Technische Universität Chemnitz-Zwickau

Fakultät für Informatik

Theoretische Informatik und Informationssicherheit

Technischer Report

TR-95-04a

Ein neues Key-Escrow System mit aktivem Abhörer

Patrick Horster · Markus Michels · Holger Petersen

März 1995

Straße der Nationen 62 Tel: 0371-531-1479

D-09111 Chemnitz Fax: 0371-531-1810

Ein neues Key Escrow System mit aktivem Abhörer*

Patrick Horster · Markus Michels · Holger Petersen

Theoretische Informatik und Informationssicherheit
Technische Universität Chemnitz-Zwickau
Straße der Nationen 62, D-09111 Chemnitz
E-mail: {pho,mmi,hpe}@informatik.tu-chemnitz.de

Zusammenfassung

Während der letzten beiden Jahre fand eine lebhafte Diskussion über Escrow Technologien statt. In diesem Beitrag geben wir einen kurzen Überblick über zwei Escrow Systeme, den hardwarebasierten Clipper Ansatz und den softwarebasierten Escrow Schlüsselaustausch mit aktivem Abhörer. Wir diskutieren die Schwächen beider Ansätze und schlagen ein neues softwarebasiertes Escrow-Schlüsselaustauschprotokoll mit aktivem Abhörer vor.

1. Einleitung

Die Verwendung von Verschlüsselungstechnologien in Geräten des alltäglichen Lebens (z.B. Telefonen) soll das unautorisierte Abhören vermeiden. Dies führt zu dem Problem, daß autorisierte Abhörer ebenfalls keine Information erhalten können. Folglich können Kriminelle solche sicheren Geräte zur Kommunikation verwenden, ohne annehmen zu müssen, abgehört zu werden. Andererseits bedroht die Verwendung von schwachen Verschlüsselungssystemen die Privatsphäre der Kommunikationspartner, da diese bei Einsatz entsprechender Ressourcen von jedem gebrochen werden können. Ein Vorschlag zur Lösung dieses Konflikts ist ein *Key Escrow System*, das den Schutz von nichtklassifizierten sensitiven Daten garantiert und gleichzeitig autorisierten Abhörern die Mithörmöglichkeit offenhält.

In einem Key Escrow System unterscheiden wir folgende Teilnehmer: Die *Escrow Agency* verwahrt die geheimen Schlüssel aller *Benutzer*. Sie kann in verschiedene *Escrow Agents* aufgespaltet werden, die jeweils Teile (shares) des geheimen Schlüssel verwahren. Um die Privatsphäre der ehrlichen Benutzer zu gewährleisten muß mindestens ein Escrow Agent vertrauenswürdig sein. Ein *Abhörer*, der durch einen Gerichtsbeschluß autorisiert

*Leicht ergänzte Übersetzung von [11]

ist, kann die Kommunikationsdaten eines verdächtigen Benutzers aufzeichnen. Ein *passiver Abhörer* leitet die aufgezeichneten Daten an die Escrow Agency weiter, die seine Autorisierung überprüft und gegebenenfalls die Daten entschlüsselt. Ein *aktiver Abhörer* erhält geheime Parameter von der Escrow Agency, die er zur eigenen Entschlüsselung der Daten verwenden kann.

Die Gegner dieses Konzepts befürchten, daß die Etablierung einer totalitären Gesellschaft ermöglicht werden könnte, wenn jede Handlung aller Bürger aufgezeichnet und durch einen „großen Bruder" überwacht werden kann (siehe Orwell [17]). Geheime Schlüssel der Benutzer können aufgrund ungeeigneter Verwahrung oder nicht vertrauenswürdiger Escrow Agents kompromittiert werden. Ein weiteres Argument ist das sogenannte Dammbruchargument: Ist das Konzept einmal realisiert, so wird es auch bei offensichtlichen Schwächen nicht mehr so schnell abgeschafft werden können. Da ferner eine Vorverschlüsselung durch Kriminelle nicht vermieden werden und die Existenz der Vorverschlüsselung nur von autorisierten Abhörern erkannt werden kann, wird das juristische Verbot von Vorverschlüsselung und damit die Einschränkung der persönlichen Verwendung von kryptographischen Algorithmen erzwungen. Es kann ein unterschwelliger Kanal (privater Kanal zwischen zwei Benutzern, dessen Existenz gegenüber Dritten verborgen ist) durch Kriminelle eingerichtet werden, daher ist die Beschränkung der privaten Verwendung von Kryptoverfahren gemäß der Argumentation der Gegner des Konzeptes unsinnig (siehe z.B. [14] für die Errichtung eines solchen Kanals bei ISDN-Telefonaten).

Die Befürworter des Konzepts heben die Notwendigkeit autorisierten Abhörens hervor, um Kriminelle zu überführen. Dieses würde ohne Key Escrow Systeme technisch wesentlich erschwert bzw. unmöglich werden. Obwohl sie zugeben müssen, daß Vorverschlüsselung nicht vermieden werden kann, führen sie an, daß bereits das Wissen um eine Vorverschlüsselung einen vorhandenen Verdacht verstärken kann. Ferner erfordert die Vorverschlüsselung oder die Etablierung eines unterschwelligen Kanals gewisse Kenntnisse sowie technische Voraussetzungen und die damit verbundene Kosten. Deshalb werden diese Schutzmethoden nicht immer von jedem Kriminellen eingesetzt werden können. Da heutzutage die meiste Kommunikation unverschlüsselt über unsichere Kanäle stattfindet, wäre der Einsatz eines durch einen Key-Escrow Mechanismus geschützten Kanals ein Fortschritt hinsichtlich dem Schutz der Privatsphäre der Benutzer.

Ein andere Ebene der Diskussion wird durch die Anforderungen an akzeptable Key-Escrow Systeme bestimmt. Wir vertreten die Ansicht, daß *wenn* ein Key-Escrow System jemals errichtet werden sollte, es *dann* dem Abhörer nur genau die Möglichkeiten zum Abhören bieten sollte, zu denen dieser vom Gericht autorisiert wurde. Es muß analysiert werden, welche vorgeschlagenen Verfahren die an sie gestellten Anforderungen erfüllen. Wir diskutieren unsere Anforderungen detailliert in Abschnitt 3.

Die Debatte über Escrow-Systeme verstärkte sich, als die US-Regierung den Escrow Encryption Standard (EES) 1993 der Öffentlichkeit präsentierte. Der Algorithmus wurde bekannter unter dem Namen „Clipper", der den verwendeten Chip bezeichnet, und den wir im folgenden ebenfalls verwenden werden. Da der EES mehrere kryptographische Schwächen enthält, gab es eine lebhafte Diskussion über den neuen Standard. Es wurde ein softwarebasierter Ansatz mit aktivem Angreifer vorgeschlagen, der einige Schwächen des Clipper Vorschlages vermeidet. Leider enthält er jedoch neue Schwächen, die in diesem Beitrag diskutiert werden. Um sie zu vermeiden, wird ein neues softwarebasiertes Verfahren mit aktivem Abhörer präsentiert, das die Vorteile des softwarebasierten Konzeptes bewahrt. Diese sind im einzelnen die Möglichkeit der preisgünstigen Realisierung anstelle eines ausforschungssicheren Chips, der öffentlich bekannte Algorithmus und das Konzept, daß das Abhören bereits in der Schlüsselaustauschphase des Protokolls stattfindet.

