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 both probabilistic and information-theoretic approaches, we analytically derive that PANACEA can offer high privacy, while preserving high search efficiency for authorized users. Our analysis and the effectiveness of the approach have been experimentally verified. Moreover, the privacy offered by the proposed system can be tuned according to the specific application needs, which is illustrated with a detailed simulation study.**

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 node 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 Kademlia [3] etc., 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. 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 in our case) of each queried resource, hence, data privacy is breached. We define the former privacy aspect concerning resource existence/non-existence as *resource privacy*, while we refer to the latter one concerning resource location as *provider privacy*. On the other extreme, unstructured P2P systems, such as Gnutella (gnutella.com), 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, when 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. We carefully quantify privacy offered by PANACEA, employing both probabilistic modeling and information-theoretic approaches. We also analytically study 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 and verifying them with simulation experiments, 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; on the other hand, 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 [4]. A resource privacy of 0 refers to the case that the adversary cannot deduce the existence/nonexistence of a resource. 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 specification of authorization policy and

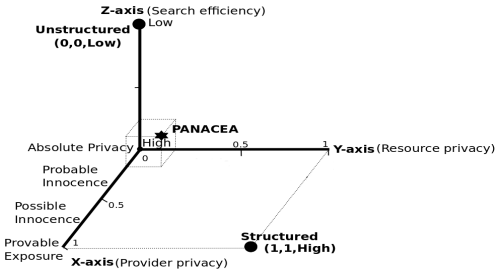

Fig. 1. Position of various systems on privacy and search efficiency axes.

the user credentials is orthogonal to the 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.

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 employing a probabilistic approach. We also quantify the resource and provider entropies of the system. In Section IV, we verify our analysis and present simulation experiments that demonstrate the effectiveness and the tunability of the system. In Section V, we discuss the related work, and finally we conclude in Section VI.

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 concerns hiding the *existence* and the *non-existence* of resources: an unauthorized user should not be able to determine either of them, regardless of the fact that there can exist multiple instances (henceforth referred to as *copies*) of the same resource owned by different peers. Our system aims to combine look up efficiency of structured P2P systems with high resource and provider privacies offered by unstructured ones. PANACEA employs a DHT that hosts a resource and provider privacy-preserving (RPP) index. 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, similarly to the unstructured P2P systems. As a result, PANACEA acts partly as a structured P2P system and partly as an unstructured one.

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. This tuning of the privacy parameters determines the position of the resulting system in the graph of Figure 1, as compared to structured and unstructured P2P systems. We describe the publishing and search mechanisms of PANACEA in Section II-A and in Section II-B respectively.

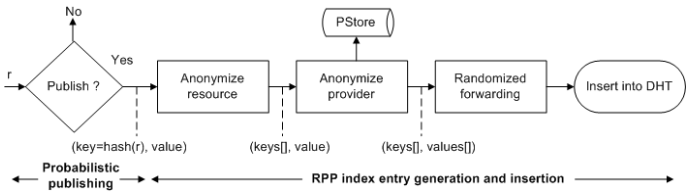

Fig. 2. Privacy preserving publishing in PANACEA

A. Privacy Preserving Publishing

PANACEA achieves the resource and provider privacy goals with an innovative privacy-aware publishing mechanism, which involves:

1) Probabilistic publishing of resources
2) RPP index generation
3) Randomized forwarding
4) Insertion into the DHT

The approach is illustrated in Figure 2 and is described as follows:

1) Probabilistic publishing: Instead of publishing every resource into the DHT, as in structured P2P systems, PANACEA announces a resource with a system-defined probability μ (as shown in Figure 2) and creates an RPP-index entry as described later. Therefore, absence of an index entry for a specific resource key in the DHT does not necessarily mean non-existence of the corresponding resource in the system. Due to probabilistic publishing, PANACEA acts as a hybrid semi-structured P2P system. All the resources, which are not announced in the DHT, are discovered using limited-hop flooding with a system-defined time-to-live (TTL).

2) RPP index generation: We employ *k-anonymization* techniques [5] 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 an adversary cannot identify it. Specifically, instead of having a $(key, value)$ pair as an index entry for a resource, as in structured P2P systems, we propose that the index entry consists of a list of keys and a list of values, i.e. $(key[], value[])$, which is derived by applying *resource* and *provider anonymization* that are subsequently explained. We refer to such an index entry as (m, n)-*index entry*, where m refers to cardinality of the key list and n refers to that of the value list. In this terminology, an index entry of the conventional structured P2P systems can be seen as a $(1, 1)$-index entry.

