On leveraging social relationships for decentralized privacy-preserving group communication

Abhishek Singh
Guido Urdaneta
University of Oslo, Norway
Department of Informatics
{abhi,guidoau}@ifi.uio.no

Maarten van Steen
VU University Amsterdam,
The Netherlands
The Network Institute and
Department of Computer Science
steen@cs.vu.nl

Roman Vitenberg
University of Oslo, Norway
Department of Informatics
romanvi@ifi.uio.no

Abstract

Millions of people use the Internet today for data exchange of sensitive information. A typical exchange usually occurs within the context of a group of users and often takes place in a centrally managed infrastructure such as Facebook. Evidence has shown that such systems are unable to properly protect the privacy of its users.

Following this observation, we present a framework to support privacy-preserving group communication. Our framework is completely decentralized and leverages the social relationships between users to bootstrap a communication overlay whose robustness is improved with the addition of extra privacy-preserving links. The framework supports multiple groups in a scalable manner and defines mechanisms to handle churn in group membership. Finally, we also outline high-level protocols for a privacy-preserving micro-news application.

Categories and Subject Descriptors C.2.4 [*Computer-Communication Networks*]: Distributed Systems

Keywords privacy, social networks, trust, peer-to-peer

1. Introduction

Nowadays, millions of people use the Internet to engage in interest-based data exchange within social communities. This message exchange is often required to provide certain privacy levels. For example, users talking about sensitive topics such as politics or a shared chronic illness may want to keep their identity, social relations and messages private.

This activity typically takes place on free, centrally managed platforms such as Facebook or Twitter. However, numerous incidents, as observed in [10], suggest that these systems cannot be trusted to safeguard sensitive user data. As an alternative, decentralized friend-to-friend (F2F) networks such as Freenet [4] have been proposed to address the needs of users wishing to exchange data in a privacy-preserving manner. These networks leverage the existence of the underlying social friendship relations between users and map them into communication links. Unlike other types of P2P networks, users in a F2F network cannot find out who else is participating beyond their own circle of friends, so that F2F networks can grow in size without compromising their users' anonymity. This also mitigates the damage to privacy if one of the user nodes is compromised by a malicious third party. However, as we have shown in our previous work [11], dissemination over a communication graph that simply mimics the social one is not very effective. Although social graphs are connected, their connectivity is weaker than that of random graphs of comparable size. Even moderate churn typical of P2P networks results in degraded connectivity and significant graph partitioning.

Our approach to privacy-preserving group communication is to bootstrap a communication overlay using the social graph but augment it with additional links between pairs of nodes that correspond to users not related by social ties. These extra links should be realized using a privacy-preserving routing mechanism such that neither node is able to learn the identity of the peer at the other end of the link.

In this paper we present a framework for privacy preserving group communication based on the aforementioned idea. In particular, we propose new extensions to state-of-the-art protocols for building a robust overlay for privacy-preserving data dissemination over a social graph. These extensions address an important need to support a large number of concurrent dissemination groups in a scalable manner. We also outline a new way to build application-level data dissemination for a privacy-preserving micro-news application on top of our framework.

SNS '12 April 10, 2012, Bern, Switzerland

2. Application characteristics

We consider applications where groups of users are interested in exchanging messages on different topics. Such groups are typically seen in applications such as forums and bulletin-boards. As these groups may be exchanging messages on a sensitive topic (e.g. discussion about a disease) it is important that the privacy of the members of such groups is preserved. For example, a group of users that discuss about ways to deal with alcohol de-addiction will require that no other user outside the group should know about the participation of a member in the group. Similarly, they also require guarantees for nondisclosure of messages exchanged in the group to any entity outside of the group. Such groups may also be formed by a subset of friends of a user u to whom u selectively disseminates messages. For example, u wants to share some photos with his college friends but not with his colleagues from work. User u requires guarantees that such messages are not received by his work colleagues.

In order to illustrate the problem better we discuss the privacy challenges in context of a Twitter-style micro-news application where a group consists of a set of users that subscribe to receive messages from another user. As messages can be sensitive, they should be delivered only to members of the group and to no one else including the communication providers (e.g. an ISP) that may be monitoring the traffic. Such groups also require that the identities of group members are not revealed to entities that are not part of the group. These entities may include an attacker who is monitoring the traffic as well as members of the group who have been compromised by an attacker. As members of a group can get compromised by an attacker, it is important that the amount of information leaked to the attacker is minimized. A possible way to achieve this is to ensure that each member of the group is aware of only a partial list of group members. Group members also require anonymity from the attacker monitoring the traffic to learn their identities. The anonymity guarantee that members require from the group communication mechanism against an attacker monitoring the traffic, is that the attacker cannot establish if a user is a member of the group.

