ESTABLISHING BILATERAL ANONYMOUS COMMUNICATION IN OPEN NETWORKS

Thomas Demuth
Department of Communication Systems
Universität Hagen
Germany
thomas.demuth@fernuni-hagen.de

Abstract Confidential and authenticated communication is an elementary necessity in open networks like the Internet and has already been realised. In the last years one more criterion of security has become conscious to users of the Internet: privacy.

Especially in the most popular service of the Internet, the World Wide Web (WWW), this need has become obvious. There, users want to anonymously access web pages to leave as little information as possible while browsing. Vice versa it is quite legitimate and understandable in the aspect of multilateral security, that provider of web pages want to offer these pages in the same way.

At first glance, these properties of confidence, authenticity, and anonymity are very contradictory. How can mutual anonymous partners contact each other and exchange authenticated keys to securely communicate with each other?

This article shows a solution to this problem, using pseudonym based signatures and an architecture providing untraceability via a mix net and two kinds of trusted third parties.

Keywords: multilateral security, anonymity, pseudonymity, mix networks, privacy, untraceability

1. INTRODUCTION

The need for security in open communication networks is evident. The main aspect of confidence has been obvious since many years. While communicating sensitive information it has to be secured that this information can only be seen by the sender and the intended receiver, or by authorized other persons, respectively. This demand has been fulfilled many years ago by introducing secure and efficient symmetric encryp-

tion (like DES, [8]), and since 1978 it has become possible to use public key encryption (RSA, [21])

But the more people around the world are using the Internet as an open network the more other criteria of security become evident: Untraceability and especially anonymity. Users of the Internet are not anonymous. While communicating they are producing a lot of personal data, and while this data is being transported, some more protocol information is being generated by the network.

This article focusses attention on the most popular service of the Internet today, the World Wide Web (short: WWW), but the statements, ideas, and the presented architecture are also valid for other (synchronous and asynchronous) services like email, text chat, etc.

In the WWW users requesting web pages are presenting information about their computers, but also about themselves, implicitly. This information is transmitted via the underlying transport protocol HTTP (*HyperText Transfer Protocol*, [12]). Among other data the address and type of the computer they are working with, their affiliation to institutions, their mail address, and other information is transfered. Furthermore, Internet users offer some meta information like the time of the day they use the Internet and other habits and preferences. These and other personal data like the postal address (inserted in a web based form of an Internet shop some time ago) can be the base of a very detailed profile of a person. In addition, data of this kind can be interchanged between providers of web servers to increase the grade of the profiling (other privacy violating mechanisms like cookies are not topic of this article).

In the context of the WWW the base of communication is the client-server-model, the web browser of the user is acting as the client in relation to the web server.

In the last years several so called *anonymising proxies* like *Anonymizer* [1] or the *Rewebber* [20] have been established. These services are offering client anonymity, and therefore the possibility for a WWW user to stay anonymous while surfing (see [6] for an overview; [2] describes a implementation).

But on the other hand, there is the need of offering the same anonymity for some providers of web pages, e.g. the providers of politically and morally correct but inconvenient content. Some years ago the owner of bookstores who offered the "Satanic Verses" by Salman Rushie were threatened by islamic fundamentalists, while it was not illegal to offer this book. Since then many bookstores have gone online.

The problem of server anonymity has been solved by two projects in principle [7, 11]. A publicly accessible service "*Rewebber*" originated from the research project *JANUS* of the Universität Hagen, Germany, is offering this service to WWW users [13].

The next step is to combine these two kinds of anonymity to meet the requirements of the so called "multilateral security" [16] to offer equivalent rights to all parties involved. Additionally one has to consider the two other important criteria of untraceability and confidentiality. But the combination of all these criteria are seemingly contradictory at a first glance. This article presents an architecture solving the problem of establishing (and performing) bilateral untraceable, anonymous, and confidential communication.

2. FUNDAMENTALS AND PROBLEMATIC NATURE

2.1. Anonymity

The very efficient and secure way of offering untraceability, unlinkability, and anonymity for sender and receiver of messages in an open network ([18] explains the terminology in detail) can be established by a so called "mix network" (Another new way for building the return path in mix networks was presented in [24]).

