Proc. GI/ITG-Conference "Kommunikation in verteilten Systemen" (Communication in Distributed Systems); Feb. 20-22, 1991, Mannheim, Informatik-Fachberichte 267, Springer-Verlag, Heidelberg 1991, 451-463

ISDN-MIXes: Untraceable Communication with Very Small Bandwidth Overhead

Andreas Pfitzmann, Birgit Pfitzmann, Michael Waidner

Institut für Rechnerentwurf und Fehlertoleranz, Universität Karlsruhe
P.O.Box 6980, D-7500 Karlsruhe 1, Federal Republic of Germany
Phone: ++49-721-608-4218, Fax: ++49-721-370455
E-mail (CSnet): WAIDNER@IRA.UKA.DE

Abstract

Untraceable communication for services like telephony is often considered infeasible in the near future because of bandwidth limitations. We present a technique, called ISDN-MIXes, which shows that this is not the case.

As little changes as possible are made to the narrowband-ISDN planned by the PTTs. In particular, we assume the same subscriber lines with the same bit rate, and the same long-distance network between local exchanges, and we offer the same services.

ISDN-MIXes are a combination of a new variant of CHAUM's MIXes, dummy traffic on the subscriber lines (where this needs no additional bandwidth), and broadcast of incoming-call messages in the subscriber-area.

1 Introduction

The need to keep communication untraceable, i.e. to keep secret who communicates with whom, has been discussed, e.g., in [Chau_81, Cha8_85, PfWa_86, PfPW_88]. Untraceable communication for services like telephony is often considered infeasible in the near future because of bandwidth limitations. We present a technique which shows that this is not the case.

Requirements: The bandwidth limitation one must deal with is that of the twisted pairs of copper wires connecting most subscribers to local exchanges, since these wires are a major investment and cannot be replaced very quickly. We presuppose that transmission on them is digitalized, and the net bit rate offered at the interface to the participant is 144 kbit/s duplex, as in CCITT's ISDN-standards [Tane_88, Kah2_85, Bock_88]. This digitalization is planned by the PTTs anyway and, in most cases, needs no additional signal regenerators.

We take for granted that two independent, bit-transparent, duplex 64-kbit/s channels are to be offered to the subscriber. Each channel can, e.g., be used for PCM-coded telephony. The remaining 16 kbit/s are available for signalling, including error control. (Like, e.g., the Deutsche Bundespost Telecom, we do not consider additional packet-switched services in the signalling channel.)

Our technique also respects that almost no additional delay can be tolerated on an established channel, and we will set the limit on the establishment of a channel to 3 seconds.

We need no changes in the long-distance network between local exchanges.

Basic technique: MIXes [Chau_81] are the only known basic mechanism for untraceable communication which offers any chance of being adaptable to this situation. In particular, in the DC-net [Chau_88] (which offers better untraceability than MIXes and has other advantages, too), each station must transmit at least half as many bits as all participants together want to send [Pfit_90 p. 98f]. Thus it is not adaptable.

Overview: ISDN-MIXes are a combination of

- a new variant of MIXes,
- dummy traffic on the subscriber lines (where this needs no additional bandwidth),
- broadcast of incoming-call messages in the subscriber-area.

Of course, we assume that the data itself is already end-to-end encrypted, if it is confidential.

In Ch. 2, we sketch MIXes as far as we need them, and introduce notation. In Ch. 3, we introduce MIX-channels, a MIX-technique that can handle a continuous stream of data almost without delay. We also explain the remaining problem with MIX-channels. This is mainly the delay in releasing connections. In Ch. 4 we present the complete technique of ISDN-MIXes. Ch. 5 contains results of a performance evaluation, and Ch. 6 a summary and the remaining problems.

For shortness, we describe most actions as those of the subscribers, although most of them would be performed by their network terminations or terminal equipment.

Most of the following is described in more detail in [PfPW1_89].

2 MIXes

MIXes are a cryptographic technique for untraceable communication, originally introduced for electronic mail [Chau_81] (cf. Fig. 1). MIXes tolerate that all lines may be tapped.

Idea of MIXes and assumptions: Each message is sent over a series of independent stations, called MIXes. A MIX collects a number of messages, called batch, discards repeats, changes the outlooks of the remaining messages, and outputs them in a different order. The change of outlook is a cryptographic operation. Since the recipient must be able to read the message, the sender and/or the recipient must perform cryptographic operations inverse to those of the MIXes.

Outsiders can only observe the path of a message if

- they can break the cryptographic operations
- or they have the cooperation of all the MIXes on the path (or, instead of any of the MIXes, all other participants who contributed a message to the same batch at this MIX).

Also, if only the sender performs cryptographic operations inverse to those of the MIXes, he knows all the different outlooks of the message, and can thus follow its path to the recipients; and vice versa.

The specific scheme needed in the following is slightly different from those in [Chau_81].

Cryptosystems needed: We need a symmetric and an asymmetric cryptosystem. The latter must be secure against active attacks. Hence at present, RSA should be used [RSA_78]; and before encryption, a random part should be added and well mingled with the message [PfPf_89].

The symbol k_x with some subscript x will always denote a key of the symmetric cryptosystem, c_x and d_x public and private keys of the asymmetric cryptosystem; en- and decryptions of a message N are denoted by $k_x(N)$, $k_x^{-1}(N)$, $c_x(N)$, and $d_x(N)$, resp. The subscript denotes the owners of the key.

We will use **hybrid encryption of minimal length**: Assume A (Alice) wants to send a message N to a recipient B (Bob). A chooses a key k_{AB}, encrypts k_{AB}, and as much of N as fits into the same block, with c_B, and encrypts the rest of N with k_{AB}. This will be denoted by $c_B{}^*(N)$.[1]

[1] For concreteness, and for the performance evaluation, we assume that the symmetric cryptosystem is a DES-variant [DES_77] with 128-bit-keys. In hybrid encryption, the randomly chosen key k_{AB} also serves as the random part for the encryption with c_B. Mingling k_{AB} and the part N' of N that fits into the RSA-block is performed by encryption with DES with a fixed and globally known key.

