In: Horster, P. (Hrsg.): *Trust Center*. Proceedings zur Arbeitskonferenz Trust Center 95, Vieweg-Verlag, S. 232-245.

Zeitabhängiges Key Escrowing

Dirk Fox

Universität Siegen
Institut für Nachrichtenübermittlung
Postfach 10 12 40, D-57068 Siegen
fox@nue.et-inf.uni-siegen.de

Zusammenfassung

Seit zwei Jahren wird in den USA heftig um ein inzwischen standardisiertes Verschlüsselungssystem für Telephonieanwendungen mit treuhänderischer Schlüsselhinterlegung (*Escrowed Encryption Standard*, EES) gestritten. Ein Teil der Kritik richtet sich dabei gegen eine Eigenschaft des Verfahrens, die den meisten heute diskutierten *Key Escrow*-Systemen anhaftet: die Hinterlegung eines „Generalschlüssels“, dessen Freigabe bei Vorliegen einer Abhörgenehmigung prinzipiell auch die Entschlüsselung älterer und zukünftiger Kommunikationsdaten ermöglicht.
Die Verwendung zeitabhängiger Schlüssel beseitigt diesen Nachteil herkömmlicher *Key Escrow*-Systeme. Exemplarisch wird der amerikanische EES vorgestellt; anschließend werden unterschiedliche Verfahren für die Realisierung eines zeitabhängigen (*Time Dependent*) *Key Escrow Systems* vorgeschlagen und hinsichtlich ihres Aufwandes sowie ihrer prinzipiellen Vor- und Nachteile diskutiert.

1 Einführung

Mit der zunehmenden Verbreitung elektronischer Kommunikation gewinnt der alte Konflikt zwischen berechtigten Sicherheitsbedürfnissen von Bürgern und Industrie auf der einen und den Forderungen von Strafverfolgungsbehörden auf der anderen Seite erheblich an Bedeutung. So sollen persönliche, sensible und wertvolle Daten während ihrer Übertragung sowohl vor unberechtigter Kenntnisnahme und Verfälschung geschützt als auch Ermittlungsbehörden bei Vorliegen einer gerichtlichen Genehmigung zugänglich sein. Der wachsende Einsatz ehemaliger Geheimdienste für Industriespionage und die weltweite Zusammenarbeit des organisierten Verbrechens verleihen dem Konflikt eine besondere Aktualität und Brisanz.

Für die Lösung dieses Dilemmas gibt es zunächst einmal zwei sehr einfache Lösungen: Entweder das generelle Verbot der Verwendung kryptographischer Verfahren zur Verschlüsselung von Daten, zumindest die Einführung einer Genehmigungspflicht. Ein solches Verbot läßt sich jedoch praktisch nicht kontrollieren; daher verfolgt bisher außer Frankreich und Italien keine Industrienation diesen restriktiven Ansatz.

Oder aber es werden ausschließlich schwache Kryptoverfahren bzw. solche standardisiert und zugelassen, für die die Ermittlungsbehörden eine *trapdoor* kennen, d.h. eine Möglichkeit, die

Daten auch ohne Kenntnis des Schlüssels zurückzugewinnen. Letzteres wurde immer wieder bei geheimgehaltenen, unter der Beteiligung von Sicherheitsbehörden entwickelten Algorithmen wie dem A5 des GSM, dem „Behördenchip“ des Bundesamtes für Sicherheit in der Informationstechnik (BSI) und dem von der *National Security Agency* (NSA) der USA entwikkelten, 1977 genormten *Data Encryption Standard* (DES) vermutet [NBS_77, ANSI_81, FuRi_94]. Nachteile dieser ‘Lösung’: Schwache Algorithmen sind ein schlechter Schutz vor Spionage. Sie werden zudem meist nur in Hardware freigegeben, um den Algorithmus geheimzuhalten; die Integration in Kommunikationssysteme ist aufwendig und teuer. Daher sind Akzeptanz und Vertrauen der Benutzer in solche Algorithmen oft gering; sie lassen sich häufig nur mit Ausschließlichkeitsregelungen etablieren.

2 Der *Escrowed Encryption Standard* (EES)

Einen dritten Weg schlug in den USA die Clinton-Administration ein. Mit der Ankündigung eines standardisierten *Key Escrow Systems* (KES) – einem unter Bush von der NSA entwikkelten Verfahren zur treuhänderischen Verwaltung von Schlüsseln – löste sie am 16. April 1993 eine heftige Diskussion im Internet und unter Experten aus. Der Vorschlag sah vor, Telefone mit einem Verschlüsselungschip (Projektname „Clipper”) auszustatten, der auf Knopfdruck für eine *online*-Verschlüsselung von Gesprächen, Faxen und (analoger, modembasierter) Datenübertragung sorgt. Trotz erheblicher Kritik wurde das Verfahren am 9. Februar 1994 ohne wesentliche Änderungen als nationaler *Escrowed Encryption Standard* genormt (EES) [NIST1_94].

2.1 Der SKIPJACK-Algorithmus

Der EES verwendet einen klassifizierten, d.h. geheimen, von der NSA in 10-jähriger Arbeit entwickelten symmetrischen Verschlüsselungsalgorithmus namens SKIPJACK. Diese Blockchiffre arbeitet intern nach dem Feistel-Prinzip, das auch dem DES zugrundeliegt [FuRi_94]. Im Unterschied zum DES verwendet SKIPJACK jedoch einen 80 bit langen Schlüssel und arbeitet intern mit 32 Runden (gegenüber 56 bit Schlüssellänge und 16 Runden beim DES). Die Blocklänge von 64 bit hält ihn kompatibel mit dem DES und den für Blockchiffren genormten Betriebsarten *Electronic Code Book* (ECB), *Cipher Block Chaining* (CBC), *Cipher Feedback* (CFB) und *Output Feedback* (OFB) [NBS_80, DaPr_89]; dies soll die Integration des SKIPJACK in existierende Anwendungen erleichtern. Näheres ist über den Algorithmus öffentlich nicht bekannt.