Wir geben einen Rückblick auf den Escrow Encryption Standard in Abschnitt 2, präzisieren unsere Anforderungen an Escrow Systeme und mögliche Lösungen in Abschnitt

3 und beschreiben das Escrow Schlüsselaustauschprotokoll in Abschnitt 4. Im folgenden diskutieren wir dessen Eigenschaften und präsentieren unseren Lösungsvorschlag in Abschnitt 6.

2. Der Escrowed Encryption Standard

Am 16 April 1993 gab die U.S. Regierung einen Vorschlag für ein symmetrisches Verschlüsselungssystem, den sogenannten Escrowed Encryption Standard (EES), bekannt. Dieser soll dem Schutz sensitiver Daten dienen [13, 15]. Der Kern dieses Vorschlages ist der ausforschungssichere (engl. tamper-proof) Verschlüsselungschip, genannt „Clipper“, zusammen mit dem Ansatz verteilter geheimer Schlüssel für den Fall des Key Escrows [6, 19]. Der Chip enthält eine klassifizierte Blockchiffre mit 64 bit Wortbreite, die auch „Skipjack“ genannt wird. Der Algorithmus benutzt 80 bit Schlüssel und besteht aus 32 Runden einer Feistel-Chiffre. Jeder Chip enthält die folgenden Komponenten:

1. einen Familienschlüssel F, der für alle Chips einer Familie gemeinsam gewählt wird,
2. eine Unit ID, die jeden Chip identifiziert sowie
3. einen geheimen Schlüssel U, mit dem alle verschlüsselten Nachrichten des Chips entschlüsselt werden können.

Nachdem ein Sitzungsschlüssel K mit Hilfe eines asymmetrischen Schlüsselaustauschprotokolls etabliert wurde, kann der Chip zum Verschlüsseln des Nachrichtenstroms m eingesetzt werden. Das Law Enforcement Field (LEAF), das zusammen mit der Nachricht übermittelt wird, hat die folgende Struktur:

$$LEAF = E_F(E_U(K)||\text{Unit ID}||checksum).$$

Der Austausch des LEAFs wird durch den Clipper Chip erzwungen, indem er das Chiffrat erst nach dem Erhalt eines gültigen LEAFs entschlüsselt. Die detaillierte Beschreibung der Chipprogrammierung und der Escrow Technik werden in [6, 19] gegeben. Der Algorithmus wird in Abbildung 1 skizziert. Die Autorisierungsprozedur für die Freigabe von Teilen des geheimen Unitkeys U an den aktiven Abhörer durch die Escrow Agencies wurden in [18] veröffentlicht.

Schwächen des Clipper Ansatzes

Es gab eine lange anhaltende Diskussion über den EES, die in [12, 15] zusammengefaßt wird. Die Hauptschwächen aus kryptographischer Sicht sind dabei:

- Der Skipjack-Algorithmus ist klassifiziert und konnte daher nicht durch die Öffentlichkeit analysiert und kritisiert werden.
- Nachdem das Abhören eines Teilnehmers einmal erlaubt ist, können alle weiteren Nachrichten, die mit diesem Chip verschlüsselt werden abgehört werden, da die Abhörer den Unit Key U von den Escrow Agents erhalten haben.
- Kriminelle, die Clipper einsetzen, können ihre eigene, geheime Vorver- bzw. Nachentschlüsselung zusammen mit dem Chip einsetzen, so daß ein aktiver Abhörer die Nachricht nicht mehr entschlüsseln kann.
- Blaze zeigte, wie man den Clipper Chip ohne die Übertragung eines gültigen LEAFs benutzen kann, so daß die Escrow Agency mit Hilfe dieses LEAFs den Schlüsseltext nicht mehr entschlüsseln kann, obwohl dieser mit Hilfe des Skipjack Algorithmus

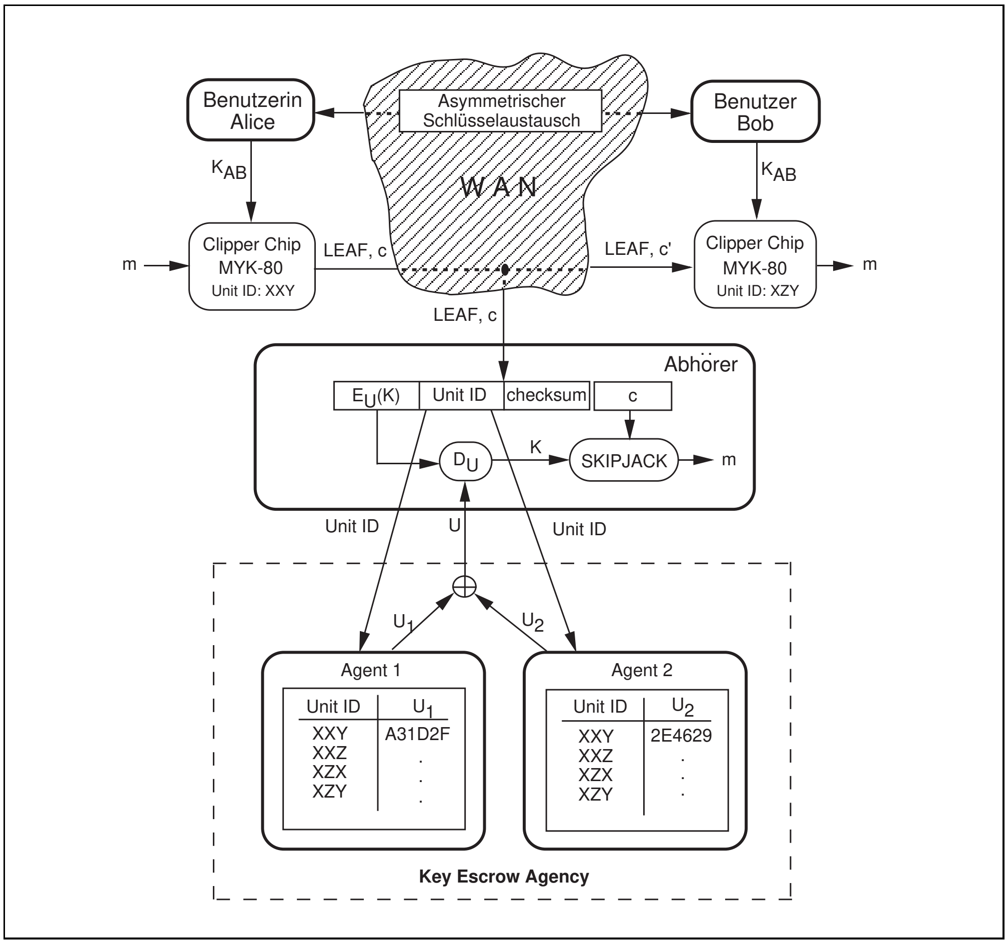

Abbildung 1: Der Escrowed Encryption Standard

verschlüsselt wurde [3]. Allerdings läßt sich diese Methode heute noch nicht für Echtzeit-Kommunikationen einsetzen [16].