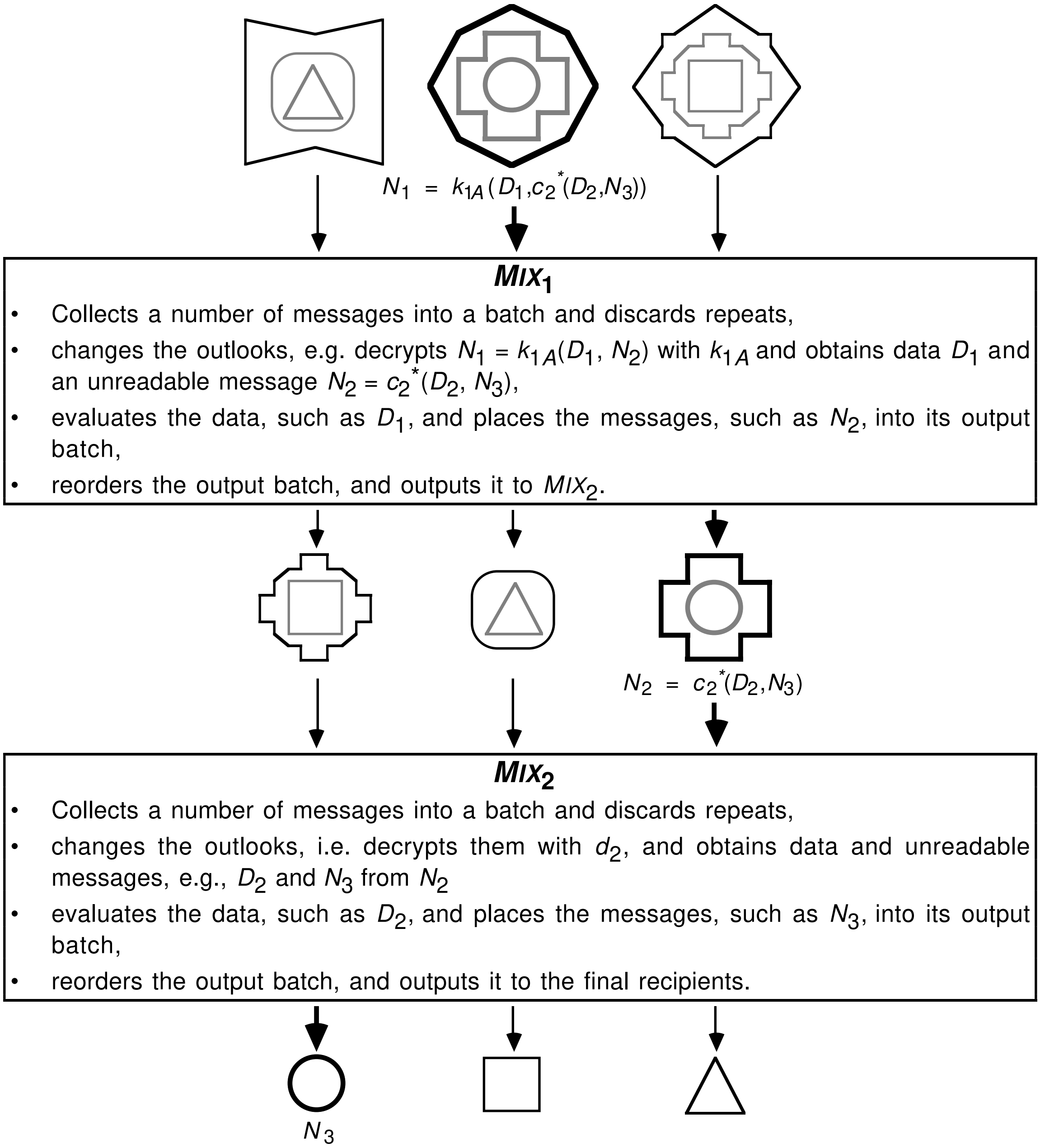

Fig. 1 Example of the basic MIX-scheme, with a cascade of two MIXes and a batch of three messages. Encryption is represented by packing into boxes.

A basic MIX-scheme: Assume A wants to send a message N to B. We prescribe that the sequence of MIXes which A must use for this purpose is fixed, say $MIX_1, \ldots, MIX_m$.[2] Such a sequence will be called a **MIX-cascade**. Each MIX_i, $i \geq 2$, has initially chosen a key pair (c_i, d_i) and published c_i; and A shares a key k_{A1} with MIX_1.

In a basic MIX-scheme, A encrypts the message several times, and each MIX decrypts it once (cf. Fig. 1). We provide for the case that A additionally wants to pass data D_i to each MIX_i. Thus A recursively forms the following encrypted messages, where N_i is the message which MIX_i will receive:

$$\begin{aligned} N_{m+1} &:= N \\ N_i &:= c_i^*(D_i, N_{i+1}) \quad \text{for } i = m, m\text{-}1, \ldots, 2, \end{aligned} \tag{1}$$

[2] This does not reduce untraceability, rather the contrary [Pfit_90]. It also reduces the problem that all messages of a batch must be of equal length, and timing problems between the MIXes. (Both are critical factors for performance.)

If one already wants to see the basic MIX-scheme (and also the MIX-channels in Ch. 3) in an ISDN-context, suppose A and B live in the same subscriber area and the MIXes are situated at the local exchange.

$$N_1 \quad := k_{1A}(D_1, N_2)$$

and sends N_1 to MIX_1. N_1 is called a MIX-**input-message**.

Each MIX_i receives the message N_i from MIX_{i-1} or A, resp.; it decrypts N_i with k_{1A} or d_i, resp., and strips D_i off. One purpose of D_i is to include a time-stamp; this eliminates the need to compare messages of different batches for repeats.

Note that the length of N_1 only grows linearly with m.

Protection of the recipient: The basic MIX-scheme does not prevent A from tracing B. In principle, there are two possibilities to protect the recipient. Variants of both will be needed in the following:

1. N is **broadcast**. For B to identify N as addressed to him, N must contain an **implicit address** (cf., e.g., [PfWa_86]). This can be provided together with end-to-end encryption: The unencrypted message inside N must fulfil a redundancy predicate; and B decrypts each message and tests this predicate. This is called an **invisible implicit address**. If A and B have communicated before, they may have exchanged a **visible implicit address**: This is just a random number, which A prefixes to the message like a normal address.
2. An untraceable return address may have been constructed by B in advance and passed to A [Chau_81]. For use with channels, we change this idea so much that we need not repeat the original scheme here.

