Techniques for Content Subscription Anonymity with Distributed Brokers

Sasu Tarkoma[1] and Christian Prehofer[2]

[1] University of Helsinki
[2] Fraunhofer Munich

Abstract. When issuing a one-shot or continuous content-based subscription, there is an inherent tradeoff between the privacy of the subscriber and the accuracy of the matching notifications. The former can be described in terms of how well the exposed information uniquely characterized the subscriber, and the latter how well the returned data items match the subscriber's real interests. In this paper, we define the partial order based k-filter anonymity, which generalizes k-location anonymity by allowing the generalization of filters using the subsumption or covering relation. We then develop techniques for assessing and maintaining content anonymity in distributed broker-based content routing and publish/subscribe networks, namely filter generalization based on partial orders, a logically centralized broker for determining pairwise k-anonymous subscription paths, and optimizations for pairwise anonymity level detection.

1 Introduction

Information targeting and delivery is a crucial requirement for current applications and services both on the fixed Internet and the mobile Internet. Publish/subscribe (pub/sub) and content-based routing offer expressive and flexible information targeting capabilities; however, they also introduce privacy concerns [22, 21, 28] that have not yet been fully addressed.

In recent years, content-based routing of information has been proposed for flexible and expressive data dissemination in distributed systems. In the well-known distributed pub/sub model, constraints called *filters* are used to make forwarding decisions on discrete messages. These decisions are made for each message separately based on its header or content.

When issuing a content-based subscription, there is an inherent tradeoff between the privacy of the subscriber and the accuracy of the result set. The former can be described in terms of how well the exposed information uniquely characterized the subscriber, and the latter how well the returned data items match the subscriber's real interests.

Issues pertaining to the privacy of a data set have been investigated in the work on k-anonymity for data sets. This technique has been applied for k-location anonymity that addresses the privacy of the end node with location-based queries [8, 4, 33]. The aim of these systems is to be able to retrieve *points*

of interest (POIs) given the current location of the client from a server without revealing this location in detail. Typically, a k nearest neighbour query is used to find the POIs. The client may require stricter privacy guarantees in the form of k-location anonymity, in which a cloaked region is used the represent the client location and this region needs to contain at least $k-1$ other client locations. In other words, given the region, it is not possible to distinguish between the k clients. As a limitation, the clients will receive POIs that are not relevant for them. We shall call such messages that contain information that the client has not subscribed beforehand as *false positives*.

The key questions addressed in the paper are the following:

- How to ensure that a given content subscription is k-anonymous (giving certain privacy protection)?
- How to ensure that in the network the subscription is not given to any broker that might violate the k-anonymity condition?

In this paper, we define the k-filter anonymity, which generalizes k-location anonymity by allowing the generalization of filters using the subsumption or covering relation. The idea is to extend anonymity requirements for logical locations in addition to physical locations. We present a formal framework for the filter-based k-anonymity and propose techniques for using the formal framework in a distributed setting. The key techniques are the following: partial-order-based generalization of filters and tracking of k-anonymity, a logically centralized broker for determining pairwise k-anonymous subscription paths, and optimizations for pairwise anonymity detection.

Subscriber privacy is enhanced by guaranteeing that a subscriber cannot be distinguished from a set of subscribers when the interests and matching content is delivered by the network. This delivery can happen in the form of broadcast within a certain area, or delivered using unicast or multicast across multiple brokers. Physical broadcast can be implemented in such a way that specific recipient identifiers are omitted; however, given the knowledge that only a single entity is interested in the data is sufficient to pinpoint the subscriber. Therefore we are motivated in enhancing the privacy of the interest registration service.

This paper is structured as follows: Section 2 presents the background and related work. Section 3 presents the basic assumptions and definitions for the paper, and Section 4 defines the k-filter anonymity in more detail. Section 5 presents the basic system model and Section 6 considers attacks against content-based pub/sub systems. The basic model is extended for multiple brokers in Section 7. Finally, we conclude the paper in Section 8.

