Peer to Peer Networks

Jan H. Knudsen (20092926) Roland L. Pedersen (20092817)
Kris V. Ebbesen (20094539)

May 5, 2014

Contents

1 A short note on cryptography

The project described in this report relies heavily on cryptographic system and practices in order to be possible. It is however the case that none of the persons working on the project have any previous experiences working with secure systems, since we come from backgrounds in algorithms and computer graphics. We have chosen such a heavy reliance on systems outside our normal line of work as a learning experience, and with the strong conviction that most cryptographic systems work well as black boxes.

Expanding further on this, when we refer to a cryptographic method, we will rarely expand on the inner workings of such components, but rather rely on their security as provided by their developers. Of course this makes us somewhat prone to making errors that would be considered mistakes by hardened security veterans, but we beg forgiveness for our bright-eyed naïveté.

In the end, what matters to us in this project, is the parts that relate to peer-to-peer systems.

2 Method of Operation

2.1 Base System

The system described is built atop the unstructured network developed during the P2PN course (TODO: ref earlier P2PN report). This network contains very little structural information, and bases its topology on the GIA network (Chawathe et al., 2003).

The choice of this network was made based on its simplicity and extendibility, and due to the fact that unstructured networks require little information about the peers involved, making it difficult the track which peers are doing what.

Note that the techniques used to extend the network could be applied to most unstructured networks, and would probably work just as well on the GIA network.

2.2 Encrypting Messages and Hiding Recipients

In order to ensure that no adversaries can divulge the content of any given message, we encrypt chat messages travelling across the network using RSA-OAEP (Bellare and Rogaway, 1995). RSA-OAEP was chosen due to its ease of use, and security against repeated plaintext attacks.

When performing this encryption, we use a pair of RSA (TODO:Mayby reference?) keys. The sender must obtain the public key of the final recipient (how to do this will be explained later), in order to encrypt the message.

When the chat message is sent, it is first encrypted by RSA-OAEP using the public key of the recipient, and then broadcast across the network using either flooding or k-walkers. Whenever a peer receives a messages travelling across the network, it will attempt to decrypt it using the corresponding RSA-OAEP decrypting using its own private key. This will fail for all peers except the recipient, ensuring that only the final recipient will be able to obtain the contents of the chat message.

Note that the encrypted message sent across the network contains no delivery address of any kind, and as such no other peers will know the final recipient.

It is also worth noting that only one RSA key pair is required to send messages. The sender needs no private key, nor do any other peers in the network.

2.3 Signed Messages and Acknowledgement

All chat messages in the system may or may not be signed by the sender. If the sender wishes to not sign his messages, in order to hide their identity to the receiver, or because he is not in possession of a private key, he may omit this signature. Additionally, any message received by a peer can be acknowledged by returning a signed digest of the message.

Both types of signatures are done according to **PKCS#1 v1.5** (TODO:Some sort of reference).

In the case of the sender signing a message, we send a signature of the plain-text message along with encrypted message. This ensures, that only after obtaining the decrypted message will it be possible to verify the signature. This ensures that we keep identity of the sender hidden to anyone except the recipient, and that the recipient can securely verify the sender given his public key. Also, should anyone attempt to tamper with the message before delivery, the signature will no longer be valid.

When verifying the delivery of a message the receiver returns a signed digest of the plaintext message, which is verified by the sender. This ensures that the sender has received the message, as he is the only one able to provide a valid signature. If the peer is using flooding we simply return this value as part of the xml-rpc call, while we answer back using a k-walker in the case that we receive a message by k-walker. Given a small random delay, it become difficult to determine whether a message was received by any given peer, or one of his neighbours. Also, should anyone tamper with the message before delivery, the receiver will no longer sign the correct data, making this easily detectable for the sender.

2.4 Encrypting Peer Communication

All traffic between peers in the network is encrypted using anonymous Diffie-Hellman (Diffie and Hellman, 1976) encryption. This encryption is provided by wrapping connections between peers in an SSL layer, with no certificates and anonymous Diffie-Hellman as the only cipher set.

This ensures that peers can communicate without outside parties snooping on the information, therefore, will make it very hard to track messages across the network, since the data sent from messages, cover traffic, and general networks operations will be indistinguishable.

Another reason to use anonymous Diffie-Hellman encryption is that it enforces no requirements on previously distributed keys or identities of the peers, keeping each peer's knowledge about its neighbours at a minimum.

In order to prevent constant Diffie-Hellman key renegotiations we provide cached pools of SSL connections, meaning that we only create a new connection when the peer runs out of idle connections to the same peer.

References

Yatin Chawathe, Sylvia Ratnasamy, Lee Breslau, Nick Lanham, and Scott Shenker. Making gnutella-like p2p systems scalable. In *Proceedings of the 2003 Conference on Applications, Technologies, Architectures, and Protocols for Computer Communications*, SIGCOMM '03, pages 407–418, New York, NY, USA, 2003. ACM. ISBN 1-58113-735-4. doi: 10.1145/863955.864000. URL `http://doi.acm.org/10.1145/863955.864000`.

Mihir Bellare and Phillip Rogaway. Optimal asymmetric encryption. In Alfredo Santis, editor, *Advances in Cryptology - EUROCRYPT 94*, volume 950 of *Lecture Notes in Computer Science*, pages 92–111. Springer Berlin Heidelberg, 1995. ISBN 978-3-540-60176-0. doi: 10.1007/BFb0053428. URL `http://dx.doi.org/10.1007/BFb0053428`.

Whitfield Diffie and Martin E Hellman. New directions in cryptography. *Information Theory, IEEE Transactions on*, 22(6):644–654, 1976.