PANACEA: Tunable Privacy for Access Controlled Data in Peer-to-Peer Systems

Rammohan Narendula, Thanasis G. Papaioannou, Zoltán Miklós and Karl Aberer

School of Computer and Communication Sciences, EPFL, Switzerland

Email: firstname.lastname@epfl.ch

Abstract—**Peer-to-peer paradigm is increasingly employed for organizing distributed resources for various applications, e.g. content distribution, open storage grid etc. In open environments, even when proper access control mechanisms supervise the access to the resources, privacy issues may arise depending on the application. In this paper, we introduce, PANACEA, a system that offers high and tunable privacy based on an innovative resource indexing approach. In our case, privacy has two aspects: the deducibility of a resource's existence/non-existence and the discovery of the provider of the resource. We systematically study the privacy that can be provided by the proposed system and compare its effectiveness as related to conventional P2P systems. Employing probabilistic reasoning, we quantify the privacy and analytically derive that PANACEA can offer high privacy, while preserving high search efficiency for authorized users. Moreover, the privacy offered by the proposed system can be tuned according to the specific application needs.**

I. INTRODUCTION

Peer-to-peer (P2P) systems are increasingly used in many distributed application domains, e.g. content distribution, file sharing, open storage grids, video streaming, etc. However, users typically expect to be able to use these systems to share access-controlled and (semi-) private data. Conventional P2P systems should be properly adapted to meet the access control requirements of such applications. Typical approaches for data access control in open environments include cryptographic methods [1], Digital Rights Management (DRM) technologies, and trust-based methods [2], which require complicated key distribution and management. We consider a simpler, yet effective, approach for data access control in P2P systems: We assume that resources reside at the publisher nodes itself, to ensure that access control is enforced safely in an untrusted P2P environment. A user directly presents his credentials to the publishing peer of a particular resource after locating the resource in the P2P overlay. The publishing peer replies the query after applying its *local authorization policies*.

P2P systems typically try to maximize search efficiency. To this extreme, structured P2P systems, such as Chord [3], Kademlia [4], employ an index implemented as a Distributed Hash Table (DHT) over the P2P overlay. Such an index typically consists of index entries of the form $(key, value)$-pairs, where the key is the resource identifier (often produced by one-way hash functions, e.g. MD5), while the value is the peer identifier, where the resource is stored. Indeed, as shown in [3], such an index significantly improves the search costs, in terms of both query latency and communication overhead. However, as index entries are hosted on arbitrary and often untrusted

nodes, access to the index entries cannot be controlled by the peers that publish their data to the index. Thus, the index reveals both the existence/non-existence and the location (i.e. publishing peer) of each queried resource, hence, data privacy is breached. We refer the former privacy aspect concerning resource existence/non-existence as *resource privacy*, while the latter one concerning resource location as *provider privacy*. On the other extreme, unstructured P2P systems, such as Gnutella [5], employ no index and limited-hop flooding is used for locating the queried data, which incurs high latency and communication overhead, yet, with no guarantees on the data discovery. However, if access-controlled, unstructured P2P systems can provide the highest data privacy by answering queries only to authorized users. Thus, there is a *trade-off* between search efficiency and data privacy in this context.

In this paper, we explore this trade-off and propose a PrivAcy preserviNg Access-ControllEd (PANACEA) P2P system that combines high data privacy (both resource and provider privacies) and high search efficiency for authorized users. To carefully quantify privacy offered by PANACEA, we define the privacy objectives using probabilistic reasoning. We analytically study the privacy and the search efficiency/overhead of the PANACEA, as related to structured and unstructured P2P systems. The parameters of PANACEA can be tuned so that the trade-off between privacy and search efficiency is set according to the application needs. Numerically evaluating our analytic results for practical systems, we demonstrate that, with proper values for the parameters of PANACEA, authorized users almost always find the queried resources at a very low search overhead, while unauthorized users can deduce the existence of a resource and its provider with a very low probability. Moreover, the communication overhead is high for unauthorized users. Figure 1 illustrates the position of PANACEA as related to structured and unstructured P2P access-controlled systems in the three-dimensional space <provider privacy, resource privacy, search efficiency>, employing the terminology of [6]. To the best of our knowledge, PANACEA is the first approach that concurrently addresses resource and provider privacies in access-controlled systems.

Note that the specific authorization policy and the format of credentials are orthogonal to current scope of the paper. As a result, PANACEA mechanism can be employed by providers with different access control techniques, such as role based access control, discretionary access control or attribute-based access control, all existing in the system simultaneously.
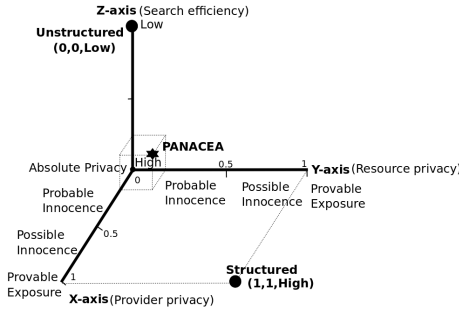
Fig. 1. Position of various systems on privacy and search efficiency axes

Fig. 2. Privacy preserving publishing in PANACEA

The remainder of the paper is organized as follows: In Section II, we describe the publishing and searching mechanisms in PANACEA. In Section III, we analytically derive the privacy properties and the search overhead. In Section IV, we discuss PANACEA's effectiveness against privacy breach attempts. In Section V, we numerically evaluate the proposed system and demonstrate its tunability. In Section VI, we discuss the related work, and we conclude in Section VII.