2 Background

Various anonymization and privacy enhancing techniques have been proposed in the literature. The techniques can be categorized in many ways, for example under database, network and distributed systems, mobile systems, and general content anonymization categories.

Distributed systems that support user anonymity include Crowds [26, 29], Hordes [17], and Mist [2]. The Mist handles the problem of routing a message though a network while keeping the location of the sender hidden from intermediate devices. The system consists of a number of routers, known as Mist routers, ordered in a hierarchical structure. The anonymity degree metric has been proposed for evaluating route selection strategies that maximize the degree of anonymity of a system [15]. Systems such as Crowds, Hordes, and Mist do not address issues pertaining to data semantics or continuous queries.

A general way to anonymize tuples is to generalize an attribute, for example a number to a range. In this case, larger the range, the more information loss is introduced by the anonymization process. This approach can be used for both published events and subscriptions in a pub/sub system. For the case of subscriptions, the more general a subscription is made, the more unwanted traffic, false positives, will be generated. Both the information loss due to generalization and the false positives can be measured.

Location privacy has become an active research topic in recent years. The system model typically includes a set of clients and a centralized server that brokers points of interest to the clients. The k-location anonymity technique uses a cloaked region to represent the client location and this region needs to contain at least $k-1$ other client locations [13, 4, 19].

Two well-known techniques for location anonymity are *spatial cloaking* [8] and *transformation-based matching*. The former enlarge the user location q into a cloaked region Q' in a way that prevents the reconstruction of q from Q'. A server returns *points of interest (POI)* to a client using the more general Q' and the client has to prune the set to find the interesting elements. The latter technique evaluates the query in a transformed space in which the points and distances are encoded, for example using Hilbert ordering. The drawback of this method is the reduced accuracy. The SpaceTwist system utilizes an *anchor* and an interactive scheme to find an accurate enough location query result [33]. This system does not use a cloaked region or require transformations.

The k-anonymity criterion does not guarantee privacy against attackers using background knowledge. A new privacy criterion called *l-diversity* has been proposed to defend against such attacks [19]. The l-diversity requires that the distribution of sensitive attributes for each quasi-identifier have high entropy.

The problem of privacy preserving publishing for multiple users is to generate the k-anonymous table in such a way that the k-anonymity requirement for each user is met and at the same time the information loss is minimized [23].

Related work also includes privacy-preserving anonymization of set-valued data [32], anonymization of sparse high-dimensional sets [14], and cluster-based techniques for anonymization [1]. We observe that the aim of these systems is to anonymize a static data set.

A number of security services have been proposed for publish/subscribe; however, privacy is typically not addressed [9, 27, 24, 5]. The techniques proposed for k-anonymity and l-diversity are suitable for anonymizing a published event or a stream of events for a single user or for multiple users. The cluster-based tech-

nique uses quasi-identifiers of data records which are first clustered and then the cluster centers are published. The techniques can also be extended for subscriptions, which is investigated in this paper.

3 Preliminaries

We follow the general publish/subscribe filter model, in which a filter F is a stateless Boolean function that accepts a message as an argument. A filter is said to match a message n if and only if $F(n) = true$. The set of all notifications matched by a filter F is denoted by $N(F)$. A filter F_1 is said to cover a filter F_2, denoted by $F_1 \sqsupseteq F_2$, if and only if all messages that are matched by F_2 are also matched by F_1, i.e., $N(F_1) \supseteq N(F_2)$. The filter F_1 is equivalent to F_2, written $F_1 \equiv F_2$, if $F_1 \sqsupseteq F_2$ and $F_2 \sqsupseteq F_1$.