Fünf ausgewählten amerikanischen Experten wurde SKIPJACK für einen Monat zur Begutachtung vorgelegt. Diese geben in ihrem am 28. Juli 1993 veröffentlichten Gutachten an, daß sie keine Schwäche des Algorithmus feststellen konnten [BDKMT_93]. Die große Zahl interner Runden und der um den Faktor 2^{24} größere Schlüsselraum des SKIPJACK gegenüber dem DES macht nach ihrer Ansicht eine schnelle Kryptoanalyse in den nächsten 30-40 Jahren unwahrscheinlich. SKIPJACK soll zudem gegen differentielle Kryptoanalyse resistent sein.

2.2 Der Behördenzugriffsmechanismus

Die SKIPJACK-Verschlüsselung der Daten erfolgt nach dem EES mit einem *Session Key* KS, der zuvor *offline* oder mit einem Schlüsselaustauschprotokoll beim Verbindungsaufbau zwischen den Kommunikationspartnern vereinbart wird (siehe Bild 1).

Die Möglichkeit eines Behördenzugriffs garantiert ein spezieller KES-Mechanismus, der an die übertragenen Daten ein *Law Enforcement Access Field* (LEAF) anhängt. Dieses Feld wird nach der *LEAF Creation Method* (LCM) erzeugt: Es enthält den mit einem Chip-Schlüssel KU (*Unit Key*) verschlüsselten *Session Key* KS, eine eindeutige Chip-Kennung UID und einen *Escrow Authentikator* EA, der nach einem geheimen Verfahren u.a. aus dem Initialisierungsvektor IV und KS berechnet wird. Alle drei Einträge werden mit einem systemweit gültigen *Family Key* KF verschlüsselt (siehe Bild 2).

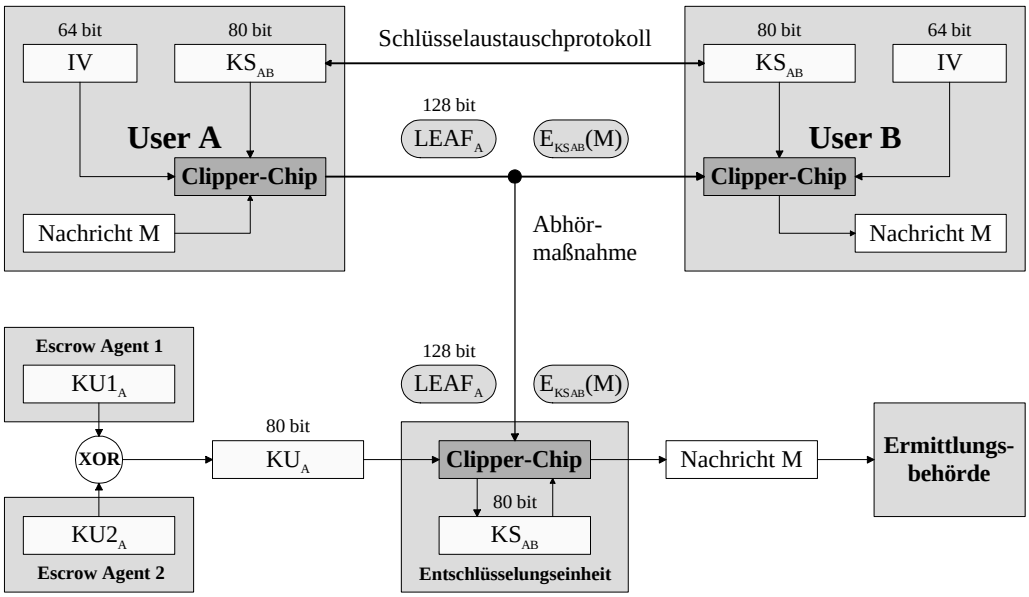

Bild 1: Arbeitsweise des Escrowed Encryption Standard (EES).[1]

Der Chip-Schlüssel KU wird nach der Personalisierung des „Clipper"-Chips in zwei Teile KU1 und KU2 zerlegt, die XOR-verknüpft wieder KU ergeben. Diese beiden Schlüsselteile werden verschlüsselt und treuhänderisch bei je einem *Escrow Agent* hinterlegt.[2] Bei Vorliegen einer gerichtlichen Anordnung händigen diese ihre Schlüsselteile der Ermittlungsbehörde zur Entschlüsselung einer abgehörten EES-Nachricht aus (siehe Bild 1).

---

1 $E_x(y)$: SKIPJACK-Verschlüsselung von y mit Schlüssel x.

2 In den USA wurden entgegen ersten Ankündigungen zwei Bundesbehörden als *Escrow Agents* eingesetzt: das NIST (*Department of Commerce*) und die *Automated Systems Division* (*Department of the Tresure*) [Froo_95].

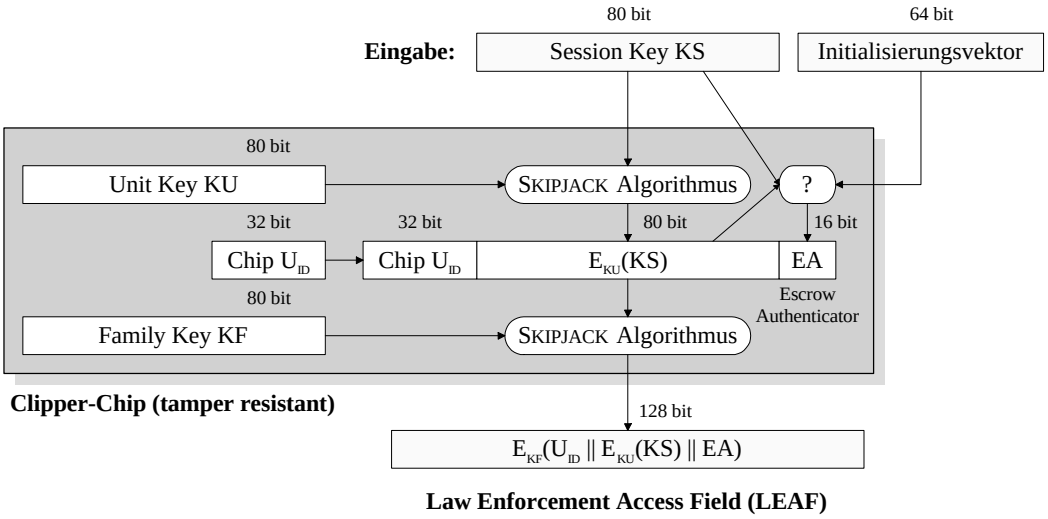

Bild 2: LEAF Creation Method (LCM): Erzeugung des Law Enforcement Access Field.