3. Preliminaries

Our main goal is to provide a solution for decentralized privacy-preserving group communication that is robust against node failures and that leverages trust among participating users in order to counter the lack of centralized trusted components. In this section we attempt to define in precise terms a number of concepts that will facilitate reasoning about this problem and its possible solutions.

The system consists of a set U of nodes, each node being managed by a unique user. We additionally assume that each user manages a single node. For brevity, we use the terms node and user interchangeably such that trust between nodes refer to the trust between users who manage such nodes. Nondisclosure of the nodes in U is one of our central privacy-preservation goals.

Node privacy We define the privacy of a node n as the fact that its participation in the system is not known by anyone except by those peers to which n discloses this information.

Trust graph Trust between two nodes a and b refers to the fact that a and b can rely on each other to never violate the privacy of their mutual identities. We assume that any node n may reveal details of their participation in the system only to other nodes that n trusts, and n can assume that those trusted nodes will never disclose such details to any third party. The set of trust relationships between the nodes in U can be modeled as a graph where each vertex represents a node, and each edge represents the existence of trust between the corresponding nodes. In this model, trust between nodes is symmetric, but not transitive.

Note that having an edge (a, b) in the trust graph does not mean that a necessarily reveals all the details about its participation in the system to b. Instead, it refers to the fact that, if a reveals some detail to b (e.g., participation in a given group), b will never disclose that detail to any third party. This assumption is equivalent to asserting that edges in the trust graph are not deleted over time.

Edge privacy Another important privacy requirement is to guarantee nondisclosure of the edges of the trust graph, which correspond to relations between the users. For example, if a has adjacent peers b and c in the trust graph, it should not be able to use the system's protocols to determine if b and c have an edge between them.

Group We refer to a group as a set of users that are interested in exchanging messages in a specific application-defined context. For example, in a micro-news application, the set of users interested in the news published by a given user may be regarded as a group. Then, each news publisher will have an associated group.

In our system, group membership is established by offline user interactions in the context of the trust graph. For example, a user wishing to publish micro-news about certain sensitive topics (e.g., politics) may establish a group and admit members using an invitation-based scheme in which members may invite some of their friends to the group.

Participation in a group can be modeled as a graph, where each vertex is a node, and each edge (a, b) represents that a and b agree to consider each other as members of the group when executing the protocols that enable communication between group members. We refer to such a graph as a *group graph*. Note that a group graph can be seen as a subgraph of the trust graph, but the semantics of an edge in the group graph are different. It may be possible to remove edges from the group graph because a user may be either expelled from the group or no longer interested in it.

Group privacy A key privacy requirement for our system is to ensure that the knowledge of the participation of a node

a in a group G is not disclosed to any other entity which is not an adjacent node to a in the group graph for G.

Other requirements depend largely on the application. However, it should be possible to establish groups such that application-level messages sent within a group can be read only by members of that group.

4. Problem statement

The problem we need to solve is the following: Given a set U of nodes, a trust graph and multiple group graphs, find a protocol for building and maintaining an overlay network that satisfies the following properties:

- *Privacy-preserving overlay links:* The links created by the protocol should be privacy-preserving so that the application disseminating data over these links would not be disclosing node identities, relations, or data about group membership as defined in Section 3.
- *Privacy-preserving overlay maintenance:* The protocol for overlay maintenance should not be disclosing node identities, relations, or data about group membership as defined in Section 3.
- *Robustness:* For each group, the subset of the overlay network intended to support communication within that group remains connected and with relatively short path lengths in presence of realistic node churn. More precisely, we want to minimize the probability that such group suboverlays become partitioned due to offline nodes.
- *Scalability:* Our main goal with regard to scalability is to minimize the fan-out of nodes participating in the system. In particular, we want the number of required overlay links to scale with the number of groups in the system.

In addition to the overlay network, it is also necessary to define a protocol for application-specific data dissemination within a group that supports group privacy as defined in Section 3.