The $\sqsupseteq$ relation is reflexive, transitive, and anti-symmetric and defines the partial order $P = (X, \sqsupseteq)$ on the ground set X. We say that $x, y \in X$ are comparable if either $x \sqsupseteq y$ or $y \sqsupseteq x$. Otherwise they are incomparable. For each $x \in X$, we have the set of predecessors (resp. successors) of x in P given by $Pred(x) = \{y \in X | y \sqsupseteq x \text{ and } y \neq x\}$. Similarly, $Succ(x) = \{y \in X | x \sqsupseteq y \text{ and } y \neq x\}$. We also define immediate cover, denoted by $x \succ y$ if $x \sqsupseteq y$ and there is no element $z \in X$ such that $x \sqsupseteq z$ and $z \sqsupseteq y$. We have the immediate predecessors and successors, $ImPred(x) = \{y \in X | y \succ x \text{ and } y \neq x\}$. Similarly, $ImSucc(x) = \{y \in X | x \succ y \text{ and } y \neq x\}$.

The information needed in order to determine the $Succ$, $Pred$, $ImSucc$, and $ImPred$ functions can be determined using a poset data structure [6]. We observe that also a forest structure can be used; however, $ImPred$ and $ImSucc$ are subsets of the poset ones [31]. Insertion and deletion operations are linear time for the forest and the structure requires linear space. The poset involves more processing, because it needs to maintain the direct predecessor and successors sets.

Filter covering may be determined efficiently for simple predicate-based filters [6] and attribute filters with disjunctions [30]. In a basic data model messages are sets of typed tuples of the following format *(name,type,value)*, and filters consists of *attribute filters*, which are constraints on typed tuples. In this case, each attribute filter is the tuple *name,type,predicate*. Algorithms exist for arbitrary conjunctive filters [16], and also conjunctive tree queries [7].

It is possible to leverage the properties of the poset P in order to understand the privacy-enhancing possibilities of a given scenario. For instance, the height of the poset (number of chains), and the width of the poset (the maximum antichain) are important [3]. A probabilistic version of Sperner's theorem states that irrespective of the underlying probability distribution of selecting two subsets A and B randomly and independently according to a probability distribution from the set of subsets of an n-set S, the probability of $A \subseteq B$ is at least $\binom{n}{\lfloor \frac{n}{2} \rfloor}^{-1}$ if $n > 1$. This means that the underlying probability distribution does not matter. This result is interesting, because it gives a probability bound for any entity to infer $A \subseteq B$ if the parameter n is known. In practice the set

S could be the set of all landmarks or values for user profiles. A and B would then be observed user landmarks or profiles.

4 Filter Anonymity Based on Partial Orders

In this section, we present the basic definitions for k-filter anonymity. The k-filter anonymity definitions are generic in nature. The motivation for our framework is that k-anonymity for filters ensures that subscriptions cannot be distinguished in the given reference group defined using a partial order. A filter can have attributes that tie this reference group to some physical concepts, for example location. From a different point of view, this is the same as k-location anonymity with logically defined locations. For network of content-based brokers or routers, this approach would allow to ensure that a subscription (and any matching data) is not routed through subnetworks that have less than k other subscribers (thus giving some protection for the traffic towards the subscribers).

We assume that there is a data structure for managing the filters based on the partial order defined by the covering relation. In addition to the $Succ$, $Pred$, $ImSucc$, and $ImPred$ functions, there are two auxiliary functions. First, the function $Source(f)$ associates the input filter f with a set of unique subscriber identifiers. $Source(S)$ returns the unique subscriber identifiers for a given set of filters S, respectively. In our system model, these identifiers correspond to entities that have installed the filter in question. Second, the function $Ano(f)$ returns the required k-anonymity level specified by the issuer of the filter. If the value of this function is one, no k-filter anonymity is required. This corresponds to the regular routing and forwarding semantics of Siena and other content-based pub/sub systems.

Figure 1 presents an overview of the four key forms of k-filter anonymity. In the most simple case, we simply examine the size of the $Source$ set for the given filter f. On the other hand, this is quite restrictive and in many cases there is only one subscribing interface. Now, the main idea behind k-filter anonymity is to utilize the covering relations to further examine the sizes of the $Source$ sets.