Diese Möglichkeit des nachträglichen Behördenzugriffs auf die Daten bietet ein EES-System allerdings nur dann, wenn erstens die Prüfung des LEAF und die Entschlüsselung der Daten nicht voneinander unabhängig durchgeführt werden können und zweitens die Fälschung eines LEAF praktisch unmöglich ist. Aus diesem Grund wird der SKIPJACK-Algorithmus geheimgehalten: Wäre er bekannt, ließe sich leicht eine Entschlüsselung programmieren, die empfangene Daten auch ohne bzw. mit ungültigem LEAF entschlüsselt.

SKIPJACK- und LCM-Algorithmus sind daher ausschließlich in *tamper resistant* Hardware erhältlich. Der „Clipper"-Chip umfaßt physikalisch geschützten, nicht auslesbaren, einmalprogrammierbaren Speicher für die bei der Personalisierung einzubrennende Seriennummer UID, den persönlichen Benutzerschlüssel KU und den allen EES-Einheiten gemeinsamen Systemschlüssel KF. Beim Verschlüsseln erzeugt der Chip das passende LEAF. Eine Entschlüsselung von Daten nimmt der Chip nur dann vor, wenn das zugehörige LEAF und der IV korrekt übergeben wurden. Die Prüfung des LEAF im Chip erfolgt dabei nach der Entschlüsselung mit dem Systemschlüssel KF anhand des 16 bit langen *Escrow Authenticators* EA. Dies soll eine Nutzung des „Clipper"-Chips mit gefälschtem LEAF (und damit eine Umgehung der Behörden-Hintertür) verhindern [Ruep_94].[3]

Entworfen wurde der „Clipper"-Chip von VLSI Technology Inc. Ein Prototyp, der mit 10 MHz getaktet wird und einen Durchsatz von 21 Mbit/s erreicht, stammt von der Firma Mykotronx (MYK-78) und ist für 30 US-Dollar erhältlich. Die zweite EES-Chip-Generation, Projektname „Capstone", enthält neben dem SKIPJACK-Algorithmus Basisoperationen für ein asymmetrisches Schlüsselaustauschverfahren (*public-key Key Exchange Algorithm*, KEA), den 1994 genormten *Digital Signature Algorithm* (DSA, [Fox_93, NIST2_94]) einen Rauschgenerator zur Erzeugung von Zufallszahlen und den 1993 standardisierten *Secure Hash*

---

3 Wie von Matt Blaze gezeigt wurde, erlaubt die Kürze des Authentikators eine *brute force*-Fälschung des LEAF [Blaz_94].

Algorithm (SHA, [NIST_93]). Ein ebenfalls von Mykotronx hergestellter, mit 10 MHz getakteter Prototyp (MYK-80) soll personalisiert maximal 85 US-Dollar kosten.

2.3 Problem „Generalschlüssel“

Eine der entscheidenden technischen Schwächen des EES wie auch anderer diskutierter *Key Escrow*-Systeme ist die Möglichkeit, mit den treuhänderisch hinterlegten Schlüsseln nach ihrer Herausgabe nicht nur die mit (richterlicher) Genehmigung abgehörten, sondern auch ältere und später kommunizierte Daten zu entschlüsseln. Diese Eigenschaft besitzen die vorgeschlagenen *Key Escrow*-Systeme unabhängig davon, ob die Schlüssel vollständig oder auf mehrere Treuhänder verteilt hinterlegt wurden.

Dadurch entsteht ein erhebliches Vertrauensproblem: Nach der Durchführung einer Abhörmaßnahme muß der abhörenden Instanz (Ermittlungsbehörde, Staatsschutz, ...) größeres Vertrauen entgegengebracht werden als den Schlüssel-Treuhändern. Dies gilt insbesondere dann, wenn bei den Treuhändern lediglich Teilschlüssel hinterlegt wurden, diese also nur dann unerlaubt auf verschlüsselte Kommunikationsdaten zugreifen können, wenn sie alle zusammenarbeiten.

In den USA wird versucht, dieser Verfahrensschwäche des EES mit organisatorischen Maßnahmen zu begegnen. Diese sollen verhindern, daß die herausgegebenen *Unit Key*-Teile KU1 und KU2 länger als für die genehmigte Abhörmaßnahme erforderlich in der Entschlüsselungseinrichtung verbleiben. Sogar eine anschließende Zerstörung des entschlüsselnden Gerätes, in das die Teilschlüssel verschlüsselt eingespeist werden, wurde vorgeschlagen [DeSm_94].

Damit wird jedoch nicht das grundsätzliche Vertrauensproblem gelöst – es erschwert lediglich (wenn auch durch den Betroffenen nicht überprüfbar) den Mißbrauch der Schlüssel zur Entschlüsselung späterer Kommunikation. Die Möglichkeit, zu einem früheren Zeitpunkt ohne richterliche Genehmigung aufgezeichnete Daten zu entschlüsseln, wird durch diese Maßnahmen jedoch nicht ausgeschlossen. Da das LEAF keinen Zeitstempel enthält, kann nachträglich nicht festgestellt werden, wann die vorliegenden Daten aufgezeichnet wurden.

3 Zeitabhängiges *Key Escrowing*

Wünschenswert ist daher eine Lösung, die technisch einen Mißbrauch der Schlüssel verhindert. Dies ist möglich durch die Verwendung von Schlüsseln, deren Gültigkeit auf den Abhörzeitraum beschränkt ist. Ein Schlüsselwechsel durch den Benutzer nach Abschluß der Abhörmaßnahme genügt allerdings nicht: Er kommt erstens in den meisten Fällen (zu) spät, da zwischen Abschluß der Abhörmaßnahme und der Benachrichtigung des Abgehörten erhebliche Zeit (in der Bundesrepublik bis zu 6 Monaten) vergehen kann, und schützt zweitens nicht die Daten, die zwischen dem letzten Schlüsselwechsel und dem Beginn der Abhörmaßnahme kommuniziert wurden. Zusätzlich würde ein solcher Wechsel erhebliche Kosten verursachen (im EES ist bspw. ein neuer Chip zu personalisieren und einzusetzen).