- Ausforschungssicherheit des Clipper Chips ist unabdingbar, da sonst der Algorithmus mit inkorrektem LEAF benutzt werden könnte [7].

3. Anforderungen und Lösungen

Um die Schwächen des Clipper Ansatzes zu vermeiden, daß nachdem einmal das Abhören genehmigt wurde, die Abhörer alle weiteren Nachrichten abhören können und ebenfalls aufgezeichnete ältere Nachrichten entschlüsseln können, formulieren wir folgende Anforderung, die sich als Verschärfung der Forderung 5 in [1] ansehen läßt.

Anforderung: Wurde das Abhören für ein präzise definiertes Zeitintervall erlaubt, das entweder in der Vergangenheit, Gegenwart oder Zukunft liegen kann, so sollte die Autorisierung durch die Escrow Agency an den Abhörer derart limitierbar sein, daß dieser genau das autorisierte Zeitintervall abhören kann.

Es gab mehrere Vorschläge, um diese Anforderung (teilweise) zu erfüllen:

1. Eine Lösung von Diffie besagt, daß der Sitzungsschlüssel K_{AB} aufgeteilt wird und der LEAF die verschlüsselten Werte $E_{U_1}(K_{AB,1})$ and $E_{U_2}(K_{AB,2})$ anstelle von $E_U(K_{AB})$ enthält. Dann muß die Escrow Agency den Sitzungsschlüssel K anstelle des Unit Keys U bekanntgeben. Diese Lösung hat mehrere Nachteile. Zum Beispiel muß die

Escrow Agency bei jeder Entschlüsselung gefragt werden, was zu einem „Flaschenhals“ in der Kommunikation führen kann [1].

2. Eine weitere Lösung besteht darin, daß die Escrow Agents ein (ausforschungssicheres) Hardwaremodul zur Verfügung stellen, daß die Entschlüsselung der Nachricht durchführt, ohne dabei den Unit Key preiszugeben [1]. Daneben existieren weitere hardwarebasierte Lösungen [2], ihre Sicherheit basieren jedoch immer auf der Ausforschungssicherheit der Hardware, die heute nicht garantiert werden kann.

3. Es existieren mehrere softwarebasierte Ansätze [1, 2, 7]. In der Lösung von Walker und Balenson wird ein aktiver Abhörer eingesetzt. Die obige Anforderung wird jedoch nicht erfüllt, da nach einmaliger Freigabe des Abhörens alle weiteren sowie alte Kommunikationen abgehört und entschlüsselt werden können, da der Abhörer den Unit Key von der Escrow Agency erhält. Desmedt schlägt den Einsatz eines passiven Abhörers vor, der nur den verschlüsselten Verkehr auf dem Kanal mithört. Da die Entschlüsselung *jedes* Sitzungsschlüssels durch die Escrow Agents erfolgen muß, kann dieses ebenfalls zu einem Flaschenhals bei der Kommunikation mit den Escrow Agents führen. Die dritte softwarebasierte Lösung von Beth, Knobloch, Otten, Simmons und Wichmann verwendet ein modifiziertes Diffie-Hellman Schlüsselaustauschprotokoll [8, 9] sowie einen aktiven Abhörer [1]. Wir werden dieses Verfahren im folgenden Abschnitt detaillierter betrachten.

4. Unsere softwarebasierte Lösung basiert ebenfalls auf einem Protokoll zum authentischen Schlüsselaustausch und setzt einen aktiven Abhörer voraus. Es erlaubt die Entschlüsselung einer Nachricht in einem festdefinierten Zeitintervall I_t ohne aktive Teilnahme der Escrow Agents. Die Lösung wird detailliert in Abschnitt 6 beschrieben.

4. Das Escrowed Schlüsselaustauschprotokoll

Die Idee dieses Ansatzes liegt in der aktiven Teilnahme des Abhörers am Protokoll, damit er erfolgreich abhören kann. Da er den geheimen Schlüssel des Benutzers nicht erfährt, sondern stattdessen nur einen verwandten Schlüssel, kann er keine vorherigen Sitzungsschlüssel aus alten Kommunikationsdaten berechnen und damit keine frühere Kommunikation entschlüsseln.

Im folgenden beschreiben wir das Escrowed Schlüsselaustauschprotokoll aus [1, 2]. Die Nachrichtenverschlüsselung kann mit Hilfe eines beliebigen, geeigneten symmetrischen Kryptosystems, z.B. auch des Skipjack-Algorithmus, erfolgen.

Initialisierung

Das Trust Center wählt große Primzahlen p, q und q' mit $q = 2q' + 1, q|(p-1)$, einen Generator α einer multiplikativen Untergruppe der Ordnung q über $\mathbb{Z}_p$ und einen Generator β in $\mathbb{Z}_q$. Diese Parameter werden allen Benutzern authentisch bekannt gemacht.

Daten für Alice's Verschlüsselungseinheit

Die Benutzerin Alice wählt ihren geheimen Schlüssel $x_A \in_R \mathbb{Z}_q$ und berechnet den zugehörigen öffentlichen Schlüssel $y_A := \alpha^{x_A} \pmod{p}$. Außerdem wählt sie einen weiteren geheimen Parameter $t_A \in \mathbb{Z}_q$, der zum Abhören verwendet wird.

Daten für die Escrow Agency

Die Escrow Agency (Agents) besitzen die Werte $\beta^{t_A} \pmod{p}$ und x_A von der Benutzerin Alice.

Schlüsselaustausch ohne Abhören

Der Schlüsselaustausch besteht aus dem Austausch zweier Diffie-Hellman Halbschlüssel, die mit einer Operation „$\circ$“ (z.B. Addition, Multiplikation, XOR) verknüpft werden um den Sitzungsschlüssel zu erhalten. Das Protokoll wird in Abbildung 2 angegeben. Das

Benutzerin Alice		Netzwerk		Benutzer Bob
		$m_A, m_B \in_R \mathbb{Z}_q$		
		$z_B := \beta^{m_B} \pmod q$	$\longrightarrow$	$z_B \neq 0, \pm 1$
$z_A \neq 0, \pm 1$	$\longleftarrow$	$z_A := \beta^{m_A} \pmod q$		
$k_A := z_A^{t_A} \pmod q$				$k_B := z_B^{t_B} \pmod q$
$r_A := \alpha^{k_A} \pmod p$				$r_B := \alpha^{k_B} \pmod p$
r_B		$\longleftrightarrow$		r_A
$K_{1,A} := r_B^{x_A} \pmod p$				$K_{1,B} := y_A^{k_B} \pmod p$
$K_{2,A} := y_B^{k_A} \pmod p$				$K_{2,B} := r_A^{x_B} \pmod p$
$K_{A,B} := K_{1,A} \circ K_{2,A}$				$K_{A,B} := K_{1,B} \circ K_{2,B}$

Abbildung 2: Der Escrowed Schlüsselaustausch ohne Abhören

Verfahren ist korrekt, da die folgende Kongruenz gilt:

$$K_{A,B} := K_{1,A} \circ K_{2,A} = r_B^{x_A} \circ y_B^{k_A} \equiv \alpha^{k_B x_A} \circ \alpha^{k_A x_B} \equiv y_A^{k_B} \circ r_A^{x_B} = K_{1,B} \circ K_{2,B} =: K_{A,B} \pmod p.$$

Schlüsselaustausch mit Abhören

Im Falle des autorisierten Abhörens, erhält der Abhörer die Werte x_A, $\gamma := \beta^{u_A} \pmod p$, $u_A \in_R \mathbb{Z}_q$ und $\gamma^{t_A} \pmod p$ von den Escrow Agents. Er sendet $z_A := \gamma^{m_A} \pmod p$ an Alice, mit zufälliger Nachricht m_A. Alice und Bob folgen dann dem obigen Protokoll. Der Abhörer berechnet $k_A := (\gamma^{t_A})^{m_A} \pmod p$ und den Halbschlüssel $K_{2,A} := y_B^{k_A} \pmod p$. Da er Alice geheimen Schlüssel x_A kennt, kann er den Wert $K_{1,A} := r_B^{x_A} \pmod p$ und damit $K_{A,B}$ berechnen. Das resultierende Protokoll ist in Abbildung 3 gegeben.

