Inhaltsverzeichnis

1 Einleitung ____ *2*

2 Geschichte ____ *3*

2.1 Maria Stuart ____ **3**

2.2 Die eiserne Maske ____ **4**

2.3 Vigenere und Charles Babbage ____ **4**

2.4 Die Enigma ____ **5**

2.5 Die Navajo Code Sprecher ____ **6**

2.6 PGP ____ **6**

3 Grundlagen der Kryptographie ____ *7*

3.1 Gebiete ____ **7**

3.2 Terminologie, Grundlagen und Protokolle ____ **8**

3.3 Kryptosysteme ____ **10**

3.4 Grundlegende Angriffsarten auf Verschlüsselungen ____ **11**

3.5 Leitsätze und Maximen für gute Algorithmen ____ **13**

3.6 Algorithmen und deren Modes ____ **13**

3.7 Informationstheorie ____ **18**

3.8 Komplexitätstheorie ____ **18**

4 Beschreibung des Data Encryption Standards (DES) ____ *19*

4.1 Geschichte und Entwicklung ____ **19**

4.2 Einführung ____ **20**

4.3 Das Feistelnetzwerk ____ **21**

4.4 Die interne Blockchiffre F ____ **21**

4.5 Die Schlüsseltransformation ____ **23**

4.6 Sicherheitsaspekte zum DES ____ **23**

5 Beschreibung des RSA-Algorithmus ____ *26*

5.1 Geschichte von RSA ____ **26**

5.2 Mathematische Beschreibung ____ **26**

6 Anhang ____ *Fehler! Textmarke nicht definiert.*

6.1 Literaturverzeichnis: ____ Fehler! Textmarke nicht definiert.

6.2 Einverständniserklärung des Schülers ____ Fehler! Textmarke nicht definiert.

1 Einleitung

In der heutigen Zeit wird die Sicherheit und Privatsphäre am Computer immer wichtiger. Die Welt der Kryptographie ist ein Thema, das in viele wichtige andere Forschungsbereiche eindringt sowie aus ihr neue Ideen empfängt. Ich entdeckte dieses Gebiet zufällig, da ich mich für Mathematik interessiere, da diese heutzutage eine wichtige Basis für die Kryptographie darstellt. Natürlich entstand diese Wissenschaft der Geheimnisbewahrung nicht aus dem Nichts. Sie hat heute eine schon mehrere Jahrtausende dauernde Entwicklung hinter sich, die aus den unterschiedlichsten Gründen angetrieben und inspiriert wurde.

Kryptographie wird von jedem unterschiedlich gesehen, schließlich ist es die Wissenschaft des Geheimnisses. Wenn Eltern vor ihren Kindern Sätze nicht aussprechen, statt dessen aber bestimmte Handbewegungen machen, benutzen sie Kryptographie, wenn man am Bankschalter seinen PIN eingibt und dann Geld bekommt benutzt man Kryptographie. Wenn ich einer vertrauten Person etwas zuflüstere, benutze ich wieder dieselbe Wissenschaft.
Wie man sieht ist das Benutzen dieses Werkzeugs fast alltäglich. Aber erst durch die rasante Verbreitung und Weiterentwicklung des Computers, wurde diese Lehre mit dem mysteriös klingendem Namen wichtig, und zwar nicht nur für Regierung und Militär, sondern auch für die Wirtschaft und private Personen. Zusätzlich zu dieser Erweiterung des Benutzerkreises wurde mittlerweile auch das Aufgabenfeld weiter gesteckt.
Mittlerweile ist das nicht mehr nur die Verschlüsselung von Nachrichten oder deren Brechen, was oft auch als Kryptologie bezeichnet wird. Es gehören jetzt dazu auch Verfahren wie Signaturen und Authentifikation. Dazu später mehr.

Trotz dieses alltäglichen Gebrauches wurde früher versucht, wie auch zum Teil jetzt noch, diese im öffentlichen Leben zu beschränken, bzw. sogar zu zensieren. Dies geschah, z.B. 1980 in Amerika, als von der NSA versucht wurde, sämtliche Erkenntnisse in diesem Teil der Wissenschaft, die dort gemacht wurden, kontrollieren zu können. Sie versuchten den Kongreß zu überzeugen, daß davon die nationale Sicherheit unter Umständen abhängen könnte. Es konnte zum Glück verhindert werden, daß die NSA die volle juristische Kontrolle über Publikationen auf diesem Gebiet erlangte. Dieser Versuch ist aber nur die Spitze des Eisbergs. Das ist eine starke Besonderheit auf diesem Gebiet. Einerseits ist in der Öffentlichkeit ein reges Interesse an der Kryptographie entstanden und diese wird auch aktiv betrieben, trotzdem ist man verglichen mit staatlichen Institutionen, um ca. 20 Jahre im Hintertreffen. Dies liegt auch daran, daß die NSA natürlich die Forschung länger betreibt, mehr Material zur Verfügung hat und nicht zuletzt der größte Arbeitgeber der Welt für Mathematiker ist. Als die NSA diese Kryptoregulierung versuchte anzutreiben, verstörte dies einige Bürgerrechtler so sehr, daß als letztendliche Reaktion Organisationen wie „Electronic Frontier Foundation“ gegründet wurde. So sehr dies die NSA in ein schlechtes Licht rückte, war die Praxis kryptographische Wissenschaft zu

verheimlichen oder im Interesse der nationalen Sicherheit zu benutzen, schon in früheren Jahrhunderten üblich.