Regarding resource anonymization, once a new resource is chosen to be published, its corresponding $(1, 1)$-index is converted to an $(m, 1)$-index by adding $m - 1$ resource keys randomly selected (that may correspond to *genuine* or *phantom* resources) from a resource namespace R to the key list of the index entry. The resource namespace R can be domain-specific or span multiple domains and even contain words and their combinations from a dictionary. Note that human-readable plain text keys (i.e resource names), which are employed by the users to refer to the resources, are mapped by a hash

function to the system key space (i.e. resource ids). For a resource namespace R, the equivalent resource key space as K, and the hash function as $H : R \rightarrow K$. When a resource with name r is selected for publishing, its key $H(r)$ in fact, is selected for anonymization. When random keys are selected for resource anonymization, $m-1$ number of entries are randomly chosen from the set R and hashed to the set K. An adversary that is able to observe the resource keys of the (m, n)-entry may be able to derive the corresponding resource names of the $m-1$ keys by employing a dictionary attack. However, he will not be able to deduce whether these keys correspond to genuine resources or not.

After resource anonymization, the resulting $(m, 1)$-index entry is fed to the provider anonymizer module, as depicted in Figure 2. 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 randomly chosen from the *Provider Store (PStore)* - a local database of provider ids. We assume that PStores at each peer are initialized with a number of well-known peers and its neighbors in the overlay, and then incrementally expanded over time with unknown providers contained in the (m, n)-entries traversing through the peer.

3) Randomized forwarding: After an (m, n)-index entry is constructed by a publishing peer, it has to be inserted into the P2P system using the DHT $put()$ method. However, this index entry must be published *anonymously*, as the next-hop node in the DHT routing could easily deduce that the initiator node is itself the publisher from the (m, n)-entry where it is contained. In order to anonymize the node that initiates the insertion request, we propose that a *randomized forwarding* phase (see Figure 2) precedes the DHT $put()$ operation. Specifically, each peer that receives the insertion request decides with a system-defined probability λ to *forward* it to a node randomly selected from the n providers in the (m, n)-entry or *initiate* the DHT routing with the $put(m, n)$ method with probability $1-\lambda$. The technique of randomized forwarding to anonymize the original sender, was originally proposed in Crowds [4] for anonymizing web access. However, in [4], the next-hop node was randomly selected from the full set of cooperating nodes before contacting the web server. Clearly, our case is more complicated than the Crowds one, since the (m, n)-entry contains the publisher itself. Hence, by randomly choosing the next hop from the set of providers in the (m, n)-entry, we achieve equal probability for each of them for being the publisher. Note that randomized forwarding precedes DHT routing, and hence it does not demand any modifications to it. The randomized forwarding phase introduces additional communication overhead of $\lambda/1-\lambda$ number of hops [4]. We assume that PStore caches the IP address along with each provider id and that the IP address for each provider is stored in the provider list of the (m, n)-index entry. Thus, the relaying of a $put()$ message can happen in $O(1)$.

4) Insertion into the DHT: Finally, for the insertion of the (m, n)-entry into the DHT, $put(m, n)$ operation is invoked, which is implemented as follows. Note that the conventional $put()$ method inserts only a $(1, 1)$-index entry. Yet, the same method can be used to insert a $(1, n)$ entry, as the $value$ field is not used in the DHT routing. Hence, we propose to convert the $put(m, n)$ request into m number of $put(1, n)$ requests, using each of the m keys as pivot ones.

Note that, since the keys in an index entry are chosen independently by peers, key collisions (i.e. conflicts) 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 (m, n)-index entry is simply appended to the list of already existing providers for the collided key. A resource r with multiple copies is expected to have a larger provider list than that of a resource published by only one provider. However, as long as a genuine resource has smaller or equal provider list size to the maximum one n_f of a phantom resource, an adversary cannot differentiate between them. We propose an extension to our basic approach that increases the provider list sizes of phantom keys. Specifically, a peer randomly selects a small partition of the set R denoted as R_L ($R_L \subset R$) and constantly employs R_L for the resource anonymization instead of R (referred to as R_L*-approach*).

B. Searching

When a peer searches for a resource with key r, it executes *get(r)*. If an (m, n)-entry was published in the DHT having r as one of its m ids, then the peer returns the provider list of this entry to the searcher. Subsequently, the searcher contacts all these providers. Note that, in general, a user does not know in advance to which providers he is authorized to for the resource r, unless he has contacted them in the past for the same resource. In the latter case, the searcher could select only certain nodes from the provider list to contact. Once an index entry is found, a provider can be reached in $O(1)$ (as in [6]). However, in case of multiple providers for the 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 flooded over the overlay can be very low for resources with a few copies and with proper selection of the publishing probability μ, as shown in Section III. No provider responds to search queries from unauthorized users, in order not to compromise the resource and provider privacies.

III. Analysis

In this section, we analytically study the privacy offered by PANACEA by employing probability theory and information theory approaches. Moreover, we estimate the expected communication overhead of our approach.

We use the following notation in our analysis. Let N be the number of peers in the system and N_c be the expected number of copies of a genuine resource r and $N_a \leq N_c$ be the number of copies that a particular user is authorized to

access. We call a user as *unauthorized* to r, when he is not authorized to access any of the N_c copies of r, i.e. $N_a = 0$.