II. THE PANACEA SYSTEM

In this section, we present the proposed PANACEA system and explain how the resource and provider privacies are achieved. As already mentioned, resource privacy has two aspects: one concerns with existing and shared resources, the second with non-existing resources: an unauthorized user should not be able to determine that a particular resource does not exist in the system, a property which is inherent to unstructured systems.

A. Overview

The proposed system aims to combine look up efficiency of structured peer-to-peer systems with high resource privacy that is offered in unstructured ones. PANACEA employs a DHT to host the proposed resource and provider privacy-preserving (RPP) index, which addresses both resource and provider privacies of the shared resources. However, as explained later in this section, PANACEA indexes only a subset of the resources into the DHT; this is a necessary characteristic for providing resource privacy. The rest of the resources are located by flooding, similar to the unstructured P2P systems. As a result, PANACEA acts partly as an unstructured P2P system for the resources not indexed in the DHT, and partly as structured systems which can be configured as explained later in the section.

The proposed indexing mechanism consists of tunable parameters that allow the application designer to choose between strong privacy guarantees and increased search efficiency based on the specific application needs. The tuning determines the position of the resulting system in the graph of Figure 1, as compared to structured and unstructured P2P systems. We highlight the publishing and search mechanisms in Section II-B and in Section II-C respectively.
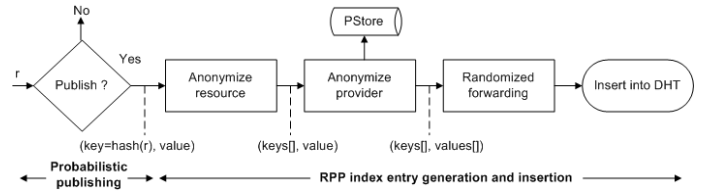
B. Privacy Preserving Publishing

PANACEA achieves the resource and provider privacy goals with privacy-aware publishing mechanism, which mainly employs two novel techniques:

1) Probabilistic publishing
2) Resource and provider privacy preserving (RPP) indexing

The approach is illustrated in Figure 2. Recall that resource privacy concerns unauthorized users deducing the presence of the existent resources and the absence of non-existent resources in the system. The *RPP indexing* is introduced to address the former aspect. However, as the absence of a key in the DHT of PANACEA could enable a user to deduce its non-existence in the network, we also introduce *probabilistic publishing* described below, which addresses the latter aspect of resource privacy.

1) Probabilistic publishing: PANACEA, instead of announcing every resource into the DHT, as in structured P2P systems, announces a resource with a system- defined probability μ- and creates an RPP-index entry, which is described later. Thus, non-existence of a particular resource in the system can not be deduced with certainty from the DHT as absence of an index entry for a key does not necessarily mean non-existence of its corresponding resource in the system. Due to probabilistic publishing, PANACEA acts as a hybrid semi-structured P2P system. All the resources that are not announced in the DHT are discovered using flooding for a limited number of hops determined by a time-to-live (TTL) set on search queries.

2) The RPP index generation: We resort to *k-anonymization* techniques to achieve both resource and provider privacies for the resources selected to be announced into the DHT by the probabilistic publishing phase. A k-anonymization technique typically anonymizes a data item by hiding it inside a list of k data items so that a user can not make out anything about the original intended data item. Instead of having a $(key, value)$ pair as an index entry for a resource, as in structured P2P systems, we propose the index entry pair to consist of a list of keys and a list of values, i.e. $(key[], value[])$, which is derived by applying *resource* and *provider anonymization* as explained in the following. We refer to such an index entry as *(m,n)-index entry*, where m refers to cardinality of key list and n refers to that of value list. In this terminology, an index entry of the conventional structured P2P systems can be seen as a $(1,1)$-index entry. From an (m,n)-entry, we claim that a user can deduce the

existence of a resource with probability $\frac{1}{m}$ and provider by $\frac{1}{n}$. Clearly, the greater the values for m and n, lower the probability to deduce and hence, higher the privacy they offer, but at an increased publishing and searching overhead, as explained later in this section. A $(1,1)$-entry denotes the fact a *particular* peer publishes a *particular* resource, whereas a (m,n)-entry means that *one or more of these* listed peers publish *one or more of the* listed resources.

Resource anonymization: Once a resource is selected for publishing, the corresponding $(1,1)$-index is selected for resource anonymization to convert it into a $(m,1)$-index. We consider two techniques to choose the $m-1$ entries of the key list for anonymization: i) select *random* keys, ii) select *semantically closer* keys. Below, we briefly outline both the techniques and compare them against potential attacks for inferring the valid key from the list.

Note that human-readable plain text keys (i.e resource names), employed by users to refer to the resources, are mapped by a hash function to the system key space (i.e. resource ids). We refer to such a resource namespace as R, the equivalent resource key space as K, and the hash function as $H : R \rightarrow K$. When a resource with name r is to be anonymized, in fact, it is the $(1,1)$-entry for key $H(r)$ that is anonymized.

In the *randomized anonymization* technique, m-1 number of entries chosen randomly from the set R are hashed to the set K to make the m entries needed for a $(m,1)$-index.

In *semantics-based anonymization*, m-1 resources are selected from the set $S \subset R$ ($|S| \ll |R|$) of resources semantically correlated to r and their corresponding keys form the m entries of a $(m,1)$-index. We assume that the user feeds such keys and argue that automating such a task is a research study in itself.