5. Diskussion des Escrowed Schlüsselaustausches

Es existieren fünf mögliche Angreifer: Die Benutzer Alice und Bob, der Abhörer, das Netzwerk und ein Außenseiter. Die Sicherheit gegenüber einer Außenseiter- und Netzwerkattacke wurden in [1] beschrieben. Hier beschränken wir uns auf Attacken von Alice, Bob und dem Abhörer.

Erste Schwäche

Eine wichtige Betrugsmöglichkeit für *Alice* und *Bob* wurde bereits in den Sicherheisbetrachtungen in [1] genannt. Alice (oder Bob analog) kann den Parameter k_A zufällig wählen, anstelle dem Protokoll zu folgen. Nachdem sie z_A vom Netzwerk erhalten hat, wählt sie entweder k_A zufällig oder sie berechnet $k_A := z_A^{u_A} \pmod p$ mit einem zufällig gewähltem Wert $u_A \in_R \mathbb{Z}_q$. Danach folgt sie dem Protokoll. Da diese Lösung hardwareunabhängig ist, ist unklar, wie man diese Attacke verhindern kann. Die Schwäche durch die die Attacke ermöglicht wird, liegt darin, daß ein Sitzungsschlüssel zwischen Alice und Bob etabliert werden kann (sogar wenn Bob unwissend ist), ohne daß das Netzwerk Abweichungen vom Protokoll erkennen kann. Der Abhörer kann später den Betrug feststellen, wenn er entweder einen anderen Schlüssel $\tilde{r}_A$ aus seiner Berechnung erhält oder er keine Nachricht zwischen Alice und Bob lesen kann. Was kann er in diesem Fall noch machen ? Sofern er Alice sagt, daß sie aufhören soll zu betrügen, ist sie gewarnt und wird keine interessanten Nachrichten mehr über das Netzwerk schicken. Daher ist es erforderlich, daß

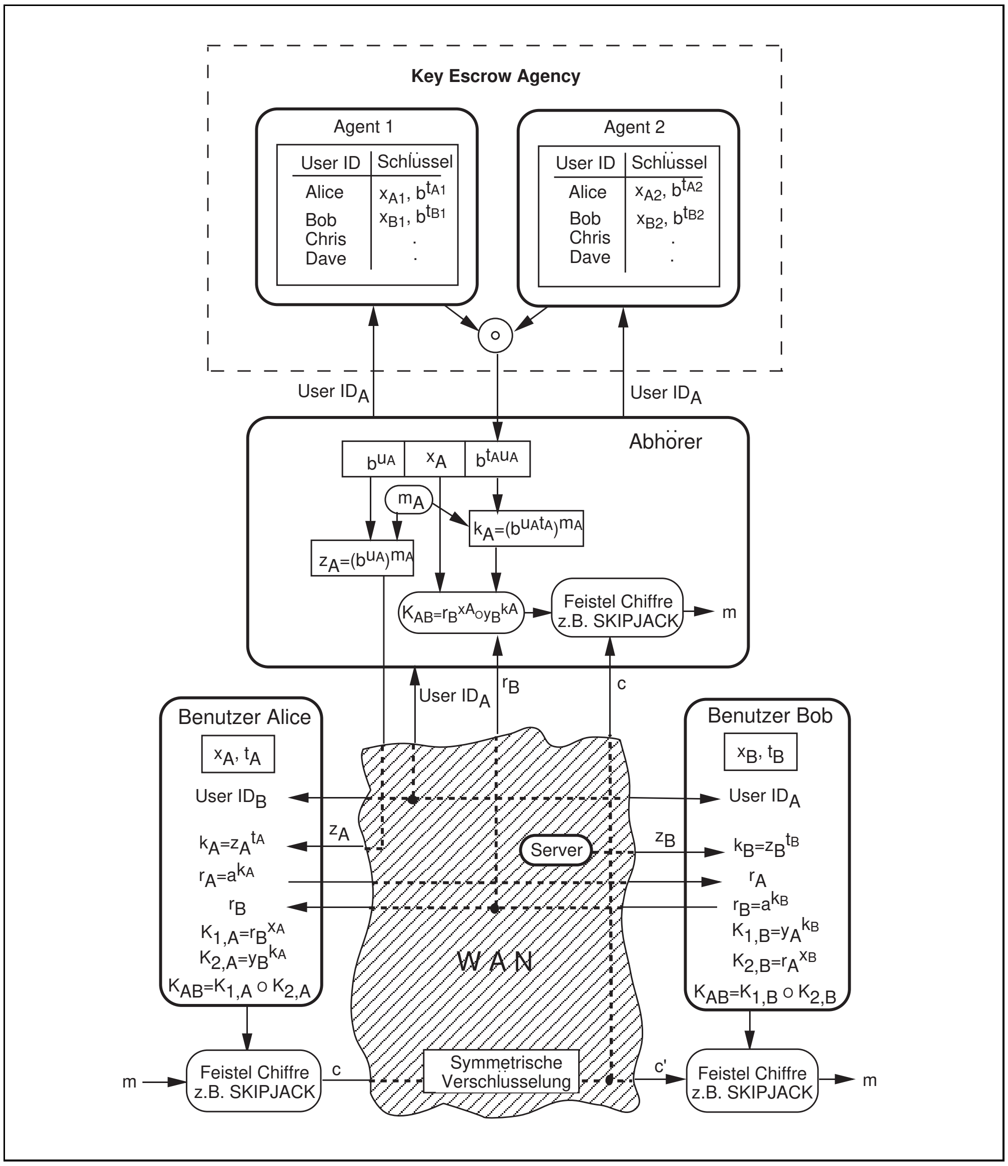

Abbildung 3: Der Escrowed Schlüsselaustausch mit Abhören

Abweichungen vom Protokoll bereits vom Netzwerk erkannt werden können. Eine hardwarebasierte Lösung hierfür wurde bereits in [1] vorgeschlagen, aber durch diese gehen die Vorteile einer softwarebasierten Lösung verloren.

