

# Stop-And-Go-MIXes Providing Probabilistic Anonymity in an Open System

Dogan Kesdogan\*, Jan Egner, Roland Büschkes

Aachen University of Technology - Department of Computer Science  
Informatik 4 (Communication Systems)  
D-52056 Aachen, Germany  
*kesdogan@informatik.rwth-aachen.de*

**Abstract.** Currently known basic anonymity techniques depend on identity verification. If verification of user identities is not possible due to the related management overhead or a general lack of information (e.g. on the Internet), an adversary can participate several times in a communication relationship and observe the honest users. In this paper we focus on the problem of providing anonymity without identity verification. The notion of probabilistic anonymity is introduced. Probabilistic anonymity is based on a publicly known security parameter, which determines the security of the protocol. For probabilistic anonymity the insecurity, expressed as the probability of having only one honest participant, approaches 0 at an exponential rate as the security parameter is changed linearly. Based on our security model we propose a new MIX variant called "Stop-and-Go-MIX" (SG-MIX) which provides anonymity without identity verification, and prove that it is probabilistically secure.

## 1 Introduction

Recently much attention has been given to the application of anonymity techniques to various networks (Internet, ISDN, GSM etc.) and for various applications and purposes (email, WWW, location management, etc.). Basic well known techniques providing anonymity are:

1. Implicit Addresses and Broadcasting [5, 16],
2. DC-Networks [2], and
3. MIXes [1].

A good overview about these techniques can be found in [18]. A short summary will also be given in the following section. Based on this introduction we show that these techniques are well applicable to closed environments<sup>2</sup>, but have shortcomings in open environments<sup>3</sup>. The reason for this is that the techniques

---

\* The work of D. Kesdogan was supported by the Gottlieb Daimler and Karl Benz Foundation.

<sup>2</sup> I.e. the number of users is some known and not too large number  $n$  (e.g.  $n < 1000$ ).

<sup>3</sup> I.e. the number of potential users of a MIX is more than one million and usually not known exactly.

depend on identity verification in order to provide security (see Section 2). Unfortunately, such information is not always available (e.g. on the Internet) and hence these basic methods are insecure against the usual attacker model. For that reason, most of the recent findings restrict the attacker to a weaker model in which he or she is not able to tap all used lines [6, 7, 10, 11, 21, 22].

We therefore conclude that the basic techniques provide security only if either user specific information is available or the adversary is restricted to a weaker model. The resulting question is whether secure techniques for an open environment exist, which do not depend on identity verification and withstand the strong attacker model. Our answer to this question follows a probabilistic model. We define this model in section 3 analogous to the one used for public key cryptosystems. Following our new probabilistic model, we present in section 4 a technique called Stop-and-Go-MIX and prove its probabilistic security.

We finish our paper with a short conclusion and an outlook on potential applications and extensions of our model.

## 2 Basic Notions And Techniques

For the following discussion of anonymity techniques we make two general assumptions:

1. The underlying communication network is global and is not subject to any topology restrictions.
2. The attacker model<sup>4</sup> used throughout this paper assumes an omnipresent attacker E. E is able to tap all transmission lines of the communication network and to control all but one intermediary switching node. The attacker E is not able to break the used cryptographic techniques.

The question now is how to hide the existence of any communication relationship, i.e. that a message was sent (sender anonymity) or received (receiver anonymity) by a user. Although the content of a message can be well protected by cryptographic techniques, the use of cryptography solely can not guarantee anonymity. The omnipresent attacker E can observe the sender of a message and follow the message up to the receiver, thereby detecting the communication relation without a need to read the content of the packets.

Hence, the decisive point of anonymity techniques is to organize additional traffic in order to confuse the adversary and conceal the particular communication relationship. The sender and/or receiver of a message must be embedded in a so-called anonymity set.

The main questions related to an anonymity set are:

1. How is the anonymity set established?
2. What is the size of the anonymity set?

---

<sup>4</sup> The attacker model is based on the one given in [1].

All of the following anonymity techniques differ in their approach towards establishing anonymity sets and therefore also differ in their possible field of application.

## 2.1 Implicit Addresses and Broadcasting

One basic technique for anonymity is the combined use of *Implicit Addresses* and *Broadcasting*. If user A wants to keep the recipient B of a message secret, he chooses additional pseudo recipients (e.g. C and D). Together with the real recipient B, these additional recipients form the anonymity set. The message is broadcasted to all members of the anonymity set (Fig. 1). To identify the real recipient within the anonymity set A uses an implicit address "x". An implicit address is an attribute by which no one else than A and B can recognize the message as being addressed to B (for other types of implicit addresses see [18]).

The technique has a clear security limit. If the additional recipients (C and D) cooperate with attacker E, B can easily be identified as the recipient of the message.