3 MIX-Channels

In this chapter, we adapt MIXes so that they can handle a continuous stream of data almost without delay.

We only consider simplex channels, since a duplex channel can be constructed as two independent simplex channels. MIX-channels will only serve as building-blocks for ISDN-MIXes. Hence we omit some details which would have to be changed in Ch. 4.

A MIX-channel will consist of two parts (Fig. 2): a MIX-sending-channel from the sender A to MIX_m (Sect. 3.1), and a MIX-receiving-channel from MIX_m to the recipient B (Sect. 3.2). The connection is described in Sect. 3.3.

3.1 MIX-sending-channels

In most MIX-variants, N is assumed to be a single message, or even only a message block. But the basic MIX-scheme from Ch. 2 can also be used if N is a stream of data of arbitrary length, if the symmetric cryptosystem is a streamcipher.[3]

However, there are two problems at the beginning of the data:

- There is some bandwidth expansion.
- Each MIX must wait for a whole block of data to arrive before it can start decrypting it using the asymmetric cryptosystem. This causes considerable delay.

They imply that this scheme cannot be used directly for the data in the 64-kbit/s-channels in an ISDN.

Therefore, we will take the asymmetrically encrypted part of the MIX-input-message out into the signalling channel. There, it is sent as a sending-channel establishment message (**SendEstab-message**) before the user data start. The innermost part, N, of the SendEstab-message only contains information which MIX_m needs to identify the corresponding MIX-receiving-channel to B (Sect. 3.3).

---

3 The streamcipher should not propagate transmission errors. Then the channels offered by the MIXes to the subscribers have similar error characteristics as the channel offered in a normal ISDN. If necessary, errors can be treated in higher layers in the normal way.

If the streamcipher is constructed from a blockcipher like DES, ECB-mode (electronic codebook mode, where each block is encrypted individually) should not be used for cryptographic reasons [PfPf_89].

A suitable choice is OFB-mode (output-feedback mode). There, each arriving bit can be decrypted at once [DaPr_89].

In addition to the normal mix operations, each MIX_i, upon receiving the N_i-part of a SendEstab-message, makes provisions for the following user data in the 64-kbit/s-channel (Fig. 2):[4]

- MIX_i reserves an outgoing 64-kbit/s-channel C_i to MIX_{i+1} for the following data.[5]
- It tells the position of C_i to MIX_{i+1}, together with the decrypted SendEstab-message N_{i+1}.
- It stores the correspondence between C_i and the incoming channel C_{i-1}. (C_{i-1} is the channel which MIX_{i-1} has reserved for the same data, and MIX_{i-1} has just told MIX_i about it, together with N_i.)
- It stores the private key k_i which it has found in N_i, as belonging to this correspondence.

Now MIX_i can immediately decrypt each bit of data arriving on C_{i-1} with k_i, and send it out on C_i. This is called a **MIX-sending-channel**.

3.2 MIX-receiving-channels

Of course, with the assumed bandwidth limitations, it is impossible to broadcast the user data of a MIX-sending-channel, even in a small area containing *B*. Thus, so far, only *A* is untraceable by *B* (and, as always, their relation is untraceable by outsiders). Therefore we need the second half of the MIX-channel, a **MIX-receiving-channel**, to protect *B* from *A*.

To establish a MIX-receiving-channel, the recipient *B* sends a receiving-channel establishment message (**RecEstab-message**) through the MIX-cascade. The RecEstab-message is formed just like a SendEstab-message; i.e., it mainly delivers a key k'_i to each MIX_i.

However, the channel is established and used in the reverse way: Channels C'_i from MIX_{i+1} to MIX_i are reserved. In particular, C'_0 leads from MIX_1 to *B*. When user data arrive on C'_i, MIX_i encrypts them (not "decrypts") and forwards it to MIX_{i-1} on C'_{i-1}. Thus *B* receives multiple-encrypted data and decrypts it with all his keys k'_i.[6]

3.3 Connecting the two halves

Each half of a MIX-channel protects only the participant who has established it. Thus, to prevent both *A* and *B* from being able to trace each other, we connect the halves at MIX_m. The resulting channel is called a **MIX-channel**.[7]

Hence, MIX_m must know which channels to connect. Hence it must receive some common information in the *N*-parts of both the SendEstab- and the corresponding RecEstab-message. We will call it a **label** l_{AB}. The label, in its turn, must be known to both *A* and *B*; e.g., it may have been chosen by one of them and told to the other one in a short normal MIX-input-message (cf. Ch. 2) which was broadcast on the signalling channel.

---

4 Small deviations from this procedure for MIX_1 and MIX_m are not described, since they will be changed in the following sections.

5 This may, e.g., be a position in a PCM-frame. It is sensible to sort the outgoing channels in the same way as the corresponding SendEst-messages in the output batch.

6 This is a variant of the untraceable return addresses in [Chau_81], cf. Ch.2. The difference is that we do not pass the 'return address' to *A* at all, but that *B* himself establishes a return channel. This will foil active attacks on *B* by *A* in Sect. 4.3.

7 This resembles an idea in [Chau_81 p. 85], but there, one message would be sent containing parts formed by *A* and *B*, whereas here, two different channel establishment messages are actually sent.