Der Anforderung, den Zugriff einer Ermittlungsbehörde technisch auf den gerichtlich festgelegten Zeitraum zu beschränken, genügen lediglich *Key Escrow* Systeme, bei denen die Abhöreinrichtung Schlüssel erhält, deren Gültigkeit auf das Abhörintervall begrenzt ist. Dabei können drei unterschiedliche Lösungsansätze unterschieden werden.

3.1 Schlüsselgenerierung mit Netzunterstützung

Zum ersten Ansatz zählen Vorschläge, die im Unterschied zum EES kein LEAF für den Behördenzugriff verwenden, sondern das Protokoll für die Vereinbarung des *Session Keys* zwischen zwei Kommunikationspartnern festlegen. Diese Protokolle räumen einer Ermittlungsbehörde bei Vorliegen einer gerichtlichen Anordnung Einfluß auf die Generierung der *Session Keys* ein: Der *Session Key* wird dabei aus „Startwerten“ gebildet, die von einem Netzwerkdienst bereitgestellt werden. Eine Abhörinstanz kann dem Netzwerk spezielle Werte vorgeben, die anschließend ein Entschlüsseln der Kommunikationsbeziehung ermöglichen (*KES with aktive investigator*) [HoMP_95].

Dieser Ansatz hat die interessante Eigenschaft, daß der Netzwerkdienst gegenüber Dritten den Nachweis führen kann, daß keine Ermittlungsbehörde die Generierung eines bestimmten „Startwerts“ beeinflußt hat und somit eine bestimmte Kommunikationsbeziehung auch nicht entschlüsselt werden konnte [BKOSW_94].

Allerdings verhindern die vorgeschlagenen Protokolle nur eingeschränkt die Entschlüsselung von Daten, die vor oder nach Ablauf des Abhörzeitraums aufgezeichnet wurden [HoMP_95]. Hinzu kommt das praktische Problem der Abhängigkeit der Protokolle von der *online*-Verfügbarkeit eines Netzwerk-Servers, der die Startwerte generiert und verteilt. Entscheidender Nachteil dieses Ansatzes ist jedoch die Beschränkung auf das Schlüsselaustauschprotokoll: Da die Ver- und Entschlüsselung der Nutzdaten unabhängig von der Schlüsselvereinbarung erfolgen, können Benutzer auch auf beliebig andere Weise ausgetauschte Schlüssel verwenden und so ein Abhören der Ermittlungsbehörden leicht verhindern. Dieser Ansatz ist daher mit der Kernidee des EES unvereinbar, ein *Key Escrow System* durch die Verbreitung preiswerter, kryptographisch starker Verschlüsselungsprozessoren zu etablieren.

3.2 *Session Keys* mit Tagesstempel

Von Whitfield Diffie stammt der Vorschlag, den im LEAF eingetragenen *Session Key* in zwei Teile zu zerlegen (KS = KS1 ⊕ KS2) und diese zusammen mit einem Tagesstempel t mit je einem *Unit Key*-Teilschlüssel zu verschlüsseln (siehe Bild 3).

Die abhörende Ermittlungsbehörde legt nach Entschlüsselung des LEAF die verschlüsselten Teile des *Session Key* den *Escrow Agents* vor. Diese geben die entschlüsselten Teile heraus, wenn der Tagesstempel t im Zeitraum der gerichtlichen Anordnung liegt [BKOSW_94].

Auf diese Weise erhält die Ermittlungsbehörde lediglich die im Abhörzeitraum verwendeten *Session Keys*, nicht aber den Generalschlüssel KU. Der zusätzliche Aufwand im Kryptomodul ist vernachlässigbar: Etwa zwei zusätzliche SKIPJACK-Verschlüsselungen je LEAF und *Escrow Agent*.

$$E_{KF}(U_{ID} \| E_{KU1}(KS1 \| t) \| E_{KU2}(KS2 \| t) \| ... \| E_{KUk}(KSk \| t) \| EA)$$

Bild 3: LEAF mit k verschlüsselten Session Key-Teilen

Dieser Ansatz hat allerdings mehrere Nachteile. So ist erstens die Zerlegung des *Unit Keys* KU im Kryptomodul zu speichern (80 Bits je *Escrow Agent*). Zweitens sind die Ermittlungsbehörden auf *online*-verfügbare *Escrow Agents* angewiesen: Diese müssen für jede Kommunikationsbeziehung die Schlüsselteile entschlüsseln. Die Analyse einer Kommunikationsbeziehung kann dadurch stark verzögert werden. Drittens wächst die Länge des LEAF mit jedem weiteren *Escrow Agent* um 128 bit.

Die Verwendung eines Tagesstempels im Kryptomodul erfordert die (authentische) Kenntnis eines aktuellen Tageszählers. Authentische Zeit-Protokolle mit einem *Time Server* sind praktisch ungeeignet, da die dafür erforderlichen Kommunikationskomponenten modifikationsgeschützt in das Kryptomodul integriert werden müßten. Anderenfalls wäre eine Täuschung des Kryptomoduls leicht durch Vorspiegelung einer falschen Zeit möglich. Auch müßte eine solche Lösung auf einen *online*-verfügbaren authentischen *Time Server* zugreifen können.

Verzichtet man auf einen *Time Server*, können alternativ synchronisierte und nicht-modifizierbare Tageszähler in den Kryptomodulen eingesetzt werden.[4] Da viele symmetrische und asymmetrische Authentifikations- und Schlüsselaustauschprotokolle wie z.B. Kerberos, SPX oder X.509 Zeitstempel zur Verhinderung von *Replay*-Angriffen und zur Reihenfolgeerhaltung der Protokollelemente verwenden, ist die Bereitstellung von zuverlässigen, synchronisierten und modifikationsgeschützten Uhren für viele Schlüsselaustauschprotokolle ohnehin erforderlich [BoMa_94, NeTs_94, StNS_88, DeSa_81]. Für diese Zeitstempel können üblicherweise Stundenzähler eingesetzt werden; für zeitabhängiges *Key Escrowing* genügen sogar schwach synchronisierte Tageszähler. Bei einem Eichintervall von 5 Jahren mit einer tolerierten Abweichung bis zu einem Tag ist eine Genauigkeit des Zählers von $\pm 5 \cdot 10^{-4}$ ausreichend.