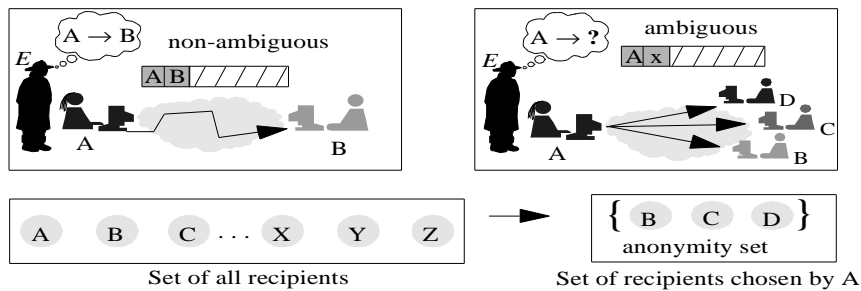


Fig. 1. Recipient anonymity by broadcast and addressing attributes

This attack scenario can be enhanced to the case that attacker E adopts different false identities (e.g. that of C and D) and therefore controls  $(n - 1)$  members of the anonymity set by simply impersonating them. In order to defend against this kind of masquerading attack, identity verification through an adequately secure and flexible identification technique is required.

**Drawbacks and Recent Directions.** What becomes clear from the simple example given above is that the broadcasting<sup>5</sup> overhead is prohibitive for a large scale network. Furthermore, the security attainable by this simple scheme is restricted to recipient anonymity. The sender of a message is observable at least by the recipients.

<sup>5</sup> We assume here a switching network, not a broadcast network (e.g. satellite network).

A major drawback is that the technique depends on a closed anonymity set, which must be actively created by the sender of the message.

Implicit addresses and broadcasting as independent techniques are used in different anonymity techniques as basic building blocks [8]. These techniques are also applied in real networks. GSM networks e.g. use implicit addresses on the air interface to hide the real identity of the users.

In research, different findings for anonymous mobility management are based on implicit addresses and broadcasting [12, 13, 14, 21]. An extension to the basic scheme called *Variable Implicit Addresses* is presented in [8].

## 2.2 DC-Network

The DC-Network [2], a powerful technique for sender anonymity, uses superposed sending. To use a DC-network, users exchange secret keys along a given key graph, i.e. a graph with the users as nodes and the secret keys as edges. This initial phase of the protocol establishes the needed anonymity set. Obviously attacker E must again be prevented from controlling a majority of the exchanged items (keys or messages). Therefore, identity verification is necessary during this initial set-up phase.

To send a message (a sequence of bits), user A superposes (adds modulo 2) the message with the previously exchanged secret key. Other users superpose in the same manner (Fig. 2). If the superposed packets are transmitted via a network, an eavesdropper is not able to decide whether a packet really contains a message or not. All sums of all users are superposed globally and the result is distributed to all user stations. This distribution process guarantees the recipient anonymity. Because every secret key is added twice, the distributed message is the message of A. If more than one message was sent in the same period a collision occurs. This collision can be detected by the respective senders and collision avoidance protocols, known from multi access channels, can be applied.

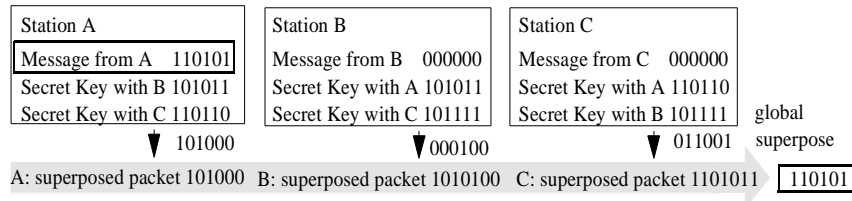


Fig. 2. DC-Network(taken from [17])

The anonymity set for the above example consists of A, B and C. Building secure groups beforehand by exchanging secret information enables the participants to generate packets for which the adversary is unable to decide whether

they contain a message or not. Therefore, DC-Networks provide an information-theoretic deterministic anonymity (see Def. 1)[2, 17].

**Drawbacks and Recent Directions.** The application of the basic method suffers from the related high overhead. For every real message  $n$  packets have to be transmitted. Therefore, [18] suggests to implement superposed sending on a physical ring network.

For special environments like e.g. broadcast networks superposed sending could be applied to reduce the cost (i.e. bandwidth). [3] proposes a superposing technique for reading anonymously from databases.

A major drawback is that DC-Networks require the installation of a closed anonymity set before the application of the superposing technique. Hence, this basic technique is not flexible. In order to include a new participant in the DC-network, a new key distribution graph has to be generated.

### 2.3 The MIX-Method

The two anonymity techniques presented so far both suffer from the same major drawback that they require the pre-installation of a closed anonymity set. The members of the anonymity set are then involved in every communication. This severely limits the flexibility of the involved users.

