Solutions for Anonymous Communication on the Internet

Joris Claessens[1] Bart Preneel[2] Joos Vandewalle

K.U.Leuven ESAT-COSIC
Kardinaal Mercierlaan 94, B-3001 Heverlee, Belgium
Tel: +32 16 32 1853 – Fax: +32 16 32 1969
joris.claessens@esat.kuleuven.ac.be
http://www.esat.kuleuven.ac.be/~joclaess/

Abstract

This paper gives a short overview on the currently existing technical solutions for anonymous communication. The problem of anonymous communication is defined, and its basic solution is described. Practical solutions, mostly based on the basic scheme, are discussed. They provide anonymity to connections in general, and anonymity in specific applications, like email and the World Wide Web. The different solutions are described, and a comparison is given. Additional remarks are made with respect to anonymity revocation, U.S. export restrictions, and the performance that can be obtained.

Keywords: anonymous communication, privacy, cryptography

1 Introduction

The Internet is used by more and more people for personal and business related communication. More and more traditional human interactions have been translated into an electronic equivalent: messaging, voting, payments, commerce, etc. Communications and data security are very important aspects in this area. Much effort has therefore already been done to achieve user/data authentication, data confidentiality, etc, in these electronic interactions.

Just as in the real world, the user's privacy should also be ensured. Solutions that provide anonymous communication in the electronic world are thus needed. In this way, it is possible that the sender's identity is not revealed to the recipients of a message, and that other people on the network cannot trace who is communicating with whom.

This paper intends to discuss technical solutions for providing anonymous communication. Privacy legislation is currently developed as well, and is a legal solution. However, making it technically impossible to trace real identities on the Internet, protects the user's privacy even more.

Providing anonymous communication on the Internet is not trivial as most applications reveal a lot of information on the user's identity. A simple example are the e-mail headers that contain the sender's and the recipient's e-mail address. More fundamentally, network addresses are required for establishing a communication between two parties. Normally, these network addresses give already enough information to reveal sender's and recipient's identity.

Offering a technical solution for anonymous communication can have disadvantages too. In some applications, anonymity can be misused by criminals or people with malicious intentions. Examples are spam e-mail and money laundering. It is therefore sometimes required that it is technically possible to revoke anonymity if requested by e.g. a judge or other legal entity.

The paper starts with a definition of anonymous communication, and an explaination of why it is difficult to remain anonymous during a normal Internet communication. The basic solution to the problem is then described. In Sections 4, 5, and 6, this basic solution is used to provide anonymous communication to Internet connections in general, e-mail, and the World Wide Web (both browsing and publishing), respectively. A rough comparison of the different described systems is then presented. Anonymity in other applications, but in payment and voting schemes in particular, is discussed in Section 8. In Section 9, some additional remarks are made, concerning revocation, the U.S. export restrictions, and performance. Finally, a conclusion is given.

[1] Funded by a research grant of the Flemish Institute for the Promotion of Industrial Scientific and Technological Research (IWT)

[2] F.W.O. postdoctoral researcher, sponsored by the Fund for Scientific Research – Flanders

Appeared in *Proceedings of the IEEE 33rd Annual 1999 International Carnahan Conference on Security Technology*
ICCST'99, Madrid, Spain, October 5-7, 1999, pp 298–303
0-7803-5247-5/99/$10.00

2 Anonymity

Anonymous communication can roughly be defined as follows: Alice communicates anonymously with Bob, if Bob does not know Alice's identity (unless Bob has more information, obtained in another way), and if Alice's and Bob's communication cannot be linked together by someone who has an overview on the global network (a third person Eve, or Bob itself). Sometimes Bob is allowed to know Alice's identity (though not obtained from the observable communication), and both Alice and Bob want to hide their communication from outside observers.

Today's communication over the Internet is not anonymous. The communicating parties' identies can be revealed in two ways.

