Das MIX-Netz

Inhaltsverzeichnis

1 Einführung in das MIX-Netz

Die bisher durchgeführten Versuche stellten immer Grundlagen und Verfahren zur Verfügung, Daten gegenüber Angreifern zu sichern. In diesem Versuch soll nun erstmals dargestellt werden, wie die sogenannten **Verkehrsdaten** geschützt werden können. Im Gegensatz zu den **Inhaltsdaten** beinhalten die Verkehrsdaten Informationen über die Kommunikationsverbindung; d.h. wer kommuniziert mit wem, zu welchem Zeitpunkt, wie lange und wie häufig. Daraus lassen sich auch die sogenannten **Interessensdaten** ableiten; Daten also, aus denen

das Interesse eines Benutzers an bestimmten Informationen gefolgert werden kann. Aus dem Bedürfnis heraus, auch die Verkehrsdaten schützen zu wollen (zu müssen), wurde erstmals in [Chau_81] ein System vorgeschlagen, welches dies leisten sollte. Die zentrale Einheit dieses Systems ist der sogenannte „Mix“ [Chau_81, Cha1_84, Pfi1_85].

Die Idee eines Mixes ist die einer schwarzen Kiste, in die Nachrichten hineinlaufen (von Teilnehmern eines Kommunikationsnetzes kommend) und die entsprechenden Nachrichten nach einer „Bearbeitung“ wieder herauslaufen (an Teilnehmer eines Kommunikationsnetzes gerichtet). Grund- und Hauptanforderung an den Mix ist die **Nichtzuordenbarkeit der einlaufenden zu den auslaufenden Nachrichten**. Um dies zu gewährleisten, muß ein Mix verschiedene Operationen ausführen, die in Abschnitt 1.2 hergeleitet werden sollen.

1.1 Grundsituation

Gegeben sei eine Anzahl von Teilnehmern, die miteinander über ein Kommunikationsnetz Nachrichten austauschen wollen. Jeder Teilnehmer erreiche jeden anderen mit Hilfe einer spezifischen Adresse, die der Nachricht beigefügt werden muß. Auf den Schutz der Inhaltsdaten soll zunächst kein besonderes Augenmerk gelegt werden. Dieser Schutz ergibt sich, wie sich später noch zeigen wird, zwangsläufig aus der Funktionsweise des Mixes bzw. kann durch nunmehr hinlänglich bekannte Maßnahmen erreicht werden. Das Hauptziel ist also nur das Verbergen oder Unkenntlichmachen der Verkehrsdaten, was nun durch die Zwischenschaltung eines Mixes geschehen soll. Jeder Teilnehmer kann Nachrichten an den Mix senden (dieser hat demzufolge auch eine Adresse) und jeder Teilnehmer kann Nachrichten von dem Mix empfangen. Der Sender der Nachricht muß sicherstellen, daß der Mix die Adresse des Empfängers erfährt.

(Mit „Sender“ und „Empfänger“ seien immer Teilnehmer des MIX-Netzes gemeint. Mit „Teilnehmer“ soll immer ein (menschlicher) Benutzer des Netzes assoziiert werden. Mit „Station“ sind alle Knoten (Teilnehmer und Mixe) des Netzes gemeint.)

Unter diesen Grundvoraussetzungen erfolgt die Herleitung der notwendigen Mixoperationen.

1.2 Bedingungen an den Mix

Es sei nochmals das Ziel eines Mixes betont: Zwei Teilnehmer eines Netzes möchten miteinander kommunizieren, diese Verbindung aber geheim halten. Sie bedienen sich dazu eines Mixes. Im folgenden sei nur der Weg vom Sender zum Empfänger betrachtet. Der Sender formuliert eine Nachricht und schickt sie an den Mix. Dieser empfängt die Nachricht, erfährt aus ihr die Adresse des Empfängers und leitet die Nachricht dementsprechend weiter. Der Umweg über den Mix macht bezüglich der Geheimhaltung der Kommunikationsbeziehung nur dann Sinn, wenn der Mix es schafft, die Zuordnung der von ihm empfangenen und der von ihm weitergesendeten Nachricht zu verhindern! Gesetzt den Fall, ein Angreifer kann alle von dem Mix

empfangenen und gesendeten Nachrichten kontrollieren (abhören), so muß es für ihn dennoch unmöglich sein herauszufinden, welche der empfangenen Nachrichten wohin weitergesendet wurden.

- Sporadisch eintreffende Nachrichten gleicher Länge von unterschiedlichen Absendern werden zunächst gesammelt („gepuffert“) und nach Ansammlung einer ausreichend großen Nachrichtenanzahl und anschließender „Bearbeitung“ wieder ausgegeben. Eine sofortige Ausgabe der eintreffenden Nachrichten zöge eine sofortige Aufdeckung der Kommunikationsbeziehung nach sich. Dies wäre ebenso der Fall, wenn die Ausgabereihenfolge der Eingabereihenfolge entspräche (FIFO-Prinzip): Die Nachrichten müssen nach der „Bearbeitung“ in einer veränderten Reihenfolge weitergegeben werden.

  Der Vorgang des Pufferns von Nachrichten kann auf zwei Arten geschehen:

  1. **Batch-Modus**: Eine genügend große Anzahl Nachrichten wird gesammelt. Wenn diese Anzahl erreicht ist, werden alle Nachrichten in einem Schub („batch“) ausgegeben. (Im weiteren Verlauf des Versuches wird von diesem Modus ausgegangen.) Die veränderte Ausgabereihenfolge könnte etwa durch alphanumerisches Sortieren verwirklicht werden.
  2. **Pool-Modus**: Es werden zunächst solange Nachrichten gesammelt, bis eine genügend große Anzahl erreicht wurde. Wenn eine weitere Nachricht eintrifft, wird diese auch in den Pool aufgenommen, und es wird eine Nachricht zufällig ausgewählt, die weitergesendet wird.

- Damit der Weg einer Nachricht über einen Mix nicht weiterverfolgt werden kann, d.h. damit eine empfangene Nachricht nicht der entsprechenden weitergereichten Nachricht zugeordnet werden kann, muß diese Nachricht ihr äußeres Erscheinungsbild ändern. Sie wird also von dem Mix ent- oder verschlüsselt, oder einfacher: umcodiert. Die Umcodierung soll — der allgemeinen Notation folgend — mit dem Schlüssel d eines Konzelationssystems geschehen.

- Gleiche Nachrichten dürfen weder gleichzeitig noch mehrmals bearbeitet werden. Zur Erklärung stelle man sich vor, in einem Schub werden mehrere gleiche Nachrichten bearbeitet. Diese werden nach dem Umcodieren das gleiche Erscheinungsbild haben. Somit stehen diese Nachrichten isoliert und leicht erkennbar da. Wird die gleiche Nachricht in mehreren Schüben verwendet, so wird sie auch immer das gleiche Erscheinungsbild nach dem Umcodieren haben — vorausgesetzt, der vom Mix verwendete Schlüssel wurde in der Zwischenzeit nicht verändert. Auf diese Weise ließe sich durch wiederholte Angriffe eine Zuordnung von empfangener und gesendeter Nachricht herstellen.

  Zur Veranschaulichung können Sie einen solchen Replay-Angriff (Wiederholung gleicher Nachrichten) im Mix-Demonstrator durchführen. (Eine Dokumentation zum Mix-Demonstrator befindet sich in Anhang A). Es genügen dafür zwei Teilnehmer und ein Mix. Um die Wirkung des Replay-Angriffs zu erkennen, muß die Wiederholungserkennung des Mixes abgeschaltet werden. (Mix-Demonstrator-Hauptfenster ⇒ Mix ⇒ Wiederholungs-Filter). Durch den Wiederholungsfilter wird dieser Angriff verhindert.

- Es darf keine Nachricht, die vor dem Empfangen aus anderen Nachrichten herausstach, auch nach dem Weiterreichen wieder herausstechen, oder allgemeiner: Keine Nachricht sollte überhaupt durch irgendwelche Merkmale von anderen Nachrichten (äußerlich) unterscheidbar sein. Dies bedeutet gleiche Länge für alle Nachrichten eines Schubes und es bedeutet gleiche zeitliche Behandlung aller Nachrichten.

Damit ergibt sich folgender Überblick: Ein Mix filtert gleiche Nachrichten und puffert die akzeptierten Nachrichten gleicher Länge[1] in einem Schub. Nachrichten eines Schubes werden gemeinsam umcodiert, umsortiert und weitergereicht. Die einzelnen Operationen sind nocheinmal im Bild aufgeführt.

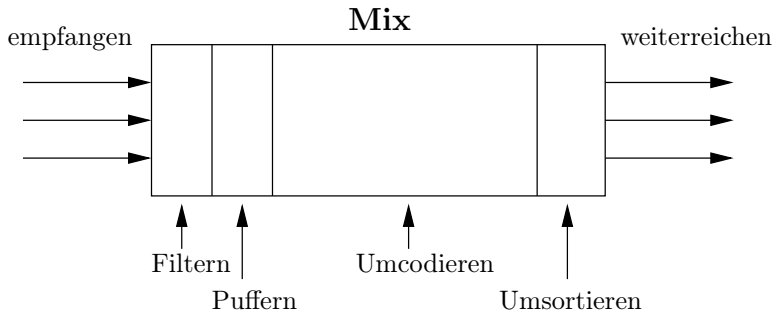


Abbildung 1: Zur Spezifikation eines Mixes

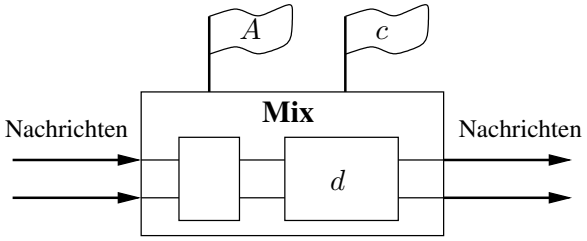
AUFGABE 1: (Batch- und Pool-Modus)
Testen Sie Batch- und Pool-Modus am Mix-Demonstrator (es genügen dazu zwei Teilnehmer und ein Mix). Welche Vor- und Nachteile haben diese beiden Modi? (Hinweis: Es soll die Zuordenbarkeit von Eingabe- zu Ausgabenachrichten und die Verzögerungszeit von Nachrichten betrachtet werden.) ◊

Eine für die Herleitung wichtige Annahme ist, daß der betrachtete Mix ein „mittlerer“ Mix ist![2] Dies bedeutet: Der betrachtete Mix empfängt Nachrichten weder direkt von einem Sender, noch schickt er sie direkt an einen Empfänger weiter. Der Mix befinde sich etwa in der Mitte einer ganzen Mixkette. Dies ist wegen der Sonderstellung des ersten und letzten Mixes einer Mixkette wichtig. Hier sei nur angemerkt, daß ein erster Mix immer den Sender und ein letzter Mix immer den Empfänger einer Nachricht kennen darf. (Aus verschiedenen Anwendungen ergibt sich dann, ob der erste bzw. letzte Mix den Sender bzw. Empfänger einer Nachricht kennen muß oder nicht.)

Eine mögliche Darstellung eines mittleren Mixes mit seinen wesentlichen Charakteristika ist in dem Bild 2 „Ein mittlerer Mix“ gegeben.

[1] Die Nachrichten müssen gleiche Länge haben, da ein Angreifer, der Ein- und Ausgabenachrichten des Mixes abhört, sonst anhand der Länge der Nachricht den Mix überbrücken kann.