A. Probabilistic approach

We evaluate the privacy breach that can be achieved by a user who has *complete* knowledge on the parameters of our PANACEA system and queries the system for a particular resource. We denote:

i) $P_{K,a}$ (resp. $P_{K,u}$) as the probability for an authorized (resp. unauthorized) user to deduce the existence of a certain genuine resource.
ii) $P_{V,a}$ (resp. $P_{V,u}$) as the probability for an authorized (resp. unauthorized) user to deduce the provider of a certain genuine resource. For brevity, we assume the simple case where the user knows that the resource is genuine. Computation of $P_{V,a}$ for the more general case is presented in [7].
iii) $P_{\_}$ as the probability for an authorized or unauthorized user to deduce the non-existence of a certain *non-existing* resource.

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

i) Lower probability for an unauthorized user to deduce a resource's presence and its provider ($P_{K,u}$, $P_{V,u}$)
ii) Lower probability for a user deducing a resource's non-existence ($P_{\_}$)

Under this definition of privacy, any *privacy-efficient* access control mechanism should aim to:

- Minimize $P_{K,u}$, $P_{V,u}$, $P_{\_}$, which should ideally be 0 as in unstructured P2P systems.
- Maximize search cost $C_{s,u}$ for unauthorized users and ideally close to that of the unstructured P2P systems.

However, the *search efficiency* of the privacy-enabling mechanism should remain high, i.e.:

- $P_{K,a}$, $P_{V,a}$ should ideally be 1 (as in structured P2P systems), and
- The search communication cost $C_{s,a}$ should be kept low and ideally close to that of the structured P2P systems.

We express the privacy and search cost metrics of PANACEA in terms of the corresponding metrics of structured and unstructured P2P systems. To this end, we use superscripts U and S to denote metrics for unstructured and structured P2P systems respectively, and no superscript for PANACEA, e.g. $P^U_{K,u}$ refers to unstructured systems and the equivalent metric for PANACEA is $P_{K,u}$.

First, we quantify provider privacy for an authorized user. There are three cases that can arise:

Case (i): If any of the N_a copies, where he is authorized to, was published to the DHT, he could deduce the provider of the resource with probability 1. The probability that at least one of N_a copies was published into the DHT is $1-(1-\mu)^{N_a}$.

Case (ii): On the other hand, consider the case that none of the N_a copies was published into the DHT (probability of which, is $(1-\mu)^{N_a}$), but at least one of the remaining $N_c - N_a$ copies was published (probability of which is $(1-(1-\mu)^{N_c-N_a})$). In this case, the user first contacts all the providers associated with $H(r)$ and then floods the search request, where he deduces the provider with probability $P^U_{V,a}$. In case of unsuccessful flooding, the user tries to deduce the provider from the provider list present in the DHT. If l is the provider list size, then the number of genuine providers for resource r can be inferred as $\frac{l}{n}$ as this number of (m,n)-entries resulted in a provider list of size l. Hence, in this case, we claim that the provider is deduced with probability $\frac{l/n}{l}$. However, the provider list size need not grow in multiples of n for each insertion, because of "collisions" (i.e. provider id conflicts) in the provider lists across the multiple (m,n)-entries of the resource copies. We account for this effect with a *collision probability* f_v. Note that collisions in the resource keys are also possible with probability f_k, which increase the expected provider list size of a key. We refrain from formalizing f_v, f_k analytically for brevity reasons.

Case (iii): When a DHT entry is not found for the resource key (probability of which is $(1-\mu)^{N_c}$)[1], the user attempts to deduce the provider by flooding ($P^U_{V,a}$). Hence,

$$P_{V,a} = \left[1-(1-\mu)^{N_a}\right]\cdot 1 + (1-\mu)^{N_a}\cdot(1-(1-\mu)^{N_c-N_a})\cdot \left[P^U_{V,a}\cdot 1 + (1-P^U_{V,a})\cdot\frac{1}{n(1-f_v)(1+f_k)}\right] + (1-\mu)^{N_c}\cdot P^U_{V,a}\,. \tag{1}$$

We apply similar reasoning to formulate resource privacy $P_{K,a}$ for an authorized user in addition to the following. Let the maximum provider list size n_f of a phantom resource key be known to the user. Then, if the size of provider list of the (m,n)-entry of the queried resource is larger than n_f, the user can deduce the existence of the resource in the system with probability 1. Otherwise, a probability $\frac{1}{m}$ can be assigned to the existence, as the queried key is mixed with $m-1$ other ones in the (m,n)-entry, in addition to what can also be deduced by flooding ($P^U_{K,a}$). Therefore,

$$P_{K,a} = \left[1-(1-\mu)^{N_a}\right]\cdot 1 + (1-\mu)^{N_a}\cdot(1-(1-\mu)^{N_c-N_a})\cdot \mathbf{I}(E(n)) + (1-\mu)^{N_c}\cdot P^U_{K,a}, \text{ where}$$
$$\mathbf{I}(l) = \begin{cases} 1, & \text{for } l > n_f \\ P^U_{K,a}\cdot 1 + (1-P^U_{K,a})\cdot\frac{1}{m}, & \text{otherwise} \end{cases} \tag{2}$$

$E(n) = \mu(N_c - N_a)n(1-f_v)(1+f_k)$ is the expected provider list size for the queried resource.