The MIX-Method [1] avoids this drawback by shifting the task of generating anonymity sets from the user to special intermediate network nodes called MIX nodes or MIXes. Centralized MIXes can serve a great amount of users without the constraint that each user has to participate in every communication. Hence, in designing a system that provides flexible access to an anonymity service the MIX approach is the most interesting and the only one suitable for open networks. MIXes collect a number of packets from distinct users (anonymity set) and process them in a way that no participant, except the MIX itself and the sender of the packet, can link an input message to an output message (Fig. 3). Therefore, the appearance (i.e. the bit pattern) and the order of the incoming packets have to be changed within the MIX. The change of appearance is a cryptographic operation, which is combined with a management procedure and a universal agreement to achieve anonymity:

1. User protocol: All generated data packets with address information are padded to equal length (agreement), combined with a secret random number ( $RN$ ) and encrypted with the public key of the MIX node. A sequence of MIXes is used to increase the reliability of the system.
2. MIX protocol: A MIX collects  $n$  packets from distinct users (identity verification), decrypts the packets with its private key, strips off the  $RNs$  and outputs the packets in a different order (lexicographically sorted or randomly delayed). Furthermore, any incoming packet has to be compared with former received packets (management: store in a local database) in order to reject any duplicates. Every MIX (except the first) must include an anonymous

loop back<sup>6</sup>, because only the first MIX can decide whether the packets are from distinct senders<sup>7</sup> or not.

E.g. assume that A wants to send a message  $M$  to Z (Fig. 3). A must encrypt the message two times with the public keys  $c_i$  of the respective MIXes and include the random numbers  $RN_i$ :  $c_1(RN_1, c_2(RN_2, Z, M))$ .

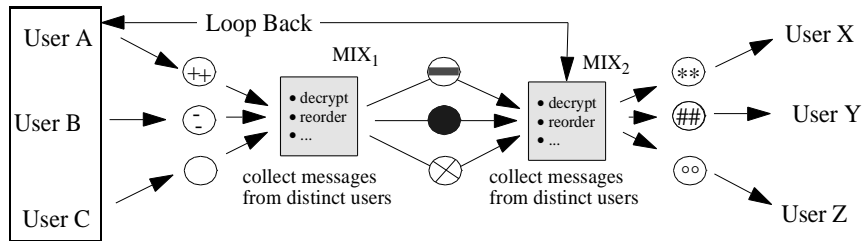


Fig. 3. Cascade of two mixes

Applying this protocol the MIX method provides full security. The relation between the sender and the recipient is hidden from an omnipresent attacker as long as he neither controls every MIX which the message passes nor cooperates with the other senders ( $n - 1$ ). [17] states that the MIX method provides information-theoretic deterministic anonymity based on complexity-theoretic secure cryptography.

**Drawbacks and Recent Directions.** Before a MIX can forward a packet, it has to collect  $n$  messages from different users. This is a typical example for an asynchronous communication model (like e.g. email). The resulting end-to-end transmission delay can be minimized if every participant sends a packet at a given time. Of course, most of these messages would be dummy messages (see e.g. ISDN-MIXes [19]). In an open environment (global network) the sending of sender originated dummies should be avoided. Therefore [6, 7, 10] propose to hide the relation between sender and recipient by using a number of distinct MIXes without distributing the collecting property over the network. But these schemes are insecure against an omnipresent attacker E. It is hence still an

<sup>6</sup> Loop back: Every MIX knows the sender anonymity set. It signs the received packets and broadcasts them to the respective users. Each user inspects whether his own message is included or not and transmits a yes or no. The MIX goes on if it receives yes from all members of the anonymity set.

<sup>7</sup> Most of the suggestions for MIX realizations in the literature work without this property and are therefore not secure, because the former MIX(es) can conspire with the opponent and generate  $(n - 1)$  dummy packets and observe the only remaining packet which is from the user of interest.

unsolved problem to provide security for synchronous communication in an open environment facing an omnipresent attacker.

All suggestions for the Internet scenario have to deal with the lack of identity verification information. If a MIX cannot decide whether the packets are from different senders or not<sup>8</sup>, the attacker can intercept the incoming packets, isolate each packet, and forward it together with  $(n-1)$  of his own packets<sup>9</sup>. This attack is well known as the  $(n-1)$ -*attack*, *blocking* or *trickle attack* [11].

Examining the  $(n-1)$ -attack we can identify two reasons for its success:

1. Time: The time until  $n$  messages are collected by a MIX and hence the end-to-end delay of a message is not known. Therefore it is possible to delay a single message without any risk of detection.
2. Deterministic output behavior: Sending messages with a high rate to the MIX results in a high output rate of the MIX.

