Pythia: A Privacy Aware, Peer-to-Peer Network for Social Search

Shirin Nilizadeh, Naveed Alam, Nathaniel Husted, Apu Kapadia
School of Informatics and Computing
Indiana University Bloomington
Bloomington, IN 47401, USA
{shirnili, nalam, nhusted, kapadia}@indiana.edu

ABSTRACT

Emerging "live social search" systems such as Aardvark.com allow users to pose questions to their social network in real time. People can thus obtain answers from real humans for questions that prove too complex for web searches. Centralized systems that broker such queries and answers, however, do not provide adequate privacy. The success of these systems will be limited since users may avoid asking or answering questions related to sensitive topics such as health, political activism, or even innocuous questions which may make the querier seem ignorant.

Since social search systems leverage the structure of the social network to better match askers and answerers, standard ideas that hide this structure such as "connect to Aardvark via Tor" fall short. Thus new techniques are needed to preserve the privacy of askers and answerers beyond the currently understood anonymity techniques. We explore the new and unique challenges for privacy, and propose Pythia, a decentralized architecture based on "controlled flooding" to enable privacy-enhanced social search that retains some degree of social network structure.

Categories and Subject Descriptors

C.2.4 [**Computer-Communication Networks**]: Distributed Systems—Distributed Applications; K.6.m [**Management of Computing and Information Systems**]: Miscellaneous—Security

General Terms

Algorithms, Security

Keywords

privacy, social search, peer-to-peer, question-and-answer systems

1. INTRODUCTION

We are all experts at something. At some level we may be e.g. movie critics, food connoisseurs, or photographers, and are happy to share our expertise with those who inquire. Often we too are faced with complicated questions that are too difficult to "google". Imagine a world where any person with a question is connected to the right (and available) person who can immediately answer that question. Such is the promise of an emerging class of what we call "live social search" systems.

Social search systems such as Aardvark (`http://www.vark.com/`) [11] leverage the power of social networks to connect askers with *online* experts (i.e., humans with domain expertise) who are close to the asker in the social network, thus facilitating *live* exchanges of information between real humans.[1] Horowitz and Kamvar [11] draw the distinction between the *library* model of search (e.g., web searches), where askers search for the *right document* to answer a question vs. the *village* model (e.g., Aardvark), where askers seek to get connected with *right human* because it is unlikely their complicated question can be satisfied by an existing document. In fact, Google recently acquired Aardvark, noting on their blog "sometimes the information just is not online in one simple place."[2] While we have used Aardvark as an exemplar of such a system, there is considerable interest in this model of live social search. Facebook Questions is currently being rolled out to its users, and provides a social search service.

Centralized systems such as Aardvark and Facebook, unfortunately, do not provide adequate privacy to users because they maintain full knowledge about the social network. For example, the identities of askers and experts, participants' interests and expertise areas, and their communication are all known to such systems. While these systems support general queries related to restaurant recommendations, home improvement, and product recommendations, the inability to ask or field queries privately limits the scope of queries and advertised expertise. For example, a pro-choice or pro-life advocate may want to field queries about the topic but not want her colleagues, or anybody (including the social search service), to know of her leanings. Similarly, experts may want to engage in political activism and keep their involvement secret. Askers, too, may ask questions on these topics and others such as health related or legal issues, only under the cover of privacy. In some cases, users may simply be embarrassed to ask "simple" questions with the fear of being perceived as ignorant. Even if the social search service kept identities private with respect to communicating askers and experts, the social service itself has knowledge of all this information. All this information is vulnerable to abuse, subpoenas, secondary use, unauthorized data aggregation, and is also a prime target for data breaches (a central point of failure). Connecting to such services over anonymizing networks such as Tor [9] to hide the identities of askers and answerers break the

WPES'11, October 17, 2011, Chicago, Illinois, USA.

[1] Aardvark claims "The vast majority of questions are answered within 10 minutes."

[2] `http://googleblog.blogspot.com/2010/02/google-acquires-aardvark.html`

utility of the system because the structure of the social network is used to match askers and answers based on their proximity in the social network. Furthermore, centralized services can manipulate or censor queries and answers and restrict communication between users in the system, anonymous or not. Thus existing routing-layer anonymity solutions do not suffice.

