The Anonymity Service Architecture

Mark Borning
Computer Science Dept., Informatik IV
Aachen University of Technology
Ahornstr. 55, 52074 Aachen, Germany
borning@i4.informatik.rwth-aachen.de

Abstract – Business-to-business is the most important area in E-Commerce. With the growth of this area, the importance of multimedia transfers increases as the business partners want to actually see each other during negotiations. In this paper, we will present an architecture that provides confidential multimedia transfers in the Internet, i.e. the transmitted data is protected against espionage from third parties by protecting the contents. Moreover, transmitted packets cannot be traced by observers in the Internet.

Keywords – Multimedia, QoS, Routing, Security, Anonymity, Privacy, Real-Time Communication

I. INTRODUCTION

The Internet is increasingly used in various areas of normal life. Today, one of the most important areas is E-Commerce, and within this area the field of business-to-business (B2B), which describes the relations between business partners, is crucial. In the B2B area, there are requirements different from those in other E-Commerce fields, as negotiations are essential in B2B. Therefore, the usage of multimedia communication systems, like e.g. videoconference systems, will become more important in the near future. They will partly replace face-to-face in negotiations, thus saving considerable time and money.

Multimedia communication tools require both fast computers and fast networks. For example, the maximum transfer rate using [H.261] for a video stream is 1.5 MBit/s. If there are n participants in a videoconference, n video streams are generated and transmitted to a multicast address. The maximum transfer rate will then be $n \times 1.5$ MBit/s. In an Ethernet network with a bandwidth of 10 MBit/s, there are at most six video parallel streams possible.

To secure multimedia communication systems, i.e. to protect the transfer of the multimedia traffic, two techniques are used. First, the data stream is encrypted; an attacker will not be able to see the contents of the transferred packets. Second, the data stream is anonymized, so an attacker is not able to determine who communicates with whom.

In chapter II, we will give an overview of the protection of individual messages. Here, section B gives a short introduction into data streams. In chapter III, the proposed anonymity architecture is introduced. The architecture is used to protect all kinds of traffic, including multimedia streams. Finally, chapter IV summarizes the results and gives an outlook on future work.

II. PROTECTION OF INDIVIDUAL MESSAGES

Negotiations between business partners may be highly confidential; as in many cases the partners do not want any third parties to know that they are negotiating, and what. In an open network, these negotiations have to be protected in the same way as in the real world. This can be done by protecting the contents and the communication scenarios of the negotiations. The protection of the content deals with the end-to-end protection by using cryptography. The protection of the communication scenario addresses questions like: "Who communicates with whom, when, how, and where?" This is done by so-called anonymity techniques that use stations between the communication partners.

A transmitted message can only be protected if an attacker cannot use any property of a message to relate incoming and outgoing messages of a station. The properties of any transmitted message are:

I) **Characteristic**: address information, bit pattern, length and content of the message.
II) **Message correlation**: temporal, spatial and textual correlation, i.e. sequence of messages and relation to other messages.

The characteristic of a message can be determined by observing it. The message correlation is determined by a traffic analysis: An observer analyses the incoming and outgoing messages from any participant or intermediate station. Therefore, any intermediate station has to protect the message's properties against observation or modification to ensure the messages untraceability.

A. Protection of the communication scenario

One method to protect the communication scenario is the Mix method [Cha81]. The protection of the message characteristic is done by a cipher; changing the order in which the messages are received protects the message correlation, because the input sequence differs from the output sequence.

A Mix is, therefore, a system that applies two methods to protect the communication scenario. The first method is substitution, as each message is substituted by another message when arriving at the Mix. The second method is a permutation, as the transmission sequence of the messages differs from the receiving sequence.

Figure 1 describes the structure of a Mix. First, the incoming messages c_i are substituted by the messages m_i by applying the deciphering function d_k. Subsequently, n messages are collected in a buffer forming the sequence $(m_1, \ldots, m_n)$. The permutation π operates on this sequence forming the sequence $(m_{\pi(1)}, \ldots, m_{\pi(n)})$ that is transmitted according to the bandwidth of the outgoing network.

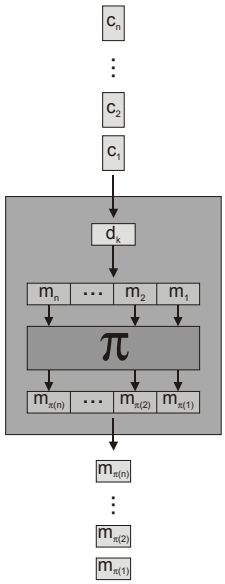

Figure 1: Structure of a Mix.