The solution given in [11] is to delay every packet individually from one batch to the other and additionally choose detours for every packet from every MIX with a given probability. The detours are over several other MIXes. As stated by the authors, this scheme is insecure against the omnipresent attacker. In the literature known to us there is no existing solution for the Internet which is also secure against an omnipresent attacker.

### 3 Probabilistic Security

The techniques discussed in the previous section share the basic requirement of identity verification. While the protocols provide complete and even perfect security under the assumption that the knowledge necessary for identity verification is available, this requirement can severely handicap or even prevent their application.

What is needed is a technique that provides security for an open network without the need of identity verification. The question is what level of security can be guaranteed without the need of identity verification. Before answering this question, we now give a formal definition for the basic terms already used in the above description:

**Definition 1.** Given an attacker model  $\mathcal{E}$  and a finite set of all users  $\Psi$ . Let  $\mathcal{R}$  be a role for the user (sender or recipient) in respect to a message  $\mathcal{M}$ . If, for an attacker according to model  $\mathcal{E}$ , the a-posteriori probability  $p$  that a user  $u$  has the role  $\mathcal{R}$  in respect to  $\mathcal{M}$  is non-zero ( $p > 0$ ), then  $u$  is an element of the *anonymity set*  $\mathcal{U} \subseteq \Psi$ .

---

<sup>8</sup> Again this is the case if it is not possible to verify the identities of the senders of the mixed packets.

<sup>9</sup> If the anonymity set is built over an indeterministic procedure (e.g. every packet is delayed randomly) then the adversary fills up the MIX with his own packets and, after forwarding the one real message, keeps on sending his own packets until the MIX outputs the one real packet.

A technique (method) provides an  $\mathcal{R}$  *anonymity set* of size  $n$  if the cardinality of  $\mathcal{U}$  is  $n$  ( $n \in \mathbb{N}$ ).

An algorithm provides *deterministic anonymity* if  $n$  is always greater than 1.

Obviously, the basic techniques presented so far provide deterministic and at their best information-theoretic anonymity, but have to verify the users identities to be secure at all. We relax the information-theoretic anonymity property of the above techniques to a notion of probabilistic anonymity in order to find a scheme which does not depend on identity verification and can be applied to open networks and groups:

**Definition 2.** Given an attacker model  $\mathcal{E}$ , let AL be an algorithm providing anonymity with a complexity parameter  $\mu$ . We say that AL is *probabilistically secure* against the attacker model  $\mathcal{E}$  if AL, for a distinct message  $\mathcal{M}$ , can be broken with probability  $\alpha$  and if

1. the a-posteriori probability of insecurity after any possible attack within the attacker model  $\mathcal{E}$  is the same as the a-priori probability before an attack occurs, i.e.  $\alpha$  remains constant for a given  $\mu$ , and
2. the probability of insecurity approaches 0 at an exponential rate as  $\mu$  is increased linearly.

If AL is *probabilistically secure* it provides *probabilistic anonymity*.

Our aim is to define a protocol providing anonymity, where  $\mu$  is given as a publicly known parameter. If the users use this parameter they can send a message spontaneously with the probability of insecurity  $\alpha$  determined by  $\mu$ , i.e.  $\alpha = f(\mu)$ .

Security Evaluation Model	Anonymity Evaluation Model
Information-theoretic security	Information-theoretic anonymity
Complexity-theoretic security	Probabilistic anonymity

**Table 1.** Anonymity Evaluation Models

Note that the above definition is analogous to the computational security of asymmetric cryptography [4, 9, 23]. To break the asymmetric protocols should be computationally as hard as solving a problem of the complexity class NP. Because the algorithms for encryption and decryption should be carried out in polynomial time  $P$  this would lead to the proof that  $P \neq NP$ . While this is a well known unproven hypothesis in complexity theory the security of the schemes depend on security parameters, which determine the length of the key etc. in accordance with the special considered problem (factoring assumption, discrete logarithm etc.) and the time complexity of the best known algorithm. The security here also depends exponentially on the parameter [20].

Table 1 compares the resulting models for security and anonymity evaluation.



## 4 The Stop-and-Go-MIX (SG-MIX)

Based on the definition of probabilistic anonymity we will now define the SG-MIX protocol and evaluate its security properties.

### 4.1 The SG-MIX Protocol

A SG-MIX (Fig. 4) operates in the same way as a classical MIX, but does not collect a fixed number of messages. Sender A selects the SG-MIXes to be used with equal probability. He calculates for every node  $i$  a time window  $(TS^{min}, TS^{max})_i$  and draws a random delay time  $T_i$  from an exponential distribution with suitable parameter  $\mu$ . This information is appended to the packet before encrypting it with the SG-MIX's public key. The SG-MIX  $i$  extracts  $(TS^{min}, TS^{max})_i$  and  $T_i$  after decryption. If the arriving time of the packet is earlier or later than given by the time window the message will be discarded. After  $T_i$  units of time have elapsed, the SG-MIX  $i$  forwards the packet to the next hop or its final destination.