If we are to realize the true potential of live social search for a range of topics, we need to develop **decentralized solutions**. Furthermore, we need to rethink the design of current anonymity systems to expose the structure of the social network just enough to make relevant matches between askers and answers. Within the context of such systems, we need to understand what the new privacy requirements are. Traditional notions of *sender or receiver anonymity*, for example, which aim to keep the identify of the sender or receiver secret do not capture properties such as *expertise unlinkability* for answerers or *interest unlinkability* for askers. For example, even if individual messages cannot be tied to Alice, one may be able to determine Alice is a pro-choice expert. Expertise areas may be leaked based on how such systems are structured (are similar experts connected to each other?), or how expertise is advertised in the system in order to attract queries. Thus providing expertise/topic unlinkability goes beyond anonymity of messages.

Towards building such systems we characterize the system model and security goals for decentralized live social search systems and propose Pythia. The central idea in Pythia is to partition the social network into *communities* or *flood zones* and use *local flooding* to send questions to online experts within the community. Such flooding provides a high degree of privacy within the community (as we explain in Section 3), yet limits the amount of flooding to maintain scalability (see our analysis in Section 3.2). When no nearby experts are found, we argue that beyond 2 or 3 hops in the social network it probably doesn't matter where the expert is located in the network, and any remote community with online experts can be contacted using *remote flooding* within the distant community. We show that for a low constant amount of overhead, Pythia supports private queries (with anonymity relating to the cluster size) while maintaining the quality of responses when nearby experts are available. We conduct a preliminary evaluation of privacy under a specific adversarial model and elaborate on future directions, demonstrating that privacy-enabled live social search is a rich area for further research. An extended version of this paper appears as a technical report [17].

2. SYSTEM MODEL AND SECURITY GOALS

We describe the high-level system model we assume for P2P social search systems, our security goals and the adversary model.

2.1 System model

We assume that a user is connected to and has knowledge of his/her list of friends in the social network (e.g., by bootstrapping off established social networks such as Facebook or instant messaging systems). Every user has a list of self-declared *expertise areas*, i.e., topics for which he/she can answer questions. For example, Alice's set of expertise areas could be {recipes, military intelligence analysis, computer networks, environmental activism}. Some of these expertise areas are *private expertise areas*, and are not known to other users in the network. In such cases the user prefers to not be known as an expert in that area.

Nodes (users) may be *online* or *offline* at any given time. Their online and idle status is visible to their friends, as with instant messaging applications. Offline nodes cannot assist in routing messages. While idle nodes can route messages, only *available* (not idle) nodes are capable of answering questions.

A peer with a query can attach a set of *query tags* to the query, where the tags indicate topics related to the query. Based on these query tags, the *query routing* protocol attempts to find experts for the specified tags close to the asker in the social network. Once the query is routed to an available expert, the expert can respond. We assume that not all available experts will answer questions, and whether they do will depend on how *responsive* the expert is.

2.2 Security goals

The following three properties are unique to P2P live social search systems:

Expertise unlinkability: A peer's private expertise areas should not be attributable to her identity beyond a certain threshold probability. For example, Bob may be a pro-choice advocate who doesn't want to advertise his association widely with the pro-choice movement. We assume a threshold that is sufficient to provide *plausible deniability*. For example, some authors consider a probability of 0.5 to be sufficient [20]. We assume more conservative probabilities on the order of 0.1 to be sufficient for plausible deniability. However, in cases where the prior probability is higher than 0.1, we say plausible deniability is attained if the threshold is below the prior probability. For example, if 33% of nodes in a community are experts on a particular topic, then if the attacker cannot infer Alice is an expert in that topic with probability more than 0.33, we say plausible deniability is attained.

Interest unlinkability: An asker's private query tags should not be attributable to her identity beyond a certain probability. For example, Alice may want to ask several queries about terrorist organizations she hears about on the news but may worry about being labeled a terrorist. We assume the same probability threshold as discussed for expertise unlinkability.

Unobservable querying and responding: Anonymous (but observable) queries and responses may allow an attacker to observe that a particular node is asking or answering a question, allowing the attacker to narrow down the set of possible nodes related to a particular answer or question. We seek to prevent this attack and hide whether nodes are asking questions or providing answers at all. Note that although nodes in the system may observe queries and answers being exchanged, they do not know whether individual nodes are issuing queries or answers.