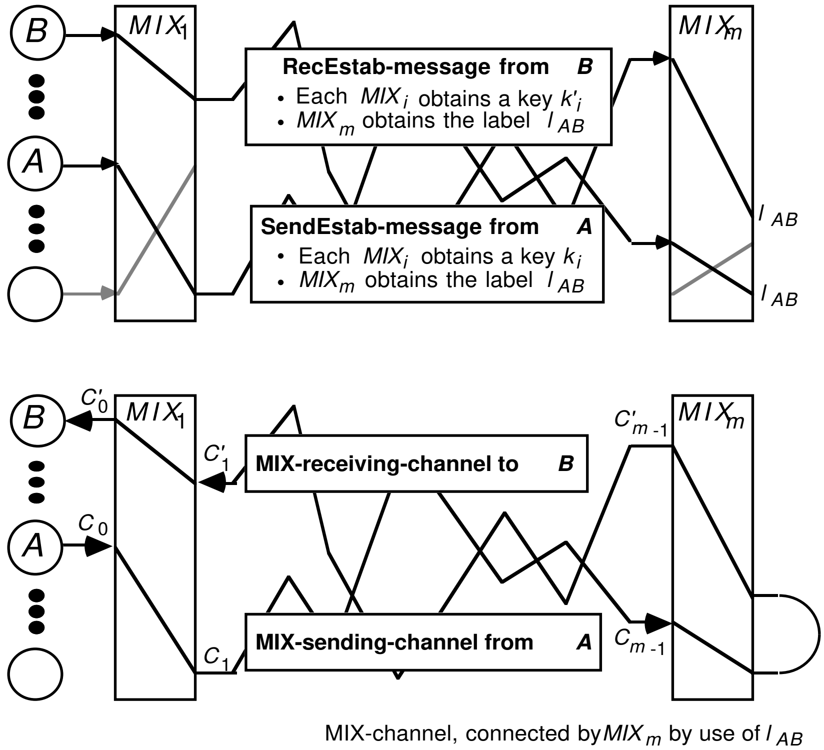

Fig. 2 MIX-channel: First *A* and *B* send SendEstab- and RecEstab-messages, resp., in the signalling channel (upper picture); later they have a MIX-sending- and a MIX-receiving-channel, resp., in the 64-kbit/s-channel (lower picture). The latter are mixed with the keys delivered in the former, and connected according to their common label.

3.4 Problems with pure MIX-channels: Releasing connections

If MIX-channels were used directly in the narrow-band ISDN, users might have to wait a very long time for the release of their channels. The reason for this is that messages mixed together must be of equal length. In the case of channels, this means: A sufficiently large number of channels must be established with the same batches of SendEstab- and RecEstab-messages; and the user data on them must start at the same time and end at the same time. Otherwise, an observer who sees the user data on *A*'s MIX-sending-channel end, and a moment later the data arriving on *B*'s MIX-receiving-channel end, too, could guess that these channels are connected.

It might be difficult enough to obtain enough connections starting at the same time, if call-establishment is bounded by 3 seconds, and it seems highly unlikely that enough of them end at the same time by themselves. Therefore, some users would have to wait for others before being allowed to release their channels (of course, this would not mean that the people must keep talking, but that the terminal equipment must send encrypted nonsense afterwards). However, each user only has two channels. Thus it cannot be tolerated that they are blocked.

4 ISDN-MIXes

In this chapter, we present the complete technique of ISDN-MIXes. It is based on the MIX-channels from Sect. 3, and it solves the problem described in Sect. 3.4.

4.1 Remarks on the network hierarchy

The following sections are simplified if we make some choices concrete now, although they could (more or less) be derived from the performance considerations at the end:

There will be one MIX-cascade at each local exchange L. Each subscriber A has a personal subscriber line with a local exchange L_A, and A will use the MIX-cascade at L_A. Thus we can assume that A has exclusive access to a 144-kbit/s duplex channel with the first MIX at L_A. The MIXes only have the function of mixing, and the local exchange keeps its normal functions. Functionally (not locally), such a local exchange has two halves (Fig. 3): The first half administers the communication between the subscribers and MIX_1, the second half the communication between MIX_m and the long-distance network.

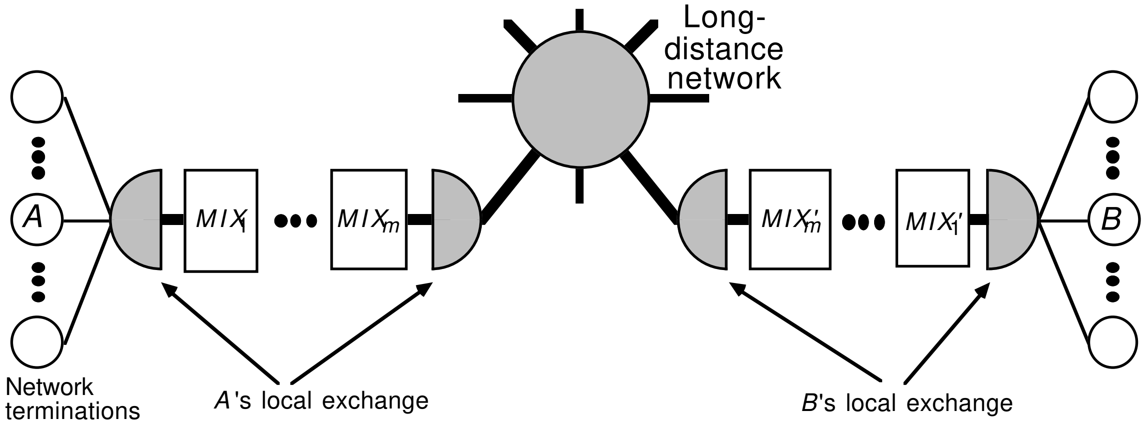

Fig. 3 Functional positions of ISDN-MIXes

Complete untraceability will be achieved within the set of all subscribers at one local exchange. This means that calls can be traced to a local exchange, but an observer obtains no additional information about which specific subscriber the call belongs to (as long as the assumptions mentioned in Ch. 2 hold). This set is called an **untraceability set**.[8]

ISDN-MIXes need no changes in the long-distance network between the local exchanges.

4.2 Solution: Time-slice channels + dummy traffic + incoming-call broadcast

The solution to the channel-release problem is to divide a connection between A and B into a sequence of **time-slice channels**, which look completely unrelated to everybody except for A and B. With each new time-slice, participants can release connections and/or establish new ones.

The time-slices also help us to solve the problem that enough channels must start at the same time: Each participant who does not use a channel during a time-slice establishes a dummy time-slice channel instead. This costs no additional bandwidth, since this channel is on the subscriber line only.

Time-slice channels: More precisely, during each time-slice, each subscriber A maintains two MIX-sending-channels and two MIX-receiving-channels, cf. Ch. 1. Each of them leads through $MIX_1, \ldots, MIX_m$ at the local exchange L_A, and ends at L_A. They are called **time-slice sending-channels** and **time-slice receiving channels**, resp.