Next, we quantify the search cost $C_{s,a}$ in terms of the number of nodes visited by the search query from an authorized user. First, a user searches in the DHT which incurs a cost of $C^S_{s,a}$. Thereafter, we account for two possible cases- none of N_c copies or some i copies of the resource are published into the DHT. The former case can happen with probability $(1-\mu)^{N_c}$ where the user employs flooding, incurring a cost of $C^U_{s,a}$. In the latter case, $i\cdot n$ number of providers are contacted.

[1]In fact, a DHT entry can also be present for the key being a phantom one. However, we assume this probability as negligible, as $|R|$ is big compared to the number of genuine resources.

If none of them has an authorized copy (probability of which, is $(1-\frac{N_a}{N_c})^i$), the user employs flooding. Overall, $C_{s,a}$:

$$C_{s,a} = C^S_{s,a} + (1-\mu)^{N_c} \cdot C^U_{s,a} + \sum_{i=1}^{N_c} \binom{N_c}{i} \mu^i (1-\mu)^{(N_c-i)} \cdot \left[i \cdot n \cdot (1-f_v)(1+f_k) + \left(1 - \frac{N_a}{N_c}\right)^i C^U_{s,a} \right] \tag{3}$$

Equations for $P_{K,u}$, $P_{V,u}$, and $C_{s,u}$ can be derived from eq. (1) to eq. (3) by having $N_a = 0$ and replacing the terms $P^U_{V,a}$, $P^U_{K,a}$, $C^S_{s,a}$, $C^U_{s,a}$ by $P^U_{V,u}$, $P^U_{K,u}$, $C^S_{s,u}$, $C^U_{s,u}$ respectively.

Finally, we derive P_{-}, i.e. the probability to deduce the non-existence of a non-existing resource. Given an event space $\Omega = \{\text{DHT}, \neg\text{DHT}\}$ that a non-existent resource is found or not in the DHT respectively, P_{-} is given by:

$$\begin{aligned} P_{-} = Pr(- \mid \Omega) =& Pr(- \mid \neg\text{DHT}) \cdot Pr(\neg\text{DHT}) + \\ & Pr(- \mid \text{DHT}) \cdot Pr(\text{DHT}) \text{ , where} \\ Pr(- \mid \neg\text{DHT}) =& P^U_{K,u} = 0 \\ Pr(- \mid \text{DHT}) =& \frac{m-1}{m} \\ Pr(\text{DHT}) =& \left[1 - \left(1 - \frac{1}{\mid R \mid} \right)^{\mu N_r (m-1)} \right] \\ Pr(\neg\text{DHT}) =& 1 - Pr(\text{DHT}) \end{aligned} \tag{4}$$

N_r is the total number of resources in the system and R is the resource namespace. $Pr(- \mid \neg\text{DHT})$ 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. $Pr(- \mid \text{DHT})$ is the probability that the resource corresponding to the key does not exist. $Pr(\text{DHT})$ expresses the probability that a phantom resource from namespace R may have been inserted into the index, while $Pr(\neg\text{DHT})$ is the complement of $Pr(\text{DHT})$. Observe that P_{-} is minimal (~ 0) for reasonable values of the various parameters. Also, we estimate the expected query cost to deduce the non-existence. The user first searches for an index entry and then employs flooding, hence,

$$C_{s,-} = C^S_{s,-} + C^U_{s,-} \, . \tag{5}$$

B. Information-theoretic approach

In [8], [9], an information theoretic approach was proposed to measure privacy offered by a system employing *entropy* H as an anonymity metric, which is defined as: $H = -\sum_i p_i \log_2 p_i$, where p_i is the attacker's estimate of the probability that a participant i was responsible for some observed action. Entropy is maximized to $\log_2 |A|$ if equal probability is assigned to all members of the anonymity set A, and it is minimized to 0 when $|A| = 1$. According to [8], a system with entropy H has *effective anonymity set* of size 2^H. As an adversary in PANACEA may have different information sets (i.e. each resulting from different observations), *conditional entropy* H_0 [9] is a more appropriate metric, which is given by:

$$H_0 = \sum_y Pr[Y = y] H(X|Y = y) = \sum_y E_y H(X|Y = y), \tag{6}$$

where X is a random variable of the private aspect to be preserved and Y models the different observations y. E_y denotes expectation with respect to observation y.