The application itself can disclose information on the sender's identity. When browsing the WWW for example, the headers of the HTTP protocol contain identifying information (Cookies, Referrer, User-Agent, etc). Also, on a lot of web sites, users have to login before obtaining certain services. E-mail is an even more obvious application that reveals the identities, as they are in the From and To fields of the message.

More importantly, at the network level, network addresses and host names are known during the communication. In many cases, these can be linked to a limited set of persons, if not one person.

Basic solutions for anonymous communication will protect against local observers. More advanced solutions will have to protect against powerful observers, who are able to overview the whole network. These solutions should protect against eavesdropping, not for confidentiality purposes, but for preventing the content of the messages to be traced from destination to source. They should also protect against traffic analysis, to prevent messages to be traced based on size and timing measurements.

Solutions that protect against revealing the identity at the application level, provide *data anonymity*, while protection at the network level, means providing *connection anonymity*.

3 Basic solution

The basic building block for an anonymous communication solution was presented by Chaum [2]. This basic building block is called a *mix*. The messages of all parties wanting to communicate anonymously are sent through the mix. The mix hides the correspondences between messages in its input and those in its output. The mix hides the order of arrival of the messages by reordering, delaying and padding traffic.

As an example, in order for A to anonymously send the message m to B, A sends the message $K_M(R_1, K_B(R_0, m), B)$ to the mix M, in which K_x means encrypted with the public key of x, and in which R_i are random values. The mix decrypts the message, delays it, reorders it with other messages, and finally sends the message $K_B(R_0, m)$ to B.

A single perfect mix adequately complicates traffic analysis. As real mixes are not ideal, a sequence of mixes is typically used in practice. An attacker might replay messages, causing the same input at the mix, and hope that the same output will occur too, and thus learn the destination of a specific message. A mix must therefore ensure that a same message is not processed more than once.

The mix system is actually the underlying solution of most of the systems described in the following sections.

4 Anonymous connections

The **Onion Routing** system [15] is a solution for anonymous connections, independent from the actual application that is used.

The network consists of a number of *onion routers*, which have the functionality of ordinary routers, combined with mixing properties. Data is sent through a path of onion routers. This path is determined by an onion. Senders will have to connect to a particular onion router via an application proxy and an onion proxy. Anonymity is only provided from the first to the last onion router. The connections from sender to first onion router, and from last onion router to receiver, are not anonymity protected (for companies, it is therefore recommended to have one onion router at the border of their own corporate network).

The core of the solution is the *onion*. An onion is a layered data structure, that is sent to an onion router. It is encrypted with the public key of the onion router to which the onion is sent. It defines the route of an anonymous connection. It contains the next hop information (the next onion router or the final recipient), key seed material for generating the symmetric keys that will be used by the onion router during the actual routing of the data, and an embedded onion that is sent to the next onion router.

When a sender wants to communicate anonymously with a particular receiver, an *anonymous connection* has to be setup. The onion proxy therefore prepares an onion that is sent to a first onion router. This onion router decrypts the onion, obtains the next hop information and key seed material, and sends the embedded onion to the next onion router. As such, an anonymous connection, identified by an ACI (Anonymous Connection Identifier) is setup.

Data can then be transmitted over this anonymous connection. The application proxy first anonymizes the data stream coming from the application (and as such provides data anonymity). Before sending the data through the onion router network (connection anonymity), it is then encrypted

Appeared in *Proceedings of the IEEE 33rd Annual 1999 International Carnahan Conference on Security Technology*
ICCST'99, Madrid, Spain, October 5-7, 1999, pp 298–303
0-7803-5247-5/99/$10.00

multiple times using the symmetric keys that were distributed to all the onion routers on the path. The data is carried by small data cells containing the appropriate ACI. Each onion router removes/adds a layer of encryption (using the symmetric keys, generated from the key seed material in the onion) depending on the direction of the data (forwards/backwards).

Many other anonymous connections will exist at the same time. Observers will see messages flowing through the onion router network, but will (in theory) not see who is communicating with whom. The onion routing system resists traffic analysis as data appears different at each onion router. Onion routers are also entry points, and traffic entering or exiting at those nodes may not be visible, making tracking packets again harder. Onion routers check replayed or expired onions.

