Simple Peer-to-Peer SIP Privacy

Joakim Koskela[1] and Sasu Tarkoma[2]

[1] Helsinki Institute for Information Technology, P.O. Box 9800,
FI-02015 TKK, Finland
joakim.koskela@hiit.fi

[2] Computer Science and Engineering Department,
Helsinki University of Technology
sasu.tarkoma@hut.fi

Abstract. In this paper, we introduce a model for enhancing privacy in peer-to-peer communication systems. The model is based on data obfuscation, preventing intermediate nodes from tracking calls, while still utilizing the shared resources of the peer network. This increases security when moving between untrusted, limited and ad-hoc networks, when the user is forced to rely on peer-to-peer schemes. The model is evaluated using a Host Identity Protocol-based prototype on mobile devices, and is found to provide good privacy, especially when combined with a source address hiding scheme. The contribution of this paper is to present the model and results obtained from its use, including usability considerations.

Keywords: peer-to-peer, privacy, P2PSIP, HIP.

1 Introduction

Privacy in peer-to-peer (P2P) systems has been an active topic for a number of years, having touched upon a wide range of aspects of this area of computing. Preventing identities or actions from being disclosed to others is difficult in P2P systems, which per-definition rely on the close cooperation between nodes.

Much of the related work has concentrated on hiding the real identities of users [1] [2] [3], as this has been the most pressing issue partly due to the legally questionable use of many P2P content-sharing systems. As long as the users cannot be traced, there is not a need to hide the actions (content requests) unless the content itself reveals something important.

Internet telephony, such as Skype, has traditionally relied on a system of trusted, centralized, servers for authentication and setting up connections. The privacy of the users, with respect to the system operator, is non-existent. Although phony accounts, source address hiding and voice scrambling can be used to protect end-users, the operator has all the means to track calls made through the system. This may not be a concern for most consumers, since they can be assumed to trust the operator; however, for companies and governments this raises more fundamental issues. For example, many companies block Skype, because call routing is proprietary and there is no way to guarantee privacy and confidentiality of the calls.

A.U. Schmidt and S. Lian (Eds.): MobiSec 2009, LNICST 17, pp. 226–237, 2009.

As we move toward mobile and ad-hoc environments, we need fully distributed systems that operate without the help of centralized nodes. This has resulted in initiatives such as the IETF's P2PSIP working group [4], aimed at standardizing a fully distributed session signaling protocol. In these systems, the control is shared amongst the peers, with the implication that anyone that is part of the network can track the calls made through it. As the operations themselves (such as call setup) implicate the users involved, simply hiding the source of these requests does little. Indeed, a more systematic privacy mechanism is needed.

Although several proposals have addressed this problem domain in related work, implementation results have not typically been elaborated. In this paper, we examine a simple application-level model for enhancing privacy in fully distributed communication systems. First, we review the details of the threat in current systems. We follow this by presenting our solution, discussing its benefits, tradeoffs and possible alterations. As the model imposes restrictions on the accessibility of users, we continue by discussing issues related to the usability of the solution. Finally we present our implementation and results obtained from its use.

The new contribution of this paper is to present and evaluate, through a prototype implementation, a concrete model for enhancing privacy in P2P communication systems. Also, we highlight and analyze issues concerning the usability of the model and present our solution for these. We hope that this paper would foster discussions of how privacy is perceived in P2P environments, possibly leading to new opportunities for further research.

2 Problem Scope

Peer-to-peer systems form a network between the participating nodes, used to collectively perform tasks and manage resources. In P2P communication systems, this network is used to perform the duties of a service provider. Most importantly to establish the voice or other calls, but also for managing presence, contact lists or for other services. A common model is to use the peer network as a distributed storage and message routing overlay, where the information needed to contact users is stored, scattered throughout the nodes of the network.

This is also the approach taken by the IETF P2PSIP working group [4], arguably the most prominent attempt at creating an open standard for these systems. The name is derived from peer-to-peer Session Initiation Protocol (SIP), as the original intent was to create a P2P version of the successful multimedia session signaling protocol.

The current version of the P2PSIP protocol draft defines a highly modular framework supporting different applications (called *Usage*s) as well as overlay network types [5]. The network module of this framework offers message routing, key-based storage and connectivity services through a common interface, independent of the underlying network structure. Although there are few restrictions, the underlying network is assumed to be a distributed hash table (DHT)-like structured network, with efficient key-based routing, where the storage service scatters the data throughout the network.