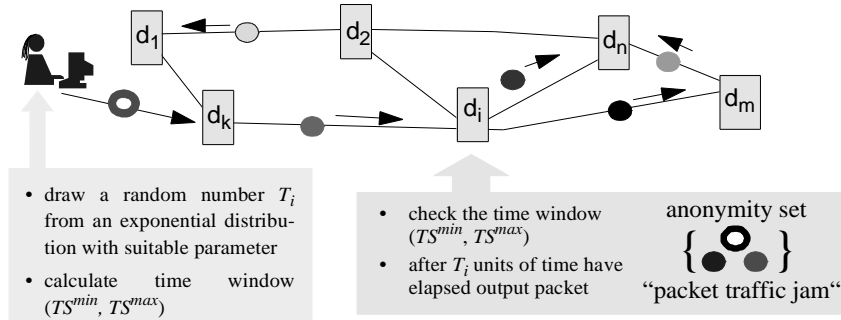


Fig. 4. SG-MIX

Selecting the SG-MIX nodes with equal probability guarantees that an user does not prefer certain SG-MIXes over others, which would enable an attack via the analysis and comparison of the amount of traffic origin from the user and processed by his favorite SG-MIXes. Furthermore, the anonymity size (see Definition 1) will be always maximal with a high probability.

### 4.2 The Security of the SG-MIX

The security of a SG-MIX does not rely on shuffling a batch of messages, but on delaying each message individually and independently by a random amount of time. If the delay times are individually drawn from the same exponential

distribution, the knowledge of the time a specific message has arrived at the SG-MIX node does not help the attacker in identifying the corresponding outgoing message as long as there is at least one other message in the queue at some time of the delay. Because of the memoryless property of the exponential distribution, if  $n$  messages are in the queue, it is equally probable for any one of them to depart next, regardless of their arrival times. Therefore, an attacker can correlate arrival and departure of a message only if during the whole delay time no other message is in the queue. This means that

1. A: the queue is empty at the arrival of the message (and the attacker knows this) and
2. B: no other message arrives during the delay time.

More formally speaking the probability of an successful attack is:

$$\mathcal{P}(\text{success}) = \mathcal{P}(\mathcal{A} \cap \mathcal{B}) = \mathcal{P}(\mathcal{A}) \cdot \mathcal{P}(\mathcal{B}) \quad .$$

Clearly, the arrivals are Poisson distributed due to the independent choice of the nodes and both events are independent from each other. Their probabilities will now be calculated.

The queue of a SG-MIX node can be modeled as an  $M/M/\infty$  server, because the arrivals are Poisson distributed, the “service times” are exponentially distributed and the queue can “serve” any number of customers independently. We denote the rate of message arrivals by  $\lambda$  and the parameter of the exponential distribution from which the delay times are drawn by  $\mu$ . Queuing theory then gives a (steady state) probability of  $\mathcal{P}(A) = e^{-\lambda/\mu}$  that this server system is idle when a message arrives [15].

Because both interarrival times and delay times are exponentially distributed with respective parameters  $\lambda$  and  $\mu$ , the probability that no message arrives during the delay time is equal to the probability that a sample drawn from  $\text{Exp}(\lambda)$  is greater than a sample from  $\text{Exp}(\mu)$ . Hence we get

$$\mathcal{P}(\mathcal{B}) = \frac{\mu}{\lambda + \mu} = \frac{1}{1 + \lambda/\mu} \quad .$$

The probability that an arbitrary message can be tracked by an eavesdropper (an active attacker has no additional advantage) is therefore

$$\mathcal{P}(\text{success}) = \frac{e^{-\lambda/\mu}}{1 + \lambda/\mu} \quad . \tag{1}$$

Let us consider an example: Assume a SG-MIX node with a mean arrival rate  $\lambda = 10$  packets/s and parameter  $\mu = 0.2$  packets/s, that is a mean delay of 5 seconds. Then the probability of an arriving packet finding the server empty is  $e^{-50} \approx 1.9 \cdot 10^{-22}$ .

**(n-1)-Attack** In order to provide probabilistic anonymity a SG-MIX must be able to fend off blocking attacks. When running such an attack the intruder must delay all incoming data packets for a certain amount of time in order to “flush” the SG-MIX. Therefore we introduce the time stamps  $(TS^{min}, TS^{max})$  to detect the delay of an incoming data packet and discard this packet. This prevents blocking attacks. The SG-MIX technique allows the calculation of the time windows very accurately as the user knows the time a message will be delayed in advance.