Every Mix method developed so far can be described in this way. The various methods differ in two aspects. First, different cryptographic methods are used. Second, the permutation differs. It will work with a constant number of messages, and on the other hand, it will work with a variable number of messages, e.g. as shown in the SG-Mix method [Kes99].

If a Mix receives a duplicate of a message, it will destroy the duplicate, as it would allow a trace of the communication. Furthermore, all messages should have the same length, because tracing messages with a specific length can also reveal a communication.

B. Anonymizing QoS Streams

A Chaum Mix only anonymizes n packets of n different participants, where any participant transmits exactly one packet. But there are two kinds of services in today's networks - connectionless services for individual packets and channel services used for data streams and connection-oriented protocols.

Any data stream can be described through certain properties [ChNa98, LoOr99, OrGu00, StZh99]. These properties are the *Quality of Service* (QoS) parameters. For example, a data stream has a transmission rate of λ and the maximum end-to-end delay has to be less then 150 ms. Then the intermediate links have to have adequate bandwidth to transmit the stream, and the delay of the transmission media should be small enough to fulfill the given time constraint, i.e. the maximum end-to-end delay.

The administration of QoS properties in the intermediate routers is much too costly. Therefore, Class of Service (CoS) properties were introduced that meet the requirements of data streams with certain QoS properties. A traffic class contains all streams with the same CoS properties.

III. Anonymity Service Architecture

As described in the previous section, there are several techniques to anonymize individual packets. In the Internet, some systems are available to anonymize e.g. web transactions [Ano01, BFK01, BSG00, JM+98, ReRu98, SGR97]. However, these systems are unable to anonymize data streams with certain QoS requirements. Beside the QoS requirements, data streams possess further properties like the correlation of several packets or cells that form the data stream, e.g., I-, P- and B-Frames of an MPEG video stream are interdependent. Likewise, several data streams may be correlated, like e.g. the transmission of audio and video through separate channels. Another problem is the prevention of dummy traffic as the network load generated by QoS streams is so high that additional dummy traffic (as potentially created by Mixes or the participant client) will overload the network or the systems. The latter can be described as follows:

Consider a system using Web Mixes [BFK01] that requires the users to transmit with identical rates. To provide anonymity, a minimum of about twenty users is necessary [Kes99]. If two participants want to establish a videoconference using the [H.261] standard, each participant may send 1.5 MBit/s, as this is the maximum transfer rate allowed in the standard, generating a total network load of 30 MBit/s. This load is definitely too high for a realistic usage of Web Mixes in multimedia transfers.

The proposed *Anonymity Service Architecture* (AServA) is designed for a wide use in today's networks. The anonymization is done by building anonymity tunnels between the participants. This technique was presented earlier in [JM+98, SGR97]. Unlike these techniques, the AServA will be integrated into the normal network infrastructure and will provide anonymity services for anyone who wants to use them. The system is able to anonymize all traffic classes ranging from real-time services with strong time constraints to best effort services like IP. To simplify the work, any connection-oriented service is considered as a data stream. To avoid dummy traffic, normal traffic that is available in the network will be used in the anonymization process.

The QoS requirements are subdivided into several classes. These classes form traffic classes which meet certain requirements. As these requirements are varying widely, each class is anonymized with a different method. Best Effort traffic, like normal IP traffic, can be anonymized with classical Mix methods, as there are no time constraints to observe. On the other hand, synchronous real-time traffic may have severe time constraints. For

