Nonesuch: a Mix Network with Sender Unobservability

Thomas S. Heydt-Benjamin
University of Massachusetts at Amherst
tshb@cs.umass.edu

Andrei Serjantov
The Freehaven Project
schnur@gmail.com

Benessa Defend
University of Massachusetts at Amherst
defend@cs.umass.edu

ABSTRACT

Oblivious submission to anonymity systems is a process by which a message may be submitted in such a way that neither the anonymity network nor a global passive adversary may determine that a valid message has been sent. We present Nonesuch: a mix network with steganographic submission and probabilistic identification and attenuation of cover traffic. In our system messages are submitted as stegotext hidden inside Usenet postings. The steganographic extraction mechanism is such that the the vast majority of the Usenet postings which do not contain keyed stegotext will produce meaningless output which serves as cover traffic, thus increasing the anonymity of the real messages. This cover traffic is subject to probabilistic attenuation in which nodes have only a small probability of distinguishing cover messages from "real" messages. This attenuation prevents cover traffic from travelling through the network in an infinite loop, while making it infeasible for an entrance node to distinguish senders.

Categories and Subject Descriptors

D.2.11 [**Software**]: Software Architectures, Information Hiding; C.2.1 [**Computer Systems Organization**]: ,Computer-Communication Networks, Network Architecture and Design; E.3 [**Data Encryption**]: ,Public Key Cryptosystems

General Terms

Algorithms, Security

Keywords

Mix Networks, Sender Unobservability, Public Key, Steganography, Oblivious Channels, Minx Packet Format

1. INTRODUCTION

Many different kinds of anonymity networks exist in which it is difficult to link senders of messages to recipients. It is common among such anonymity systems that certain kinds of adversaries can easily determine the identity of all entities submitting messages to the network. In Mixminion [5] and other mix-based designs, for example, a sender can be identified with certainty by a global passive adversary or even a single corrupt entry node [17]. In other words, many systems (with the notable exception of DC-nets) provide unlinkability rather than sender unobservability and hence the identities of all the senders are easily known.

There are many situations, however, in which mere knowledge of submission is too much knowledge to permit the adversary to achieve. Even if the contents and the recipient of a message are occluded by the anonymity network, a sender may wish to keep secret the very fact of participation in an anonymity protocol. Users who need this level of privacy protection include, for example, citizens of oppressive governments with widespread surveillance, and corporate whistle blowers. In addition to possible threats of negative consequences for participating in anonymous communication, these users may be prevented from participation by an adversarial service provider.

We propose Nonesuch: a high latency mix network which supports oblivious submission. Nonesuch provides the same sender-receiver unlinkability provided by other mix networks, while providing better sender anonymity and strong protection against tagging attacks. Nonesuch offers improved sender anonymity against both compromised nodes and against passive adversaries.

In Nonesuch, users steganographically embed messages in images which they then post to the most popular Usenet newsgroups. Note that by the nature of popularity, neither subscribing to nor posting to these newsgroups is a suspicious activity. The majority of images in Usenet will not contain stegotext and will serve as cover traffic. Nonesuch nodes operate by performing a steganographic extraction on each new posting to the protocol-specified range of Usenet newsgroups. These messages are then routed based on a tunably sparse routing table. This routing mechanism permits correct routing of messages while only slightly reducing the anonymity set of a valid message from the set of all valid messages and cover traffic combined. Cover traffic is probabilistically revealed as such and is attenuated out of the network as soon as it is detected.

Anonymity of messages and quantity of cover traffic are inversely proportional. As a message travels deeper in the anonymity network it becomes more difficult to link it to its entrance and thus its sender. In Nonesuch, cover traffic is attenuated as it travels deeper. Therefore we concentrate

WPES'06, October 30, 2006, Alexandria, Virginia, USA.