The first form of k-filter anonymity is given by Definition 1 that considers the immediate successors of a given filter f. Second form is to consider all successors of f given by Definition 2. The interpretation of these two definitions is that the filter f contains subspaces that have at least k unique subscribers. Therefore this offers privacy in the sense that messages that match f can also be matched to at least $k-1$ other subscribers.

Definition 1. *A filter f has the property of k-immediate successor anonymity if and only if $|Source(ImSucc(f))| \geq k$. In other words, if the source set size of the direct successors is greater or equal to k.*

Definition 2. *A filter f has the property of k-successor anonymity if and only if $|Source(Succ(f))| \geq k$. In other words, if the source set size of the successors is greater or equal to k.*

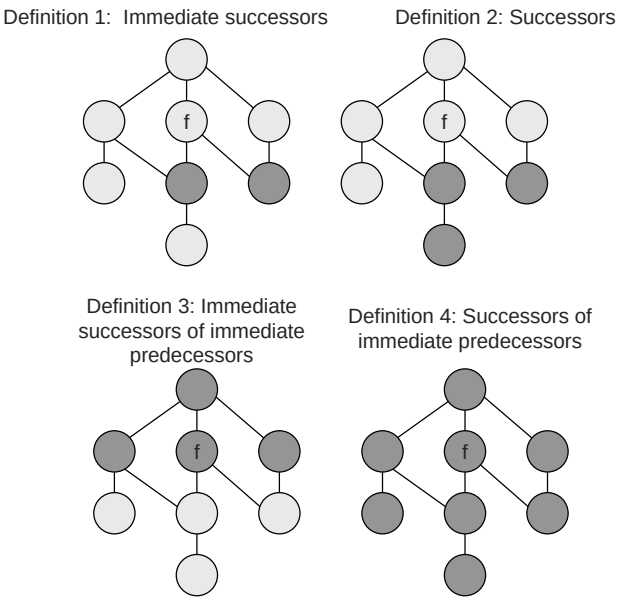

Fig. 1. Examples of k-filter anonymity.

Definition 4 and 3 offer stricter view to k-filter privacy. In the former, the immediate successors of the immediate predecessors of f must meet the criterion. In the latter, this successors of the immediate predecessors of f must meet the criterion, respectively. This can be seen as being stronger form of privacy since now f cannot be distinguished among at least k filters covered by its predecessors.

Definition 3. *A filter f has the property of K-immediate predecessor-successor anonymity if and only if $|Source(ImSucc(ImPred(f)))| \geq k$. In other words, if the number of subscribers of the immediate successors of the immediate predecessors to f is greater or equal to k.*

Definition 4. *A filter f has the property of k-immediate predecessor anonymity if and only if $|Source(Succ(ImPred(f)))| \geq k$. In other words, if the number of subscribers of the successors of the immediate predecessors to f is greater or equal to k.*

We observe that $|Succ(f)| \geq |Source(Succ(f))|$ and $|Pred(f)| \geq |Source(Pred(f))|$. $Source(Succ(ImPred(f))$ returns the set of identifiers associated with all the successors of the predecessors of the filter f. The size of this set therefore denotes the number of distinct identifiers associated with filters that have the

same parent as f and are thus more general than f. The interpretation is that if the size of this set is greater than k, the filter therefore cannot be distinguished among the $Source(Succ(ImPred(f)) \geq k$ identifiers.

The above definitions assume that there is an existing set of active filters (the base set X of the poset). We observe that the number of sources as used by in the definitions can be realized with minor book-keeping, for example for successors traversing to the root and updating the value of predecessors. As such the definitions are compatible with a data structure, such as a poset or forest, used for filter cover-based routing tables.

Now, it is also desirable to find a filter g that covers the input filter f, satisfies the given k-filter anonymity condition, and is minimal in terms of generality to f. We call this *augmenting k-filter anonymity*.