Sender anonymity: One important note is that we assume that the primary goal of an attacker is to uncover the expertise or interest area of a user. For such a requirement sender anonymity is necessary but not sufficient. Clearly if sender anonymity is broken for a message, the expertise area of the sender is also broken. Sender anonymity, however, does not imply expertise unlinkability. As already mentioned, expertise areas may be leaked based on how such systems are structured (are similar experts connected to each other in clusters?), or how expertise is advertised in the system in order to attract queries. Thus, providing expertise or topic unlinkability goes beyond message anonymity, and the system must be carefully designed to preserve querier/expertise unlinkability in addition to sender anonymity. As a result our proposed system Pythia uses underlying sender anonymity techniques as just one building block to provide the requisite privacy.

2.3 Attack model

We evaluate two classes of adversaries: *global attackers* can view all messages exchanged in the system and infer the online/offline and idle status of all the nodes at any time. *Colluding attackers* have only partial knowledge of this type—we assume some

fraction of nodes c ($0 < c < 1$) are compromised and these colluding attackers can infer the online and idle status of their neighbors only, and view messages exchanged with their neighbors.

Attackers can have two different capabilities related to what they can infer about messages: the messages are *linkable* if adversaries can tell the answers (or questions) are authored by the same answerer (or asker). For example, perhaps the writing style is unique enough to link answers by the same expert. Otherwise, the messages are *unlinkable*. Using and linking observed information, attackers try to determine the asker of a particular topic or the answerers of a particular topic. We assume all adversaries have full knowledge of the structure of the social network, and are *honest but curious*, i.e., they participate in the protocol correctly, but try to infer what they can based on their observations.

Thus we have four adversaries: *Global-Linkable*, *Global-Unlinkable*, *Colluding-Linkable*, and *Colluding-Unlinkable*.

3. ARCHITECTURE

Figure 1 shows the high-level architecture of Pythia. The central idea in Pythia is to partition the *nodes* in the social network into *anonymizing communities* or *flood zones*. Questions from a community are received by all nodes in the community by means of a *local flood*. The local flood allows anonymous answerers to receive questions without having to reveal their expertise areas. Furthermore, to provide asker/answerer unobservability, all nodes ask and answer questions at regular intervals (including dummy questions and answers) and thus attackers cannot readily pinpoint which nodes are forwarding, asking or answering questions. If no answerers are found in the local community, questions are forwarded to a remote community as part of a *remote flood*.

Communities are small enough to limit the overhead of flooding as well as to target answerers who are close to the asker in the social network, but large enough to provide plausible deniability as explained in Section 2.2. Our work does not seek to provide near-complete anonymity (i.e., where the answerers can be any of a several million nodes in the network), but attempts to strike a good trade-off between privacy and performance. The size of communities, however, is a system parameter, and represents a trade-off between privacy and performance. Smaller communities provide lower privacy, but ensure that the overhead of query flooding is a low constant (as opposed to a full flood of the entire network, which has overhead linear in the size of the social network for example).

In Pythia, time is divided into intervals $\{t_1, ..., t_n\}$, where questions and answers are exchanged and distributed at the end of each time interval as coordinated by a *representative* who is reelected periodically. As we will show later, longer time intervals provide better privacy against attackers, but delay communication times. For practical purposes, one can assume time slots are a few minutes long. We note the representative can be adversarial (and pose as one of the four types of attackers outlined in Section 2.3).

3.1 Creating social communities

In Pythia, all nodes in the social network are grouped into self-organizing clusters called *communities*. We assume a distributed community-forming algorithm such as the one proposed by Ramaswamy et al. [19], which results in a *representative* for that community. In this scheme, nodes are clustered based on social relationships and each node belongs to a single community and all community members are known to each other. Communities may have more than one representative, although, for simplicity, we assume that communities have only one representative. Since it is not the focus of our paper, we refer the reader to our technical report for more details on the community creation protocol [17].