Zweite Schwäche

Der *Abhörer* kann den Kanal abhören, obwohl er dazu nicht durch einen Gerichtsbeschluß befugt wurde. Nachdem er einmal die Parameter kennt, kann er *jeden* weiteren Sitzungsschlüssel berechnen. Die fünfte Anforderung in [1] besagt, daß die Autorisierung derart beschränkt sein muß, daß genau ein präzise definiertes Zeitintervall I_t abhörbar ist. Diese Anforderung erfüllt das vorgeschlagene Protokoll nicht, da nach einmaligem erfolgreichem Abhören jeder weitere Sitzungsschlüssel vom Abhörer ohne Zeitbegrenzung berechnet werden kann.

Dritte Schwäche

Die dritte Schwäche betrifft den Fall, indem der *Abhörer* einen Verdacht hat und daher eine Telefonleitung abhören möchte. Sobald er den Gerichtsbeschluß vorliegen hat, muß er zunächst noch auf die Daten der Escrow Agents warten bevor er mit dem Abhören beginnen kann. Diese Eigenschaft wurde von Beth et. al. [1] deswegen so gewählt, um unauthorisiertes Abhören alter Kommunikationen prinzipiell zu vermeiden. Jedoch sind in diesem Fall alle Informationen, die vor dem Erhalt dieser Daten ausgetauscht wurden, für den authorisierten Abhörer verloren, da er aktiv am Protokoll teilnehmen muß. Dieses ist dann besonders problematisch, wenn einer der Escrow Agents längere Zeit nicht erreichbar ist oder wegen Überlastung nicht alle Anfragen sofort bearbeiten kann. Daher sollte es möglich sein, auch ohne aktive Teilnahme am Protokoll, noch nachträglichen Zugriff auf frühere Kommunikationen zu ermöglichen, die seit dem Zeitpunkt des Gerichtsbeschlusses stattgefunden haben.

Im folgenden schlagen wir ein Protokoll vor, bei dem der Abhörer auch Kommunikationen, die vor dem Zeitpunkt des Erhalts der Schlüsseldaten von den Escrow Agents lagen abhören kann, sofern er dafür einen Gerichtsbeschluß vorlegen kann.

6. Das neue Escrowed Schlüsselaustauschprotokoll

Jetzt stellen wir unsere Lösung vor, die alle drei gerade diskutierten Schwächen vermeidet. Die Idee ist, daß die zufälligen Parameter k_A und r_A, die im obigen Protokoll nicht vom Netzwerk verifiziert werden konnten, in einem ElGamal ähnlichen Signaturverfahren benutzt werden [10]. Dadurch existiert ein Zusammenhang und sie können nicht zufällig durch die Benutzer ersetzt werden, ohne daß das Netzwerk dieses merkt. Dabei machen wir drei *grundsätzliche Annahmen:*

1. Erstens sollte im Netzwerk ein vertrauenswürdiger Server (bzw. mehrere) existieren, der die Signatur gewissenhaft überprüft und nur dann die Kommunikation weiterleitet, wenn die Signaturgleichung erfüllt ist. Dies bedeutet, daß er nicht mit einem der Benutzer kollaborieren darf und vor Hackern (die die Überprüfung beeinflussen können) geschützt ist.

2. Zweitens sollte kein zusätzlicher Kanal zwischen zwei Benutzern existieren, über den diese ohne den vertrauenswürdigen Server kommunizieren können. Das heißt, Protokollabweichungen sollten auf jeden Fall von mindestens einem vertrauenswürdigen Server unterbunden werden können.

3. Die dritte Annahme besagt, daß die beiden Server nicht bereits über ein gemeinsames Geheimnis verfügen. Diese Annahme ist essentiell für die Errichtung unterschwelliger Kanäle und die Verwendung von Vorverschlüsselung. Beides ist problemlos möglich, wenn sich beide Parteien bereits im voraus auf ein Geheimnis einigen konnten.

Dafür bracht die Software des Benutzers nicht gegen Modifikation geschüzt zu sein, da jegliche Manipulation dem Netzwerk bei seiner Überprüfung auffallen wird.

Um das Abhören für ein festes Zeitintervall $I_t = [t \cdot temp : (t+1) \cdot temp]$ zu erreichen, besitzt Alice verschiedene geheime und öffentliche Schlüssel für jedes Zeitintervall. Dieses hat den Vorteil, daß der Abhörer nach Verstreichen der Zeitdauer $temp$ den neuen geheimen Schlüssel nicht mehr kennt und daher den Sitzungsschlüssel nicht mehr ohne eine erneute Erlaubnis der Escrow Agency berechnen kann.

Das Protokoll kann in zwei Schritte unterteilt werden: Der erste beschreibt die Erzeugung unterschiedlicher zertifizierter Schlüsselpaare für jedes Zeitintervall und der zweite den authentischen Schlüsselaustausch unter Verwendung dieser zertifizierten Schlüsselpaare.

6.1 Erzeugung der zertifizierten Schlüsselpaare

Angenommen der betrachtete Gesamtzeitraum läßt sich in n Zeitintervalle mit Dauer $temp$ unterteilen. Wenn z.B. jeder Benutzer sein (Basis-)Schlüsselpaar einmal im Jahr erneuert und jedes Zeitintervall eine Woche lang ist, existieren insgesamt $n = 52$ Intervalle. Daher benötigt jeder Benutzer 52 zertifizierte Schlüsselpaare.

Eine einfache Lösung das Schlüsselpaar $(x_{A,i}, y_{A,i})$, $1 \leq i \leq n$, mit $y_{A,i} := \alpha^{x_{A,i}} \pmod p$ für die Benutzerin Alice zu generieren ist, daß jede Escrow Agency j, $1 \leq j \leq l$, einen Share $x_{A,i,j}$ des geheimen Schlüssels $x_{A,i}$ von Alice erhält, so daß die Summe aller dieser Shares ihren geheimen Schlüssel ergibt, d.h. $x_{A,i} := \sum_{j=1}^{l} x_{A,i,j} \pmod q$. Dann kann jede Escrow Agent den zu ihrem Share gehörigen Teil des öffentlichen Schlüssels als $y_{A,i,j} := \alpha^{x_{A,i,j}} \pmod p$ berechnen und das Produkt $y_{A,i} := \prod_{j=1}^{l} y_{A,i,j} \pmod p$ inklusive einem Zeitstempel zertifizieren. Dieses Zertifikat kann von jedem verifiziert werden und deutet an, daß die Escrow Agency den geheimen Schlüssel zum Abhören kennt.

Um den Speicheraufwand für die Escrow Agency zu reduzieren, wird vorgeschlagen, daß der Share $x_{A,i,j}$ vom geheimen Schlüssel $x_{A,i}$ berechnet wird als $x_{A,i,j} := E_{SK_{A,j}}(x_{A,i-1,j})$, wobei ein geheimer Masterschlüssel $SK_{A,j}$, der dem Benutzer und dem Escrow Agent j bekannt ist, sowie ein Share $x_{A,0,j}$ verwendet wird. Ferner wird ein symmetrisches Kryptosystem mit Verschlüsselungsfunktion E eingesetzt. Dadurch braucht der Escrow Agent j nur die beiden Parameter $x_{A,0,j}$ und $SK_{A,j}$ für Alice zu speichern. Da jedes $SK_{A,j}$ niemals an das Netzwerk oder den Abhörer verraten wird, gibt es keine Bedenken, daß der Abhörer den geheimen Schlüssel für das Zeitintervall I_t berechnen kann, sofern er irgendeinen geheimen Schlüssel eines anderen Zeitintervalls kennt. Der Nachteil dieser Lösung besteht darin, daß das Netzwerk und die anderen Benutzer die zugehörigen öffentlichen Schlüssel $y_{A,i}$, $1 \leq i \leq n$ nicht selbst mit einem ähnlichen Ansatz unter Verwendung eines zertifizierten öffentlichen Schlüssels $y_{A,0}$ berechnen können. Daher müssen ihnen die zertifizierten öffentlichen Schlüssel der Benutzer von der Escrow Agency oder den Benutzern zur Verfügung gestellt werden.

