Secure and anonymous communication in the cloud

Risto Laurikainen
Aalto University School of Science and Technology
rlaurika@cc.hut.fi

Abstract

Many communication systems provide confidentiality, integrity and availability to communicating parties, yet few can also provide anonymous communication in a way that allows communicating parties to authenticate each other while still preventing third parties from discovering the end points of a particular communication.

Cloud based anonymizing systems provide one solution to the anonymity problem, but the existing systems are not designed with confidentiality of the communicating parties in mind. It is possible to combine anonymizing systems with systems that provide confidentiality, but such combinations may have unforeseen security issues that may compromise anonymity.

This paper establishes a list of requirements for a secure and anonymous communication system and then looks at how existing systems can be used to fulfill those requirements. Finally cloud based solutions to the problem will be examined generally.

1 Introduction

Secure and anonymous communication has many applications. Dissidents living in totalitarian states need to communicate without fear of retribution. Companies need to protect their trade secrets from industrial espionage. Whistleblowers need to be able to communicate with the press without fear of punishment. Freedom of expression sometimes needs anonymity to survive attacks mounted against it.

Confidentiality, integrity and availability are requirements for any secure communication system. No system can fulfill all three requirements in all circumstances, but the probability of a succesful attack that compromises any of the three should be minimal.

The confidentiality requirement means that a message's content must remain secret between its sender and its intended recipient. The content must therefore be protected in some way against unauthorized readers. This is usually done using some type of encryption, but ciphering alone is insufficient: the communicating parties must also authenticate each other to make sure the message actually comes from the correct source and reaches the correct destination. Without an authentication mechanism it is possible for a man in the middle to intercept the message by posing as the recipient to the sender and as the sender to the recipient, rendering any encryption useless.

Local law enforcement requirements may also pose challenges to communications privacy that are not directly linked to encryption and authentication. A recent example of this is the disabling of certain features of Research in Motion's BlackBerry phone in the United Arab Emirates due to concerns over the difficulty of monitoring communications between the users of the devices[15]. Another example of recent developments is a bill proposed in the United States that would mandate all communication systems – including those that use encryption – to be able to comply with a wiretap order[20]. The infrastructure required to comply with such orders has been shown to be impractical to implement in a secure and cost-effective way[3]. While there are legitimate reasons for signal intelligence, there are also legitimate reasons for messages between two parties to remain between those two parties only.

Maintaining availability is important. If an adversary can disable a secure means of communication, it may be able to force the communication to a less secure channel. In order to maintain availability, a communication system should not have any single point of failure. The point of failure can be technical, such as reliance on a single server. In such a case a malfunction in the server or a denial of service attack against it can disable the system. It can also be organizational, in which case a single entity has the power to disable the system. This entity is usually the organization that runs the system, but it can also be a government entity as was the case in the BlackBerry ban in the United Arab Emirates. Fully decentralized systems do not have this vulnerability, but attacks involving multiple hostile nodes may still be able to disable them[12]. For a system to remain secure in an untrusted environment, it must be resilient against attacks that aim to either disable or compromise it.

Anonymity is also important. The mere knowledge of two parties having exchanged messages may be compromising even if the content of the correspondence remained secret. However, neither party of the message exchange should be anonymous to the other party, as is the case in some systems[8].

The previous paragraphs have listed many requirements for a secure and anonymous communication system. These requirements are summarized in the following list:

- The system must use cryptographically strong encryption to protect the content of messages.
- The communication parties must have the ability to authenticate each other using a cryptographically strong authentication scheme.
- The system must have a robust and secure procedure to manage and exchange encryption keys and codes.

- There must be no single point of failure in the system, whether it be organizational or technical.
- The system must be resilient to attacks against its infrastructure that aim to compromise availability.
- The system must have the ability to hide the identities of communication participants from third parties.

Section 2 examines a system that is able to fulfill the requirements listed above and breaks it into three different components. Each component fulfills a subset of the requirements listed above. This division is due to the abilities of existing systems, many of which can provide the functionalities of only one or two of these components. We also take a look at existing systems that offer communication with varying degrees and types of security features. Section 3 examines how each of the previously listed requirements can be fulfilled in a cloud environment.