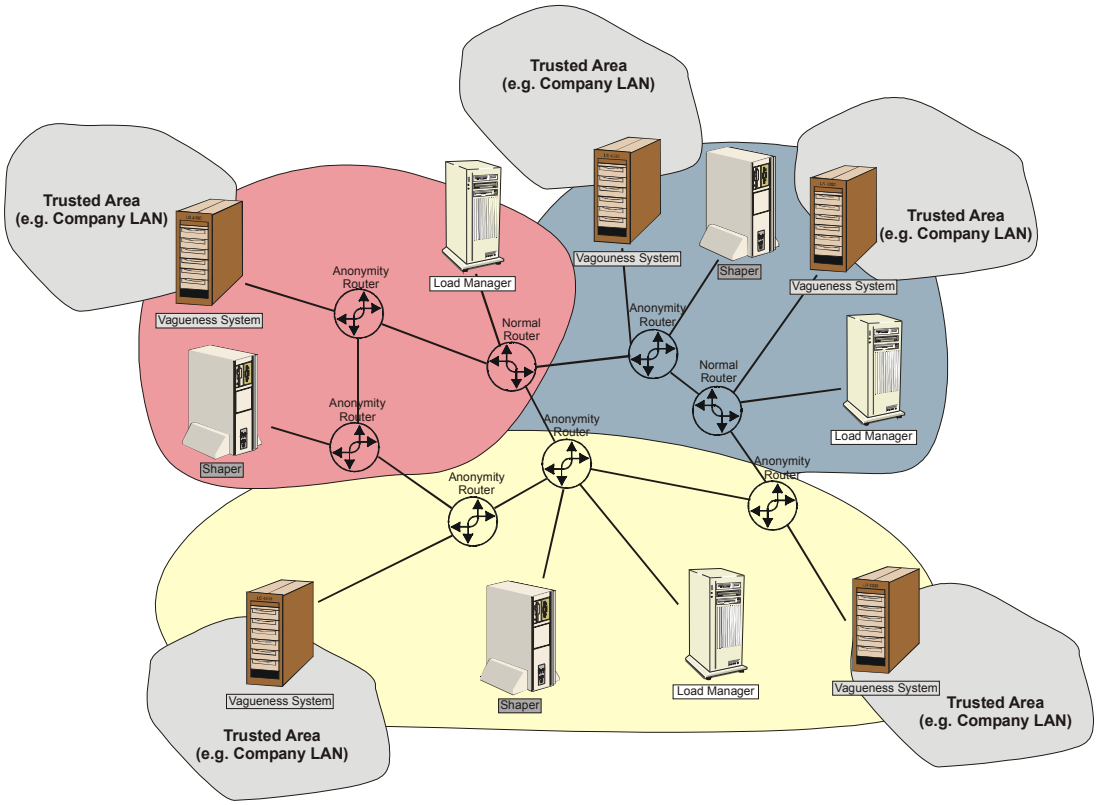

Figure 2: Anonymity Service Architecture

example, the end-to-end transmission time has to be less than 150 ms [G.114], thus another anonymization technique that fulfills this constraint has to be used.

The AServA (Figure 2) can be integrated into an open network like the Internet. The network is separated into a number of network regions, each containing a *Shaper*, a *Load Manager* and a couple of *Anonymity Routers*. These regions and their components are typically administrated by Internet Service Providers (ISPs). To use the provided services, a *Vagueness System* is used. The Vagueness System is an infrastructural component, like a firewall or an intrusion detection system (IDS), to anonymize outgoing streams. It is positioned in a trusted area with several users, like e.g. a company intranet.

Tunnels are established to transfer data anonymized between two Vagueness Systems. These tunnels form a path through the network using several Anonymity Routers providing the anonymization methods. All messages transmitted in the AServA have the same size, so a specific message cannot be identified by its length. The transmitted messages are called 'cells'.

In the following section, the components of the architecture are described in detail. Subsequently, we will describe the services provided by the AServA.

A. Components

The AServA consists of several components. The component relevant to the end user is the *Vagueness System*. It is positioned in a trusted area. Its task is to anonymize the data streams coming from, or going to, a user in the internal network. The Vagueness System communicates with *Shapers*, *Load Managers* and *Anonymity Routers* that are operated by ISPs.

1. Vagueness System

The *Vagueness System* provides the anonymization of network traffic in a trusted area like a company network. It is deployed at the boundary of the trusted area and acts as a security component like a Firewall or an Intrusion Detection System. It provides its service without any normal user noticing it. The Vagueness System consists of a System Manager, a Traffic Classifier, a Routing Subsystem, a Cell Subsystem and a Network Interface.

The *System Manager* controls the Vagueness System and assures its proper functionality. It is called whenever a new data stream arrives at the Vagueness System, and initiates the corresponding actions. These actions are classification of the stream, creation of anonymous cells, selection of routes through the network etc.

The *Traffic Classifier* is used to select an appropriate traffic class of a data stream with certain QoS requirements as described in section 2. The data stream is then

assigned to this traffic class and transmitted according to the properties of the class. In the network, all cells of a traffic class are treated the same way, e.g. the maximum delay of a cell in a real-time traffic class is guaranteed over the whole path.

The *Routing Subsystem* selects an adequate route to the target host. It consists of a *Shaper Interface* and a *Route Cache*. Before sending a request to a Shaper, the Routing Subsystem selects a possible route description in the Route Cache. If no adequate route description is available, the Shaper is queried. The request sent to the Shaper includes the destination, the traffic class and the expected data rates and distribution for each direction.

