UNIVERSITÄT
KOBLENZ · LANDAU
Institut für Wirtschafts-
und Verwaltungsinformatik

FB 4
Informatik

Anwendungsbeispiele für Kryptographie

Rüdiger Grimm
Helge Hundacker
Anastasia Meletiadou

Nr. 2/2007

Arbeitsberichte aus dem Fachbereich Informatik

Universität Koblenz-Landau • Campus Koblenz • Universitätsstraße 1 • 56070 Koblenz
Telefon +49 261 287-0 • http://www.uni-koblenz.de

Arbeitsberichte des Fachbereichs Informatik

I**SSN:** 1864-0346

Herausgeber / Edited by:

Der Dekan:
Prof. Dr. Paulus

Die Professoren des Fachbereichs:
Prof. Dr. Bátori, Jun.-Prof. Dr. Beckert, Prof. Dr. Burkhardt, Prof. Dr. Diller, Prof. Dr. Ebert, Prof. Dr. Furbach, Prof. Dr. Grimm, Prof. Dr. Hampe, Prof. Dr. Harbusch, Jun.-Prof. Dr. Hass, Prof. Dr. Krause, Prof. Dr. Lautenbach, Prof. Dr. Müller, Prof. Dr. Oppermann, Prof. Dr. Paulus, Prof. Dr. Priese, Prof. Dr. Rosentahl, Prof. Dr. Schubert, Prof. Dr. Staab, Prof. Dr. Steigner, Prof. Dr. Troitzsch, Priv.-Doz. Dr. von Kortzfleisch, Prof. Dr. Walsh, Prof. Dr. Wimmer, Prof. Dr. Zöbel

Kontaktdaten der Verfasser

Rüdiger Grimm, Helge Hundacker, Anastasia Meletiadou
Institut für Wirtschafts- und Verwaltungsinformatik
Fachbereich Informatik
Universität Koblenz-Landau
Universitätsstraße 1
D-56070 Koblenz
EMail: grimm@uni-koblenz.de, hundacker@uni-koblenz.de, nancy@uni-koblenz.de

Inhaltsverzeichnis

Einleitung

In den folgenden acht Beispielen wird gezeigt, wie elektronische Anwendungen mithilfe kryptographischer Verfahren abgesichert werden. In jedem Beispiel erklären wir das Ziel der Verschlüsselung, erläutern den technischen Hintergrund, besprechen im Detail den Ablauf der Anwendung mit ihrer Verschlüsselung und diskutieren Stärken und Schwächen des Verfahrens.

D Das erste Beispiel der Verschlüsselungsprotokolle ist noch nicht sehr konkret, sondern es bildet die Grundlage der folgenden Beispiele. Denn jede Anwendung setzt eines der beiden Typen von Verschlüsselungsprotokollen ein, die im ersten Beispiel erklärt werden. Dann folgen die Beispiele Sicheres World Wide Web mit SSL, die Verschlüsselung der Luftschnittstelle im Mobilfunk, die sichere Identifikation des Karteninhabers einer ec-Karte am Geldauszahlungsautomaten, der Schutz von WLANs gegen fremde Lauscher und Eindringlinge, die sichere Identifikation elektronischer Schlüssel, darunter der Funkschlüssel bei Autotüren, das kontaktlose Flugticket zum Ausdrucken einer Boardingkarte und schließlich die Verschlüsselung im Pay-TV.

Die Beschreibung der kryptografischen Anwendungen dieses Kapitels wird in einem Arbeitspapier des Instituts für Wirtschafts- und Verwaltungsinformatik der Universität Koblenz-Landau weitergepflegt und dort stets aktuell zum Download bereitgehalten [Grimm, Hundacker, Meletiadou 2006]. http://www.uni-koblenz.de/FB4/Institutes/IWVI/AGGrimm/Downloads

1. Protokolle (symmetrisch und asymmetrisch)

Anwendung

Um Verschlüsselungen nutzbringend anwenden zu können, müssen sie in die Kommunikation zwischen den Partnern integriert werden. Wenn Alice und Bob verschlüsselte Botschaften austauschen wollen, dann müssen sie genau wissen, in welcher Reihenfolge sie sich gegenseitig zu erkennen geben, ihr Verfahren verabreden, Schlüssel austauschen und sich schließlich die verschlüsselten Texte zusenden. Solche Regeln, wer wem was in welcher Reihenfolge zusendet, nennt man ein „Protokoll“. Wenn ein Funkschlüssel seinem Schloss eine verschlüsselte Botschaft zusenden will, „bitte jetzt öffnen“, dann müssen Schlüssel und Schloss über ein wohl verabredetes Protokoll verfügen, sonst würden sie sich nicht „verstehen“, das heißt, sie würden ihr Ziel, das Schloss auf Entfernung zu öffnen – und zwar nur dem berechtigten Schlüssel – nicht erreichen.

Ziel von Verschlüsselungsprotokollen

Das Ziel von Verschlüsselungsprotokollen ist es, dass Alice und Bob voneinander genau wissen, dass der jeweils andere wirklich der ist, der er vorgibt zu sein (Authentizität) und dass sie dann

verschlüsselte Botschaften austauschen können, die andere vielleicht mitlesen, aber niemals entschlüsseln und daher auch nicht verstehen können (Vertraulichkeit). Ohne die gegenseitige Authentifizierung als ersten Schritt ist die Verschlüsselung als zweiter Schritt nur halb so viel wert – leider setzt sich diese Erkenntnis nur langsam durch.

Es gibt grundsätzlich zwei verschiedene Typen von Protokollen: In dem einen Typ wird die gegenseitige Authentifizierung mit einem symmetrischen Algorithmus abgesichert. Deswegen heißt dieser Protokolltyp symmetrisch. Bei symmetrischen Protokollen müssen Alice und Bob *vorab* einen symmetrischen Schlüssel über einen sicheren Kanal austauschen, zum Beispiel über Telefon, Brief oder bei einem persönlichen Treffen.

In dem anderen Typ wird die gegenseitige Authentifizierung mit einem asymmetrischen Algorithmus abgesichert, er heißt daher asymmetrisch. Bei asymmetrischen Protokollen brauchen Alice und Bob vorab keinen Schlüsselaustausch über einen sicheren Kanal zu organisieren, sondern sie verschlüsseln ihre Botschaften, die sie im Laufe des Protokolls austauschen, mit asymmetrischen Verfahren. Allerdings erfordert die Authentifizierung mit asymmetrischen Verfahren ein glaubwürdiges Zertifikat der verwendeten öffentlichen Schlüssel. Ein solches Zertifikat wird von einer vertrauenswürdigen neutralen Instanz, einer so genannten Zertifizierungsstelle, ausgestellt.

In einer Mischform, bei den so genannten hybriden Verfahren, tauschen Alice und Bob einen spontan erzeugten symmetrischen Schlüssel asymmetrisch verschlüsselt aus und schicken sich dann ihre vertraulichen Botschaften mit diesem „Sitzungsschlüssel“ symmetrisch verschlüsselt zu.

Technische Grundlagen

Protokolle sind Kommunikationsregeln, die die erlaubten Datenformate und die Reihenfolge der Kommunikationsschritte festlegen. Außerdem beschreiben Protokolle eine Schnittstelle zu ihrer Anwendung, aus der hervorgeht, wem und wozu dieses Protokoll dient. Und schließlich beschreiben Protokolle die Hilfsmittel, die sie benötigen, um ausgeführt zu werden, also etwa welche Netzressourcen sie brauchen, um ihre Protokolldaten auszutauschen. Verschlüsselungsprotokolle legen also je nach Anwendung ihre spezifischen technischen Grundlagen fest.

Eingesetzte Algorithmen

Alle Algorithmen kommen in Verschlüsselungsprotokollen vor. Welches in einem konkret implementierten Protokoll eingesetzt wird, wird von der jeweiligen Anwendung festgelegt, oder sogar im Laufe des Protokolls miteinander verabredet.

Ablauf

Der symmetrische Verschlüsselungsprotokolltyp:

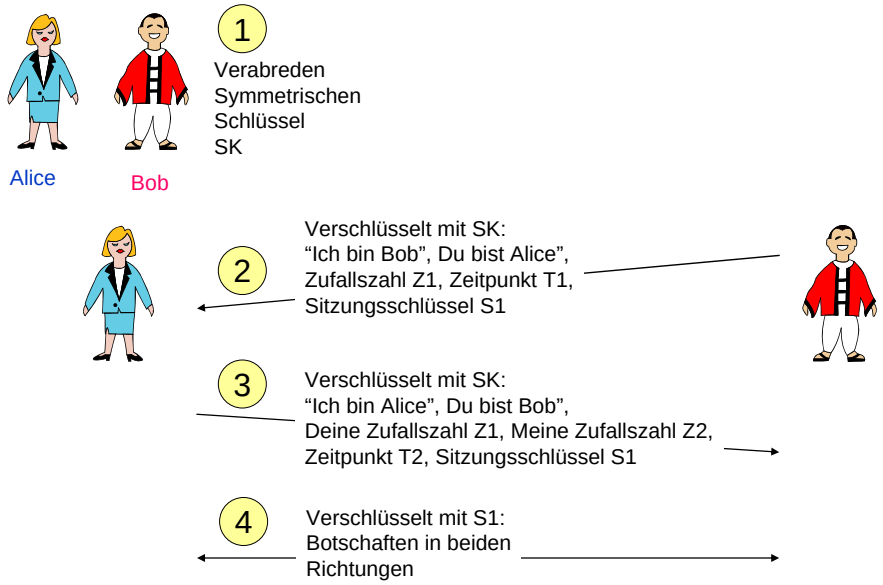

Abb. 1.1: Symmetrische Verschlüsselungsprotokolle

Schritt 1: Vorab verabreden Alice und Bob einen geheimen symmetrischen Schlüssel SK. Wenn sie sich zum Beispiel bei einer Konferenz treffen und dort verabreden, Triple-DES zu verwenden, dann kann Alice sich von ihrem lokalen Verschlüsselungstool einen 112-Bit-Triple-DES-Schlüssel erzeugen lassen und diesen auf einen Memory-Stick speichern, den Bob in sein Laptop einsteckt und ausliest. Es ist allerdings auch möglich, den symmetrischen Schlüssel über ein asymmetrisches Verfahren über das Internet auszutauschen: Bob könnte einen 112-Bit-Triple-DES-Schlüssel lokal bei sich erzeugen und diesen mit Alices öffentlichem PGP-Schlüssel verschlüsseln und ihr per Email zuschicken. Alice würde dann die Email entschlüsseln und den darin enthaltenen 112-Bit-Triple-DES-Schlüssel auslesen und ihrem lokalen Verschlüsselungstool zur Verfügung stellen. Ganz analog könnte Alice auch Bob einen von ihr erzeugten symmetrischen Schlüssel sicher zukommen lassen

Wenn Alice und Bob nun über einen gemeinsamen symmetrischen Schlüssel SK verfügen, dann können sie sich gegenseitig an der Verwendung genau dieses einen Schlüssels erkennen und ihn zur exklusiven Verschlüsselung von Botschaften zwischen sich einsetzen. Voraussetzung dafür ist, dass sie den Schlüssel wirklich für sich behalten und nicht anderen Personen weitergeben.

Ein möglicher Ablauf eines symmetrischen Verschlüsselungsprotokolls könnte etwa so vor sich gehen:

Schritt 2. Bob möchte die Kommunikation mit Alice eröffnen und schickt ihr eine Botschaft, die er mit SK verschlüsselt. Darin nennt er ihr seinen Namen „Bob“, sagt ihr, dass er mit ihr, „Alice“, reden will, nennt die aktuelle Uhrzeit T1 und eine Zufallszahl Z1. Alice erkennt an der Uhrzeit, dass Bob sich tatsächlich jetzt an sie wendet und dass nicht etwa ein fremder Lauscher eine vorher gesendete Eröffnungsbotschaft eingespielt hat. Wenn Alice sämtliche Zufallszahlen, die Bob vorher einmal verwendet hatte, abgespeichert hat und bei jeder neuen Eröffnung die neue Zufallszahl auf ihre Frische prüft, dann erkennt Alice die Aktualität der Eröffnung auch daran, dass die Zufallszahl von Bob neu ist.

Schritt 3: Alice antwortet Bob, indem sie beide Namen sowie Bobs Zufallszahl wiederholt. Daran erkennt Bob, dass Alices Antwort wirklich frisch ist. Das würde eigentlich genügen. Aber wenn sie sich die Frische jeder Antwort sichern wollen, kann Alice ebenfalls ihrerseits eine Zufallszahl Z2 senden, die Bob bei seiner nächsten Botschaft zitiert, u.s.w.

Schritte 4 und folgende: Bob hatte in seiner Eröffnungsbotschaft einen neuen symmetrischen Schlüssel S1 mit gesendet, den Alice und Bob in dieser Sitzung für die verschlüsselten Botschaften nutzen werden. Der Vorteil von immer neuen Sitzungsschlüsseln liegt darin, dass wechselnde Schlüssel Angreifern das Raten von Schlüsseln erschwert. Sollte einmal tatsächlich ein Sitzungsschlüssel geknackt werden, so ist das für die nächsten Sitzungen nicht gefährlich. Allerdings: Der ursprünglich gemeinsame Schlüssel SK sollte nicht geknackt werden können, denn der eröffnet ja jede neue Sitzung.

Dieser symmetrische Verschlüsselungsprotokolltyp wird zum Beispiel von Autofunkschlössern und elektronischen Türschlössern verwendet. In geschlossenen Netzen eines Universitätscampus oder einer größeren Firmenniederlassung setzen die Rechenzentren oft das so genannte Kerberos-Verfahren ein, damit sich Anwender und Netzdienste gegenseitig erkennen und verschlüsselte Aufträge schicken können. Das Kerberos-Protokoll baut zwar auf diesem symmetrischen Protokolltyp auf, ist aber etwas kompliziert. Bei Kerberos hat jeder Teilnehmer einen (anderen) symmetrischen Schlüssel mit einem zentralen Server, dem so genannten Kerberos-Server. Bevor dann ein Nutzer, sagen wir Alice, einen Dienst-Server, sagen wir einen Netzdrucker, anspricht, besorgt sie sich vom Kerberos-Server einen symmetrischen Verschlüsselungsschlüssel SK für diese eine Sitzung. Kerberos erzeugt einen solchen Session-Schlüssel SK für Alice und den Netzdrucker und schickt ihn Alice mit dem symmetrischen Schlüssel von Alice verschlüsselt zu. Außerdem schickt er SK Alice noch einmal anders verschlüsselt zu, nämlich mit dem symmetrischen Schlüssel dieses Druckers. Wenn Alice also diesen zweiten Datensatz an den Drucker weiterleitet, kann dieser SK mit Hilfe seines eigenen symmetrischen Schlüssels, den er mit dem Kerberos-Server teilt, entschlüsseln und für die weitere Kommunikation mit Alice verwenden.

Der asymmetrische Verschlüsselungsprotokolltyp:

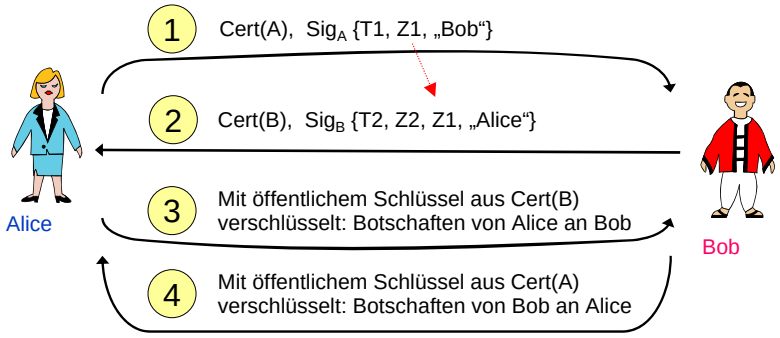

Abb. 1.2: Asymmetrische Verschlüsselungsprotokolle

Am Beispiel der X.509-Zwei-Wege-Authentifizierung kann der asymmetrische Verschlüsselungsprotokolltyp erklärt werden. Alice und Bob verfügen dabei jeweils über einen eigenen öffentlichen Schlüssel, der von einer Zertifizierungsstelle in einem Zertifikat Cert(Alice), bzw. Cert(Bob) bestätigt ist. Das Zertifikat enthält den öffentlichen Schlüssel selbst, den Namen seines Eigentümers (also „Alice“ in Cert(A) und „Bob“ in Cert(B)) und einen Hinweis darauf, mit welchem Algorithmus das Zertifikat ausgestellt worden ist und mit welchem der zertifizierte öffentliche Schlüssel gebraucht werden kann. Hier werden die Zertifikate erst nur zum Signieren und Verifizieren benutzt. Nach der erfolgreichen Authentifizierung werden sie auch zur Verschlüsselung benutzt.

Schritt 1: Diesmal soll Alice die Kommunikation eröffnen. Sie erzeugt eine einmalige Zufallszahl Z1 und signiert diese zusammen mit der aktuellen Zeitangabe T1 und dem Namen des gewünschten Gesprächspartners Bob und schickt diesen Datensatz an Bob, dem sie noch das Zertifikat ihres eigenen öffentlichen Schlüssels Cert(A) hinzufügt.

Schritt 2: Bob prüft die Signatur von Alice mit Hilfe des öffentlichen Schlüssels aus Alices Zertifikat Cert(A), dann prüft er, ob er gemeint ist („Bob“), ob die Zeitangabe T1 aktuell ist und ggf. zusätzlich noch, ob er die Zufallszahl Z1 schon einmal vorher von Alice bekommen hatte. Daran erkennt er, dass Alice ihm eine frische Eröffnungsnachricht geschickt hat und behält sowohl die Zufallszahl Z1, als auch den öffentlichen Schlüssel von Alice zur Verfügung. Er erzeugt nun seinerseits eine einmalige Zufallszahl Z2 und signiert diese zusammen mit der aktuellen Zeitangabe T2, dem Zitat von Alices Zufallszahl Z1 und dem Namen des gewünschten Gesprächspartners

Alice und schickt diesen Datensatz an Alice zurück, dem er seinerseits noch das Zertifikat Cert(B) seines eigenen öffentlichen Schlüssels hinzufügt.

Schritt 3: Alice prüft, ob Bobs Nachricht wirklich von Bob kommt, indem sie die Signatur von Bobs Datensatz mit Hilfe des öffentlichen Schlüssels aus Cert(B) prüft. Außerdem prüft sie, ob die Zeit T2 aktuell ist, ob Bobs Zufallszahl Z2 (jedenfalls für sie) neu ist und ob Bob ihre Zufallszahl Z1 richtig zitiert hat. Nun vertraut Alice, dass Bob wirklich Bob ist und auf ihre weiteren Nachrichten wartet. Sie verwendet also nun Bobs öffentlichen Schlüssel aus Cert(B) und verschlüsselt damit eine vertrauliche Botschaft an Bob. Sie könnte zum Beispiel ein hybrides Verfahren einschlagen, indem sie einen symmetrischen Sitzungsschlüssel erzeugt und diesen mit dem öffentlichen Schlüssel von Bob verschlüsselt an Bob schickt.

Schritt 4: Bob tut analog dasselbe wie Alice in Schritt 3. Er könnte fortan alle vertraulichen Botschaften an Alice mit ihrem öffentlichen Schlüssel aus Cert(A) verschlüsseln, oder er verwendet dazu einen symmetrischen Schlüssel, den ihm Alice in Schritt 3 vielleicht zugesendet hatte.