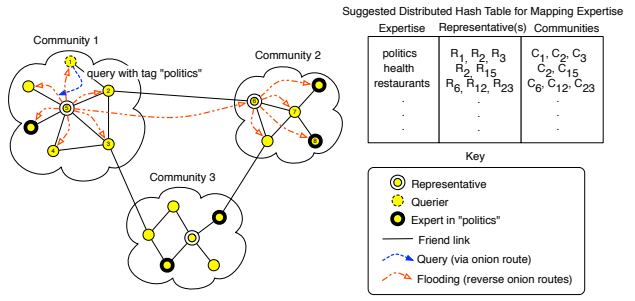

Figure 1: Node 1 in Community 1 initiates a query with the tag *politics*. The query via an onion route is sent to the *representative* node 5. The representative then floods the query to the local community (along with other received queries). The answerer in the community is unresponsive. The representative, having received no responses, may choose a random community or ask the DHT for communities with answerers in politics, and initiates a remote flood in Community 2 by contacting representative node 6. Node 8 is a responsive answerer whose response is sent to representative node 6 via an onion route (like a query message), then relayed to representative 5, and finally received by node 1 in the next flood by node 5.

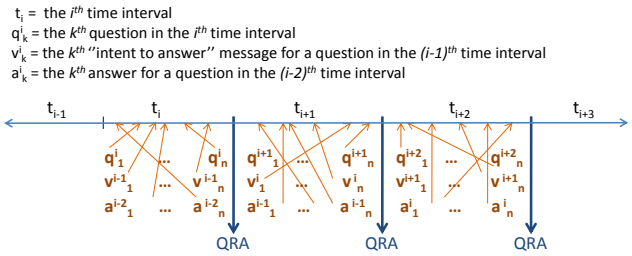

Figure 2: Timeline in Pythia

3.2 Routing questions and answerers

To simulate a distributed social search, nodes in Pythia participate in three phases: asking, showing their intent to answer, and answering. In all phases the same protocol is used with different message types. All the messages in a community are forwarded to the representative of the community, and are padded to have the same size to thwart traffic analysis of messages in transit.

As a building block, Pythia uses Onion routing [10] to deliver messages from nodes to representatives. We assume that the list of IP addresses in a community is available to all nodes after community creation. The sender of a message can pick a set of n random nodes from the community and progressively build a circuit through these nodes. We conservatively set $n = 6$, which provides adequate "mixing time" in social networks, such that from the receiver's point of view the message could have originated anywhere in the community. The particular choice for this parameter is orthogonal to this work, and it suffices to pick a value that provides adequate mixing. To create an "onion packet", the message is encrypted with the public key of the representative and each node in the circuit. As a node receives a message it decrypts the outer layer and passes it to the next node in the circuit. This approach prevents attackers in collusion with the representative to link a particular message with a particular sender, providing asker/answerer anonymity unless all nodes along the path are compromised. As with mix networks [7, 16] we assume nodes along the route add

delays and reorder messages to thwart timing attacks. Since messages are exchanged periodically, tens of seconds of delay at each hop should be sufficient. Note that to initiate a two-way conversation, an asker generates a "reply onion" [3, 5] to send the question. The reply onion is transmitted to the representative who is the last node of the chain. Then representative uses the reply onion to initiate the return chain for sending answers.

To obtain the public keys of nodes for setting up onion routes, Pythia can benefit from existing centralized or distributed key management approaches applied in peer-to-peer systems [8, 12, 14, 25]. However, to avoid revealing any information about the circuit, the sender should be given a list of public keys of all the nodes within the community instead of the nodes in the circuit. An approach could be using a DHT to store several replicas of the communities' lists of public keys, and to obtain and use keys based on agreement. The nodes on the DHT responsible for storing the lists would verify if the authorized IP address is updating or retrieving the list. To defend against global passive adversaries an extra step is needed to provide asker/answerer unobservability. As illustrated in Figure 2, during every time period t_i, *every* online node sends three separate messages to the representative, along three different routes, at randomly chosen times in the interval. Due to the time delays imposed by nodes along the onion route, messages are received by the representative at random times in the interval.[3]

For every time period, a user can ask up to one question, and sends q^i in time period t_i. Each online user also sends one "intent to answer" message v^{i-1} for a question received in t_{i-1}. If questions with topic tags for which there exists no expertise in the local community are found, then they are forwarded to a remote community. In Pythia the remote community is chosen randomly. However, this selection can be done more efficiently and purposefully if users can advertise (see Section 5) their topics of interests anonymously and a DHT mechanism provides the correlations between communities and topics so that questions can be directed to appropriate communities with potential answerers in that area.