2 Geschichte

2.1 Maria Stuart

Schon zu Zeiten von Maria Stuart wurden Briefe verschlüsselt, damit sie nur von den berechtigten Personen lesbar waren.
So der Briefwechsel zwischen Maria Stuart und ihren Anhängern während ihrer Gefangenschaft. Die Briefe wurden von Gilbert Gifford übermittelt. Antony Babington, der die Verschwörung zur Befreiung Maria Stuarts und der Ermordung Elisabeths I. leitete, verschlüsselte die Briefe, damit sie nur für Maria und ihre Anhänger lesbar waren.
Er verschlüsselte Zeichen und Wörter auf folgende Weise:

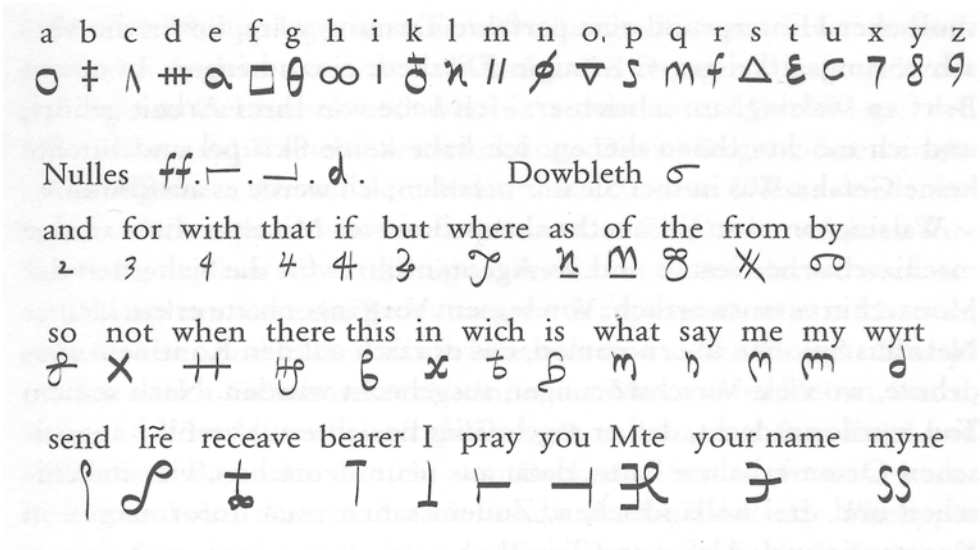


Bild 1 Geheimtextalphabet von Maria Stuart ([3], S.57)

Für die damalige Zeit war die eine verhältnismäßig starke Verschlüsselung. Allerdings gab es eine Tatsache die Babington und Maria Stuart nicht bewußt war. Gifford war ein Doppelagent, der für den englischen Minister Walsingham arbeitete. Dieser Minister war für die Sicherheit er Königin verantwortlich und beobachtete Maria schon länger. Er führte überall im Land Agentennetze, wodurch ihm nichts entging. Dieser hatte auch einen brillianten Kryptographen, Thomas Phelippe, als Geheimsekretär. Phelippe brach schließlich mit der Häufigkeitsanalyse die Verschlüsselung von Maria Stuarts Briefen. Walsingham benutzte sogar die Verschlüsselung um an die Daten der Mitverschwörer zu kommen, indem er sich selbst als Maria ausgab.
Durch die entschlüsselten Botschaften konnte man beweisen, daß Maria Stuart in das Babingtonkomplott verstrickt war und wurde deshalb durch das Henkersbeil am 8 Februar 1587 hingerichtet.