Der im folgenden vorgestellte Ansatz arbeitet ebenfalls mit Tagesstempeln, kommt aber ohne Modifikationen des EES-LEAF-Formats aus und erfordert nicht für jedes abgehörte Gespräch einen Zugriff der Ermittlungsbehörden auf die *Escrow Agents*.

3.3 Tagesschlüssel als *Unit Key*

Eine Alternative zu Whitfield Diffies Vorschlag ist ein automatischer täglicher Wechsel des *Unit Keys*. Da Abhörgenehmigungen üblicherweise für einen in Tagen festgelegten Zeitraum ausgestellt werden, wird damit sichergestellt, daß nur die im genehmigten Zeitraum aufgezeichnete Kommunikation mit den herausgegebenen (Tages-)Schlüsseln entschlüsselt

---

4 Die Treuhänder sind davon nicht betroffen; sie können die aktuelle Zeit ihrerseits problemlos von einem beliebigen authentischen *Time Server* beziehen.

werden kann. Diese müssen daher auch nicht vor einer mißbräuchlichen Weiterverwendung geschützt werden. Bei diesem Ansatz ist keine Modifikation des LEAF erforderlich.

Für die Erzeugung und Verwendung täglich wechselnder Unit Keys bieten sich unterschiedliche technische Realisierungen an, die im folgenden vorgestellt und diskutiert werden. Alle Lösungen müssen eine wesentliche Eigenschaft besitzen: Für einen Angreifer bzw. eine abhörende Ermittlungsbehörde muß es praktisch (d.h. unter Annahme realistischen Speicher- und Rechenaufwands) unmöglich sein, auch nur einen Tagesschlüssel systematisch (d.h. zu einem beliebigen vorgewählten Datum) auch bei Kenntnis eines oder mehrerer Tagesschlüssel vorhersagen zu können.

3.3.1 Zufällige Tagesschlüssel

Eine sehr einfache Lösung ist die Vorausgenerierung sämtlicher Tagesschlüssel für die Lebenszeit eines Kryptomoduls durch einen (echten) Zufallsgenerator. Diese Schlüssel würden sowohl im Kryptomodul als auch beim Treuhänder (ggf. auch verteilt auf mehrere) in einer festgelegten Reihenfolge hinterlegt und besäßen genau einen Tag Gültigkeit. Die zufällige Generierung der Schlüssel macht das Schließen auf einen unbekannten Tagesschlüssel sogar *informationstheoretisch* unmöglich.

Im Kryptomodul kann die dabei anfallende Schlüsselmenge mit herkömmlicher Technik bewältigt werden (5 Jahre entsprechen bei einer Schlüssellänge von 80 bit etwa 18 kByte). Aufwendig ist allerdings die Speicherung der Tagesschlüssel bei den Treuhändern; der Speicherbedarf steigt dort bei einer durchschnittlichen Modul-Lebensdauer von 5 Jahren auf mehr als das 1800-fache.

Eine Verteilung des Geheimnisses auf mehrere *Escrow Agents* läßt sich bei dieser Lösung sehr leicht realisieren: Es genügt, die einzelnen Tagesschlüssel in k Teile zu zerlegen, die XOR-verknüpft wieder den Tagesschlüssel ergeben, und diese an die k Treuhänder zu verteilen. Dabei entsteht kein zusätzlicher Speicher- oder Rechenbedarf; die informationstheoretische Sicherheit des Verfahrens bleibt erhalten.

Die mit dieser Lösung einhergehende genaue Festlegung der maximalen Lebenszeit des Kryptomoduls kann für bestimmte Anwendungen vorteilhaft sein. Jedoch erfordert die zufällige Generierung von 80 Schlüsselbits je Tag und Kryptomodul Aufwand bei der Personalisierung der Module und der Verteilung der Schlüssel auf die *Escrow Agents*.

3.3.2 Pseudozufällige Tagesschlüssel

Da für praktische Zwecke *kryptographische Sicherheit* hinreichend ist, kann die Vorausgenerierung der Tagesschlüssel durch die Verwendung eines praktisch unvorhersagbaren Pseudozufallsgenerators (PZG) ersetzt werden. Dann genügt es, bei der Personalisierung die geheimen Parameter des Generators (Startwert, Schlüssel) zufällig zu wählen und diese im Kryptomodul und beim Treuhänder zu speichern. Für die Bestimmung des Tagesschlüssels ist jedoch zusätzlicher Rechenaufwand erforderlich. Im Kryptomodul ist diese einmal täglich anfallende Berechnung (bei Speicherung der Zwischenergebnisse) praktisch vernachlässigbar.

Anders bei den Treuhändern: Dort müssen zur Bestimmung des ersten Tagesschlüssels eines Abhörzeitraums ausgehend von dem geheimen, zufälligen Startwert des PZG alle Tagesschlüssel seit der Personalisierung des Kryptomoduls nachträglich generiert werden.

Aufwendig wird dieses Verfahren, wenn die Schlüsselhinterlegung verteilt auf mehrere Treuhänder erfolgen soll. In diesem Fall ist je Treuhänder und Teilnehmer ein eigener PZG (bzw. ein anderes Geheimnis) erforderlich. Im Kryptomodul müssen ebenfalls alle von den PZGs generierten Teilschlüssel erzeugt werden können, d.h. das Modul muß Schlüssel und Startwerte aller PZGs der Treuhänder kennen. Die Verknüpfung (z.B. via XOR) der Teilschlüssel liefert dann den Tagesschlüssel. Damit vervielfacht sich der Speicher- und Rechenaufwand im Kryptomodul mit der Zahl der Treuhänder.

Als PZG bieten sich vor allem zwei Verfahren an: eine Blockchiffre im OFB-Mode (*Output Feedback Mode*, [FuRi_94]) oder ein kryptographisch unvorhersagbarer Generator.