Dieses Protokoll ist hier etwas vereinfacht dargestellt worden. Denn aus Sicherheitsgründen werden öffentliche Schlüssel, die zur Verifikation einer digitalen Signatur verwendet werden, niemals zum Verschlüsseln von Nachrichten verwendet. Deshalb müssten Alice und Bob sich jeweils zwei Zertifikate zuschicken: Eines, das den öffentlichen Schlüssel zur Verifikation der Signatur enthält, und ein anderes, das einen davon verschiedenen öffentlichen Schlüssel zum Verschlüsseln von Nachrichten enthält.

Das populärste Protokoll, das auf X.509-Zertifikaten aufbaut – allerdings in einer erheblich veränderten Form – ist das so genannte **SSL-Protokoll** („Secure Sockets Layer“), das Web-Browser und Web-Server verwenden, wenn sie sichere https-Verbindungen aufbauen.

Stärken

Es gibt starke symmetrische Verschlüsselungsalgorithmen wie Triple-DES, AES und IDEA. Wenn Alice und Bob ihre Schlüssel geheim halten, dann ist dieser Protokolltyp nach unserem heutigen Wissensstand nicht zu knacken. Er hat gegenüber asymmetrischen Verfahren den großen Vorteil der Effizienz. Er ist schnell und verbraucht wenige Extradaten.

Es gibt ebenfalls starke asymmetrische Verschlüsselungsverfahren wie RSA und ElGamal. Wenn jeder seinen privaten Schlüssel bei sich geheim hält, dann ist dieser Protokolltyp nach unserem heutigen Wissensstand auch nicht zu knacken. Er hat gegenüber symmetrischen Verfahren den großen Vorteil, dass Alice und Bob sich vorher nicht treffen müssen, um einen gemeinsamen Schlüssel auszutauschen. Er ist daher besonders für offene Netze wie das World Wide Web geeignet, in dem sich die Partner vorher nicht treffen und doch vertraulich miteinander kommunizieren wollen.

Schwächen

Das Problem symmetrischer Verschlüsselungsprotokolle liegt darin, dass beide Partner über denselben Schlüssel verfügen. Eine Schwäche liegt darin, dass sie diesen erst einmal vorab und sicher vereinbaren müssen. Auch danach ist noch nicht alles in Butter. So lange sie sich vertrauen und beide den gemeinsamen Schlüssel geheim halten, ist es zwar gut. Sobald aber einer den Schlüssel weitergibt, dann kann der andere sich nicht mehr darauf verlassen, dass er noch den richtigen Kommunikationspartner hat. Und selbst wenn kein Schlüssel weiter gegeben wurde, könnten Alice und Bob, wenn sie sich uneins sind, nach außen hin (also zum Beispiel gegenüber einem Gericht) nicht nachweisen, dass nur sie miteinander geredet haben. Einen Einwand von Bob, Alice habe ihren gemeinsamen Schlüssel an andere weitergeben und deshalb sei es nicht erwiesen, dass mit diesem Schlüssel verschlüsselte Versprechen wirklich von ihm stammten, kann Alice dem Gericht gegenüber nicht entkräften, sondern nur ihr Wohlverhalten beteuern.

Das ist mit asymmetrischen Verfahren besser. Alice kann den privaten Schlüssel von Bob nicht weiter gegeben haben, da sie ihn nie zur Verfügung hatte. Der Nachteil der asymmetrischen Verfahren ist dagegen ihre Langsamkeit. Es dauert erheblich länger, Nachrichten ausreichend sicher asymmetrisch zu ver- und entschlüsseln, als die besten symmetrischen Verfahren.

Ein Man-in-the-middle-Angriff, in dem ein Angreifer gegenüber Bob vorgibt, Alice zu sein, und gegenüber Alice, er sei Bob, wobei er in Wirklichkeit alle Nachrichten zwischen den Beiden nach Belieben mitliest und modifiziert, ist in der heutigen Form des X.509-Protokolls noch nicht gefunden worden. Dafür müssen aber die eingesetzten öffentlichen Schlüssel auf vertrauenswürdige Weise zertifiziert sein, und im Laufe des Protokolls müssen die Zertifikate überprüft werden.

2. SSL (Secure Sockets Layer), das Sicherheitsprotokoll des World Wide Web

Anwendung

Anwendungen im Internet, wie E-Mail, Filetransfer (Übertragung von Dateien) und das World Wide Web, nutzen das standardisierte Verfahren TCP/IP („Transmission Control Protocol“, „Internet Protocol“) zum Transport ihrer Anwendungsdaten. TCP-Transportdaten sind unverschlüsselt. Also kann sie jeder, der an irgendeiner Stelle Zugang zu dem Transportweg hat, unbemerkt mitlesen und sogar konsistent verändern. Zugang zum Transportnetz hat aber zum Beispiel jeder Router-Administrator. Und da zwischen Sender- und Empfangsrouter im Internet durchschnittlich 20-30 Zwischenrouter liegen, gibt es sehr viele, die unberechtigt zugreifen könnten. Die Lösung liegt darin, dass ein Anwender seine Daten nur verschlüsselt auf den Transportweg gibt. Allerdings braucht der vorgesehene Empfänger dann die richtigen

Entschlüsselungsschlüssel. Das nennt man Ende-zu-Ende-Verschlüsselung zwischen Sender und Empfänger.

Das SSL ist eine Funktionsschicht zum verschlüsselten Austausch von Anwendungsdaten über das TCP-Transportsystem. Um die Anwendungen bei der Verschlüsselung zu entlasten, bietet es eine einheitliche Schnittstelle zwischen dem Anwendungssystem und dem Transportsystem. Statt also den Anschluss an das Transportsystem direkt zu programmieren, programmiert der Anwendungsentwickler den Anschluss an das SSL. Die Programmanschlüsse von SSL enthalten dabei eine ganze Reihe starker Verschlüsselungsfunktionen.

Typische Nutzer von SSLsind Webbrowser und Web-Server. Das Protokoll **https** des World Wide Web ist das um den SSL-Service erweiterte Protokoll http. Die Nutzung von SSL im World Wide Web wird durch ein Schlosssymbol im Browserfenster angezeigt, zum Beispiel beim Homebanking und in Online-Shops.

Ziel von SSL

Das Ziel von SSL ist ein sicherer Transportkanal zwischen Anwendern des Internet. „Sicher“ bedeutet, dass die Vertraulichkeit der Daten, die Unverletztheit der Daten und die Authentizität der Kommunikationspartner gewährleistet werden. Insbesondere signalisiert der SSL-Anschluss eines Browsers, dass ein Web-Server, der SSL unterstützt, der richtige ist, dass alle Daten zwischen Browser und Server verschlüsselt sind und auf dem Transportweg nicht verändert wurden.

Die Absicherung soll interoperabel zwischen allen Browser- und Server-Typen funktionieren und effizient sein. Die Schnittstelle zur Integration von SSL in die Anwendungen ist durch einen internationalen Standard offen gelegt.

Technische Grundlagen

Das SSL ist ein Anwendungsprotokoll des Internet. Es liegt zwischen der Transportschnittstelle und den eigentlichen Anwendungen. Zum Datentransport verwendet es die Internetprotokolle TCP und IP. Anwendungsprogrammierer nutzen die Programmierschnittstelle von SSL aus einem der zahlreichen SSL-Produkte statt die des TCP und bekommen dafür den abgesicherten Datentransport geliefert.

Eingesetzte Algorithmen

Im SSL-Protokoll verabreden die Anwendungspartner, welche Verschlüsselungsverfahren sie einsetzen wollen, sie tauschen über ein asymmetrisches Verschlüsselungsverfahren symmetrische Schlüssel aus und verschlüsseln anschließend ihre Daten mit diesen symmetrischen Schlüsseln. In

der Regel kommen RSA oder Diffie-Hellman für die asymmetrische Phase, und Triple-DES, AES oder IDEA für die symmetrische Phase zum Einsatz.

Ablauf

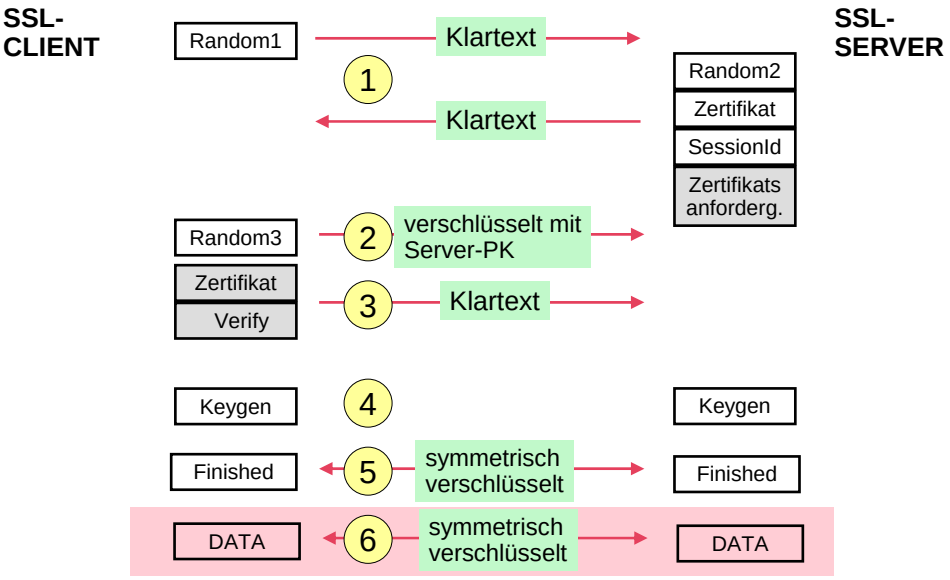

Abb. 2.1: SSL-„Handshake“-Protokoll

Schritt 1: Das SSL-Verfahren kann von jedem der beiden Anwendungspartner, im Falle des WWW also sowohl vom Browser als auch vom Server, angestoßen werden. Der Server würde dem Browser eine Aufforderung schicken, mit einem so genannten „Client-Hello“ zu beginnen. Der Browser würde gleich mit einem „Client-Hello“ loslegen. Die „Client-Hello“-Nachricht würde vom Server jedenfalls mit einem „Server-Hello“ beantwortet werden. Die „Hello“-Phase, in Abb. 2.1 als Schritt 1 dargestellt, eröffnet das so genannte „SSL-Handshake-Protokoll“, in dessen Verlauf der Server sich gegenüber dem Browser authentifiziert und beide Partner die Verschlüsselungsverfahren und die symmetrischen Schlüssel für die spätere Datenphase (Schritt 6) austauschen.

Die Verabredung darüber, welche Verschlüsselungsalgorithmen in welcher Schlüssellänge eingesetzt werden, findet bereits in der „Hello“-Phase statt: Der Browser sendet dem Server in seiner „Client-Hello“-Nachricht eine Liste kryptographischer Algorithmen und Schlüssellängen, die er selbst beherrscht. Der Server wählt die Verfahren aus, die er ausführen kann und sendet diese Auswahl zurück, oder, falls er kein angebotenes Verfahren kennt, antwortet er mit einer „Handshake Failure“-Nachricht, die die Kommunikation zwischen Browser und Server beenden würde. Typischerweise wird in dieser Phase das symmetrische Verfahren AES mit einer als sicher geltenden Schlüssellänge von 256 Bits ausgewählt, das heutzutage alle Server und Browser

beherrschen. → Man sollte Web-Server so einstellen, dass sie keine schwachen Verfahren akzeptieren!

Im unmittelbaren Anschluss an sein „Server-Hello“ sendet der Server ein Zertifikat seines öffentlichen Schlüssels, der für die nun folgende asymmetrische Phase gebraucht wird. Falls er eine Authentifizierung des Clients wünscht, fügt er eine Aufforderung an den Client hinzu, seinerseits auch ein Zertifikat zu schicken. Diese Nachricht zählen wir in Abb.2.1 ebenfalls noch zur Eröffnungsphase von Schritt 1.

Und schließlich werden in der „Hello“-Phase in Schritt 1 zwei Zufallszahlen ausgetauscht, von denen jeder der beiden Partner einen erzeugt und dem anderen mitteilt.

Schritt 2: Während die beiden ersten Zufallszahlen im Klartext ausgetauscht werden, teilt der Browser eine dritte Zufallszahl, die er erzeugt, dem Server verschlüsselt mit, und zwar verschlüsselt mit dem öffentlichen Schlüssel, den er dem Zertifikat des Servers entnommen hat. Der Browser kann also sicher sein, dass nur der berechtigte Inhaber des zugehörigen privaten Schlüssels, also der „richtige“, im Zertifikat bestätigte Server, die dritte Zufallszahl bekommt.

Schritt 3 wird nicht immer ausgeführt. Er dient der Authentifizierung des Browsers gegenüber dem Server. Da nur die wenigsten Browser über aussagekräftige Zertifikate verfügen, verzichtet der Server in der Regel darauf, in seiner „Hello“-Nachricht ein Browser-Zertifikat anzufordern. In höherwertigen Anwendungen allerdings, zum Beispiel in einer Homebanking-Anwendung nach dem Standard HBCI, sind Browser mit einem Bankenzertifikat ausgestattet, das dann in einem Schritt 3 des SSL-Handshake zum Einsatz kommt.

In Schritt 4 erzeugen nun Browser und Server, jeder auf seiner Seite, nach einem gemeinsamen, von SSL für alle Teilnehmer festgelegten Verfahren, einen gemeinsamen Satz symmetrischer Schlüssel. Sie kommen zum selben Ergebnis, weil sie dieselben Zufallszahlen als Parameter einsetzen. In die Schlüsselerzeugung gehen alle drei Zufallszahlen ein, so dass erstens beide Partner an der Entwicklung der Schlüssel beteiligt sind, und zweitens niemand anderes als nur diese zwei Partner den Schlüsselsatz erzeugen können, denn die dritte Zufallszahl wurde ja vom Browser erzeugt und dem Server asymmetrisch verschlüsselt zugestellt.

Und zwar werden sechs symmetrische Schlüssel erzeugt: Jeder Partner kennt alle sechs Schlüssel. Die ersten drei sind dem Browser zugeordnet und heißen ClientWriteKey, ClientWriteMAC und ClientIV. Die nächsten drei sind dem Server zugeordnet und heißen ServerWriteKey, ServerWriteMAC und ServerIV. ClientIV und Server IV sind so genannte Initialisierungsvektoren, die den Gebrauch des symmetrischen Verschlüsselungsverfahrens jeweils auf ihrer Seite steuern. Mit dem ClientWriteKey verschlüsselt der Client seine Nachrichten an den Server, und der Server nutzt diesen, um Nachrichten vom Browser zu entschlüsseln. Der ServerWriteKey wird analog verwendet. Mit dem ClientWriteMAC erzeugt der Client zu jeder Nachricht einen so genannten

Message Authentication Code (MAC), das ist ein Hashwert der Nachricht, in den eine fortlaufende Nummer, die Nachrichtenlänge, der Nachrichteninhalt und eben der ClientWriteMAC eingehen. Der ServerWriteMAC wird analog verwendet. Das wirkt gegen den Versuch Nachrichten zu löschen oder deren Reihenfolge zu ändern.

Den Schritten 1-4 des „SSL-Handshake“ folgt ein Zwischenschritt, der in der Abb. 2.1 nicht dargestellt ist: Client und Server tauschen nämlich die so genannten „Change Cipher Specs“-Nachrichten aus und erklären damit, dass der Schlüsselaustausch und die Authentifizierung erfolgreich waren.

Schritt 5: Die im Anschluss hieran ausgetauschten „Finished“-Nachrichten sind die ersten Nachrichten, die mit Hilfe der ausgehandelten Parameter verschlüsselt werden. Die Empfänger müssen jeweils nachprüfen, ob der Inhalt korrekt ist. Schritt 5 ist gewissermaßen der Probelauf für die Datenphase. Denn an dieser Stelle sind ja die Verschlüsselungsverfahren verabredet, niemand anderes als die beiden Partner kennen die Schlüssel.

Schritt 6: Die „Handshake“-Phase ist nun abgeschlossen und die verabredeten Verfahren und Schlüssel können nun sicher in der folgenden Datenphase eingesetzt werden.

Das SSL-Programm ist entweder in den Browser integriert, oder es ist in einen externen Prozess in der lokalen Umgebung des Browsers ausgelagert, über den der Browser alle Kommunikation ins Web leitet. Ein extern ausgelagerter SSL-Prozess wird „SSL-Proxy“ genannt. Der Nutzer des Browsers muss darauf vertrauen, dass der SSL-Proxy ordentlich programmiert ist. In den Verbindungseinstellungen eines Browsers kann ein Nutzer einstellen, ob und über welchen SSL-Proxy sein Browser ins Internet gehen soll.

Eine gut lesbare Darstellung des SSL-Protokolls findet sich bei [Esslinger, Müller 1997].

Stärken

Der große Vorzug des SSL-Protokolls liegt darin, dass Anwendungsdaten ohne weiteres Zutun der Nutzer von den Anwendungsinstanzen automatisch verschlüsselt werden und mit einem Unverletztheitskennzeichen versehen sind. Das erkennt der Nutzer daran, dass eine Web-Verbindung am Anfang ihrer URL-Adresse den Protokollnamen „https“ statt „http“ erhält. Alle Browser zeigen außerdem bei der Adresse und im Rahmen des Browserfensters ein deutlich sichtbares Symbol eines Bügelschlosses. Das zeigt dem Nutzer an, dass die ausgetauschten Daten nicht von dritter Seite mitgelesen, verändert, gelöscht oder in der Reihenfolge verändert werden können.

Optional können sich beide Partner auch noch sicher authentifizieren. Dazu braucht der betreffende Partner (im Web ist das immer der Server und gelegentlich zusätzlich auch der Browser) nur anfänglich sein Zertifikat einzustellen, und dann läuft die Authentifizierung jedes Mal automatisch

ab. Der Nutzer eines Browsers, bzw. der Administrator eines Servers kann über die Sicherheitseinstellungen eigene Zertifikate, die er zuvor von einer Zertifizierungsstelle erworben hat, importieren.

Server und Client können sich dann mit SSL darauf verlassen, dass sie wissen, mit wem sie es auf der anderen Seite zu tun haben und dass niemand anderes in ihre Kommunikation hineinlesen oder sie verfälschen kann. Allerdings sind die Voraussetzungen der Sicherheit einzuhalten.

Im „SSL-Handshake“-Protokoll werden die kryptographischen Parameter ausgetauscht und optional die Kommunikationspartner authentifiziert. Sofern die Authentifizierung ordentlich durchgeführt wird, kann eine Maskerade durch einen Betrüger ausgeschlossen werden. Ebenfalls verhindert würde ein Angriff eines „Man-in-the-middle“, der sich zwischen zwei Partner einschiebt und jedem vortäuscht, der jeweils andere zu sein. Die bei der Authentifizierung ausgetauschten Zertifikate müssen gültig sein und vom Empfänger auf Echtheit überprüft werden. Um das Server-Zertifikat zu prüfen, klickt der Nutzer im Browserfenster das Schloss an und erhält die in der Abb. 2. dargestellte Seiteninformation. Bei einem weiteren Klick auf den „Sicherheits“-Reiter wird das Zertifikat gezeigt. Es ist wichtig, dass der Nutzer die Organisation kennt, die als Zertifikatausstellter bezeichnet ist, und ihr vertraut.

Abb. 2.2: SSL-Sicherheitsinformation für den Browser

Allerdings kann auch eine unzertifizierte SSL-Verbindung nützlich sein. Jedenfalls ist auch in diesem Fall die Kommunikation verschlüsselt und mit dem Integritätsschutz versehen. Und auch bei anonymem Schlüsselaustausch werden die übermittelten Zufallszahlen in den erzeugten Schlüssel integriert und so eine Replay-Attacke vermieden. Die produzierten „Finished“-Nachrichten bestätigen beiden Seiten die Nutzung der zuvor verabredeten Verfahren.