Thus, before each time-slice, A must send two SendEstab-messages and two RecEstab-messages. MIX_m passes the innermost part, N, of these messages to L_A. One part of N is the label which L_A needs to connect the channels. (Thus, in contrast to Ch. 3, MIX_m does not evaluate N itself, because connecting channels is no MIX-function.)

---

8 In any case, it is good to have fixed untraceability sets, since otherwise someone who exchanges several messages with the same partner could eventually trace him by intersecting the untraceability sets.

If A has a real connection with B, A's SendEstab-message contains the address of B's local exchange L_B, i.e., the area code, and the same label as B's RecEstab-message and vice versa.

If A has no real connection on one of the two 64-kbit/s channels, the corresponding SendEstab- and RecEstab-messages carry the same label, i.e., A sets up a simplex MIX-channel with herself. Hence these dummy-channels look exactly like local calls. This makes local calls completely unobservable.

The data in the SendEstab- and RecEstab-messages is summarized in Fig. 4.

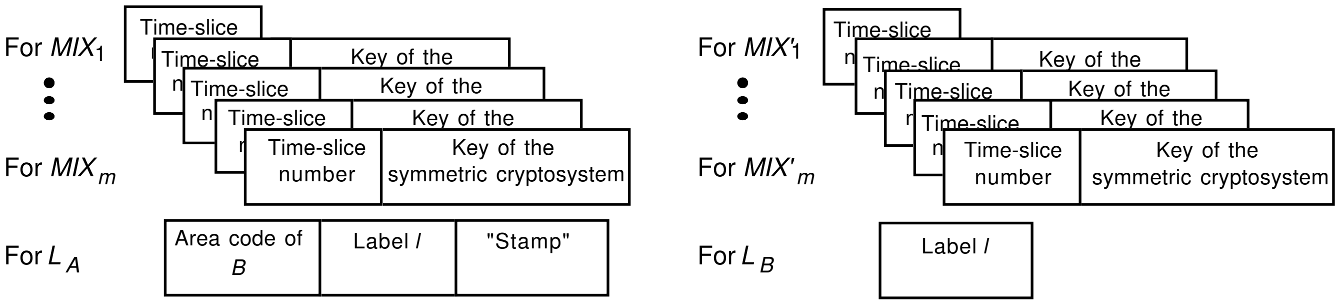

Fig. 4 Data needed by the MIXes and the local exchanges for establishing a time-slice channel from A to B. The data in the SendEstab-message from A are shown on the left, those in the RecEstab-message from B on the right. The "stamp" will be explained in Sect. 4.5.

The local exchanges should not release an established channel in the long-distance network automatically after each time-slice. Instead, they wait to see if a SendEstab-message with the same area code for the next time-slice arrives. Thus for each call, a channel through the long-distance network needs to be switched at most once.

Establishing calls: Of course, some coordination between A and B is needed. In particular, if A wants to call B, A needs a way of telling B that B must stop setting up dummy-channels with himself and set them up to meet A's channels instead. Since B does not know about the arrival of such messages, he cannot make provisions for them in the same way as for channels. Thus these **incoming-call messages** are broadcast in B's untraceability set.

Apart from, or rather within, the implicit address, an incoming-call message contains the number t_0 of the time-slice where the call is to start.

A and B must also use the same labels in the corresponding SendEstab- and RecEstab-messages. This can be achieved efficiently if A includes a **seed** s for a pseudo-random number generator in the incoming-call message. From s, all the labels are derived. Denote the i-th label derived from s by l_i. For example, the labels with even index i can be used for the simplex time-slice channels from A to B, and those with odd i for the channels from B to A. The labels of different time-slices look totally unrelated to everybody except for A and B.

The incoming-call message may also contain a key k_{AB} for fast end-to-end-encryption; but to keep the message short, k_{AB} can also be derived from the seed s.

Of course, A sends this incoming-call message as N in a MIX-input-message, called **call-establishment message**. It must be possible to send two call-establishment messages every time-slice, and A should be untraceable in her whole untraceability set for call-establishment messages. Hence every subscriber must send two dummy call-establishment messages every time-slice, if they have no real ones. Dummy call-establishment messages are marked so that L_A can discard them (Fig. 5). Thus they do not go out into the long-distance network and are never broadcast.

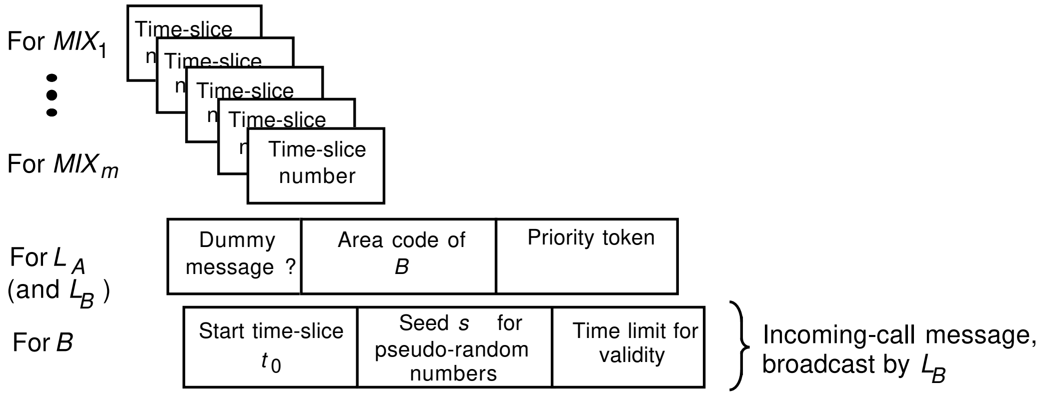

Fig. 5 A call-establishment message. The priority token and the time limit will be explained in Sect. 4.3.

A successful call-establishment is shown in Fig. 6.