We cannot use centralized node directories to bootstrap the overlay network, since such a directory can be compromised. On the other hand, leveraging the mutual knowledge of the neighbors in the trust graph is an appealing way of bootstrapping overlay links between the nodes that would otherwise be unable to learn of each other.

A difficulty on the way to solving the problem is that direct communication channels between any pair of nodes might be monitored by a passive external observer (e.g., an ISP). A naive data exchange over these channels may reveal both the identities of the nodes and the fact that there is a trust relation between them. Therefore, it is necessary to create an indirect communication link that prevents observers from learning that the trusted nodes are exchanging data in the context of our system. Fortunately, this specific issue can be addressed by existing anonymity systems [5].

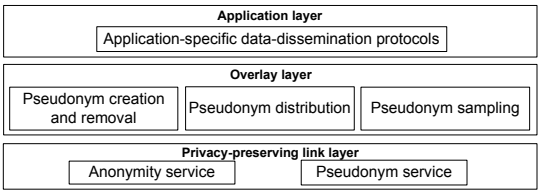

Figure 1. Architecture for privacy-preserving data dissemination

However, establishing overlay links between trusted nodes is not enough to make the overlay robust. As we have shown in previous work [11], in typical trust graphs, a significant fraction of the online participating nodes may become disconnected under moderate node churn. Hence, we also have to create overlay links between untrusted nodes (i.e., not connected by an edge in the trust graph). In this case, the link must not only protect against external observers, but it must also prevent both nodes from learning each other's IDs.

In this paper we present a framework that aims to solve the problem by generalizing the architecture of a previous solution designed for single groups to the case of multiple groups whose membership can change. In particular, we outline protocols for overlay maintenance and a Twitter-style micro-news application.

5. Proposed framework

Our solution for privacy-preserving robust group communication is based on a layered framework shown in Figure 1. The lowest layer of our framework is a privacy-preserving link layer that allows any pair of users to exchange messages privately. The next layer is an overlay layer, which is responsible for the creation and maintenance of privacy-preserving communication links among nodes such that the resulting overlay graph exhibits good properties for data dissemination such as good connectivity and short path lengths. Under our approach, the overlay is bootstrapped by mapping edges in the trust graph to overlay links, similar to traditional F2F networks. Unlike F2F networks, we improve the overlay's data dissemination properties by carefully adding extra privacy-preserving links among nodes that do not trust each other. The topmost layer of the framework consists of protocols that use the overlay links to disseminate data in an application-specific manner. In the rest of this section, we describe these layers in more detail.

5.1 Privacy-preserving link layer

The privacy-preserving link layer consists of an anonymity service and a pseudonym service. This layer allows two nodes to establish a privacy-preserving communication link between them. Privacy-preserving links allow nodes to exchange messages without revealing their participation in the system to external observers monitoring underlying communication channels.

The anonymity service allows any node to create privacy-preserving links to any peer whose ID is known. The main

use for the anonymity service in our system is to establish overlay links between nodes that trust each other and, therefore, know each other's IDs. Each node n uses this service to establish links with its neighbors in the trust graph when n becomes online. We refer to the set of privacy-preserving links built in this way as the *trusted links* of n.

The anonymity service can be realized using existing solutions based on the concept of mix networks [2]. Mix networks allow the implementation of privacy-preserving links between two nodes by employing a set of relay nodes. The sender applies multiple encryption layers to the message and sends it to the first relay in the mix, each relay removes one encryption layer and passes the message to the next relay until it reaches the destination. To send a response back to the sender, the receiver sends the response to the first relay, each relay adds an encryption layer, and the destination applies all the decryption operations.

In a mix network, relays do not know their position in the chain, so a given relay cannot know if the previous node in the chain is the sender or another relay, or if the next node is the destination or another relay. It is also difficult for an external observer who can monitor communication channels to associate the sender with the receiver. A recent survey [5] describes in great detail the state of the art in mix-based anonymity systems.

Together with the anonymity service we use a pseudonym service, which allows any node to create pseudonyms and to establish privacy-preserving links to any peer for which a pseudonym is known. A pseudonym $P(n)$ of a node n is an address that any other node m can use in conjunction with the pseudonym service to build a link to n such that n's ID is not disclosed to m and vice versa, and that the participation of both m and n in the system remains undisclosed to external observers. We use the pseudonym system to establish links between nodes that do not trust each other and hence must not be able to learn each other's IDs. For any node n, we refer to the set of privacy-preserving links established by n using the pseudonym service as the *pseudonym links* of n.