Schwächen

Das SSL-Verfahren hat zwei Schwächen und eine Einschränkung, über die sich ihre Nutzer klar sein müssen. Die eine Schwäche ist die Unsicherheit darüber, wer der Partner auf der anderen Seite wirklich ist, wenn die Authentifizierung entweder nicht stattfindet (was das SSL-Protokoll zulässt) oder nicht richtig ausgeführt wird. Ein Web-Server, der sich mit einem gefälschten, ungültigen oder abgelaufenen Zertifikat bei einem Browser zurückmeldet, erzeugt beim Browser eine Warnmeldung. Viele Nutzer von Web-Browsern können mit einer solchen Meldung nichts anfangen und ignorieren sie. Sie wiegen sich dann in der falschen Sicherheit einer mit einem hübschen Bügelschloss markierten SSL-Verbindung, kommunizieren auch tatsächlich verschlüsselt, allerdings mit einem falschen Partner, möglicherweise mit einem direkten Angreifer, der ihnen nun alle möglichen vertraulichen Informationen absaugt, wie zum Beispiel PINs und TANs für Homebanking.

Eine zweite Schwäche liegt darin, dass die automatisierte Sicherheit von SSL nur dann gewährleistet ist, wenn die Client- und Server-Komponenten unmanipuliert funktionieren. Ein so genannter SSL-Trojaner könnte alle Sicherheitsfunktionen ausschalten, indem es alle Warnungen des Browsers unterdrückt und dem Nutzer dadurch suggeriert, dass noch alles in Ordnung sei. In diesem Fall glaubt der Nutzer, mit SSL sicher zu kommunizieren, spielt aber in Wirklichkeit direkt einem Angreifer in die Hand. Also hilft SSL bei einem verseuchten Betriebssystem nicht weiter.
Daraus ergibt sich, dass Nutzer von SSL diese beiden Dinge sicherstellen muss: Erstens seine Umgebung viren- und trojanerfrei zu halten, und zweitens genau auf die Zertifikate des Partners zu achten. Dazu klickt ein Nutzer auf das Bügelschloss und liest das angezeigte Server-Zertifikat. Er sollte den Namen des Zertifikatausstellers kennen und ihm vertrauen. Wenn das nicht der Fall ist, ist es ratsam, die Web-Verbindung nicht weiter zu verfolgen.

Ein dritter Punkt ist nicht eigentlich eine Schwäche von SSL, sondern nur eine Einschränkung seiner Wirkung: Der Integritätsschutz von SSL wirkt nur für die Dauer der SSL-Verbindung. Es werden nämlich nicht die Anwendungsdaten im Ganzen geschützt, sondern nur ihre Fragmente im Transport-System. Ein elektronischer Vertrag also, der über eine SSL-Verbindung ausgehandelt wird, ist erst dann gegen spätere Abstreitung gesichert, wenn er insgesamt mit einer digitalen Signatur versehen ist, die auf Anwendungsebene ausgestellt, verifiziert und aufbewahrt wird. Denn die SSL-Signaturen sind nach Ende einer SSL-Verbindung alle verschwunden. Wir sagen daher, dass SSL ein flüchtiges Sicherungsverfahren darstellt.

3. Mobilfunk GSM auf der Luftschnittstelle

Anwendung

Überall, wo sich Menschen befinden, vor allem in Siedlungsgebieten und entlang Straßen, haben die Betreiber von Mobilfunknetzen Basisstationen errichtet zur Ortung von Mobiltelefonen und

zum Austausch von Gesprächsdaten zwischen Mobiltelefonen und dem Telefonnetz über die Luft. Die Luft als Übertragungsmedium ist aber unbeschränkt zugänglich. Um zu verhindern, dass Gesprächsdaten zwischen Handy und Basisstationen abgehört werden, werden sie verschlüsselt.

Ziel der GSM-Verschlüsselung

Handys heißen in der Fachsprache „Mobilstationen (MS)“. Um zu prüfen, ob eine Mobilstation berechtigt ist, das Mobilfunknetz zu nutzen, muss sie sich dem Netzbetreiber gegenüber authentifizieren. Dafür wird ein symmetrisches Verschlüsselungsverfahren verwendet, dessen geheimer Schlüssel sowohl der SIM-Karte in der Mobilstation als auch dem Netzbetreiber bekannt ist. Vor Auslieferung der SIM-Karte an den Kunden wird dieser geheime Schlüssel vom Betreiber des Netzes auf die SIM-Karte gespeichert.

Ist die Mobilstation erst einmal authentifiziert, kann die Kommunikation zwischen GSM-Teilnehmer und Basis-Station (und damit die eigentliche Verwendung des Netzes) erfolgen. Damit dieser Verkehr nicht abgehört werden kann, findet zusätzlich zur Anmeldung eine Verschlüsselung der Kommunikation statt. Dafür wird ein weiteres symmetrisches Verfahren eingesetzt.

Die persönliche Geheimzahl PIN (Personal Identification Number) einer SIM-Karte hat nichts mit der Verschlüsselung der Daten zu tun. Sie dient allein der Freischaltung eines Handys gegenüber seinem berechtigten Benutzer.

Technische Grundlagen

Das GSM-Netz ist in Zellen unterteilt. Eine Zelle ist die kleinste Einheit, in der mobile Geräte kommunizieren können. Das GSM-Netz besteht aus drei Teilsystemen:

- Das Funkteilsystem (Radio Subsystem, RSS) ist für die Funkübertragung zuständig. Hierzu gehören die Mobilstation (das Handy) und das Basisstation-Teilsystem (Base Station Subsystem, BSS), welches die Funkübertragung zwischen Mobilstation und dem Netzbetreiber übernimmt. Das BSS besteht aus der BTS (Base Transceiver Station, das sind z.B. die Antennen) und dem BSC (Base Station Controller), welcher als Kontrolleinheit agiert. Ein BSC verwaltet 10-100 BTS und ist über ein MSC (Mobile Switching Center) mit dem Vermittlungsteilsystem verbunden.

- Das Vermittlungsteilsystem (Network and Switching Subsystem, NSS) beinhaltet erstens das MSC (Mobile Switching Center), welches die zentrale Kontrolle und die Verwaltung der Verbindungen übernimmt. Zweitens enthält es das HLR (Home Location Register), in dem Daten aller registrierten Teilnehmer eines Betreibers gespeichert sind und das VLR (Visitor Location Register), in dem Daten zu den Kunden gespeichert sind, die sich nur temporär in diesem Netz aufhalten.

- Das Betreiberteilsystem (Operation Subsystem, OSS) übernimmt die Betriebs- und Wartungsaufgaben und verwaltet die Daten aller Mobilegeräte eines Netzbetreibers. Hier ist auch das AC (Authentification Center) angesiedelt, in dem die Verschlüsselungsalgorithmen für die Netzauthentifikation und für die Verschlüsselung der Daten gespeichert sind.

Jedes mobile Endgerät hat eine international eindeutige Seriennummer, die so genannte IMEI (International Mobile Equipment Identity). Zusätzlich enthält es eine SIM-Karte. Diese verwaltet eine Kundennummer (die so genannte IMSI, International Mobile Subscriber Identity), eine PIN (Personal Identification Number), einen PUK (Personal Unblocking Key), die teilnehmerbezogene Rufnummer sowie die Algorithmen und Schlüssel für die spätere Authentifizierung und Datenverschlüsselung.

Eingesetzte Algorithmen

Für die Authentifizierung der SIM-Karte im Handy gegenüber dem Netzbetreiber wird ein symmetrisches Verfahren namens A3 verwendet. Die Verschlüsselung und Entschlüsselung der Daten zwischen Handy und Basisstation durch die Luft erfolgt durch einen anderen Algorithmus namens A5. Der dafür erforderliche Schlüssel wird mit Hilfe eines weiteren Algorithmus namens A8 erzeugt.

Alle Algorithmen sind geheim, netzbetreiberabhängig und nicht standardisiert. Allerdings ist seit 1998 eine Implementierung einer Kombination der Algorithmen A3 und A8 mit dem Namens COMP128 bekannt geworden. Diese wird von vielen Netzbetreibern eingesetzt.

Ablauf

Authentifizierung:

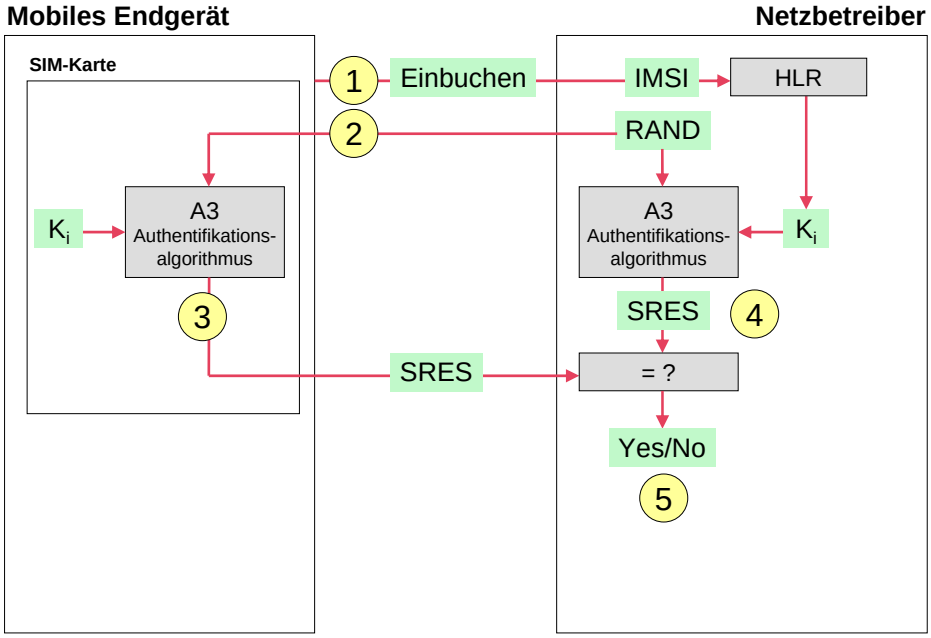

Abb. 3.1: Authentifizierung eines Endgerätes im GSM-Netz

Im GSM-Netz erfolgt die Authentifizierung mit dem folgenden Challenge-Response-Verfahren:

Schritt 1: Beim Anmelden wird die IMSI von der Mobilstation unverschlüsselt an den Netzbetreibern übertragen.

Schritt 2: Daraufhin wird beim Betreiber eine 128-Bit lange Zufallszahl RAND erzeugt und diese erstens der Mobilstation als jedes Mal frische Herausforderung („Challenge“) geschickt und zweitens für eine eigene Berechnung (s.u. Schritt 4) aufgehoben.

Schritt 3: In der Mobilstation wird die Zufallszahl RAND und ein teilnehmerbezogener geheimer Schlüssel K_i (128-Bit), der auf der SIM-Karte gespeichert ist, mit Hilfe des Algorithmus A3 verknüpft. Dieser Algorithmus ist auf der SIM-Karte gespeichert und wird innerhalb der SIM-Karte ausgeführt. Das Ergebnis ist eine 32-Bit-Zahl SRES, die als „Response“ an das Netz zurückgeschickt wird.

Schritt 4: Beim Netzbetreiber wird aus dem HLR der teilnehmerbezogene geheime Schlüssel K_i abgerufen. Dieser entspricht genau dem in der Karte gespeicherten Schlüssel K_i. Auch beim Betreiber wird dieser mit der Zufallszahl RAND und dem Algorithmus A3 verknüpft. Das beim Betreiber berechnete Ergebnis wird mit dem vom mobilen Endgerät gesendeten SRES-Wert verglichen.

Schritt 5: Ist die Überreinstimmung positiv, so ist die Mobilstation authentifiziert und der Teilnehmer kann die Dienste des Funknetzes nutzen.

Verschlüsselung:

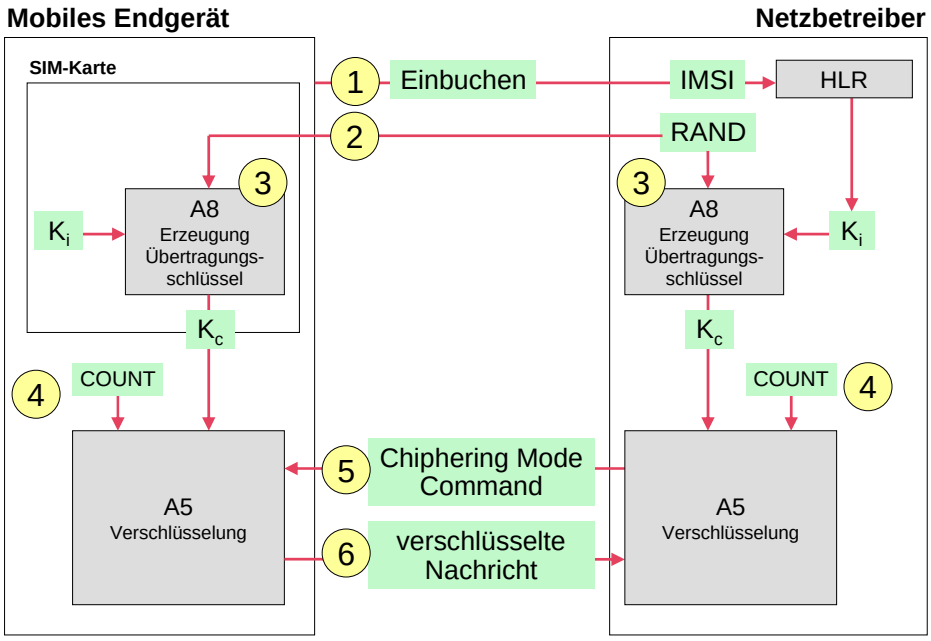

Abb. 3.2: Verschlüsselung der Übertragung im GSM-Netz

Alle teilnehmerbezogenen Daten zwischen Mobilstation und Netzbetreiber werden im GSM-Netz verschlüsselt übertragen. Der dafür benötigte Schlüssel wird folgendermaßen erzeugt:

Aus der vorher gegangenen Authentifizierung (s. o. die Schritte 1 und 2) stehen auf beiden Seiten (Betreiber und Mobilstation) noch die Werte K_i und RAND zur Verfügung.

Schritt 3: Aus diesen beiden Werten wird mit einem weiteren Algorithmus A8 ein 64-Bit Chiffrierschlüssel K_c erzeugt. Diese Berechnung wird sowohl in der SIM-Karte als auch beim Netzbetreiber durchgeführt. Der erzeugte Schlüssel K_c wird dann bei der eigentlichen Verschlüsselung der übertragenen Daten verwendet.

Schritt 4: Um sicherzustellen, dass Mobilstation und Netzbetreiber zu jedem Zeitpunkt den gleichen Schlüssel K_c verwenden, wird dieser bei der Erzeugung mit einer Schlüsselnummer COUNT versehen.

Schritt 5: Der Netzbetreiber initiiert die verschlüsselte Übertragung mit einem so genannten „Ciphering Mode Command“.

Schritt 6: Ab diesem Zeitpunkt werden unter Berücksichtigung des Schlüssels K_c und der Schlüsselnummer COUNT die übertragenen Daten mit einem Stromverschlüsselungsverfahren (Algorithmus A5) verschlüsselt.

Stärken

Da die genannten Verschlüsselungsalgorithmen (A3, A5 und A8) netzbetreiberabhängig und teilweise geheim sind, ist es nicht einfach diese zu umgehen. Die verwendeten symmetrischen Algorithmen lassen sich im Gegensatz zur asymmetrischen Verfahren effizient implementieren und sind daher für die Umsetzung auf mobilen Endgeräten mit geringer Rechenkapazität geeignet.

Die Verwendung netzbetreiberspezifischer Algorithmen ist auch bei weltweiter Nutzung eines Endgerätes (Roaming) möglich, da diese Verfahren nur in der vom Netzbetreiber ausgegebenen SIM-Karte und im AC des Heimatnetz gespeichert und ausgeführt werden. Die Systeme eines besuchten Fremdnetzes brauchen keine Kenntnis davon zu haben.

Schwächen

Eine Schwäche der GSM-Authentifizierung liegt in der Einseitigkeit. Nur die Mobilstation authentifiziert sich gegenüber dem Netz, nicht umgekehrt. Dadurch ist z. B. mithilfe eines so genannten IMSI-Catchers ein Man-in-the-middle-Angriff möglich, da sich ein Mobiltelefon automatisch mit dem stärksten Sender verbindet. Sendet der Catcher ein entsprechend starkes Signal aus, dann geht die Mobilstation davon aus, dass dies die Gegenstelle des Netzbetreibers ist. Durch die Vortäuschung einer schlechten Verbindung bringt der Angreifer die Mobilstation dazu, das Verschlüsselungsverfahren temporär abzuschalten. Auf diese Weise kann der Angreifer die Kommunikation mithören.

Ein weiterer Schwachpunkt ist, dass die Sicherheit der verwendeten Verschlüsselungsalgorithmen auf deren Geheimhaltung und nicht auf starken Schlüsseln basiert. Dies widerspricht einer Anforderung für moderne Kryptoverfahren, dem Kerckhoffs-Prinzip: „Die Sicherheit eines kryptographischen Systems darf nur von der Wahl des Schlüssels und nicht von der Geheimhaltung des Algorithmus abhängen”.

Als Folge davon ist es im Falle eines Bekanntwerdens der o. g. Algorithmen sehr aufwendig, diese flächendeckend zu aktualisieren. Von einer solchen Aktualisierung sind potentiell alle SIM-Karten des Netzbetreibers betroffen.

Eine Vertraulichkeitslücke bilden die SMS-Nachrichten, die unverschlüsselt über den so genannten Kontrollkanal und nicht verschlüsselt über den Datenkanal, über den die Telefonanrufe laufen, übertragen werden.

4. Geldauszahlungsautomat und ec-Cash

Anwendung

Die ec-Karte (Debitkarte) ist ein alltägliches Instrument zum elektronischen Bezahlen an einem Händler-Terminal im Laden und zum Abheben von Bargeld von einem Geldauszahlungsautomaten. Die ec-Karte funktioniert in Europa und teilweise auch weltweit grenzüberschreitend. Die Abkürzung ec stand ursprünglich für „Eurocheque“. Seit 1998 steht diese Abkürzung für „Electronic Cash“ und damit für einen bestimmten elektronischen Bezahlvorgang mit Karte und PIN an einem spezifischen Händler-Terminal, zum Beispiel an der Ladenkasse oder an der Tankstelle. Die ec-Karten sind mit einem Magnetstreifen und seit 1996 von einigen Kreditinstituten auch mit einem Chip versehen.

Je nach Anwendung (z. B. beim elektronischen Bezahlen beim Händler oder beim Abheben eines Geldbetrags von einem Automaten) werden Informationen entweder vom Magnetstreifen oder vom Chip benötigt. Im folgenden Beispiel wird die Verschlüsselung der Magnetstreifendaten einer ec-Karte zur Berechnung der PIN (Persönliche Identifikationsnummer) betrachtet.

An dieser Stelle soll explizit erwähnt werden, dass die folgenden Ausführungen auf dem alten PIN-Verfahren basieren. Dieses Verfahren wies schwerwiegende Schwächen auf (vgl. 0). Deshalb wurde ab 1997 ein neues ec-PIN-Verfahren eingeführt, das Triple-DES statt DES nutzt und die Prüfung der PIN-Eingabe immer online zum Kartenherausgeber oder dessen Bevollmächtigten vornimmt. Es handelt sich dabei um ein nicht veröffentlichtes Verfahren, welches im Gegensatz zu dem alten PIN-Verfahren nicht einheitlich von den Banken implementiert wurde. Aussagen über das neue Verfahren werden im Folgenden gesondert gekennzeichnet.