Mixes have been introduced by David Chaum in 1981 [3]. A mix instance collects, encrypts, and reorders messages before sending them to the next mix, which is working the same way. As a result, no adversary can trace the path of a message. For encryption, each mix has a pair of a secret key d and a publicly known key e. Sender and receiver are unlinkable as long as at least one mix is not corrupted.

To establish *anonymity for a sender*, it chooses n mixes $(M_1, \ldots, M_n)$ (with addresses $(A_{M_1}, \ldots, A_{M_n})$) for a path through the mix net, encrypts a message N with the public key e_R of the receiver with address A_R and then successively with each public key of the mixes e_{M_i} on the route, including address information A_{M_i} of the mixes and some random data for blinding k_i for each mix, and sends the result to the first mix (see Figure 1):

$$\begin{aligned} N_{n+1} &= e_R(N) \\ N_i &= e_{M_i}(A_{M_{i+1}}, k_i, N_{i+1}), i \in (1 \ldots n), A_{M_{n+1}} = A_R \end{aligned}$$

If the sender (with address A_S) wants to receive an answer (still staying anonymous), it has to provide a so called *untraceable* (or *anonymous*) *return address* ($anRA$) with the message it sends to the receiver. Therefore this mechanism establishes *anonymity for a receiver.*

$$\begin{aligned} anRA_{n+1} &= A_S \\ anRA_i &= e_{M_i}(anRA_{i+1}, A_{M_i}, k_i), i \in (1 \ldots n) \end{aligned}$$

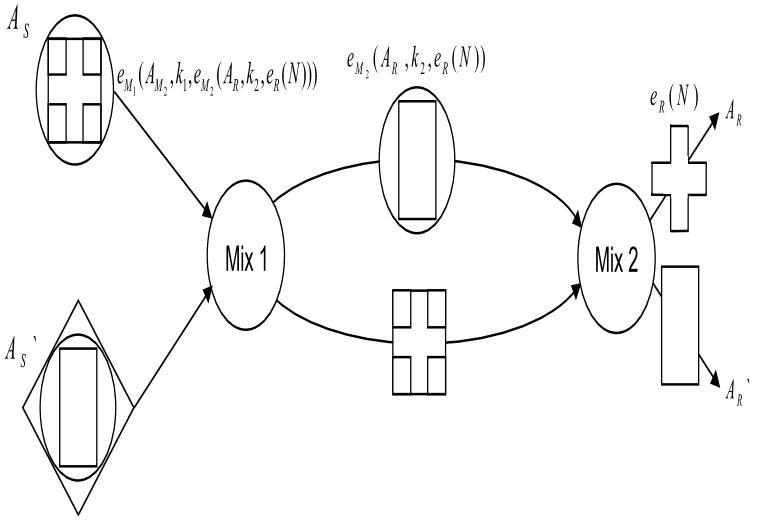

Figure 1. Message processing in a mix network

On the path of the message, each mix can decrypt the outmost layer of encryption of the message with its secret key d_{M_i}.

k_i are randomly selected blinding keys of a symmetric cipher which are used by the mixes to encrypt the message while transporting them through the mix net. For establishing bilateral anonymity and untraceability both mechanisms of anonymity can be combined. Then one mix on the route represents the turning point. Until this point a sender can follow the message, but not further, and after that point, but not before, the receiver can trace it.

Mixes are vulnerable for *replay attacks* in principal: the same input messages are matched to the same messages at the output of the mix. An adversary could investigate correlations by means of the frequency of messages. A mix can be bridged by determination of differences and intersections of ingoing and outgoing messages.
To block these replay attacks, each mix must not transport a message or handle an address twice. Therefore each mix provides a database to store messages, or hashes of them to increase performance.
But this security mechanism implies a problem with respect to the architecture described below: addresses, which are neccessary for the establishing of communication, can be used once at most.