SIP-based applications use this framework for three primary operations, referred to as the *SIP Usage*; registration, lookup and connection establishment. Registering a SIP identity (SIP Address of Record, SIP AOR) with the overlay is done by creating a data packet containing the SIP AOR and a NodeId, the identifiers used for nodes in the overlay, or another SIP AOR through which the users can be reached. This packet is stored in the network under a key made from the hash of the SIP AOR.

The session signaling is exchanged directly between peers. To establishing this connection, peers use the *Attach* function provided by the network module. This initiates an Interactive Connectivity Establishment (ICE) [6] procedure, during which *address candidates* are gathered, and sent to the other peer through the overlay network. These address candidates are all Internet protocol (IP) address and transport-level port combinations through which the peer might be reachable (due to multihoming, network address translation or firewalls). These three operations offer a number of opportunities for curious, or malicious, users to eavesdrop.

Assuming a DHT-like overlay (such as Chord [7]), where responsibility for data is assigned using a key proximity function, any node with a suitable NodeId, along with every node in-between, is able to intercept registrations. Even if these nodes were trustworthy, it is trivial to request the information using the public SIP AOR.

To establishing a SIP session, the caller fetches the registration packet for the responder, and connects using the Attach function. Again, the node maintaining the registration packet, and all nodes in-between, can easily monitor from whom the user receives calls. And even though that path is secured, Attach-related messaging offers yet another opportunity to track the call.

As noted in the draft [5], end-to-end encryption of the payload could be used to mitigate some of these, although still leaving the storage keys and NodeIds exposed. Also, a strong authentication mechanism and a protective identity acquirement procedure would prevent peers from situating themselves suitably in the network, decreasing the possibility for eavesdropping and sabotage in large networks [8].

The problem at hand is to introduce privacy mechanisms to P2PSIP that guarantee end-users that their sessions are not intercepted and tapped by possible malicious peers. The threat is anticipated by the P2PSIP Internet Draft; however, to our understanding this paper presents the first analysis of P2PSIP privacy issues based on experimentation.

3 Solution Model

In the design of the privacy-enhancement for P2PSIP, we identified the overlay data storage as a crucial component. The aim was to design a simple model which sets no additional requirements on the underlying storage service, using only the get, put and remove primitives. This way the model can be adopted to other, similar systems as well. Besides, this does not prevent the use of additional privacy-related enhancements, such as source address hiding, for even greater security.

We did make the assumption of a strong, cryptography-based, identity scheme. The identities are associated with the public key of a key pair, with the private key kept secret by the rightful owner. This is a reasonable assumption, as it reflects the current trend in these systems and is a common solution for identity management in distributed systems in general. It allows users to protect the integrity of their data and make it accessible only for specific peers using encryption. How these keys are generated or bound to the identities, e.g., using a trusted third party certificates or leap-of-faith, is not relevant for the privacy scheme.

This leads us to the first step of our solution, which is to encrypt the registration packet using the public key of peers that may want to establish a session. To protect the integrity, the packet is signed prior to encryption, preventing either the data, source or intended recipients from being revealed without the proper decryption key. We thus *publish* this data only for a specific set of trusted peers. As each peer uses a different key pair, we need to encrypt and publish the package multiple times or use a common encryption key appended as a set of attachments, each encrypted using a different public key.

3.1 Storage Key Obfuscation

As the keys used for storing registration packets map directly to the public identities, encrypting the content of these packets alone does not prevent intermediate peers from tracking the calls. Size analysis can be used identify registration packets and by monitoring the keys used to store and request these, we can determine both who is on-line as well as the recipient of a call. Although buffering and *decoy* packets can be used to make it harder, it seems unavoidable that the storage keys should also be obfuscated.

Until now, we have assumed that users need to possess only the public keys of the trusted peers to publish registration packets. We could consider a scheme were the storage key is also encrypted using these. However, this is easily broken in systems were the public keys are well known, which is our assumption. An attacker could not make a general query whether a specific user is on-line, but being aware of even one peer which that user trusts, be able to query the state of the user with regard to that peer.