[2] d.h. es soll im Folgenden soll davon ausgegangen werden, daß nicht ein einzelner Mix verwendet wird, sondern daß Nachrichten jeweils eine Kette mit mehreren Mixen durchlaufen müssen, bevor sie den Empfänger erreichen. Zum Sinn dieser Annahme siehe Abschnitt 1.3.



A: öffentliche Adresse
c: öffentlicher Schlüssel
d: geheimer Schlüssel
kleiner Kasten: Filtern und Puffern
großer Kasten: Umcodieren und Umsortieren

Abbildung 2: Ein mittlerer Mix

Da der Mix den Sender bzw. den Empfänger nicht kennen soll (Ausnahmen bilden eben der erste und der letzte Mix, die aber nicht betrachtet werden), können diese mit dem Mix auch keine (geheimen) Schlüssel austauschen. Dies ist aber für alle symmetrischen Konzelationssysteme notwendig. Damit ist zwangsläufig nur noch ein asymmetrisches Konzelationssystem mit einem öffentlichen Chiffrierschlüssel c und einem geheimen (nur dem Mix bekannten) Dechiffrierschlüssel d möglich. Der Mix kann sein Schlüsselpaar (c, d) selbst generieren. Die öffentlichen Schlüssel c der Mixe werden (z.B. über eine Schlüsselzentrale) allen Netzteilnehmern bekannt gegeben. Mit dem geheimen Schlüssel d werden alle empfangenen Nachrichten entschlüsselt, also umcodiert.

Zur Notation: Schickt ein Teilnehmer eine Nachricht N an einen Mix, so wendet dieser seinen geheimen Schlüssel auf die Nachricht an und gibt das Ergebnis $d(N)$ aus. Verschlüsselt der Teilnehmer die Nachricht N vorher mit dem öffentlichen Schlüssel des Mixes c (Notation: $c(N)$) und schickt sie an den Mix, der den dazugehörigen geheimen Schlüssel d kennt, so gibt dieser $d(c(N)) = N$ aus.

Die Sicherheit eines Mixes steht und fällt mit dem von ihm verwendeten Konzelationssystem. Angriffe gegen Mixe sind darum in der Regel Angriffe gegen das verwendete Konzelationssystem. Darüber hinaus muß der Mix gegen **adaptive aktive Angriffe** [Boett_89, Pfit_89, Bleu_91, Versuch "Asymmetrische Konzelationssysteme"] sicher sein. Ein Angreifer könnte nicht nur alle Leitungen abhören, sondern auch selbst Teilnehmer des MIX-Netzes sein. Er könnte also selbst Nachrichten formulieren und dem Mix zusenden (aktiver Angriff) bzw. aus den daraus gewonnenen Erkenntnissen neue Nachrichten formulieren und den Angriff wiederholen (adaptiver Angriff).

Ein besonders einprägsames Beispiel beschreibt die Situation, in der ein Angreifer von den n Nachrichten, die ein Mix in einem Schub bearbeitet, selbst $n-1$ geliefert hat. In diesem Fall ist der Mix bezüglich der von einem anderen Teilnehmer gelieferten Nachricht sinnlos geworden. Weiter: Arbeiten alle Sender und Empfänger von Nachrichten eines Zeitintervalls zusammen,

können sie einen weiteren Teilnehmer trotz der Wahl beliebig vieler Mixe beobachten [Pfit_89].

AUFGABE 2
Das verwendete asymmetrische Konzelationssystem sei deterministisch, d.h. Verschlüsselung von gleichen Nachrichten mit dem gleichen Schlüssel ergibt deterministisch gleiche Schlüsseltexte. Eine formalere (und andere) Darstellung dieses Sachverhalts ist: $d(X) = d(Y) \Rightarrow X = Y$. Ein Angreifer höre alle von einem Mix empfangenen und weitergereichten Nachrichten ab. Er kennt das vom Mix verwendete asymmetrische Konzelationssystem und den öffentlichen Schlüssel c. Welcher Angriff ermöglicht sich ihm unter diesen Bedingungen?

(Hinweis: Was ist die Aufgabe des Mixes? Wie kann der Angreifer den Mix „überbrücken", d.h. ihn sinnlos machen?) ◊

Werden nicht ausreichend Nachrichten gesendet, so kann es bei den Mixen zu langen Verzögerungszeiten kommen. Um diesem Problem zu entgehen, kann auf bedeutungslose Nachrichten (*dummy traffic*) zurückgegriffen werden, die von den Stationen (Sender und Mixe) gesendet werden. Auf diese Art und Weise kann auch der Sender vor Beobachtbarkeit durch Angreifer geschützt werden. Das Senden bedeutungsloser und bedeutungstragender verschlüsselter Nachrichten läßt sich nicht unterscheiden.

ZUSAMMENFASSUNG
Der oben entwickelte Mix M besitzt eine öffentliche Adresse A, über die ihm Nachrichten N zugeschickt werden können. Außerdem generiert er sich ein Schlüsselpaar (c, d) eines asymmetrischen Konzelationssystems. c ist sein öffentlicher, d ist sein geheimer Schlüssel. Es gilt $d(c(X)) = X$ für beliebige Nachrichten X. Die Mixoperationen bestehen aus:

1. **Filtern:** Nachrichten werden auf Wiederholung geprüft.
2. **Puffern:** Nachrichten gleicher Länge von unterschiedlichen Absendern werden zu einem Schub zusammengefaßt.
3. **Umcodieren:** Nachrichten werden umcodiert (mit d entschlüsselt).
4. **Umsortieren:** Nachrichten werden umsortiert (z.B. alphanumerisch).

Anschließend werden die „gemixten" Nachrichten in einem Schub an die entsprechenden Adressen weitergegeben.

AUFGABE 3 (Reihenfolge der Mixoperationen)
Was passiert, wenn die Mixoperationen Umcodieren und Umsortieren vertauscht würden? Ist der Mix dann überbrückbar? ◊

1.3 Verwendung vieler Mixe

Obwohl gemäß Spezifikation ein Mix zur Verdeckung der Kommunikationsbeziehung ausreicht, ist es dennoch sicherer, eine Nachricht über verschiedene und voneinander unabhängige Mixe zu senden. (Das Wort „unabhängig" bezieht sich auf die Entwerfer, die Hersteller, die Betreiber und Warter des Mixes [Chau_81, Cha1_84, Pfit_86].) In diesem Fall sinkt zumindest die

Wahrscheinlichkeit, daß alle gewählten Mixe von einem bzw. von mehreren kooperierenden Angreifern kontrolliert werden.

Für die Festlegung der Reihenfolge der Mixe gibt es folgende Möglichkeiten:

1. Jeder Teilnehmer kann sich die Reihenfolge und Anzahl der Mixe selbst wählen, durch die er seine Nachrichten schicken will. Dadurch kann er sich Mixe auswählen, denen er vertraut. Er muß nicht eine feste Anzahl von Mixen auswählen.

2. Es werden sogenannte **Mix-Kaskaden** festgelegt. Alle Teilnehmer, die eine Mix-Kaskade benutzen, senden ihre Nachrichten in gleicher Reihenfolge durch alle Mixe der Kaskade.

Mix-Kaskaden haben den Vorteil, daß bei maximal vielen angreifenden Mixen, (Bei einer Kaskade der Länge n sind das $n-1$ Mixe) die größtmögliche Anonymitätsgruppe entsteht. Hingegen gibt es bei der freien Auswahl von Mixreihenfolgen dann einen Vorteil, wenn nur wenige Mixe angreifen. In diesem Fall entstehen größere Anonymitätsgruppen.

Zur Notation: Damit bei frei wählbarer Mixreihenfolge eine Nachricht vom Sender ihren Weg über die Mixe zum Empfänger findet, sind Adressen notwendig. Jede Station besitze eine Adresse A. Zur Unterscheidung der Adressen werden Indizes benutzt. Mix_1 habe etwa die Adresse A_1, der Empfänger die Adresse A_E, u.s.w. Die Adressen werden der Nachricht beigefügt, damit jeder Mix weiß, an welche Station er die bearbeitete Nachricht weiterschicken muß. Eine Nachricht N, die an die Adresse A geschickt werden soll, schreibt sich als (A, N). Wird diese Gesamtnachricht verschlüsselt, so erhält man $c(A, N)$. Der Mix mit dem zugehörigen Schlüssel d kann nun $d(c(A, N)) = (A, N)$ berechnen und gibt die Nachricht N an die Adresse A weiter.

AUFGABE 4

Angenommen, ein Angreifer kontrolliert maximal viele Mixe. (D.h. es gibt nur noch einen vertrauenswürdigen Mix in der Mix-Kette.) Die Mixreihenfolgen sind frei wählbar, aber die Teilnehmer wählen immer eine *feste Anzahl* von Mixen. Diese Anzahl ist dem Angreifer bekannt. In diesem Fall kann der Angreifer auch den verbliebenen vertrauenswürdigen Mix überbrücken.

a) Überlegen Sie, wie ein solcher Angriff aussehen könnte.

   Hinweis: Neben diversen aktiven Angriffen (durch Einschleusen geeigneter Nachrichten) ist auch ein passiver Angriff (d.h. der Angreifer beobachtet lediglich an allen ihm zugänglichen Stellen im Netz) möglich. Überlegen Sie dazu zunächst, was der Angreifer über die Anzahl der von einer Nachricht durchlaufenen Mixe weiß, die er selbst kontrolliert.

b) Schlagen Sie Möglichkeiten vor, diesen Angriff zu verhindern. ◊

2 Erläuterung des direkten Umcodierungsschemas für Senderanonymität

Aus dem oben Gesagten und der Aufgabe 2 läßt sich nun ein Schema ableiten, welches die Anonymität der Kommunikationsbeziehung und die Anonymität des Senders (aus der Sicht

Empfängers) gewährleistet.

Aus dem Determinismus des verwendeten Konzelationssystems ergibt sich die Notwendigkeit einer zusätzlichen Veränderung der Nachrichten (ohne sie unleserlich zu machen!). Es werden der Nachricht Zufallsbits angefügt, die einen Test nach Aufgabe 2 verhindern. Diese Zufallsbits dürfen vom Mix nicht ausgegeben werden. Betrachtet man sie als Geheimnis zwischen Sender und Mix, so kommt die Wahrung dieses Geheimnisses der Wahrung der Kommunikationsbeziehung zwischen Sender und Empfänger gleich.

Wie die Zufallsbits im einzelnen in die Nachricht mit eingehen und welchen Einfluß sie beim Umcodieren der Nachricht haben, hängt von dem verwendeten Konzelationssystem und den Betriebsarten ab [Bras_88, AssPf2_90]. Der Übersichtlichkeit halber soll diese Prozedur nicht ausführlich in die Notation mit übernommen, sondern nur durch das Anfügen von Zufallsbits Z angedeutet werden. Eine mit Zufallsbits Z versehene Nachricht N, die an die Adresse A geschickt werden soll, stellt sich dann formal als (Z, A, N) dar.

Die Funktionen der Netzteilnehmer und Mixe bei diesem Umcodierungsschema werden im folgenden vorgestellt. Es wird eine Nachricht von einem Sender über verschiedene Mixe zu dem gewünschten Empfänger betrachtet. Die notwendigen Ver- und Entschlüsselungen und die Adreßbehandlung werden für jede einzelne Station betrachtet. Die Bewertung der erreichten Anonymität erfolgt zum Schluß.