This property does not fit to the situation in the WWW (The sender above can be viewed as a web (page) server, the receiver as a web user): A web page provider (or server), who wants to stay anonymous, has to publish one anonymous address (or anonymous URL) for each access to its page. But a web user (or client) wants to access a page more than

once. Because of communication problems the transport of a request for a web page can be interrupted and has to be repeated.

Problem 1: As a result of the mechanism for the prevention of replay attacks, a web page provider has to create and publish an enormous number of anonymous URLs or at some point no more requests for a web page can be made or would be transported trough the mix net.

Problem 2: Anonymous URLs built in the way described above are very lengthy and difficult to handle for a regular web user. The URL `http://www.sec2002.eun.eg/`, anonymised with the service *Rewebber* becomes:

```
http://www.rewebber.com/surf_encrypted/MTCIZviiwqq7HNPB2NY
EzTC5y7LfWEdZd34HCiQiOAGOANNjI3b8J$eKKnxh5$bLidZIWS$BDAtWc
ba43Pmrukjsg43vpj$1mEwlolmDrcHUWf5chzCBuNmrqz2LRT8os+A=
```

Problem 3: The third problem is the lack of confidentiality because of the non-existing authenticity of keys in the situation described above. In the context of the WWW a publisher of an anonymous URL (a server) does not know the future user of this address and therefore not its (public) key. Vice versa a client does not know the identity of the web server and even if a key is provided with an anonymous URL, it can not trust this key, which can be changed by a potential adversary. How to establish an explicit key authentication while staying mutual anonymous?

2.2. Pseudonym based signatures

In [23] Shamir describes a signature scheme, which eliminates the need of a channel between communicating users to exchange public keys, or between users and a list of public keys, respectively. Instead of this, the verifying key is the identity of a user and the signing key is generated by a trusted party.

In this system i is the identity of a user, g the corresponding secret key, generated such that

$$g^e = i(\,mod\, n)$$

Furthermore, n is the product of two large primes p and q,
e a large prime relatively prime to $\phi(n)$ choosen at random, and
f a one way function.
These parameters, except the primes p and q, are known to all instances in the system. With the knowledge of p and q, only the trusted third party can easily calculate a secret key g for a given identity i. To sign a message m, the user chooses a random number $r \neq 0$ and computes

$$\begin{aligned} t &= r^e(\,mod\, n) \\ s &= g * r^{f(t,m)}(\,mod\, n) \end{aligned}$$

The pair (s, t) represents the signature on m. To verify the signature, the receiving user has to test if the following holds:

$$s^e = i * t^{f(t,m)} (\, mod\, n)$$

The mechanism of signing a value m with a pseudonym based key psd will be visualised as $s_{psd}(m)$ in the following.

In the proposed architecture below, the values n, e, and the function f are publicly known to all participants.

3. PROPOSED ARCHITECTURE

The architecture (Figure 2) consists of following instances:

1 A group of (web) clients, that want to stay anonymous while browsing untraceable and communicating confidentially.

2 A group of (web) servers with the same requirements. Additionally this group want to provide a set of authentic credentials as a base of an anonymous, untraceable, and confidential way for communicating and accessing web pages. These credentials shall be easy to handle for a standard web user and are therefore bound to a pseudonym.

3 An address generator (AG), which holds a list of pseudonym signed URLs and is able to produce anonymous URLs dynamically for any given pseudonym.

4 A trusted third party (KGC, key generation center) generating secret keys for pseudonyms, to be used for signing digital data and therefore enabling the binding of the set of credentials mentioned above to a pseudonym. The KGC is maintaining a system wide database of valid pseudonyms to detect a request for a signature for a pseudonym already existing.

5 One mix network specialised to handle various data streams like web requests and responses and to enable therefore untraceable communication between client and server. Work in the topic of efficient transport of data in mix nets has been done in detail in [**?**, 14, 15, 17]. To simplify the considerations, the mix nets in the architecture are reduced to abstract instances and to their elementary function of providing untraceable communication.

KGC, AG, and the mixes have public keys of a public key cryptosystem like RSA, known to all communicating parties.

