cause spam we isolate spam information itself. The reasoning behind this approach is that a node isolation based solution may jeopardize legitimate nodes as malicious nodes will try to manipulate them with the usage of tools such as viruses and worms. Moreover information items can be identified by self-signed ids, e.g., by ids that are based on the result of a hash function over the item data. Finally it is easier to determine in an objective way whether a specific information item is malicious or not, rather than to determine if a node behaves maliciously or not–especially when it comes to Byzantine nodes. We evaluate our approach against a node ranking based approach in the evaluation section.

As far as the publish/subscribe architecture security is concerned, several solutions have been proposed. Event-Guard [12], is a mechanism that aims at providing security for content-based publish/subscribe systems by using "guards", that secure the critical operations of the publish/subscribe system. QUIP [13] is a lighter version of Eventguard that secures less operations. Pallickara et al. [14] propose a centralized framework for encrypting messages in publish/subscribe architectures while Belokosztolszki et al [15] modify Hermes publish/subscribe [16] system in order to support access control.

All these solutions are not focused on preventing spam communication and they demand heavy modification of the existing publish/subscribe architecture. Our solution does not add any additional entity in the network, it tries to use already existing functionality and it imposes a minimum communication and state overhead.

A. Inforanking

Inforanking is a vote-based approach for ranking information items. It was initially developed for isolating polluted pieces of information in file sharing networks and its development was based on the observation that in these networks malicious users, provide numerous polluted versions, in order to avoid blacklisting. Its design was driven by the requirement to add the least possible overhead to the already deployed architecture. Inforanking has been proved to be more effective than user-ranking based solutions–such as Credence [17]–in terms of polluted objects isolation. Moreover it has minimum impact to the architecture.

ItemID	Users	Score
Item01	U1, U2, U3, U4	0.25 + 0.5 + 0.5 + 0.5 = 1.75
Item02	U1, U2, U3, U4	0.25 + 0.5 + 0.5 + 0.5 = 1.75
Item03	U1, U5, U6, U7	0.25 + 1 + 1 + 1 = 3.25
Item04	U1	0.25

TABLE I
INFORANKING VOTING EXAMPLE

In inforanking users may vote only positively regarding a specific information item. Moreover a user may vote only once. When it comes to a file sharing system the fact that a user shares a file is considered as a positive vote. Therefore there is no need for the deployment of a separate voting subsystem. Each vote of a user U in a context C is weighted by a factor w computed as $w = 1/(\sum U_C)^a$ where $\sum U_C$ is the sum of U's votes in C and a a fixed value. As an example consider a system in which users query for information items using keywords. The result set of user's query is the context in which w is calculated using the above formula. Table 1 is an example of inforanking usage with $a = 1$. The first column of this table contains all the items that are included in the result set. The second column contains a list of users that share each item and the third column contains the inforanking of each item. As it can be seen in the table user U1 has voted for 4 items in the result set, so his vote is weighted by 0.25, On the other hand users U5, U6, and U7 have voted only once so their votes are weighted by 1.

III. A Publish/Subscribe Reference Architecture

Various publish/subscribe architecture proposals exist in the literature. They are classified into two broad categories, namely topic-based and content-based. In the rest of this section we describe a conceptual content-based publish/subscribe architecture, which is used as reference architecture in this paper.

In the majority of the proposed architectures, a publish/subscribe system consists of publishers, subscribers and routing nodes -also called brokers [18]. Our reference architecture adopts this approach. Publishers are information providers that advertise information, service, or content. Subscribers are consumers that explicitly express their interests in a specific published element. Brokers are elements that match publishers' advertisements with subscribers' interests. They initiate routing, forwarding, and distribution decisions, eventually leading to the delivery of the content from publishers to subscribers. A node where the matching of the publisher

content with the subscriber interests takes place is referenced to as the rendezvous point.

Publishers feed an information element into the publish/subscribe system by virtue of publications. A subscriber expresses her interest in receiving a piece of information by issuing a subscription message that contains keywords on what kind of information to be delivered. The rendezvous point which is responsible for handling these keywords will receive the subscription message. This rendezvous point will decide which is the appropriate publication that corresponds to subscribers keywords.

For the rest of this paper we make the following assumptions:

- A publication can be provided by more than one publisher.
- Every publication is identified by a unique identifier, e.g., the result of a hash function over the publication data, and it is impossible for a publication to use a false identifier.
- Publishers and subscribers are not anonymous. We assume the existence of an authentication service. Moreover we assume that it is difficult for a single user to create multiple accounts.
- A specific keyword is handled by a single rendezvous point.
- All rendezvous points are reliable.