Ziel der Verschlüsselung der Magnetstreifendaten

Durch die PIN-Prüfung soll erreicht werden, dass der Karteninhaber als authentisch verifiziert wird. Die PIN liegt nämlich im Gedächtnis des Karteninhabers, und hoffentlich auch nur dort. Durch die Verschlüsselung der Datenübertragung soll gewährleistet werden, dass dabei geheime Daten wie die eingegebene PIN nicht in die Hände Dritter gelangen können. Es gibt prinzipiell zwei Varianten für die PIN-Erzeugung und –Prüfung, in denen die Verschlüsselung eine Rolle spielt: (1.) Die Kartendaten werden verschlüsselt, aus diesem Chiffrat wird die PIN abgeleitet. Beim Geldautomaten wird dann die PIN nachgerechnet und mit der eingegebenen PIN verglichen. Das ist das alte ec-PIN-Verfahren. Die neuen PIN-Verfahren basieren auf der folgenden Variante: (2.) Die PIN wird zufällig berechnet und dem Kunden zugewiesen. Bei der Prüfung wird ein Abgleich mit der zentralen Datenbank durchgeführt, dabei ist die Verbindung zu der Bank verschlüsselt. Die Verifizierung erfolgt anhand von Prüfwerten, die in einer Datenbank beim Kartenherausgeber abgelegt sind.

Für das alte ec-PIN-Verfahren gilt konkret folgendes: Erstens ist die PIN das Ergebnis der Verschlüsselung der Magnetstreifendaten mit einem geheimen institutsspezifischen Bankenschlüssel. Zweitens wird der Datenaustausch zwischen Geldauszahlungsautomat bzw. Händler-Terminal einerseits und Banken-Server andererseits zum Schutz vor Lauschangriffen verschlüsselt.

Damit nicht nur die kartenausgebende Bank sondern auch assoziierte Geldinstitute eine PIN berechnen können, (und zum verbesserten Schutz vor der Analyse der geheimen Bankenschlüssel) werden neben dem Institutsschlüssel der kartenausgebenden Bank mehrere Schlüssel, so genannte Pool-Schlüssel, eingesetzt.

Technische Grundlagen

Im Bereich des Zahlungsverkehrs gibt es drei Arten von Karten:

- Kreditkarten – Dies sind meistens Magnetstreifenkarten ohne Chip. Man kann mit der Karte einen Betrag bezahlen und dieser wird zu einem späteren Zeitpunkt vom Konto abgebucht.

- Debitkarten – Diese gibt es sowohl als reine Magnetstreifenkarten als auch in Form einer Kombination von Magnetstreifen und Chip. Der Betrag wird unmittelbar nach der Zahlung vom Konto abgebucht. Die ec-Karte ist eine Debitkarte.

- Karten für eine elektronische Geldbörse – Es wird auf dem Chip mittels eines Aufladevorgangs ein Betrag gespeichert, der später für eine Zahlung genutzt werden kann, hier wird der Betrag also vor der Zahlung auf die Karte geladen. Die GeldKarte ist von diesem Typ.

Damit sind drei Zahlungsverfahren möglich:

- Electronic Cash – Die benötigten Daten werden vom Kartenterminal aus dem Magnetstreifen ausgelesen. Das Kartenterminal baut eine Verbindung zu einem Banksystem auf, bei dem die PIN, vorliegende Sperren und die Kontodeckung überprüft werden.

- Electronic Cash offline – Hier wird ein Limit im Chip gespeichert. Die Zahlung wird ebenfalls durch Eingabe der Karten-PIN legitimiert. Allerdings wird die PIN offline durch den Kartenchip geprüft. Ist das Limit nicht überschritten und ist die PIN richtig, dann wird die Transaktion ohne zusätzlichen Verbindungsaufbau zur Bank autorisiert und das Limit um den Zahlungsbetrag reduziert.

- POZ (Point of Sale ohne Zahlungsgarantie) und ELV (Elektronisches Lastschriftverfahren) – Hier muss im Gegensatz zum Electronic-Cash-Verfahren zur Autorisierung keine PIN

eingegeben werden. Stattdessen bestätigt der Kunde einen ausgedruckten Lastschriftbeleg mit seiner Unterschrift. POZ wird Ende 2006 eingestellt.

Eingesetzte Algorithmen

Für die Karteninhaberauthentifizierung im alten ec-PIN-Verfahren wird das symmetrische Verschlüsselungsverfahren DES mit einem Institutsschlüssel (Bankschlüssel) oder einem Pool-Schlüssel verwendet, welche im Geldausgabeautomat vorgehalten werden. Die Schlüssellänge beträgt 56 Bits.

Dasselbe Verfahren wird auch während eines Bezahlvorgangs zwischen der ec-Karte und – vermittelt über das Händler-Terminal – dem Banken-Server eingesetzt. Zur Abwehr von Lauschangriffen zwischen Händler-Terminal und Banken-Server werden die Daten ebenfalls mit DES verschlüsselt, allerdings mit einem individuellen Schlüssel, den die Bank jedem Händler-Terminal vor der Auslieferung an den Händler zuteilt.

Der reine DES-Algorithmus gilt nicht mehr als sicher, so wird im neuen ec-PIN-Verfahren der Triple-DES-Algorithmus mit einem 112-Bit-Schlüssel eingesetzt.

Ablauf

Geldausgabeautomat:

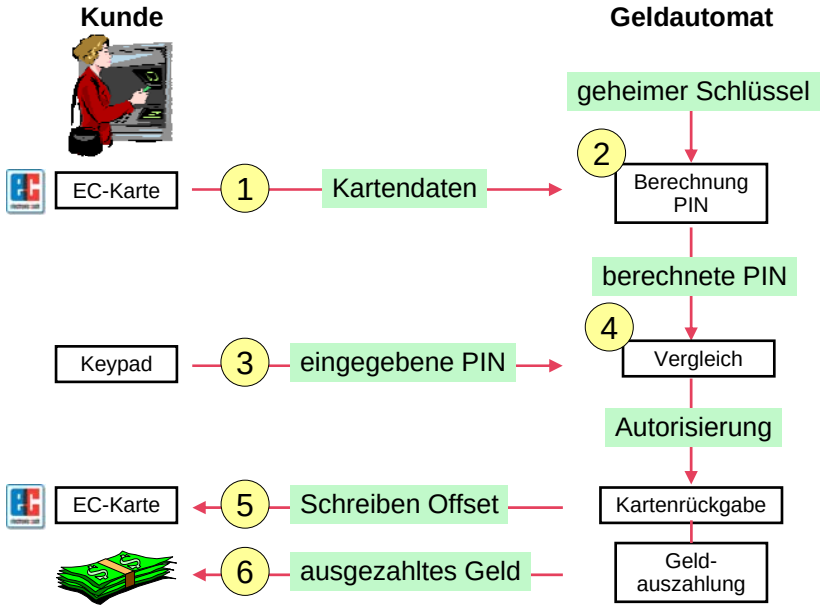

Abb. 4.1: Offline-Autorisierung am Geldautomaten (altes ec-PIN-Verfahren)

Auf dem Magnetstreifen der ec-Karte sind der Name des Karteninhabers, die Bankleitzahl, Kontonummer, eine Kartenfolgenummer sowie der Verfügungsrahmen gespeichert. Um verschiedene Schlüssel (Pool-Schlüssel) in der Weise einsetzen zu können, dass sie alle dasselbe Verschlüsselungsergebnis erzielen, werden außerdem so genannte Offset-Felder auf dem Magnetstreifen gespeichert, die bei Schlüsselwechsel auch verändert zurück geschrieben werden können. Im Geldausgabeautomaten sind mehrere geheime Pool-Schlüssel gespeichert. Ein Geldautomat kann mit seinen Schlüsseln und anhand der Magnetstreifendaten die PIN anhand der Offsets verifizieren. Diese Überprüfung läuft folgendermaßen ab:

Schritt 1: Der Kunde schiebt die Karte in den Automat und die Kartendaten werden ausgelesen.

Schritt 2: Der Bankautomat erzeugt aus der Bankleitzahl, der Kontonummer und einer Kartenfolgenummer einen Datenblock. Aus diesem Datenblock und einem geheimen Schlüssel, der im Automat gespeichert ist (geheimer Schlüssel, Bankschlüssel oder Derivation Key), wird mit Hilfe des symmetrischen DES-Verfahrens ein Ciphertext berechnet, der 64 Bits umfasst. Das entspricht sechzehn Blöcken à 4 Bits, die als hexadezimale Zeichen 0-9 und A-F dargestellt werden. Von diesen sechzehn Stellen werden die vier Stellen an den Positionen 3-6 ausgewählt, die mit Hilfe einer Abbildungsvorschrift ($0\rightarrow0$, …, $9\rightarrow9$, $A\rightarrow0$, …, $F\rightarrow5$) die PIN ergeben.

Schritt 3: Der Automat fordert den Kunden auf, seine PIN einzugeben.

Schritt 4: Die berechnete PIN (Schritt 2) und die eingegebene PIN (Schritt 3) werden verglichen.

Schritt 5: Stimmen die Werte überein, wird das Auszahlungsergebnis (Buchungsvorgang) sowie ggf. eine neue Offset-Nummer auf die Karte zurück geschrieben und die Karte wieder ausgegeben.

Schritt 6: Außerdem erfolgt die Auszahlung an den Kunden.

Stimmen in Schritt 4 die Werte nicht überrein, wird in der Regel zwei weitere Male nach der PIN gefragt. Bei wiederholten Falscheingaben wird die Karte gesperrt oder vom Automat eingezogen. Die Häufigkeit der erfolglosen PIN-Abfrage wird ebenfalls auf der Chipkarte in Schritt 5 zurück geschrieben.

Bei dem neuen ec-PIN-Verfahren ist die Entscheidung, ob eine Transaktion z.B. Geldausgabe durchgeführt wird, immer das Ergebnis einer Online-Prüfung zwischen Automat und Bank. Diese Anfrage enthält Daten wie Kontonummer, Bankleitzahl, Datum, die Nummer des Automaten, die PIN-Nummer. All diese Daten, außer der PIN, werden im Klartext an die Bank übertragen. Die Prüfung der PIN kann also nicht mehr offline erfolgen [Lochter, Schindler 2006].

Zahlungsvorgang am Händler-Terminal:

Möchte der Kunde beim Händler mit seiner Karte und dem Electronic-Cash-Verfahren bezahlen, läuft die Prozedur ganz analog ab, mit dem Unterschied, dass das Händler-Terminal die

Kartendaten verschlüsselt an den Banken-Server weiterreicht und von dort die Ergebnisse der Berechnungen, Freigaben und Rückschreibwerte geliefert bekommt.

Schritt 1: Der Kunde steckt die Karte in das Terminal und bestätigt den angezeigten Betrag. Dann fordert das Terminal den Kunden auf, die PIN einzugeben.

Schritt 2: Der Kunde gibt die angeforderte PIN über die Tastatur ein. Die PIN wird von der Tastatur unmittelbar verschlüsselt (Encrypting PIN Pad). Für die Verschlüsselung wird ein Schlüssel verwendet, der individuell dem Terminal zugeordnet ist. Außerdem liest das Terminal die Kartendaten aus und verschlüsselt diese ebenfalls mit demselben individuellen Terminalschlüssel. Die verschlüsselten Datensätze, die eingegebene PIN und die Kartendaten werden an den angeschlossenen Banken-Server (Autorisierungsstelle) übermittelt.

Schritt 3: Dort wird überprüft, ob die PIN richtig ist. Und zwar macht das die Autorisierungsstelle genauso wie der Geldauszahlungsautomat in Schritt 2 oben. Falls sie die mitgelieferte PIN als korrekt erkennt, wird weiterhin geprüft, ob diese Karte gesperrt ist und ob das Konto eine ausreichende Deckung aufweist. Falls auch das gut ausgeht, wird die Bezahlung akzeptiert und die Händlerbank zieht den Betrag von der Käuferbank ein. Andernfalls wird die Bezahlung abgewiesen.

Schritt 4: Bei positiver Rückmeldung seitens der Autorisierungsstelle erfolgt auf dem Händler-Terminal die Anzeige „Zahlung erfolgt“ und das Händler-Terminal schreibt die Rückschreibwerte Offset und Buchungsvorgang auf die Karte zurück.

Stärken

Auf der ec-Karte sind auf dem Magnetstreifen keine geheimen Informationen gespeichert. Die darin erhaltenen personenbezogenen Informationen sind dieselben, die auch auf der Karte mit Klartext angedruckt sind (Name, Kontonummer, Bankleitzahl, Kartenfolgenummer).

Entgegen anders lautenden Gerüchten ist die PIN weder im Klartext noch in verschlüsselter Form auf dem Magnetsreifen vorhanden. Die PIN wird für jede Transaktion (Geldausgabe, Bezahlvorgang Terminal) neu verifiziert. Durch ein beschreibbares Offset-Feld auf dem Magnetstreifen können mehrere Schlüssel (Poolschlüssel) zur Berechnung der PIN eingesetzt werden. Nur einer der Schlüssel ist gültig. Außerdem können die Poolschlüssel wechseln. Das erschwert die Analyse des geheimen Bankenschlüssels. Nach dem neuen Verfahren wird nicht die PIN neu nachgerechnet, sondern es werden nur Prüfsummen verifiziert (vgl. die beiden Varianten in 0).

Die ec-Karten enthalten als Schutz gegen physische Kopien ein individuelles so genanntes „Moduliertes Merkmal (MM)“, das ist eine in den Kartenkörper eingebrachte geheime

maschinenlesbare Substanz, die von einem speziellen Sensor im Kartenterminal ausgelesen wird. Dieses Merkmal wird nur in Deutschland von den Automaten überprüft.

Ein weiterer Vorteil des ec-Verfahrens liegt darin, dass seitens der Banken von Zeit zu Zeit neuere Technologien eingesetzt werden, um bekannte Sicherheitslücken zu beseitigen. Die Überprüfung der PIN findet ausschließlich beim Kartenherausgeber oder dessen Beauftragten (Autorisierungszentrale) statt.

Des Weiteren wird durch die Einführung eines Chips auf der Karte und neuere Geldautomaten zusätzlich zur Authentifizierung der Karte gegenüber dem Automat auch die Authentifizierung des Geldautomaten gegenüber der Karte möglich. Auf diese Weise kann verhindert werden, dass eine Karte Kundendaten an einen manipulierten oder nachgebauten Geldautomaten herausgibt.

Schwächen

Die beim alten ec-PIN-Verfahren verwendeten 56-Bit langen Schlüssel des klassischen DES-Verfahrens sind gegen Brute-force-Angriffe nicht mehr sicher. Es ist zweifelhaft, ob die 112-Bit langen Triple-DES-Schlüssel einem Brute-force-Angriff ausreichend lang standhalten. Krimineller PIN-Diebstahl von einer ec-Karte geschieht nach bisherigem Kenntnisstand jedoch nicht durch eine Kryptoanalyse des Verfahrens, sondern durch das Ausspähen der PIN, entweder direkt durch eine Person, durch einen dem Geldauszahlungsautomaten aufgepflanzten Vorsatz oder über eine Kamera, die in der Nähe des Ziffernblocks (des Geldautomaten oder Zahlungsterminals) angebracht wird.

Eine Schwachstelle des heute üblichen ec-Verfahrens liegt darin, dass die Daten auf dem Magnetstreifen mit einem entsprechenden Lese- und Schreibgerät leicht auslesbar und überschreibbar sind. Als Gegenmaßnahme gegen Kartenkopien verfügen die deutschen ec-Karten über das MM-Merkmal. Leider wird dieses Verfahren nicht weltweit eingesetzt. Daher ist es möglich, auf Grundlage gestohlener Informationen eine gefälschte Karte zu erzeugen und dann mittels des Lastschriftverfahrens oder mittels einer abgelauschten PIN Transaktionen auf Kosten der gestohlenen Karte zu tätigen.

Manipulationen am Kartenterminal oder gefälschte Kartenterminals wären wirkungslos, wenn sich nicht nur die Karte gegenüber dem Automaten ausweisen müsste (einseitige Authentifikation), wie das heute der Fall ist, sondern wenn auch umgekehrt das Kartenterminal sich gegenüber der Karte authentifizieren müsste. Dies erfolgt mit der Einführung des EMV-Chips für Zahlungskarten und für die zugehörigen POS-Terminals und Geldautomaten. Die Buchstaben EMV stehen dabei für die drei Gesellschaften Europay, MasterCard und VISA, die den Standard entwickelt haben.

5. WLAN – WEP, WPA und WPA2 – kabellos (un-)sicher

Anwendung

Die LAN-Technologie (Local Area Network) verbindet einzelne Arbeitsrechner zu einem Netzwerk. Die klassische Form ist dabei eine Infrastruktur aus Koaxial- oder CAT5-Kabeln. Drahtlose Netze sind nicht nur billiger, sondern erlauben zusätzlich mobile Nutzung. Die einfachste Form des WLAN (Wireless Local Area Network) ist dabei unverschlüsselt und erlaubt Zugang für jeden ohne Überprüfung der Identität. Die Verbindung der teilnehmenden Computer untereinander und ins Internet übernehmen Geräte, die nach ihrer Funktionalität „Access-Point“ oder „Router“ genannt werden.

Sind drahtlose Netze unverschlüsselt, so ist jeder im Umkreis von 30 bis 100 Metern (je nach Umgebung) in der Lage auf dieses Netzwerk zuzugreifen. Eine unrechtmäßige Person (Angreifer) könnte nicht nur den Netzdienst frei in Anspruch nehmen, sondern auch jegliche Kommunikation innerhalb des Netzwerkes belauschen und somit auch persönliche Daten und sogar Passwörter, die bei Anwendungen im Internet eingesetzt werden, abgreifen. Das unberechtigte Nutzen des Internetzugangs birgt neben den höheren Verbindungskosten auch die Gefahr, dass illegale Handlungen im Internet dem Eigentümer des Internetanschlusses zur Last gelegt werden können, und dieser kaum Möglichkeiten haben wird, seine Unschuld zu beweisen.

Ziel der Verschlüsselung des WLAN

Um die oben beschriebenen Gefahren zu reduzieren, werden Verschlüsselungsverfahren eingesetzt. Sie zielen darauf, die Authentizität der Partner, sowie die Vertraulichkeit und Integrität der Kommunikation durch die Luft zu gewähren.

Zur Überprüfung der Authentizität der berechtigten Nutzer bietet die WLAN-Technologie verschiedene Varianten, nämlich

- die Möglichkeit die Adresse der Netzkarte (MAC – Media Access Control) zu überprüfen,
- ein vorher definiertes Kennwort zu verwenden (PSK – Pre Shared Key)
- oder die Authentizität implizit über die Verschlüsselung zu überprüfen.
- Weitere Authentifikationsmöglichkeiten sind in dem Protokoll EAP (Extensible Authentication Protocol) zusammengefasst.

Über die Authentifizierung hinaus wird der Datenverkehr in der Luft verschlüsselt und dadurch dem Zugriff von Mitlauschern entzogen. Aktuelle Verfahren dafür sind WEP (Wired Equivalent Privacy), WPA (Wireless Fidelity Protected Access) und WPA2. Zusätzlich können die

Verschlüsselungsverfahren die Integrität der Daten schützen, so dass Angreifer Daten auf dem Übertragungsweg nicht verfälschen oder gar falsche Daten einspielen können.

Technische Grundlagen

Die IEEE Gruppe 802 befasst sich allgemein mit den grundlegenden Protokollen der Netzwerktechnik, welche in der ISO-OSI-Schicht 1 (Physical Layer) und 2 (Data Layer) anzusiedeln sind. Alle Teilnormen, die das WLAN betreffen, sind in der IEEE Standardgruppe 802.11 zusammengefasst, welche 1990 ins Leben gerufen wurde. Zielsetzung dieser Gruppe war es die bestehenden LAN-Technologien im gebührenfreien Frequenzband ISM (Industrial, Scientific, and Medical) im Bereich 2,4 GHz verfügbar zu machen, und so das Medium Luft mit der Funktechnologie einzuschließen.