In order to infer the valid key from the list of m keys, an adversary first has to derive the resource names from the resource ids using brute force approach. Such a dictionary attack is computationally easier on semantics-based technique as $|S| \ll |R|$. However, inferring a valid name from the random entries selected in randomized anonymization is simpler than inferring from semantically closer keys. In the rest of the draft, we assume that a user can learn from an (m,n)-entry that each of the m keys is equally probable of being a valid entry with probability $\frac{1}{m}$.

Provider anonymization: After resource anonymization, the resulting $(m,1)$-index entry is fed into the provider anonymizer. The provider list is populated with n number of entries with the providing peer itself being one of them. The other $n-1$ entries are chosen randomly from the *Provider Store (PStore)*- a local database of provider ids. We assume that PStores at each peer are populated with some providers in the network during the network bootstrapping phase, which are selectively, expanded incrementally over time with providers listed in the the $put()$ requests traversing through the peer.

3) The RPP index insertion: After an (m,n)-index entry is constructed by a publishing peer, it has to be inserted into the network using the DHT's $put()$ method, with a slightly modified semantics for the $put()$ interface over the conventional structured systems. However, this index entry must be published *anonymously* as next-hop node in the routing, can easily deduce the valid publisher node from the (m,n)-entry, as the initiator node itself is the publisher.

Anonymous publishing: To anonymize the node that initiates the insertion request, we propose that a *randomized routing (RR)* phase preceding the DHT's $put()$ operation. Each node that receives the insertion request decides to forward it to a random node in its PStore with a certain probability λ or initiate the DHT routing of the $put(m,n)$ method with probability $1-\lambda$. Forwarding $put()$ requests to random nodes in the overlay instead of neighboring ones, as was introduced in [6], with a few modifications is necessary; otherwise, if a node received a $put()$ request from a neighboring node that is contained in the provider list, it could assign a high probability that its neighbor is the publishing node. However, randomly choosing next hop nodes from PStore at each peer ensures that the identity of the valid resource provider is not leaked during resource publishing in the DHT, as there is no way for a peer that receives $put(m,n)$ request to determine whether the preceding peer is the publisher or a relay node.

The RR phase introduces additional communication overhead compared to DHT's $put()$. This process can be viewed as a geometric distribution with parameter λ. Therefore, if X is the random variable that describes the number of hops of a $put()$ request, then the probability that it travels x hops is given by:

$$P(X = x) \quad = \quad \lambda^{x-1}(1-\lambda)$$

The expected number of hops in randomized routing is given by the mean of the geometric distribution, i.e. $E(X) = 1/\lambda$. To this end, we assume that PStore caches the IP address along with each provider id. Thus, the relaying of $put()$ can happen in $O(1)$. Moreover, the IP address for each provider is stored in the provider list of the (m,n)-index entry.

Insertion into the DHT: When a node in the RR routing decides to enter the DHT routing, it invokes $put(m,n)$ operation, which is implemented using conventional $put()$ interface as follows. Note that the $put()$ operation inserts only a $(1,1)$-index entry. Yet, the same can be used to insert a $(1,n)$ entry, as the $value$ field is not used in the DHT routing. Thus $put()$ can act as $put(1,n)$. Hence, we propose to convert the $put(m,n)$ request into m number of $put(1,n)$ requests, with each of the m keys (pivot), acting as the *key* and the n number of providers acting as the *value list* required for a $put(1,n)$. Subsequently, for every $get()$ request on any of the m keys, the same set of n providers is returned, as the index entry is present on m nodes responsible for each of the keys. However, such a reply to $get()$ request can not reveal the presence or absence of the resource in the system, because of the way the index was constructed.

Note that, since the keys in an index entry are chosen independently by peers, key collisions in the DHT are possible. Key collisions also happen when multiple copies of the same resource are inserted into the DHT. We propose that the list

of providers in the new index entry is simply appended to the list of already existing providers for the collided key.

C. Searching

When a peer searches for a resource with id r, it executes *get(r)*. If a (m, n)-index entry was published in the DHT with r being one of the m ids, then the peer returns this index entry to the searcher. Subsequently, the searcher contacts all the n providers. Note that, in general, a user does not know in advance to which providers he is authorized for resource r, unless he has contacted them before for the same resource. In the latter case, the searcher could select only certain nodes from n to contact. Once a provider is selected, it can be reached in $O(1)$, since its IP address is maintained in the index entry along with the provider id. In case of multiple (m, n) entries matched because of key collisions, we assume that the searcher contacts all the providers listed in all the entries. If no (m, n)-entry exists in the DHT, then the searcher floods the search request to its neighbors in the overlay with a limited hop-count (Time-to-Live, TTL), as would happen in an unstructured peer-to-peer system.

However, in case of multiple providers for same resource, an (m, n)-index entry for an existing resource may not contain all the providers of that resource in the system because of probabilistic publishing in PANACEA. In other words, the index entry is not always *complete*. As a result, a searcher may not be able to reach the provider where he is authorized through the RPP index. Therefore, even if an (m, n)-index entry is present in the index, the searcher may have to employ limited-hop flooding. However, the probability that a query is both sent to the DHT and flooded over the overlay is very low, as shown in Section III, for reasonable values of μ, the DHT index entries are complete with high probability and contain all potential providers of a resource.

A simple modification in the publishing algorithm can significantly increase the completeness of the DHT. In the randomized routing, each relay hop of the $put()$ request adds itself in the provider list of the request, in case it has one of the resources listed. This approach does not violate privacy and increases the look up efficiency of the DHT.