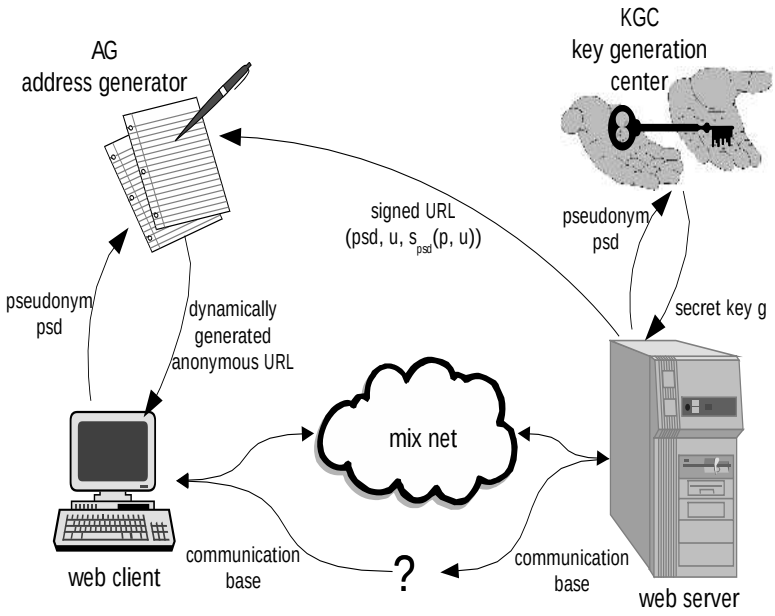

Figure 2. Proposed architecture

Establishing communication via the proposed architecture works in six phases:

1 Generation of a secret key g for a chosen pseudonym psd.

2 Sending of a signed URL to the AG.

3 Distribution of a signed communication base.

4 Request for and generation of a dynamically built anonymous URL.

5 Establishing communication between client and server and exchanging a session key.

6 Communication between client and server using dynamically built anonymous URLs.

These phases are explained in detail in the following.

3.1. Generation of a secret key for a chosen pseudonym

An author or a provider of a web page accessible via an URL who wants to stay anonymous, chooses a pseudonym psd which could be

named arbitrarily but should be as informative as possible and has to be unique across the system. This pseudonym is sent to the KGC.

Only the KGC knows the factorisation of n and can therefore calculate the secret key g for the pseudonym as described above and returns it to the sender. The server now possesses a system wide valid pseudonym, which acts as a public key, and the corresponding secret signing key g.

3.2. Sending of a signed URL to the AG list

The server chooses the URL u of a start web page and signs the URL and the pseudonym psd with its secret key g to bind u and psd together. Therefore it authenticates the tuple $(psd, u, s_{psd}(p, u))$ which is then sent to the AG. The AG is able to verify the signature and adds the pair (psd, u) to its address list.

The server has to send the URL in clear text and not in an anonymised way. Otherwise the route through the mix network using this address could be determined from a certain point when used by the the client afterwards. The mix at this point would not handle the address twice to avoid replay attacks as described above.

3.3. Distribution of a signed communication base

As mentioned above, the mixes use the blinding keys k_i to encrypt the transported message and to change its appearance. To ensure their confidential communication and to speed up encryption, client and server need a common symmetric key. They have to agree on this key without knowing each other.

The server creates a so called *communication base*, consisting of the pseudonym psd, a random value a, which serves as one part of a key based on the Diffie-Hellman key agreement protocol[1], and a public key p chosen by the server to enable the client sending back its part of the Diffie-Hellman key. A signature is made on these three items

$$(psd, a, p, s_{psd}(psd, a, p))$$

together with an optional description of the web pages that are accessible via the communication base. By signing the communication base, no adversary can replace the number a by a different value. The communication base itself is distributed via anonymous channels like newsgroups or anonymous remailer.

One way for the server to distribute the communication base are anonymous mailing systems like remailers [19], offering anonymity in the context of email.

3.4. Request for and generation of a dynamically built anonymous URL

A client, which wants to establish communication to the server and knows the communication base, sends a request to the AG, providing the pseudonym psd.