Das erste Protokoll wurde 1997 verabschiedet, mit dem Übertragungsraten von 1-2MBit/s erreicht wurden. Seit 2003 sind mit 802.11g im gebührenfreien ISM-Band und mit 802.11h im gebührenpflichtigen 5GHz-Band Protokolle mit einer Datenrate bis zu 54Mb/s verfügbar.

Weitere Standards der 802.11-Familie regeln Spezialanwendungen wie zum Beispiel Virtuelle LANs oder den Zugang aus mobilen Fahrzeugen. Wir beschränken uns in diesem Kapitel auf den Standard 802.11i und deren Vorgänger, welche die verbesserte Verschlüsselung des Datenverkehrs als Ziel haben.

Eingesetzte Algorithmen

Das WEP Verfahren verwendet den RC4 Algorithmus zur Erzeugung eines Pseudozufalls-Bitstroms, mit dem die Übertragungsdaten bitweise verschlüsselt werden. WPA setzt dasselbe Verfahren ein, allerdings wird durch die Verwendung temporärer Schlüssel nach dem TKIP Protokoll (Temporal Key Integrity Protocol) eine höhere Sicherheit erreicht.

WPA2 hingegen verwendet den AES-Algorithmus mit CBC-MAC (Counter Cipher Block Chaining – Massage Authentication Code).

Ablauf

Im Folgenden werden die üblichen Authentifikations- und Verschlüsselungsverfahren für die Absicherung von WLANs vorgestellt. Da das EAP-Protokoll in aktuellen Verfahren eine besondere Rolle spielt, wird sein Verbindungsaufbau exemplarisch erläutert. Eine aktuelle Beschreibung der verschiedenen Verfahren findet sich bei [Detken 2006].

Offener Zugang:

Grundsätzlich sollte man nicht auf eine Authentifizierung verzichten. In den meisten Fällen bedeutet dies, dass überdies keine Verschlüsselung verwendet wird, und ein entsprechendes Netzwerk komplett schutzlos ist.

Überprüfung der MAC-Adresse (Media Access Control):

Die MAC-Adresse ist eine weltweit eindeutige Kennung von jedem netzwerkfähigen Gerät. Zur Authentifikation trägt der WLAN-Betreiber erlaubte MAC-Adressen in einer Zugriffskontrollliste entweder direkt im Zugangsgerät (Access-Point) oder auf einem zentralen Server im Netzwerk ein, die der Access-Point zur Authentifikation heranzieht. Da Angreifer in der Lage sind, die MAC-Adresse zu fälschen, sollte dieses Verfahren nicht als alleinige Authentifikation verwendet werden. In Kombination mit anderen Authentifikationsverfahren ist die Überprüfung der MAC-Adressen durchaus sinnvoll, da ein Angreifer dann eine zusätzliche Hürde zu überwinden hat.

Einsatz eines geheimen Schlüssels (PSK – Pre Shared Key):

Um Netze abzusichern, nutzt man üblicherweise ein Geheimnis, das nur autorisierten Personen bekannt ist. Die einfachste Variante sieht dabei ein gemeinsames Kennwort für alle Nutzer des Netzwerkes vor. Da alle Teilnehmer dieses Kennwort kennen müssen und die Netzwerkkomponenten dementsprechend konfiguriert werden müssen, ist diese Variante nur für kleine Netzwerke oder Heimnetzwerke sinnvoll. Weiterhin wäre es aus Sicherheitsgründen nicht sinnvoll große Netzwerke auf diese Weise zu betreiben, da man sich auf die Geheimhaltung aller Teilnehmer verlassen muss.

Bei der Verschlüsselung mit WEP (s.u.) dient das gemeinsame Geheimnis außerdem noch als Schlüssel für das Verschlüsselungsverfahren, so dass ein Angreifer bei Kenntnis des Schlüssels außerdem in der Lage ist, abgefangene Nachrichten zu entschlüsseln.

WEP (Wired Equivalent Privacy):

WEP ist eine frühe Form der Verschlüsselung drahtloser Netze. Das Ziel war dabei eine etwa gleich hohe Sicherheit wie in drahtgebundenen Netzen. WEP enthält eine Authentifizierung und eine Verschlüsselung, die auf den RC4-Algorithmus basieren.

Die WEP-Authentifikation setzt ein vorab verabredetes gemeinsames Geheimnis zwischen Client und Access-Point (Pre Shared Key) für ein einfaches Challenge-Response-Verfahren ein: Der Access-Point sendet eine 1024-Bit lange Zufallszahl an den Client. Der Client erzeugt zunächst mit Hilfe des RC4-Algorithmus und des zuvor verabredeten Geheimnisses eine Bitfolge, die von außen wie eine Zufallsfolge aussieht, aber natürlich vom Access-Point nachgeahmt werden kann, da es das gemeinsame Geheimnis auch kennt. Im nächsten Schritt verknüpft der Client diese (Pseudozufalls-)Bitfolge bitweise (XOR) mit der vom Access-Point erhaltenen Zufallszahl und sendet das Kryptogramm an den Access-Point zurück. Der Access-Point kann das Kryptogramm

seinerseits erzeugen und mit dem vom Client empfangenen Kryptogramm vergleichen. Wenn die Kryptogramme übereinstimmen, wird der Client zugelassen, andernfalls abgelehnt.

Die WEP-Verschlüsselung ist ähnlich wie die WEP-Authentifikation. Dabei wird eine Pseudozufalls-Bitfolge erzeugt, welche dann mit der Nachricht bitweise (XOR) verknüpft wird. Bei der Erzeugung der Pseudozufalls-Bitfolge wird wiederum der RC4-Algorithmus verwendet. Dabei fließen der vorab verabredete Schlüssel (Pre Shared Key), der eine Länge von 64 Bit oder 128 Bit hat, und ein vom Access-Point übermittelter Initialisierungsvektor ein, der eine Länge von 24 Bit besitzt.

Implizite Authentifikation:

Bei der impliziten Authentifikation wird auf ein Authentifikationsverfahren selbst verzichtet, allerdings ist die Verschlüsselung nur mit einem zuvor ausgetauschten Schlüssel (Pre Shared Key) möglich. Gerade beim WEP wird empfohlen, lieber auf den WEP-Authentifikationsmechanismus zu verzichten, da es einem potentiellen Angreifer zusätzliche Angriffspunkte bietet, den verwendeten Schlüssel herauszufinden (s.u. unter „Schwächen“). Beim reinen WEP wird deshalb die implizite Authentifikation empfohlen.

EAP (Extensible Authentication Protocol):

EAP ist kein einzelnes Authentifikationsverfahren, sondern ein Rahmen für verschiedene Authentifikationsverfahren wie zum Beispiel die Authentifikation mit User-Id und Passwort oder mit Signatur und Zertifikat. Das Grundprinzip besteht in einer Ende-zu-Ende-Authentifikation zwischen dem Client und einem Authentifikation-Server über den vermittelnden Access-Point hinweg. Da in drahtlosen Netzen in der Regel keine direkte Ende-zu-Ende-Kommunikation möglich ist, muss diese von einem anderen Protokoll wie IEEE 802.1X realisiert werden. Im Folgenden werden die einzelnen Schritte einer Authentifikation mit Hilfe von EAP beschrieben.

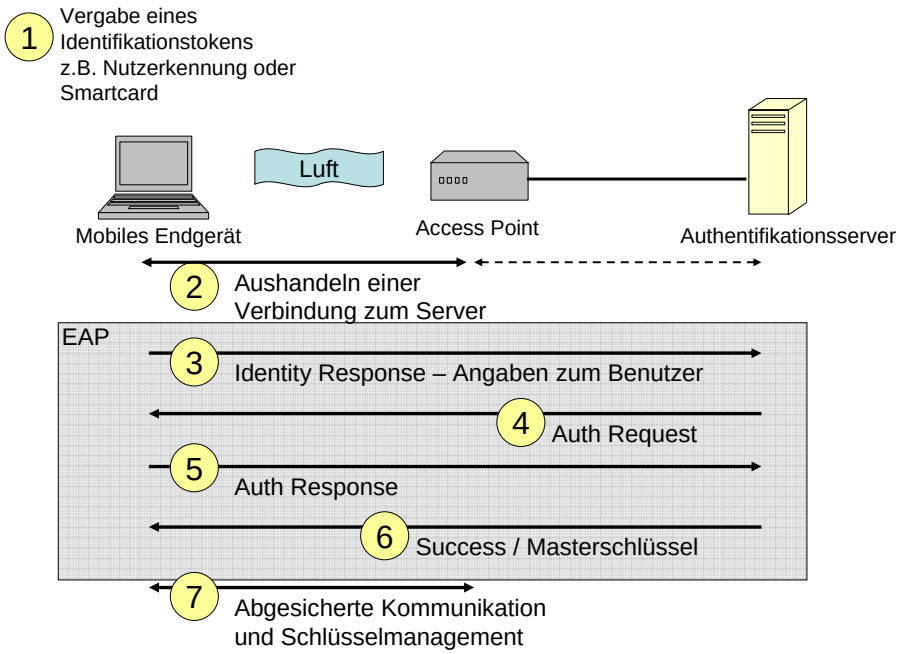

Abb. 5.1: Authentifikation mit EAP (Extensible Authentication Protocol)

Schritt 1: Im Vorfeld wird ein Authentifikationsverfahren festgelegt. Der Nutzer erhält ein „Token“, mit dem er seine Identität nachweisen kann. Das kann eine Benutzerkennung mit Kennwort, die sich der Nutzer merken muss, oder eine Smartcard sein, die zertifikatsbasierte Signaturen ausführen kann.

Schritt 2: Vor dem eigentlichen EAP-Protokoll bittet der Client den Access-Point um die Vermittlung zu einem Authentifikation-Server. Der Access-Point baut daraufhin eine Verbindung zwischen Client und Server her. Dabei können weitere Sicherungsprotokolle wie zum Beispiel eine SSL-Verschlüsselung initialisiert werden.

Schritt 3: Der Client teilt dem Server seine Identität mit, zum Beispiel in Form einer Nutzerkennung („Identity Response“).

Schritte 4 und 5: Auf die Anfrage vom Server an den Client, die in Schritt 3 mitgeteilte Identität zu beweisen („Authentication Request“), überträgt der Client seine Authentifikationsdaten an den Server („Authentication Response“). Je nach Ausprägung findet hier ein symmetrisches oder ein asymmetrisches Authentifikationsprotokoll statt, zum Beispiel das X.509-Protokoll mit Signatur und Zertifikat.

Schritt 6: Der Authentifikation-Server überprüft den Identitätsnachweis und übermittelt seine Entscheidung. Im Erfolgsfall („Success“) überträgt er einen Masterschlüssel, der für die Verschlüsselung der weiteren Kommunikation verwendet wird.

Schritt 7: Für die weitere Kommunikation ist der Authentifikation-Server nicht mehr notwendig. Der Client und der Access-Point können nun anhand des Masterschlüssels alle weiteren benötigten Schlüssel erzeugen und ggf. miteinander abgleichen. Darüber hinaus gibt der Access-Point den Zugang zum Netzwerk frei, und ermöglicht eine abgesicherte drahtlose Kommunikation.

WPA und WPA2 (Wireless Fidelity Protected Access):

Bei WPA und WPA2 unterscheidet man zwei Authentifikationsmodi. Neben dem Pre-Shared-Key-Verfahren, welches für Heimanwendungen beibehalten wird, werden nun die EAP-Verfahren bevorzugt, welche über das IEEE 802.1X Protokoll eine direkte Server-Client-Authentifikation durch den Access-Point hindurch ermöglichen.

Zur Verschlüsselung selbst gibt es bei WPA und WPA2 eine Reihe von Schlüsseln, die aus einem Masterschlüssel, der für jeden Client unterschiedlich ist, abgeleitet werden. Im Pre Shared-Key-Modus wird der Masterschlüssel aus dem gemeinsamen geheimen Schlüssel, der SSID (Name einer WLAN-Verbindung) und einigen anderen Werten abgeleitet. Wird EAP verwendet, so legt der Authentifikation-Server den entsprechenden Masterschlüssel fest, und übermittelt ihn im Laufe des EAP-Protokolls (s.o. Schritt 6) gesichert an den Client und an den Access-Point.

Aus dem Masterschlüssel werden im nächsten Schritt verschiedene Schlüssel für die Gruppenkommunikation, die Integritätsprüfung, den Schlüsselaustausch und für die Datenverschlüsselung abgeleitet. Außer den Schlüsseln für die Gruppenkommunikation gibt es alle Schlüssel paarweise, je einen zum Senden und einen zum Empfangen.

WPA verwendet zur Verschlüsselung ein so genanntes Temporal Key Integrity Protocol (TKIP), welches aus Kompatibilitätsgründen auf die WEP-Verschlüsselung aufsetzt. Verbesserungen gegenüber WEP wurden zum Beispiel durch Verwendung eines besseren Integritätschecks, durch längere Initialisierungsvektoren und ein besseres Schlüsselmanagement erreicht.

Wie bei WEP wird bei WPA eine Pseudozufalls-Bitfolge erzeugt, welche bitweise mit der Nachricht verknüpft wird. TKIP verwendet für jedes versendete Datenpaket einen anderen Schlüssel.

Bei WPA2 greift man auf den standardisierten AES-Algorithmus (Advanced Encryption Standard) im CBC-MAC-Modus (Cipher-Block-Chaining-Massage Authentication Code) zurück. In diesem Modus wird die zu übertragende Nachricht in 128-Bit lange Blöcke zerlegt. Mit dem Startblock beginnend fließt das Kryptogramm eines jeden Blocks in die Verschlüsselung des jeweiligen Folgeblocks mit ein, um auf diese Weise die komplette Nachricht zu verschlüsseln.

Stärken

WLAN-Betreiber können mit angemessenen Authentifikations- und Verschlüsselungsverfahren Angriffe auf ihre Netzwerke deutlich erschweren oder verhindern. Damit schützen sie sich nicht nur vor Lauschen und Ressourcendiebstahl, sondern kommen auch ihren Haftungspflichten gegenüber illegalen Handlungen in ihren Netzen nach. Selbst geringe Schutzmaßnahmen (z.B. MAC-Adressen-Überprüfung) sind besser als gar keine, da sie jedenfalls das unbeabsichtigte Einwählen und das Einwählen durch unerfahrene Angreifer abhalten.

Standardtechniken bieten heutzutage die Möglichkeit, sich mit stärkeren Mitteln zu schützen. Das WPA2-Verfahren erreicht dabei schon ein recht hohes Sicherheitsniveau durch die Verwendung des starken AES-Algorithmus. EAP bietet eine allgemein verfügbare Schnittstelle für verschiedene Authentifikationsverfahren.

Schwächen

Die hier erwähnten Schutzmechanismen sind aufwändig zu konfigurieren. Ein durchschnittlicher Nutzer hat oft keine ausreichenden administrativen Kenntnisse, um bei einem WLAN-Access-Point die notwendigen Einstellungen vorzunehmen, so dass in diesem Fall komplett auf Schutz verzichtet wird. Hinzu kommt, dass häufig voreingestellte WLAN-Namen (SSID) und Kennwörter verwendet werden. Angreifer kennen diese Voreinstellungen, und können sie zu ihrem Vorteil einsetzen.

Die Überprüfung erlaubter MAC-Adressen bietet zwar einen geringen Schutz, aber versierte Angreifer sind mit entsprechender Zusatzsoftware sehr wohl in der Lage, die MAC-Adresse zu fälschen und diesen Schutz zu umgehen.

Die WEP-Verschlüsselung weist Mängel im Schlüsselmanagement auf, so dass Angreifer das gemeinsame Geheimnis (Pre Shared Key) ermitteln können. Und zwar kann ein Lauscher der WEP-Authentifikation aus der Zufallszahl, die der Access-Point dem Client überträgt, und dem RC4-Kryptogramm, das der Client aus dieser Zufallszahl und dem Pre Shared Key gebildet und dem Access-Point zurückgeschickt hat, den Pre Shared Key berechnen. Damit kann der Angreifer nicht nur eine falsche Identität vortäuschen, sondern auch die verschlüsselte Kommunikation anderer Nutzer entschlüsseln und belauschen.

Das bei WPA und WPA2 eingesetzte Protokoll IEEE 802.1X war ursprünglich nicht für drahtlose Kommunikation gedacht. Durch Einspielen falscher Signale bei der Authentifizierung (Man-in-the-middle), ist es daher möglich, die Sitzung eines Benutzers zu übernehmen (Session Hijacking). Dies kann durch eine Verschlüsselung der Authentifikation durch zusätzliche Protokolle (zum Beispiel PEAP – protected EAP) verhindert werden.

Die höchste Sicherheitsstufe (WPA2) bei WLAN ist leider sehr selten im Einsatz. Viele WLAN-Hardware-Komponenten beherrschen AES nicht. Andere Komponenten unterstützen das Verfahren nur durch softwareseitige Berechnungen, so dass erhebliche Einbußen bei der Übertragungsgeschwindigkeit die Folge sind. Das macht WPA2 für viele Nutzer unattraktiv.

6. Elektronische Türschlösser

Anwendung

Zutritt zu geschützten Objekten, zum Beispiel zu Gebäuden oder in ein Fahrzeug, wird traditionell mit physischen Schlüsseln gewährt. Wer den passenden Schlüssel hat, kommt hinein, andere nicht. Bei Diebstahl oder Verlust eines physischen Schlüssels muss das ganze Schloss ersetzt werden, das kann bei größeren Schließanlagen mit Gruppenschlüsseln sehr teuer werden. Bei elektronischen Schließanlagen dagegen brauchte nur das Schloss umprogrammiert zu werden, und der verlorene oder gestohlene Schlüssel wäre wertlos. Auch bei einem Wechsel der Zutrittsberechtigung müssten bereits verteilte elektronische Schlüssel nicht umverteilt werden, sondern die betroffenen Schlösser würden umprogrammiert werden und jeder könnte seinen Schlüssel behalten. Elektronische Schießanlagen könnten also die Schlüsselverwaltung erleichtern und sicherer machen.

Für elektronische Schließanlagen bietet sich ein weites Spektrum an Technologien an. In Frage kommen zum Beispiel kontaktfreie Systeme wie Infrarot, RFID oder Bluetooth. Komfortable Funkschlüssel sind zum Beispiel bei Autos heute gang und gäbe. Des Weiteren gibt es elektronische Schließsysteme, die zwar kontaktbehaftet sind, aber statt der Auswertung von physischen Strukturen des Schlüsselbarts auf eine Datenübertragung zurückgreifen.

Die elektronischen Übertragungswege müssen natürlich gegen Fehlbedienung, Lauschangriffe und Manipulation geschützt werden. Das gilt insbesondere für kontaktlose Schlüssel, die ein Schloss durch die Luft bedienen.

Ziel der Verschlüsselung bei elektronischen Türschlössern

Verschlüsselungsverfahren werden dafür eingesetzt, dass Schlösser und Schlüssel sich gegenseitig sicher erkennen und dass die Schließbefehle nicht von unautorisierter Seite mitgehört und ggf. kopiert werden können.