Der Sender:
Bei dem **direkten Umcodierungsschema für Senderanonymität** wählt der Sender S eine Reihenfolge von Mixen, über die er eine Nachricht N an den Empfänger E schicken möchte. Er **verschlüsselt** N **vorab** mit dem öffentlichen Schlüssel c_E des Empfängers und den öffentlichen Schlüsseln c_i der Mixe. Mit Hilfe der Adressen A_i bzw. A_E, die der Nachricht beigefügt werden, teilt der Sender jeder mittleren Station jeweils die Station mit, an die die Nachricht weitergesendet werden soll. Die Adressen werden derart in die Gesamtnachricht verpackt, daß ein Mix immer nur die Adresse des nächsten Mixes (oder die des Empfängers) erfährt.

Bei Verwendung eines deterministischen asymmetrischen Konzelationssystems müssen bei den Verschlüsselungen Zufallsbits Z_i angefügt werden. Mit den Z_i beeinflußt der Sender die Umcodierung der einzelnen Mixe, weil sich die Zufallsbits auf die Erscheinungsbilder der Nachricht auswirken.

Die Z_i werden von dem Sender der Nachricht alleine gewählt (denn es ist nicht nötig und macht auch keinen Sinn, bzgl. der Wahl von Geheimnissen andere zu fragen!). Der Mix erfährt — bedingt durch das Protokoll — dieses Geheimnis, benötigt es für seine Aufgabe jedoch nicht.

Die Mixe:
Die einzelnen Mixe **entschlüsseln** die bei ihnen eintreffenden Gesamtnachrichten mit ihrem geheimen Schlüssel d_i. Sie erfahren so die Adresse der nächsten Station und — bei Verwendung eines deterministischen asymmetrischen Konzelationssystems — die Zufallsbits. Letztere werden ignoriert und die Restnachricht wird an die gefundene Adresse weitergesendet.

Der Empfänger:
Der Empfänger **entschlüsselt** die empfangene Nachricht mit seinem geheimen Schlüssel d_E und erfährt die von S gesendete Nachricht N. Es soll davon ausgegangen werden, daß ein

Empfänger die Nachricht N als ein Geheimnis betrachtet und sie nicht ausgibt. Sollte dies trotzdem geschehen, dann nur in modifizierter Form (z.B. formuliert er die Nachricht ausreichend um oder verwertet die Informationen nur auszugsweise). Ein Angriff nach Aufgabe 2 ist dann nicht mehr möglich. Das Mitschicken von Zufallsbits auf dem Weg vom letzten Mix zum Empfänger ist darum nicht notwendig.

Die Anonymität:
Bei diesem Schema kennt jede Station (bis auf den Sender) immer die Vorgängerstation und jede Station (bis auf den Empfänger) die Nachfolgestation. (Ein MIX-Netz mit nur einem Mix ist also schon dann überbrückt, wenn dieser eine Mix in der Hand eines Angreifers ist.) Der Sender kennt alle verwendeten Mixe, ihre Reihenfolge und Informationen über den Empfänger (eine Adresse, öffentlichen Schlüssel).

Der Sender einer Nachricht ist bei diesem Protokoll anonym gegenüber dem Empfänger, solange mit diesem nicht alle beteiligten Mixe zusammenarbeiten (d.h. Austausch von Verkehrsdaten).

Ein Angreifer kann durchaus alle vom Sender ausgewählten Mixe und deren Reihenfolge kennen! Dennoch wird er nichts über die Kommunikationsbeziehung zwischen Sender und Empfänger erfahren, weil der Sender der Nachricht die Umcodierung der Mixe implizit durch die Wahl der geheimen (!) Zufallsbits vorgegeben hat.

AUFGABE 5

Betrachten Sie das folgende Bild 3. Jürgen möchte Sven Eric eine Nachricht N über die Mixe MIX 1 und MIX 2 (in dieser Reihenfolge!) schicken. A_i bezeichne die Adressen, c_i die öffentlichen und d_i (nicht eingezeichnet) die geheimen Schlüssel des verwendeten deterministischen asymmetrischen Konzelationssystems ($i \in S, 1, 2, E$). Wie sieht die Gesamtnotation dieser Nachricht an den Punkten (a), (b) und (c) aus? (Hinweis: Denken Sie an Schlüssel, Adressen und Zufallsbits. Beachten Sie die vorgegebene Reihenfolge der Mixe.)

Was fällt bezüglich der Gesamtnachricht auf? ◊

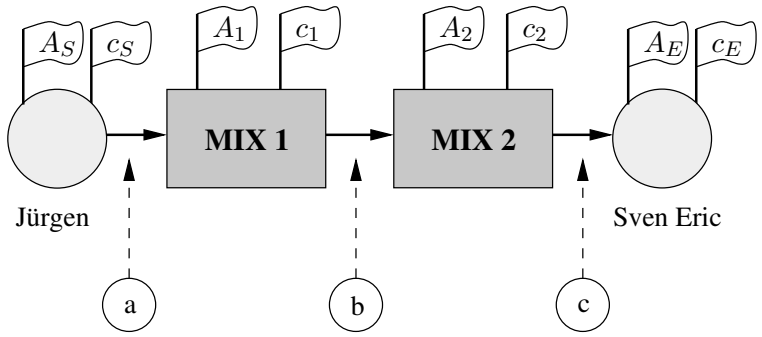


Abbildung 3: Zu Aufgabe 5

AUFGABE 6

Gib ein rekursives Schema für die Nachrichtenbildung des Senders bei n Mixen an. Die Nutznachricht N soll über die Mixe i, $i = 1, \ldots, n$ (in dieser Reihenfolge!) laufen. Der Empfänger

sei o.B.d.A. der letzte (n-te) Mix. ◊

3 Brechen der direkten RSA-Implementierung

Aus dem ersten Teil dieses Versuchs ging hervor, daß zur Geheimhaltung der Kommunikationsbeziehung ein vom Sender gewähltes Geheimnis nötig ist. Wie sich im Laufe dieses Versuches zeigen wird, läßt sich dieser in Abschnitt 1 konstruierte Mix bei schlechter Wahl des asymmetrischen Konzelationssystems immer noch überbrücken. Dies bedeutet: Trotz Wahrung des Geheimnisses lassen sich vom Mix empfangene Nachrichten den weitergereichten zuordnen. Diese Sicherheitsmängel stehen im Zusammenhang mit dem gewählten Konzelationssystem. Es soll weiterhin ein „mittlerer“ Mix betrachtet werden.

3.1 Grundlagen und Definitionen

In diesem Versuch spielen verschiedene Modelle über die Sicherheit von und die Angriffsmöglichkeiten bei Kryptoverfahren (und damit auch bei Mixen) eine Rolle. Sie sollen darum nochmals kurz erläutert werden (Siehe auch [DiHe_76, Boett_90, Bleu_91, Versuch “Asymmetrische Konzelationssysteme”]). Wichtig ist zunächst einmal die Trennung in informationstheoretische und komplexitätstheoretische Sicherheit. Dazu jedoch erst die

Definition 1a:

> Ein Konzelationssystem heißt **nachrichtenbezogen gebrochen**, wenn ein Angriff bekannt ist, der zu dem eigentlich zu schützenden Klartext führt.[3]

und die

Definition 1b:

> Ein Konzelationssystem heißt **vollständig gebrochen**, wenn ein Angriff bekannt ist, der zu dem (geheimen) Schlüssel des Konzelationssystems führt.

Ein erfolgreicher Angriff auf die verwendeten Schlüssel des Konzelationssystems ist natürlich immer ein erfolgreicher Angriff auf den zu schützenden Klartext. Mit den verwendeten Schlüsseln in der Hand läßt sich jeder damit erzeugte Schlüsseltext in den Klartext umwandeln.
(Bemerkungen über das Brechen von RSA stehen in [Pfit_89, Seite 25].)

Definition 2:

> Ein Konzelationssystem heißt **informationstheoretisch** sicher, wenn es trotz aller zur Verfügung stehender Informationen (ausschließlich der geheimen) auch mit unbeschränkt zur Verfügung stehender Zeit und (Rechen-)Mitteln nicht gebrochen werden kann.

[3] In manchen Anwendungen muß man ein Konzelationssystem schon dann als nachrichtenbezogen gebrochen ansehen, wenn ein Angriff bekannt ist, der **partielle Information** über den zu schützenden Klartext liefert. Beispiele hierfür könnten sein: der Klartext dieser verschlüsselten Nachricht ist gleich (ist ungleich) dem Klartext jener verschlüsselten Nachricht. Oder: Von den beispielsweise 1000 in der Beobachtungssituation des Angreifers zu erwartenden Klartexten können es 950 nicht sein.

Im Gegensatz dazu die

Definition 3:

> Ein Konzelationssystem heißt komplexitätstheoretisch sicher, wenn es trotz erheblicher (Rechen-)Mittel nicht in vertretbarer Zeit mit relevanter Wahrscheinlichkeit gebrochen werden kann.

Wünschenswert ist natürlich immer eine bewiesene informationstheoretische Sicherheit [Sha1_49]. Sie versagt bei Bekanntwerden der geheimen Information (etwa: Schlüssel); dies ist aber bei jedem Konzelationssystem der Fall, weshalb dies tunlichst immer (sozusagen trivialerweise) vermieden werden sollte.

Bei der komplexitätstheoretischen Sicherheit bleibt immer die Ungewißheit über die tatsächlichen Möglichkeiten des Angreifers. Eventuell hat er doch ausreichend Rechenleistung, um in vertretbarer Zeit alle Möglichkeiten (etwa alle möglichen Schlüssel) durchzutesten. Denkbar wäre auch, daß der Angreifer über geschicktere Algorithmen verfügt, Klartexte oder gar Schlüssel zu berechnen. Ist das in Abschnitt 1 verwendete Konzelationssystem nur komplexitätstheoretisch sicher, dann besitzt auch der Mix nur komplexitätstheoretische Sicherheit [Pfit_89].

Anmerkung: Bezüglich der Bewertung von komplexitätstheoretischer Sicherheit sollte nach obigen Ausführungen in zweierlei Hinsicht vorsichtig verfahren werden. Zum einen: Selbst wenn wir unterstellen, daß der nach Sicherheit Lechzende alle Möglichkeiten (insbesondere: Algorithmen) zum Brechen kennt, so müssen von jedem, der behauptet, ein Konzelationssystem sei komplexitätstheoretisch sicher, drei Fragen beantwortet werden, die genau den vagen Begriffen in Definition 3 entsprechen:

1. Welche (*erheblichen*) (Rechen-)Mittel werden dem Angreifer unterstellt?
2. Welche (*vertretbare*) Zeit wird dem Angreifer zugebilligt?
3. Welche Wahrscheinlichkeit für das Brechen wird noch als *irrelevant* betrachtet?

Zum anderen: Kennt der Angreifer, unabhängig von der Beantwortung der drei Fragen, wirklich nur die Algorithmen, die man ihm unterstellt — oder gar noch ausgefeiltere?

Kurzum: Man kann den Angreifer auf zwei Arten unterschätzen: Man hält ihn für zu arm oder für zu dumm. Gegen ersteres kann man sich durch pessimistische Abschätzungen gut absichern; gegen zweiteres nur sehr bedingt: Man macht ausschließlich seit langem übliche, unwiderlegte und in diesem Sinne validierte Annahmen, z.B. „Faktorisierung ist schwierig“. Bezüglich diesen ist ein plötzlicher algorithmischer Durchbruch nicht zu erwarten und ein gradueller Fortschritt kann extrapoliert werden.