2 Possible solutions

The components of a secure and anonymous communication system can roughly be divided into the following three categories:

- Encryption and authentication service for maintaining confidentiality and integrity
- Message delivery system that needs to be resilient enough to maintain availability and integrity
- Anonymizing system that hides the identities of communicating parties while still allowing them to authenticate each other.

Some systems can provide more than one of these components, but few systems have all three. Combining multiple systems each of which can provide one or more of the features is one possible solution, but such combinations may introduce unforeseen security issues. One example is the watermark attack on anonymized Skype calls described by Wang et al.[22] and mentioned in section 2.1.2. Predicting and finding security issues such as the one described by Wang et al. is something the end user cannot be expected to do, which makes combining multiple systems problematic.

Further in this section we will look at some examples of systems categorized by the functions they fulfill in the complete system.

2.1 Examples of existing systems

2.1.1 Encryption/authentication: OpenPGP

OpenPGP[1][7] is a protocol for encrypting email. It defines standard formats for signatures, certificates and encrypted messages. Its implementations include PGP, which is the original source of the protocol and GnuPG, which is an open source implementation. It can be used in conjunction with an anonymizing service and a message delivery system to provide secure and anonymous communication.

The basis of OpenPGP is public key cryptography, in which each entity is assigned a public key and a private key. The public key is known to all while only the owner of a key pair knows the private key. A message encrypted using the public key can be opened using the private key and vice versa. This way, messages can be sent confidentially to entities by using the public key of an entity. These messages can then only be opened using the corresponding private key.

Another operation is signing, which is used to prove that a given message did in fact originate from the holder of a particular key pair. The private key is used to encrypt some identifiable piece of data. If decryption using the public key yields the correct data, this proves that the encrypting entity had to have had access to the private key associated with that public key.

2.1.2 Message delivery, encryption/authentication: Skype

Skype[2] is a proprietary Voice over IP -client. Its architecture is mostly decentralized, although there is a separate login server for authentication purposes and to ensure the uniqueness of login names in the network[4]. Communication between clients is done in a peer-to-peer fashion using an overlay network divided into two layers: some clients act as supernodes that communicate to other supernodes, while most clients are regular nodes that communicate to other clients through supernode intermediaries[14].

The proprietary nature of the software restricts independent scrutiny of its security features. The company behind Skype has commisioned at least one third party security evaluation whose author had access to the source code of the program[6]. However, as the author notes in the paper, the code base has since evolved, and current versions of the software may be vastly different. According to the paper, Skype uses standard cryptographic primitives: AES for encrypting data in an established session, RSA for authentication, SHA-1 for hashing passwords and in random number generation and the ISO 9796-2 signature padding scheme. As is noted in the paper, this is a sound practice from a security perspective. However, SHA-1 may no longer provide adequate security[23]. While newer versions of Skype may no longer use SHA-1, the proprietary nature of the software prevents users from finding out if this is the case.

Skype itself does not provide an anonymous service and must be used in conjuction with an anonymizing system. Even if such a system is used, calls may still be traceable: Wang et al.[22] describe a watermarking attack that allows an adversary that has access to the Skype flow at both the caller's and the callee's end to know when the caller and callee are communicating with each other. The attack requires an ability to introduce delays to the packets, which can then be observed at the other end of the call. While the anonymizing system may have worked correctly with other types of traffic, the nature of the Skype flow allowed the attack to compromise the anonymity of the call. This is because the Skype conversation contained enough data to allow watermarking of the packets in the flow. According to the paper, it is possible to compromise the anonymity of calls that

[1] `http://www.openpgp.org`

[2] `http://www.skype.com`

last 90 seconds or longer. Other more space efficient forms of communication such as email would not be susceptible to this attack. It should be noted that Skype also contains a text based instant messaging capability, which would probably be safe from this attack.

2.1.3 Message delivery, anonymizing service: Freenet

Freenet[3][9] is a distributed data storage system that promises to protect the identity of both the publisher and the reader of a given piece of information stored in the system. It is designed for one-to-many communication instead of one-to-one communication, but it can still be used for the latter purpose according to the Freenet Frequently Asked Questions: *"Freenet is designed to make communication possible even if there's just one publisher and one reader, and this is already reasonably feasible on the current freenet."*[2] One-to-one communication has also been implemented in practice in the form of Freemail, which is a plugin for the Freenet program[1].