We could also consider a scheme where all users publish their registration packets (encrypted) for a specific peer using a common storage key. Either as such, or under a secondary, random, key with a *link* stored under the common key. As the key for this *index* is shared, intermediate nodes can not determine whom the packets concern. This protects the publishers, but reveals information about the user for whom they are intended, such as the number of friends and call intentions. Although a minor threat, mitigated using decoys, the scheme is inefficient and easily sabotaged, as the caller needs to process a large number of packets before finding the right one.

The most practical solution we found is to simply use shared secrets to obfuscate the storage keys. After encrypting the registration packet, the storage key is formed by appending a shared secret to the SIP AOR, and computing a hash digest (using algorithms such as MD5 [9] or SHA-1 [10]) of the result. As the hash function can not be reversed, the new key will not reveal the source or recipient of the packet.

This concludes the core of our model. Currently the static nature of the shared secrets is seen as the weakest link, although this can be mitigated using techniques such as auto-renegotiation of the shared secret, hash chains [11] or time-bound appendices. The use of these is left open, to be chosen suitably to best fit the nature of the application.

Although our model does well in protecting the privacy of users from intermediate nodes, it has a number of obvious tradeoffs. It significantly increases the amount of data stored in the overlay, as the registration packets need to be published separately for each peer. However, as the packets contain only the information necessary to contact the user, they are relatively small. We could also consider a slightly less secure linking-scheme, similar to the one described previously. The packets stored using the shared secrets would contain only an encryption key and a link and pointing to the actual registration packet, encrypted with that key.

Our model requires also that peers perform a large number of public-key cryptography operations. However, the frequency of these should be low, keeping the load well within the range of even resource-limited computing devices.

3.2 Usability Considerations

Besides the technical complications, the scheme introduces a number of usability issues that need consideration. The need for a privacy enhancing scheme might be unclear for many users accustomed to traditional communication systems operated by a trusted provider. Also, how such a scheme affects the usability of the system, how certain contacts might become unreachable as a result, might not be apparent.

When activated, we need to have both the public key and a shared secret with everyone that might wish to contact us. Otherwise we will appear unavailable, *offline*. This might seem as an extension of a presence scheme, allowing us to lay hidden from certain unwanted contacts.

The purpose is not to limit who is able to contact you, but to hide who *does*. This privacy scheme sits a level below presence, on top of which visibility can be filtered. Accordingly, although shared secrets is often associated with close relationships, the model encourages users to establish these with everyone. The aim is to affect only the visibility the sessions, not the sessions themselves.

As the scheme requires users to possess each other's public keys, as well as have established a shared secret prior to contact, initialization is a problem. Operators of high-security systems and concerned users might go through the trouble of creating such a database manually. However, most users do not appreciate the limitations and extra work, and would be satisfied with a partial open exposure, at least at times, but with the privacy enhancements used whenever possible.

The shared secrets could then be agreed on during the first contact, stored to be used for subsequent sessions.

This leads us to the question of how to present these options, *modes*, to the user. Although related to the traditional concept of presence, it has a slightly different meaning and effect, not familiar from centralized systems. In section 4.3, we present how this is solved in our implementation, discussing the pros and cons of our solution.

4 Implementation

To evaluate the feasibility of our model, and investigate the usability issues, we implemented the privacy enhancements into Helsinki Institute for Information Technology's (HIIT) Host Identity Protocol (HIP)-based P2P communication prototype.

4.1 Technology

The prototype is implemented as a light Linux-based, locally run, SIP proxy for ordinary SIP user agents (SIP UAs). It intercepts SIP signaling, converting them into the appropriate operations in the P2P model [12]. The prototype has been adopted to mobile and embedded Linux environments, and is currently used in a trial on Nokia N810 Internet tablets.

HIP [13] is a communication framework which splits the notion of identifier and locator from their union in the current Internet. HIP adds a layer between the network and transport layer, allowing applications to address end-points using cryptographically generated identifiers, instead of network addresses. HIP is used in the prototype to establish secure and robust connections between the peers, as it provides advanced features such as multihoming, mobility and NAT traversal.

HIP uniquely identifies hosts using Host Identifiers (HIs), the public part of an asymmetric key pair. As HIs are cumbersome to manage, applications use Host Identity Tags (HITs) for addressing remote hosts. These are 100 bit hashes of the HIs, which combined with the ORCHID [14] prefix are expressed as IPv6 addresses. The translation of HITs to network locations (IP addresses) is done by the underlying HIP stack, which also handles connection establishment and mobility management.