We define for the pair of time stamps of node  $i$   $(TS^{min}, TS^{max})_i$  the time window  $\Delta t_i$  during which a packet must arrive at SG-MIX  $i$ . If a  $\Delta t$  value is given, we can calculate the success probability of a blocking attack:

A packet leaves a SG-MIX after at most  $\Delta t$  units of time with probability  $\mathcal{P}(X \leq t) = 1 - e^{-\mu\Delta t}$ . Assuming there are  $n$  packets in the SG-MIX the success probability of a blocking attack is  $\mathcal{P}(\text{success}|X = n) = (1 - e^{-\mu\Delta t})^n$ . The number of packets in the SG-MIX at an arbitrary point of time is Poisson distributed with parameter  $\rho = \lambda/\mu$ :

$$\mathcal{P}(X = i) = \frac{\rho^i}{i!} e^{-\rho} \quad .$$

Therefore the overall success probability of an  $(n - 1)$ -attack is

$$\mathcal{P}(\text{success}) = \sum_{i=0}^{\infty} \frac{(1 - e^{-\mu\Delta t})^i \cdot \rho^i \cdot e^{-\lambda/\mu}}{i!} = \exp\left(\frac{-\lambda e^{-\mu\Delta t}}{\mu}\right) \quad . \quad (2)$$

Obviously, when  $\Delta t$  is given, a linear decrease of  $\mu$  leads to an exponentially decreasing success probability of a blocking attack. The only successful attack is if the adversary blocks the incoming messages of all SG-MIXes quite long before the attacked message arrives, due to the lack of knowledge which SG-MIX node would be selected from the user. This is usually impossible to do “on demand” and, in any case, would block the whole network, i.e. result in the loss of many messages due to time-outs, which would surely not go undetected.

**Calculating Time Stamps.** When defining the time stamps one has to take into account that computer clocks are not perfectly synchronized. Many different clock synchronization mechanisms have been proposed, each of them providing a different quality of synchronization. For the following discussion we assume that the clocks of all involved hosts are synchronized with parameter  $\text{syn}$ . That is, the maximal clock offset between any pair of hosts is at most  $\text{syn}$ .

All together, the following parameters are relevant for time stamp calculation:

1.  $\text{syn}$  : maximum clock deviation of two clocks (max. offset)
2.  $t_S$  : local time of the sender
3.  $n$  : number of SG-MIXes
4.  $T_i$  : delay time of SG-MIX  $i$
5.  $d_{ij}$  : unidirectional transmission delays

Define SG-MIX 0 as the sender. The time stamps are calculated as follows:

$$TS_i^{min} = t_S + \sum_{j=1}^{i-1} T_j + \sum_{j=1}^i d_{j-1,j}^{min} - \text{syn}$$

$$TS_i^{max} = t_S + \sum_{j=1}^{i-1} T_j + \sum_{j=1}^i d_{j-1,j}^{max} + \text{syn} \quad .$$

Therefore, the length of the time window is given by

$$\Delta t = TS_i^{max} - TS_i^{min} = 2\text{syn} + \sum_{j=1}^i \Delta d_{j-1,j}$$

with

$$\Delta d_{i,j} = d_{j-1,j}^{max} - d_{j-1,j}^{min} \quad .$$

The length of the time windows increases with the number of SG-MIXes used. Additionally it depends on the propagation delay jitter and the accuracy of synchronized clocks.

The crucial measure for the quality of the time stamps is the question whether an  $(n - 1)$ -attack can be fended off. Let  $T_{empty}$  denote the time that is needed by an attacker to flush a SG-MIX if all incoming packets are blocked. Then the time windows should satisfy  $\Delta t \leq T_{empty}$ . The security parameter  $\mu$  must be decreased until this requirement is fulfilled with a given probability. Of course, this results in longer security delays and the time an attacker needs to flush a SG-MIX increases.

**Size of the Anonymity Set.** As the last subsection has shown how to calculate the timestamps, we will now prove our statement that the size of the anonymity set will always be maximal with a high propability.

The anonymity set  $\mathcal{U}$  produced by a SG-MIX for a single message  $\mathcal{M}$  is composed of the recipients of the messages already present at the SG-MIX at the arrival of  $\mathcal{M}$  and the messages which are received by the SG-MIX during the same busy period. This property of the anonymity set is obvious. For no message which arrives after  $\mathcal{M}$  and departs before the SG-MIX runs empty, the attacker can exclude that it is the one of interest.