On the negative side, the system is intended for real-time and bidirectional communication, which limits the possibility to delay traffic, and which decreases the mixing properties of onion routers. As onions are forwarded along the same path which later on forms the anonymous connection, they should be shuffled and delayed as well.

PipeNet [5] is a similar, but more robust system. In the Onion Routing system, an attacker could delay data at one input of an onion router, causing a delay at a specific output of that onion router, and therefore disclosing part of the anonymous connection. The PipeNet switches on the other hand will detect this disruptions, and delay all other connections through that switch at the same time. Padding data is added to maintain constant data flow over all connections, making traffic analysis very hard. There is however no indication of an actual implementation of this system.

The **Freedom Network** Architecture [17] has also the strong PipeNet properties. Besides the implementation of IP based anonymous connections, an interface for e-mail and WWW is also provided (see next sections). A traffic shaper manages possible bandwidth requirements.

5 Anonymous e-mail

While the previous section described solutions for real-time, bidirectional, anonymous communication, this section describes solutions for a connectionless Internet application, in particular anonymous email.

A first obvious and easy way to obtain a certain level of anonymity, is using a **pseudonymous account**, e.g., `anonuser@email.com`. An ordinary recipient cannot figure out the real name of the sender. A more powerful observer can however easily trace the message through the network, especially if the IP address of the sender is still visible in the message. Pseudonymous accounts can be used in combination with the following anonymous e-mail solutions.

The Pseudonymous server or **Type 0 remailer**, is a simple system, which just strips identifying headers, and forwards the message to the intended recipient. As the message is sent via an intermediate server, the originator's IP address is not revealed to the recipient. This does however not protect against someone who can observe the communication from the sender to the pseudonymous server. Examples can be found at [11] (the service is closed because of spam problems) and [12].

A higher level of anonymity can be obtained with the **Type 1 remailer** [4]. In this system, an e-mail message consists of a nested set of encrypted messages (conceptually, the same structure as the onion structure in the previous section). The message is sent through a path of mixes. A mix decrypts the message, and obtains a number of instructions, and an embedded encrypted message. These instructions include for example how long the message should be delayed. The mix is also instructed to forward the embedded message to the given next mix or the intended recipient. To allow the recipient to reply, an alias address can be added. Each mix will keep a list which maps alias addresses and return addresses. Mixes will only forward messages, after receiving N messages.

The Type 1 remailer is vulnerable to a spam attack. An attacker can flood a mix with e-mail messages. When the sender's message is forwarded by the mix, the attacker will detect its destination, as it is the only message for which he does not know the next destination. Other weaknesses are the fact that the message decreases in size by each hop, allowing some traffic analysis, and the fact that replay is not prevented.

The last two weaknesses are solved in the **Type 2 remailer** [4], also called the **Mixmaster**. The message, intended for the recipient, is encrypted multiple times with different symmetric keys. For each mix, through which the message will be sent, a header is prepended. This header is first encrypted with the public key of that mix, and then encrypted multiple times using the symmetric keys of all previous mixes. A header contains the address of the next mix (or the final destination), a Packet ID (avoids replay attacks), an optional Message ID (a message can be sent in different packets), the symmetric key which has to be used to remove a layer of decryption from all following headers and from the internal message.

When a mix receives this kind of message, it will decrypt the top header, obtain the symmetric key included in that header, decrypt all following headers and the internal message. It will shift all headers one place up (at this stage, all the layers of encryption have been removed from the new first header, enabling the next mix to decrypt it with its private key), and add garbage (or the original encrypted top header, as both look random for all observers and other mixes) at the end of the header part (in this way, messages do not decrease in size as they are forwarded by mixes, providing more resistance against traffic analysis). After delaying and reordering, the message is forwarded to the following mix, which in its turn performs the same task, so that finally the internal message is sent to the recipient.

Example Mixmaster based systems can be found at [1] and [13].