A HIP connection is established through a four-way Diffie-Hellman compliant handshake called the base exchange (BEX). During this denial of service (DoS)-protected process the hosts are authenticated and keying material is established. This keying material is used to establish a security encapsulation (using the IPSec Bound End-to-End Tunnel (BEET) mode) to protect the data traffic. As the encapsulation and routing are performed transparently by HIP, it provides advanced features (including security and IP mobility) for applications with no additional implementation effort.

4.2 Prototype Overview

As with the IETF P2PSIP protocol, the prototype adds a level of abstraction on top of the P2P overlay, accessing it through a hash table-like interface (*get*, *put* and *remove*). This allows for flexibility in its implementation, but offers no control over how the data is managed in the P2P overlay. As message routing through the overlay is not supported, it is used only to retrieve the contact information needed establish the HIP data connections between peers.

Although not the focus of the privacy model, the use of HIP could also improve privacy during the connection set-up procedure. As described in section 2, the IETF P2PSIP protocol embeds ICE address candidates in the Attach-message. Even though the content is encrypted, the NodeIds of peers are still visible, providing clues about a call. By using HIP we have a wider choice of how the connections are established. The HIP connection handshake, could be routed through the overlay, but we also have the option of using trusted relays according to the HIP NAT traversal draft [15]. In addition, the use of the BLIND extensions [16] and one-time HITs would hide the identity of the end-points.

The identity scheme of the prototype is based on trusted third-party certificates binding SIP AORs to the public keys, in line with the assumptions of section 3. Normally the SIP AORs are used as keys when storing registration packages. In addition to this *open* contact information, the privacy extensions adds the notion of *secret* storage keys, constructed using the shared secrets, under which encrypted registration packages are *hidden*.

4.3 Privacy Enhancements

The implementation of the privacy extensions included creating a local peer database and modifying how the data in the distributed storage is managed. The prototype maintains a database of the public keys and shared secrets of all the peers it has been in contact with. Although it is possible to manually configure a shared secret, a key-negotiation protocol was implemented, which automatically establishes (or reconfigures) a shared secret when connected.

This protocol is a simple three-part protocol which checks whether both peers have the same secret, and in case they differ, re-initializes one from two parts. Both peers propose one (randomly generated) part each, which are combined into the final secret. This allows for easy reconfiguration, in case either peer looses it or wants to prevent intermediate nodes from tracking these between sessions. The prototype can be configured to re-initialize these secrets each time a new session is established.

The privacy enhancements are activated using a configurable setting of the prototype, affecting how the registration packets are retrieved and published. After analyzing different use-cases, we came to the conclusion that three different modes are needed:

Open. The registration package is published using both the open and secret storage keys. Lookup is done using only the open, even when a shared secret has been established with the recipient.

Relaxed. As in the Open mode, the registration package is published using both keys. The lookup differs; if a shared secret has been established, only the secret storage key is used, otherwise the open.

Paranoid. When set to paranoid, only the secret keys are used, both when publishing and performing lookups.

The reasoning behind the Paranoid mode is understandable; the prototype uses the privacy enhancement fully, revealing as little as possible. But as discussed in section 3, for the enhancements to be usable, we also need to have an option of being openly exposed, at least temporarily. The need for two, nearly identical, modes is based on the assumption that nodes might get reset at times, loosing the database of shared secrets.

The Relaxed mode operates using a *best-effort* principle. Whenever a shared secret has been established, the privacy extensions are used without even trying the open key. By falling back to the open key, we would avoid the additional Open mode. But this could be seen as a violation of the privacy of the recipient, if in Paranoid mode. By falling back the use of open keys in the Relaxed mode, either by using subsequent or simultaneous lookups, we might reveal a relationship between the two. Although the risk of this may be low, the usability also seemed better, as the user knows beforehand whether the privacy enhancements are used. This is however something that might still change depending on the feedback we receive.

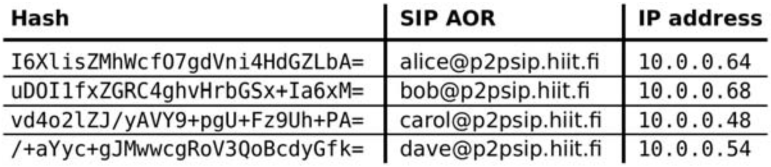
4.4 Evaluation

As a demonstration of our model, we created a small test bed of four user accounts to simulate a group of privacy conscious peers. The prototype, with the privacy enhancements, was deployed on hand-held Nokia N810 Internet tablets, with an appropriate user account configured on each. The tablets were given to a set of test users, who were asked to use them for voice- and video calls, and instant messaging.