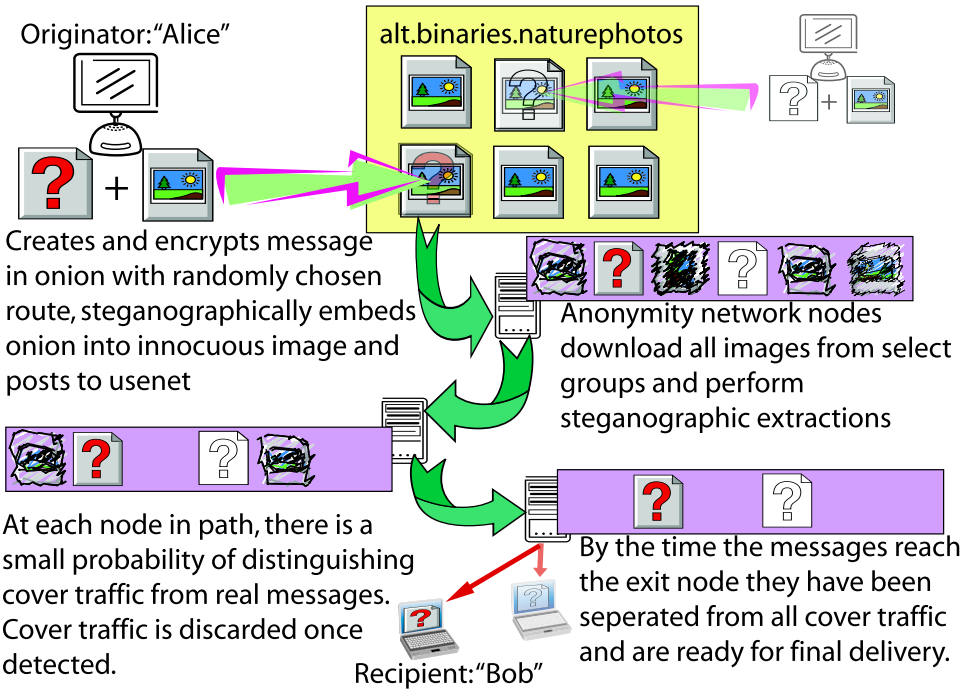

Figure 1: In the Nonesuch protocol submission is accomplished over a subliminal channel, and anonymity nodes can only distinguish real messages from cover traffic with small probability.

cover traffic where it is most needed; to occlude submission. Cover traffic then becomes sparse where it is not needed; at the end of a real message's path.

In this paper we:

1. Present a novel design for oblivious submission
2. Utilize the Minx packet format in what we believe is a novel way
3. Present a Baysean inference based attack against uncertainty about the number of real and dummy messages in the mix
4. Demonstrate that this attack does not alter the advantage of Nonesuch over more traditional Mix networks

2. RELATED WORK

Chaum [4] introduced the concept of anonymizing mix networks. ore modern systems, such as Mixminion [5] allow features such secure anonymous replies, resistance to tagging, and more security against active attacks. More advanced analysis techniques allow us to prove [2] or at least to some extent analyze mix networks [15, 7].

Nevertheless, mix net protocols offer strong sender-receiver unlinkability, especially when few messages are exchanged. However, none of these systems conceal the fact that an originator has submitted a message to the mix net when the originator's outgoing traffic is monitored by a passive adversary. Nonesuch offers secret submission to a mix net using a steganographic covert channel.

Our work is inspired in part by Matthias Bauer's work on limited unobservability in the context of a mix network [1]. This system uses covert channels in HTTP to make it more difficult for an adversary to distinguish senders from non-sending participants. An important difference between Bauer's system and ours is that his system requires the participation of the servers belonging to popular websites, whereas our protocol uses only existing features of Usenet. We observe that it is much easier for a user to submit a posting to Usenet than to submit software to a major website and ask them to run it on their servers.

The Minx packet format uses probabilistic attenuation in order to gain resistance to active tagging attacks [6]. While this goal is quite different from our central one of oblivious submission we find the Minx packet format compatible with our system as discussed in section 3.3.

All mixes are vulnerable to timing attacks [13]. However, Since all non-Nonesuch postings to the most popular Usenet image newsgroups serve as cover traffic in our system, the volume of cover traffic in Nonesuch is both high and difficult for an adversary to control. Our system should therefore be more resilient than mix-based systems which have less cover traffic, or which rely on artificial (and therefore more easily manipulated) dummy traffic for cover.

DC-nets [3] provide strong sender anonymity, in which a passive adversary is not aware of when an originator is sending a message. However, the adversary can still determine that the sender is a member of an anonymity network. Disclosure of this knowledge is unacceptable in situations where entities such as governments or companies may prohibit involvement in anonymity networks. Nonesuch provides strong sender anonymity without requiring the sender to join an anonymity network. Our system does not, however, provide recipient anonymity which is a property of DC-nets.