A last mix based anonymous email system is **Babel** [8]. In this system, an onion type structure, used for the return address (called Return Path Information), can be included in the message. This provides the recipient the ability to reply, without having the intermediate mixes to keep a list of return addresses (stateless remailers).

6 Anonymity on the World Wide Web

The WWW is perhaps the most frequently used application on the Internet. The importance of anonymous web browsing is certainly increasing, as more and more web servers keep track of users, while they are surfing from site to site (e.g., by providing banners on all these sites, and using cookies for tracking purposes).

The **Anonymizer** [1] provides the same level of anonymity as the Type 0 remailer. The Anonymizer is used as a web proxy, which strips all identifying information, and which forwards the requests to the intended web servers. It thus makes web browsing private in the absence of any eavesdropping or traffic analysis.

The Lucent Personalized Web Assistant, or **LPWA** [6], provides anonymous, yet personal, web browsing. Many web sites require the user to provide a username and password. This allows the web site to offer a personalized service. Unfortunately, users will mostly choose easy-to-remember usernames that can be linked to their real identity. The LPWA provides privacy concerned users with a different, anonymous, and unlinkable, username/password, for each different web site, while the users only have to remember one secret. The *Janus* function is based on pseudo-random functions and collision-resistant hash functions. It transforms an identity id, a web site w and a secret S, into a username u and a password p for the web site w: $J(id, w, S) = (u, p)$. Before browsing the WWW, the user has to login into LPWA by giving its identity and the secret. From then on, the LPWA is used as an intermediate web proxy. Whenever a web site asks for a username and password, the user types in '\U' and '\P', which is replaced by LPWA with the u and p values.

The LPWA provides an anonymous e-mail service in the same way as well. Just as the Anonymizer, the LPWA provides data anonymity, and a low level of connection anonymity. However, Janus could be used in combination with any anonymous communication solution, adding personalized web browsing to it. The LPWA service is now being commercialized as ProxyMate [14].

In the **Crowds** [16] solution, users join a group (crowd) of users, who collectively perform requests. As such, they can anonymously browse the web, as for the web servers, the request are equally likely originating from any of the users.

Each user is represented by a *jondo* on its machine. This jondo is a web proxy that can forward both the user's as other users' requests. Users send their requests to their jondo. This jondo strips all identifying information. A jondo either submits the request to the end server, or forwards the request to another jondo. This path of jondos is randomly chosen initially, but remains the same for the whole session. Requests and replies follow the same path. The communication between the jondos is encrypted. As jondos cannot tell if a request is initiated by the previous jondo, or the one before it, etc, users also maintain their anonymity within the crowd itself.

The Crowds system is essentially a distributed and chained Anonymizer, with encrypted links between Crowds members, providing both data anonymity and a high level of connection anonymity. A possible disadvantage of this system is that for each request, an end-user's IP address is revealed to the web server, instead of a proxy's or router's IP address. A web server (and other people, if the logging information is made public) cannot see the difference between an end-user's jondo, making a request on behalve of an anonymous end-user of the crowd, and an end-user itself making the request. The latter case will thus be supposed, linking that end-user to the particular request.

The previous solutions all addressed anonymous browsing on the WWW. In some cases, anonymous publishing on the WWW is desired. **Rewebbers** and **TAZ servers** [7] provide a solution for this. Rewebbers are web proxies that understand nested URLs (locators). Conceptually, this nested URL has again the same structure as an onion or a Type 1 remailer message: e.g., `http://A/`K_A`(http://B/`K_B`(http://C/`K_C`(http://url/)))` (the real URL `http://url/` is reached after sending the request through three rewebbers A, B, and C). As locators look quite ugly, and are very hard to remember, they can be associated with a .taz address. Unlike anonymous remailers, there is no spam risk involved, due to the pull nature of the WWW.