IV. Approach

The target of our approach is to enable rendezvous points to isolate spam publications and respond to subscriptions with valid ones. Inforanking is applied to the result set that occurs whenever a subscriber requests for a subscription using keywords. This result set contains all the publications that match the subscription's keywords. The purpose of inforanking in this approach is to give **bigger** rank to spam publications. Publications are ranked based on the number of publishers that provide these publications as well as based on subscribers' votes. Figure 1 gives an overview of the suggested approach. Publishers publish their publications to a rendezvous point and subscribers issue subscriptions that contain some keywords. Whenever the rendezvous point receives a subscription that can be matched to one or more publications, it forwards the appropriate publication. When the publication is received, subscribers vote if they consider that it was a spam. Our anti-spam mechanism is triggered after subscription operation and before forwarding, and it takes into consideration previous publication and vote messages.

A. Publisher-based Ranking

The rule of thump in this step is that the publications that are provided by many well-behaved publishers, are probably valid publications. Well-behaved publishers are those who publish a normal number of publications. Inforanking assures that the bigger the number of publications a publisher provides the lesser is the effect he has on the publication's rank. The fact that a publisher provides a publication, is considered as a positive vote for this publication. This vote is weighted

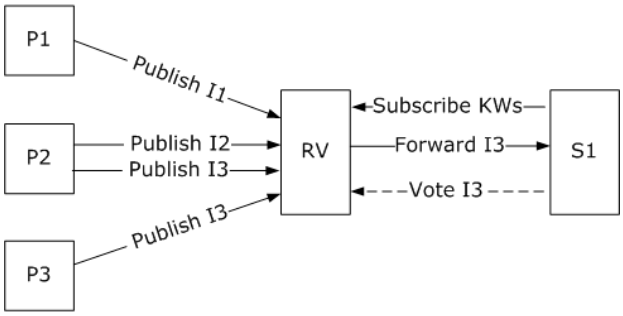

Fig. 1. Overall Architecture

with respect to the total number of publications the publisher provides in the same result set. The rank of every publication i in the result set R is $\sum V_i^R$ where V_i^R is a vote for i in R. As a vote is considered the fact that a publisher P provides i in R and it is weighted by $1/\sum Pu_P^R$ where Pu_P^R is a publication of P in the result set R. As it can be observer inforanking is used with $a = 1$. As an example, if a publication A is provided by two publishers and each of these publishers has 4 publications in the result set that contains A then the publisher-based rank of A is $PR(A) = (1/4 + 1/4) = 0.5$. In our approach we normalized publisher-based ranks using the following formula: $NPR(A) = PR(A)/\sum P^R$ where $\sum P^R$ is the total number of publishers in the result set R. The bigger the publisher-based rank is the better a publication is, therefore in our approach we consider $1 - NPR$.

During this step, the ranking of an information item is calculated based on data and functionality provided by the publish/subscribe infrastructure, i.e., no extra state or communication overhead is added to the network.

B. Subscriber-based Ranking

During this step subscribers vote for spam publications, i.e, whenever a subscriber receives a spam publication she sends a message towards the rendezvous point and informs it about this specific publication. Subscribers may vote only once for a specific publication and there is no vote that indicates that a publication is **not** spam. Subscribers votes are considered when a result set is created. Every vote V of a subscriber S is weighted by $1/\sum V_S^R$ where $\sum V_S^R$ is the total votes of S in the result set R. So, if a publication A has received two votes from two different subscribers and each of these subscribers has already voted for 10 publications in the result set that contains A then the subscriber based rank of A is $SR(A) = (1/10 + 1/10)$. The subscriber-based ranking is also normalized using this formula: $NSR(A) = SR(A)/\sum S^R$ where $\sum S^R$ is the total number of subscribers that have vote in the result set R. The bigger the subscriber-based rank is the bigger is the possibility for a publication to be a spam.

This step requires some additional state and communication overhead. Each rendezvous point should maintain a list of subscriber votes and each subscriber vote is an extra message in the network. Nevertheless the state is fully distributed to

all rendezvous points and the vote message may be possible encapsulated in other messages, e.g., in an ACK message.