III. ANALYSIS

In this section, we analytically study the privacy offered by PANACEA and estimate its communication overhead. We evaluate the privacy breach that can be achieved by a user who has *complete information* of the underlying system mechanism (PANACEA, structured or unstructured) and queries the system for a particular resource. We consider this as the worst case scenario for privacy breach with a single-query. Other cases where users have limited or no knowledge of the systems and case of attacks employing multiple queries to violate privacy of PANACEA, are discussed in Section IV.

In the analysis, we employ probabilistic reasoning and use the following notation:

  i) P_K the probability for a user to deduce the existence of a certain *existing/shared* resource.

  ii) P_V the probability for a user to find the provider of a certain resource.

  iii) P_- the probability for a user to deduce the non-existence of a certain *non-existing* resource.

Moreover, we consider these probabilities separately, for the cases that a user is authorized or not (unauthorized), to access a copy of the requested resource and denote them as $P_{K,a}$, $P_{V,a}$ and $P_{K,u}$, $P_{V,u}$ respectively. Note that it does not make sense to consider separately P_- for authorized and unauthorized users for non-existent resources. We use superscript U and S to denote metrics for unstructured and structured systems respectively, and a metric without any superscript is used for PANACEA i.e., $P_{K,u}^U$ refers to unstructured systems and equivalent metric for PANACEA is denoted by $P_{K,u}$.

Definition 1: An access-controlled system is said to provide *higher* privacy if it promises:

  i) Lower probabilities for $P_{K,u}$, $P_{V,u}$, which addresses an unauthorized user deducing a resource's presence and its provider.

  ii) Lower probabilities for P_-, which addresses a user deducing a resource's non-existence.

Under this definition of privacy, any privacy-enabling access control mechanism should try to minimize P_u, P_-. However, the search efficiency of the privacy-enabling mechanism should remain high, i.e. : a) $P_{K,a}$, $P_{V,a}$ should ideally be 1, which addresses authorized users ability to access the resources and discovering the providers, and b) the search communication cost $C_{s,a}$ should be kept low.

To this end, we require that the PANACEA should achieve the best of the unstructured and structured systems, w.r.t the privacy and search efficiency. Specifically, the *privacy efficiency objectives* for PANACEA are:

- $P_{K,u} \sim P_{K,u}^U$ and $P_{V,u} \sim P_{V,u}^U$ i.e., PANACEA should be closer to privacy-strong unstructured systems in the case of unauthorized searches
- $P_- \sim P_-^U$ i.e., PANACEA should act like unstructured systems in hiding non-existence of resources
- $C_{s,u} \sim C_{s,u}^U$ i.e., unauthorized searches should be as costly as that of unstructured systems
- $C_{s,-} \sim C_{s,-}^U$

On the other hand, the *search efficiency objectives* for PANACEA are:

- $P_{K,a} \sim P_{K,a}^S$ and $P_{V,a} \sim P_{V,a}^S$, i.e. PANACEA's privacy mechanisms should be transparent to authorized users like in privacy-free structured systems
- $C_{s,a} \sim C_{s,a}^S$, i.e. the search performance of PANACEA should be closer to that structured systems

Next, we analytically model the privacy and performance properties of the three systems by assuming a user assuming the roles of an authorized and unauthorized user to access an existing resource r. Let N be the number of peers in the system and N_c be the expected number of copies of r in the system. $N_a \leq N_c$ be the number of providers where the user is authorized to access r. For an existent resource $N_c \geq 1$ and $N_a \geq 1$, $N_a = 0$ for an authorized and unauthorized

user, respectively. For non-existent resources, $N_c = 0$. For simplicity, we assume, without loss of generality, that both N_c and N_a nodes are uniformly distributed in the network.

In an unstructured system, the search involves limited-hop flooding. Once a resource is found, if the user is authenticated, the query is directly replied to him. Otherwise, the query is further flooded. In PANACEA, if the requested resource that the user is authorized to access, is not indexed in PANACEA, he can discover its presence, only when at least one of the N_a providers is contacted in the search process. We assume that the providers do not respond to search queries from unauthorized users, in order not to compromise the resource and provider privacies.

A. Structured P2P systems

As structured P2P systems have index available for all users, they offer no resource and provider privacies. Therefore,

$$P_{K,a}^S = P_{V,a}^S = P_-^S = P_{K,u}^S = P_{V,u}^S = 1 \qquad (1)$$

The associated publishing and discovery costs C_p, C_s for a single resource (copy) are given by:

$$C_{p,a}^S = C_{s,a}^S = C_{s,u}^S = C_{s,-}^S = O(logN) \qquad (2)$$

B. Unstructured P2P systems

We assume that the peers are organized in an overlay graph with average degree d. We first quantify the *expected number* of nodes α that are visited by flooding with TTL set to ttl. Since it is assumed that resource copies are uniformly distributed, we derive that $p = \frac{N_a}{N}$ is the probability of visiting one of the N_a nodes in next-hop. To this end, α is given by:

$$\alpha = \sum_{i=1}^{ttl} T_i, \qquad \text{where} \qquad (3)$$

$$T_i = \begin{cases} d, \quad \text{for } i = 1 \\ \sum_{j=0}^{T_{i-1}} \binom{T_{i-1}}{j} p^j (1-p)^{(T_{i-1}-j)} (T_{i-1} - j)(d-1), \\ \quad \text{for } i > 1 \end{cases}$$