Pseudonym services can be realized with the help of an anonymity service. A few deployed anonymity services have this extra functionality built in. Examples are I2P's "eepsites" [16] and Tor's "hidden services" [6], in which a node n wishing to be contacted establishes a mix network, with the address of the last relay acting as a pseudonym. A node wishing to contact n can send a request to the last relay (i.e., the pseudonym) and negotiate a separate mix for further communication.

5.2 Overlay layer

The overlay layer is responsible for the creation and maintenance of overlay links and for letting the application layer use those links to implement data-dissemination protocols.

Maintaining trusted links is simple. Initially, every node knows the IDs of its neighbors in the trust graph and can therefore use the anonymity service to establish trusted links. On the other hand, nodes initially have no knowledge about pseudonyms, and thus cannot readily create pseudonym links. To solve this problem, the overlay layer executes a maintenance protocol that creates and removes pseudonyms, distributes pseudonyms across the overlay, and adds and removes pseudonym links such that the resulting overlay is robust for data dissemination.

In previous work [11] we have proposed and evaluated overlay-layer protocols to support robust data dissemination in the context of a single group. In this section we describe these protocols and formulate new extensions intended to support multiple concurrent groups in a scalable manner.

5.2.1 Single group

Pseudonym creation and removal Each node creates a pseudonym of itself when the node starts. Our pseudonyms have a limited lifetime, so that whenever a pseudonym expires all pseudonym links involving the expired pseudonym are removed from the overlay. Therefore, every node must periodically create a new pseudonym. Pseudonyms remain valid even if the corresponding node leaves and rejoins the overlay at a later time, provided the node rejoins before the pseudonym's expiration time.

Having ephemeral pseudonyms can help improve the privacy of our system against certain types of external observers, and makes it easier to defend against replay attacks. On the other hand, ephemeral pseudonyms may result in an unstable overlay if their lifetime is too short. Our previous work [11] showed that having pseudonym lifetimes that are at least three times longer than the expected offline time of nodes leads to robust overlays.

Pseudonym distribution Pseudonyms are distributed across the overlay by means of a shuffling gossiping protocol [12, 14]. Under this scheme, each node n maintains a pseudonym cache of a configurable size. Periodically, n selects one of its overlay links uniformly at random and exchanges a set of pseudonyms with the node at the other end of the link. The set includes the node's own pseudonym and up to $\ell - 1$ pseudonyms from the node's cache. Upon receiving a set over the link, the node updates its own cache to include all entries in the received set (with the exception of its own pseudonym, if present). The cache replacement policy is the same as in in [14]. Additionally, all pseudonyms in the received set, whether already in the cache or not, are sampled as described next.

Pseudonym sampling Each node establishes pseudonym links with a carefully selected sample of the pseudonyms received by the shuffling protocol. The maximum allowed number of overlay links per node sets the balance between potentially higher overhead and better overlay robustness.

Our goal is to select a sample of pseudonyms such that the resulting overlay resembles a random graph. We achieve this with a mechanism based on the Brahms protocol [1], which guarantees that the pseudonym sample for every node

n will always be a random sample of all the pseudonyms n has received by means of the shuffling protocol, regardless of how frequently each pseudonym is received.

5.2.2 Multiple groups

Here we present possible extensions to the single-group mechanisms so that we are able to support multiple groups.

Pseudonym creation and removal In the case of multiple groups there are two obvious ways to extend our original mechanism. The first is to have a separate pseudonym for each group, which is equivalent to having separate instances of the single-group case. This is a simple solution, but it does not scale well in the number of groups since it will result in nodes with high fan-out.

The second obvious extension is to use a single pseudonym for all groups. This, combined with appropriate pseudonym distribution and sampling mechanisms, could result in a very efficient overlay where nodes with similar interests establish overlay links, reducing the total number of overlay links required to establish robust overlay graph. One disadvantage of this approach is that it might be easier to correlate different pseudonyms that correspond to the same node, since such pseudonyms would present the same interests.

An intermediate solution is have each node use a small number of pseudonyms for all the groups in which it participates (i.e., fewer pseudonyms than groups per node), and change the mapping of groups to pseudonyms every time a new set of pseudonyms needs to be generated due to the expiration of the previous set. This solution would still improve scalability with respect to having a different pseudonym per group, while making it difficult to correlate pseudonyms.