First, we calculate the entropy of PANACEA for provider anonymity against an authorized searcher. For brevity in the analysis, we assume that the searcher already knows the existence of the queried resource. More detailed analysis can be found in [7]. Here, the random variable X models the publisher of the requested resource and the random variable Y models possible information sets observed by the searcher: (i) an authorized copy is found in the DHT, (ii) an unauthorized copy is found in the DHT and an authorized copy is found by flooding, (iii) an unauthorized copy is found in the DHT but no authorized copy is found by flooding, (iv) no copy is found in the DHT but an authorized copy was found by flooding, and (v) no copy is found in the DHT and no authorized copy was found by flooding. If the searcher has made the observations (i), (ii) or (iv), then the provider entropy is 0. In case of observation (iii), where an unauthorized copy of the resource is found in the DHT and no authorized copy was found by flooding, we calculate $H(X|Y = iii)$ according to the following logic: $\mu(N_c - N_a)$ copies are expected to be published in the DHT resulting in a provider list of size $E(n)$. Therefore, the probability that a copy out of the $\mu(N_c - N_a)$ ones resides at one of these providers is $\frac{\mu(N_c-N_a)}{E(n)} = \frac{1}{n(1-f_v)(1+f_k)}$. Also, $(1-\mu)(N_c - N_a)$ copies are not published in the DHT and the probability that a copy resides at any other provider (i.e. apart from the $E(n)$ ones) is $\frac{(1-\mu)(N_c-N_a)}{N-E(n)}$. However, since there exist $N_c - N_a$ copies in total in this case, the sizes of the aforementioned sets of $E(n)$ and $N - E(n)$ number of providers are divided by $N_c - N_a$ to derive the effective anonymity set sizes. Finally, in the case of observation (v), where no copy is found in the DHT and no authorized copy was found by flooding, each peer in the set of N peers has a probability of $\frac{N_c}{N}$ for being a provider. Therefore, the provider entropy $H_{V,a}$, is given by:

$$\begin{aligned} H_{V,a} =& E_{\text{iii}} H(X|Y = \text{iii}) + E_{\text{v}} H(X|Y = \text{v}) \\ =& -(1-\mu)^{N_a}(1-(1-\mu)^{N_c-N_a})(1-P^U_{V,a}) \cdot \\ & \left[\frac{E(n)}{N_c - N_a} \frac{1}{n(1-f_v)(1+f_k)} \log\left(\frac{1}{n(1-f_v)(1+f_k)} \right) + \right. \\ & \left. \frac{N-E(n)}{N_c - N_a} \frac{(1-\mu)(N_c-N_a)}{N-E(n)} \log\left(\frac{(1-\mu)(N_c-N_a)}{N-E(n)} \right) \right] \\ & - (1-\mu)^{N_c}(1-P^U_{V,a}) \left(\frac{N}{N_c} \right) \left(\frac{N_c}{N} \right) \log\left(\frac{N_c}{N} \right) \end{aligned} \tag{7}$$

Next, we calculate the system entropy for resource anonymity. To this end, the random variable X models the existence of a resource, i.e. whether a resource name from the resource namespace R exists in the system or not. The random variable Y models the observations of the searcher for

a requested resource as in the case of the provider entropy. The analysis follows a similar reasoning to the case of the provider entropy. Note that in the information set (iii), if the expected size of the provider list of the (m, n)-entry is greater than n_f, then the resource is genuine and there is no anonymity. Also, $E(m) = \mu(N_c - N_a)m(1 - f_k)$ is the expected number of keys in the DHT when $N_c - N_a$ copies may be published. Overall, the resource entropy $H_{K,a}$ is given by:

$$\begin{aligned} H_{K,a} =& E_{\text{iii}} H(X|Y = \text{iii}) + E_{\text{v}} H(X|Y = \text{v}) \\ =& -(1-\mu)^{N_a}(1-(1-\mu)^{N_c-N_a})(1-P^U_{K,a}) \cdot \mathbf{I}'(E(n)) \\ & -(1-\mu)^{N_c}(1-P^U_{K,a}) \log\left(\frac{N_c}{|R|}\right), \text{ where} \end{aligned}$$

$$\mathbf{I}'(l) = \begin{cases} \log(1), \text{ for } l > n_f \\ \left[\frac{E(m)}{N_c-N_a}\frac{1}{m(1-f_k)}\log(\frac{1}{m(1-f_k)}) + \frac{|R|-E(m)}{N_c-N_a}\cdot \right. \\ \left. \frac{(1-\mu)(N_c-N_a)}{|R|-E(m)}\log\left(\frac{(1-\mu)(N_c-N_a)}{|R|-E(m)}\right)\right], \text{otherwise} \end{cases} \tag{8}$$

The equations for $H_{V,u}$ and $H_{K,u}$ can be derived from eq. (7) and (8), respectively by replacing $N_a = 0$ and $P^U_{K,a}$, $P^U_{V,a}$ with $P^U_{K,u}$, $P^U_{V,u}$ respectively.