Es wäre wünschenswert, daß das Netzwerk und die anderen Benutzer selbst die öffentlichen Schlüssel $y_{A,i}, 1 \leq i \leq n$, unter Verwendung einer öffentlichen Funktion F_A berechnen könnten, wobei der zertifizierte Schlüssel $y_{A,0}$ als Argument diente. Die Existenz einer solchen Funktion ist leider ein ungelöstes Problem und Gegenstand aktueller Forschung.

6.2 Das Escrowed Schlüsselaustauschprotokoll

Initialisierung

Das Trust Center wählt große Primzahlen p, q und q mit $q|(p-1)$, einen Generator α einer multiplikativen Untergruppe der Ordnung q über $\mathbb{Z}_p$ und eine kollisionsfreie Hashfunktion h. Diese Werte werden authentisch unter allen Benutzern verbreitet.

Daten für Alice's Verschlüsselungseinheit

Die Benutzerin Alice wählt n geheime Schlüssel $x_{A,i} \in_R \mathbb{Z}_q$ gemäß dem vorherigen Abschnitt und berechnet die zugehörigen öffentlichen Schlüssel $y_{A,i} := \alpha^{x_{A,i}} \pmod p$. Diese werden von der Escrow Agency zusammen mit einem Zeitstempel, der ihr Gültigkeitsintervall I_t angibt, zertifiziert.

Daten für die Escrow Agency

Die Escrow Agency (Agents) besitzt alle n geheimen Schlüssel $x_{A,i}$ der Benutzerin Alice bzw. kann sich diese wie oben beschrieben erzeugen.

Schlüsselaustausch ohne Abhören

Während des Zeitintervalls I_t, in dem Alice ihr Schlüsselpaar $(x_{A,t}, y_{A,t})$ und Bob sein Schlüsselpaar $(x_{B,t}, y_{B,t})$ verwendet, wird das Protokoll in Abbildung 4 ausgeführt, wobei die Verknüpfung $\oplus$ das bitweise XOR bezeichnet.

Benutzerin Alice		Netzwerk		Benutzer Bob
		$y_{A,t}, y_{B,t}$ authentisch		
m_A	$\longleftarrow$	$m_A, m_B \in_R \mathbb{Z}_q$	$\longrightarrow$	m_B
$k_A \in_R \mathbb{Z}_q$				$k_B \in_R \mathbb{Z}_q$
$r_A := \alpha^{k_A} \pmod p$				$r_B := \alpha^{k_B} \pmod p$
$s_A := x_{A,t}(r_A \oplus m_A) + k_A$	$\longrightarrow$	r_A, s_A		
		$\alpha^{s_A} \equiv y_{A,t}^{r_A \oplus m_A} r_A$?	$\longrightarrow$	r_A
		r_B, s_B	$\longleftarrow$	$s_B := x_{B,t}(r_B \oplus m_B) + k_B$
r_B	$\longleftarrow$	$\alpha^{s_B} \equiv y_{B,t}^{r_B \oplus m_B} r_B$?		
$K_{1,A} := r_B^{x_{A,t}} \pmod p$				$K_{1,B} := y_{A,t}^{k_B} \pmod p$
$K_{2,A} := y_{B,t}^{k_A} \pmod p$				$K_{2,B} := r_A^{x_{B,t}} \pmod p$
$K_{A,B} := h(K_{1,A} \circ K_{2,A})$				$K_{A,B} := h(K_{1,B} \circ K_{2,B})$

Abbildung 4: Der neue Escrowed Schlüsselaustausch ohne Abhören

Das Protokoll ist korrekt, da die folgende Kongruenz gilt:

$$\begin{aligned} K_{A,B} &:= h(K_{1,A} \circ K_{2,A}) = h(r_B^{x_{A,t}} \circ y_{B,t}^{k_A}) \equiv h(\alpha^{k_B x_{A,t}} \circ \alpha^{k_A x_{B,t}}) \\ &\equiv h(y_{A,t}^{k_B} \circ r_A^{x_{B,t}}) = h(K_{1,B} \circ K_{2,B}) =: K_{A,B} \pmod p. \end{aligned}$$

Die Funktion h wird als kollisionsfreie Hashfunktion gewählt. Dieses ist erforderlich, da das zugrundeliegende Schlüsselaustauschprotokoll Dreiecksattacken nicht standhält, sofern als Sitzungsschlüssel $K_{A,B}$ verwendet wird [5]. Ein unehrlicher Benutzer Chris kann einen Sitzungsschlüssel aus einer früheren Kommunikation zwischen Alice und Bob berechnen, sofern er Alice und Bob glaubhaft machen kann, daß er mit jedem von ihnen einen Sitzungsschlüssel austauschen möchte. Aus der Kenntnis dieser Protokolle (bei denen er sich unehrlich verhält) *und* zusätzlicher Information, die er freiwillig von Alice und Bob erhalten hat, kann er den alten Sitzungsschlüssel zwischen den beiden berechnen. Diese Attacke ist eher von theoretischem Interesse, da weder Alice noch Bob ihm diese freiwilligen zusätzlichen Informationen geben wird. Sie kann jedoch durch den Einsatz der Hashfunktion h prinzipiell vermieden werden [5].

Schlüsselaustausch mit Abhören

Jetzt betrachten wir den Fall, indem ein autorisierter Abhörer am Protokoll während des Zeitintervalls I_t teilnimmt. Er erhält den zeitbehafteten geheimen Schlüssel $x_{A,t}$ von der Escrow Agency, nachdem er seinen Gerichtsbeschluß vorgelegt hat. Unter Verwendung dieses Schlüssels berechnet er $K_{1,A} := r_B^{x_{A,t}}$ und den geheimen Wert k_A als:

$$k_A := s_A - x_{A,t} \cdot (r_A \oplus m_A) \pmod q.$$

Mit diesem Wert erhält er den Sitzungsschlüssel als

$$K_{A,B} := h(K_{1,A} \circ y_B^{k_A} \pmod p))$$

den er zum Entschlüsseln des Chiffrats c benötigt. Das Protokoll wird in Abbildung 5 veranschaulicht.

6.3 Sicherheitsanalyse

Wir zeigen, wie die drei Schwächen aus Abschnitt 5 in unserem Protokoll vermieden werden.

1. **Erste Schwäche:** Da das Netzwerk den öffentlichen Schlüssel $y_{A,t}$ für jedes Zeitintervall I_t kennt, kann Alice den Wert $x_{A,t}$ nicht durch irgendeinen zufälligen Wert ersetzen, den die Escrow Agency nicht berechnen kann. Daher kann der Abhörer immer den Wert k_A berechnen, da er das korrekte $x_{A,t}$ kennt.