7 Overview

Table 1 gives an overview of the different existing solutions. The level of anonymity and the availability of the solutions are indicated. The table only gives a very rough measurement, and presents a relative comparison of the different solutions. Comparing solutions across different applications is sometimes difficult, as the anonymity proper-

	Application	Anonymity	Availability
Onion Routing	connection	++	++
PipeNet	connection	++++	–
Freedom Network	connection	++++	++
Pseudonym	e-mail		++++
Type 0 remailer	e-mail	+	+++
Type 1 remailer	e-mail	++	++
Type 2 remailer	e-mail	+++	+++
Babel	e-mail	++	-
Anonymizer	WWW browse	+	+++
LPWA	WWW browse	+	+++
Crowds	WWW browse	+++	++
Rewebber/TAZ	WWW publish	++	-

Table 1. Anonymity level and Availability of different solutions

ties are dependent of the kind of application, e.g., a real-time connection-oriented (WWW) versus a connectionless (e-mail) application.

For the e-mail solutions, it is clear that only using a pseudonym still can reveal your own IP address to the recipient. A Type 0 remailer will only protect against a local observer. The Type 1 and Type 2 remailers protect against global observers, but the Type 2 has stronger properties than the Type 1 remailer. Babel seems to provide the same anonymity level as a Type 1 remailer. With respect to the availability, a pseudonym is actually always used when sending e-mail (though, a normal e-mail address mostly contains the real name of the sender). WWW interfaces exist for all remailers.

For the anonymous connection solutions, Onion Routing has more or less the same level of anonymity as a Type 1 remailer, because of the equivalent structure of an onion and a Type 1 remailer message. PipeNet and the Freedom Network Architecture are designed with a very powerful adversary in mind, and provide therefore a high level of anonymity. Both Onion Routing and Freedom Network are implemented, and can be used by end-users. PipeNet is more a conceptual design, and an implementation is not available.

The Anonymizer is conceptually the same as a Type 0 remailer. The LPWA has the same anonymity properties as the Anonymizer. The LPWA however introduces personalized browsing, which can be used in combination with other anonymity solutions too. Crowds protects anonymity against global observers. Anonymizer and LPWA are freely available services. Crowds is available too, but some more effort (installing additional software at the client side) is necessary. The Rewebber/TAZ concept is equivalent with a Type 1 remailer, but is not available anymore.

8 Anonymity in other applications

Many solutions are already proposed for e-mail and WWW in particular. To provide anonymity to other applications, a general anonymous connections solution must be used.

Payment and voting protocols are two more high-level applications which are of special interest with respect to anonymity. Anonymity is a specific requirement in these applications. Much effort has therefore already been done to provide anonymity at the application level. However, when applying these protocols on the Internet, an anonymous connection solution has to be used as well.

The *Practical mix* [9] is a threshold decryption mix network for ElGamal encrypted messages. Basically, the mix network transforms an input list of ElGamal encrypted messages into a permuted output list of decrypted messages. This idea can be used in the design of mix based payment [10] and voting schemes, in which data and connection anonymity are provided at the same time, and not by two separate mechanisms.

9 Additional remarks

Solutions for anonymous communication are of great value for a privacy concerned society. Unfortunately, they can sometimes be abused. The spam problems with anonymous remailers are a perfect example. The higher the level of anonymity, the more difficult it is to trace the origin of abuse. In the limit, when perfect anonymity is provided, one can abuse the system without being caught. For anonymous electronic payment systems, this would mean that perfect crimes are possible. **Anonymity revocation** is here a necessary technical feature. Especially in the area of anonymous electronic payment systems, the addition of anonymity control has therefore been studied [3]. These systems still provide a very high level of anonymity, but the anonymity can be revoked only with the help of a trustee or judge. The solutions described in this paper do not address the idea of anonymity revocation. Though, the level of possible abuse is not that severe, in Email and WWW applications. Moreover, the anonymity service providers mostly know the real identities of their users, and will reveal them, or will disallow further use of their service, once abuse has been detected, or signaled by victims.

Most described solutions use encryption to provide connection anonymity. Due to the U.S. **export restrictions**, solutions developed in the U.S. cannot be exported outside the U.S. This certainly limits the possible world-wide deployment of particular anonymous communication solutions.