To determine the mean size of the anonymity set we can follow the already used model of an  $M/M/\infty$  server with arrival rate  $\lambda$  and service rate  $\mu$ . The life cycle of such a system consists of alternating busy and idle periods. The process  $\{N_t\}$ , denoting the number of messages in the SG-MIX at time  $t$ , is regenerative due to the memoryless property of the exponential distribution. According to the fundamental theorem for regenerative processes the following relation between the probability  $\pi_0$  that the system is empty, the expectation of the length of an

idle period  $E(t_i)$ , and the mean length of a cycle  $E(t_i + t_b)$  ( $t_b$  denotes the length of the busy period) holds:

$$\pi_0 = \frac{E(t_i)}{E(t_i + t_b)} \quad . \quad (3)$$

In addition the following properties hold:

1.  $E(t_i + t_b) = E(t_i) + E(t_b)$ , due to the linearity of the expectation
2.  $\pi_0 = e^{-\lambda/\mu}$
3.  $E(t_i) = \frac{1}{\lambda}$ , because the  $t_i$ 's are distributed equally according to the exponential distribution  $\mathcal{E}(\lambda)$

This results in

$$e^{-\lambda/\mu} = \frac{1/\lambda}{1/\lambda + E(t_b)} \quad ,$$

and therefore

$$E(t_b) = \frac{e^{\lambda/\mu} - 1}{\lambda} \quad .$$

Hence the mean number  $Y$  of served packets during a busy period is:

$$E(Y) = \lambda \cdot E(t_b) + 1 = e^{\lambda/\mu} \quad . \quad (4)$$

Now we can examine a single message  $\mathcal{M}$  arriving during a busy period in which  $n$  messages are processed.

$X$  denotes the number of messages in the busy period arriving after  $\mathcal{M}$  and  $Y$  denotes the total number of messages served during the busy period.

Because the message arrivals are Poisson distributed and therefore fulfill the memoryless property the following relation holds:

$$\mathcal{P}(X = m | Y = n) = \begin{cases} \frac{1}{n} & : m \leq n \\ 0 & : else \end{cases} \quad .$$

This leads to a conditional expectation of

$$E(X | Y = n) = \sum_{m=1}^n \frac{m}{n} = \frac{n+1}{2} \quad .$$

The summation over  $Y$  (theorem of total probability) results in:

$$E(X) = \sum_{n=1}^{\infty} \frac{n+1}{2} \cdot P(Y = n) = \frac{1}{2} E(Y) + \frac{1}{2} \quad . \quad (5)$$

Because the expectation of the number of message already in the SG-MIX at the arrival of  $\mathcal{M}$  is  $\lambda/\mu$ , the expectation of the size of the anonymity set is:

$$E(|\mathcal{U}|) = \frac{\lambda}{\mu} + E(X) = \frac{\lambda}{\mu} + \frac{e^{\lambda/\mu} + 1}{2} \quad . \quad (6)$$

Hence we can deduce that the size of the anonymity set will always be maximal with high propability.

**The  $M/M/\infty$  Model.** All security statements in this paper are based on the assumption of using an infinite server system ( $M/M/\infty$  model). Obviously this assumption is not realistic for a concrete realization of a SG-MIX. A real-world SG-MIX can only approximate the ideal model depicted above, because it can only serve a finite number of packets. Therefore it must be analyzed according to the multiple server system model ( $M/M/n$  model). Using the  $M/M/n$  model it is possible that an arriving packet finds a situation in which all servers within a SG-MIX are busy. The probability that an arriving packet finds no idle server can be calculated by Erlang's C formula [15]. Taking this formula and the parameters  $\mu$  and  $\lambda$  into account it is possible to design a concrete SG-MIX in a way that fulfills the demanded level of approximation to the ideal model.

Nonetheless the potential ranges of the arrival rate  $\lambda$  and the service rate  $\mu$  must be taken carefully into account during the design of an SG-MIX to provide a reasonable servicing capacity.

And finally, as every real-world implementation can only provide a limited servicing capacity, the SG-MIX is like every other MIX vulnerable to denial of service attacks, in which an attacker floods the SG-MIX with packets. As a consequence, most of the other packets will time out during such an attack, but this will not go undetected.

## 5 Conclusions

The currently known basic anonymity techniques depend on identity verification, but provide perfect anonymity. Following the general idea of complexity-theoretic security we have introduced the notion of probabilistic anonymity in order to find a scheme which does not need these identity verification procedures.

Probabilistically secure algorithms provide untraceability with a probability which depends exponentially on a publicly known parameter. We have proposed the Stop-And-Go-MIX as a probabilistically secure protocol, where  $\mu$  is given as a publicly known security parameter. This parameter can be chosen to fulfil the practical considerations and security demands. The users can predict the delays of their message very accurately and hence time stamp protocols can be applied to counter  $(n - 1)$ -attacks. For the same reason the network delay can be calculated from packet turnaround times and congestion control algorithms can be employed, allowing the use of SG-MIX on ISO/OSI layer 3.

*Untraceable Return Address* can also be supported by the SG-MIX for an open system so fully anonymous communication is possible.

## 6 Acknowledgement

We would like to thank Andreas Pfitzmann for his support of this work and many fruitful discussions.