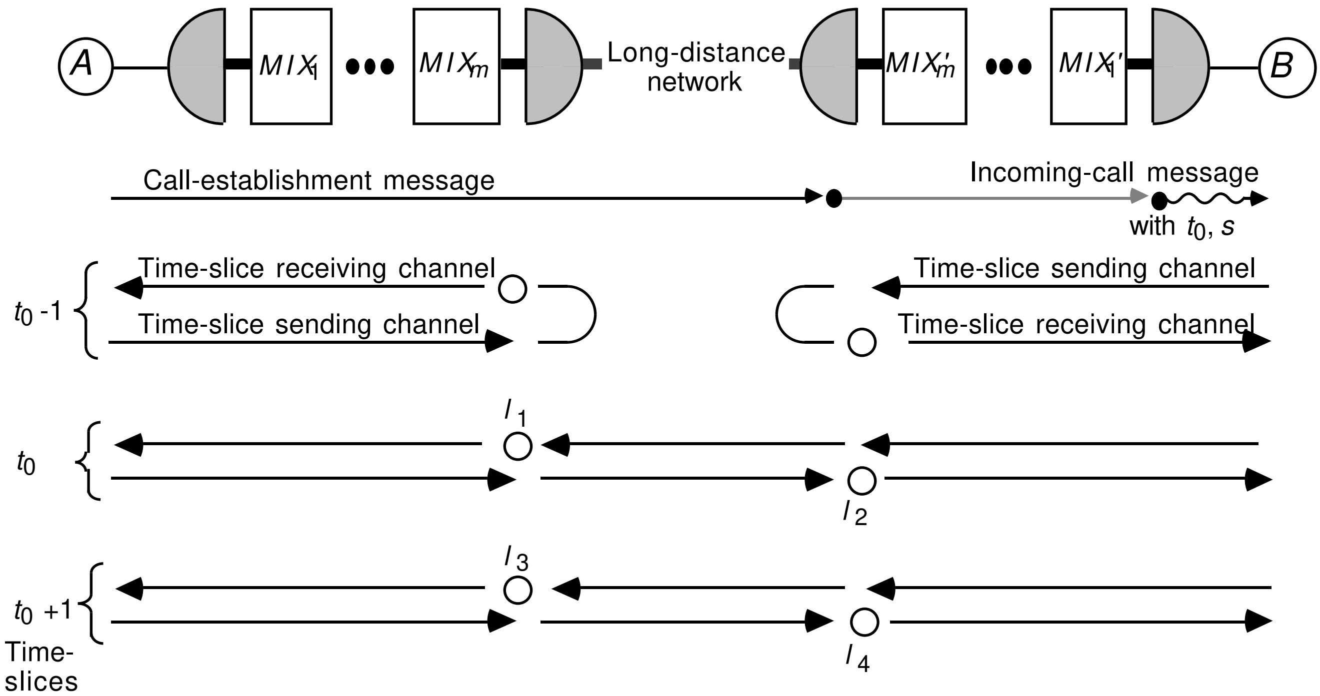

Fig. 6 Call-establishment in the easiest case. The incoming-call message does not pass the MIX-cascade at L_B. Instead, it is broadcast by L_B. The SendEstab- and RecEstab-messages corresponding to each time-slice channel are not shown.

4.3 Rejected calls and active attacks

Rejected calls and a related attack: So far, A would be traceable if B cannot accept the call, since in that case, no data would arrive on A's time-slice receiving-channel. This can easily be corrected: MIX_m (at L_A) must fill up all time-slice receiving-channels for which no time-slice sending-channel with the same label exists (e.g., with zeroes).

Now B can also accept the call some time-slices later than previewed; of course, he must then use a different starting label.[9] A starting signal within the (still unused) 64-kbit/s channel is provided by MIX_m to

[9] If B accepts in time-slice t_0+t, he must start with the labels l_{2t+1}, l_{2t+2}. B need not compute all the intermediate labels, if pseudo-random functions are used instead of pseudo-random generators [GoGM_86], or a key of a symmetric block cipher is exchanged. The pseudo-random function or the block cipher is applied to the time-slice number (concatenated with a bit for even/odd, cf. Sect. 4.2). The result (or part of it) is taken as the label.

tell A when the connection has been established. The signal is recognized by the terminal equipment of A after decryption with all the keys k_i. To avoid that B accepts the call when A has already given up waiting (i.e., she uses different labels for her time-slice channels again), A may include a time-limit in the incoming-call message.

Any MIX could use the same idea of leaving a channel empty for an active attack to see where this channel leads to; hence each MIX_i, not just MIX_m, must fill up all empty channels (in either direction).

Some more difficult active attacks, however, cannot be prevented completely. They all exploit situations where an attacker knows that an anonymous participant X should react in some way. The attacker then prevents a real participant A from reacting, e.g., by cutting her line. Then, if X reacts, $X \neq A$ has been proved, and if X does not react, it is probable that $X = A$:

Attack on A as a sender: If MIX_1, or an active attacker on A's subscriber line, changes the content of A's channel and colludes with B, they can test whether A is B's current communication partner, because B will receive nonsense in this case. However, since this attack needs the collusion of B and one must disturb lots of participants to carry it out, this seems acceptable in practice.[10]

Attack on the broadcast of incoming-call messages: A second attack is to disturb the broadcast of incoming-call messages: Assume L_A, or an active attacker on the subscriber lines, causes an incoming-call message to be handed to A only. Then, if the call is accepted, A must be the recipient. This attack can be detected: Each network termination sends a digital signature [GoMR_88] of all messages received back to each MIX, e.g., once per time-slice. If a MIX detects an inconsistency, one can try to localize the attacker. To prevent A from falsely claiming that she received wrong messages, the MIXes, too, should sign the messages which are broadcast. However, if A claims that she does not receive any correctly signed messages, and no fault can be localized, one cannot do anything in practice.

Blocking A's ressources: An attacker may, e.g., establish two calls with A. If he can then establish a call with X, too, then $X \neq A$. This kind of attack cannot be prevented. However, it needs the collusion of the communication partner of X again; and as long as A's ressources are not fully used, she knows that she is not a victim of such an attack.

Since an attacker could particularly easily clog the signalling-channel by sending large numbers of senseless call-establishment messages, we introduce priority tokens for them (cf. Fig. 5). The local exchange of the recipient broadcasts incoming-call messages according to their priorities. For the lowest priority, no token is needed. Each participant only obtains a limited number of high-priority tokens from the PTT. They are signed by the PTT, using blind signatures (like in an untraceable payment system [Cha8_85, Chau_89], but free of charge).