The *Cell Subsystem* creates anonymous cells that are transmitted using the AServA. It consists of a *Cryptography Object*, a *Key Cache*, a *KDC Interface* and a *Cell Creation Object*. The Cryptography Object does all necessary cryptographic operations on the incoming and outgoing messages. The Key Cache contains several cryptographic keys of other network components like Shapers, Anonymity Routers and other Vagueness Systems. If the key of a specific component is not available, it is acquired using the KDC Interface. Finally, the Cell Creation Object creates any cell needed in the anonymization process. This includes channel creation and initialization, channel usage, etc.

The *Network Interface* receives all incoming cells and outgoing messages. It contains a table with currently registered streams. Each stream is identified by a unique identifier giving information on its source and destination. If an outgoing message of a registered stream is received, the first Anonymity Router is identified; an anonymous cell is generated by the Cell Subsystem and transmitted to the first Anonymity Router. If an incoming cell of a registered stream is received, the destination in the internal network is identified; the cell is transformed into the internal message format and transmitted to the final destination.

If the Network Interface receives a message or a cell of an unregistered stream, the System Manager is invoked.

Any incoming cell of an unregistered stream is either a datagram message or a tunnel establishment message. In the latter case it contains the address of the last Anonymity Router through which it passed, a symmetric key of the stream, and additional information about the stream, e.g. whether it is unidirectional, bidirectional, a datagram service etc. The System Manager identifies the destination of the stream, generates a unique identifier and returns it to the Network Interface that registers the stream with its internal table.

An outgoing message contains the address of the final destination and some QoS requirements. The System Manager determines the lowest possible traffic class using the Traffic Classifier, and an adequate route using the Routing Subsystem. The Cell Subsystem then creates an initialization message, and the System Manager establishes a tunnel to the destination by using the Network Interface. A unique identifier is generated for the stream and registered with the Network Interface.

2. Shaper

The Shaper calculates a route for a tunnel. A *tunnel* is a route between two Vagueness Systems that is used by a QoS stream with a specific traffic class and rate. The functionality can be compared to a DNS server, which obtains a canonical name of a host and determines the IP address of that host. Accordingly, a Shaper receives a canonical name of a Vagueness System, the expected data rate and distribution in each direction, and a traffic class identifier.

The Shaper calculates the route taking into account the current network traffic and the available bandwidth using various Load Managers. It specifies certain Anonymity Routers that are able to anonymize the data stream with their current volume of network traffic.

3. Load Manager

The Load Manager is a network management tool in a network region. It administrates the Anonymity Routers in its region and regularly obtains the in- and outgoing data rates from them. These rates are deterministic for a specific time. As there are static connections between the Anonymity Routers, the Load Manager can determine the status of its network region. If a static connection changes, the Load Manager is informed by the Anonymity Routers involved.

4. Anonymity Router

An Anonymity Router performs the main task in anonymizing a data stream as it receives and forwards the cells of the data stream. As described before, each stream belongs to a specific traffic class. An Anonymity Router uses all cells of a traffic class to anonymize the incoming data streams. As an Anonymity Router is integrated into the network infrastructure, it is able to route normal network traffic as well. To use this traffic in the anonymization process, each packet is transformed into an AServA cell.

An Anonymity Router is logically interconnected to other Anonymity Router through static connections. These connections are protected by a block cipher, as e.g. Rijndael, and are used to multiplex several data streams.

If an Anonymity Router receives a tunnel-initializing cell, it determines the predecessor and successor for this tunnel as well as the maximum response time for a cell of this stream. Obviously, the maximum response time is not a stream but a traffic class property. The Anonymity Router creates an associated FIFO queue for the cells of this stream with respect to traffic class and maximum response time. Each stream is registered with a unique identifier. The Anonymity Router changes this label before forwarding the cell to its successor. These labels are used to route the stream and its cells through the network, see section B.

While receiving cells of a registered data stream, the Anonymity Router appends the cell to the associated

queue. If it receives a cell of an unregistered stream, it is appended to the generic queue of the corresponding traffic class.

Priority Scheduling is used to select a traffic class. Within a traffic class, cells of several queues of a traffic class are transmitted in parallel.

If not enough are cells available in a traffic class to ensure anonymity, cells of a traffic class with lower priority are used. If there are not enough cells in a lower class either, the cells of the current traffic class become members of a higher priority class. As a last resort, dummy traffic will be generated.