The inforank of a specific publication is the sum of $1-$ the normalized publisher-based rank and the normalized subscriber-based rank, i.e, $IR = 1 - NPR + SR$, and the publication chosen by the rendezvous point is the one with the smaller inforank.

V. Evaluation

We evaluate our solution using two threat models. The first threat model concerns a publish/subscribe architecture in which there exist malicious publishers who publish spam publications. In the second threat model in addition to the malicious publishers we consider malicious subscribers who collude and vote against valid publications.

A. Simulation Setup

Using OMNeT++ [19] and OverSim framework [20], we simulate a network consisting of 100 publishers and 100 subscribers. A publisher may, or may not, be malicious and–in the second threat model–a subscriber may, or may not, collude. Non malicious publishers share in average 5 information items. These items are selected through a pool of 80 valid information items, using a zipf distribution. All the published information items concern the same keywords, therefore are published to the same rendezvous point. Subscribers query this rendezvous point in specified time intervals requesting an information item. The simulation ends when all subscribers obtain one valid information item.

B. Threat model A

In this threat model we examine the case in which a percentage of the publishers publishes spam publications. We consider two scenarios, one in which 50% of the publishers behaves maliciously and another in which 80% of the publishers behaves maliciously. In each scenario we compare our solution against a solution based on publisher's ranking as described in [8]. More precisely when publisher's ranking is used, whenever a subscriber receives a spam publication, he updates a global accessible black list that contains publishers that publish spam publications. Moreover when publisher's ranking is used, the publication chosen by the rendezvous point is the one that has the biggest number of non black-listed publishers.

Malicious publishers publish as many objects as needed in order to achieve the maximum negative impact to the network. As it can be seen in Figure 2, when inforanking is used, the number of the subscriptions that lead to spam publications, depends on the number of objects that malicious publishers choose to share from their pool of publications. This number of objects in every experiment has been determined through simulations.

Figure 3 shows the total number of subscriptions that leads to spam publications, when 50% of the publishers are malicious and choose their objects from a pool of 10 to 120 spam publications. Figure 4 shows the simulation outcome

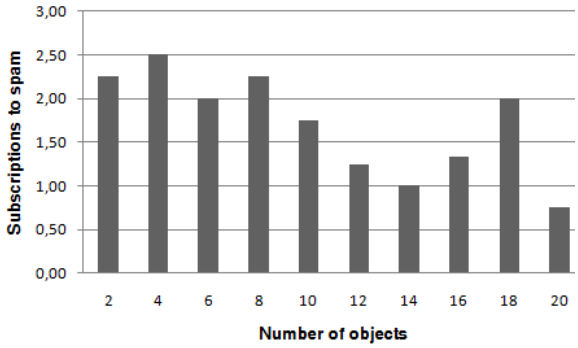

Fig. 2. Total subscriptions to spam publications when malicious publishers choose a number of objects to publish, from a pool of 20 publications

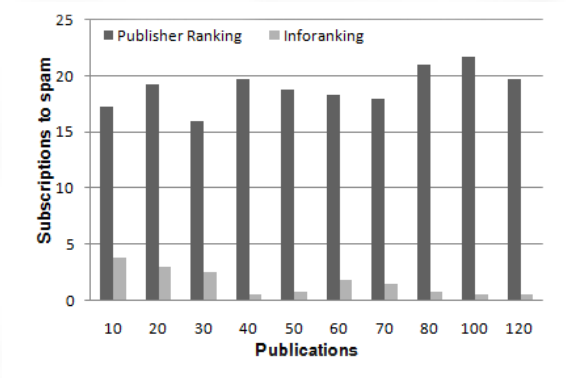

Fig. 3. Total subscriptions to spam publications in a network in which 50% of the publishers are malicious

when 80% the publishers are malicious. As it can be observed inforanking is much more effective than publisher's ranking. Moreover as the number of publications that a malicious publisher may publish augments, the number of subscriptions that led to spam publication tends to zero when inforanking is used, whilst it remains almost constant when publisher's ranking is used.