In unstructured P2P systems, $P_{K,a}^U$ and $P_{V,a}^U$ are always the *same* as they are discovered at the same time, and are given by:

$$P_{K,a}^U = P_{V,a}^U = 1 - \left(1 - \frac{N_a}{N}\right)^\alpha \qquad (4)$$

The expected publication and query costs are given by:

$$C_{s,a}^U = \alpha \qquad \text{and} \qquad C_{p,a}^U = 0 \qquad (5)$$

Since no provider would reply to unauthorized queries, the presence of the resource can never be deduced. Therefore, these systems offer the highest resource and provider privacies.

$$P_{K,u}^U = P_{V,u}^U = 0 \qquad (6)$$

The expected query cost is given by:

$$C_{s,u}^U = \alpha \qquad (7)$$

Regarding the deduction of the non-existence of a particular resource, it suffices to calculate the probability to discover if a *single* copy exists into the system. However, observe that discovering a non-existent resource is similar to discovering an unauthorized existent resource in the unstructured P2P system. Thus, non-existence can be deduced with probability:

$$P_-^U = P_{K,u}^U \qquad (8)$$

Also, the associated query cost is given by:

$$C_{s,-}^U = \alpha \qquad (9)$$

C. PANACEA

Note that if the user finds (m, n)-entry for r in the DHT, as an *authorized* user, he can deduce the existence of r with probability 1, in case one of the N_a nodes has published that (m, n)-entry, since as part of the search process, he contacts all the providers listed in the index entry. However, if i number of nodes, where the user is not authorized to, publish (m, n)-entries, the user can deduce the existence of the resource with the probability of being one out of the expected number of distinct keys $(i(m-1)+1)(1-f_k(N_r))$ in the DHT entries for this resource and with probability $P_{K,a}^U$ to find it by flooding. The parameter f_k is the probability of *key collisions* in the key lists of the different (m, n)-index entries in the DHT when N_r resources are stored in the system and it is derived as the number of m disjoint partitions of $|R| - N_r$ resource keys over the number of their total combinations into sets of size m. If the requested resource is not published into the DHT, the user learns about the resource with probability $P_{K,a}^U$. This reasoning is captured in the following equation:

$$P_{K,a} = \left[1 - (1-\mu)^{N_a}\right] \cdot 1 +$$
$$(1-\mu)^{N_a} \cdot \sum_{i=1}^{N_c-N_a} \binom{N_c - N_a}{i} \mu^i (1-\mu)^{(N_c-N_a-i)}$$
$$\cdot \left[\frac{1}{(i(m-1)+1)(1-f_k(N_r))} + P_{K,a}^U\right] + (1-\mu)^{N_c} \cdot P_{K,a}^U,$$
$$\text{where} \quad f_k(N_r) = \frac{\frac{|R|-N_r}{m}}{\binom{|R|-N_r}{m}}$$
$$(10)$$

We apply the same analogy used in (10) to derive $P_{V,a}$. However, when a number i of providers where the user is not authorized to, publish into the DHT, the expected number of distinct providers in the index entry is approximated by $i \cdot n \cdot (1-f_v)$, where f_v is the probability of *provider collisions* among the provider lists of different (m, n)-index entries in the DHT and it is derived similarly to f_k.

$$P_{V,a} = \left[1 - (1-\mu)^{N_a}\right] \cdot 1 +$$
$$(1-\mu)^{N_a} \cdot \sum_{i=1}^{N_c-N_a} \binom{N_c - N_a}{i} \mu^i (1-\mu)^{(N_c-N_a-i)}$$
$$\cdot \left[\frac{1}{(i \cdot n)(1-f_v)} + P_{V,a}^U\right] + (1-\mu)^{N_c} \cdot P_{V,a}^U$$
$$(11)$$

Recall that, for each (m,n)-index entry insertion, m number of index entries have to be inserted into the DHT, each incurring a cost of $C_{p,a}$. The searching cost depends on whether one of N_a nodes published into the DHT or not. If none of them published, $C_{s,a}^U$ should be accounted for. Each search also incurs the look up cost on the DHT. Hence, the expected publishing and searching costs are given by:

$$C_{p,a} = \sum_{i=1}^{N_c} \binom{N_c}{i} \mu^i (1-\mu)^{(N_c-i)} \cdot i \cdot \left(\frac{1}{\lambda} + m \cdot C_{p,a}^S \right)$$

$$C_{s,a} = C_{s,a}^S + (1-\mu)^{N_c} \cdot C_{s,a}^U + \sum_{i=1}^{N_c} \binom{N_c}{i} \mu^i (1-\mu)^{(N_c-i)}$$
$$\cdot \left[i \cdot n \cdot (1-f_v) + \left(1 - \frac{N_a}{N_c} \right)^i C_{s,a}^U \right] \tag{12}$$

We can derive $P_{K,u}, P_{V,u}$ from (10) and (11) respectively by observing that for an unauthorized user $N_a = 0$ and replacing $P_{K,a}^U, P_{V,a}^U$ with $P_{K,u}^U, P_{V,u}^U$ respectively.