We consider two practical techniques to achieve this. First, the data structure can be preloaded with a typical subscription workload (set of filters). This approach requires no changes to the algorithms. Second, we can identify a set of filters whose union satisfies the k-anonymity condition. Then filter merging techniques [10, 20, 18, 30], either perfect or imperfect, can be applied to derive a single filter g that satisfies the conditions for augmenting k-filter anonymity.

5 Basic Technique Using a Broker

Our basic system model follows the models of well-known distributed pub/sub systems, such as Siena [6], Rebeca [12], and Hermes [25]. We consider two usage environments for k-filter anonymity. First, the definitions can be applied in a client-server fashion in which a trusted server or broker manages filters on behalf of the clients. The server can be a cluster-head in an ad hoc network, a local access server, or a rendezvous-point in an overlay network. Second, the definitions can be applied between pub/sub brokers so that local subscriber information exchanged by the brokers is k-anonymous.

Figure 2 presents an overview of the environment with the two usage scenarios. In the first usage case, the broker stores only filters that meet the required k-filter anonymity conditions. In the second case, the broker ensures that all client filters that are inserted into the external routing table, and propagated in the environment, meet the conditions. We discuss the realization of the latter environment in the next section.

In the client-server case, the following invariants illustrate k-filter anonymity during the subscription process from the viewpoint of a trusted broker that manages filters on behalf of the client:

- If unique subscribers for $f \geq k$ then follow normal processing semantics.
- If unique subscribers for $f < k$ then result in failure and remove the subscription.

In the broker to broker case, each broker is responsible for propagating only those filters that meet the given k-filter anonymity condition. Thus the condition can be seen as an input parameter for *route selection*. The problem of maintaining

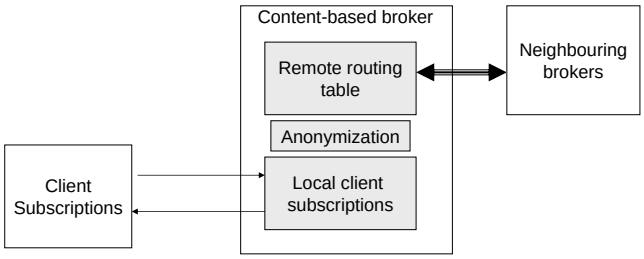

Fig. 2. Overview of the distributed environment.

k-anonymous routing paths requires recomputing the paths when the subscriber state changes in the network.

6 Attacks Against Subscribers

The main strategy of an active adversary is to introduce a bogus broker into the network that accepts subscriptions, and inject false subscriptions into the network and thus effectively reduce the value of k. Thus the approach is prone to the sybil attack [11], in which attackers gain quorum in the network by creating new identities.

In addition, if the attackers can influence the partial order, this may be used to compromise the privacy of the system by injecting specific subscriptions that are then indicative of the real content being subscribed. Attacks against the partial order can be mitigated by instrumenting the partial order beforehand and controlling updates to it.

These attacks against the subscription system can be mitigated by utilizing a centralized service that is used to activate subscriptions. The rendezvous points used by many overlay-based pub/sub systems are examples of such fixed points. We observe that the detection of malicious nodes depends on the network environment, for example ad hoc networks and wide-area networks are expected to utilize differing solutions. We outline an anonymizer service for the buildup and maintenance of k-anonymous paths in the presence of active malicious nodes in the next section.

7 Content-based k-anonymity Tomography

This technique involves a logically centralized trusted proxy that is used by clients and brokers to determine whether or not a given filter and destination combination satisfies the required k-filter anonymity condition. The idea is to offer a kind of network anonymity tomography service through the proxy. Each answer returned by such a service is only valid for some duration of time. This

approach requires that the service has information on subscribed filters and their respective frequencies, or some ways of approximating these without revealing any filters given to it by subscribers. This model also requires that as in the above case, the edge broker used by a client is trusted; however, the other brokers may or may not be trusted.