When comparing normal communication to anonymity enhanced communication for a certain application, a **performance** decrease is expected. For providing anonymity,

Appeared in *Proceedings of the IEEE 33rd Annual 1999 International Carnahan Conference on Security Technology*
ICCST'99, Madrid, Spain, October 5-7, 1999, pp 298–303
0-7803-5247-5/99/$10.00 ©1999 IEEE

the data has to be cryptographically processed. Moreover, the data is not transmitted directly from sender to receiver, but through a path of mixes, multiplying the amount of network traffic by a specific factor (the average number of hops).

10 Conclusion

From a technical point of view, this paper shows that there are already many practical solutions available for providing anonymous communication to Internet applications. These solutions are all based on the same principles.

From a social point of view, anonymous communication seems only be desired by a minority of privacy concerned people, and it is even not wanted by many governments and organizations. The solutions for anonymous communication are therefore not yet really integrated in existing products or infrastructure, but are offered as independent services, for which an extra effort has to be made, in order to use them.

Due to the success of electronic commerce, the Internet will be even more frequently used by an increasing number of people. Privacy concern is expected to increase as well. More research is therefore certainly needed on how the technology of providing anonymous communication can be technically integrated in electronic commerce applications.

References

[1] Anonymizer. Anonymizer service. http://www.anonymizer.com/.

[2] D. L. Chaum. Untraceable Electronic Mail, Return Addresses, and Digital Pseudonyms. *Communications of the ACM*, 24(2):84–88, February 1981.

[3] J. Claessens, B. Preneel, and J. Vandewalle. Anonymity Controlled Electronic Payment Systems. In *Proceedings of the 20th Symposium on Information Theory in the Benelux*, 1999.

[4] L. Cottrell. Mixmaster & Remailer Attacks. http:// www.obscura.com/ ˜loki/remailer/remailer-essay.html.

[5] W. Dai. PipeNet 1.1. http://www.eskimo.com/ ˜wei-dai/pipenet.txt.

[6] E. Gabber, P. B. Gibbons, Y. Matias, and A. Mayer. How to Make Personalized Web Browsing Simple, Secure, and Anonymous. In *Proceedings of Financial Cryptography '97 - LNCS 1318*. Springer-Verlag, 1997.

[7] I. Goldberg and D. Wagner. Taz servers and the rewebber network: Enabling anonymous publishing on the world wide web, 1997. http://www.cs.berkeley.edu/˜daw/classes/cs268/taz-www/rewebber.html.

[8] C. Gülcü and G. Tsudik. Mixing Email with Babel. In *ISOC Symposium on Network and Distributed System Security*, February 1996.

[9] M. Jakobsson. A Practical Mix. In *Eurocrypt '98*, 1998.

[10] M. Jakobsson and D. M´Raihi. Mix-based Electronic Payments. In *Fifth Annual Workshop on Selected Areas in Cryptography (SAC'98)*, 1998.

[11] J.Helsingius. anon.penet.fi anonymous remailer. http://www.penet.fi/.

[12] MailAnon. The anonymous e-mail remailer. http://www.mailanon.com/.

[13] Nymserver. Nymserver service. http://www.nymserver.com/.

[14] ProxyMate. Proxymate. http://www.proxymate.com/.

[15] M. G. Reed, P. F. Syverson, and D. M. Goldschlag. Anonymous Connections and Onion Routing. *IEEE Journal on Selected Areas in Communication*, 1998. http://www.onion-router.net/.

[16] M. K. Reiter and A. D. Rubin. Crowds: Anonymity for Web Transactions, 1997. http://www.research.att.com/projects/crowds/.

[17] Zero-Knowledge Systems. The Freedom Network Architecture. http://www.zeroknowledge.com/.

Appeared in *Proceedings of the IEEE 33rd Annual 1999 International Carnahan Conference on Security Technology*
ICCST'99, Madrid, Spain, October 5-7, 1999, pp 298–303
0-7803-5247-5/99/$10.00