The expected searching cost for an unauthorized user is given by:

$$C_{s,u} = C_{s,u}^S + (1-\mu)^{N_c} \cdot C_{s,u}^U +$$
$$\sum_{i=1}^{N_c} \binom{N_c}{i} \mu^i (1-\mu)^{(N_c-i)} \left[i \cdot n \cdot (1-f_v) + C_{s,u}^U \right] \tag{13}$$

Finally, we derive P_-, i.e. the probability to deduce the non-existence of a non-existent resource. Given an event space $\Omega = \{Find, NotFind\}$, representing that a non-existent resource is found or not in the DHT respectively, the probability of non-existence $(Pr(NE))$, as deduced by a user, is given by:

$$P_- = Pr(NE \mid \Omega) = Pr(NE \mid NotFind) \cdot Pr(NotFind)$$
$$+ Pr(NE \mid Find) \cdot Pr(Find) \text{, where}$$
$$Pr(NE \mid NotFind) = P_{K,u}^U$$
$$Pr(NE \mid Find) = \frac{m-1}{m}$$
$$Pr(Find) = \left[1 - \left(1 - \frac{1}{|R|} \right)^{\mu N_r (m-1)} \right]$$
$$Pr(NotFind) = 1 - Pr(Find) \tag{14}$$

N_r is the total number of resources in the system and R is the resource namespace. $Pr(NE \mid NotFind)$ expresses the probability that a resource is non-existent, given that it is not found in the DHT. This is similar to the probability of deducing the existence of an unauthorized resource for a user in unstructured P2P systems, because an existing resource is same as a non-existing resource for an unauthorized user. $Pr(NE \mid Find)$ is the probability that the resource corresponding to the key does not exist. $Pr(Find)$ expresses the probability that an arbitrary resource name from space R may have been inserted into the index, while $Pr(NotFind)$

is the complement of $Pr(Find)$. Observe that P_- is minimal (~ 0) for reasonable values of the various parameters.

Finally, we estimate the expected query cost to deduce the non-existence. The user first searches for an index entry and then employs flooding, hence,

$$C_{s,-} = C_{s,-}^S + C_{s,-}^U \tag{15}$$

IV. PRIVACY BREACH ATTEMPTS

In this section, we analyze the privacy properties of PANACEA with repetitive querying attempts of privacy breach by an adversary with complete information on the parameters of the system. First, we consider the case where the adversary is incapable of employing traffic sniffing and analysis in the network. Consider a resource r published by multiple providers into the DHT. Since multiple (m,n)-index entries exist for r, the provider list is expected to be larger than that of a resource published by only one provider. An adversary, observing such larger provider lists than the usual ones in his repetitive queries for different resources, can learn that r exists in the system with higher probability, than other resources. However, this breach in resource privacy is expected to be small for two reasons: i) The provider lists of (m,n)-index entries also grow for randomly selected keys over time, as for each $put(m,n)$ request $m-1$ keys are randomly selected from $|R|$ and they may collide with existing keys in the DHT. ii) Provider lists are not expected to grow linearly with the number of copies of real or fake keys due to collisions in the provider lists.

If we assume that traffic sniffing and analysis are also employed by the adversary, then provider privacy can be negatively affected. Specifically, if the adversary monitors the input and output interfaces of a node publishing a new resource in the DHT, he can deduce that the node is the provider of one of the m resource keys that the adversary observes. However, resource privacy is not breached as resource existence can only be deduced with probability $1/m$.

If the adversary does not have complete information for PANACEA, then the achievable resource and provider privacies of the system further improve. For example, we consider the extreme case where the adversary has no information on the parameters of the system. In this case, the adversary obtains a provider list in a query for a certain resource that exists in the DHT. However, assuming that he is not authorized to that resource, there is no way for him to deduce whether a key in the DHT corresponds to a real resource or not, and therefore his probabilities $P_{K,u}, P_{V,u}, P_-$ become 0.

The study of the effectiveness of PANACEA against more sophisticated attack strategies is left for future work.

V. EVALUATION

In this section, we numerically evaluate PANACEA as related to unstructured and structured access-controlled P2P systems based on the analysis of Section III. Unless otherwise specified, we assume that each system consists of $N = 10000$ peers with the unstructured system organized in an overlay
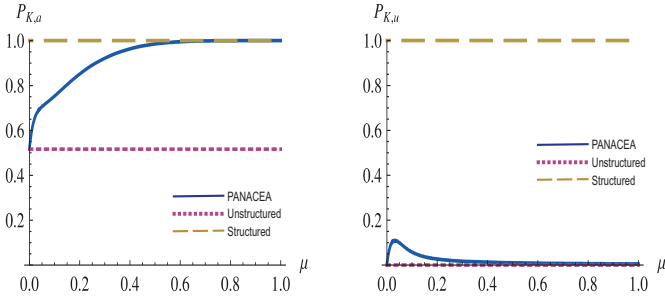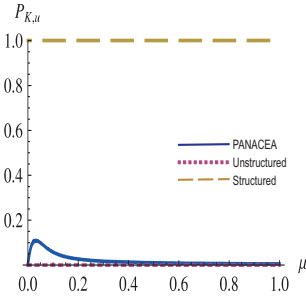
Fig. 3. $P_{K,a}$ vs μ

Fig. 4. $P_{K,u}$ vs μ

Fig. 7. $C_{s,a}$ vs μ

Fig. 8. $C_{s,u}$ vs μ

graph of average degree $d = 4$. The searcher is assumed to be authorized to $N_a = 5$ out of $N_c = 50$ copies of the queried resource. The TTL employed for limited-hop flooding is $ttl = 6$ and the parameters of PANACEA are: $m = 5, n = 5, \lambda = 0.2, \mu = 0.6$. The collision parameters are specified as $f_k = 0.1$, $f_v = 0.4$.