In diesem Versuch soll der Mix aus Abschnitt 1 gebrochen werden, wenn als Konzelationssystem RSA [RSA_78] direkt verwendet wird. „Direkt“ bedeutet die Verwendung von RSA ohne zusätzliches Redundanzprädikat. Die verwendete Betriebsart sei ECB [Bras_88, AssPf2_90].

Das Brechen des Mixes ist nicht notwendigerweise gleichbedeutend mit dem Brechen des verwendeten Konzelationssystems. Die Definition eines gebrochenen Mixes ist schwächer als

die des gebrochenen Konzelationssystems. Ist aber im umgekehrten Fall das verwendete Konzelationssystem gebrochen, so natürlich auch der Mix.

Definition 4:

> Ein **Mix** heißt **gebrochen**, wenn ein Angriff die Zuordnung der vom Mix empfangenen zu den zugehörigen vom Mix weitergereichten Nachrichten ermöglicht hat.

Ein Angriff auf einen Mix bezeichnet also den Versuch, die Zuordnung (und damit die Kommunikationsbeziehungen) von Nachrichten aufzudecken. Der Angriff richtet sich demnach gegen die Spezifikation des Mixes, also ist das vom Mix verwendete Konzelationssystem mit involviert. Ein Angriff auf ein Konzelationssystem ist aber zunächst immer unabhängig von der Anwendung zu sehen. Die Konsequenzen für diese Anwendung müssen für den Fall eines erfolgreichen Angriffes gesondert betrachtet werden.

Hier wird ein Angriff auf einen Mix gestartet, der die Eigenschaften des Mixes (Ausgabe von Nachrichten) und des Konzelationssystems RSA (Multiplikativität) ausnutzt. Zunächst soll die Schwäche von RSA vorgestellt werden. Dann wird auf die spezielle Mix-Situation eingegangen. Schließlich soll die Versuchsumgebung beschrieben werden.

3.2 Angriff auf RSA

Im folgenden sei nun zunächst der klassische Angriff von Davida [Davi_82] (auch [Merr_83, Denn_84, Bras_88, Versuch "Digitale Signatursysteme"]) auf RSA beschrieben. Verwendet wird die Darstellung von Judy Moore in [Denn_84]. Dieser Angriff stellt einen Angriff auf den Schlüsseltext dar. Wie sich später zeigen wird, läßt sich dies für einen Angriff auf einen Mix mit RSA als verwendetes Konzelationssystem ausnutzen.

Ein Angreifer A (siehe Bild 4) hört auf einer Leitung (etwa die zu einem Empfänger) die Nachricht $c(X)$ ab. Dabei ist X die Klartextnachricht, die mit dem öffentlichen Schlüssel c von RSA verschlüsselt ist. A möchte nun wissen, was in der Nachricht X steht.

Dafür wählt er einen beliebigen Faktor f und verschlüsselt diesen mit dem bekannten Schlüssel c. Dann multipliziert er den verschlüsselten Faktor mit der abgehörten Nachricht. (Diese Multiplikation wird natürlich modulo m ausgeführt, wobei m der öffentliche Modulus von RSA ist.) Das Produkt $P = c(X) \cdot c(f) = (X^c \bmod m) \cdot (f^c \bmod m) = (X \cdot f)^c \bmod m = c(X \cdot f)$ schickt er nun an den Empfänger und hofft, von ihm die entschlüsselte Nachricht $X \cdot f$ zu bekommen. Hier wird die Multiplikativität von RSA ausgenutzt, denn es gilt (siehe oben): $c(X) \cdot c(f) = c(X \cdot f)$ und somit $d(P) = d(c(X \cdot f)) = X \cdot f$. Es ist anzumerken, daß $X \cdot f$ für den Empfänger in der Regel bedeutungslos ist. Es ist also nicht selbstverständlich, daß er diese Nachricht gedankenlos veröffentlichen wird. Darum sei angenommen, daß der Angreifer A sich diese Nachricht $X \cdot f$ auf irgendeine (dubiose) Weise beschaffen kann. $X \cdot f$ wird auch für den Angreifer zunächst bedeutungslos sein. Er kennt aber seinen Faktor f und kann — bei richtiger Wahl von f — die zugehörige Inverse $f^{-1} \pmod m$ dazu berechnen (Siehe auch Versuch "Zahlentheoretische Algorithmen"). Multipliziert er nun $X \cdot f \cdot f^{-1}$, so erhält er den Klartext X. RSA ist damit **nachrichtenbezogen** gebrochen [Boett_89, Pfit_89].

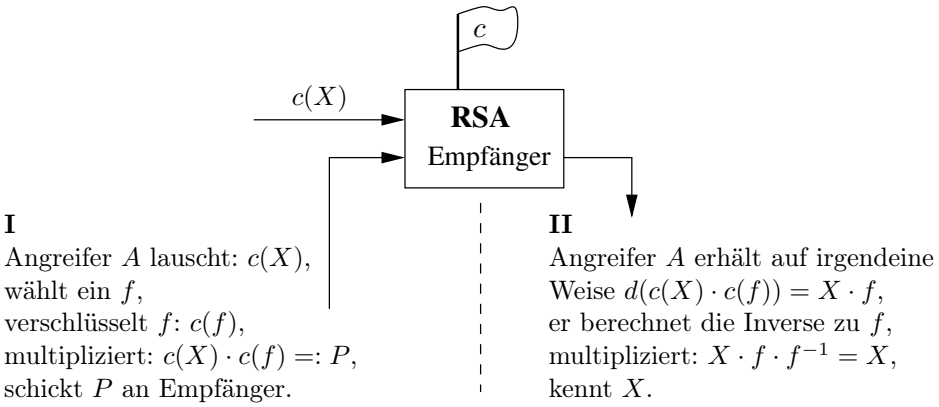


Abbildung 4: Angriff auf RSA nach Judy Moore

AUFGABE 7 (Umsetzung des Angriffes auf Mixe)
Versuchen Sie den obigen Angriff auf die Mix-Situation anzupassen. Beantworten Sie dazu im wesentlichen die folgenden Fragen:

a) Welche Tatsache, die im obigen Angriff problematisch ist, ist bei Mixen besonders einfach?

b) Stellen Sie sich in Bild 4 den Empfänger als Mix vor. Warum wäre mit diesem einfachen Schema ein erfolgreicher Angriff auf den Klartext X gleichzeitig ein erfolgreicher Angriff auf den Mix? (Hinweis: Siehe obige Definition 4 und Abschnitt 1)

c) Warum funktioniert der obige Angriff bei Mixen nicht ohne weiteres? Welche Information fehlt? (Hinweis: Denken Sie an Aufgabe 2.) ♢

3.3 Angriff auf Mixe

Allein das Fehlen einer **vollständigen** Nachrichtenausgabe bei Mixen verhindert das direkte Umsetzen des Angriffes nach Judy Moore auf Mixe. Dies ist jedoch kein Hinderungsgrund, den Angriff dennoch erfolgreich bei den gegebenen Rahmenbedingungen durchzuführen. Ausgangspunkt soll der in Abschnitt 1 konstruierte Mix sein. Dort enthält jeder RSA-Block X einen „echten“ Nachrichtenanteil N der Länge B (in Bits) und eine Zufallsbitkette Z der Länge b. (In diesem Unterkapitel bezeichnet N also einen Teil eines Nachrichtenblocks und nicht, wie in Aufgabe 5, die ganze Nachricht. Entsprechend werden hier Adressen auch nicht gesondert betrachtet — sie können Teil eines „Nachrichten“blocks sein.) Jeder RSA-Block sei L Bits ($L = B + b$) lang. Zur Vereinfachung werde nur ein RSA-Block betrachtet, auch wenn die Nachricht aus mehreren solcher Blöcke besteht. Dann sieht eine betrachtete Gesamtnachricht folgendermaßen aus:

Ein Sender verschlüsselt diese Nachricht mit dem öffentlichen Schlüssel c des belauschten Mixes. Der Mix erhält so $c(X)$. Kryptographisch bedeutet dies die Potenzierung von X mit

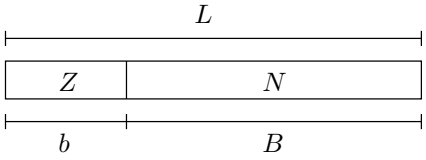


Abbildung 5: Nachrichtenformat für den Angriff auf Mixe

dem öffentlichen Exponenten c des Mixes modulo m. m ist der ebenfalls öffentliche Modulus des von dem Mix verwendeten Schlüsselpaares des Konzelationssytems RSA. Als Formel geschrieben erhält man:

$$c(X) = X^c \bmod m = (Z \cdot 2^B + N)^c \bmod m =: M$$

M wird an den Mix geschickt und M ist auch die Nachricht, die ein Angreifer A erhalten wird, wenn er die Nachrichten an den betreffenden Mix abhört. Der Mix seinerseits gibt nun $d(M)$ ohne die Zufallsbits Z (also nur N) an den (gedachten) Empfänger weiter. Es sollte klar sein, daß N keinen Klartext bezüglich des Senders darstellt, da der betreffende Mix sich in einer Mix-Kette befindet. N ist demnach ebenfalls eine nach dem gleichen Schema (mit einem anderen Schlüssel) verschlüsselte Nachricht. Dieser Sachverhalt ist aber für den Angriff nicht von Bedeutung. Aus der Sicht des Mixes jedoch stellt N Klartext dar. Dies wird von dem Angreifer ausgenutzt.

Der Angreifer A verfährt nach dem in Bild 4 dargestellten Schema. Er berechnet aus der von ihm abgehörten Nachricht M eine Nachricht M^* wie folgt: Er wählt einen Faktor f, verschlüsselt ihn mit dem öffentlichen Exponenten c des Mixes und multipliziert ihn dann mit M. Damit entsteht die Nachricht M^*, welche an den Mix geschickt wird.

AUFGABE 8

a) Beschreiben Sie M^* formal.

b) M^* wird an den Mix geschickt. Dieser entschlüsselt mit dem geheimen Schlüssel d. Wie sieht $d(M^*)$ aus der Sicht des Mixes und aus der Sicht des Angreifers formal aus? Wie sieht die Ausgabe dieser Nachricht aus der Sicht des Mixes und wie aus der Sicht des Angreifers aus? ◊

Damit ist der aktive Anteil des Angriffes beendet. Der Angreifer wartet jetzt im Schritt II (siehe Bild 4) auf die Ausgaben des Mixes. Zur besseren Übersicht gelte folgende (aber nicht unbedingt notwendige) Annahme: Der Mix empfängt M zusammen mit anderen (nicht weiter relevanten) Nachrichten in *Schub 1*. In der Ausgabe des Mixes von *Schub 1* befindet sich dann N. Der Angreifer sendet seine berechnete Nachricht M^* in einem zweiten Schub (*Schub 2*) an den Mix. In der Mixausgabe dieses Schubes befindet sich dann neben anderen, nicht weiter relevanten Nachrichten auch N^* (Aufgabe 8b)).