The tablets were connected using a standard IEEE 802.11b Wireless LAN (WLAN) access point with a Dynamic Host Configuration Protocol (DHCP) server to provide IP addresses. To simulate a P2P overlay, we used LAN broadcast as the back-end for the distributed storage. To evaluate the enhancements, we recorded and analyzed the network traffic generated by the tablets, simulating the wost-case scenario where an intermediate can log all traffic. The purpose was to compare what can be deduced from the logs before and after the privacy enhancements.

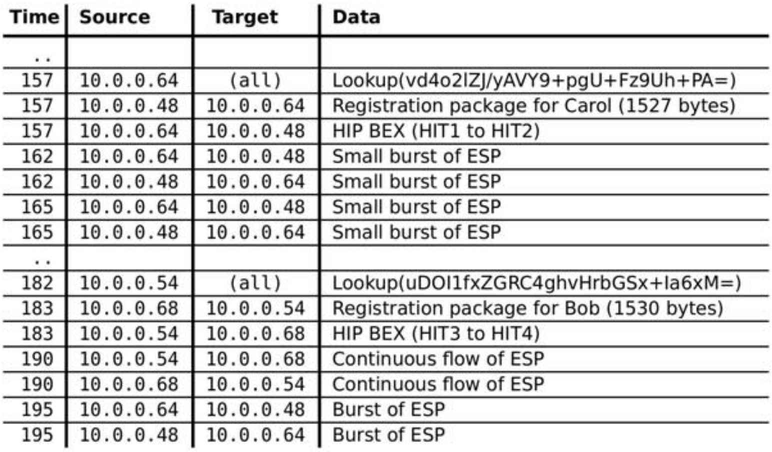
Initially the prototype was set to the Open mode. Although the keys used in the lookup are hashes of the SIP AORs, the identifiers they correspond to can be read from the clear-text response. After a short while, we had compiled a mapping of these, together with the IP address used by each peer (presented in figure 1),

A sample from the recorded traffic is presented in figure 2. From the log we see Alice establishing a connection to Carol at 157 seconds. Even though the actual data traffic is secured using HIP, the pattern of relatively small ESP burst provides clues about an instant messaging session. At 182 seconds, we see



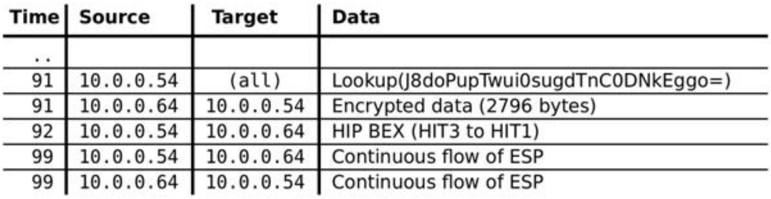
Hash	SIP AOR	IP address
I6XlisZMhWcfO7gdVni4HdGZLbA=	alice@p2psip.hiit.fi	10.0.0.64
uDOI1fxZGRC4ghvHrbGSx+Ia6xM=	bob@p2psip.hiit.fi	10.0.0.68
vd4o2lZJ/yAVY9+pgU+Fz9Uh+PA=	carol@p2psip.hiit.fi	10.0.0.48
/+aYyc+gJMwwcgRoV3QoBcdyGfk=	dave@p2psip.hiit.fi	10.0.0.54