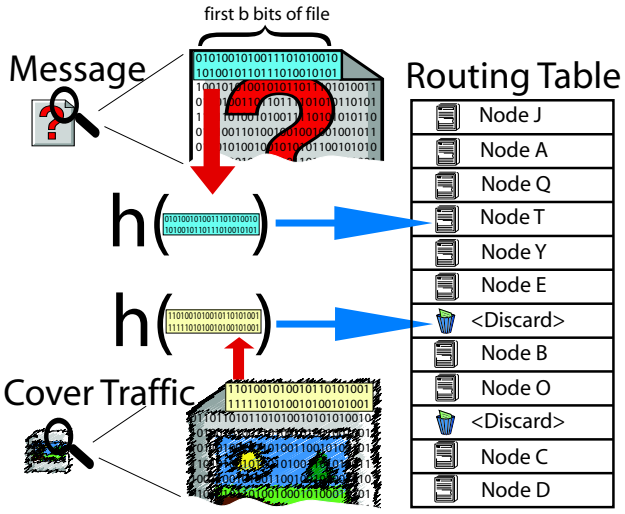

Figure 2: A Nonesuch message is structured such that some portion of the header will hash to the routing table index of the next node in the path through the anonymity network. Cover traffic will usually hash to a valid node, and thus be indistinguishable from a real message. With some probability a cover message will hash to an empty slot in the routing table and be discarded.

3. DESIGN

3.1 Subliminal Channel

In order for Nonesuch to be effective at concealing submission, we require a subliminal channel in which data with and without embedded hidden messages are indistinguishable from each other in polynomial time up to some reasonable assumption. Established literature suggests that this requirement is reasonable. For example; Hopper, Langford, and von Ahn prove the existence of steganographic protocols with the aforementioned property if one-way functions exist [9].

In a follow-up paper, Von Ahn et. al. introduce a public-key steganography protocol which is well suited to our requirements. In this protocol a steganographic extraction from an image is indistinguishable from random noise (up to standard cryptographic assumptions) unless the extracting entity possesses the appropriate private key [18]. Some stegosystems require a-priori knowledge of an unmodified cover image in order to successfully extract stegotext. Such a requirement is incompatible with Nonesuch. The stegosystem introduced in [18] has no such requirement; the appropriate private key is sufficient for extraction.

3.2 Network Setup and Global Parameters

The Nonesuch network has several global parameters of which all parties must have knowledge. These Parameters include a global routing table that is published once per long term time epoch (e.g., one week), a fixed message length (short messages will be padded), a time epoch duration, and the public keys of all participating anonymity nodes. The generation and propagation of these parameters are discussed below.

Propagation of anonymity network global parameters is a difficult problem, and is not the focus of this work. We note that the global parameter distribution problem for Nonesuch is almost exactly the same as that for Tor and Mixminion. A satisfactory solution for either of these systems will also suffice for Nonesuch. At the moment both Tor and Mixminion rely on trusted directory servers [8, 5]. For simplicity of discussion, let us assume the existence of a trusted certificate authority, which we will call CA. In implementation, CA can be replaced with any suitable directory server scheme.

Nodes wishing to participate in the protocol contact CA before the beginning of a new time epoch. The nodes must participate throughout the epoch or else some messages will be lost. We consider this loss to be acceptable in view of the increased security resulting from the stability of the network. The routing table is constructed as a hash table in which each Nonesuch node's address appears once, and the table contains a number of blank entries. The ratio of non-empty addresses entries to total table size (R) is a security parameter. This and the other remaining global parameters may be selected depending on the degree of desired security and empirical measurements of network performance.

At the beginning of the time epoch, CA publishes all of the global parameters marshalled as article headers in postings to each of the most popular Usenet image newsgroups. Potential users of the anonymity system will be able to download this information without revealing their intent to participate in the anonymity protocol because it is typical for a Usenet client to download all header information of new articles in a newsgroup to which its user subscribes [11]. By virtue of the popularity of the top newsgroups, subscription to one or more of these groups will similarly not disclose meaningful information regarding intent to send an anonymous message. Usenet news messages may have a great diversity of data in the header information, and the RFC specified behavior for compliant servers and clients is to pass through all unrecognized headers unchanged [10]. This makes the header space a suitable publication medium for Nonesuch global parameters.

3.3 Packet Format

There are two essential requirements that a packet format must fulfill in order to be suitable for use in a Nonesuch network.

- The packet must be indistinguishable from random noise until the final layer of encryption is removed from the message body.
- The packet must be impervious to tagging attacks.