Der Angreifer weiß damit, daß

$$Z^* \cdot 2^B + N^* \equiv f \cdot (Z \cdot 2^B + N) \pmod m$$

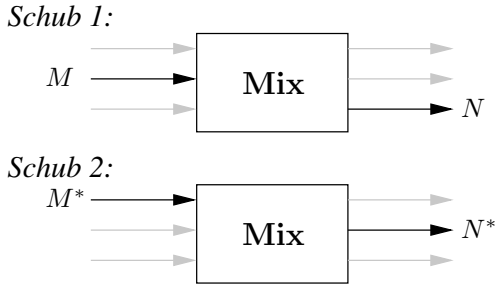


Abbildung 6: Angriffssituation bei Mixen

gilt. (Es handelt sich ja um die gleiche Nachricht, nur aus verschiedenen Perspektiven formuliert.) Da es eine Inverse zu 2^B gibt, kann zu

$$(N^* - f \cdot N) \cdot 2^{-B} \equiv f \cdot Z - Z^* \pmod{m}$$

umformuliert werden. Da aber Z bzw. Z^* Zahlen in $\{0, \ldots, 2^b - 1\} \pmod{m}$ darstellen, gilt:

$$f \cdot Z - Z^* \in \{-2^b + 1, \ldots, 0, \ldots, f \cdot (2^b - 1)\} \pmod{m}$$

Ein Angreifer kann somit alle Nachrichten des ersten Schubes mit allen Nachrichten aus *Schub 2* gemäß $(N^* - f \cdot N) \cdot 2^{-B} \bmod m$ „vergleichen“, d.h. diesen Ausdruck für alle N aus *Schub 1* und alle N^* aus *Schub 2* berechnen. Er prüft, ob das Resultat dieser Rechnung in der Menge $\{-2^b + 1, \ldots, 0, \ldots, f \cdot (2^b - 1)\} \bmod m$ liegt. (Der Modulus m auf eine Menge angewendet bedeute, daß jedes Element dieser Menge bezüglich m reduziert wird.) Es kann gegebenenfalls zu Mehrdeutigkeiten kommen (insbesondere, wenn der Angreifer den Faktor f dummerweise groß statt klein wählt), d.h. die Bedingung trifft für mehrere Nachrichtenpaare aus den beiden Schüben zu. In diesem Fall muß der Angriff mit einem anderen Faktor wiederholt werden. Um Mehrdeutigkeiten zu vermeiden, können auch mehrere RSA-Blöcke einer Nachricht mit dem Faktor multipliziert werden. Der auszuführende Test muß dann für alle diese Blöcke erfüllt sein, was die Wahrscheinlichkeit von Mehrdeutigkeiten reduziert. Wird genau ein Nachrichtenpaar gefunden, welches die Bedingung $f \cdot Z - Z^* \in \{-2^b + 1, \ldots, 0, \ldots, f \cdot (2^b - 1)\} \bmod m$ erfüllt, so ist der Mix überbrückt und der Angriff ist gelungen.

Sind alle Mixe in einem Kommunikationsnetz auf diese Art und Weise zu überbrücken, dann läßt sich der Weg einer Nachricht (vom Angreifer beobachtet und ausgewählt) durch das Kommunikationsnetz (soll heißen: über die verwendeten Mixe) nachvollziehen. Kann der Angreifer seiner beobachteten Nachricht obendrein einen Sender zuordnen und erkennt er das Empfangen von Nachrichten durch Netzteilnehmer, dann läßt sich die Anonymität einer Kommunikationsbeziehung aufdecken: Das verwendete MIX-Netz erfüllt dann seinen Zweck nicht mehr. Letztendlich schadet der Angriff also nicht dem Mix, sondern den Netzteilnehmern.

ZUSAMMENFASSUNG: (Übersicht zur Durchführung des Angriffes)

1) Wähle einen Mix aus dem MIX-Netz aus. Er habe den öffentlichen Schlüssel c und den geheimen Schlüssel d.

2) Höre einen Schub einlaufender Nachrichten dieses Mixes ab. Nenne ihn *Schub 1*. Greife aus *Schub 1* die Nachricht heraus, deren Weg weiterverfolgt werden soll. Nenne sie M.

3) Wähle einen Faktor f und berechne $c(f)$.

4) Berechne $c(f) \cdot M =: M^*$ und schicke M^* in *Schub 2* an den Mix.

5) Höre alle zu *Schub 1* gehörigen auslaufenden Nachrichten des Mixes ab. Berechne für jede Nachricht Y aus dieser Ausgabe den Ausdruck $f \cdot Y \cdot 2^{-B}$. Setze $V_1 := \{f \cdot Y \cdot 2^{-B} | \, Y \text{ aus der Ausgabe zu } \textit{Schub 1}\}$. (Damit enthält die Menge V_1 auch $f \cdot N \cdot 2^{-B}$, wobei N die zu M gehörige Ausgabe des Mixes ist.)

6) Höre alle zu *Schub 2* (oder gegebenenfalls weiteren Schüben) gehörigen auslaufenden Nachrichten des Mixes ab. Berechne für jede Nachricht Y^* aus dieser Ausgabe den Ausdruck $Y^* \cdot 2^{-B}$. Setze $V_2 := \{Y^* \cdot 2^{-B} | \, Y^* \text{ aus der Ausgabe zu } \textit{Schub 2}\}$. (Damit enthält die Menge V_2 auch $N^* \cdot 2^{-B}$, wobei N^* die zu $c(f) \cdot M = M^*$ gehörige Ausgabe des Mixes ist.)

7) Prüfe, ob für ein $v_1 \in V_1$ und ein $v_2 \in V_2$ der Ausdruck $v_2 - v_1 \in \{-2^b + 1, \ldots, 0, \ldots, f \cdot (2^b - 1)\} \bmod m$ gilt. (Ausführlich beschrieben bedeutet dies: Betragskleinste Darstellung von $((v_2 - v_1) \bmod m) \in \{(-2^b + 1) \bmod m, \ldots, (f \cdot (2^b - 1)) \bmod m\}$

   Wenn ja, dann gib die zu v_1 gehörende Nachricht aus. Beende den Angriff. Wenn nein, dann versuche es in dem nächsten Eingabe- und Ausgabeschub des Mixes nocheinmal (weiter mit Schritt 3).

AUFGABE 9

Implementieren Sie diesen Angriff im Projekt `MixRSAAngriff`, Klasse `Attack`. Es ist dort bereits ein Objekt `mix` vorbereitet, das den anzugreifenden Mix darstellt. Dieser Mix stellt Funktionen bereit, die es ermöglichen, für den Angriff benötigte Daten auszulesen und Nachrichten zu einem Schub hinzuzufügen. Im Gegensatz zur hier angegebenen Beschreibung werden die Schübe im Programm von 0 beginnend durchnummeriert. Die Bearbeitung eines Schubes durch den Mix wird durch Aufruf der Methode `executeBatch()` durchgeführt, und muß im Programm durch den Angreifer selbst aufgerufen werden. Weitere Hinweise zur Implementierung werden im bereits vorgefertigten Teil der Klasse `Attack` gegeben. Außerdem existiert eine JavaDoc-Dokumentation zu allen benötigten Klassen und Methoden. ♢

AUFGABE 10

Nehmen Sie an, der Angreifer könnte gezielt einzelne Nachrichtenbits in der abgehörten Gesamtnachricht, die an einen Mix gesendet wird, verändern. (Anders als im bisher beschriebenen Angriff auf die direkte RSA-Implementierung: Dort wird die gesamte Nachricht geändert.) Da der Mix ohnehin nur N ausgibt und Z geheimhält, müßte der Angreifer dann auch nur N verändern. Geht das, bzw. reicht es für den Angriff auf die direkte RSA-Implementierung aus? Wenn ja, warum? Wenn nein, warum nicht? ♢

Dieser Angriff auf RSA gelingt nur bei der Verwendung von (zusammenhängenden) Zufallsbits, die erst bei der Verschlüsselung über den gesamten Nachrichtenblock „verschmiert werden“. Dennoch sollte eine allgemeinere Lehre aus diesem Versuch gezogen werden. RSA sollte

nur mit Redundanzprädikaten [Pfit_89, Seite 88] verwendet werden, welche die Multiplikativität der Konzelation zerstören. Solche Redundanzprädikate sind bezüglich ihrer Sicherheit noch nicht vollkommen erforscht. Bei Mixen könnten die den Nachrichten beigefügten Adressen Redundanzprädikate darstellen. Bei einem modifizierenden Angriff auf die verschlüsselte Nachricht (wie gerade vorgestellt) müßte dann die Zerstörung der als nächstes benötigten Adresse sichergestellt werden. Der Mix fände in einem solchen Fall keine gültige Adresse mehr vor und könnte demnach die Nachricht auch nicht mehr weiterleiten. Adaptive Angriffe fielen somit aus.

Eine andere Schutzmöglichkeit gegen den beschriebenen Angriff stellt die sogenannte hybride Verschlüsselung dar [Pfit_89, PfPW_91]. Hier wird die Gesamtnachricht mit einem (schnelleren) symmetrischen Konzelationssystem verschlüsselt, welches keine multiplikativen Eigenschaften aufweist. Der eigens dafür erzeugte Schlüssel wird nach der Verschlüsselung der Nachricht angefügt und dieser (wesentlich kürzere) Teil wie üblich mit dem asymmetrischen Konzelationssystem verschlüsselt. Der Schlüssel des symmetrischen Konzelationssystems stellt gleichzeitig eine zufällige Bitkette dar. Ein Angriff auf den mit RSA verschlüsselten Teil wird mit hoher Wahrscheinlichkeit den Schlüssel verändern und somit ein korrektes Entschlüsseln des übrigen Teiles unmöglich machen. Um dies zu erkennen, wäre zusätzliche Redundanz vorzusehen.

4 Empfängeranonymität und Kanäle

Bisher ist immer nur von einem Senderanonymitätsschema gesprochen worden. Auch der vorgestellte MIX-Netz-Demonstrator beinhaltet nur ein Senderanonymitätsschema. Wünschenswert aber wäre auch eine Empfängeranonymität, d.h. der Sender einer Nachricht kennt den wahren Empfänger der Nachricht nicht, bzw. das Empfangen von Nachrichten bleibt unerkannt. In diesem (zweigeteilten) Trockenversuch nun soll Ihnen als Abschluß zum einen die Empfängeranonymität nahegebracht und in einem zweiten Teil eine andere Art und Weise der Nachrichtenvermittlung (sogenannte Kanäle) dargeboten werden.

4.1 Empfängeranonymität

Die einfachste Methode der Empfängeranonymität ist die Verteilung (*broadcast*). In dem unten gezeigten Bild 7 ist unklar, welcher der vier Empfänger Sven Eric, Anke, Helga und Ullrich der Empfänger einer von Jürgen geschickten, geheimnisvollen Nachricht ist. Die physische Empfangsoperation ist für alle Beteiligten gleich, aber nur der wahre Empfänger wird nach der Entschlüsselung eine sinnvolle (hier: lesbare) Nachricht erhalten.

An diesem Beispiel (Verteilung einer Nachricht ohne weitere Protokolle) soll der Begriff Empfängeranonymität genauer untersucht werden. Verschiedene Sichtweisen ergeben nämlich ganz andere Inhalte für diesen Begriff:

- Ein **Außenstehender** (der die Verteilung der Nachricht durch den letzten Mix beobachtet) erfährt nichts über den Empfänger der Nachricht. Für ihn kommen aber nur Netzteilnehmer als Empfänger in Frage. Die Menge aller möglichen Empfänger von Nachrichten wird für ihn also auf die Menge der Empfänger mit einem Netzabschluß

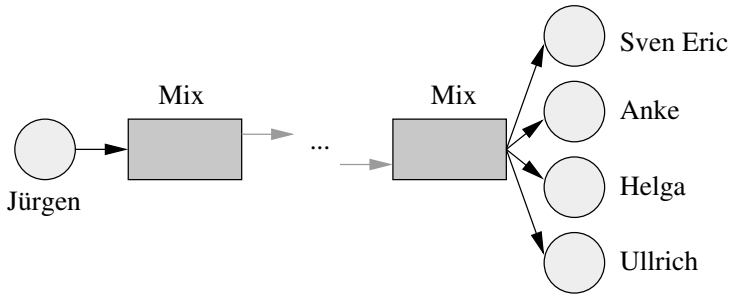


Abbildung 7: Zur Empfängeranonymität

eingeschränkt. Ist diese Menge groß genug (z.B. Menge aller Telefon- oder Fernsehbesitzer in Europa), kann immer noch ohne Bedenken von Empfängeranonymität gesprochen werden.