## References

- [1] D.L. Chaum, "Untraceable Electronic Mail, Return Addresses, and Digital Pseudonyms", *Comm. ACM*, Feb. 1981, Vol. 24, No. 2, pp. 84-88.
- [2] D.L. Chaum, "The Dining Cryptographers Problem: Unconditional Sender and Recipient Untraceability", *J. Cryptology*, Vol. 1, No. 1, Springer-Verlag, 1988, pp. 65-75.
- [3] D.A. Cooper and K.P. Birman, "Preserving Privacy in a Network of Mobile Computers", 1995 Symposium on Research in Security and Privacy, IEEE Computer Society Press, Los Alamitos, 1995, pp. 26-38.
- [4] W. Diffie and M.E.Hellman, "New Directions in Cryptography", *IEEE Transactions on Information Theory*, 22 (1976), pp. 644-654.
- [5] D.J. Farber and K.C. Larson, "Network Security Via Dynamic Process Renaming", Fourth Data Communication Symp., Quebec City, Canada, Oct. 1975, pp. 8-18.
- [6] A. Fasbender, D. Kesdogan, and O. Kubitz, "Analysis of Security and Privacy in Mobile IP", 4th International Conference on Telecommunication Systems, Modelling and Analysis, Nashville, 1996.
- [7] A. Fasbender, D. Kesdogan, and O. Kubitz, "Variable and Scalable Security: Protection of Location Information in Mobile IP", VTC'96, Atlanta, 1996.
- [8] H. Federrath, A. Jerichow, D. Kesdogan, A. Pfitzmann, and D. Trossen, "Minimizing the Average Cost of Paging on the Air Interface - An Approach Considering Privacy", *IEEE VTC' 97*, May 1997, Phoenix, Arizona.
- [9] S. Goldwasser and S. Micali, "Probabilistic Encryption", *Journal of Computer and System Science* 28 (1984), pp. 270-299.
- [10] D.M. Goldschlag, M.G. Reed, and P.F. Syverson, "Hiding Routing Information", *Information Hiding*, Springer-Verlag LNCS 1174, 1996, pp. 137-150.
- [11] C. Gülcü and G. Tsudik, "Mixing Email with Babel", *Proc. Symposium on Network and Distributed System Security*, San Diego, IEEE Comput. Soc. Press, 1996, pp. 2-16.
- [12] S. Hoff, K. Jakobs, and D. Kesdogan, "Secure Location Management in UMTS", *Communications and Multimedia Security*, Proceedings of the IFIP TC6/TC11 International Conference on Communications and Multimedia Security at Essen, Germany, September 1996, Chapman & Hall, ISBN 0-412-79780-1.
- [13] D. Kesdogan, H. Federrath, A. Jerichow, and A. Pfitzmann, "Location Management Strategies increasing Privacy in Mobile Communication Systems", *IFIP SEC 96*, 12th International Information Security Conference, May 1996, pp. 39-48.
- [14] D. Kesdogan and X. Fouletier, "Secure Location Information Management in Cellular Radio Systems", *IEEE Wireless Communication Systems Symposium WCSS 95*, Wireless Trends in 21st Century", New York, 1995, pp. 35-40.
- [15] L. Kleinrock, "Queuing Systems, Vol. I: Theory", John Wiley & Sons, 1975.
- [16] P.A. Karger, "Non-Discretionary Access Control for decentralized Computing Systems", Master Thesis, MIT, Laboratory for Computer Science, Report MIT/LCS/TR-179, 1977.
- [17] A. Pfitzmann, "Dienstintegrierende Kommunikationsnetze mit teilnehmerüberprüfbarem Datenschutz", IFB 234, Springer-Verlag, Heidelberg, 1990.
- [18] A. Pfitzmann and M. Waidner, "Networks without User Observability", *Computers & Security* 6, 1987, pp. 158-166.

- [19] A. Pfitzmann, B. Pfitzmann, and M. Waidner, "ISDN-MIXes: Untraceable Communication with Very Small Bandwidth Overhead", Information Security, Proc. IFIP/SEC 91, Brighton, UK, 15-17 May 1991, D.T. Lindsay, W.L. Price (eds.), North-Holland, Amsterdam 1991, pp. 245-258.
- [20] B. Pfitzmann, "Digital Signature Schemes. General Framework and Fail-Stop Signatures", Springer-Verlag LNCS 1100, Springer 1996.
- [21] M.G. Reed, P.F. Syverson, and D.M. Goldschlag, "Protocols using Anonymous Connections: Mobile Applications", 1997 Workshop on Security Protocols, Paris, France, April 1997.
- [22] M.K. Reiter and A.D. Rubin, "Crowds: Anonymity for Web Transactions", DIMACS Technical Report 97-15, <http://www.research.att.com/projects/crowds/>.
- [23] R.L. Rivest, A. Shamir, and L.M. Adleman, "A Method for Obtaining Digital Signatures and Public-Key Cryptosystems", Communications of the ACM, 21 (1978), pp. 96-99.