The AG, knowing the corresponding URL u to the pseudonym psd chooses a route through the mix network by selecting a number of mixes and builds an anonymous URL using the described method. It selects r mixes $(M_1, \ldots, M_r)$ from the mix net, chooses blinding factors k_i and constructs an anonymous URL:

$$anRA(u) \quad = \quad (A_{M_1}, k_1, e_{M_1}(\ldots e_{M_{r-1}}(A_{M_r}, k_r, e_{M_r}(u))\ldots))$$

This anonymous URL is sent back to the client.

3.5. Establishing communication between client and server and exchanging a session key

The client is now in possession of a valid and dynamically built anonymous URL directing to the server, especially to the home page associated with the pseudonym. It selects a second number b at random as the second item for a Diffie-Hellman key agreement and encrypts it for hiding with the public key p.

Furthermore, to be able to be identified by the server in the upcoming communication, it adds a client identification number c_{id} and a token CE for signalling the request for establishing a communication. This request is an inquiry for the web page attached with the pseudonym simultaneously:

$$[p(b, c_{id}, CE, psd)]$$

This request is being sent using the mix network, the previously received anonymous URL $anRA(u)$, and another anonymised return address, built by the client and pointing back to it, so that the server can respond while the client mutually stays anonymous. This results in the following message constructed of the concatenated components described previously:

$$anRA(u)||[p(b, c_{id}, CE, psd)]|anRA(c)$$

3.6. Communication between client and server using dynamically built anonymous URLs

The server responds with the requested web page, associated with a new dynamically built anonymous URL. This is done again via the mix network to preserve anonymity.

In the further communication, the client has to add a prefix of mixes to the anonymous URL sent by the server to stay anonymous. The server knows the anonymous URLs generated by itself and therefore it is able to observe the mixes chosen by itself and can therefore identify a client.

4. ANALYSIS

The main function of the architecture is to establish bilateral untraceable, anonymous, but nonetheless confidential communication in the World Wide Web providing pseudonyms as an simple way of handling anonymous URLs. In the described architecture, the client has to trust the two instances KGC and AG. These are trusted third parties (TTP) like in various public key infrastructures.

In the system, a digital signature, which is produced using the corresponding secret key to the pseudonym (owned only by the holder of the pseudonym), can be verified by any instance.

The three problems shown above are solved as follows:

- Problem 1: With use of the AG dynamically built anonymous URLs can be produced in any number.
- Problem 2: Instead of handling lengthy anonymised URLs, a client uses pseudonyms easy to memorise.
- Problem 3: Even mutual anonymous partners can confidentially communicate with pseudonym based signed keys.

A client using a communication base can be sure that the provided part of the Diffie-Hellman key is authentic, modification of the provided part by a man-in-the-middle attack can be recognised by every instance by checking the signature.

By using the generated key, client and server can establish a confidential end-to-end-communication staying mutually anonymous.

Pseudonyms establish a simple way of communication and are easier to handle and used by non-experts.

5. SUMMARY AND OUTLOOK

The proposed architecture enables the establishing of bilateral untraceable and confidential communication between anonymous partners using pseudonym based signatures.

With light modifications it is possible for client and server to stay anonymous even against KGC and AG: To stay anonymous to the KGC, the server communicates with it via a mix network using sender anonymity and provides an anonymous return address to the KGC. The KGC does not need to know the real address of the sender. Even if the service of the KGC must be paid, this requirement can be fulfilled by mechanisms which have been presented to pay anonymously with anonymous cash [4].

In order to stay anonymous to the AG, the client can use sender and receiver anonymity preserving mechanisms.

These modifications result in handling all communication via mix nets by client, server, KGC and AG. To increase performance this communication can be done using adapted mix nets for each kind of data.

Acknowledgments

The author would like to thank the Department of Communication Systems under supervision of Prof. Dr.-Ing. Firoz Kaderali, and especially Prof. Dr. rer. nat. Werner Poguntke for the support in his research.

Notes

1. In this article the Diffie-Hellman key agreement is assumed as a fundamental cryptologic algorithm and is therefore not explained further. For details see [9].