IV. Evaluation

In this section, we verify our analysis and evaluate the privacy and search efficiencies of the PANACEA system as related to unstructured and structured access-controlled P2P systems using simulation experiments.

In our simulated system (implemented in Java), we assume $N = 10000$ peers that use the system both to publish and search for resources. The PStores on the peers are initialized with 25 random entries. The providers are organized in a Kademlia-like structured topology, but they are also connected over an unstructured overlay power-law network with average degree 7.5 and maximum degree 150. We conducted two types of simulation experiments, which differ in their resource distributions and the type of the generated queries. Each resource is randomly assigned a publisher peer and a list of user peers who are authorized to access the resource. Any other peer is said to be an unauthorized user for this resource and publisher pair. We compute $P_{V,a}$ and $P_{V,u}$ as follows:

- $P_{V,a} = 1$ if a user is able to contact a publisher where he is authorized to access the requested resource.
- If the resource key is found in the DHT, then $P_{V,a} = P_{V,u} = \frac{1}{n}$.
- Otherwise, if the resource key is not found in the DHT and no authorized copy is located by flooding, then $P_{V,a} = P_{V,u} = 0$.

The PANACEA's parameters are taken as follows: key list size $m = 5$, value list size $n = 5$ and forwarding probability $\lambda = 0.6$. Also, $ttl = 4$ was employed for limited-hop flooding in the unstructured overlay.

A. Provider privacy and search cost

Initially, we aim to verify the correctness of eqs. (1), (3) using the simulation results with a rather static setting regarding resource popularity. Specifically, we assume 100 resources with $N_c = 50$ copies for each (thus $5K$ resources in total). 100 peers are randomly selected, each of which is inserted into the authorization list of random $N_a = 5$ copies. Each resource is then given to randomly chosen peers that publish them using the PANACEA publishing mechanism. The collision probability for provider lists is experimentally found to be $f_v = 0.002$, while the collision probability for resource keys is assumed to be $f_k = 0$. We also experimentally found that by searching in the unstructured overlay $P^U_{V,a} = 0.38$, $P_{V,u} = P_{-} = 0$, a total of 912 distinct nodes are visited and 1687 messages are sent per query on the average. In order to measure $P_{K,a}, C_{s,a}$, we generate authorized searches from the above 100 authorized peers for all of the 100 resources, thus $10K$ search queries in total. Also, in order to measure $P_{K,u}, C_{s,u}$, we randomly select 100 unauthorized users that query the system for the same 100 resources. These experiments have been run 10 times each and the mean values are plotted in Figures 3 to 6. As depicted in Figures 3, 4, the analytical equations model the privacy properties of the simulated PANACEA system very accurately. As the probability of publishing μ increases, PANACEA approaches the search efficiency of a structured system (see Figure 3). Note that for only $N_a = 5$ authorized copies in the system, a small value of $\mu = 0.6$ makes the search efficiency of PANACEA close to that of structured systems. On the other hand, for unauthorized users, provider privacy of our system is always close to that of unstructured P2P systems, as shown in Figure 4. Therefore, PANACEA design meets its privacy objectives introduced in Section III. For $\mu = 0$, $P_{V,u} = 0$. From $\mu = 0.1$ onwards, a provider list is found in the DHT for the queried resource, resulting in a constant privacy value of $\frac{1}{n(1-f_v)(1+f_k)}$. This is captured in Figure 4. In Figure 5, the effect of μ on the