B. Services

The AServA provides several services: *Anonymous Datagram Service* (ADServ), *Anonymous Request-Reply Service* (ARRServ) and *Anonymous Tunnel Service* (ATServ). The former two services are datagram services; the latter is used to transmit data streams.

The transmitted service messages possibly exceed the cell size, so a message will be transmitted using more than one cell if necessary.

1. Anonymous Datagram Service

The ADServ is a basic service used to transmit a packet from one Vagueness System to another. First, the Vagueness System selects a route to the destination using the Shaper and creates an *Anonymous Datagram Message* (ADMsg). The ADMsg is then transmitted to the destination using the Anonymity Routers on the selected route. No further action from the sending or receiving Vagueness System or the intermediate Anonymity Routers is necessary.

The Datagram belongs to a specific traffic class, too, so it is routed according to this properties of class. As no associated queue is necessary in the Anonymity Routers, the packet is inserted into the unassociated queue.

Figure 3: Encapsulated Message using exactly one Mix.

As in classical Mix methods, the ADMsg is an encapsulated message, as shown in Figure 3. The ADMsg to the first Anonymity Router contains a header for the Anonymity Router as clear text and the ADMsg to the second Anonymity Router as encrypted payload. The payload is encrypted with the public key of the first Anonymity Router. The payload area of the second packet contains the third Anonymity Router's packet that is encrypted with the public key of the second Anonymity Router, and so on.

2. Anonymous Request-Reply Service

The ARRServ is used to query a server, e.g. a web server or a DNS server. The Vagueness System selects a route to the server using the Shaper and creates an *Anonymous Request Message* (AReqMsg). The AReqMsg is transmitted to the server's Vagueness System using the Anonymity Routers on the selected route.

The AReqMsg is a repeatedly encapsulated message with a header for each Anonymity Router of the selected route. This header contains, among other things, the Reply ID of the request. The Anonymity Router uses the Reply ID to transmit the reply along the same path through the network.

The payload contains the Reply ID of the next hop and a symmetric key. The symmetric key is used for the transmission to the previous Anonymity Router (or Vagueness System, if the current Anonymity Router is the first one).

The Vagueness System at the queried server finally receives the request and creates an *Anonymous Response Message* (ARespMsg). The message is transmitted to the first Anonymous Router in the reverse direction by using the Reply ID and the symmetric key for this hop and so on. Finally, the client's Vagueness Systems receives the reply and delivers it to the requesting client.

3. Anonymous Tunnel Service

The Anonymous Tunnel Service is used to anonymously transport data streams in the network. The Anonymous Tunnel can either be unidirectional or bidirectional.

The ATServ has two phases. The first phase is the *Tunnel Establishment*; the second phase is the *Tunnel Usage*.

During the Tunnel Establishment phase, the Tunnel Initiator, the Vagueness System of the Sender, creates an *Anonymous Tunnel Creation Message* (ATCMsg). This Message contains several fields in its payload that initialize the Anonymity Routers, including

- a label used on the incoming link,
- a label used for the outgoing link,
- a symmetric key used to encipher the incoming cells, and
- a flag, indicating whether the channel is used uni- or bidirectional.

An Anonymity Router will create an associated queue for that stream. If the Tunnel is bidirectional, two queues are created.

There are other fields in the payload, too, that describe the properties of the Tunnel, as e.g. *Lifetime*, *Data Rate*, *Link Key*, etc.

After the establishment of the Tunnel, there are two types of messages:

- *Anonymous Tunnel Data Message* (ATDMsg): This type of message is used to transfer data from one Vagueness System to another.
- *Anonymous Tunnel Management Message* (ATMMsg): This type of message is used to manage the Tunnel. The actions are key changing, extension of the Tunnel lifetime, etc.

When the lifetime of a tunnel is expired it is automatically shut down. Then, all Anonymous Routers will delete the associated queues and all other information about the tunnel.

The ATDMsg is constructed very simply. It contains the current label and the encrypted data. Every Anonymity Router used changes the label and encrypts the data with its key.

C. Inter-Component Communication

In the previous section, the communication between various Vagueness Systems was described. This communication should not be visible to any observer. In contrast, the communication between other components might be visible, as there is no need for any additional protection. First, we will describe the communication between an Anonymity Router and a Load Manager, followed by a description of the communication between Shaper and Load Manager, and between a Vagueness System and a Shaper. Finally, the communication between two Anonymity Routers is described, which uses the multiplexing algorithm mentioned earlier. The communication is not described in detail, but we will show the concepts.