Fig. 1. The hash to SIP AOR relationships found, with responsible IP address



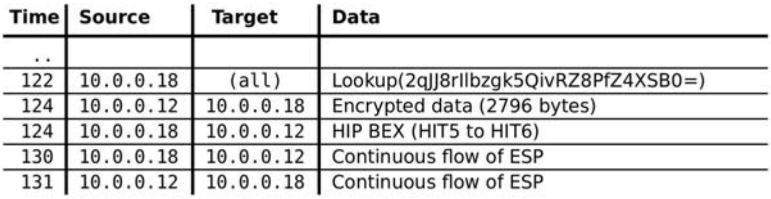
Time	Source	Target	Data
..			
157	10.0.0.64	(all)	Lookup(vd4o2lZJ/yAVY9+pgU+Fz9Uh+PA=)
157	10.0.0.48	10.0.0.64	Registration package for Carol (1527 bytes)
157	10.0.0.64	10.0.0.48	HIP BEX (HIT1 to HIT2)
162	10.0.0.64	10.0.0.48	Small burst of ESP
162	10.0.0.48	10.0.0.64	Small burst of ESP
165	10.0.0.64	10.0.0.48	Small burst of ESP
165	10.0.0.48	10.0.0.64	Small burst of ESP
..			
182	10.0.0.54	(all)	Lookup(uDOI1fxZGRC4ghvHrbGSx+Ia6xM=)
183	10.0.0.68	10.0.0.54	Registration package for Bob (1530 bytes)
183	10.0.0.54	10.0.0.68	HIP BEX (HIT3 to HIT4)
190	10.0.0.54	10.0.0.68	Continuous flow of ESP
190	10.0.0.68	10.0.0.54	Continuous flow of ESP
195	10.0.0.64	10.0.0.48	Burst of ESP
195	10.0.0.48	10.0.0.64	Burst of ESP

Fig. 2. Samples of the traffic log before the privacy enhancements



Time	Source	Target	Data
..			
91	10.0.0.54	(all)	Lookup(J8doPupTwui0sugdTnC0DNkEggo=)
91	10.0.0.64	10.0.0.54	Encrypted data (2796 bytes)
92	10.0.0.54	10.0.0.64	HIP BEX (HIT3 to HIT1)
99	10.0.0.54	10.0.0.64	Continuous flow of ESP
99	10.0.0.64	10.0.0.54	Continuous flow of ESP

Fig. 3. Samples of the traffic log with the privacy enhancements in use



Time	Source	Target	Data
..			
122	10.0.0.18	(all)	Lookup(2qJJ8rIlbzgk5QivRZ8PfZ4XSB0=)
124	10.0.0.12	10.0.0.18	Encrypted data (2796 bytes)
124	10.0.0.18	10.0.0.12	HIP BEX (HIT5 to HIT6)
130	10.0.0.18	10.0.0.12	Continuous flow of ESP
131	10.0.0.12	10.0.0.18	Continuous flow of ESP

Fig. 4. Samples of the traffic log with the IP and HIT reset, and privacy enhancements in use

how Dave contacts Bob. The continuous flow of ESP suggests a voice- or video call. At 195 seconds we see again traffic, most likely instant messages, exchanged between Alice and Carol.

After the initial session, the privacy mode was set to Paranoid, and the devices rebooted to reset any existing IPSec security associations. A short sample from the traffic is shown in figure 3. Although we can not determine what was sought at 91 seconds, it is fairly certain that a call was made between Dave and Alice. The lookup response is encrypted, but the size fits within what we would expect of an encrypted registration package, and the data traffic matches the pattern of a voice- or video call. The IP addresses (and the HITs) reveal the peers involved.

To simulate the use of additional source address hiding techniques, the HIP Host Identity database was reset on all devices, and the DHCP server configured to provide addresses from a different IP range. A sample of the traffic is shown in figure 4. We can still see that a voice- or video call is made, but the peers remain unknown.

5 Discussion

As the traffic analysis shows, the privacy enhancements does a fair job ensuring that the data managed in P2P communication systems reveal as little as possible on the application-level. However, for complete privacy, we need to consider other factors as well.

Using HIP to secure the data connections provides many benefits, but identifies the end-points to outsiders as well. Even without HIP, IP, hardware Media Access Control (MAC) addresses, or other host identification schemes might be used for the same purpose. Although the session data is encrypted, traffic analysis may reveal the type of content. However, these issues are considered out of scope for our work, as they relate to the general problem of communications privacy, and is being addressed by work such as the SlyFi [17] design. Our focus is only on the data managed by the distributed storage, and what it reveals; an issue specific to P2P systems.

The privacy model has usability issues which need to be examined further. The proposed Open, Relaxed and Paranoid modes seems reasonable from a technical point of view, but the usability of these needs to be verified.

6 Summary and Future Work

In this paper, we reviewed the differences between fully distributed and centralized communication systems, and the privacy issues in both. We presented a simple model for enhancing privacy in P2P communication systems, such as the one being developed by the IETF P2PSIP working group. The model obfuscates the data managed by the overlay-based storage by using public-key encryption and shared secrets. As it affects only the application-level data handling, it can be used in conjunction with addition lower-level techniques for greater security.