The Minx packet format introduced by Danezis and Laurie satisfies both of these requirements [6]. The first bits of a Minx packet consist of either the routing table index of the next node in the path, or a special prefix called "final" to indicate that a message has reached its terminal node and should be delivered to the recipient. By simply expanding the routing table such that there are many "final" entries in the routing table at randomly distributed indeces, we find the Minx packet format compatible with Nonesuch. In Minx, when a "final" prefix is revealed to a node, the message body is decrypted. If the message body is well formed, it is delivered, and if it is garbled it is discarded as the likely result of an attempted active attack. By contrast, in our system most messages are cover traffic. Cover messages are

disposed of via the same probabilistic mechanism as in Minx, adding to the difficulty of learning useful information from active attacks on real messages.

3.4 Message Submission

To submit a message to the Nonesuch network, a message sender A begins by selecting a path through the network. The path selection algorithm is as follows: A generates random bitstrings of length b (where b is a global parameter), and checks to see to which entry of the routing table each random bitstring maps. Any bitstring that maps to an empty entry is rejected, and A continues to generate random bitstrings until a complete random route has been selected.

In abstract (without regard to a particular packet format): A then creates a multi-layered onion using a key at each layer n which corresponds to the node dictated by the nth random bitstring. At each layer a header consisting of the random bitstring representing the next node in the path must appear, and the message must be padded as necessary such that all packets will be the same length regardless of their progress through their path.

If the Minx packet format is adopted, the algorithm given in [5] for the construction of a Minx packet without reply block suffices with the following modification: Whereas in Minx an onion layer is prefixed with a routing bitstring and a "final tag" ($k_n|tag$ in equation 6 of [5]) appears, a random bitstring which hashes to an empty routing table entry must be substituted. This is quite similar in intent to the corresponding value in Minx, but is slightly different in execution. We make this change because we wish to avoid the use of fixed low-entropy tags which appear in cleartext for fear that they may aid in making messages statistically distinguishable from cover traffic.

Next, A steganographically conceals the completed packet in an image file, and posts it to one of the most popular Usenet image newsgroups. As mentioned previously, the steganographic extraction will only succeed when the private key of the entry node is used. This means that every valid message will result in $M-1$ cover packets and 1 real message packet, where M is the number of mix nodes in the Nonesuch network. In contrast, posting of a non-message containing Usenet image will result in M cover packets.

3.5 Operation of Anonymity Nodes

The Nonesuch network monitors the most popular Usenet image newsgroups, and each node downloads all images that are large enough to contain stegotext of the protocol specified fixed length. Each node performs a steganographic extraction on each image (truncating the output to the fixed packet length as necessary) resulting in a packet which may be an encrypted message or may be the random result of performing an extraction on an image file which contains no message. We will henceforth call such a fixed length potential message a packet. After extraction, an entering packet is treated just like a packet received from another Nonesuch node.

To thwart timing based-attacks, every Nonesuch node must individually delay each packet it receives in a queue for a random amount of time before forwarding it to the next node. Thus, an adversary cannot link an exiting packet to an entering packet, provided the queue contains multiple packets.

3.5.1 Non-exit Nodes

The behavior of a non-exit node in the Nonesuch network is essentially the same as the behavior of nodes in a Mixminion network using the Minx packet format. Briefly: each non-exit node decrypts the header of each incoming packet with its private key. This header reveals a symmetric decryption key for the remainder of the packet. Symmetric decryption then reveals the next layer of the onion and the address of the next node in the path. This address appears in the form of a random bitstring which must be looked up in the routing table. The new onion layer is then padded up to the fixed packet length.

The routing table is tunably sparse, and it is by this means that cover traffic is probabilistically attenuated. For example, if the routing table contains 5% blank entries then 95% of cover traffic will be indistinguishable from valid messages with respect to routing at any given node. We call this a routing table ratio of $R = 0.95$.

3.5.2 Exit Nodes

When a real message reaches the final node in its path, the header which would normally specify the next node in the path maps to an empty routing table entry. When this occurs, the exit node checks the message body, and if it is well formed (for example, if it contains cleartext email headers) it will then be delivered outside of the Nonesuch network. Because of the properties required of the packet format, the message body cannot be well formed if the packet has been tagged by an active adversary.

Only the exit node has sufficient knowledge to distinguish a cover packet from a message containing packet. Because the exit node is at the far end of a multi-hop path from the packet's entrance to the network, the adversary has a negligible probability of determining the sender of any message.