The anonymity tomography service proxy maintains a poset or a forest data structure and augments it with frequency counts and origin. We note that some book-keeping is required to track this information.

Given that we have two brokers A and B, and A wants to determine whether or not the filter f is k-anonymous from A to B (in the network/topology). A sends a request to the proxy, P, and asks for the broker to determine this. P either has this information already, or it needs to contact B and then retrieve at least k different statements regarding subscribers. The statements need to be verifiable (signature, some third party asserting that the subscription is f or is in relation to f). In addition, if the latter scheme is used, P has to be careful that the original f is not disclosed in such a way that violates the given value of k. The most straightforward way is to gather filter privacy data at the trusted proxy or proxies and then use that information locally at the proxy. In order for the system to be able to keep local subscriptions private, the first broker that a subscriber connects needs to be trusted; however, other brokers do not need to be trusted.

The proxy will then be consulted by clients and brokers to determine what parts of the subscription topology meet the given k-anonymity requirement. Figure 3 illustrates the distributed model in which a trusted anonymizer service is used to monitor subscription status and maintain k-filter anonymity requirements given by subscribers. In this case, a subscriber at broker A triggers anonymity testing by the anonymizing service. First, testing happens locally with broker A, then between A and C, and finally between C and B. The service can offer both passive monitoring and also more active anonymity maintenance in which bogus filters are created in order to maintain k-anonymity of a subscription.

The distributed solution suffers from the bootstrap problem, in which the propagation of filters is limited because the anonymity condition is not being met. The creation and placement of bogus filters solves this problem; however, it introduces overhead in terms of additional routing state and false positives. Another solution to the bootstrap problem is adaptive probing by first using a general filter and a smaller value of k, and then increasing detail and incrementing the value of k. This latter strategy also introduces overhead in terms of communication rounds, but does not necessarily increase state or the number of false positives (due to its probing nature).

This technique lends itself well to several optimizations. First, the brokers can cache the results for certain period of time. Second, if they know the maximum value of k, they can omit to update the proxy when this value is exceeded. Third, the brokers and the proxy can leverage the transitive nature of the partial order to perform optimizations, namely for Definitions 1 and 2 if a filter has already

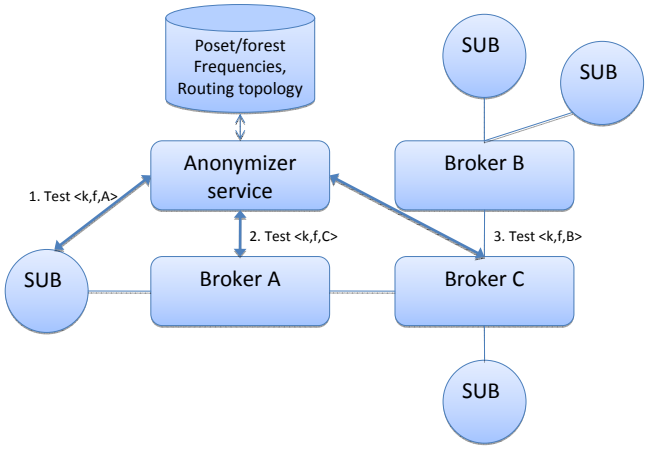

Fig. 3. The anonymizer service.

been found to meet have k-anonymity property, then any covering filter will also have the property.

8 Conclusions

In this paper, we have presented definitions for k-filter anonymity, investigated how it can be applied for content-based pub/sub systems, and outlined a distributed solution based on trusted edge brokers and a proxy service. Privacy of client subscriptions is expected to be an important requirement and techniques are needed to assess and enforce privacy requirements.

The definitions for k-filter anonymity generalize k-location anonymity by allowing the generalization of filters using the subsumption or covering relation. The framework has two important parameters, namely the structure of the partial order, and the value of k. In addition, a proxy service is needed if the anonymity property needs to be verified with untrusted brokers. We briefly outlined two techniques for verifying the property, namely creation of bogus subscriptions and adaptive probing.