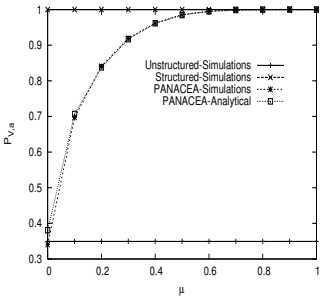

Fig. 3. $P_{V,a}$ vs μ

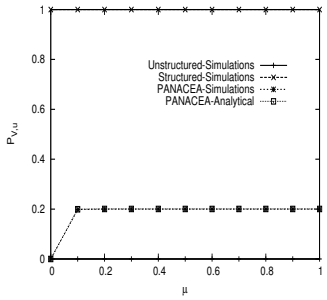

Fig. 4. $P_{V,u}$ vs μ

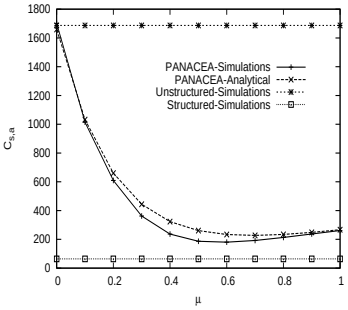

Fig. 5. $C_{s,a}$ vs μ

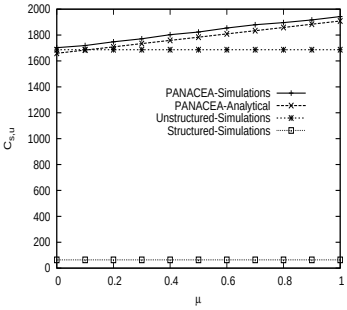

Fig. 6. $C_{s,u}$ vs μ

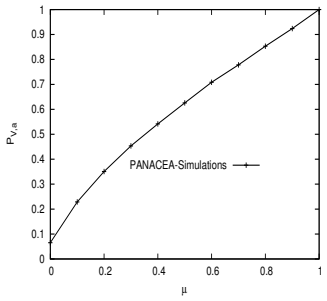

Fig. 7. $P_{V,a}$ vs μ

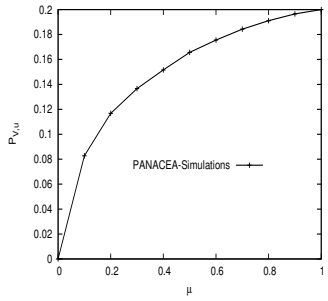

Fig. 8. $P_{V,u}$ vs μ

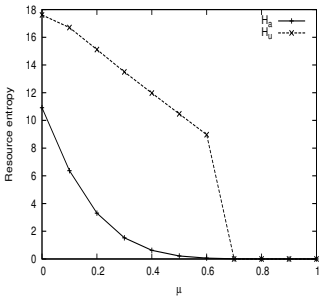

Fig. 11. H_K vs μ

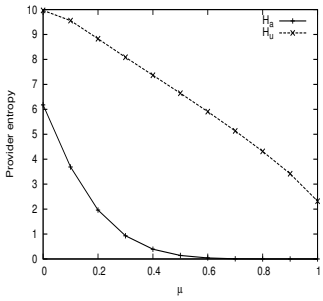

Fig. 12. H_V vs μ

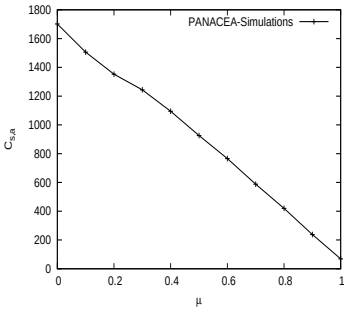

Fig. 9. $C_{s,a}$ vs μ

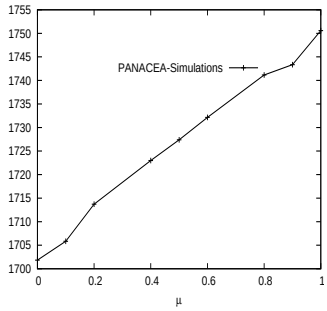

Fig. 10. $C_{s,u}$ vs μ

search communication cost for authorized users is depicted. As μ increases, the probability to find a provider where the user is authorized also increases. After $\mu = 0.6$, there is no more search cost improvement, because the size of the provider list of the queried resource slightly increases. As shown later, in general, the search cost significantly decreases as μ increases. However, as observed from Figure 6, the search cost for unauthorized users significantly increases (over the cost of limited-hop flooding) with μ, which is a highly desirable property of PANACEA.

Next, we evaluate the provider privacy and the search cost of PANACEA for authorized and unauthorized users in a more general setting, where $10K$ resources whose popularity (N_c) follows Zipf distribution are published in the system. The maximum number of resource copies is 150 and their mean number is 10 thus resulting in a total of $100K$ resources. A random number of 5 to 50 peers are chosen to be authorized to each resource. Each resource of the $100K$ ones is randomly assigned to a peer that initiates PANACEA publishing. We randomly generated $20K$ number of authorized and unauthorized search queries separately. Again, the experiments are repeated 10 times and mean values of the results are plotted in Figure 7 to Figure 10. As depicted in Figure 7, search efficiency increases with μ for an authorized user. The search efficiency can be increased by choosing high μ for unpopular resources. However, as shown in Figure 8, provider privacy is always minimal for unauthorized users. Also, Figure 9 depicts that the search cost for authorized users decreases with μ, as opposed to that of unauthorized users as shown in Figure 10.

B. Resource privacy

As mentioned in Section II-A, resource privacy may be breached by observing large sizes of the provider lists. Although it is difficult to preserve resource privacy of highly popular resources, we do not focus on them as their presence in the system can be easily taken for granted. Our goal for resource privacy in PANACEA is to preserve privacy for the other resources.

We observe that as long as an existing resource has a provider list size less or equal to that of a phantom resource, an adversary cannot differentiate between them. We assume again $100K$ resources Zipf-distributed with mean 10 and maximum 150 copies. For the R_L*-approach* (Section. II-A4) with $|R_L| = 25$, we observed that phantom keys have provider lists longer than those of the 87.6 percentile of the existing resources in the DHT (for $\mu = 1$). For this percentile of resources, the resource privacy for unauthorized users is $P_{K,u} = 1/m = 0.2$ and for the authorized users is $P_{K,a} = 1$ for $\mu = 1$. For more popular resources, the adversary can exploit the provider list sizes to conclude their existence.