- Für den Sender der Nachricht ist der Empfänger anonym, weil er durch die Beobachtung der Verteilung nicht mehr über den Empfänger erfährt, als er ohnehin schon weiß. Die ihm bekannte Adresse kann aus einem öffentlichen Verzeichnis stammen, welches nichts über den eigentlichen Empfänger aussagt.

Empfängeranonymität ist demnach ein relativer Begriff. Das folgende Schema garantiert eine Empfängeranonymität auf der Basis von Mixen. Die Sichtweise soll die des Senders sein.

Dem in Abschnitt 2 beschriebenen Senderanonymitätsschema liegt die Idee zugrunde, daß ein Sender S, welcher anonym bleiben möchte, die Gestalt einer Nachricht über die Mixe selbst bestimmen kann. Dies erreicht er durch die Zufallsbits, die er seiner Nachricht hinzufügt. Diese Geheimnisse gewähren ihm die gewünschte Anonymität gegenüber den anderen Netzteilnehmern oder außenstehenden Angreifern.

Anmerkung: Der Weg einer Nachricht kann allgemein bekannt sein! Selbst wenn ein Angreifer den Weg einer Nachricht über die Mixe kennt, erfährt er trotzdem nichts über die Kommunikationsbeziehung, weil die Umcodierung eine Zuordnung von den vom Mix empfangenen zu den vom Mix weitergereichten Nachrichten verhindert. Diese Umcodierung wird vom Sender der Nachricht insofern festgelegt, als daß er die Zufallsbits wählt. Darum wird von einer **Vorgabe der Gestalt** oder von einer **Vorgabe der Umcodierung** gesprochen.

Verfährt man nun andersherum, bestimmt also der Empfänger die Umcodierung der Nachricht, die ein Sender ihm zukommen lassen soll, so wäre die Empfängeranonymität in dem Sinne erreicht, daß der Sender den Empfänger nicht kennt. Damit ergeben sich sofort zwei Probleme:

1. Wie erfährt der Sender, daß der Empfänger etwas empfangen möchte bzw. wie meldet der Sender einen Sendewunsch an?

2. Wie kann die Vorgabe der Umcodierung des Empfängers den beteiligten Stationen (inklusive Sender) vorgegeben werden, ohne daß diese mehr als nötig davon erfahren, die Vorgabe aber verwendbar bleibt?

Um das erste Problem zu umgehen, führen wir ein zweischrittiges Protokoll ein. Ein Netzteilnehmer (etwa Helga) möchte eine Information von einem anderen Netzteilnehmer (etwa Jürgen) haben und dabei unerkannt bleiben. Helga schickt dafür zunächst die Frage (Anforderung, *request*) an Jürgen und dieser sendet die Antwort (*response*) zurück. Mit dieser Maßnahme kann nun die (Empfänger-) Anonymität von Helga gegenüber Jürgen gewährt werden. Helga gibt die Umcodierung der Nachricht vor und schickt die Vorgabe mit der Frage an Jürgen. Dieser beantwortet die Frage, fügt die Vorgabe der Umcodierung bei und überläßt es den Mixen, daraus schlau zu werden...

Dies läßt sich allerdings auch etwas genauer und sorgfältiger formulieren:

AUFGABE 11

Helga möchte von Jürgen die Anfangszeit des Theaterstückes wissen, welches er am Abend sehen möchte. Sie will auch dorthin kommen und ihn überraschen; will also bei der Auskunft ihren Namen nicht preisgeben. Damit Jürgen ihr die Antwort über die Mixe MIX 1 und MIX 2 (in dieser Reihenfolge, siehe Bild 8) schicken kann (was er, gutmütig wie er ist, bei allen anonymen Anfragen tut) liefert sie ihm eine **anonyme Rückadresse** [Chau_81] mit.

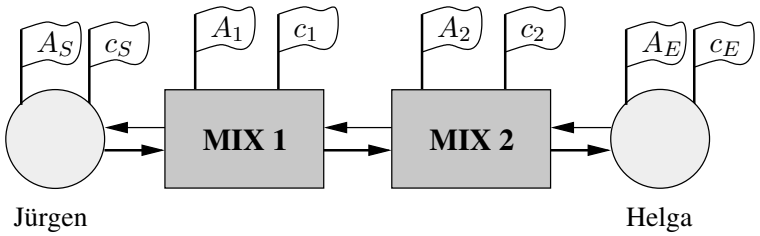


Abbildung 8: Zu Aufgabe 11

Die anonyme Rückadresse sei zunächst mit *„aR“* bezeichnet, die Frage von Helga laute *„Wann?“* und Jürgen wird mit *„20 Uhr“* antworten. Weiterhin weiß Helga um das tolle MIX-Netz mit Senderanonymität und bedient sich dessen, um Jürgen die Frage zu schicken. Wie sieht die Nachricht aus, die Helga gemäß Bild 8 an Jürgen schicken wird? (Hinweis: Dies ist nur eine Auffrischung des Abschnittes 2. Verwenden Sie die dort eingeführte Notation.) ♢

Jürgen erhält, wenn alles gut geklappt hat, sowohl *„Wann?“* als auch *„aR“*. *„aR“* soll nun so aufgebaut sein, daß

- Jürgen erfährt, an welchen Mix er seine Antwort zuerst senden soll und
- Jürgen nicht erkennen kann, an welchen Netzteilnehmer (Empfänger) und über welche Mixe die Nachricht zurückgehen wird,
- Jürgen *„aR“* nur an seine (vielleicht verschlüsselte) Antwort *„20 Uhr“* anfügen und gemäß dem Schema an den ersten Mix zurückschicken muß.

Die Idee und der Aufbau der anonymen Rückadresse soll nun anhand dieser drei Punkte Stück für Stück hergeleitet werden.

Jürgen muß notwendigerweise den Mix erfahren, an den er seine Antwort zuerst schicken soll. Dies geschieht durch Angabe einer Adresse, die von Helga vorgegeben wird. Helga wollte die Antwort von Jürgen über die Mixe MIX 1, MIX 2 zurückgeschickt wissen (siehe Bild 8). Darum teilt sie Jürgen als Adresse des ersten Mixes A_1 mit.

„aR“ enthält natürlich weitere Weginformationen, sonst verliefe sich Jürgens Antwort bei MIX 1. Diese Information, ein **Rückadreßteil R_1**, darf für Jürgen nicht lesbar, muß demnach verschlüsselt sein. Dies gilt aber auch für alle verwendeten Mixe: Jeder Mix soll (wie Jürgen) nur die Adresse der nächsten Station erfahren, nicht aber alle weiteren Stationen. Darum müssen die anonymen Rückadreßteile verschlüsselt, für die Mixe aber zumindest teilweise entschlüsselbar sein. Demnach kann dies nur mit den öffentlichen Schlüsseln des asymmetrischen Konzelationssystems der Mixe geschehen. Die Konstruktion garantiert, daß jede Station nur die Adresse der nächsten erfährt, aber keine weiteren.

Natürlich sollen die verwendeten Mixe die Antwort von Jürgen nicht lesen können. Darum verschlüsselt Jürgen die Antwort mit einem Schlüssel k_0, der ein Schlüssel eines symmetrischen Konzelationssystems sein kann. Dieser Schlüssel ist speziell für diesen Zweck von Helga generiert worden und stand mit in der anonymen Rückadresse aR.

Vollständig lautet aR also (k_0, A_1, R_1). A_1 ist die Adresse des Mixes, an den Jürgen seine Antwort zuerst schicken soll. Es sei hier der MIX 1. R_1 ist der Teil der Rückadresse, den Jürgen nicht lesen, der von MIX 1 aber interpretiert (entschlüsselt) werden kann. Demnach sendet Jürgen pflichtbewußt eine Nachricht $N_1 := R_1, I_1$ mit $I_1 := k_0$(„20 Uhr“) an MIX 1.

AUFGABE 12

a) MIX 1 entschlüsselt den von Jürgen erhaltenen Rückadreßteil R_1 mit dem eigenen, geheimen Schlüssel d_1. Was muß MIX 1 nach dem oben Gesagten alles finden, damit die Antwort von Jürgen weitergesendet werden kann. Wie also muß R_1 ausgesehen haben (verwenden Sie die formale Notation)?

b) Nach wie vor sollen die Mixe das Erscheinungsbild von Nachrichten ändern, sprich die Nachrichten (hier: die Antwort von Jürgen) umcodieren. Warum kann dies weder mit dem geheimen noch mit dem öffentlichen Schlüsseln des Mixes geschehen. Was also muß der MIX 1 nach dem Entschlüsseln von R_1 noch alles finden? (Analog zu dem, was Jürgen empfangen hat.)

   Wie geht es weiter, was findet MIX 2,...?

c) Geben Sie nun formal die komplette Rückadresse aR an, wie sie von Helga formuliert wurde. ◊

AUFGABE 13 (optional)

Warum sind bei der anonymen Rückadresse keine Zufallsbits Z_i erforderlich? Nach welchem rekursiven Bildungsschema wird aR gebildet und nach welchem rekursiven Schema kommt Jürgens Antwort bei Helga an? ◊

Wie gezeigt wurde, gibt es je ein Umcodierungsschema für Senderanonymität und für Empfängeranonymität. In beiden Fällen aber war die eine Seite nicht vor der anderen anonym. Diese Schwäche ließe sich durch Kombination der beiden Umcodierungsschemata erreichen,

doch ist hierfür eine weitere Instanz notwendig. Ein Sender läßt dieser Instanz eine Nachricht auf senderanonymen Weg zukommen. Diese Instanz, als „Umschalter“ zwischen Sender- und Empfängeranonymität denkbar, sendet die erhaltene Nachricht auf Empfängeranonymität garantierende Weise weiter. Die Instanz kennt durch diese Konstruktion weder Sender noch Empfänger und auch diese beiden Netzteilnehmer sind voreinander geschützt. Einen ähnlichen Weg beschreitet auch das letzte Kapitel des Versuches.

4.2 Kanäle

In diesem zweiten Teil des Versuches soll eine etwas effizientere Methode vorgestellt werden, wie sich Nachrichten übertragen lassen. Bisher wurden Nachrichten samt Netzinformationen gemeinsam verschickt und bearbeitet. Die gesamte Nachricht wird von den Mixen zunächst gepuffert und dann gemeinsam mit anderen Nachrichten bearbeitet und weitergereicht. Auf diese Art und Weise ist der eigentliche Nachrichteninhalt (die Nutzdaten) sehr lange unterwegs.

Das notwendige asymmetrische Konzelationssytem bei Mixen bedingt eine weitere Zeitverzögerung, wenn der gesamte Nachrichteninhalt hiermit verschlüsselt ist. In diesem Fall müssen, da es sich um eine Blockchiffre handelt, immer ganze Blöcke abgewartet werden, bevor mit der Entschlüsselung begonnen werden kann. Außerdem sind asymmetrische Konzelationssysteme immer noch bedeutend langsamer als symmetrische [Bras_88].