   Alice kann nicht k_A durch einen zufälligen anderen Wert ersetzen, da k_A und r_A für die Signaturverifikation durch das Netzwerk benötigt werden. Das Netzwerk leitet den Wert r_A nur dann weiter, wenn die Verifikationsgleichung erfüllt war. Daher muß r_A korrekt berechnet worden sein.

2. **Zweite Schwäche:** Da es verschiedene (für dem Abhörer unzusammenhängende) Schlüsselpaare $(x_{A,t}, y_{A,t})$ für jedes Zeitintervall I_t gibt, kann der Abhörer nur die Nachrichten, die in diesem Zeitintervall ausgetauscht wurden, entschlüsseln.

3. **Dritte Schwäche:** Da alle Berechnungen auch unter Verwendung alter Protokolldaten $\{ID_A, (m_A, r_A, s_A), c\}$, die gegebenenfalls vom Netzwerk aufgezeichnet worden sein können, durchgeführt werden können, kann der Abhörer genau dann alte Chiffrate abhören, wenn er die dazupassenden alten geheimen Schlüssel von der Escrow Agency erhält.

4. **Attacken durch das Netzwerk:** Um zu vermeiden, daß das Netzwerk die Nachrichten m_A und m_B adaptiv wählen kann, um Informationen über die geheimen Schlüssel $x_{A,t}$ oder $x_{B,t}$ zu erhalten, kann man das Protokoll folgendermaßen modifizieren: Der Benutzer kann entweder eine zweite zufällige Nachricht $\tilde{m}_A$, wählen, $h(m_A, \tilde{m}_A)$ unterschreiben und $\tilde{m}_A$ zusammen mit der Signatur übertragen oder er kann zu Beginn einen Diffie-Hellman Schlüsselaustausch mit dem Netzwerk durchführen und den so ausgetauschten Schlüssel als Nachricht verwenden. Die zweite Lösung schützt ihn außerdem vor dem Abhören der signierten Nachricht durch externe Angreifer.

In einer vereinfachten Version des Protokolls könnten die Parameter (k_A, r_A) als zertifizierte Schlüsselpaare direkt verwendet werden. Dadurch entfällt die Signaturerzeugung und Verifikation. Der Berechungsaufwand für das Netzwerk reduziert sich. Der Nachteil dieser Modifikation liegt darin, daß zwei Parteien entweder den gleichen Sitzungsschlüssel für ein ganzes Zeitintervall verwenden müssen oder sie neue zertifizierte Schlüsselpaare (k_A, r_A) für jede Sitzung benötigen, während sie beliebig viele verschiedene in unserem Originalprotokoll für ein Schlüsselpaar $(x_{A,t}, y_{A,t})$ während eines Zeitintervalls I_t haben können. Daher reduziert sich entweder die Sicherheit oder der Berechnungsaufwand erhöht sich. Wir empfehlen diese Modifikationen nicht, sondern schlagen vor, besser die Dauer des Zeitintervall geeignet zu wählen, um die optimale Lösung hinsichtlich Sicherheit und Aufwand zu finden.

7. Ausblick

Wir haben den Escrow Encryption Standard und das Escrowed Schlüsselaustauschprotokoll mit aktivem Abhörer betrachtet. Dabei haben wir die Hauptschwächen der beiden Ansätze aufgezeigt und einen Überblick über mögliche Lösungen gegeben, die allerdings nicht immer der von uns gestellten Anforderung genügten. Schließlich haben wir ein neues softwarebasiertes Escrowed Schlüsselaustauschprotokoll mit aktivem Abhörer vorgestellt, das alle zuvor beschriebenen kryptographischen Schwächen vermeidet. Unser Ansatz ist

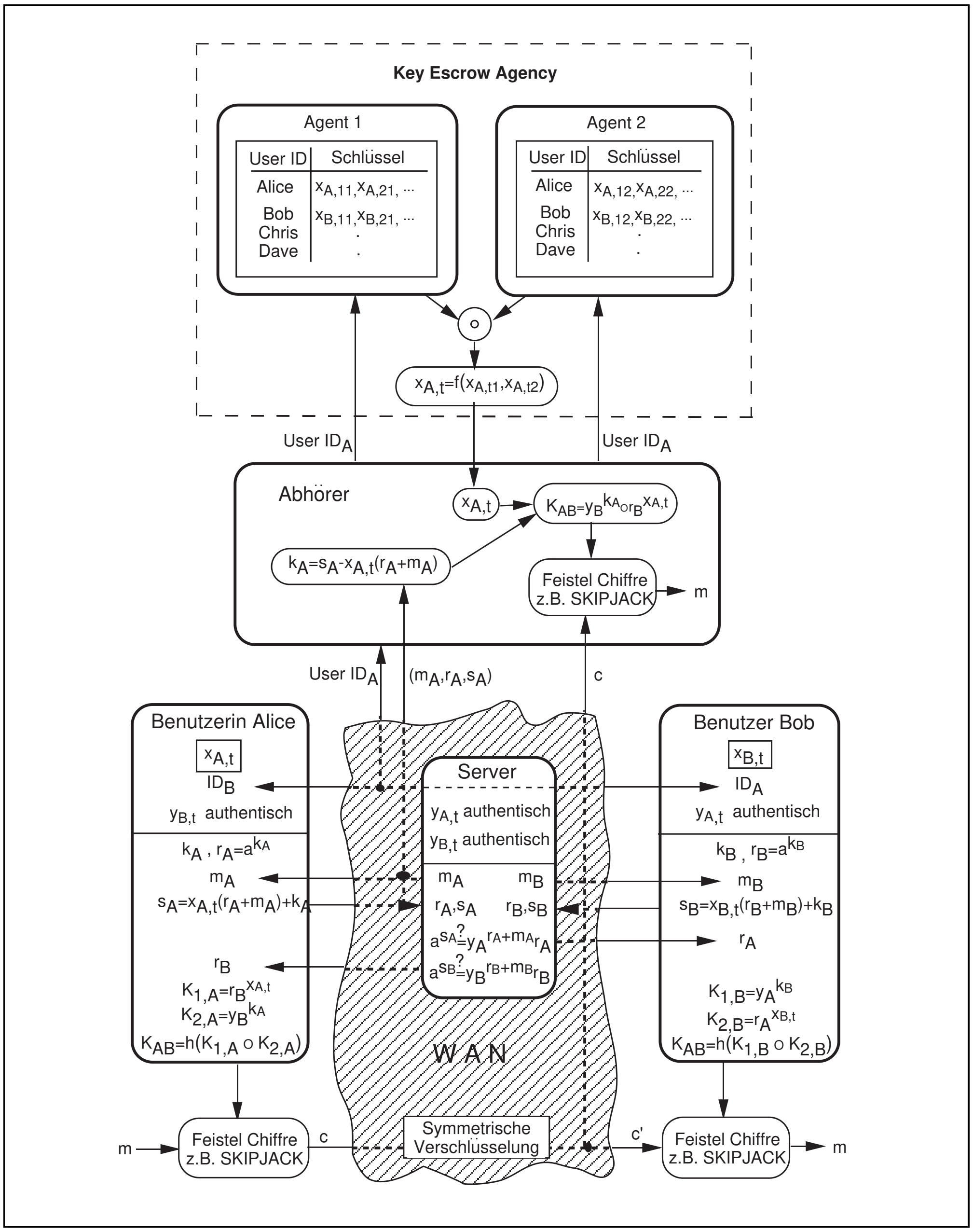

Abbildung 5: Der neue Escrowed Schlüsselaustausch mit Abhören

ein Schritt zum Design von Schlüssel-Escrow-Protokollen, die nur die für die Escrow Agency absolut erforderlichen minimalen Bedürfnisse erfüllen, und kein zusätzliches Abhören ermöglichen.