4. ANONYMITY ANALYSIS OF NONESUCH

In this section we look at the anonymity properties provided by Nonesuch. As we already mentioned above, Nonesuch is very similar to traditional mix networks. Its unlinkability properties are correspondingly similar. One key difference is in the amount of traffic – by design Nonesuch has many dummy messages flowing through it, hence one expects the anonymity sets to be larger and entropy of anonymity probability distributions to be higher. For this to hold, however, it is key that the attacker should not be able to distinguish between real and dummy messages. The packet format we have chosen supports this idea; in this section we investigate a stronger property, namely the extent to which the attacker can determine the number of real and dummy messages.

Such an analysis is clearly of interest to the adversary: if he can show that all the messages are "real messages" then Nonesuch has not provided any additional properties compared to a traditional mix network. If, however, there is no way of doing this and the attacker has to settle for dealing with probability distributions of the number of real messages, then information has been hidden in excess of a normal mix network. Note that even if the attacker has obtained the knowledge that 50 out of 100 messages are real, he is no closer to determining which ones are real and which ones are not!

Let us introduce the following definitions:

f	–	unconditional average fraction of good messages in network
N	–	total number of packets in the mix
r	–	number of "good" (anonymous) messages in the mix
$N-r$	–	number of dummies
z	–	number of messages leaving the mix
R	–	fraction of good entries in the routing table

We start by looking at the distribution of the number of real messages given that z messages were observed leaving the mix.

$$\begin{aligned} P(z|N,r) &= 0 && \text{if } z < r \\ &= \left(\begin{array}{c} N-r \\ z-r \end{array} \right) R^{z-r}(1-R)^{N-z} && \text{if } z \geq r \end{aligned}$$

Hence r messages are good, and leave the local mix node, $z-r$ messages are dummies but get through with probability R^{z-r} and $N-z$ messages are dummies and get attenuated.

Of course what we are really interested in is the probability distribution of real messages given that z messages were observed leaving the mix, i.e. $P(r|z,N)$.

Using Bayes theorem,

$$P(r|z,N) = \frac{P(z|r,N)P(r|N)}{P(z|N)}$$

And $P(z|N) = \sum_{r \in N} P(z|r,N)P(r|N)$. We can approximate $P(r|N)$ by a binomial distribution: $P(r|N) = Nf^r(1-f)^{N-r}$.

Let us take an example. The attacker is observing a mix node and sees 100 messages enter it. He does not know how many of these are "real" (r), but he does have an idea that around 50% of the traffic is comprised of real messages (this could be obtained from observing the input and the output of the mixes over a long period of time). For example, let us consider an attenuation coefficient of $R = 0.85$ (the value of R should be tuned to desired security versus efficiency tradeoff, we choose 0.85 for the sake of exposition as it offers a reasonable compromise between these). Suppose then the attacker observes that all 100 submitted packets leave the mix as real messages. This leads the adversary to suspect that more messages are real. If only 20 messages come out, his estimate of the number of real messages decreases. This is illustrated in Figure 3.

Note that this inference can be made simply by observing one mix node. Naturally, if all the messages from the first node go to the second node, the inference can be repeated to obtain a more accurate estimate for the number of good messages. If, however, messages from other mixes arrive to the second mix, the inference we have used above would not be accurate as the attacker may have already inferred something about the likelihood of them being a real message. An alternative inference procedure may be developed where the attacker computes an estimate of the probability of a packet being a real message *for each packet* and then combined these when the packets enter the mix. Such a method would be much more complex and possibly computationally intensive; we leave this for future work.

This example shows that while an attacker, such as an entry node or a passive adversary around an entry node, can gain some knowledge about the number of real messages travelling through the node[1], there is still significant uncertainty about this, even in the case where all the messages are real (see for example Figure 3, case for $z = 100$). More concretely, we have shown that unlike in more traditional mix systems, the entry mix or a passive attacker *cannot determine the number of real messages entering the mix* and hence their senders.