The notion of k-filter anonymity appears to be useful in determining and maintaining certain levels of anonymity in distributed content-based systems, and it can serve as a building block for more sophisticated privacy enhancing technologies.

References

1. Aggarwal, G., Khuller, S., Feder, T.: Achieving anonymity via clustering. In: PODS. pp. 153–162 (2006)
2. Al-Muhtadi, J., Campbell, R., Kapadia, A., Mickunas, M.D., Yi, S.: Routing Through the Mist: Privacy Preserving Communication in Ubiquitous Computing Environments. In: ICDCS '02: Proceedings of the 22 nd International Conference on Distributed Computing Systems (ICDCS'02). p. 74. IEEE Computer Society, Washington, DC, USA (2002)
3. Anderson, I.: Combinatorics of Finite Sets. Dover (1987)
4. Bayardo, R.J., Agrawal, R.: Data privacy through optimal k-anonymization. In: ICDE '05: Proceedings of the 21st International Conference on Data Engineering. pp. 217–228. IEEE Computer Society, Washington, DC, USA (2005)
5. Belokosztolszki, A., Eyers, D.M., Pietzuch, P.R., Bacon, J., Moody, K.: Role-based access control for publish/subscribe middleware architectures. In: Proceeding of the 2nd International Workshop on Distributed Event-Based Systems (DEBS'03). ACM SIGMOD, San Diego, CA, U.S.A. (2003), `citeseer.ist.psu.edu/belokosztolszki03rolebased.html`
6. Carzaniga, A., Rosenblum, D.S., Wolf, A.L.: Design and evaluation of a wide-area event notification service. ACM Transactions on Computer Systems 19(3), 332–383 (Aug 2001), `http://www.cs.colorado.edu/~carzanig/papers/`
7. Chan, C.Y., Fan, W., Felber, P., Garofalakis, M.N., Rastogi, R.: Tree pattern aggregation for scalable XML data dissemination. In: VLDB. pp. 826–837 (2002)
8. Chow, C.Y., Mokbel, M.F., Liu, X.: A peer-to-peer spatial cloaking algorithm for anonymous location-based service. In: GIS '06: Proceedings of the 14th annual ACM international symposium on Advances in geographic information systems. pp. 171–178. ACM, New York, NY, USA (2006)
9. Corman, A.B., Schachte, P., Teague, V.: QUIP: a protocol for securing content in peer-to-peer publish/subscribe overlay networks. In: ACSC '07: Proceedings of the thirtieth Australasian conference on Computer science. pp. 35–40. Australian Computer Society, Inc., Darlinghurst, Australia, Australia (2007)
10. Crespo, A., Buyukkokten, O., Garcia-Molina, H.: Query merging: Improving query subscription processing in a multicast environment. IEEE Trans. Knowl. Data Eng. 15(1), 174–191 (2003)
11. Douceur, J.R.: The sybil attack. In: IPTPS '01: Revised Papers from the First International Workshop on Peer-to-Peer Systems. pp. 251–260. Springer-Verlag, London, UK (2002), `http://portal.acm.org/citation.cfm?id=687813`
12. Fiege, L., Gärtner, F.C., Kasten, O., Zeidler, A.: Supporting mobility in content-based publish/subscribe middleware. In: Endler, M., Schmidt, D.C. (eds.) Middleware. Lecture Notes in Computer Science, vol. 2672, pp. 103–122. Springer (2003)
13. Gedik, B., Liu, L.: Protecting location privacy with personalized k-anonymity: Architecture and algorithms. IEEE Trans. Mob. Comput. 7(1), 1–18 (2008)
14. Ghinita, G., Tao, Y., Kalnis, P.: On the anonymization of sparse high-dimensional data. In: ICDE. pp. 715–724 (2008)
15. Guan, Y., Fu, X., Bettati, R., Zhao, W.: An optimal strategy for anonymous communication protocols. In: ICDCS '02: Proceedings of the 22nd International Conference on Distributed Computing Systems (ICDCS'02). p. 257. IEEE Computer Society, Washington, DC, USA (2002)
16. Kiani, A., Shiri, N.: Containment of conjunctive queries with arithmetic expressions. In: CoopIS. pp. 439–452 (2005)