Die Grenzen der softwarebasierten Escrowed Schlüsselaustauschprotokolle liegen in den drei zuvor gemachten Annahmen, daß zwei Parteien nicht bereits über ein gemeinsames Geheimnis verfügen, mit dessen Hilfe sie vorverschlüsseln können und daß das Netzwerk die Signaturen ehrlich verifiziert und entsprechend keine Daten weiterleitet, wenn diese nicht erfüllt sind. Daneben kann nicht vermieden werden, daß Kriminelle andere Kanäle zum Austausch eines Sitzungsschlüssels verwenden. Die letzten beiden Annahmen lassen sich durch einen hybriden Ansatz, bei dem jeder Benutzer über ein kryptographisches Modul in seinem Benutzerendgerät verfügt, vermeiden, wenn dieses bereits die Verifikation der Unterschrift durchführt. Diesem muß der Benutzer nicht das gleiche Vertrauen wie z.B. dem Clipper-Chip entgegenbringen, da es keine geheimen Algorithmen enthält. Um die Funktionalität zu gewährleisten, muß dieses Modul jedoch ausforschungssicher sein, so daß der Benutzer die Verifikation nicht umgehen kann und der geheime Benutzer-schlüssel nicht kompromitiert wird. Werden zusätzlich stichprobenartig die Unterschriften von vertrauenswürdigen Servern verifiziert, so kann ein Betrugsversuch bei Verletzung der Ausforschungssicherheit erkannt werden, ohne daß diese Überprüfung unbedingt vorhanden sein muß.

Um die Privatsphäre des Benutzers zu schützen, falls Shares seines geheimen Schlüssels durch unsachgemäßen Gebrauch oder Speicherung bekannt werden, kann es von Vorteil sein, nur einen Teil des geheimen Schlüssels unter den Escrow Agents aufzuteilen und einen kleinen Teil (z.B. 30–40 bit) ganz für sich zu behalten. Im Falle eines autorisierten Abhörens kann dieser geheime Teil durch vollständige Suche vom Abhörer berechnet werden. Dadurch ist der Benutzer zumindestens durch ein schwaches Verschlüsselungssystem geschützt, selbst wenn alle Shares seines geheimen Schlüssels bekannt werden sollten.

8. Danksagung

Die Autoren möchten den Teilnehmern der Podiumsdiskussion über den Kryptokonflikt auf der VIS'95 in Rostock für ihre zahlreichen Diskussionsbeiträge danken. Daneben sind die Autoren einem anonymen Reviewer, dessen Name nicht escrowed werden soll, für hilfreiche Diskussionen und zahlreiche Hinweise verpflichtet.

Literatur

[1] T.Beth, H.-J.Knobloch, M.Otten, G.Simmons, P.Wichmann, „Towards Acceptable Key Escrow Systems“, Proc. 2nd ACM Conference on Computer and Communications Security, Fairfax, Nov. 2–4, (1994), S. 51–58.

[2] T.Beth, M.Otten, (ed.) „E.I.S.S.–Workshop on Escrowed Key Cryptography“, E.I.S.S.-Report 94/7, University of Karlsruhe, June 22 – 24, (1994), 160 Seiten.

[3] M.Blaze, „Protocol Failure in the Escrowed Encryption Standard“, Proc. 2nd ACM Conference on Computer and Communications Security, (1994), S. 59–67.

[4] E.F.Brickell, D.E.Denning, S.T.Kent, D.P.Mahler, W.Tuchman, „SKIPJACK Review“, Interim Report, July 28, (1993), 8 pages.

[5] M.Burmester, „On the Risk of Opening Distributed Keys“, Lecture Notes in Computer Science , Advances in Cryptology: Proc. Crypto '94, Berlin: Springer Verlag, (1994), S. 308–317.

[6] D.Denning, „The Clipper Chip: A Technical Summary“, April 21, (1994), 3 Seiten.

[7] Y.Desmedt, „Securing Traceability of Ciphertexts – Towards a Secure Software Key Escrow System“, Lecture Notes in Computer Science 921, Advances in Cryptology: Proc. Eurocrypt '95, , Berlin: Springer Verlag, (1995), S. 147 – 157.

[8] W. Diffie, M. Hellman, „New directions in cryptography“, IEEE Transactions on Information Theory, Vol. IT-22, No. 6, November, (1976), S. 644–654.

[9] C.G.Günther, „An identity based key exchange protocol“, Lecture Notes in Computer Science 434, Advances in Cryptology: Proc. Eurocrypt '89, Berlin: Springer Verlag, (1990), S. 29–37.

[10] P.Horster, M.Michels, H.Petersen, „Meta-ElGamal signature schemes“, Proc. 2. ACM conference on Computer and Communications security, Fairfax, Virginia, 2.-4. Nov., (1994), S. 96 – 107[1].

[11] P.Horster, M.Michels, H.Petersen, „A new key escrow system with active investigator“, Proc. Securicom, Paris, La Defense, 8.-9. Juni, (1995), S. 15–28[1].

[12] S.Landau, S.Kent, C.Brooks, S.Charney, D.Denning, W.Diffie, A.Lauck, D.Miller, P.Neumann, D.Sobel, „Codes, Keys and Conflicts: Issues in U.S. Crypto Policy“, Report of a Special Panel of the ACM U.S. Public Policy Committee (USACM), June, (1994), 74 Seiten.

[13] J.Markoff, „Communications plan to balance government access with privacy“, New York Times, April 16, (1993).

[14] S.Möller, A.Pfitzmann, I.Stierand, „Rechnergestützte Steganographie: Wie sie funktioniert und warum folglich jede Reglementierung von Verschlüsselung unsinnig ist“, Datenschutz und Datensicherheit, 18/6, (1994), S. 318–326.

[15] National Institute for Standards and Technology, „Escrowed Encryption Standard“, Federal Information Processing Standards Publication 185, U.S. Dept. of Commerce, February 9, (1994).

[16] National Institute for Standards and Technology, „Technical Fact Sheet on Blaze Report and Key Escrow Encryption“, June 15, (1994), 2 Seiten.

[17] G.Orwell, „1984“, (1948).

[18] J.Reno, “Key Escrow Encryption Announcements”, Department of Justice, U.S. Government, February 4, (1994).

[19] R.A.Rueppel, „Clipper – Der Krypto-Konflikt am Beispiel der Amerikanischen ESCROW Technologie“, Datenschutz und Datensicherheit, 18/8, (1994), S. 443–451.

[1] Verfügbar über anonymous FTP im Verzeichnis *ftp.tu-chemnitz.de/pub/Local/informatik/crypto* oder über WWW unter URL <http://www.tu-chemnitz.de/~hpe/techrep.html>