Note that a new privacy concern with respect to the single-group solution is to prevent nodes from being able to determine if two different pseudonyms correspond to the same node.

Pseudonym distribution We envision two possible extensions to the pseudonym distribution mechanism. One possibility is to have a single instance of the shuffling protocol and attach to each pseudonym information that allows fellow group members to discover if a given pseudonym participates in a group. If each group g_i has a group-specific symmetric key, then each pseudonym P could be distributed as a tuple $(P, K_1(P), \cdots, K_m(P))$, where P is a pseudonym (expressed as a bit string), and $K_i(P)$ is P encrypted with the symmetric key of group g_i. This allows any node n to determine if a given pseudonym participates in groups where n participates, but prevents n from discovering the other groups in which the pseudonym participates.

Another possibility is to have a separate instance of the shuffling protocol for each group. In each instance, only pseudonyms that participate in the corresponding group are propagated. This solution has the advantage that nodes only receive data they are interested in, and therefore, there is no need to have per-group keys to obfuscate any data and propagation can be faster. A possible disadvantage is that it may require more careful parameter tuning in order to keep overhead at reasonable levels.

Note that a new privacy concern posed by multiple groups is to prevent any node n from discovering the participation of any given pseudonym in groups in which n does not participate.

Pseudonym sampling Establishing links with a uniform random sample of pseudonyms per group would limit the scalability of the system in the number of groups. Such a policy would result in an overlay with a number of links proportional to the number of groups in the system. A better choice is to use a sampling method that biases pseudonym samples towards pseudonyms that share multiple groups. The challenge in this case is to introduce bias in a way that still results in group-level suboverlays with good data-dissemination properties. A number of protocols have been proposed to achieve this goal including SpiderCast [3] and Vicinity [15], among others.

Note that, unlike pseudonym creation and distribution, changing the pseudonym sampling method to support multiple groups does not raise evident privacy concerns. However, it does raise important scalability concerns.

5.2.3 Churn in group membership

A node n should not receive messages from a group if n is no longer a member of that group. A node can voluntarily leave a group or it can be removed from the group if the node becomes disconnected from the group graph due to the removal of group-graph edges.

When a node voluntarily decides to leave the group, it can inform its neighbors in the corresponding group graph and stops providing a pseudonym for the group. Since pseudonyms have limited lifetime, the user will be removed from the group after expiration of its older pseudonym.

The case when a user is removed from the group can be handled if we make the assumption that each group has an ephemeral secret key distributed to group members using exclusively the trusted links of the overlay to ensure that only nodes connected to the group through a trust path receive the updated key. We can use the group key to encrypt the group-specific information distributed by the pseudonym-distribution protocol, effectively preventing a nonmember from establishing new pseudonym links to the group; and we can also encrypt application-level messages, preventing nonmembers from reading them.

The mechanism to establish group keys is application specific. For applications where there is a single application-level data publisher (e.g., a Twitter-style application), the group key can be periodically generated by the publisher. For applications where any group member can publish data (e.g., a discussion forum), a leader-election algorithm may be run periodically across the group graph (using only trusted links) to decide who will generate the new key.

5.3 Application layer

In this section, we outline how to leverage our framework to implement a Twitter-style micro-news application. We assume that each group is initiated by a single user and all publications for that group (i.e., the micro-news) are generated by the group's creator. Unlike in Twitter, membership to the group is established using offline means, such as invitations or other social interaction.

Micro-news dissemination within a group can be achieved with well-known protocols such as anti-entropy gossiping [7], or even flooding. In order to prevent nonmembers from reading the group's micro-news, the group owner can encrypt the micro-news messages with a group key, following the approach presented in Section 5.2.3. Apart from group-key encryption, when a node forwards a micro-news message to a neighbor, it can further encrypt the message using a neighbor-specific (public) key, making it more difficult for a potential observer to discover membership in the group just by looking at which nodes receive the same messages. Such keys may be established during the offline process of joining a group for the case of trusted links, and, for the case of pseudonym links, either as part of pseudonym distribution protocol or a with separate link-establishment protocol.