Freenet is completely decentralized. Data is stored on the client machines and there is no centralized index of the files stored in the system. Each file is given a hash identifier based on its name, and a request for a specific file will go through multiple nodes. Each node on the path will redirect the request to another node that it thinks is most likely to hold the file. Once a node holding the file is found, the file is sent along the request path and is stored on each of the nodes on that path.[9]

Files are stored in encrypted form in the system, but this is only to provide plausible deniability for the storing nodes[9]. Nodes storing files have no way of knowing their contents, but anyone searching for a given file will still be able to open it. This is becaused the files are encrypted using their original descriptive text string as the key: anyone with knowledge of the string will be able to open the file, but storing nodes do not know the descriptive strings of the files they store[9]. This scheme does not provide confidentiality. It is possible to pre-encrypt files before storing them in the system, but the system itself provides no means of key exchange or authentication for end-to-end encryption between the sender and the recipient in a one-to-one communication. These necessary functions would have to be provided by some other system.

Another problem with using Freenet for one-to-one communication with end-to-end encryption is that it does not provide file lifetime guarantees. A node will delete files when it runs out of space starting with its least popular files, which means that files that have not been requested for some time will eventually disappear[9].

2.1.4 Anonymizing service: Tor

Tor[4] is a *"circuit-based low-latency anonymous communication service"*[11] that routes the user's messages through a random circuit of nodes to hide the origin of those messages. Onion routing[13] is the key concept behind Tor, although various changes and improvements – such as perfect forward secrecy – have been made to the design of earlier systems[11]. In onion routing, messages are encrypted multiple times in layers using the public keys of the nodes on the message's circuit. Each router can then open one layer of the encryption to find out the next hop on the circuit of the message. This way no node on the circuit knows the complete circuit – only the previous and the next node.

The purpose of the encryption in Tor is to make traffic analysis more difficult. It does not provide confidentiality, since the last node on the path will see the plain text of the message if nothing is done to disguise the message beforehand. If confidentiality is needed, another system must be used to provide end-to-end encryption.

Any user running the Tor client can act as a Tor router. However, the network is not completely decentralized, as it relies on a small set of well known directory servers run by independent parties that provide clients with information about the trusted nodes in the system. One of the functions of these servers is to restrict the introduction of too many malicious nodes into the system.[11]

The Tor network is a compromise between perfect anonymity and high performance[11]. One of the features it has for ensuring low latencies is favoring those nodes that have the highest uptimes and the most bandwidth. This feature has been exploited by attacks where malicious nodes exaggerate their resources and reliability, thus making them more likely to be selected to relay data on a given circuit[5]. If an attack can compromise both the entry and the exit node on a circuit, then that circuit will no longer provide perfect anonymity[5, 11]. These and other issues are being worked on[10], but Tor may not yet be mature enough to provide strong anonymity.

3 Cloud based systems

The National Institute of Standards and Technology gives the following definition to cloud computing:

> "Cloud computing is a model for enabling convenient, on-demand network access to a shared pool of configurable computing resources (e.g., networks, servers, storage, applications, and services) that can be rapidly provisioned and released with minimal management effort or service provider interaction."[18]

According to this definition, all of the systems described in section 2.1 except for OpenPGP incorporate architectural features that are similar to those in cloud based systems. These systems all use the shared resources of the peers that form the systems.

It seems that cloud based architectures are especially well suited for anonymizing traffic. Tor and Freenet both use a cloud architecture, and the ways in which they anonymize their users are very similar. These systems are also either completely decentralized, as is the case with Freenet, or mostly decentralized, as is the case with Tor. This makes them resistant to denial of service attacks, as there is no single point of failure.

While the properties of cloud systems are well suited for providing anonymous message delivery, authentication and

[3] `http://freenetproject.org/`

[4] `http://www.torproject.org/`

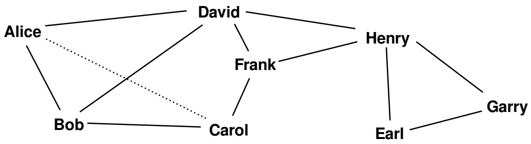

Figure 1: The web of trust model of PGP. A chain of trust is created between Alice and Carol through Bob.

encryption need to be provided through some other means. Public key cryptography offers both, but the communicating parties must first have some reliable way to acquire each other's public keys. To achieve this, some type of public key infrastructure is needed. The main problem is establishing trust: how does the recipient of a key know if that key actually belongs to the right person? Perlman gives an overview of different public key infrastructure trust models[19]. Many of these models involve some type of centralized certificate authority (CA) that verifies the identity of other entities. However, as established earlier in section 1, this would introduce a single point of failure into the system – the private key of the CA. If this key were to be compromised, the identities of all users in the system would also be compromised. Given the decentralized nature of the cloud, the best solution would be to also decentralize the process of establishing trust. This way there would be no single point of failure.