Bei der sogenannten *hybriden Verschlüsselung* wird dieser Nachteil behoben. Nur die eigentliche Verwaltungsinformation ist in einem kurzen Teil, dem *Nachrichtenkopf*, mit dem asymmetrischen Konzelationssystem verschlüsselt. Der restliche Teil der Nachricht ist durch ein symmetrisches Konzelationssystem geschützt. Der zugehörige Schlüssel steht mit in dem Nachrichtenkopf und wird dort von dem entschlüsselnden Mix gefunden. Auf diese Art und Weise kann auch der restliche Teil der Nachricht von dem Mix umcodiert werden. Doch auch hier ist die Nutznachricht mit der Verwaltungsinformation gekoppelt und deswegen länger unterwegs als notwendig.

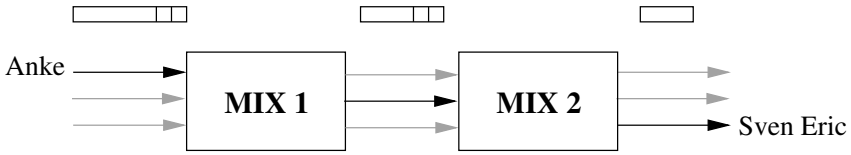


Abbildung 9: Nachrichtenbearbeitung durch Mixe

Es liegt daher nahe, die Verwaltungsinformation von der eigentlichen Nachricht zu trennen, um diese nach Festlegung des Nachrichtenweges schnell übermitteln zu können. Steht ein solcher Nachrichtenweg ersteinmal fest, so kann er auch auf längere Zeit für die Datenübertragung genutzt werden. Fallen sehr große Datenmengen an, so sind diese auf alle Fälle schneller am Ziel als bei den oben beschriebenen Schemata. Die Nachrichtenübermittlung teilt sich nun in zwei Teile: Eine Aufbauphase, bei der die Mixe nach den bisher vorgestellten Schemata verfahren, und eine Übertragungsphase, bei der die Nachricht von den Mixen mit einer

(schnellen) symmetrischen Stromchiffre entschlüsselt und weitergereicht wird (siehe Bild 10 und 11).

In der Aufbauphase (Bild 10) wird ein **Kanal** vom Sender zum Empfänger geschaltet. Derjenige, der den Kanal aufbauen möchte (ein Sender oder ein Empfänger), verwendet hierzu eine kurze **Kanalaufbaunachricht**. Sie enthält für jeden Mix und den Empfänger je zwei Informationen:

- Die Adresse der Station, an die die Kanalaufbaunachricht (und anschließend die eigentliche Nachricht) weitergereicht werden soll. (Der Empfänger benötigt diese Information natürlich nicht.)
- Den Schlüssel eines symmetrischen Konzelationssystems, mit dem der Mix (und der Empfänger) die erhaltene Nachricht entschlüsseln soll.

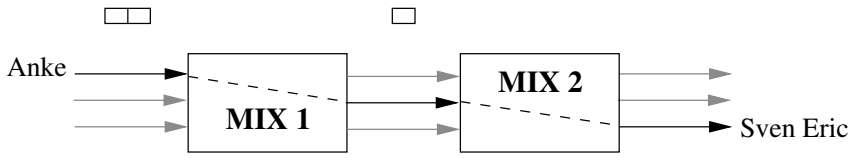


Abbildung 10: Aufbauphase bei Kanälen

Die Kanalaufbaunachricht wird nach dem bekannten, indirekten Umcodierungsschema für Senderanonymität verschickt. Nach Bild 10 und der allgemeinen Notation sähe eine von Anke an Sven Eric verschickte Kanalaufbaunachricht folgendermaßen aus:

$$A_1, c_1(sK, k_1, A_2, c_2(sK, k_2, A_{SE}, c_{SE}(sK, k_{SE})))$$

Die Mitteilung sK interpretieren die Mixe als „schalte Kanal“ und der Empfänger erkennt hieran, daß es sich um eine Kanalaufbaunachricht gehandelt hat. Diese Aufbaunachricht „bahnt“ sich einen Weg über die Mixe und jeder Mix erfährt die Adresse, an die diese Aufbaunachricht weitergeschickt werden soll. Anke kann nun ihre Nachricht mit den Schlüsseln k_{SE}, k_2, k_1 (in dieser Reihenfolge!) verschlüsseln und an den ersten Mix (MIX 1) schicken. Dieser entschlüsselt mit k_1 und leitet die Daten umgehend an MIX 2 weiter. Dieser entschlüsselt wiederum und so empfängt Sven Eric die nur noch mit k_{SE} verschlüsselten Daten. Er selbst besitzt aber diesen Schlüssel, da er ihn von Anke erhalten hat. So liest er voller Spannung, was sie ihm geschickt hat.

Der große Vorteil dieses Verfahrens ist nun, daß nur noch kurze Kanalaufbaunachrichten von den Mixen gepuffert und auf Wiederholungen getestet werden müssen [Pfit_89, Seite 82, 158]. Der eigentliche Nachrichtenstrom, der in der Regel den Hauptteil der Gesamtnachricht darstellen wird, kann dagegen schnell und fortwährend übermittelt werden (siehe auch Bild 11). Die Mixe entschlüsseln mit den ihnen mitgeteilten Schlüsseln.

In dem obigen Beispiel hat Anke eine Schlüsselfolge generiert, die Kanalaufbaunachricht formuliert und gesendet. Die verwendeten Mixe entschlüsseln die Nachricht mit den ihnen mitgeteilten Schlüsseln. Da Anke der Sender und Sven Eric der Empfänger der Nachricht ist,

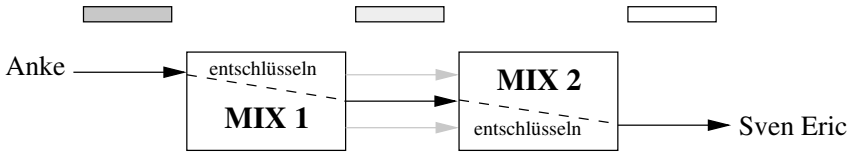


Abbildung 11: Übertragungsphase bei Kanälen

spricht man sinnigerweise von einem **Sendekanal**. Hierbei ist Anke (als Sender) anonym vor Sven Eric (als Empfänger).

Vollkommen analog hierzu hätte auch Sven Eric eine Schlüsselfolge generieren, die Kanalaufbaunachricht senden und trotzdem von Anke eine Nachricht empfangen können. In diesem Fall spricht man von einem **Empfangskanal**. Die verwendeten Mixe und Anke erhalten von Sven Eric mit der Kanalaufbaunachricht je einen Schlüssel. Die Mixe verschlüsseln die bereits von Anke verschlüsselte und gesendete Nachricht und Sven Eric kann, da er selbst den Mixen (und Anke) die Schlüssel mitgeteilt hat, die Nachricht entschlüsseln. Hier ist Sven Eric (als Empfänger) anonym vor Anke (als Sender).

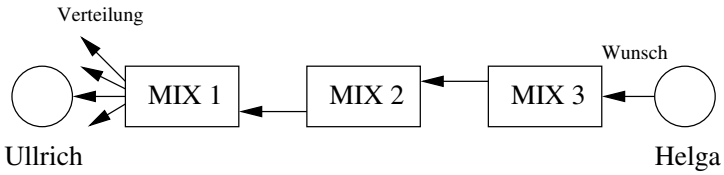
Um wirkliche Anonymität zu erreichen, müssen Kanäle gemeinsam aufgebaut und gemeinsam aufgebaute Kanäle auch wieder gemeinsam abgebaut werden. Dies entspricht einer gemeinsamen Bearbeitung der Nachrichten bei den bereits beschriebenen Umcodierungsschemata. Damit ausreichend Kanalaufbaunachrichten über das Netz gehen, greift man auf bedeutungslose Kanalaufbaunachrichten zurück. Mit diesen schalten Netzteilnehmer Kanäle zu sich selbst, ohne daß dies von außen erkennbar wäre.

Allgemein gesprochen muß jeder Teilnehmer immer gleich viele Sende- und Empfangskanäle unterhalten. Dies gilt auch für die Mixe. Besteht kein Bedarf für eine Nachrichtenübermittlung, so werden diese Kanäle mit bedeutungslosen Nachrichten gefüllt (*dummy traffic*).

Wie auch bei den Umcodierungsschemata lassen sich beide Verfahren kombinieren. Dies geschieht hier durch Verknüpfung eines Sendekanals und eines Empfangskanals bei einem Mix. Damit die richtigen Kanäle verknüpft werden, benötigt man ein **gemeinsames Kanalkennzeichen**, welches dem Mix mitgeteilt werden muß. Die Problematik liegt hier bei „gemeinsam“. Eine der beiden Parteien, ob nun Sender oder Empfänger, muß dieses Kanalkennzeichen festlegen und der anderen auf Anonymität wahrende Weise mitteilen. Diese *rufende* Partei teilt ihrem Kommunikationspartner (der *gerufenen* Partei) mit dem Kanalkennzeichen auch mit, ob er eine Nachricht empfangen oder senden möchte. Anschließend baut der Sender einen Sendekanal und der Empfänger einen Empfangskanal zu einem gemeinsamen Mix auf. (Auch diese Angabe kann in der Kanalwunschnachricht stehen. Der verbindende Mix kann aber auch eine im Netz festgelegte Instanz sein.) Von beiden Seiten wird dem Mix das vereinbarte Kanalkennzeichen mitgeteilt, so daß eine beidseitig anonyme Verbindung zustande kommt.

Im folgenden soll das gesamte Protokoll noch einmal im Überblick dargestellt werden. Zur Vereinfachung sind nur drei Mixe dargestellt. Weitere Nachrichten, die im Netz gesendet werden, sind ebenfalls nicht eingezeichnet. Es dürfte aber klar sein, daß zwischen den Mixen noch beliebig viele weitere Mixe liegen können und daß diese nur dann arbeiten, wenn sie ausreichend viele Nachrichten empfangen haben. MIX 2 sei o.B.d.A. der verbindende Mix.

Helga möchte den Plausch beginnen...

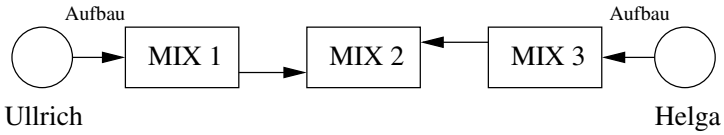


Helga, als rufende Partei, sendet eine Kanlwunschnachricht *„Wunsch“* über die von ihr gewählten Mixe. Mit dieser Nachricht erreicht sie dreierlei:

1. Helga teilt Ullrich das von ihr gewählte Kanalkennzeichen mit. Helga bleibt dabei anonym, weil sie ein Senderanonymitätsschema verwendet. Ullrich bleibt durch die vom Protokoll vorgesehene Verteilung anonym.

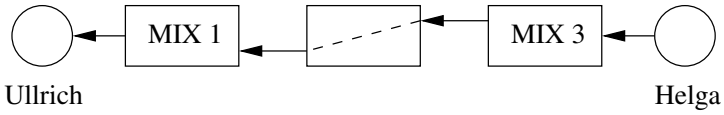
2. Helga teilt Ullrich ihren Kommunikationsschlüssel k eines symmetrischen Konzelationssystems mit.

3. Helga gibt die Richtung des Kanals an. In diesem Beispiel sei die Richtung Helga $\rightarrow$ Ullrich gewählt, weil Helga viel zu erzählen hat.