Figure 3 shows the effect of μ on the search efficiency for an authorized user. As the probability of publishing μ increases, PANACEA approaches the search efficiency of a structured system. It is interesting to observe that for only 0.05% of authorized copies of the resource in the system, setting a small value of $\mu = 0.6$ makes the search efficiency of PANACEA close to that of structured systems. Hence, we claim that $\mu = 0.6$ is the *appropriate* value for this parameter for the given network and configuration. On the other hand, for unauthorized users, increasing μ makes the search efficiency of the proposed system close to that of unstructured P2P systems, as shown in Figure 4. Therefore, PANACEA design meets its privacy objectives of Section III. As μ increases, so is the number of RPP index entries, which reduces the probability of deducing the existence of the resource as the list of potential providers grows.   In Figures 5 and 6, we highlight the effect of
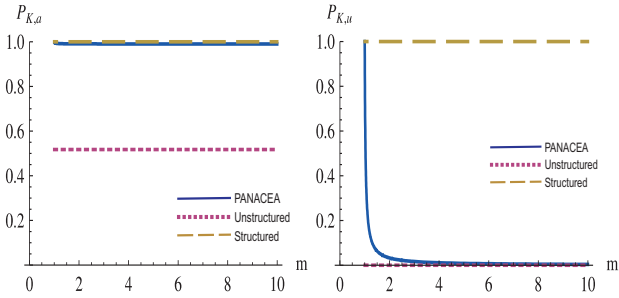

Fig. 5. $P_{K,a}$ vs m
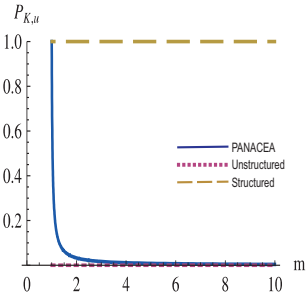

Fig. 6. $P_{K,u}$ vs m

m on resource privacy for authorized and unauthorized users respectively. Clearly, the value of m has no effect on the search efficiency of authorized users for a given value of μ, as they discover the resource as long as at least one provider with an authorized copy of the resource publishes in the DHT (as the provider is then directly contacted). However, for unauthorized users, as m increases, their probability to deduce the existence of the resource decreases due to the increased number of fake resource keys in the index entry, which is shown in Figure
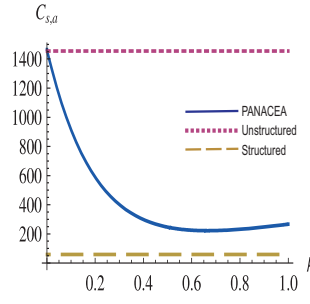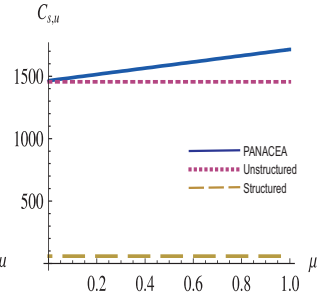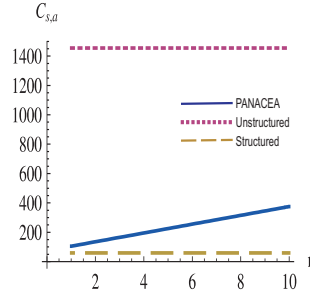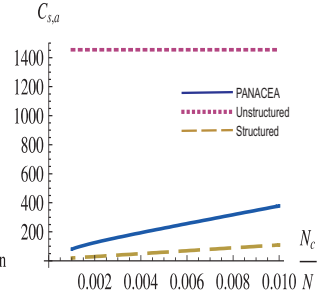

Fig. 9. $C_{s,a}$ vs n

Fig. 10. $C_{s,a}$ vs $\frac{N_c}{N}$

6. Thus, with increase in m, PANACEA search efficiency reaches that of unstructured P2P systems for unauthorized users. Similar results were obtained for the provider privacy of authorized and unauthorized users, while varying μ and m. We omit these results for brevity reasons.

Figure 7 depicts the effect of μ on the search communication cost for authorized users. As μ increases, the probability to find a provider where the user is authorized also increases. Hence, the user finds the resource in the DHT, and thus need not employ flooding. After $\mu = 0.6$, there is no more cost improvement. The almost constant difference between the PANACEA and structured systems costs is due to: i) the decreasing probability that the resource is not found in the DHT and then flooding has to be employed, and ii) the increasing number of providers that have to be contacted as m grows and more index entries are published in the DHT (refer to (12)). However, it should be observed from Figure 8 that increase in μ proves to be more *costly* for unauthorized users, which is a highly desirable property of PANACEA. The search cost with respect to n is depicted in Figure 9. As n increases, the search cost increases as the user has to contact more number of providers. However, observe that, even for $n = 10$, the search cost of PANACEA is several times lower compared to that of unstructured systems.

Clearly, Figures 3 to 9 prove our initial claims on PANACEA positioning as related to the structured and unstructured P2P systems as illustrated in Figure 1. In Figure 10, as the number of providers of the resource N_c increases in the overlay, we observe a linear increase in the search cost of PANACEA. However, this cost remains significantly lower than that of the unstructured P2P systems, and it can

be considered as highly tolerable given the privacy benefits of PANACEA. If, for certain application, this is undesirable in terms of scalability, it could be addressed by properly adjusting values of the parameters μ and n. The search cost can also be improved having a user contacting the providers listed in an RPP index sequentially instead of concurrently.

Finally, as explained in Section II-B, probabilistic publishing was introduced to hide the non-existence of resources. As shown in equations (8) and (14), the PANACEA system meets its design objectives (Section III) in this case as well, as $P_- \sim P_-^U$. Moreover, as shown in equations (15), $C_{s,-} \sim C_{s,-}^U$. We omit the relevant graphs for brevity reasons.