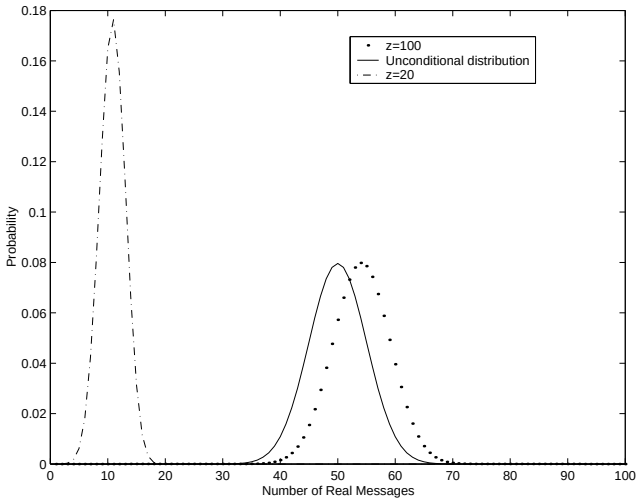

Figure 3: Probability distributions of real messages: unconditional; after observing 20 messages exiting the mix; after observing 100 messages exiting the mix

A more powerful adversary can mount a different attack on the extra property provided by Nonesuch. If the attacker owns one or more Nonesuch nodes, then he can monitor many messages which are sent by Alice and observe the probability with which they are retained by the mix. If this is significantly higher than R, the attacker may conclude that A is, in fact a sender of Nonesuch messages. This is, of course, merely an instance of a long term intersection attack on anonymity properties which have been paid significant attention in the literature, e.g. [16, 12].

5. EFFICIENCY

The Nonesuch network derives much of its security from the large amount of cover traffic that circulates. The cover traffic retention ratio R (the probability that a cover message is retained by any given node) is one of the most critical parameters to the protocol. When selecting an appropriate value for R there is a tension between desired security properties and the possibility of overwhelming the network with more messages than can be processed. In considering R values we have made several observations.

A contributing factor to the amount of traffic in the Nonesuch network is the rate at which new image files are posted to the set of monitored Usenet newsgroups. We determined that the average posting rate to the top ten most populous Usenet newsgroups combined is 0.583 messages per second (daily observation of 10 most populous newsgroups visible to the Easynews Usenet provider from 04/13/2005 to 05/18/2005). We explicitly ignore the bursts that are common in such traffic since by the nature of mixing our system

[1]and clearly the identities of the all the posters of the newsgroup messages which enter the node, including the real senders of anonymous messages

will queue messages during overly busy periods to be forwarded during less busy periods. If over time the average rate significantly increases or decreases, the protocol can respond by choosing to monitor fewer or more newsgroups in order to keep the input rate consistent. Such changes would be communicated to clients and nodes along with the rest of the published global parameters.

The total number of packets in the system can be expressed as:

$$\hat{r}l + \sum_{i=0}^{\infty}(\hat{N} - \hat{r})(R^i)$$

where $\hat{r}$ is the total number of real messages, $\hat{N}$ is the total number of packets entering the system and R is the cover traffic retention ratio, and M is the number of nodes in the Nonesuch network. Given that $\hat{r}$ is more or less fixed, R and M are the factors that will determine the per node amount of traffic in the Nonesuch network. Clearly, the amount of traffic in the system is easily tunable by altering R.

In order to demonstrate the efficiency in a more concrete manner, we make the following comparison to an existing network: setting $R = 0.5$ attenuates the dummy traffic in an average of two hops, which makes adding dummy traffic much less expensive than increasing capacity of the network for real traffic (typical route length of an anonymous email is between 2 and 5). Hence, if we take 50% of the traffic to be real (with an average route length of 4), then we need to increase the capacity of the network by 25% compared to a Mixminion-like system without the extra sender untraceability properties. This can easily be done by adding extra nodes to the system (the typical number of Mixmaster or Mixminion nodes has been around 30 over the period of 2000-2005) [14].

Processor overhead is unlikely to be the limiting factor in the potential throughput of the Nonesuch system. We ran an OpenSSL benchmark on a Dell Precision 470 running linux (2GHz microprocessor), which we take to be representative of relatively ubiquitous consumer hardware. This benchmark indicates that such a computer can process 4.434 MB/s of 1024 bit RSA decryption (average of 10 runs). Relatively expensive asymmetric decryption must only be performed on the small (around 128 bits) header of a packet. The rest of the decryption is symmetric, and can most likely be performed at line speed.

It is far more likely that communications will be the limiting factor to network size, especially in a hybrid network consisting of some corporate or university servers, and some servers belonging to home users.