One way of establishing trust in a decentralized manner is PGP's web of trust[24] model illustrated in Figure 1. In it there is no centralized authority, and users act as trusted entities that can vouch for their own public key or the public keys of others. This creates a graph where trust can be established by following the edges on that graph. If, say, Alice trusts that she has Bob's correct public key and Bob trusts that he has Carol's correct public key, then Alice can obtain Carol's public key through Bob. In other words, a chain of trust is created between Alice and Carol. This type of an arrangement works well for small groups of people. In such groups, members usually have the ability to reliably verify each other's identities by meeting each other. However, this arrangement does not scale well to larger groups, where long chains of trust have questionable trustworthiness[19]. With large geographical separations between the group members, even small groups may have difficulties in verifying each other's public keys.

A better method for establishing trust would be one that could leverage the resources in the cloud so that the users would not need to use ad-hoc ways to verify each other's public keys. The optimal solution would be to store the user's public keys in a distributed manner inside the cloud without any governing entity. This way, user's could easily obtain each other's keys. There are proposed systems that distribute the certification task among peers[17, 16]. These systems are based on distributing the secret key used to sign certificates among peers so that no one has the complete secret key. The mechanism for this distribution has originally been described by Shamir[21]. While these systems can reliably authenticate entities for whom certificates already exist, issuing new certificates can be a problem. Obtaining a certificate under a false identity should be prevented in some way. Kong et al.[16] suggest some type of physical proof to establish identity. This approach to issuing new certificates has the same problems of scalability that are also present in the web of trust scheme.

4 Conclusion

There are many systems that allow end-to-end encrypted communications between users once the users have authenticated each other. These systems offer confidentiality through authentication and strong encryption, but they do not anonymize their users.

On the other hand, there are many systems that allow users to anonymize themselves, but this often means that they are anonymous to everyone including the other party of the communication instead of only being anonymous to third parties. Another problem with using these anonymizing systems is that they do not provide confidentiality.

When combined with an anonymizing system, public key cryptography offers one solution for providing confidentiality and anonymity: users can send messages to each other through the anonymizing system encrypted using the public key of the intended recipient. If the messages also include a cryptographic signature of the sender, then the recipient can be sure that the messages came from the correct source. Even if the message ends up in the wrong hands, there is no way to determine its contents and, because of the anonymizing network, no way to determine its source. The problem with this system is that user's must first have some way to reliably exchange public keys.

Cloud based systems are a good way to provide the anonymizing component of the secure and anonymous communication system. They can offer high availability through replication of resources and the lack of a single point of failure gives them resilience to attacks. Tor and Freenet are existing systems that use a cloud architecture. However, cloud systems do not currently offer any new solutions to the authentication problem. Using a centralized authentication scheme would mean the loss of some of the advatages given by the decentralization. The web of trust introduced by PGP would not be able to function solely in the cloud and would require users to verify their public keys in some ad-hoc way. There are ways to distribute the roles of a certificate authority among many nodes, but widely available implementations are not available.

It is possible to communicate in a secure and anonymous fashion by using a combination of widely available programs, each of which provides a subset of the required features. However, users may not have the knowledge required to find and configure such a combination in a truly secure way. It would be beneficial if all the features required for secure and anonymous communication could be found in a single program.

References

[1] Freemail. Cited: 18.10.2010, Available: `http://freenetproject.org/freemail.html`.

[2] Freenet frequently asked questions. Cited: 18.10.2010, Available: `http://freenetproject.org/faq.html`.

[3] H. Abelson, R. Anderson, S. M. Bellovin, J. Benalob, M. Blaze, W. Diffie, J. Gilmore, P. G. Neumann, R. L. Rivest, J. I. Schiller, and B. Schneier. The risks of key recovery, key escrow, and trusted third-party encryption. *Center for Democracy & Technology Digital Issues*, (3), June 1998.