In Verbindung mit dem „Clipper"-Chip empfiehlt sich die Verwendung des SKIPJACK-Algorithmus im OFB-Mode. Für die Berechnung eines 80-bit-Tagesschlüssels sind bei k Treuhändern dann täglich 2•k Verschlüsselungen im Kryptomodul erforderlich. Im Modul sind 144•k bit lange PZG-Parameter (ein 80-bit-Schlüssel und ein 64-bit-Start- bzw. Zwischenwert je Treuhänder) zu speichern. Die Periodenlänge des Generators sollte – sofern SKIPJACK kryptographisch stark ist – bei 2^{80} bit liegen.

Mit dem „Capstone"-Chip ließe sich auch der kryptographisch starke, bewiesenermaßen dem Faktorisierungsproblem äquivalente PZG von Blum, Blum und Schub (BBS) einsetzen [BlBS_86]. Ausgehend von einem zufälligen Startwert werden durch modulares Quadrieren pseudozufällige, nichtvorhersagbare Bits generiert. Die Stärke des Generators (d.h. dessen Periodenlänge und Unvorhersagbarkeit) läßt sich über die Schlüssellänge beeinflussen: Mit einem m bit langen Modulus erreicht der BBS-Generator eine Periodenlänge von 2^m bit [BlBS_86]. Der Aufwand steigt allerdings auch mit wachsender Schlüssellänge: Je Generatoroperation können die log(|m|) letzten Bits als Zufallswerte verwendet werden. Wählt man also eine BBS-Moduluslänge von 512 bit, sind 10•k Operationen für die Generierung eines Tagesschlüssels erforderlich. Im Kryptomodul sind 1024•k bit (Modulus, Zwischenergebnis) zu speichern.

Treuhänder müssen zur Bestimmung des ersten Tagesschlüssels einer Abhörperiode sämtliche Schlüssel seit der letzten Abhörmaßnahme, schlimmstenfalls seit der Personalisierung des Kryptomoduls mit je 10 modularen Quadrierungen erzeugen; die Schlüssel der Folgetage ergeben sich dann durch jeweils weitere 10 modulare Quadrierungen. Bei einer 5 Jahre zurückliegenden Personalisierung kann sich dieser Vorgang auf 18.000 Operationen summieren; mit weniger als einer Sekunde sind diese jedoch in akzeptabler Zeit durchführbar.

3.3.3 Tagesstempel mit Einwegfunktion

Eine Generierung sämtlicher Tagesschlüssel bei den *Escrow Agents* ist nicht erforderlich, wenn das Tagesdatum nicht indirekt (über die Ordnungsnummer der Zufalls- oder Pseudozufallszahl), sondern direkt in den Tagesschlüssel eingeht. Dies muß so erfolgen, daß eine Ablei-

tung weiterer Tagesschlüssel ohne Berechtigung (d.h. ohne Kenntnis eines Geheimnisses) praktisch unmöglich ist, z.B. durch die Verwendung einer Einwegfunktion mit geheimem Schlüssel.

Eine gute Blockchiffre ist eine sehr schnelle Einwegfunktion. Dabei wird das Tagesdatum mit einem zufälligen, geheimen Schlüssel KU verschlüsselt, um den aktuellen Tagesschlüssel KD zu erhalten. Voraussetzung für die Unvorhersagbarkeit der Tagesschlüssel ist allerdings, daß erstens praktisch kein aktiver Angriff (Unterschieben eines falschen Tagesdatums) möglich ist und das Verfahren zweitens auch nicht existentiell gebrochen werden kann (Finden wenigstens eines Klartextes zu einem gegebenen Schlüsseltext). Dazu kann der SKIPJACK-Algorithmus eingesetzt werden. Von Otto Leiberich stammt der folgende Vorschlag zur Berechnung des Tagesschlüssels [Leib_95]:

$$KD(t) = E_{KU \oplus t}(KU)$$

Offen ist dabei, ob durch die Verwendung des *Unit Keys* KU als Schlüssel und Klartextblock ein Entschlüsseln des KU erleichtert wird (t ist einem Angreifer natürlich bekannt).

Auch bei dieser Lösung ist es erforderlich, daß bei verteilter Hinterlegung im Kryptomodul alle k Teil-Tagesschlüssel im Kryptomodul generiert und verknüpft werden. Dafür sind zusätzlich 80•k Bits (geheime Schlüssel für die Teilschlüsselbestimmung) im Kryptomodul zu speichern. Der Rechenaufwand ist vernachlässigbar: 2•k SKIPJACK-Operationen je Tagesschlüssel.

Eine besonders elegante Lösung ist auf der Basis des diskreten Logarithmusproblems möglich. Dazu wird systemweit ein primer Modulus p (mit $|p| \geq 512$ bit) festgelegt. Jeder Chip wird mit einer einmaligen, nicht notwendig geheimen Primitivwurzel $a < p$, $|a| = |p|$ personalisiert. Als *Unit Key* KU dient ein geheimer Exponent e (mit ggT(e, p-1) = 1). Der Tagesschlüssel (*Day Key*) KD(t) berechnet sich nun aus Tagesstempel t und *Unit Key* KU durch eine modulare Multiplikation und eine Exponentiation:[5]

$$KD(t) = a^{e \bullet h(t)} = a^{(KU \bullet h(t)) \bmod (p-1)} \bmod p.$$

Über die zur Durchführung dieser Tagesschlüsselbestimmung erforderliche modulare Exponentiationsfunktion und eine standardisierte, kollisionsresistente Hashfunktion h (SHA) verfügt der „Capstone"-Chip (siehe Bild 4).[6]

---

[5] Natürlich genügen die letzten m Bits des Ergebnisses (mit m = Schlüssellänge der Blockchiffre; für SKIPJACK demnach 80 Bits); oder aber KD(t) wird auf die Länge m gehasht.

[6] Die Hashfunktion sollte so gewählt werden, daß ggT(h(t), p-1) = 1 gilt. Das kann leicht durch die Wahl von p und h sichergestellt werden (z.B. p-1 = 2•q, q prim, und h(t) = 2•SHS(t)+1).