When inforanking is used, the number of the subscriptions

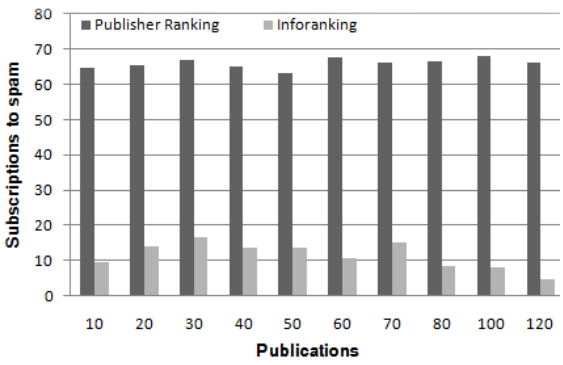

Fig. 4. Total subscriptions to spam publications in a network in which 80% of the publishers are malicious

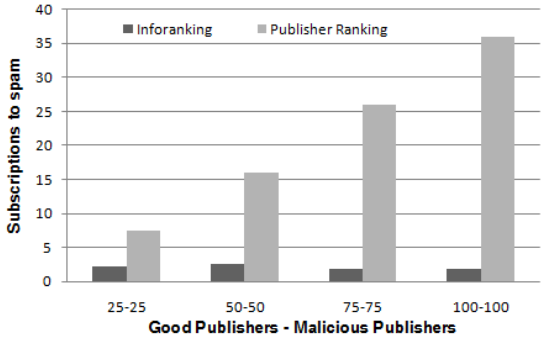

Fig. 5. Total subscriptions to spam publications when the number of publishers is variable

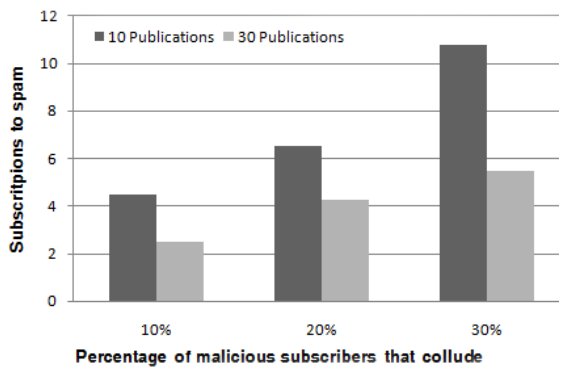

Fig. 6. Total subscriptions to spam publications in a network in which 50% of the publishers are malicious and malicious subscribers collude

that lead to spam publications is not affected by the number of publishers. On the other hand the efficacy of a publisher's ranking based solution is greatly affected by the number of publishers. Figure 5 shows the number of subscriptions that lead to spam publications when 50 publishers (25 good, 25 malicious), 100 publishers (50 good, 50 malicious), 150 publishers (75 good, 75 malicious) and 200 publishers (100 good, 100 malicious) are considered. In all cases the number of the subscribers remains constant (100) as well as the number of publications each publisher publishes. As it can been seen inforanking is not affected by the variable number of publishers.

C. Threat model B

In this threat model malicious subscribers are also considered. Malicious subscribers collude and vote against valid publications. Publisher's ranking is not examined in this model as collusions are not considered in this approach. Malicious subscribers are the first nodes that enter the network and they vote at the same time intervals as the time intervals in which valid subscribers issue subscription messages. Moreover 50% of the publishers are malicious, and they choose their publication from a pool of 10 or 30 spam publications–this is the number of spam publications that had the biggest negative impact in threat model A.

Figure 6 shows the total number of subscriptions that led to spam publications when malicious publishers choose spam publications from a pool of 10 or 30 publications and the percentage of subscribers that collude is 10% or 20% or 30%. As it can be seen even if 30% of the subscribers collude, if inforanking is used, the maximum number of subscriptions that leads to spam publications is less than the number of subscriptions that leads to spam publications if publisher's ranking is used and there are **no** malicious subscribers.

VI. Conclusions and Future Work

In this paper we presented a light-weighted solution for fighting spam in publish/subscribe networks. We used a 2-step approach, based on publisher's behavior and subscriber's votes. We compared our solution to a publisher's ranking based solution and we found out that our solution leads to smaller number of spam publications. Moreover our solution uses, to its biggest extent, functionality already deployed in a publish/subscribe network and it needs only a few extra messages towards rendezvous-points as well as some extra state maintained by the rendezvous points. We also examined the case in which malicious subscribers exist in the system and try to affect it, in favor of spammers and we found out that even in this case our solution is robust enough.

Future work includes the development of a large scale publish/subscribe system and the deployment of inforanking functionality in this system. We believe that fighting spam in only one of the many possibilities that inforanking may offer. We anticipate to incorporate many inforanking-based solution in our publish/subscribe system including malicious file isolation, faulty vote elimination, effective publication selection, denial of service attack prevention.