Although many experts may be volunteered to answer a particular question so as not to overburden experts who are willing to answer only a couple of times a day, representatives pick at most α answerers (at random) to answer the question (e.g., we set $\alpha = 2$ in our simulations), and thus at most α selected answerers from the set of answerers who sent volunteer messages for the question are informed about this selection in the previous time period t_{i-1}. Answerers that have been picked to answer, find their one-time pseudonym in the respond block.

Online nodes send one answer a^{i-2} for the question they selected in t_{i-2} and askers must wait between 2 to 3 time periods to receive answers. For all these messages, if the online node does not have a legitimate question, intent to answer, or answer, then the node sends a dummy message through the onion routing. Thus, attackers cannot directly observe who is asking or answering questions. Moreover, encrypted messages have fixed sizes so the adversary cannot infer the content from the message size. We note that the representative selects the α answers after discarding the dummy messages. We note that reordering the messages within each block is not necessary. We assume the representative is adversarial and thus all adversaries know the order in which messages were received by the representative. Pythia relies on onion routing to provide anonymity to the sender of a message.

[3]The route length and average delays at each hop should be tuned so that messages are likely to be received by the representative within the time period.

3.3 Messaging overheads

Our controlled flooding approach does not induce high overheads. At each time interval, the traffic generated in each community can be analyzed by calculating the messages sent by each member and the representative. Based on the format of messages and Pythia's protocol, even with a large community size of 10,000, every time period only 17MB of traffic is received by the representative and it sends out only 7MB to the community. With smaller community sizes of hundreds to a thousand nodes, the traffic is on the order of a few hundred kilobytes to a megabyte *total* per time period. Thus the overhead of dummy messages is low and the representative is not overwhelmed with traffic. We provide details of this analysis in our technical report [17].

4. EVALUATION

Pythia uses two-way onion routing with mixing and dummy traffic to provide unobservability of question asking and answering, sender unobservability and resistance against traffic analysis attacks, especially for the global adversaries who can observe all traffic in the system. Moreover, the use of two-way onion routing with mixing ensures that even a malicious representative cannot actively modify questions and answers to great benefit because the representative does not know who the recipients of the messages are. While this architecture also provides resistance against attacks to interest/expertise unlinkability, attackers can attempt to correlate information about who is online and not idle to deduce the expertise of a victim. We now evaluate our system under such attacks.

To demonstrate our hypothesis that privacy and utility can indeed be balanced for privacy-aware social search, we simulated a P2P system of 60,000 nodes partitioned into social communities of around size 100. To test Pythia with various potential topologies of a social network, we created 5 randomly generated scale-free graphs with 60,000 nodes using the Network Work Bench[4] (NWB) tool and the Barabasi-Albert (BA) model. Each graph was used for 5 different simulation experiments. Communities were created using the process discussed in Section 3.1. A subset of communities was taken due to the computational complexity of the simulation. The subset communities were sized between 85 nodes and 115 nodes ($\mu = 95.526, \sigma = 8.6702$).

Human models from Skype and Aardvark usage were used to simulate queries and answers. We defined three types of expertise categories: Common, Uncommon and Rare where 70%, 30% and 6% of nodes respectively have expertise in these topics. Each user can ask questions on 36 different topics by tagging the query with a topic. In our system users are assigned different numbers of expertise topics according to the rough distribution of users and topics in Aardvark as described by Horowitz and Kamvar [11]. In our simulation, every node issues on an average 2 queries per week while 20% of users are *active* [11]. If an active answerer gets a query, she responds with an 85% probability. We assumed users are all from a country with 3 time zones and are online for 10 hours.[5] We make a simpler assumption in this paper that the user is online for 10 contiguous hours (but we vary the idle times randomly), and in the future we plan to use a better model for online/offline behavior. We provide details of usage models in our technical report [17].