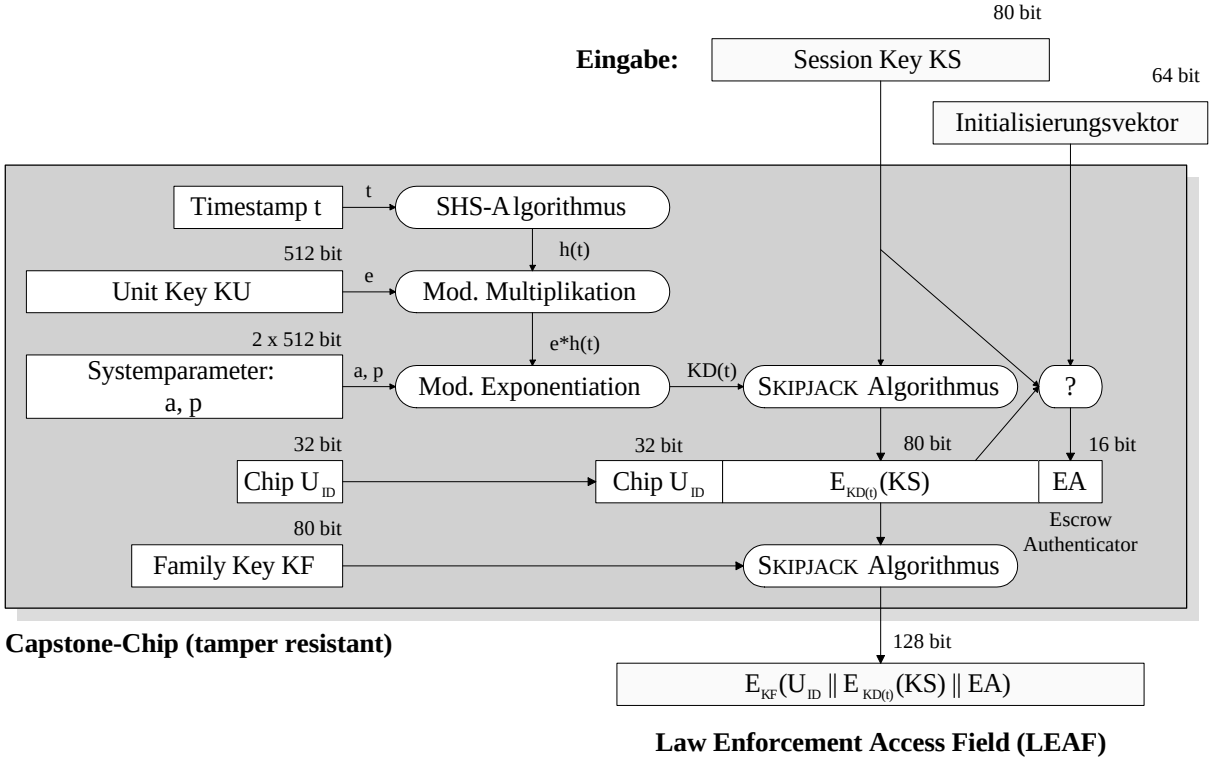

Bild 4: LEAF Creation Method für täglich wechselnde Unit Keys

Die Stärke dieses Ansatzes zeigt sich, wenn der *Unit Key* KU auf mehrere *Escrow Agents* verteilt wird. Dazu genügt es, den geheimen Exponenten e in Teile e_1, e_2, ..., e_k mit $e = e_1 + e_2 + ... + e_k$ und $ggT(e_i, p-1) = 1$ zu zerlegen ($KU_i = e_i$). Dann gilt:

$$KD(t) = a^{e \bullet h(t)} = a^{e1 \bullet h(t)} \bullet a^{e2 \bullet h(t)} \bullet ... \bullet a^{ek \bullet h(t)} = a^{(e1+e2+...+ek) \bullet h(t)}$$

Ohne Kenntnis des geheimen Exponenten e kann ein Tagesschlüssel nur über die Bestimmung des Diskreten Logarithmus gewonnen werden. Die Hashfunktion h im Exponenten zerstört die Struktur des Tagesstempels und verhindert, daß aus einem Tagesschlüssel systematisch ausgewählte weitere gewonnen werden können (z.B. KD(n•t)). Selbst ein Angreifer, der den Tagesschlüssel KD(t) und alle Teilschlüssel KU_i bis auf einen (e_x) kennt, kann diesen nur gewinnen, wenn er den Diskreten Logarithmus berechnen kann. Sei $g = a^{h(t)} \bmod p$, dann gilt:

$$KD = g^e = g^{e1} \bullet g^{e2} \bullet ... \bullet g^{ek} \bullet g^{ex} \bmod p \Rightarrow KD \bullet g^{-(e1+e2+...+ek)} = g^{ex} \bmod p$$

Bei dieser Lösung genügt – unabhängig von der Zahl der *Escrow Agents* – die Haltung eines geheimen *Unit Key* e und der Systemparameter p und a im Kryptomodul. Der Rechenaufwand reduziert sich auf eine modulare Exponentiation je Tagesschlüssel. Ermittlungsbehörden, die eine gerichtliche Anordnung vorlegen, erhalten von den *Escrow Agents* die Teil-Tagesschlüssel für jeden Tag des genehmigten Abhörzeitraums:

$$KD_i(t) = a^{ei \bullet h(t)}$$

Diese können daraus die Tagesschlüssel bestimmen:

$$KD(t) = \prod_{j=1..k} KD_i(t) \bmod p$$

Fazit

Es wurden unterschiedliche Ansätze zur Realisierung eines zeitabhängigen *Key Escrowing* vorgestellt und hinsichtlich ihres Aufwandes und ihrer wesentlichen Eigenschaften diskutiert. Mit einem solchen *Time Dependent Key Escrow System* läßt sich ein schwerwiegender Nachteil herkömmlicher *Key Escrow* Systeme beseitigen: die prinzipielle Möglichkeit von Ermittlungsbehörden, die Entschlüsselung beim *Law Enforcement*-Zugriff unzulässig auch auf ältere Daten auszudehnen oder den Schlüssel für später aufgezeichnete Kommunikation aufzubewahren.

Die folgende Tabelle gibt eine kurze vergleichende Übersicht zu den in Kapitel 3 vorgestellten Lösungsansätzen für zeitabhängiges *Key Escrowing* hinsichtlich des Speicher- und Rechenaufwandes im Kryptomodul.