Von besonderer Bedeutung bei elektronischen Schließsystemen ist die eindeutige Identifizierung des richtigen Schlüssels im Besitz der berechtigten Person (Authentizität). Ein fest installiertes Schloss oder Terminal kann als vertrauenswürdige Instanz angesehen werden, da die berechtigte Person mit dem richtigen Schloss ja physisch davor steht und es erkennen kann. Auf eine Authentifikation des Schlosses gegenüber dem Schlüssel kann dann verzichtet werden. Werden Schließ- und Entriegelungsbefehle aber in einem Netzwerk zu einem entfernten Schloss übertragen,

so müssen sich die Schlösser ihrerseits ebenfalls identifizieren. Wenn dann noch ein Schlüssel-Server im Netz die Verteilung der Schlüssel organisiert, dann muss sich der Schlüssel-Server ebenfalls authentifizieren, um zu verhindern, dass ein Angreifer dem Schloss eigene Schlüssel zusendet. Zusätzlich muss die Kommunikation verschlüsselt stattfinden.

Findet eine Authentifikation kontaktlos statt, wie bei Funkschlössern, so muss sichergestellt werden, dass ein Angreifer nicht in der Lage ist, die Schließ- und Entriegelungsbefehle durch die Luft mitzulauschen und als eigene Identifikation wieder in das System einzuspielen (Replay-Attacke).

Technische Grundlagen

Bei elektronischen Schließsystemen sind zwei Phasen zu unterscheiden: erstens die Schlüsselverwaltung, bei der es darum geht, Schlüsselträger herzustellen, unter den berechtigten Personen zu verteilen und ggf. umzuverteilen; zweitens die Schließvorgänge, bei denen ein Schlüsselträger über einen elektronischen Kontakt oder durch die Luft per Funk oder Infrarot Schließ- und Entriegelungsbefehle an ein Schloss überträgt. Wir erläutern hier nur die zweite Phase. Für die Schlüsselverwaltung setzen wir dabei voraus, dass die Schlüsselträger, seien es elektronische Kontaktschlüssel oder kontaktlose Transponder, vom Betreiber der Schließanlage vor der Ausgabe richtig programmiert wurden, wobei Algorithmen, Identifikationsnummern und geheime kryptographische Schlüssel eingebracht werden können.

Für die Schließvorgänge selbst ist wiederum zu unterscheiden, wer die Entscheidung trifft, ob ein Schließ- oder Entriegelungsbefehl ausgeführt wird. Elektronische Schlösser können eigenständig und ohne Anschluss an einen zentralen Server entscheiden, wenn ihnen die Berechtigungen dezentral vorab einprogrammiert wurden. Das ist zum Beispiel bei Funkschlössern für Fahrzeuge der Fall. Alternativ können elektronische Schlösser mit einem zentralen Server verbunden sein, dem sie die Schlüsselbefehle vom Schlüsselträger weiterleiten und von dem sie die Entscheidung zum Schließen oder Entriegeln mitgeteilt bekommen. Dies ist in der Regel bei Schließanlagen in Gebäuden oder Gebäudekomplexen der Fall. Im Fall der mit einem Server vernetzten Schlösser ist die Kommunikation zwischen Schlössern und Schlüssel-Server ebenfalls abzusichern. Wir gehen hier davon aus, dass dieses Netz gegen Lauschangriffe und Manipulation geschützt ist, zum Beispiel dadurch, dass sie im physisch geschützten Gebäudebereich fest verdrahtet sind und unter zentraler Aufsicht stehen.

Für die Schließ- und Entriegelungsvorgänge zwischen elektronischem Schlüsselträger und Schloss gibt es verschiedene technische Ausführungen von Identitätsnachweisen. Es gibt biometrische Sensoren, sowie kontaktbehaftete und kontaktlose elektronische Schlüsselträger.

Die biometrische Variante löst die Authentifizierung anhand eines biometrischen Merkmals aus. Ein Sensor für Fingerabdrücke (oder für Augenhintergrundmuster oder für Gesichtsformen) ist

direkt neben dem Schloss angebracht und mit einem Prozessor verbunden, der sowohl neue Berechtigungen aufnehmen als auch bestehende Berechtigungen prüfen kann. Die Datensätze, welche die biometrischen Informationen beinhalten, (so genannte „Templates“) können als Schlüssel interpretiert werden, welche direkt im Schloss gespeichert sind. Separat von der Schlosseinheit ist lediglich noch eine Schaltung notwendig, welche das Relais zur Öffnung der entsprechenden Tür mit Spannung versorgt. Von der Firma „ekey biometric systems“ in Linz/Oberösterreich werden zum Beispiel Varianten von einer einzelnen Tür für die Heimanwendung bis zur komplexen Vernetzung in Firmengebäuden angeboten.

Kontaktbehaftete Schlüsselträger kommunizieren mit ihrem Schloss vor Ort über einen elektrischen Schleifkontakt.

Kontaktlose Schlüsselträger können die Datenübertragung zum Schloss über Bluetooth, RFID oder Infrarot realisieren. Eine allgemein etablierte Anwendung kontaktloser Schlüsselträger sind Autoschlüssel. Die frühen Formen von Autoschlüsseln übertrugen die Daten per Infrarot. Das hat den Nachteil, dass eine Sichtlinie zwischen Schlüsselträger und Empfänger erforderlich war. Heute wird hauptsächlich die Funktechnologie verwendet. Die meisten Autoschlüssel senden im lizenzfreien ISM-Band (Industrial, Scientific, and Medical Band) bei 433 MHz.

Zusätzlich beinhalten moderne Autoschlüssel einen RFID-Transponder, der die elektronische Wegfahrsperre schaltet. Während RFID-Chips eine Reichweite von ca. 10 cm besitzen, hat die Funkübertragung eine Reichweite von etwa 10 m.

Eingesetzte Algorithmen

In elektronischen Schließsystemen kommen in der Regel symmetrische Verfahren zu Einsatz. Bei vielen Anwendungen wie zum Beispiel bei einem KFZ-Schlüssel werden die kryptographischen symmetrischen Schlüssel während der Produktion in den Schlüsselträger eingebettet, so dass auf eine Aushandlung der Schlüssel mit asymmetrischen Verfahren verzichtet werden kann. Dabei kommen alle symmetrischen Algorithmen, wie Triple-DES, AES, IDEA, RC4 zum Einsatz.

Viele Hersteller neigen dazu, verwendete Algorithmen geheim zu halten, da sie davon ausgehen, dass ihre Systeme dadurch sicherer sind. Allerdings gibt es genügend viele öffentlich bekannte Verschlüsselungsverfahren, die als sicher gelten. Deshalb kann das Publizieren des Algorithmus ein Qualitätsmerkmal sein, während die Geheimhaltung den Verdacht nährt, dass aus Kosten- und Effizienzgründen unsichere Verfahren eingesetzt werden.

Ablauf

Die Identifizierung eines Schlüsselträgers (ob als biometrie-gesteuerter Schlossprozessor, als elektronischer Kontaktschlüssel oder als kontaktloser Transponder) gegenüber einem Schloss

basiert auf der einseitigen symmetrischen Authentifikation, wie sie in Abschnitt 1 oben über symmetrische Protokolle beschrieben wird. Da der Schlüsselträger vom Schloss keine sicherheitsrelevanten Daten erhält, aufgrund dessen er eine Entscheidung fällen muss, kann auf eine Authentifikation des Schlosses verzichtet werden. Die einseitige Authentifizierung eines Schlüsselträgers gegenüber einem Schloss führt zu einer Entscheidung des Schlosses sich zu öffnen oder nicht zu öffnen, bzw. sich zu schließen oder nicht zu schließen.

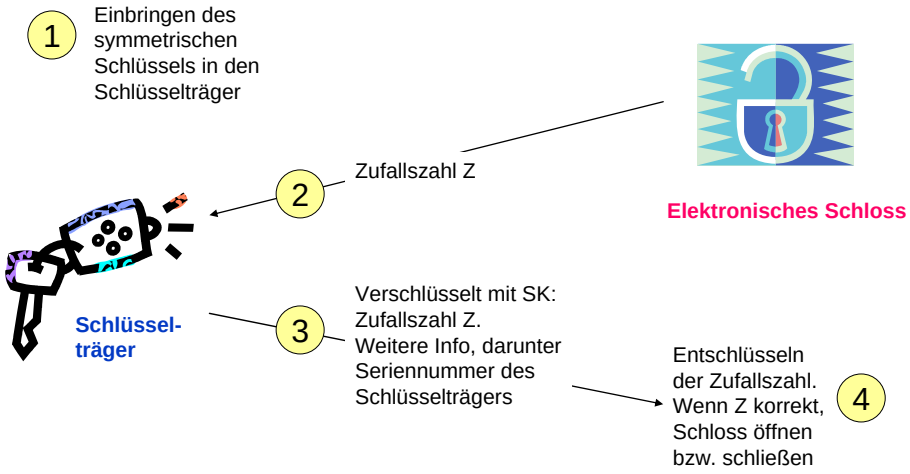

Abb. 6.1: Einseitige Authentifizierung eines Schlüsselträgers gegenüber seinem elektronischen Schloss

Das symmetrische Authentifikationsprotokoll in Abschnitt 1, das hier zum Einsatz kommt, nutzt als Frischekennzeichen eine Zufallszahl, die das Schloss dem Schlüsselträger zusendet und die der Schlüsselträger dem Schloss verschlüsselt zurücksendet.

Schritt 1: Ein Schlüsselträger und sein elektronisches Schloss verwenden einen gemeinsamen geheimen symmetrischen Schlüssel. Diesen kryptographischen Schlüssel, anhand dessen das Schloss den berechtigten Schlüsselträger erkennt, berechnet das elektronische Schloss aus einem Hauptschlüssel, den er für alle Schlüsselträger verwendet, und aus der Seriennummer des Schlüsselträgers. Das elektronische Schloss braucht sich deswegen nur seinen Hauptschlüssel zu merken, aber keine geheimen Schlüssel der Schlüsselträger, die er sich ja bei Bedarf immer wieder ausrechnen kann. Allerdings merkt er sich dazu die Seriennummern aller Schlüsselträger, die ihn öffnen und schließen dürfen. Dem Schlüsselträger wird der gemeinsame Schlüssel bei der Herstellung vor der Ausgabe an die berechtigte Person eingespeichert. Zum Beispiel programmiert ein Autohersteller die Autoschlüssel und Türschlösser unmittelbar vor der Auslieferung von Auto und Schlüssel an den Händler bzw. an den Kunden.

Schritt 2: Bei Annäherung des Schlüsselträgers an das elektronische Schloss löst der Mensch, der den Schlüsselträger in der Hand hält, ein Signal aus, mit dem er anzeigt, ob er das Schloss öffnen oder schließen will. Daraufhin sendet das elektronische Schloss eine Zufallszahl Z aus, die der Schlüsselträger empfängt.

Schritt 3: Der Schlüsselträger verwendet den ihm eingepflanzten symmetrischen Schlüssel, um die Zufallszahl Z zu verschlüsseln. Das Kryptogramm sendet er dann gemeinsam mit weiteren Informationen, darunter seiner Seriennummer, an das elektronische Schloss zurück.

Schritt 4: Das elektronische Schloss prüft erst, ob die Seriennummer bei ihm eingespeichert ist. Wenn das der Fall ist, dann berechnet es aus der Seriennummer und seinem Hauptschlüssel den geheimen Schlüssel des Schlüsselträgers und verwendet diesen zur Entschlüsselung des zugesandten Kryptogramms. Wenn das Ergebnis mit der von ihm in Schritt 2 ausgesendeten Zahl Z übereinstimmt, dann ist der Schlüsselträger authentisch, und das Schloss kann den Befehl ausführen, das Schloss zu schließen bzw. zu öffnen.

Aufwändigere elektronische Schließanlagen wie in Firmengebäuden arbeiten mit zusätzlichen Entscheidungsregeln (so genannten „Policies“). Regeln können sich auf Zeiten, Personen, Personengruppen und Anzahl Schließvorgängen beziehen. So kann es zum Beispiel Regeln geben, die den Zutritt nur zu bestimmten Zeiten oder verschiedenen Gruppen zu verschiedenen Zeiten erlaubt. Andere Regeln können einmaliges Schließen und Öffnen oder eine andere vorherbestimmte Anzahl von Schließvorgängen erlauben. Wieder andere Regeln können ein Schloss dazu veranlassen, einen Schlüssel nach einer bestimmten Anzahl von Fehlversuchen zu sperren. Soweit Regeln programmierbar sind, können sie von Schließanlagen mit Hilfe digitaler Prozessoren realisiert werden.

Stärken

Ein Vorteil elektronisch gesteuerter gegenüber physischen Schließanlagen liegt darin, dass sie leichter und billiger zu verwalten sind. Die Nutzung kontaktloser Schlüsselträger ist für die Anwender zudem bequemer. Geht bei den beschriebenen Verfahren ein Schlüsselträger verloren, so reicht es aus, den entsprechenden Schlüssel zu sperren und einen neuen Schlüsselträger mit neuem Schlüssel auszuhändigen, während bei physischen Schlüsseln ein Austausch aller betroffenen Schlösser erforderlich ist.

Schließregeln („Policies“) bringen zusätzlichen Nutzen wie zum Beispiel die Ermittlung des Status einer Tür, das Festlegen von Zeiten und die flexible Vergabe von Gruppenschlüsseln.

Schließanlagen werden zentral verwaltet und kommen daher mit den effizient arbeitenden symmetrischen Verschlüsselungsprotokollen aus. Sofern sie auf starken Verschlüsselungsverfahren wie AES mit entsprechend großen Schlüssellängen beruhen, sind sie sicher.

Schwächen

Bei elektronischen Schließsystemen ist elektrische Energie notwendig. Für den Fall eines Stromausfalls oder bei technischen Problemen muss eine alternative Lösung verfügbar sein. Da in den meisten Fällen zwei Schließsysteme parallel bestehen, reduziert sich auch die Gesamtsicherheit, da ein Angreifer nun die Auswahl hat, welches System er angreift. Zusätzlich müssen auch bei Stromausfall Fluchtwege verfügbar bleiben.

Die Festlegung von Policies kann bei großen Systemen sehr komplex werden. Hinzu kommt, dass sowohl die Policies als auch die Schlüssel in die Schlösser verteilt werden müssen. Je nach Lösung kann dies einen hohen Aufwand bedeuten.

Wenn die eingesetzten Verschlüsselungsverfahren schwach sind, dann sind elektronische Schlüssel leichter angreifbar als physische Schlüssel. Insbesondere Transponder reagieren auf ein Funksignal, auch wenn es nicht von einem autorisierten Schloss stammt, sofern die Authentifizierung nur einseitig erfolgt, was die Regel ist. Dabei gibt der Schlüsselträger Daten preis, die kopiert und für Fälschungen ausgenutzt werden könnten. Wenn sie gut verschlüsselt sind und Frischemerkmale wie Zufallszahlen und Zeitinformationen tragen, dann kann nichts passieren. Wenn sie aber schlecht verschlüsselt sind, dann können die zugehörigen kryptographischen Schlüssel geknackt und mit ihnen perfekt gefälschte Funkschlüssel nachgebaut werden. Ein physischer Diebstahl wäre dann nicht mehr erforderlich.

7. Kontaktloses Lufthansaticket mit einer RFID-Chipkarte, die ausschließlich Authentifizierungsdaten enthält

Anwendung

Um Personal und Zeit zu sparen, verfolgte die Lufthansa mit dem Projekt „Fliegen ohne Ticket“ das Ziel, die Abfertigung von Fluggästen beim Einstieg („Boarding“) so zu automatisieren, dass es für die Fluggäste bequem, einfach und schnell geht. Dazu wurde eine Kunden-Chipkarte für elektronische Tickets eingeführt, die zusätzlich andere bereits bestehende Dienste wie Kreditkartenfunktion, Telefonkarte und Bonussystem integriert. Wir beschränken uns bei diesem Beispiel auf die Funktion der kontaktlosen Identifikation der Chipkarte zur Ausstellung der Boardingkarte. Das Pilotprojekt wurde von Mai bis Dezember 1995 auf mehreren tausend Flügen auf der Strecke Frankfurt-Berlin mit 600 Kunden erfolgreich durchgeführt, aber bis heute noch nicht in den Regelbetrieb übernommen [Rankl, Effing 2002].

Ziel der Verschlüsselung bei kontaktlosen Flugtickets

Kontaktlose Dienste nutzen die Luft als Übertragungsmedium, das auf einen Umkreis von mehreren Zentimetern bis Metern frei abstrahlt. Der freie technische Zugriff auf das

Übertragungssignal wird durch Verschlüsselung in der Weise eingeschränkt, dass Mitlauscher das Signal zwar empfangen, aber nicht verstehen oder sinnvoll beeinflussen können. Das Ziel der Verschlüsselung ist in diesem Beispiel die sichere Authentifizierung der Chipkarte des Passagiers gegenüber dem Lufthansa-Terminal, aber nicht umgekehrt, da ein vorgetäuschtes Terminal zwar vielleicht ein Ticket unberechtigt auslesen könnte, aber eine dabei erschlichene Boradingkarte praktisch wertlos wäre. Weiterhin werden keine Inhalte wie Namen, Flugziele oder Preise übertragen, so dass die Vertraulichkeit auch durch Datensparsamkeit gewahrt wird.

Technische Grundlagen

Die kontaktlose Chipkarte verwendet die RFID-Technologie („Radio Frequency Identification“). Die Chipkarte selbst enthält keine eigene Spannungsquelle, sondern wird vom Terminal durch die Luft mit Strom versorgt. Dazu sendet das Terminal ein Funksignal an die Chipkarte. Das elektromagnetische Wechselfeld des Funksignals induziert in einer Magnetspule in der Chipkarte eine Spannung. Dieser Induktionswechselstrom dient sowohl der Energieversorgung in der Chipkarte, als auch der Datenübertragung an die Chipkarte. In der Chipkarte laufen dann die unten beschriebenen automatisierten Prozesse ab. Die Rückübertragung funktioniert so, dass die Chipkarte mit ihrer vom Terminal gewonnenen Energie einen inneren Widerstand steuert und dadurch das innere elektromagnetische Feld verändert, das wiederum umgekehrt das elektromagnetische Feld des Terminals beeinflusst und als Signal gewertet wird. Bei dem hier verwendeten System können Terminal und Chipkarte Entfernungen von 10 cm durch die Luft überwinden.

Das System ist denkbar einfach gebaut: Die Chipkarte enthält nur ganz wenige Daten, nämlich (für den hier beschriebenen Zweck) eine Kartenseriennummer und einen symmetrischen Schlüssel, die ihr bei Ausgabe durch die Lufthansa eingespeichert wurden, sowie die Fähigkeit, eine Zahl vom Terminal zu empfangen, diese (durch einen eigenen Co-Prozessor) symmetrisch zu verschlüsseln und das Ergebnis an das Terminal zurückzusenden.

Weitere technische Daten: Kartenformat: ID-1; Transaktionszeit: ca. 100 ms; Sendefrequenz: 13,56 kHz; Speicherchip: SLE 44R35 (1kB EEPROM Speicher, davon 48 Bytes benutzt); Taktfrequenz der Karte: 3,5 MHz; Kommando INTERNAL AUTHENTIFICATE nach ISO/IEC 7816-4.

Eingesetzte Algorithmen

Das hier verwendete Verfahren der einseitigen symmetrischen Authentifizierung könnte jeden symmetrischen Verschlüsselungsalgorithmus verwenden. Die üblichen Verfahren wie Triple-DES, AES, RC4 oder IDEA sind alle sicher und effizient genug für den Einsatz in diesem einfachen RFID-System.