As one example depicted in Figure 3, we studied the degradation of anonymity after four weeks of system operation under the *Global-Linkable* model defined in Section 2.3. Anonymity sets for experts were measured depending on how many questions they an-

[4]`http://nwb.slis.indiana.edu/`

[5]According to [18, 21], in Skype 95% of peers disappear after 10 hours of activity.

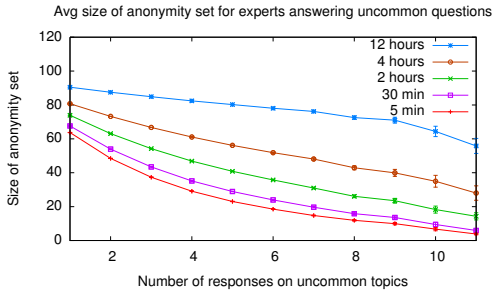

Figure 3: Anonymity set vs. number of questions answered for Uncommon topics against a Global-Linkable attacker after 4 weeks for various time aggregations.

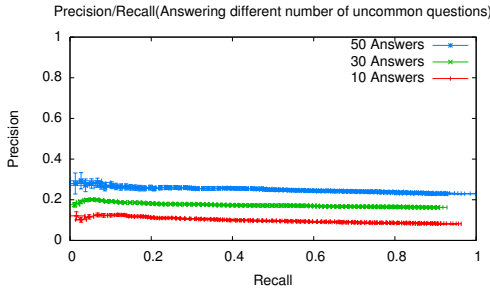

Figure 4: Precision vs. Recall for Uncommon topics against a Global-Unlinkable attacker after 4 weeks.

swered in the four-week period. Adversaries were able to observe the online/offline/idle status of all users in the network and correlate questions and answers with those observations. As a private expert answers more and more questions, adversaries are able to narrow down the set of potential users that may be that expert. We note that while our analysis focuses on questions answered in a single category only, one could interpret these results for questions answered in multiple categories (if answers from the same person are linkable) since the attack would be the same. We can see that for various levels of aggregation (time scales at which questions and answers are exchanged in the network) the anonymity set degrades faster for lower aggregation times and slower for higher aggregation times. We posit this adversary is too strong, since we assume the adversary can link responses from the same expert together. Further research is needed into determining how effective stylometry is (i.e., linking texts to a particular author, or recognizing a set as coming from the same author), and how to defeat it [4].

Figure 4 shows that experts have a high degree of anonymity against *Global-Unlinkable* adversaries. In this attack, adversaries maintain presence counts of users while answers are received for sensitive topics. After 4 weeks, adversaries sort the list of answerers by descending order of counts, and draw a line after each answerer in the list with the hope that a large number of answerers are included in the list above the line. Figure 4 plots the precision vs. recall curves for the adversaries, where each point corresponds to a particular position of the line in the sorted list of counts. The flat precision shows this attack is not successful in finding the answerers at top of the list and answerers are uniformly distributed in the list. Even after 4 weeks, the precision is at best 0.1 for rare topics, which provides plausible deniability.

Finally, in our simulations, we found the average social distance between askers to local answerers was about 3 as a result of the clustering algorithm. We hope future work can further reduce this distance based on better clustering.

Implications of our results. Thus, if we assume adversaries can link responses from experts by noticing similarities in text, the anonymity of users degrades over time (albeit less so for Colluding-Linkable adversaries). *This is a fundamental limitation of anonymity systems that cannot control the content of messages being exchanged.* On the other hand our results are promising by showing that if answers are not linkable, anonymity improves greatly. Users must therefore ensure that their messages do not contain revealing characteristics [4].

5. DISCUSSION

Shielding. We propose the use of *shielding sets*, where participants would recognize the set of users that are usually online while they are online, and only ask or answer questions when a large fraction of these users are online. Participants can thus keep track of their anonymity sets, and compute their loss of anonymity when they ask or answer questions. Such a study needs long term data about the online/offline patterns of users.

Metrics. For expertise unlinkability, it would be better to characterize the probability of a person being an expert using the prior probability to then calculate the posterior probability of being an expert based on the size of the anonymity set. This information could be combined with other information relating to the estimate of how many experts are suspected to exist within the anonymity set. We leave such refinements to future work.