Finally, by numerically evaluating eqs. (4) and (5) with $|R| = 2M$, $N_r = 100K$, $\mu = 1$, we observed the PANACEA system meets its design objectives in this case as well, as $P_- \sim P_-^U$ and $C_{s,-} \sim C_{s,-}^U$. We omit the verification of these formulas with simulations for brevity reasons.

C. System entropy

We demonstrate the entropy analysis discussed in Section III-B in Figures 11 and 12 with parameters $N_a = 5, N_c = 10, n_f = 30$. Note that this value for n_f corresponds to $f_k = 0.14$. For $\mu = 0$, the entropy $H_{K,u} \approx 18$ which is roughly equivalent to an anonymity set of size $\approx 200K (= \frac{|R|}{N_c})$. As μ increases, the entropy decreases due to the resource presence in the DHT. After $\mu = 0.6$ the provider list size in the DHT becomes greater than n_f, and from then on, an unauthorized user sees no entropy in the system. For authorized user, entropy gradually reduces with μ and becomes zero after $\mu = 0.6$. The provider entropy is demonstrated in Figure 12. $H_{V,u} \approx 10$ (=N_c/N) for $\mu = 0$. After $\mu = 0.6$, the provider entropy becomes zero for authorized user. At $\mu = 1$, for an unauthorized user the anonymity set size would be $n = 5$, which is verified by the plot as the entropy is approximately 2.3 as shown in Figure 12.

V. 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 [10] proposes a fine-grained access control framework based on super-peer-based P2P overlays where the access-control policies of sub-peers are enforced by the super-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 [6]. In PANACEA, peers can share their resources through index hosted on untrusted nodes, and yet, can enforce access control.

Regarding the privacy of access-controlled content, a privacy-preserving approach for centralized indexing of such data is proposed in [6]. 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, [6] does not address resource privacy. Furthermore, new resources can be easily inserted in the index of PANACEA, while index reconstruction is required in [6].

The OneSwarm system proposed in [11] 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, similarly to access-controlled unstructured systems.

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], [4], [12]. Additionally, the anonymity of a node hosting an index entry (resource) is also considered [13]. In Freenet [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, involve complicated key distribution and management overhead. Furthermore, resource discovery is not guaranteed and involves 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 used to encrypt the content accessible them. 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 involving structured and unstructured topologies to achieve sender and receiver anonymity, was discussed in [12] and referred to as Agyaat. Agyaat, provides mutual anonymity for the sender and receiver, which is not among the goals of PANACEA. Agyaat offers three alternative resource discovery approaches: semantic groups, centralized directory service, and dynamic services. In the first case, peers that host semantically similar resources are grouped into a cloud. Then, some sort of resource and provider privacies can be provided at the expense of resource discovery, which is flooding-based, as opposed to our approach. For improving resource discovery, a centralized directory service or dynamic services can be also employed. Then, a resource is mapped to a cloud and the index is stored at a central server or at the coordinator peers of the clouds in a distributed manner. These are similar to the privacy preserving indexing employed in [6]. However, the anonymous construction of this index is not presented and it does not analytically quantify its effectiveness, as opposed to PANACEA.

VI. 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 employing probabilistic and information-theoretic approaches. Our analysis was verified by simulation experiments, while we analytically and experimentally showed that PANACEA meets its design objectives. As a future work, we intend to employ the mechanism in a Kademlia client and observe the privacy offered in a real testbed.

Acknowledgment

This work was partly supported by the Swiss Nano-Tera OpenSense project (Nano-Tera ref. 839_401).

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] P. Maymounkov and D. Mazières, "Kademlia: A peer-to-peer information system based on the xor metric," in *Proc. of IPTPS*, 2002.

[4] 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.

[5] L. Sweeney, "k-anonymity: a model for protecting privacy," *Int. J. Uncertain. Fuzziness Knowl.-Based Syst.*, vol. 10, no. 5, 2002.

[6] 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.

[7] N. Rammohan, T. G. Papaioannou, and K. Aberer, "PANACEA: Tunable privacy for access controlled data in peer-to-peer systems," *Technical report, EPFL-REPORT-148337, EPFL*, 2010.

[8] A. Serjantov and G. Danezis, "Towards an information theoretic metric for anonymity," in *Proc. of PET*, San Francisco, CA, USA, April 2002.

[9] C. Diaz, S. Seys, J. Claessens, and B. Preneel, "Towards measuring anonymity," in *Proc. of PET*, San Francisco, CA, USA, April 2002.

[10] B. Crispo, S. Sivasubramanian, P. Mazzoleni, and E. Bertino, "P-hera: Scalable fine-grained access control for p2p infrastructures," in *Proc. of the ICPADS*, Washington, DC, USA, 2005.

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

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

[13] R. Dingledine, M. J. Freedman, and D. Molnar, "The free haven project: distributed anonymous storage service," in *Proc. of PET*, 2001.