6. Related work

The most common approach for building decentralized privacy-preserving communication overlays is to use F2F networks in which the overlay includes only links between nodes who trust each other. Examples of this approach are Turtle [8] and Freenet's so-called darknet mode [4]. However, it has been observed that overlays built with this approach do not provide optimal properties for data dissemination as they tend to get partitioned under node churn [11].

Another approach for enabling privacy-preserving decentralized group communication is to use social relationships to control who is able to participate in the group, but do not take such relationships into account for building the communication overlay. One example of this is Whisper [9], which allows building private invitation-only groups using random overlays that have privacy-preserving links based on mix networks. However, Whisper's privacy model is limited to preventing members of one group from learning the identities of members of other groups. No attempt is made to protect the identity of members within a single group. Another example is Membership-Concealing Overlay Networks (MCONs) [13]. As in Whisper, membership in a MCON is by invitation, but their primary goal is to protect the identity of users even from other participants. Their approach is to organize nodes in a DHT, such that each node is connected to a limited number of other participants. This helps prevent "celebrity" attacks, in which compromising a hub in the social graph allows the attacker to gain significant information about the whole system. However, the degree limitation in MCONs is achieved with the help of a trusted online central authority, which might also be compromised.

7. Conclusions and future work

The paper discusses a framework for robust and scalable privacy-preserving group communication, suitable for situations in which users need to exchange messages about sensitive topics. Our framework relies on social relationships rather than centralized components to bootstrap a communication overlay. We build on our previous protocol for improving the robustness of such an overlay by adding extra links between users not connected to social ties, and extend it to support multiple groups in a scalable manner. The next step will be to evaluate our proposed protocols with trace-driven simulation experiments in order to assess their effectiveness and efficiency.

References

[1] E. Bortnikov, M. Gurevich, I. Keidar, G. Kliot, and A. Shraer, "Brahms: Byzantine resilient random membership sampling," *Computer Networks*, vol. 53, no. 13, 2009.

[2] D. L. Chaum, "Untraceable electronic mail, return addresses, and digital pseudonyms," *CACM*, vol. 24, 1981.

[3] G. Chockler, R. Melamed, Y. Tock, and R. Vitenberg, "SpiderCast: a scalable interest-aware overlay for topic-based pub/sub communication," in *DEBS*, 2007.

[4] I. Clarke, O. Sandberg, M. Toseland, and V. Verendel, "Private communication through a network of trusted connections: The dark freenet."

[5] G. Danezis and C. Diaz, "A survey of anonymous communication channels," Microsoft Research, Tech. Rep., 2008.

[6] R. Dingledine, N. Mathewson, and P. Syverson, "Tor: the second-generation onion router," in *SSYM*, 2004.

[7] K. Petersen, M. Spreitzer, D. Terry, M. Theimer, and A. Demers, "Flexible update propagation for weakly consistent replication," in *SOSP*, 1997.

[8] B. Popescu, B. Crispo, and A. Tanenbaum, "Safe and private data sharing with turtle: Friends team-up and beat the system," in *Security Protocols*, ser. LNCS, 2006, vol. 3957.

[9] V. Schiavoni, E. Riviere, and P. Felber, "Whisper: Middleware for confidential communication in large-scale networks," in *ICDCS*, 2011.

[10] A. Shakimov, H. Lim, R. Caceres, L. Cox, K. Li, D. Liu, and A. Varshavsky, "Vis-à-vis: Privacy-preserving online social networking via virtual individual servers," in *COMSNETS*, 2011.

[11] A. Singh, G. Urdaneta, M.van Steen, and R. Vitenberg, "Robust overlays for privacy-preserving data dissemination over a social graph," in *ICDCS*, 2012.

[12] A. Stavrou, D. Rubenstein, and S. Sahu, "A lightweight, robust p2p system to handle flash crowds," in *ICNP*, 2002.

[13] E. Vasserman, R. Jansen, J. Tyra, N. Hopper, and Y. Kim, "Membership-concealing overlay networks," in *CCS*, 2009.

[14] S. Voulgaris, D. Gavidia, and M. Steen, "CYCLON: Inexpensive membership management for unstructured P2P overlays," *JNSM*, vol. 13, no. 2, 2005.

[15] S. Voulgaris and M.van Steen, "Epidemic-style management of semantic overlays for content-based searching," in *EuroPar*, 2005.

[16] B. Zantout and R. Haraty, "I2P data communication system," in *ICN*, 2011.