4.4 Additional practical considerations

Here we just make some remarks:

Synchronization: Of course, the technique needs global synchronization. This is provided in the ISDN planned by the PTTs, anyway. Note, however, that time-slices start at different times everywhere in the network. E.g., time-slices for time-slice sending-channels at MIX_i start shortly after they started at MIX_{i-1}, whereas MIX_i can only start decrypting SendEstab-messages after MIX_{i-1} has sent the first complete block of them. Also, time-slices for receiving-channels at MIX_m only start when data from time-slice sending-channels everywhere in the long-distance network has had time to arrive. Hence local calls must be buffered during this time.

---

[10] This attack, and the following one, could be prevented in principle by measures which imply that such an attack leads to an immediate end of the communication in the whole network. But then, participants could prevent all members of their untraceability set from communicating. Such techniques for the DC-network are described in [Waid_89, WaPf_89].

Releasing connections: Calls could be released by messages similar to incoming-call messages. Because of bandwidth-limitations, one should use a very long in-band pattern instead, or reserve a channel of 1 bit per time-slice in the signalling-channel for this purpose. The same holds, e.g., for signalling a change of service.

User signalling: If an end-to-end connection has been established, and before there is user data, the connection can be used for signalling. For example, the caller may receive a sign when the recipient picks up the receiver. This does not disturb transparency while sending user data.

Large subscriber-areas: In Ch. 5, we will see that the use of only 16 kbit/s for signalling limits untraceability sets to about 5000 participants. In subscriber areas with more participants, the participants must therefore be virtually partitioned into several fixed untraceability sets. The same local exchange, and the same physical MIX-cascade, can be used. However, each incoming-call message is only broadcast within one untraceability set, and only messages from the same untraceability set are mixed together.

Avoiding call-repetition and unused connections in the long-distance network: The number of incoming-call messages is critical for performance. It is reduced if call-repetition is avoided, e.g. if *A*'s terminal equipment waits for *B*'s to answer, even if *B*'s channels are both busy (cf. Sect. 4.3). In this case, two further changes are needed: First, one cannot expect *A* personally to wait at her phone. Thus she must receive a signal when the connection is established. Secondly, *A*'s MIX-sending channels would unnecessarily use the long-distance network. This can be avoided with a very small decrease in untraceability, if the same label is used for all time-slice channels belonging to one call. Then the channel in the long-distance network need only be established by L_B when *B* has accepted the call.

Connecting subscribers with and without MIXes: *A* and *B* can also communicate, with restricted untraceability, if just one of their local exchanges has a MIX-cascade [PfPW1_89]. If L_A has a MIX-cascade, but L_B has not, this is easy: L_A (or MIX_m at L_A) serves as a gateway. In the call-establishment message, *A* includes a normal address of *B* and the start time-slice t_0. From then on, L_A connects *A*'s MIX-sending- and -receiving-channels with a normal channel to *B*. Hence *A* is untraceable.

If L_B has a MIX-cascade, but L_A has not, the channels can be connected at L_B in the same way. However, this only makes sense if *A* can use an implicit address of *B* in the call-establishment message. If *A* and *B* can use a visible implicit address, it can be coded as a normal telephone number, after the area code. 15 decimal digits should be enough. Thus it could even be used if *A* has a normal telephone. To use an invisible implicit address, *A* needs some computing power, and the address is so long that it must be transmitted in the data channel of a connection between *A* and L_B.

Billing: To bill untraceable connections, local counters in network terminations or an untraceable payment system [Cha8_85, Chau_89] can be used. In the latter case, the subscribers buy so-called stamps from the PTT from time to time, which are special messages signed by the PTT using blind signatures. These stamps can be included, e.g., in the SendEstab-messages [PfPW1_89], cf. Fig. 4.

5 Performance

The main bottleneck is the bandwidth of the signalling channels in both directions. (The second strictest requirement is the number of time-slices one may have to wait to establish a channel.)

Length of a time-slice: From the subscriber to the MIXes, there are 7 messages per time-slice: 2 SendEstab-, 2 RecEstab-, and 2 call-establishment messages (Sect. 4.2), and 1 broadcast-check message (Sect. 4.3). We assume the following field lengths (cf. Fig. 4, 5) in bits:

- Time-slice number: 30
- Area code: 22
- Stamp: 670
- Dummy message?: 1
- Seed for pseudo-random numbers: 128
- Time limit: 20

- Label: 28
- Priority token: 670

Let m be the number of MIXes per cascade, n_{sym} the key length of the symmetric cryptosystem, and b_{asym} the block length of the asymmetric cryptosystem, and also the length of a signature. For the total length of the 7 messages, we obtain (cf. [PfPW1_89])

$$Len7 = (m\text{-}1) \bullet 180 + (6m\text{-}8) \bullet n_{sym} + 5 \bullet b_{asym} + 2946.$$

(Note that the keys depicted in Fig. 4 are the same which are used for hybrid encryption.)

The total bandwidth of the signalling channel, in each direction, is 16 kbit/s. Since some of it must be used for error control, we assume that 12 kbit/s are available for these messages. Thus a lower bound on the length z of a time-slice (in seconds) is $z \geq Len7 / 12000$.

For $n_{sym} = 128$, $b_{asym} = 660$, and $m = 10$, we obtain

$$z \geq 1.22 \text{ s.}$$

Size of an untraceability set: The bandwidth from the MIXes to the subscriber is needed to distribute the incoming-call messages to the subscriber's untraceability set. To evaluate this, data about busy-hour call-establishment is needed. Let λ be an upper limit on the average arrival rate of incoming-call messages for one user. From some PTT-data, we derived $\lambda = 1/300$ (in s^{-1}) as a very conservative assumption. For the acceptable busy-hour system time, i.e. the average time an incoming-call messages needs from L_B to B, we used $T = 0.5$ s. Using an M/D/1-model, we obtain an upper limit on the size of untraceability sets of about 5000, if all incoming-call messages use invisible implicit addresses (cf. Ch. 2) [PfPW1_89]. The sets can be larger if some of the implicit addresses are visible, and thus shorter.

Call-establishment time: The time needed for the computations, both at the terminal equipment and at the MIXes, is almost negligible (cf. [PfPW1_89]), we take an upper limit as 0.01 s per MIX. Thus time until a call from A to B has been established mainly consists of the following parts, if B accepts this call at once:

- Waiting until A's MIX-cascade mixes call-establishment messages ($\leq z$).
- Call-establishment in the long-distance network (≤ 0.2 s).
- Waiting until the incoming-call message is broadcast to B (in peak hours T on average).
- Waiting until B can establish a MIX-channel with A ($\leq z$).