A Load Manager queries an Anonymity Router for traffic data. The Load Manager receives the number of incoming and outgoing cells from each Anonymity Router. Furthermore, the Load Manager limits the number of static connection of the Anonymity Router and sets the parameters for the number of cells transferred in parallel in the multiplexing algorithm.

A Shaper has to communicate with a Load Manager to determine a route for a QoS stream. The Shaper connects to a Load Manager and transmits a *RequestStatus*-Message containing the expected data rate and the traffic class of the QoS stream. The Load Manager will then compute the traffic situation in its network region by querying the Anonymity Routers. It will establish which links are capable of carrying the stream and send a list of these links back to the Shaper.

A Vagueness System needs a Shaper to compute a route. It transmits the destination address, the expected data rate and the traffic class of a specific QoS Stream to the Shaper. The Shaper first determines the network regions which will be used to transmit the stream. Then the Shaper queries the Load Managers of those regions to compute the route. This route will be returned to the Vagueness System.

The communication between two Anonymity Routers is realized through static connections. The connection is protected with a stream cipher.

IV. Summary and Outlook

AServA is a security architecture that provides the ability to create anonymous connections for all types of QoS streams, especially multimedia streams, for instance for videoconferences. The architecture is not positioned on top of an existing network but integrated into it. Therefore, normal network traffic can be used to anonymize the confidential traffic. The costs are split into two parts. The first part refers to the introduction of so-called Vagueness Systems. They will become a basic security system like a Firewall or an Intrusion Detection System. The second part is the replacement of some routers with Anonymity Routers by Internet Service Providers. Furthermore, the ISPs need to set up Shapers and Load Managers. A Shaper computes routes between Vagueness Systems for a specific QoS stream; a Load Manager controls the load of network traffic between the Anonymity Routers in its network region.

The implementation of the architecture will commence this summer; it will be available as a prototype in summer 2002.

Literature

[Ano01] Anonymizer: http://www.anonymizer.com, Anonymizer Inc.

[BFK01] O. Berthold, H, Federrath, S. Köpsell: *Web MIXes: A System for Anonymous and Unobservable Internet Access*. In: H. Federrath (Ed.): Anonymity 2000, LNCS 2009, pp. 115-129, 2001.

[BSG00] Ph. Boucher, A. Shostack, I. Goldberg: *Freedom System 2.0 Architecture*. White Paper, Zero Knowledge Systems, Inc., 2000.

[Cha81] D. Chaum: Untraceable Electronic Mail, Return Addresses, and Digital Pseudonyms. In: Communications of the ACM 24/2 (1981) 84-88.

[ChNa98] S. Chen, K. Nahrstedt: *An Overview of Quality-of-Service Routing for the Next Generation High-Speed Networks: Problems and Solutions*. IEEE Network Magazine, Special Issue on Transmission and Distribution of Digital Video, 1998.

[G.114] ITU-T Recommendation G.114: *One-Way Transmission Time*. 02/96.

[H.261] ITU-T Recommendation H.261: *Video Codec for Audiovisual Services at $p \times 64$ KBits*. 03/93.

[JM+98] Anja Jerichow, Jan Müller, Andreas Pfitzmann, Birgit Pfitzmann, and Michael Waidner: *Real-Time Mixes: A Bandwidth-Efficient Anonymity Protocol*. In IEEE Journal on Selected Areas in Communications. Vol. 16, No. 4, May 1998.

[Kes99] D. Kesdogan: *Privacy im Internet – Vertrauenswürdige Kommunikation in offenen Umgebungen*. DuD-Fachbeiträge, Vieweg Verlag, 1999.

[LoOr99] D. H. Lorenz and A. Orda: *Optimal Partition of QoS Requirements on Unicast Paths and Multicast Trees*. Proceedings of IEEE INFOCOM'99, New York, March 1999.

[OrGu00] A. Orda and R. Guerin: *QoS Routing: The Precomputation Perspective*. In IEEE Infocom, 2000.

[StZh99] I. Stoica and H. Zhang: *Providing Guaranteed Services Without Per Flow Management*. In Proceeding of SIGCOMM '99, 1999.

[ReRu98] M. K. Reiter and A. D. Rubin. *Crowds: Anonymity for Web Transactions*. ACM Transactions on Information and System Security, 1(1):66—92, 1998.

[SGR97] P.F. Syverson, D.M. Goldschlag, and M.G. Reed: *Anonymous Connections and Onion Routing*. In: Proceedings of the 1997 IEEE Symposium on Security and Privacy, May 1997