As far as the spam prevention mechanism described in this paper is concerned, future work includes the usage of pre-trusted subscribers, whose votes will have bigger weight as well as the distribution of the system functionality among multiple points, e.g., publisher-based ranking may take place in rendezvous points whereas subscriber-based ranking may take place in local brokers.

Acknowledgment

The work reported in this paper was supported by the ICT PSIRP project under contract ICT-2007-216173.

References

[1] "Ccnx project," March 2010, `http://www.ccnx.org`.
[2] "Psirp project," March 2010, `http://www.psirp.org`.
[3] "4ward project," March 2010, `http://www.4ward-project.eu`.
[4] M. Castro, P. Druschel, A. Kermarrec, and A. Rowstron, "SCRIBE: A large-scale and decentralized application-level multicast infrastructure," *IEEE Journal on Selected Areas in communications*, vol. 20, no. 8, pp. 1489–1499, 2002.
[5] K. Katsaros, N. FOTIOU, G. POLYZOS, and G. XYLOMENOS, "Overlay Multicast Assisted Mobility for Future Publish/Subscribe Networks," *Proc. ICT Mobile Summit*, 2009.
[6] I. Stoica, D. Adkins, S. Ratnasamy, S. Shenker, S. Surana, and S. Zhuang, "Internet indirection infrastructure," *Peer-to-Peer Systems*, pp. 191–202.

[7] K. Katsaros, G. Xylomenos, and G. Polyzos, "MultiCache: an incrementally deployable overlay architecture for information-centric networking," *INFOCOM Work-in-Progress (WiP)*.

[8] S. Tarkoma, "Preventing Spam in Publish/Subscribe," in *26th IEEE International Conference on Distributed Computing Systems Workshops, 2006. ICDCS Workshops 2006*, 2006, pp. 21–21.

[9] D. Lagutin, K. Visala, A. Zahemszky, T. Burbridge, and G. Marias, "Roles and Security in a Publish/Subscribe Network Architecture," in *Proceedings of the 2010 IEEE Symposium on Computers and Communications*. IEEE, 2010, to appear.

[10] A. Wun, A. Cheung, and H. Jacobsen, "A taxonomy for denial of service attacks in content-based publish/subscribe systems," in *Proceedings of the 2007 inaugural international conference on Distributed event-based systems*. ACM, 2007, p. 127.

[11] N. Fotiou, G. Marias, and G. Polyzos, "Information Ranking in Content-Centric Networks," in *Proceedings of the Future Network and Mobile-Summit 2010*, 2010, to appear.

[12] M. Srivatsa and L. Liu, "Securing publish-subscribe overlay services with eventguard," in *Proceedings of the 12th ACM conference on Computer and communications security*. ACM, 2005, p. 298.

[13] A. Corman, P. Schachte, and V. Teague, "QUIP: a protocol for securing content in peer-to-peer publish/subscribe overlay networks," in *Proceedings of the thirtieth Australasian conference on Computer science-Volume 62*. Australian Computer Society, Inc., 2007, p. 40.

[14] S. Pallickara, M. Pierce, H. Gadgil, G. Fox, Y. Yan, and Y. Huang, "A Framework for Secure End-to-End Delivery of Messages in Publish/Subscribe Systems," in *Proceedings of the 7th IEEE/ACM International Conference on Grid Computing (GRID 2006). Barcelona, Spain*. Citeseer, 2006, pp. 28–29.

[15] A. Belokosztolszki, D. Eyers, P. Pietzuch, J. Bacon, and K. Moody, "Role-based access control for publish/subscribe middleware architectures," in *Proceedings of the 2nd international workshop on Distributed event-based systems*. ACM, 2003, p. 8.

[16] P. Pietzuch and J. Bacon, "Hermes: A distributed event-based middleware architecture," 2002.

[17] K. Walsh and E. Sirer, "Fighting peer-to-peer spam and decoys with object reputation," in *Proceedings of the 2005 ACM SIGCOMM workshop on Economics of peer-to-peer systems*. ACM, 2005, p. 143.

[18] P. Eugster, P. Felber, R. Guerraoui, and A. Kermarrec, "The many faces of publish/subscribe," *ACM Computing Surveys (CSUR)*, vol. 35, no. 2, p. 131, 2003.

[19] "Omnet++ community site," March 2010, `http://www.omnetpp.org`.

[20] "The oversim p2p simulator," March 2010, `http://www.oversim.org`.