Ablauf

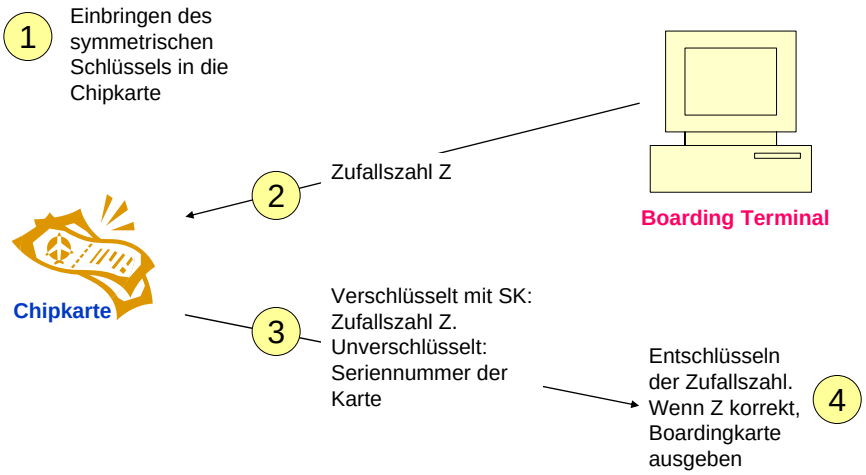

Abb. 7.1: Einseitige Authentifizierung der Chipkarte des Fluggastes gegenüber dem Lufthansa-Terminal

Schritt 1: Bei der Erstellung der Chipkarte bringt der Betreiber des Systems, in diesem Fall die Lufthansa, eine Kartenseriennummer und einen symmetrischen Schlüssel SK in den Speicher der Chipkarte des Fluggastes ein. Üblicherweise wird der Schlüssel aus einem Hauptschlüssel, der in einem sicheren Bereich des Firmennetzwerks gespeichert ist, und der Seriennummer der Karte abgeleitet. Dazu werden Teile der Seriennummer mit dem Hauptschlüssel verschlüsselt. Das Ergebnis ist dann der geheime symmetrische Schlüssel, der bei jeder Chipkarte verschieden ist. Würde ein Angreifer den Hauptschlüssel erfahren, so wäre es ihm möglich, für jeden Passagier den individuellen symmetrischen Schlüssel zu errechnen.

Der Vorteil des errechneten Schlüssels liegt darin, dass das Firmensystem nicht für jeden Kunden den jeweiligen symmetrischen Schlüssel vorhalten muss, da das Terminal diesen aus der Seriennummer der Karte berechnen (lassen) kann. Das Terminal ist mit dem Firmennetz der Lufthansa verbunden und kann daher auf die Liste der Kartenseriennummern und auf den Hauptschlüssel zugreifen, bzw. durch einen Server im Hintergrund sicher zugreifen lassen.

Die Chipkarte wird dem Fluggast ausgehändigt: Danach kann er sie für jedes Flugticket verwenden. Er bestellt seine Flüge in Zukunft telephonisch oder im Internet. Das Flugticket wird nicht etwa auf der Chipkarte gespeichert, sondern in einer Datenbank des Lufthansanetzes in einem sicheren Ticket-Server.

Schritt 2: Will sich ein Fluggast eine Boardingkarte erzeugen, so nähert er seine Chipkarte dem Lufthansaterminal und drückt auf den Auslöseknopf. Nun sendet das Terminal eine unverschlüsselte Zufallszahl Z an die Chipkarte. Diese Zufallzahl muss jedes Mal frisch sein. Diese Anfrage wird als „Challenge“ (Herausforderung) bezeichnet.

Schritt 3: Die Chipkarte verschlüsselt die erhaltene Zufallszahl Z mit Hilfe des auf der Karte hinterlegten symmetrischen Schlüssels SK. Zu diesem Zweck haben die Chipkarten einen gesonderten Verschlüsselungschip, da die Transponder (Sender automatisierter Antworten) für solche komplexen Operationen nicht ausgestattet sind.

Das Ergebnis der Verschlüsselung sendet die Chipkarte an das Terminal zurück. Da das Terminal zur Berechnung des symmetrischen Schlüssels auch die Seriennummer der Karte benötigt, wird diese ebenfalls übertragen.

Schritt 4: Das Terminal gibt die verschlüsselte und unverschlüsselte Zufallszahl und die Kartenserienummer über das Lufthansanetz an einen sicheren Server im Hintergrund weiter, der die weiteren Berechnungen ausführt. Dieser errechnet nun, genau wie bei der Erstellung der Chipkarte, den geheimen symmetrischen Schlüssel der Karte und entschlüsselt damit die Zufallszahl, die er mit der unverschlüsselten Zufallszahl Z des Terminals vergleicht. Sind sie gleich, ist die Chipkarte echt, sind sie ungleich, so war es nicht die Chipkarte, die mit dem Terminal kommuniziert hatte. Dieses Ergebnis meldet der Lufthansa-Server an das Terminal zurück. Auf diese Weise kann das Terminal sicher sein, dass es sich um die richtige Chipkarte handelt, denn keine andere hätte diese frische Zufallszahl Z richtig verschlüsseln können. Daher kann auch kein Eindringling die übertragenen Daten abhören und bei späterer Gelegenheit wieder einspielen, denn das nächste Mal würde das Terminal ja eine neue, frische Zufallszahl senden. Das Terminal kann also bei erfolgreicher Authentifizierung der Chipkarte das aktuelle Ticket des Fluggastes in der Lufthansa-Datenbank abfragen und die zugehörige Boardingkarte ausstellen. Mit Hilfe eines Touchscreens und eines Druckers kann ein Benutzer so unter anderem einem Sitzplatz wählen und weitere Informationen zum Flug ausdrucken.

Stärken

Dieses Verfahren ist sehr datensparsam, daher ist es nicht nur effizient und schnell, sondern es schützt auch die Privatsphäre des Fluggastes. Es werden nämlich außer der Kartenseriennummer, deren Personenbezug nur der Lufthansa-Server kennt, keine weiteren personenbeziehbaren Daten ausgetauscht. Alle weiteren personenbezogenen Daten werden im sicheren Firmennetzwerk verwaltet. Da alle Terminals in diesem Netz verbunden sind, sind alle notwendigen Informationen stets verfügbar, ohne auf die Speicherkapazität der Karte zurückgreifen zu müssen. Weder das Abhören der Kommunikation, noch das Auslesen einer Karte bringt dem potentiellen Angreifer einen Nutzen.

Das Identifikationsverfahren ist in dem Sinn sicher, als das Terminal sich darauf verlassen kann, bei erfolgreicher Authentifizierung die richtige Chipkarte zu haben. Weder druckt es aus Versehen Passagieren falsche Boardingkarten aus, noch kann ein gefälschter Sender, etwa eine andere Chipkarte, sich die Boardingkarte eines anderen Fluggastes (auf dessen Kosten) ausdrucken lassen.

Schwächen

Da das Terminal sich gegenüber der Chipkarte nicht authentifiziert, ist theoretisch ein Man-in-the-middle-Angriff möglich, der allerdings im praktischen Ergebnis wertlos ist und daher keine Schwäche des Systems darstellt. Ein gefälschtes Terminal (etwa ein prozessorbehafteter Aufsatz vor dem Kartenschlitz des Lufthansaterminals) könnte vom Terminal eine Zufallszahl abrufen, diese an die Chipkarte weiterleiten und dort verschlüsseln lassen. Die verschlüsselte Zufallszahl würde der Angreifer von der Chipkarte an das Terminal weiterleiten und dort das Ausdrucken der Boardingkarte veranlassen. Allerdings könnte dieser Eindringling nichts damit anfangen: Entweder das Terminal druckt die Boradingkarte aus, dann hat sie der berechtigte Fluggast in der Hand. Oder er fängt den Ausdruck der Boardingkarte ab, um sie bei späterer Gelegenheit (bis zum Abflug) einer anderen Person (dem Angreifer) auszugeben. Der aber stünde nun in Person am Abfluggate und würde dort vermutlich dem empörten berechtigten Fluggast begegnen, der nach seiner Boardingkarte verlangt. Der technische Aufwand eines Man-in-the-middle-Angriffs ist sehr hoch, dagegen ist ein solcher Diebstahl fast sicher wirkungslos.

Die Schlüssel auf der Chipkarte und im Lufthansa-Netz bilden allerdings einen Angriffspunkt. Da sich der individuelle symmetrische Schlüssel auf der Karte befindet, muss der Besitzer der Karte besonders darauf aufpassen. Wer die Karte (bzw. die darauf gespeicherte Seriennummer und den symmetrischen Schlüssel) hat, den erkennt das Lufthansaterminal als berechtigt an und stellt ihm zu vorhandenen Flugtickets Boardingkarten aus. Allerdings müsste ein unberechtigter Boardingkartenbesitzer persönlich durch das Boarding hindurch und damit rechnen, dort dem empörten Fluggast zu begegnen.

Der Hauptschlüssel muss im Lufthansasystem sicher aufbewahrt werden, denn mit diesem sind sämtliche Chipkarten aller Fluggäste, die an dieses System nutzen, imitierbar.

8. Pay-TV (Bezahl-Fernsehen)

Anwendung

Das öffentlich-rechtliche und privat-rechtliche Programm-Fernsehen erreicht alle Fernesehempfänger in gleicher Weise und wird darum Free-TV genannt. Als Alternative dazu wird das so genannten Pay-TV verschlüsselt ausgesendet, so dass sein Signal zwar ebenfalls von allen Geräten empfangen wird, aber nur von solchen Geräten dekodiert werden kann, welche über einen

frei geschalteten Entschlüsselungsschlüssel verfügen. Der zugehörige Entschlüsselungsschlüssel wird nur denjenigen Nutzern zugänglich gemacht, die dafür bezahlt haben.

Pay-TV hat zwei Anwendungsvarianten: Beim so genannten „Video on Demand“ entscheidet sich ein Kunde für einen Film, bezahlt für ihn und bekommt bei Zahlungseingang einen filmspezifischen Entschlüsselungsschlüssel (Service Key) mit den Fernsehdatenstrom ausgehändigt. Mit dem kann er dann die im Datenstrom eingebetteten verschlüsselten Nutzdaten entschlüsseln und unmittelbar konsumieren. Er braucht auf keine Programmzeiten zu warten. Das so genannte „Pay-per-View“ dagegen ist ein Programmfernsehen, das flächendeckend ausgestrahlt wird (Broadcast), allerdings verschlüsselt. Nutzer können sich daraus Filme, Zeiten oder Serien auswählen, bezahlen gezielt nur für diese und bekommen für diese die zugehörigen Entschlüsselungsschlüssel mit dem Fernsehdatenstrom ausgehändigt. Bekannteste Anbieter in Deutschland sind „Premiere“, „Arena“ und „Kabel Digital Home“.

Ziel der Pay-TV-Verschlüsselung

Das Ziel der Pay-TV-Verschlüsselung ist es, dass diejenigen Nutzer, die bezahlt haben, Zugang zum Klartext des Fernsehstroms erhalten, während die anderen, die nicht bezahlt haben, den verschlüsselten Fernsehstrom nicht entziffern können und nur Bild- und Tonsalat empfangen. Die Entschlüsselung kann auf einen Film, auf eine Serie, auf einen Sender, auf eine Zeitperiode oder auf eine Kombination davon begrenzt werden.

Dazu haben die Fernsehnutzer von ihrem Anbieter eine Smartcard erhalten, die einen kundenspezifischen Nutzerschlüssel enthält. Die Smartcard wird in eine Set-Top-Box gesteckt, die das verschlüsselte Fernsehsignal empfängt, ggf. entschlüsselt und dann im Klartext an das Ausgabegerät weitergibt.

Der Fernsehstrom ist portionsweise symmetrisch verschlüsselt. Der zugehörige symmetrische Schlüssel ist seinerseits verschlüsselt und in regelmäßigen Abständen, alle paar Sekunden, in den verschlüsselten Fernsehstrom eingestreut. Mit Hilfe seiner Smartcard kann der Nutzer diese Schlüsselinformation auslesen, dechiffrieren, und mit der so gewonnen Schlüsselinformation schließlich den Fernsehstrom entschlüsseln. Damit die Dechiffrierfähigkeit der Nutzer auf spezifische Teile des Fernsehstroms und zeitlich begrenzt werden kann, wird die Schlüsselinformation mehrfach verkettet im Fernsehstrom verschlüsselt.

Technische Grundlagen

Fernsehinformation werden in so genannte digitale Transportdatenströme kodiert und diese als elektromagnetische Signale ausgestrahlt. Durch Aufteilung der Frequenzbereiche kommt dabei pro Sender ein Frequenzbereich zwischen 8 MHz (Kabel) und 40 MHz (Satellit) zustande, das führt zu einer Übertragungsrate für die Nutzdaten von über 50 MB/s. Daran haben die Videodaten den

Löwenanteil, die Audiodaten belegen nur etwa 3% davon. Bis zu weiteren 4% der Daten belegen die Schlüsselinformationen: In einer Rate zwischen 100 KB/s und 2 MB/s werden die relativ seltenen, aber umfangreichen „Entitlement Management Messages (EMM)“ eingespielt, die nutzerspezifische Schlüssel enthalten (Service Keys) und etwa 10.000 Nutzer pro Sekunde mit Schlüsseln versorgen können; nur einen Bruchteil davon kosten die viel häufiger eingespielten „Entitlement Control Messages (ECM)“, die den alle paar Sekunden wechselnden Datenschlüssel „Control Word (CW)“ enthalten, und zwar mit dem Service Key verschlüsselt. Wenn also ein Angreifer das relativ leichtgewichtige Control Word (48 Bits) knacken kann, so nützt es ihm nichts, denn in wenigen Sekunden hat es gewechselt.

Zur Dekodierung und Entschlüsselung nutzen die Empfänger spezifische Geräte, so genannte Set-Top-Boxen, die das verschlüsselte Signal vom Satelliten, vom Kabel oder von der terrestrischen Antenne empfangen, entschlüsseln und an ein Ausgabegerät (Bildschirm und Lautsprecher) weiterleiten.

Der Fernsehdatenstrom ist nach dem ISO-Standard MPEG2 kodiert. MPEG steht für „Moving Pictures Expert Group“, das ist die Gruppe, die auch den MP3-Standard geschrieben hat. Die Verschlüsselung erfolgt ebenfalls nach dem MPEG2-Standard, allerdings nach spezifischen Regeln des so genannten „Digital Video Broadcast (DVB)“, an das sich alle Video- und Pay-TV-Anbieter halten. Unter DVB-C, DVB-T, bzw. DVB-S sind die Übertragungsprotokolle für das Kabel (C), die terrestrische Antenne (T), bzw. den Satelliten (S) standardisiert.

Eingesetzte Algorithmen

Unter DVB-CI (Common Interface) ist ein symmetrisches Verschlüsselungsverfahren standardisiert, das dort unter dem Namen „Common Scrambling Algorithm (CSA)“ im Detail ausgeführt ist. Der CSA ist eine konkrete Ausführung für ein „Conditional Access System (CAS)“, da es die Entschlüsselung von spezifischen Teilen und Zeiten und damit einen „bedingten Zugriff“ (conditional access) zulässt. Das geschieht durch den Einsatz verketteter Schlüssel, die aufeinander verweisen. Der eigentliche Datenschlüssel ist das so genannte „Control Word“, das seinerseits mit einem „Service Key“ verschlüsselt im Sendestrom liegt, und auch der Service Key liegt für jeden Kunden mit seinem Kundenschlüssel verschlüsselt im Datenstrom. Die Schlüssellänge des symmetrischen Control Word beträgt nominell 64 Bits, praktisch aber nur 48 Bits, da 16 von den 64 Schlüsselbits aus anderen Schlüssel-Bits berechnet werden können.

Der Datenstrom wird zweimal mit verschiedenen Verfahren, aber mit demselben Schlüssel (dem Control Word) verschlüsselt. Bei der Verschlüsselung wird erst eine Blockchiffre (Verschlüsselung ganzer Blocks à 188 Bits), dann eine Stromchiffre (Bit für Bit-Verschlüsselung) eingesetzt, bei der Entschlüsselung umgekehrt. Das Ziel der doppelten Verschlüsselung ist es, den Klartext mehr zu verwirren und damit die Analyse des Kryptogramms noch mehr zu erschweren.

Interessant ist die Konstruktion des relativ selten wechselnden Service Keys. Der Service Key ist Kundengruppen zugeordnet. Für jede Kundengruppe wird ein Gruppenschlüssel berechnet, in den jeder Kundenschlüssel als Parameter eingeht (z.B. ein Hash über die Kundenschlüssel). Alle Gruppenschlüssel, die auf eine Sendung passen, gehen in den Service Key ein (z.B. ein Hash über die Gruppenschlüssel). Der Service Key braucht also nur dann ausgetauscht zu werden, wenn ein Kunde oder eine ganze Kundengruppe zur Laufzeit der Sendung aus- oder neu einsteigt. Alternativ zum Austausch eines Service Key können weitere Service Keys erzeugt werden, die dasselbe Control Word in anderen ECMs verschlüsseln, die dem Datenstrom zusätzlich hinzugefügt sind. Wie das genau funktioniert, ist ausführlich in [Reimers 2004] und [Weinmann, Wirt 2004] beschrieben.

Ablauf

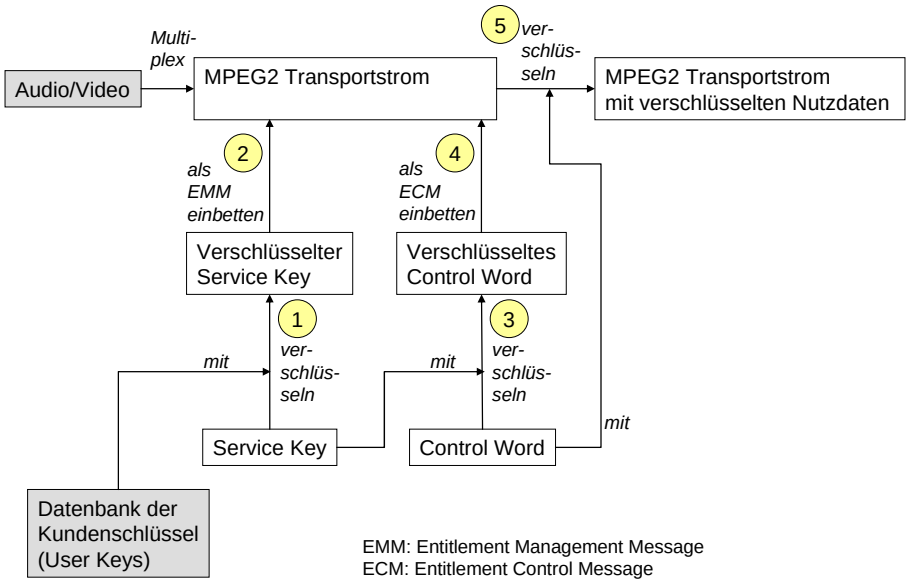

Abb. 8.1: Common Scrambling Algorithm (CSA) – Verschlüsselung

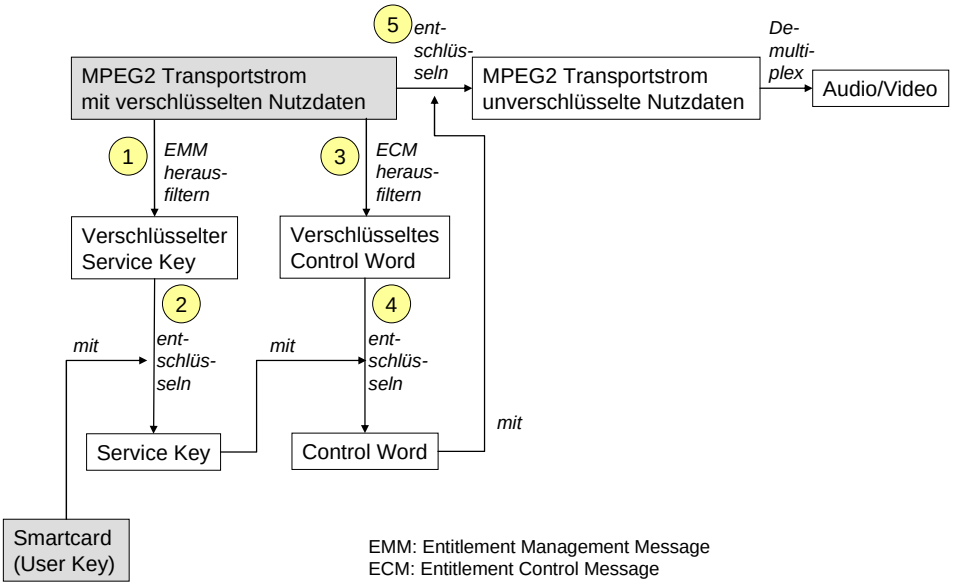

Abb. 8.2: Common Scrambling Algorithm (CSA) – Entschlüsselung

Der Common Scrambling Algorithm (CSA) zur Ver- und Entschlüsselung von Pay-TV-Sendedaten funktioniert wie folgt.