VI. Related Work

There is significant research work in the literature related to PANACEA, particularly in the areas of access control in P2P systems, privacy of access-controlled content, and anonymous P2P systems.

To enable access control in P2P systems, PHera [7] proposes a fine-grained access control framework based on super-peer-based P2P overlays where sub-peers specify their access control policies, which are enforced by the super-peers on behalf of the former. Super peers index the data of sub-peers and they could preserve data privacy by not replying to the queries from unauthorized peers. However, this approach assumes that all super-peers are unanimously trusted by their sub-peers to enforce their data privacy and access control policies, which is difficult in general [8].

Regarding the privacy of access-controlled content, a privacy-preserving approach for centralized indexing of such data is proposed in [8]. A group of data providers iteratively circulate a bloom filter representing the content hosted on the providers, bits of which are set probabilistically by the proposed algorithm. At the end of this iterative process, the index -represented by the bloom filter- emerges, which preserves data privacy regarding its location (i.e. provider privacy). However, as opposed to PANACEA, [8] does not address resource privacy. Furthermore, new resources can be easily inserted in the index of PANACEA, while index reconstruction is required in [8].

The OneSwarm system proposed in [9] preserves the privacy of a peer's location using cryptographic mechanisms. It employs an unstructured friend-to-friend overlay for privacy preserving content sharing. The system allows users to define permissions for data sharing among trusted friends. Peers search for data objects using flooding techniques, similar to access-controlled unstructured systems analyzed in this work.

There exists a large number of works in the area of anonymous P2P systems that achieve publisher (source) or reader (searcher) anonymity or both [1], [10], [6], [11]. Additionally, the anonymity of a node hosting an index entry (resource) is also considered [10]. In Freenet system [1], resource identifiers are generated in several cryptographic ways and are inserted into the system based on these identifiers. It achieves access control and resource and provider privacies using cryptographic techniques, which however, involves complicated key

distribution and management overhead. Furthermore, resource discovery is not guaranteed and involve significant search communication overhead compared to structured systems. In addition, the searchers have to be associated with the providers *a priori*, in order to be informed about the cryptographic keys. Instead, in our approach, search efficiency is high and new searchers can be dynamically authorized by providers to access the resources. P2P access control system based on such cryptographic indexing was discussed in [2].

A hybrid P2P system was discussed in [11], which involves structured and unstructured topologies to achieve sender and receiver anonymity. By connecting unstructured sub-overlays via a DHT, sender and receiver identities remain hidden.

Finally, note that searcher anonymity was not among the design objectives of PANACEA, but it could be easily achieved in a similar way the anonymous publication was realized in Section II-B.

VII. Conclusion

In this paper, we have proposed PANACEA, a P2P infrastructure to share access-controlled data, which combines high resource and provider privacies with high search efficiency for authorized users. We have analytically derived the privacy and search efficiency properties of the system and numerically evaluated to show that it meets its design objectives. As a future work, we intend to study the system behavior in case of repetitive attacks against system privacy and derive lower bounds on its effectiveness.

References

[1] I. Clarke, O. Sandberg, B. Wiley, and T. W. Hong, "Freenet: A distributed anonymous information storage and retrieval system," *Lecture Notes in Computer Science*, vol. 2009, pp. 46–67, 2001.

[2] N. Rammohan, Z. Miklos, and K. Aberer, "Towards access control aware p2p data management systems," in *2nd International workshop on data management in peer-to-peer systems*, 2009.

[3] I. Stoica, R. Morris, D. Karger, F. Kaashoek, and H. Balakrishnan, "Chord: A scalable Peer-To-Peer lookup service for internet applications," in *Proceedings of the 2001 ACM SIGCOMM Conference*.

[4] P. Maymounkov and D. Mazières, "Kademlia: A peer-to-peer information system based on the xor metric," in *Proc. of 1st International Workshop on Peer-to-peer Systems (IPTPS)*, Cambridge, MA, USA, March 2002.

[5] "Gnutella," *http://www.gnutella.com*.

[6] M. K. Reiter and A. D. Rubin, "Crowds: anonymity for web transactions," *ACM Trans. Inf. Syst. Secur.*, vol. 1, no. 1, pp. 66–92, 1998.

[7] B. Crispo, S. Sivasubramanian, P. Mazzoleni, and E. Bertino, "P-hera: Scalable fine-grained access control for p2p infrastructures," in *ICPADS '05: Proceedings of the 11th International Conference on Parallel and Distributed Systems (ICPADS'05)*. Washington, DC, USA: IEEE Computer Society, 2005, pp. 585–591.

[8] M. Bawa, R. J. Bayardo, Jr, R. Agrawal, and J. Vaidya, "Privacy-preserving indexing of documents on the network," *The VLDB Journal*, vol. 18, no. 4, pp. 837–856, 2009.

[9] T. Isdal, M. Piatek, A. Krishnamurthy, and T. Anderson, "Privacy-preserving p2p data sharing with oneswarm," *Technical report, University of Washington*, 2009.

[10] R. Dingledine, M. J. Freedman, and D. Molnar, "The free haven project: distributed anonymous storage service," in *International workshop on Designing privacy enhancing technologies*, 2001.

[11] A. Singh, B. Gedik, and L. Ling, "Agyaat: Mutual anonymity over structured p2p networks," *Emerald Internet Research Journal*, vol. 16, no. 2, 2006.