Advertising. When nodes join the network, they may advertise their expertise using a DHT based approach. A representative can then select remote communities more efficiently and purposefully and direct questions to communities with potential answerers for a topic. However, a sophisticated method is needed to resist linkage attacks where joins/leaves of nodes in communities cannot be easily correlated with entries added/removed from the DHT.

Reputation. Nodes could advertise their expertise along with reputation information for that expertise. For example, users could accumulate anonymous digital cash [2] for answering questions, and then prove their "wealth" as an indicator of reputation. A detailed reputation mechanism is outside the scope of this paper and we leave details for such a scheme to future work.

Incentives. Our current model largely relies on altruism. In addition to reputation mechanisms, simple policies can help control freeriding; e.g., the number of questions that users can ask may depend on how recently the user has joined or on the number of questions that a user has answered. Incentive mechanisms could take into account the quality of answers provided, which could then be combined with a reputation system.

6. RELATED WORK

Torrey et al. [23] study how information is searched for and learned in the web. Adamic et al. [1] studied knowledge sharing and its relation to Yahoo Answers. Popular Q&A services include Quora, Yahoo! Answers, Amazon Askville, Wiki-answers, and Google Groups. These services allow users to post their questions and answer other questions. None of these are *live* social search systems as they offer only offline communication, and the service does not actively seek experts. Some social search applications use an individual's social network to filter out the most relevant search results. For example, Cha-Cha sets up a "human middleman" to maximize the number of relevant results returned to the user. Such services demonstrate the power of humans to filter

out inconsequential data during searches, but still fall within the library model of searching for documents.

Cutillo et al. [6] describe a peer-to-peer architecture implementation for social networks. Li et al. [15] study the feasibility of P2P as a web search engine infrastructure. These systems however do not support live social search. Wu et al. present a P2P based distributed search system called Sixearch.org [24]. However, Sixearch locates static content. Related to the flooding-style of routing in Pythia, P5 [22] is a protocol for scalable anonymous communication over the Internet. Although it provides anonymous communication, it lacks specific features required by Q&A networks. Last, and most related to our work, Kacimi et al. [13] present a protocol that allows anonymous opinion exchange among users connected over an untrusted social network platform. However, it does not protect against honest-but-curious nodes within the P2P network and thus has weaker protections for privacy.

7. CONCLUSIONS

We present *Pythia*, a privacy-aware P2P system for live social search. We have made the first significant attempt at designing such a distributed system with strong privacy guarantees, and show the feasibility of our approach through extensive simulations. While this work provides an important first step, we hope to spur further research in areas such as privacy-aware query routing, defenses against intersection attacks, incentivizing use of such systems for sensitive queries, and assigning reputation to anonymous experts. We believe social search is bound to succeed through services such as Aardvark and Facebook Questions in this social networking age. Yet much work remains to be done to support private queries about sensitive issues. Without privacy-aware systems, the full potential for social search will not be realized.

8. ACKNOWLEDGMENTS

This research was funded in part by a grant from the Indiana University Center for Applied Cybersecurity Research. We thank Johan Bollen, Filippo Menczer, Katy Börner, Russell Duhon, Chris Schneider, and our anonymous reviewers for their comments.

9. REFERENCES

[1] L. Adamic, J. Zhang, E. Bakshy, and M. Ackerman. Knowledge sharing and yahoo answers: everyone knows something. In *Proceeding of the 17th international conference on World Wide Web*, pages 665–674. ACM, 2008.

[2] M. H. Au, S. S. M. Chow, and W. Susilo. Short e-cash. In S. Maitra, C. E. V. Madhavan, and R. Venkatesan, editors, *INDOCRYPT*, volume 3797 of *Lecture Notes in Computer Science*, pages 332–346. Springer, 2005.

[3] C. Beaver, R. Schroeppel, and L. Snyder. A design for anonymous, authenticated information sharing. In *Proceedings of the 2001 IEEE, Workshop on Information Assurance and Security*, 2001.

[4] M. Brennan and R. Greenstadt. Practical attacks against authorship recognition techniques. *Proceedings of the Twenty-First Innovative Applications of Artificial Intelligence Conference*, 2009.

[5] D. L. Chaum. Untraceable electronic mail, return addresses, and digital pseudonyms. *Commun. ACM*, 24(2):84–90, 1981.