References

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

[2] O. Berthold, H. Federrath, S. Köpsell, 'Web MIXes: A System for Anonymous and Unobservable Internet Access', International Workshop on Design Issues in Anonymity and Unobservability 2000, LNCS 2009, p. 115-129

[3] D. Chaum, 'Untraceable Electronic Mail, Return Addresses, and Digital Pseudonyms', Communications of the ACM, February 1981, Vol. 24, No. 2

[4] D. Chaum, A. Fiat, M. Naor, 'Untraceable electronic cash', Advances in Cryptology - Crypto 88, Lecture Notes in Computer Science, Springer-Verlag

[5] R. Clayton, G. Danezis, M. Kuhn, 'Real World Patterns of Failure in Anonymity Systems', Information Hiding Workshop 2001, April 2001

[6] J. Claessens, B. Preneel, J. Vandewalle, 'Solutions for Anonymous Communications on the Internet', ICCST - IEEE International Carnahan Conference on Security Technology, 1999, pp. 298-303

[7] T. Demuth, A. Rieke, 'On Securing the Anonymity of Content Providers in the World Wide Web', Proceedings of SPIE '99, Vol. 3657, San Jose, California, January 1999, pp. 494-502

[8] National Bureau of Standards, NBS FIPS PUB 64, 'Data Encryption Standard' National Bureau of Standards, U.S. Department of Commerce, Jan. 1977

[9] W. Diffie, M.E Hellman, 'New Directions in Cryptography', IEEE Transactions on Information Theory, Vol. IT-22, No. 6, November 1976, pp. 644-654

[10] E. Franz, A.Graubner, A. Jerichow, A. Pfitzmann, 'Modelling mix-mediated anonymous communication and preventing pool-mode attacks', *Global IT Security*, Proceeding of the XV IFIP World Computer Congress, 1998, pp. 554-560

[11] I. Goldberg, D. Wagner, 'TAZ Servers and the Rewebber Network', `http://www.cs.berkeley.edu/ daw/classes/cs268/ taz-www/rewebber.html`

[12] HTTP: `http://ds.internic.net/rfc/rfc1945.txt`

[13] JANUS: `http://janus.fernuni-hagen.de/`

[14] D. Kesdogan, R. Büschkes, O. Spaniol, 'Stop-and-Go-MIXes Providing Probabilistic Anonymity in an Open System', pp. 365-380, in 'G. Müller, K. Rannenberg (Eds.): Multilateral Security for Global Communication - Technology, Application, Business', Addison-Wesley-Longman, 1999

[15] D. Kesdogan, J. Egner, R. Büschkes, 'Stop-And-Go-MIXes Providing Probabilistic Anonymity in an Open System', Proceedings of the Second Workshop on Information Hiding (IHW98), Portland, Oregon, USA, LNCS 1525, Springer

[16] G. Müller, K. Rannenberg (Eds.): 'Multilateral Security for Global Communication - Technology, Application, Business', Addison-Wesley-Longman, 1999

[17] A. Pfitzmann, M. Waidner, 'Networks without User Observability', *Computers & Security*, Vol. 6 (1987), pp. 158-166

[18] A. Pfitzmann, M. Köhntopp, 'Anonymity, Unobservability, and Pseudonymity - A Proposal for Terminology', International Workshop on Design Issues in Anonymity and Unobservability, 2001, LNCS 2009, pp. 1-9

[19] Remailer: `http://www.obscura.com/loki/`

[20] Rewebber: `http://www.rewebber.com/`

[21] R.L. Rivest, A. Shamir, L.M. Adleman, 'A Method for Obtaining Digital Signatures and Public-Key Cryptosystems', Communication of the ACM, Vol. 21, No. 2, pp. 120-126

[22] A. Shamir, 'How to Share a Secret', Communication of the ACM, Vol. 24, No. 11,Nov. 1979, pp. 612-613

[23] A. Shamir, 'Identity-Based Cryptosystems and Signature Schemes', Advances in Cryptology - Crypto 84, pp. 47-53

[24] C. Shields, B. Neil Levine, 'A Protocol for Anonymous Communication Over the Internet', ACM - CCS: Conference on Computer and Communication Security, 2000, pp. 33-42