Ullrich baut nun (weil er neugierig ist) einen Empfangskanal zu MIX 2 auf. Mit der dazu verwendeten Kanalaufbaunachricht teilt er allen von ihm gewählten Mixen einen Schüssel eines symmetrischen Konzelationssystems mit. Der verbindende Mix (MIX 2) erhält zusätzlich das von Helga gewählte Kanalkennzeichen.

Helga ihrerseits baut einen Sendekanal zu MIX 2 auf. Auch sie teilt allen gewählten Mixen einen Schlüssel eines symmetrischen Konzelationssystems mit. Ebenso erhält der verbindende Mix (MIX 2) zusätzlich das gemeinsame Kanalkennzeichen.



MIX 2 kann nun anhand des gemeinsamen Kanalkennzeichens den Empfangskanal von Ullrich und den Sendekanal von Helga verbinden. Helga verschlüsselt ihre Nachricht mit ihrem Kommunikationsschlüssel k und anschließend gemäß dem Sendeschema mit allen von ihr den Mixen mitgeteilten Schlüsseln. Diese entschlüsseln sukzessive und MIX 2 entschlüsselt mit dem von Helga gewählten Schlüssel (danach ist die Nachricht von Helga nur noch mit k verschlüsselt!) und verschlüsselt mit dem von Ullrich gewählten Schlüssel. Alle weiteren Mixe verschlüsseln nun mit Ullrichs Schlüssel und dieser kann, da er sie alle und den Kommunkationsschlüssel k von Helga kennt, die Neuigkeiten erfahren.

Literatur

[AssPf2_90] Ralf Aßmann, Andreas Pfitzmann: Betriebsarten von Blockchiffren, Versuchsunterlagen zum 2. Versuch des Praktikums „Angriffstoleranz in Rechnernetzen“ (1990), Institut für Rechnerentwurf und Fehlertoleranz der Universität Karlsruhe.

[Bleu_91] Gerrit Bleumer: Eine Praktikumsumgebung für Kryptoprotokolle sowie Erstellung der Versuche „Zahlentheoretische Algorithmen“ und „Asymmetrische Konzelationssysteme“. Diplomarbeit, Institut für Rechnerentwurf und Fehlertoleranz der Universität Karlsruhe (1991).

[Boett_89] Manfred Böttger: Untersuchung der Sicherheit von asymmetrischen Kryptosystemen und MIX-Implementierungen gegen aktive Angriffe (1989), Studienarbeit am Institut für Rechnerentwurf und Fehlertoleranz der Universität Karlsruhe.

[Boett_90] Manfred Böttger: Realisierung des DC-Netz-Versuchs und einer einheitlichen Praktikums-Netzschnittstelle. Diplomarbeit, Institut für Rechnerentwurf und Fehlertoleranz der Universität Karlsruhe (Juli 1990).

[Bras_88] Gilles Brassard: Modern Cryptology - A Tutorial. LNCS 325, Springer, Berlin (1988).

[Chau_81] David Chaum: Untraceable Electronic Mail, Return Addresses, and Digital Pseudonyms. Communications of the ACM, Band 24(2) (1981), 84–88.

[Cha1_84] David Chaum: A New Paradigm for Individuals in the Information Age. In: 1984 IEEE Symposium on Security and Privacy, IEEE Computer Society Press, Washington (1984), 99–103.

[Davi_82] G. Davida: Chosen Signature Cryptanalysis of the RSA (MIT) Public-Key-Cryptosystem. Technischer Bericht TR-CS-82, University of Wisconsin, Milwaukee (October 1982).

[Denn_84] Dorothy E. Denning: Digital Signatures with RSA and other Public-Key-Cryptosystems. Communications of the ACM, Band 27(4) (1984), 388–392.

[DiHe_76] Whitfield Diffie, Martin E. Hellman: New Directions in Cryptography. IEEE Transactions on Information Theory, Band 22(6) (1976), 644–654.

[FGJP_98b] Elke Franz, A. Graubner, Anja Jerichow, Andreas Pfitzmann: Comparision of Commitment Schemes Used in Mix-Mediated Anonymous Communication for Preventing Pool-Mode Attacks. In: 3rd Australasian Conference on Information Security and Privacy (ACISP ’98), Band 1438 von *LNCS*, Springer-Verlag, Berlin (1998), 111–122.

[FGJP_98a] Elke Franz, A. Graubner, Anja Jerichow, Andreas Pfitzmann: Modelling mix-mediated anonymous communication and preventing pool-mode attacs. In: 14th IFIP International Conference on Information Security (IFIP/SEC'98), Chapman & Hall, London (1998).

[Merr_83] Michael John Merritt: Cryptographic Protocols. Dissertation, School of Information and Computer Science, Georgia Institute of Technology (February 1983).

[Pfi1_85] Andreas Pfitzmann: How to implement ISDNs without user observability - Some remarks. Fakultät für Informatik, Universität Karlsruhe, Interner Bericht 14/85.

[Pfit_86] Andreas Pfitzmann: Die Infrastruktur der Informationsgesellschaft: Zwei getrennte Fernmeldenetze beibehalten oder ein wirklich datengeschütztes errichten? Datenschutz und Datensicherung DuD, Band 6 (1986), 353–359.

[Pfit_89] Andreas Pfitzmann: Diensteintegrierende Kommunikationsnetze mit teilnehmerüberprüfbarem Datenschutz; Universität Karlsruhe, Fakultät für Informatik, Dissertation, Feb. 1989. IFB 234, Springer-Verlag, Heidelberg (1990).

[PfPW_91] Andreas Pfitzmann, Birgit Pfitzmann, Michael Waidner: ISDN-MIXes - Untraceable Communication with Very Small Bandwidth Overhead. In: Proc. Kommunikation in verteilten Systemen, Band 267 von *Informatik-Fachberichte*, Springer-Verlag, Heidelberg (1991), 451–463.

[RSA_78] R. L. Rivest, A. Shamir, L. Adleman: A Method for Obtaining Digital Signatures and Public-Key Cryptosystems. Communications of the ACM, Band 21(2) (1978), 120–126, reprinted: 26/1 (1983) 96-99.

[Sha1_49] C. E. Shannon: Communication Theory of Secrecy Systems. The Bell System Technical Journal, Band 28(4) (1949), 656–715.

A Beschreibung des Mix-Demonstrators

A.1 Konfiguration des Mix-Netzes

Das Hauptfenster:

Links im Hauptfenster können neue Teilnehmer und neue Mixe mit den entsprechenden ausgewählten Namen dem Mix-Netz hinzugefügt werden. Das Feld „Sichtbar“ gibt an, ob Mix oder Teilnehmer am Bildschirm angezeigt werden sollen. **Achtung:** Beim Start des Demonstrators gibt es noch keine Teilnehmer und Mixe. Sie müssen also wenigstens einen Mix und einen Teilnehmer hinzufügen, um Aktionen im Mixnetz ausführen zu können.

In der **Mitte** des Hauptfensters können Eigenschaften für Mixe und Teilnehmer und Häufigkeiten für Dummy-Nachrichten angegeben werden.

Unter „Teilnehmer“ wird das Senden von Pseudonachrichten konfiguriert. Pseudonachrichten sind bedeutungslose Nachrichten, die von Teilnehmer zu Teilnehmer gesendet werden. Der

Empfänger wirft derartige Nachrichten weg. Das Feld „markieren“ gibt an, ob die Nachrichtenkennung die Nachricht als Pseudonachricht ausweisen soll. (Eine solche Kennung wird zur visuellen Verfolgung der Nachrichten benötigt.) Weiterhin kann angegeben werden, über wie viele Mixe eine Pseudonachricht maximal gesendet werden soll und mit welcher Wahrscheinlichkeit eine gesendet wird. Diese Einstellungen gelten jeweils für alle Teilnehmer.

Unter „Mix“ wird der Arbeitsmodus (Pool- oder Batch-Modus) aller Mixe konfiguriert. Für den ausgewählten Modus kann dann die Schubgröße bzw. die minimale Poolgröße eingestellt werden. Der Schalter „Wiederholungs-Filter“ gibt an, ob die Mixe Wiederholungen von Nachrichten ignorieren.

Unter „Dummy“ wird das Senden von Dummynachrichten konfiguriert. Dummynachrichten werden von Teilnehmern und/oder Mixen über eine beliebige Anzahl von Mixen gesendet, wobei der Empfänger ein Mix ist. Dummynachrichten werden wie Pseudonachrichten vom Empfänger weggeworfen. Es kann konfiguriert werden, ob Teilnehmer und/oder Mixe Dummynachrichten senden sollen. Weiterhin kann konfiguriert werden, ob Dummynachrichten eine entsprechende Kennung erhalten sollen („markieren“) und mit welcher Häufigkeit sie gesendet werden.

Rechts im Hauptfenster wird die Ablaufgeschwindigkeit der Schritte des Mix-Netzes konfiguriert. Bei Periodendauer > 0 beginnt der nächste Schritt jeweils nach Ablauf der Periodendauer. Mit „Trigger“ kann manuell der nächste Schritt durchgeführt werden.

Der Teilnehmer:

Ein Teilnehmer kann Nachrichten senden und empfangen. Bevor er Nachrichten über das Mix-Netz senden kann, muß er Mixe auswählen, über die er senden will. („Mixfolge“ ⇒ Auswahl eines Mix-Namen ⇒ „→Liste“ oder „zufällig“). „Lösche Mixliste“ macht diese Auswahl rückgängig. Zum Senden einer Nachricht muß ein Empfänger ausgewählt und Nachrichteninhalt und Kennung angegeben werden. Mit Hilfe der Kennung kann eine Nachricht durch die einzelnen Mixe verfolgt werden. Die Taste „Senden“ sendet die Nachricht ab. Im Feld „Empfänger“ werden die empfangenen sinnvollen Nachrichten (d.h. keine Pseudo- oder Dummynachrichten) angezeigt.

Der Mix:

Im oberen Teil wird der Nachrichtenpuffer eines Mixes angezeigt. Im unteren Teil werden jeweils die Nachrichten angezeigt, die im nächsten Schritt ausgegeben werden.

Der Netzwerkmonitor:

Unter „Spezial“ ⇒ „Netzwerkmonitor“ wird der Netzwerkmonitor gestartet. Dort werden die Schritte des Mix-Netzes in Diagrammform dargestellt. Es wird jeweils das Senden von Nachrichten mit der zugehörigen Nummer des Schrittes angezeigt. Ein Quadrat zeigt an, daß die entsprechende Nachricht an einen Mix gesendet wurde. Eine Raute gibt an, daß die Nachricht an einen Teilnehmer gesendet wurde. Die Farbe gibt den Empfänger der Nachricht an.

Der Replay-Angreifer:

Dieser Angreifer kann unter „Spezial“ ⇒ „Replay-Angreifer“ gestartet werden. Er vervielfältigt gesendete Nachrichten seines Opfers und sendet die Kopien an den Mix, der auch die Original-Nachricht empfangen hat. Der Angriff erfolgt entweder nur auf eine Nachricht oder beliebig

lange. Dazu muß der Angreifer „aktiviert“ sein. „Nachrichten/Angriff“ gibt die Anzahl der Kopien an, die von der Originalnachricht gemacht werden.

Allgemeiner Hinweis:
Eine Mixnetz-Konfiguration kann auch gespeichert „Mix-Demo“ ⇒ „Speichern“ und wieder geladen „Mix-Demo“ ⇒ „Laden“ werden.