Der Betreiber eines Pay-TV-Senders übergibt jedem seiner Kunden eine Smartcard, auf der je ein individueller symmetrischer Kundenschlüssel (User Key) abgespeichert ist. Diese Karte bleibt dem Kunden mit demselben Kundenschlüssel so lange erhalten, wie er Kunde dieses Betreibers ist. Der Betreiber hält für den Sendeprozess eine Datenbank mit allen Kundenschlüsseln vor. Die Datenbank enthält darüber hinaus eine Zuordnung der Kunden zu Kundennutzungsgruppen und zu den Zugriffsrechten der Nutzungsgruppen. Beispielsweise hat Alice ein Monatsabonnement für Mai 2007 und gehört daher in die Nutzungsgruppe aller Mai-2007-Abonnementen, während Bob den Film „Ben Hur“ gebucht hat, der am 7. Mai 2007 um 20:15-23:50 ausgestrahlt wird, er ist daher der Gruppe der „Ben Hur“-Kunden zugeordnet. Wenn nun „Ben Hur“ am 7. Mai 2007 um 20:15 anläuft, dann haben Alices und Bobs Nutzungsgruppe Zugriff. Wenn der Film um 23:50 zu Ende geht, verliert Bobs Gruppe das Zugriffsrecht, während Alices Gruppe ihr Zugriffsrecht auf die Nachfolgesendung behält.

Das Nutzungsrecht wird wie folgt vergeben:

Zunächst kodiert der Sender den Audio/Video-Datenstrom (das sind die Nutzdaten) in ein MPEG2-Format, den so genannten MPEG2-Transportdatenstrom. Dieser hat Platz für die Nutzdaten und zusätzliche weitere Informationen.

In Schritt 1 erzeugt der Sender für alle berechtigten Nutzungsgruppen zusammen einen Service Key (alternativ: für jede Nutzungsgruppe einen eigenen Service Key) und verschlüsselt diesen genauso oft, wie es berechtigte Kunden gibt, und zwar für jeden Kunden mit dessen Kundenschlüssel, den der Sender der Kundendatenbank entnimmt. Das Ergebnis ist eine Menge von so genannten „Entitlement Management Messages (EMM)“, jede EMM enthält 100-1000 Pakete, jedes Paket enthält bis zu 256 Kundeninformationen. In Schritt 2 werden die EMMs in einer Datenrate zwischen 100 KB/s und 2 MB/s neben den Nutzdaten in den MPEG2-Transportstrom eingefügt.

In Schritt 3 wird ein 64-Bit Datenschlüssel, das so genannte Control Word erzeugt, und zwar alle paar Sekunden ein neues. Dann erzeugt der Sender so genannte „Entitlement Control Messages (ECM)“, indem er das Control Word mit dem Service Key verschlüsselt. Wenn mehrere Service Keys verwendet werden, dann wird mit jedem Service Key ein eigenes ECM erzeugt, das etwa 400-1000 Bits lang ist. Ein ECM-Generator produziert etwa 30 ECMs pro Sekunde und fügt sie in Schritt 4 in einer durchschnittlichen Datenrate von 20 KB/s neben den Nutzdaten und den EMMs in den MPEG2-Transportstrom ein.

Da es viel weniger Service Keys als Kundenschlüssel gibt, werden zwar pro Takt weniger ECMs als EMMs gebraucht, dafür werden die ECMs wesentlich häufiger in den Transportstrom gestreut, da der Inhalt der ECMs, das Control Word, häufig wechselt, während die Service Keys nur beim Wechsel von Nutzungsberechtigungen wechselt.

In Schritt 5 schließlich verwendet der Sender das in Schritt 3 erzeugte Control Word und verschlüsselt damit die Nutzdaten und bettet sie neben den EMMs und ECMs in den MPEG2-Transportstrom ein.

Derselbe Datenstrom kann mit verschiedenen Service Keys verschlüsselt werden, um damit verschiedene Nutzungsgruppen auseinander zu halten. Beispielsweise kann ein Film an eine Kundengruppe frei geschaltet werden, die ein Monatsabonnement für Filme hält (Alices Gruppe). Gleichzeit kann derselbe Film für eine andere Kundengruppe frei geschaltet werden, die nur diesen einen Film gebucht hat (Bobs Gruppe). Die Inhaltsdaten werden für beide Nutzergruppen mit demselben Control Word verschlüsselt, das im selben Takt für alle Nutzungsguppen wechselt. Aber für Alices Nutzungsgruppe wird das Control Word in einer eigenen Control Message ECM1 mit einem gruppenspezifischen Service Key SK1 verschlüsselt. Für Bobs Nutzungsgruppe dagegen wird das Control Word in einer anderen Control Message ECM2 mit einem gruppenspezifischen Service Key SK2 verschlüsselt. Beide Typen von Control Messages werden dem Datenstrom zugefügt. Die Anzahl der Management Messages EMM ändert sich dadurch nicht. Der Sender muss lediglich dafür sorgen, dass die richtigen Service Keys mit den richtigen Kundenschlüsseln verschlüsselt werden. Aber nach wie vor gibt es für jeden Kunden einen eigene Kundeninformation in einer EMM.

Bei der Entschlüsselung geht die Set-Top-Box des Kunden in umgekehrter Reihenfolge vor: Sie entnimmt den Kundenschlüssel der eingelegten Smartcard, filtert aus dem MPEG2-Transportstrom eine passende EMM und entschlüsselt diese. Dann entnimmt sie der EMM den Service Key, filtert aus dem MPEG2-Transportstrom eine passende ECM und entschlüsselt diese. Dem entschlüsselten ECM entnimmt sie nun das Control Word und entschlüsselt damit die Nutzdaten, die die Set-Top-Box nun dem Fernseher zur Ausgabe zuleiten kann.

Die Set-Top-Box liest übrigens kontinuierlich die ECMs aus, da die darin enthaltenen Control Words laufend wechseln. Die abgelaufenen Control Words werden gelöscht, die aktuell gültigen werden zur Entschlüsselung der Nutzdaten verwendet, die im Vorrat erzeugten Control Words werden aufgehoben, bis sie gültig werden und das bis dahin gültige Control Word ablösen. Die EMMs braucht die Set-Top-Box erst dann wieder herauszufiltern und zu entschlüsseln, wenn ein bisher gültiger Service Key ausgelaufen ist, bspw. bei einem neuen Abonnement.

Stärken

Das Verschlüsselungsverfahren CSA ist leichtgewichtig und erlaubt daher die synchrone Verschlüsselung von über 50 MB/s, das reicht für Fernsehsignale inklusive der erforderlichen Schlüsselinformationen, die etwa 4% der Nutzdaten kosten. Theoretische Analyseansätze haben bisher noch zu keinem praktischen Erfolg geführt.

Das Verschlüsselungsverfahren ist durch die mehrfache Verkettung von Schlüsseltypen und ihre Einfügung in den Datenstrom einer Sendung so flexibel gehalten, dass es einen bedingten Zugriff („conditional access“) von Nutzern und Nutzergruppen im praktischen Betrieb erlaubt, d.h. der verschlüsselte Sendestrom kann gleichzeitig von den verschiedensten Nutzungsgruppen, begrenzt auf individuelle Zeiten und Teile des Stroms, entschlüsselt werden. Das liefert die technische Geschäftsgrundlage sowohl für Angebote zum „Video-on-Demand“ als auch zur „Pay-per-View“-Variante des Pay-TV. Für das Pay-per-View wiederum können verschiedene Geschäftstypen gleichzeitig unterstützt werden: zeitliche Abos, Buchung von Serien mit mehreren Sendungen oder einzelner Sendungen.

Der Vorteil in der Trennung von Datenschlüsseln (Control Word), Service Keys und Kundenschlüsseln liegt darin, dass zwei Rhythmen von Schlüsselwechseln unabhängig voneinander ausgeführt werden können: Beim Wechsel von Kundengruppen und Gruppenrechten wird der Service Key ausgetauscht, herausgenommen oder neu hinzugefügt. Ansonsten bleibt er stabil. Unabhängig davon kann aus Sicherheitsgründen der Datenschlüssel ausgetauscht werden, das geschieht in der Praxis im Sekundenrhythmus.

Schwächen

Bisher sind keine Schwächen in der Stromverschlüsselung gefunden worden, die es ermöglichen, das Control Word ohne Zugangsschlüssel auszulesen. Auch die in der Blockverschlüsselung analysierten Schwächen reichen nicht aus, das Control Word im CSA, so wie er in der Praxis eingesetzt wird, ohne Zugangsschlüssel auszulesen.

Allerdings liefert die Schlüssellänge des CSA eine theoretische Schwachstelle. Die CSA-Schlüssel haben zwar nominell eine Länge von 64 Bit, in der Praxis werden sie aber entsprechend dem „DVB Conformance Mechanism" auf 48 Bit reduziert, indem 16 Bits aus den anderen 48 Bits des Schlüssels berechnet werden. Der Grund dafür dürfte in Exportbeschränkungen liegen, die für starke Kryptographiesysteme zum Zeitpunkt der Standardfestlegung bestanden, z.B. in den USA. Der Aufwand für einen Brute-force-Angriff, bei dem alle möglichen Schlüssel ausprobiert werden, verringert sich dadurch erheblich. Kommt dazu noch eine fehlerhafte Implementierung einer wichtigen Komponenten wie etwa des Control Word Generators, dann kann ein Brute-force-Angriff sogar schon in weniger als zwei Minuten zur unberechtigten Entschlüsselung des alles entscheidenden Control Word führen. Dagegen wirkt allerdings der rasche Wechsel des Control Words alle paar Sekunden, so wie es das Verfahren in der Praxis tut.

Schwächen hat das Verfahren daher weniger in der verwendeten Verschlüsselungsart, als im organisatorischen Bereich des Schlüsselmanagements. Ein Gruppenangriff kann dadurch erfolgreich sein, dass ein berechtigter Nutzer das Control Word ausliest und unter einer Gruppe von unberechtigten Nutzern in Echtzeit verteilt, die dann ohne zu bezahlen mit sehen können.

Literaturhinweise

Bewic, Simon (1998): Descrambling dvb data according to etsi common scrambling specification. Technical Report GB2322994A, GB2322995A, UK Patent Application, 1998.

Blake-Wilson, S.; et al. (2006): Transport Layer Security (TLS) Extensions. Internet Official

Breitschaft, Markus; Krabichler, Thomas; Stahl, Dr. Ernst; Wittmann, Georg (2005): Sichere Zahlungsverfahren für E-Government, ibi research an der Universität Regensburg GmbH In E-Government-Handbuch, Bundesamt für Sicherheit in der Informationstechnik (BSI) http://www.bsi.bund.de/fachthem/egov/download/4_Zahlv.pdf

Bundesamt für Sicherheit in der Informationstechnik (2003): GSM-Mobilfunk – Gefährdungen und Sicherheitsmaßnahmen / Referat III. Bonn, 2003 http://www.bsi.de/literat/doc/gsm/index.htm

CENELEC. Common interface specification for conditional access and other digital video broadcasting decoder applications. Technical Report EN 50221, Technical Committee TC 206, October 1997.

Detken, Kai-Oliver (2006): WLAN-Sicherheit von WEP bis CCMP, in DACH Security 2006, Syssec, 2006, 187-201

Dierks, Tim; Riscorla, Eric (2006): The Transport Layer Security (TLS) Protocol, Version 1.1. Internet Official Protocol Standard (STD 1) RFC 4346 (obsoletes RFC 2246, June 2003). April 2006, 87 pages. http://www.ietf.org/html.charters/tls-charter.html

Eckert, C. (2006): IT-Sicherheit. Konzepte-Verfahren-Protokolle, Odenburg Verlag, 2006

ekey biometric systems GmbH - Produktübersicht: http://www.ekey.at/products/products.asp

Esslinger, Bernhard; Müller, Maik (1997): Secure Sockets Layer (SSL) Protokoll – Sichere Internetkommunikation mittels SSL und Sicherheits-Proxy. DuD 12/1997, 691-697.

Freier, Alan; Karlton, Philip; Kocher, Paul (1996): The SSL Protocol, (Secure Socket Layer), Version 3.0. Internet Draft, 18 Nov 1996, 63 pages, draft-freier-ssl-version3-02.txt. Deleted from Internet-drafts server. Now included in Transport Layer Security (TLS) standardization efforts of the IETF.

Hill Keith, Bormans, Jan (2002): Mpeg-21 overview v.5. Technical Report JTC1/SC29/WG11/N5231, ISO/IEC, Requirements Group, October 2002.

Hirsch, Frederick J. (1997): Introducing SSL and Certificates Using SSLeay. In Web Security – A Matter of Trust (World Wide Web Journal). O'Reilly. Sebastopol, 1997, 141-173.

IEEE 802.11-Gruppe: http://grouper.ieee.org/groups/802/11/ [10.10.2006]

ISO/IEC 9594-8, ITU X.509 (1988/92): Information technology – Open Systems Interconnection – The Directory – Authentication Framework. 1988/1993(E).

Lochter, Manfred; Schindler, Werner (2006): „Missbrauch von PIN-gestützten Transaktionen mit ec- und Kreditkarten aus Gutachtersicht“. In: Multimedia und Recht, 5/2006, 292-297

Meyer, Carsten (1996): Nur Peanuts, der Risikofaktor Magnetkarte, In: c’t Juli 1996, S. 94, http://www.heise.de/ct/96/07/094/

Needham, R.M.; Schroeder, M.D (1978): Using Encryption for Authentication in Large Networks. In: Comm. ACM 21 No.12, 1978, 993-999.

Protocol Standard (STD 1) RFC 4366. Updates (but does not replace RFC 4346). April 2006, 30 pages. http://www.ietf.org/html.charters/tls-charter.html

Rankl, Wolfgang; Effing, Wolfgang (2002): Handbuch der Chipkarten, Aufbau – Funktionsweise - Einsatz von Smart Cards, Verlag Carl Hanser, München Wien.

Reimers, Ulrich (2004): DVB, The Family of International Standards for Digital Video Broadcasting. Springer, second edition, September 2004.

Steiner, J.G.; Neumann, C.; Schiller, J.I. (1988): Kerberos: An Authentication Service for Open Network Systems. USENIX Winter Conference, Dallas Texas, 9-12 Feb 1988. Proceedings pp. 191-202. (Project Athena, MIT, Boston MA).

Taylor, Jim (2000): DVD Demystified. McGraw-Hill Professional, second edition, December 2000

The Cable Guy (2005): Überblick zu Wi-Fi Protected Access 2 (WPA2) Veröffentlicht: 06. Mai 2005, http://www.microsoft.com/germany/technet/datenbank/articles/600761.mspx

The Cable Guy (2004): Datenverschlüsselung und -integrität mit WPA, Veröffentlicht: 01. Nov 2004, http://www.microsoft.com/germany/technet/datenbank/articles/600513.mspx, Stand:

Walke, B (2000): Mobilfunknetze und Ihre Protokolle. Band 1, Teubner Verlag, 2000

Weinmann, Ralf-Philipp and Wirt, Kai (2004). Analysis of the dvb common scrambling algorithm. In Proceedings of the 8th IFIP TC-6 TC-11 Conference on Communications and Multimedia Security (CMS 2004). Springer LNCS.

Bisher erschienen

Arbeitsberichte des Fachbereichs Informatik

(http://www.uni-koblenz.de/fb4/publikationen/arbeitsberichte)

Rüdiger Grimm, Helge Hundacker, Anastasia Meletiadou: Anwendungsbeispiele für Kryptographie, Arbeitsberichte aus dem Fachbereich Informatik, 2/2007

Anastasia Meletiadou, J.Felix Hampe: Begriffsbestimmung und erwartete Trends im IT-Risk-Management, Arbeitsberichte aus dem Fachbereich Informatik, 1/2007

„Gelbe Reihe“

(http://www.uni-koblenz.de/fb4/publikationen/gelbereihe)

Lutz Priese: Some Examples of Semi-rational and Non-semi-rational DAG Languages. Extended Version, Fachberichte Informatik 3-2006

Kurt Lautenbach, Stephan Philippi, and Alexander Pinl: Bayesian Networks and Petri Nets, Fachberichte Informatik 2-2006

Rainer Gimnich and Andreas Winter: Workshop Software-Reengineering und Services, Fachberichte Informatik 1-2006

Kurt Lautenbach and Alexander Pinl: Probability Propagation in Petri Nets, Fachberichte Informatik 16-2005

Rainer Gimnich, Uwe Kaiser, and Andreas Winter: 2. Workshop "Reengineering Prozesse" – Software Migration, Fachberichte Informatik 15-2005

Jan Murray, Frieder Stolzenburg, and Toshiaki Arai: Hybrid State Machines with Timed Synchronization for Multi-Robot System Specification, Fachberichte Informatik 14-2005

Reinhold Letz: FTP 2005 – Fifth International Workshop on First-Order Theorem Proving, Fachberichte Informatik 13-2005

Bernhard Beckert: TABLEAUX 2005 – Position Papers and Tutorial Descriptions, Fachberichte Informatik 12-2005

Dietrich Paulus and Detlev Droege: Mixed-reality as a challenge to image understanding and artificial intelligence, Fachberichte Informatik 11-2005

Jürgen Sauer: 19. Workshop Planen, Scheduling und Konfigurieren / Entwerfen, Fachberichte Informatik 10-2005

Pascal Hitzler, Carsten Lutz, and Gerd Stumme: Foundational Aspects of Ontologies, Fachberichte Informatik 9-2005

Joachim Baumeister and Dietmar Seipel: Knowledge Engineering and Software Engineering, Fachberichte Informatik 8-2005

Benno Stein and Sven Meier zu Eißen: Proceedings of the Second International Workshop on Text-Based Information Retrieval, Fachberichte Informatik 7-2005

Andreas Winter and Jürgen Ebert: Metamodel-driven Service Interoperability, Fachberichte Informatik 6-2005

Joschka Boedecker, Norbert Michael Mayer, Masaki Ogino, Rodrigo da Silva Guerra, Masaaki Kikuchi, and Minoru Asada: Getting closer: How Simulation and Humanoid League can benefit from each other, Fachberichte Informatik 5-2005

Torsten Gipp and Jürgen Ebert: Web Engineering does profit from a Functional Approach, Fachberichte Informatik 4-2005

Oliver Obst, Anita Maas, and Joschka Boedecker: HTN Planning for Flexible Coordination Of Multiagent Team Behavior, Fachberichte Informatik 3-2005

Andreas von Hessling, Thomas Kleemann, and Alex Sinner: Semantic User Profiles and their Applications in a Mobile Environment, Fachberichte Informatik 2-2005

Heni Ben Amor and Achim Rettinger: Intelligent Exploration for Genetic Algorithms – Using Self-Organizing Maps in Evolutionary Computation, Fachberichte Informatik 1-2005