With the values from the previous paragraphs, we obtain a time of 3.34 s. On average, the call-establishment will take about half as long.

6 Summary

We have shown that telephony with MIXes is feasible under the technical assumptions made by the PTTs for narrow-band ISDN. Participants are untraceable within fixed untraceability sets of about 5000 subscribers, i.e. calls can only be traced to such a group. Local calls are completely unobservable.

Some problems remain: It seems difficult to organize the responsibility for the MIXes, so that every participant trusts at least one MIX. Of course, ISDN-MIXes are based on cryptographic assumptions. Finally, it seems impossible to take countermeasures against some specific active attacks on untraceability, since that would enable participants to prevent all members of their untraceability set from communicating (see Sect. 4.3). For practical purposes, however, the security achieved seems quite satisfactory.

As usual, note that untraceable communication does not complicate identification on higher layers, e.g. by digital signatures, where that is needed.

Also note that if optical fibers are introduced as subscriber lines, DC-nets can be used and offer many advantages [Cha8_85, Chau_88, WaPf_89, BoBo_89, Waid_89, Pfit_90].

Acknowledgement

We are pleased to thank *Manfred Böttger* and Prof. Dr. *Winfried Görke* for helpful discussions, and the *German Science Foundation* (*DFG*) for financial support.

References

BoBo_89 Jurjen Bos, Bert den Boer: Detection of Disrupters in the DC Protocol; Eurocrypt '89; Houthalen, 10.–13. April 1989, Abstracts; Proc. to appear in the series LNCS, Springer-Verlag Heidelberg.

Bock_88 Peter Bocker: ISDN – The Integrated Services Digital Network; Concepts, Methods, Systems; In collaboration with G. Arndt, V. Frantzen, O. Fundneider, L. Hagenhaus, H. J. Rothamel, L. Schweizer; Springer-Verlag, Heidelberg 1988.

Cha8_85 David Chaum: Security without Identification: Transaction Systems to make Big Brother Obsolete; Communications of the ACM 28/10 (1985) 1030-1044.

Chau_81 David Chaum: Untraceable Electronic Mail, Return Addresses, and Digital Pseudonyms; Communications of the ACM 24/2 (1981) 84-88.

Chau_88 David Chaum: The Dining Cryptographers Problem: Unconditional Sender and Recipient Untraceability; Journal of Cryptology 1/1 (1988) 65-75.

Chau_89 David Chaum: Privacy Protected Payments – Unconditional Payer and/or Payee Untraceability; SMART CARD 2000: The Future of IC Cards, Proceedings of the IFIP WG 11.6 International Conference; Laxenburg (Austria), 19.-20. 10. 1987, North-Holland, Amsterdam 1989, 69-93.

DaPr_89 D. W. Davies, W. L. Price: Security for Computer Networks, An Introduction to Data Security in Teleprocessing and Electronic Funds Transfer; (2nd ed.) John Wiley & Sons, New York 1989.

DES_77 Specification for the Data Encryption Standard; Federal Information Processing Standards Publication 46 (FIPS PUB 46), January 15, 1977.

GoGM_86 O. Goldreich, S. Goldwasser, S. Micali: How to construct random functions; Journal of the ACM 33/4 (1986) 792-807.

GoMR_88 Shafi Goldwasser, Silvio Micali, Ronald L. Rivest: A Digital Signature Scheme Secure Against Adaptive Chosen-Message Attacks; SIAM J. Comput. 17/2 (1988) 281-308.

Kah2_85 Peter Kahl (Hrsg.): ISDN, Das künftige Fernmeldenetz der Deutschen Bundespost; R. V. Decker's Taschenbuch Telekommunikation (TTK), R. V. Decker's Verlag G. Schenk, 1985.

Pfit_90 Andreas Pfitzmann: Diensteintegrierende Kommunikationsnetze mit teilnehmerüberprüfbarem Datenschutz; IFB 234, Springer-Verlag, Heidelberg 1990.

PfPf_89 Birgit Pfitzmann, Andreas Pfitzmann: How to Break the Direct RSA-Implementation of MIXes; Eurocrypt '89; Houthalen, 10.–13. April 1989, Abstracts; Proc. to appear in the series LNCS, Springer-Verlag Heidelberg.

PfPW_88 Andreas Pfitzmann, Birgit Pfitzmann, Michael Waidner: Datenschutz garantierende offene Kommunikationsnetze; Informatik-Spektrum 11/3 (1988) 118-142.

PfPW1_89 Andreas Pfitzmann, Birgit Pfitzmann, Michael Waidner: Telefon-MIXe: Schutz der Vermittlungsdaten für zwei 64-kbit/s-Duplexkanäle über den (2•64+16)-kbit/s-Teilnehmeranschluß; Datenschutz und Datensicherung DuD /12 (1989) 605-622.

PfWa_86 Andreas Pfitzmann, Michael Waidner: Networks without user observability -- design options; Eurocrypt '85, LNCS 219, Springer-Verlag, Berlin 1986, 245-253; Extended version in: Computers & Security 6/2 (1987) 158-166.

RSA_78 Ronald L. Rivest, Adi Shamir, Leonard Adleman: A Method for Obtaining Digital Signatures and Public-Key Cryptosystems; CACM 21/2 (1978) 120-126, reprinted in: CACM 26/1 (1983) 96-99.

Tane_88 Andrew S. Tanenbaum: Computer Networks; 2nd ed., Prentice-Hall, Englewood Cliffs 1988.

Waid_89 Michael Waidner: Unconditional Sender and Recipient Untraceability in spite of Active Attacks; Eurocrypt '89; Houthalen, 10.–13. April 1989, Abstracts; Proc. to appear in the series LNCS, Springer-Verlag Heidelberg.

WaPf_89 Michael Waidner, Birgit Pfitzmann: Unconditional Sender and Recipient Untraceability in spite of Active Attacks – Some Remarks; Fakultät für Informatik, Universität Karlsruhe, Interner Bericht 5/89, March 1989.