[6] L. A. Cutillo, R. Molva, and T. Strufe. Privacy preserving social networking through decentralization. In *WONS'09: Proceedings of the Sixth international conference on Wireless On-Demand Network Systems and Services*, pages 133–140, Piscataway, NJ, USA, 2009. IEEE Press.

[7] G. Danezis, R. Dingledine, and N. Mathewson. Mixminion: Design of a type III anonymous remailer protocol. *Security and Privacy, IEEE Symposium on*, 0:2, 2003.

[8] M. E. Dick and E. Pacitti. Leveraging p2p overlays for large-scale and highly robust content distribution and search. In *VLDB PhD Workshop*, 2009.

[9] R. Dingledine, N. Mathewson, and P. Syverson. Tor: The second-generation onion router. In *Proceedings of the 13th USENIX Security Symposium*, August 2004.

[10] D. Goldschlag, M. Reed, and P. Syverson. Onion routing for anonymous and private internet connections. *Communications of the ACM*, 42:39–41, 1999.

[11] D. Horowitz and S. D. Kamvar. The anatomy of a large-scale social search engine. In *WWW '10: Proceedings of the 19th international conference on World wide web*, 2010.

[12] T. Isdal, M. Piatek, A. Krishnamurthy, and T. Anderson. Privacy-preserving p2p data sharing with OneSwarm. *SIGCOMM Comput. Commun. Rev.*, 40, 2010.

[13] M. Kacimi, S. Ortolani, and B. Crispo. Anonymous opinion exchange over untrusted social networks. In *SNS '09: Proceedings of the Second ACM EuroSys Workshop on Social Network Systems*, pages 26–32, 2009.

[14] M. Kondo, S. Saito, K. Ishiguro, H. Tanaka, and H. Matsuo. Bifrost : A novel anonymous communication system with dht. *Parallel and Distributed Computing Applications and Technologies, International Conference on*, 0, 2009.

[15] J. Li and F. Dabek. F2F: Reliable storage in open networks. In *5th IPTPS*. Citeseer, 2006.

[16] U. Möller, L. Cottrell, P. Palfrader, and L. Sassaman. Mixmaster protocol — version 3, Dec. 2004.

[17] S. Nilizadeh, N. Alam, N. Husted, and A. Kapadia. Pythia: A Privacy Aware, Peer-to-Peer Network for Social Search. Technical Report TR687, Indiana University Bloomington, Oct. 2010.

[18] J. A. Pouwelse, P. Garbacki, J. Yang, A. Bakker, J. Yang, A. Iosup, D. Epema, M. Reinders, M. V. Steen, and H. J. Sips. Tribler: A social-based peer-to-peer system. In *In The 5th International Workshop on Peer-to-Peer Systems*, 2006.

[19] L. Ramaswamy, B. Gedik, and L. Liu. A distributed approach to node clustering in decentralized peer-to-peer networks. *IEEE Transactions on Parallel and Distributed Systems*, 16:814–829, 2005.

[20] M. K. Reiter and A. D. Rubin. Crowds: anonymity for web transactions. *ACM Trans. Inf. Syst. Secur.*, 1(1):66–92, 1998.

[21] D. Rossi, M. Mellia, and M. Meo. Understanding Skype signaling. *Comput. Netw.*, 53(2):130–140, 2009.

[22] R. Sherwood, B. Bhattacharjee, and A. Srinivasan. P5: a protocol for scalable anonymous communication. *J. Comput. Secur.*, 13(6):839–876, 2005.

[23] C. Torrey, E. F. Churchill, and D. W. McDonald. Learning how: the search for craft knowledge on the internet. In *CHI '09: Proceedings of the 27th international conference on Human factors in computing systems*, pages 1371–1380, New York, NY, USA, 2009. ACM.

[24] L.-S. Wu, R. Akavipat, and F. Menczer. Adaptive query routing in peer web search. In *WWW '05: Special interest tracks and posters of the 14th international conference on World Wide Web*, pages 1074–1075, 2005.

[25] Z. Xu and H. Jiang. A framework of decentralized pki key management based on dynamic trust. In *Proceedings of the International Conference on Security & Management*, 2008.