[4] S. Baset and H. Schulzrinne. An analysis of the Skype peer-to-peer internet telephony protocol. In *INFOCOM 2006. 25th IEEE International Conference on Computer Communications. Proceedings*, Barcelona, Spain, April 2006.

[5] K. Bauer, D. McCoy, D. Grunwald, T. Kohno, and D. Sicker. Low-resource routing attacks against Tor. In *WPES '07: Proceedings of the 2007 ACM workshop on Privacy in electronic society*, pages 11–20, New York, NY, USA, 2007. ACM.

[6] T. Berson. Skype security evaluation, Anagram Laboratories, October 2005.

[7] J. Callas, L. Donnerhacke, H. Finney, D. Shaw, and R. Thayer. OpenPGP Message Format. RFC 4880 (Proposed Standard), Nov. 2007. Updated by RFC 5581.

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

[9] I. Clarke, O. Sandberg, B. Wiley, and T. Hong. Freenet: A distributed anonymous information storage and retrieval system. In H. Federrath, editor, *Designing Privacy Enhancing Technologies*, volume 2009 of *Lecture Notes in Computer Science*, pages 46–66. Springer Berlin / Heidelberg, 2001. 10.1007/3-540-44702-4_4.

[10] R. Dingledine. Tor development roadmap, 2008-2011. Technical report, The Tor Project, 2008.

[11] R. Dingledine, N. Mathewson, and P. Syverson. Tor: the second-generation onion router. In *SSYM'04: Proceedings of the 13th conference on USENIX Security Symposium*, Berkeley, CA, USA, 2004. USENIX Association.

[12] J. Douceur. The Sybil attack. In P. Druschel, F. Kaashoek, and A. Rowstron, editors, *Peer-to-Peer Systems*, volume 2429 of *Lecture Notes in Computer Science*, pages 251–260. Springer Berlin / Heidelberg, 2002. 10.1007/3-540-45748-8_24.

[13] D. Goldschlag, M. Reed, and P. Syverson. Onion routing. *Commun. ACM*, 42(2):39–41, 1999.

[14] S. Guha and N. Daswani. An experimental study of the Skype peer-to-peer VoIP system. 2005.

[15] M. Kiselyova, D. Tripathy, and T. Virki. Analysis: Secure data blessing and curse for BlackBerry. `http://www.reuters.com/article/idUSTRE6713I820100802`, August 1, 2010.

[16] J. Kong, P. Zerfos, H. Luo, S. Lu, and L. Zhang. Providing robust and ubiquitous security support for mobile ad-hoc networks. In *ICNP 2001: Ninth International Conference on Network Protocols*, 2001.

[17] F. Lesueur, L. Mé, and V. V. T. Tong. An efficient distributed PKI for structured P2P networks. In *P2P '09: IEEE Ninth International Conference on Peer-to-Peer Computing*, 2009.

[18] P. Mell and T. Grance. The NIST definition of cloud computing. Technical report, National Institute of Standards and Technology, 2009.

[19] R. Perlman. An overview of PKI trust models. *IEEE Network*, 13(6):38–43, 1999.

[20] C. Savage. U.S. tries to make it easier to wiretap the internet. `http://www.nytimes.com/2010/09/27/us/27wiretap.html`, September 27, 2010.

[21] A. Shamir. How to share a secret. *Commun. ACM*, 22(11):612–613, 1979.

[22] X. Wang, S. Chen, and S. Jajodia. Tracking anonymous peer-to-peer VoIP calls on the internet. In *CCS '05: Proceedings of the 12th ACM conference on Computer and communications security*, pages 81–91, New York, NY, USA, 2005. ACM.

[23] X. Wang, Y. L. Yin, and H. Yu. Finding collisions in the full SHA-1. In V. Shoup, editor, *Advances in Cryptology - CRYPTO 2005*, volume 3621 of *Lecture Notes in Computer Science*, pages 17–36. Springer Berlin / Heidelberg, 2005. 10.1007/11535218_2.

[24] P. R. Zimmermann. *The official PGP user's guide*. MIT Press, Cambridge, MA, USA, 1995.