We highlighted possible usability problems, and concluded with an overview of our implementation and traffic analysis from its use, demonstrating its effect. The model does well in protecting the privacy of users on the application-level. It does increase the load on the system, but not unreasonably.

Future work will concentrate on evaluating the usability and combining the model with additional techniques to improve the overall privacy. This include examining the impact of different link-layer techniques and methods for preventing traffic pattern analysis. Also, wider trials and simulations of real user behavior, possible on a global test-bed such as PlanetLab [18], should be conducted to evaluate the feasibility of the solution.

Acknowledgments. The authors like to thank everyone associated with the design and development of the work presented in this paper. First and foremost Cheevarat Jampathom, responsible for a large share of it, but also the people involved in the TrustInet and InfraHIP projects at HIIT.

References

1. Mondal, A., Kitsuregawa, M.: Privacy, security and trust in p2p environments: A perspective. In: 17th International Conference on Database and Expert Systems Applications, pp. 682–686 (2006)
2. Good, N.S., Krekelberg, A.: Usability and privacy: a study of kazaa p2p file-sharing. In: CHI 2003: Proceedings of the SIGCHI conference on Human factors in computing systems, pp. 137–144. ACM Press, New York (2003)
3. Lu, Y., Wang, W., Bhargava, B., Xu, D.: Trust-based privacy preservation for peer-to-peer data sharing. IEEE Transactions on Systems, Man and Cybernetics 36(3), 498–502 (2006)
4. IETF P2PSIP working group, `http://www.ietf.org/html.charters/p2psip-charter.html`
5. Jennings, C., Lowekamp, B., Rescorla, E., Baset, S., Schulzrinne, H.: REsource Location and Discovery (RELOAD)(2008) (Work in progress)
6. Rosenberg, J.: Interactive Connectivity Establishment (ICE): A Protocol for Network Address Translator (NAT) Traversal for Offer/Answer Protocols (2007) (Work in progress)
7. Stoica, I., Morris, R., Karger, D., Kaashoek, M.F., Balakrishnan, H.: Chord: A scalable peer-to-peer lookup service for internet applications. In: Proceedings of the 2001 conference on applications, technologies, architectures, and protocols for computer communications, pp. 149–160. ACM Press, New York (2001)
8. Douceur, J.R.: The sybil attack. In: IPTPS 2001: Revised Papers from the First International Workshop on Peer-to-Peer Systems, pp. 251–260. Springer, London (2002)
9. Rivest, R.: The MD5 Message-Digest Algorithm. RFC 1321 (Informational) (1992)
10. Eastlake III, D., Hansen, T.: US Secure Hash Algorithms (SHA and HMAC-SHA). RFC 4634, Informational (2006)
11. Lamport, L.: Password authentication with insecure communication. Communications of the ACM 24(11), 770–772 (1981)
12. Koskela, J.: A HIP-based peer-to-peer communication system. In: ICT 2008: Proceedings of the 15th International Conference on Telecommunications, pp. 1–7 (2008)

13. Moskowitz, R., Nikander, P.: Host Identity Protocol (HIP) Architecture. RFC 4423, Informational (2006)
14. Nikander, P., Laganier, J., Dupont, F.: An IPv6 Prefix for Overlay Routable Cryptographic Hash Identifiers (ORCHID). RFC 4843, Experimental (2007)
15. Komu, M., Henderson, T., Tschofenig, H., Melen, J., Keränen, A. : Basic HIP Extensions for Traversal of Network Address Translators (2009) (Work in progress)
16. Ylitalo, J., Nikander, P.: Blind: A complete identity protection framework for endpoints. In: Christianson, B., Crispo, B., Malcolm, J.A., Roe, M. (eds.) Security Protocols 2004. LNCS, vol. 3957, pp. 163–176. Springer, Heidelberg (2006)
17. Greenstein, B., McCoy, D., Pang, J., Kohno, T., Seshan, S., Wetherall, D.: Improving wireless privacy with an identifier-free link layer protocol. In: MobiSys 2008: Proceeding of the 6th international conference on Mobile systems, applications, and services, pp. 40–53. ACM, New York (2008)
18. PlanetLab: An open platform for developing, deploying and accessing planetary-scale services, `http://www.planet-lab.org/`