17. Levine, B.N., Shields, C.: Hordes: a multicast based protocol for anonymity. J. Comput. Secur. 10(3), 213–240 (2002)
18. Li, G., Hou, S., Jacobsen, H.A.: A unified approach to routing, covering and merging in publish/subscribe systems based on modified binary decision diagrams. In: ICDCS. pp. 447–457. IEEE Computer Society (2005)
19. Machanavajjhala, A., Gehrke, J., Kifer, D., Venkitasubramaniam, M.: ℓ-diversity: Privacy beyond κ-anonymity. In: ICDE '06: Proceedings of the 22nd International Conference on Data Engineering. p. 24. IEEE Computer Society, Washington, DC, USA (2006)
20. Mühl, G., Fiege, L.: Supporting covering and merging in content-based publish/subscribe systems: Beyond name/value pairs. IEEE Distributed Systems Online (DSOnline) 2(7) (2001), `http://computer.org/dsonline/0107/features/muh0107.htm`
21. Opyrchal, L., Prakash, A., Agrawal, A.: Supporting privacy policies in a publish-subscribe substrate for pervasive environments. Journal of Networks 2(1), 17–26 (2007)
22. Pareschi, L., Riboni, D., Agostini, A., Bettini, C.: Composition and generalization of context data for privacy preservation. In: PerCom. pp. 429–433 (2008)
23. Pei, J., Tao, Y., Li, J., Xiao, X.: Privacy preserving publishing on multiple quasi-identifiers. In: ICDE (March 2009)
24. Pietzuch, P., Bacon, J.: Hermes: A distributed event-based middleware architecture. In: Proceedings of the 1st International Workshop on Distributed Event-Based Systems (DEBS'02) (2002)
25. Pietzuch, P.R.: Hermes: A Scalable Event-Based Middleware. Ph.D. thesis, Computer Laboratory, Queens' College, University of Cambridge (February 2004)
26. Reiter, M.K., Rubin, A.D.: Anonymous Web transactions with crowds. Communications of the ACM 42(2), 32–48 (1999), `citeseer.ist.psu.edu/reiter99anonymous.html`
27. Srivatsa, M., Liu, L.: Securing publish-subscribe overlay services with EventGuard. In: CCS '05: Proceedings of the 12th ACM conference on Computer and communications security. pp. 289–298. ACM, New York, NY, USA (2005)
28. Srivatsa, M., Liu, L.: Secure event dissemination in publish-subscribe networks. In: ICDCS '07: Proceedings of the 27th International Conference on Distributed Computing Systems. p. 22. IEEE Computer Society, Washington, DC, USA (2007)
29. Sui, H., Wang, J., Chen, J., Chen, S.: The cost of becoming anonymous: on the participant payload in crowds. Inf. Process. Lett. 90(2), 81–86 (2004)
30. Tarkoma, S., Kangasharju, J.: Filter merging for efficient information dissemination. In: CoopIS. pp. 274–291 (2005)
31. Tarkoma, S., Kangasharju, J.: Optimizing Content-based Routers: Posets and Forests. Distributed Computing 19(1), 62–77 (Sep 2006)
32. Terrovitis, M., Mamoulis, N., Kalnis, P.: Privacy-preserving anonymization of set-valued data. Proc. VLDB Endow. 1(1), 115–125 (2008)
33. Yiu, M.L., Jensen, C.S., Huang, X., Lu, H.: Spacetwist: Managing the trade-offs among location privacy, query performance, and query accuracy in mobile services. In: ICDE. pp. 366–375 (2008)