6. CONCLUSION

In this paper we considered the problem of making senders of anonymous messages indistinguishable from Usenet newsgroup users. We have presented a novel design of Nonesuch, an anonymity system with these desired properties. We have found that Nonesuch can make use of an existing packet format, Minx, and draw on some of the ideas from [6]. We argue that the additional traffic that passes through Nonesuch provides extra anonymity and outline one procedure which the adversary can use to estimate the number of real messages passing inside the mix.

7. ACKNOWLEDGEMENTS

We thank Boris Margolin for having asked us the questions that Nonesuch was eventually developed to answer. His conversation was invaluable in motivating this work. We thank George Danezis for co-inventing the Minx packet format, and then travelling to the University of Massachusetts to share it with us. We thank Brian Levine for advising an earlier version of Nonesuch, and Kevin Fu for his support and writing advice. Finally we thank WPES anonymous reviewers for their exceptionally thorough and helpful comments.

8. REFERENCES

[1] Bauer, M. New covert channels in http: adding unwitting web browsers to anonymity sets. In *WPES '03: Proceedings of the 2003 ACM workshop on Privacy in the electronic society* (New York, NY, USA, 2003), ACM Press, pp. 72–78.

[2] Berman, R., Fiat, A., and Ta-Shma, A. Provable unlinkability against traffic analysis. In *Proceedings of Financial Cryptography (FC '04)* (February 2004), A. Juels, Ed., Springer-Verlag, LNCS 3110.

[3] Chaum, D. The dining cryptographers problem: unconditional sender and recipient untraceability. *J. Cryptol. 1*, 1 (1988), 65–75.

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

[5] Danezis, G., Dingledine, R., and Mathewson, N. Mixminion: Design of a Type III Anonymous Remailer Protocol. In *Proceedings of the 2003 IEEE Symposium on Security and Privacy* (May 2003).

[6] Danezis, G., and Laurie, B. Minx: A simple and efficient anonymous packet format. In *Proceedings of the Workshop on Privacy in the Electronic Society (WPES 2004)* (Washington, DC, USA, October 2004).

[7] Díaz, C. *Anonymity and Privacy in Electronic Services*. PhD thesis, Katholieke Universiteit Leuven, Leuven, Belgium, December 2005.

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

[9] Hopper, N. J., Langford, J., and von Ahn, L. Provably secure steganography. In *CRYPTO '02: Proceedings of the 22nd Annual International Cryptology Conference on Advances in Cryptology* (London, UK, 2002), Springer-Verlag, pp. 77–92.

[10] Horton, M., and Adams, R. Standard for interchange of USENET messages. RFC 1036, Dec. 1987.

[11] Kantor, B., and Lapsley, P. Network News Transfer Protocol. RFC 977 (Proposed Standard), Feb. 1986.

[12] Kesdogan, D., Egner, J., and Büschkes, R. Stop-and-Go MIXes: Providing Probabilistic Anonymity in an Open System. In *Proceedings of Information Hiding Workshop (IH 1998)* (1998), Springer-Verlag, LNCS 1525.

[13] Levine, B. N., Reiter, M. K., Wang, C., and Wright, M. K. Timing attacks in low-latency mix-based systems. In *Proceedings of Financial*

Cryptography (FC '04) (February 2004), A. Juels, Ed., Springer-Verlag, LNCS 3110.

[14] Mixminion Stats V1.0. Statistics generated by the remailer grove. WWW, June 2006. http://privacy.outel.org/minion/nlist.html.

[15] Serjantov, A. *On the Anonymity of Anonymity Systems*. PhD thesis, University of Cambridge, June 2004.

[16] Serjantov, A., and Danezis, G. Towards an information theoretic metric for anonymity. In *Proceedings of Privacy Enhancing Technologies Workshop (PET 2002)* (April 2002), R. Dingledine and P. Syverson, Eds., Springer-Verlag, LNCS 2482.

[17] Syverson, P., Tsudik, G., Reed, M., and Landwehr, C. Towards an Analysis of Onion Routing Security. In *Proceedings of Designing Privacy Enhancing Technologies: Workshop on Design Issues in Anonymity and Unobservability* (July 2000), H. Federrath, Ed., Springer-Verlag, LNCS 2009, pp. 96–114.

[18] von Ahn, L., and Hopper, N. J. Public-key steganography. In *EUROCRYPT* (2004), C. Cachin and J. Camenisch, Eds., vol. 3027 of *Lecture Notes in Computer Science*, Springer, pp. 323–341.