Verfahren	Speicherbedarf	Rechenaufwand	Bemerkung
3.2 *Session Keys* mit Tagesstempel (Diffie)	80•k bit	3•k SKIPJACK-Operationen je *Session Key*	LEAF-Verlängerung, je *Session Key* Anfrage bei *Escrow Agents*
3.3.1 Zufällige Tagesschlüssel	80•365•j bit	-	sehr viele Schlüssel (vorausgeneriert)
3.3.2 Pseudozufällige Tagesschlüssel (SKIPJACK)	144•k bit	2•k SKIPJACK-Operationen täglich	Nacherzeugung aller Tagesschlüssel bei *Escrow Agents*
3.3.2 Pseudozufällige Tagesschlüssel (BBS)	1024•k bit	10•k modulare Quadrierungen täglich	Modulararithmetik erforderlich
3.3.3 Einwegfunktion (SKIPJACK)	80•k bit	2•k SKIPJACK-Operationen täglich	
3.3.3 Einwegfunktion (mod. Exponentiation)	1536 bit	1 modulare Exponentiation täglich	Modulararithmetik erforderlich

k: Anzahl der beteiligten *Escrow Agents* bei verteilter Schlüsselhinterlegung

j: Lebensdauer eines KES-Kryptomoduls in Jahren

Literatur

ANSI_81 American National Standards Institute (ANSI): *Data Encryption Algorithm*. ANSI X3.92, 1981.

BDKMT_93 Brickell, Ernest F.; Denning, Dorothy E.; Kent, Stephen T.; Maher, David P.; Tuchmann, Walter: *The SKIPJACK Algorithm*. Interim Report, 28. Juli 1993.

BKOSW_94 Beth, Thomas; Knobloch, H.-J.; Otten, Marcus; Simmons, Gustav J.; Wichmann, Peer: *Towards Acceptable Key Escrow Systems*. Proceedings of the 2nd ACM Conference on Computer and Communications Security, Fairfax, 1994, S. 51-58.

Blaz_94 Blaze, Matt: *Protocol Failure in the Escrowed Encryption Standard*. Proceedings of the 2nd ACM Conference on Computer and Communications Security, Fairfax, 1994, S. 59-67.

BlBS_86 Blum, L.; Blum, M.; Schub, M.: *A Simple Unpredictable Pseudo-Random Number Generator*. SIAM J. Computing, 15/2, 1986, S. 364-383.

BoMa_94 Boyd, Colin; Mao, Wenbo: *Designing Secure Key Exchange Protocols*. In: Gollmann, D. (Hrsg.): Proceedings of ESORICS '94, LNCS 875, Springer, Heidelberg 1994, S. 93-105.

DaPr_89 Davies, Donald W.; Price, Wyn L.: *Security for Computer Networks*. 2. Auflage, John Wiley & Sons Ltd., Chichester 1989.

DeSa_81 Denning, Dorothy E.; Sacco, Giovanni Maria: *Timestamps in Key Distribution Protocols*. Communications of the ACM, Vol. 24, No. 8, August 1981, S. 533-536.

DeSm_94 Denning, Dorothy E.; Smid, Miles: *Key Escrowing Today*. IEEE Communications Magazine, 9/94, S. 58-68.

DiHe_76 Diffie, Whitfield; Hellman, Martin E.: *New Directions in Cryptography*. IEEE Transactions on Information Theory, Bd. IT-22, Nr. 6, 1976, S. 644-654.

Fox_93 Fox, Dirk: *Der 'Digital Signature Standard': Aufwand, Implementierung und Sicherheit*. In: Weck, G.; Horster, P. (Hrsg.): Verläßliche Informationssysteme, Proceedings der GI-Fachtagung VIS '93, DuD-Fachbeiträge 16, Vieweg, Braunschweig 1993, S. 333-352.

Froo_95 Froomkin, Michael A.: *The Metaphor is the Key: Cryptography, the Clipper Chip and the Constitution*. Erscheint 1995.

FuRi_94 Fumy, Walter; Rieß, Hans Peter: *Kryptographie*. Schriftenreihe Sicherheit in der Informationstechnik, Band 6. Oldenbourg Verlag, München, 2. Auflage 1994.

HoMP_95 Horster, Patrick; Michels, Markus; Petersen, Holger: *Ein neues Key Escrow System mit aktivem Abhörer*. Trust Center 95, in diesem Tagungsband.

Leib_95 Leiberich, Otto: *Verschlüsselung und Kriminalität (II)*. BSI-Forum, Kommunikations- und EDV-Sicherheit (KES), 1/95, S. 60-61.

NBS_77 National Bureau of Standards (NBS): *Data Encryption Standard (DES)*. Federal Information Processing Standards Publication (FIPS-PUB) 46-1, US Department of Commerce, 1/1977.

NBS_80 National Bureau of Standards (NBS): *DES modes of Operations*. Federal Information Processing Standards Publication (FIPS-PUB) 81, US Department of Commerce, 12/1980.

NeTs_94 Neumann, B. Clifford; Ts'o, Theodore: *Kerberos: An Authentication Service for Computer Networks*. IEEE Communications Magazine, Sept. 1994, S. 33-38.

NIST_93 National Institute of Standards and Technology (NIST): *Secure Hash Standard (SHS)*. Federal Information Processing Standards Publication (FIPS-PUB) 180-1, US Department of Commerce, 31. Mai 1993.

NIST1_94 National Institute of Standards and Technology (NIST): *Escrowed Encryption Standard (EES)*. Federal Information Processing Standards Publication (FIPS-PUB) 185, US Department of Commerce, 9. Februar 1994.

NIST2_94 National Institute of Standards and Technology (NIST): *Digital Signature Standard (DSS)*. Federal Information Processing Standards Publication (FIPS-PUB) 186, US Department of Commerce, 19. Mai 1994.

Ruep_94 Rueppel, Rainer A.: *Clipper - Der Krypto-Konflikt am Beispiel der Amerikanischen ESCROW Technologie*. Datenschutz und Datensicherung (DuD), 8/94, S. 443-451.

StNS_88 Steiner, Jennifer G.; Neumann, Clifford; Schiller, Jeffrey I.: *Kerberos: An Authentication Service for Open Network Systems*. USENIX, Winter '88, Dallas, 1/1988, S. 1-15.