2.2 Die eiserne Maske

In der Geschichte gibt es noch weitere Erlebnisse bei denen die Kryptographie eine so starke Rolle spielte.
So auch bei der epischen Geschichte der Eisernen Maske. In den Zeiten des Absolutismus wurde bei staatlichen Stellen viel verschlüsselt, da aber die herkömmliche Verschlüsselung trotz einiger Modifikationen nicht sicher war (wie bei Maria Stuart gesehen), mußte man sich etwas neues einfallen lassen. Das war die homophone Verschlüsselung. Sie wurde auch in Frankreich benutzt, von den Rossignols entwickelt und erhielt durch ihre damalige kryptographische Stärke den Namen „Grosse Chiffre“. Nach ein paar Jahrhunderten konnte schließlich die Grosse Chiffre durch den französischen Offizier Bazeries gebrochen werden. Nach der vollständigen Entschlüsselung der grosssen Chiffre wurde dann auch unter anderem in einem Dokument das Geheimnis der Maske gelüftet. Man hatte ja früher vermutet, hinter der Maske verberge sich der Zwillingsbruder des damaligen Königs, der Erbstreitereien vermeiden wollte.
Es wurde der Maske sehr viel angedichtet, es gab den Roman von Alexandre Dumas sowie in der Neuzeit den Film „Die Maske“ mit Leonardo di Caprio. Aber es stellte sich heraus, daß es ein Kommandeur namens Vivien de Bulonde war. Dieser war General und hatte Befehl Stellung in einem Krieg zu halten. Durch das Kommen von gegnerischen Truppen wurde er so nervös, daß er floh und seine Soldaten im Stich ließ. Dadurch wurde der ganze Kriegszug von Piemont gefährdet. Wegen diesem feigen Verhalten befahl der König, Bolunde in die Festung Pignerole zu sperren und ihn eine eiserne Maske tragen zu lassen. Manche vermuten allerdings, daß das Dokument in dem die Identität der Maske enthüllt wurde, gefälscht wurde, da die Erklärung zu banal sei, und nur als Ablenkungsmanöver dient.

2.3 Vigenere und Charles Babbage

Mit der Zeit verloren alle Varianten von monoalphabetischen Codierungen ihre Sicherheit. Diese wurden nämlich am Fließband von den schwarzen Kammern geknackt. Dies waren Arbeitsgruppen von Kryptographen die mit der Haufigkeitsanalyse vertraut waren und für die lokalen Regierungen arbeiteten. Als Beispiel will ich die effizienteste schwarze Kammer anführen, die geheime Kabinettskanzlei von Wien. Briefe für Botschaften in Wien wurden beim Postweg automatisch über die Kabinettskanzlei umgeleitet. Dort wurden sie kopiert und zu ihrem ursprünglichem Empfänger geschickt. Diese Abschriften wurden dann an Kryptographen in kleinen Kammern weitergereicht, die diese dechiffrieren sollten.
Aus diesem Grund wurde die schon viel früher entwickelte, aber weil so schwierig einzusetzende, Vigenere Verschlüsselung, mehr und mehr benutzt. Weitere Informationen sind in [3] ausgearbeitet

Gegen sie waren die Kryptographen bis Babbage machtlos. Babbage war ein Universalgenie aus dem 19. Jahrhundert und ist auch für den ersten Entwurf eines modernen Computers bekannt. Mit der Zeit erwarb er sich einen

beeindruckenden Ruf als Kryptoanalytiker in der englischen Gesellschaft. Er amüsierte sich oft damit verschlüsselte Botschaften in Zeitungen zu dechiffrieren, die von jungen englischen Pärchen geschrieben wurden. Er selbst schuf mit dem Baron Lyon die Verschlüsselung „Playfair“, die sogar im Burenkrieg eingesetzt wurde. Erst durch einen nicht in der Kryptographie vorbelasteten Mann namens Thwaites, der die VigenereVerschluesselung unwissenderweise patentieren lassen wollte, versuchte Babbage diese zu knacken.
Als nämlich Babbage dem zuständigen Patentamt, an das Thwaites geschrieben hatte, sagte, daß diese Verschlüsselungsart nicht neu sei, forderte Thwaites ihn heraus, sie zu brechen. Auch wenn das gar nichts mit Babbages Aussage zu tun hatte, daß die Verschlüsselung alt sei, nahm er die Herausforderung an.
Er fand schließlich eine Möglichkeit das Verfahren zu knacken, und zwar die Periodenanalyse.

2.4 Die Enigma

1918 erwarb der deutsche Erfinder namens Scherbius sein erstes Patent für die Enigma. Es war nur eine kleine Kiste mit 12 kg Gewicht. Um 1924 konnte er die Maschine an das deutsche Militär verkaufen, das im ersten Weltkrieg eine kryptographische Niederlage erlitten hatte. So konnte Scherbius über 30000 Enigmas allein an das Militär verkaufen. Scherbius erlebte nicht mehr die späten Mißerfolge seiner Chiffriermaschine.
Deutschland hatte nun das kryptographisch sicherste System der Welt, und fast alle Alliierten bemühten sich nicht mehr die Nachrichten des, schon im ersten Weltkrieg geschlagenen, Deutschlands, zu analysieren. Nur Polen, das eine Invasion befürchten mußte, versuchte, die Enigma zu knacken. Nur durch den Verrat des Deutschen Hans Thilo Schmidt, der geheime Dokumente über die Enigma verkaufte, war es möglich mehr über die Technik herauszufinden. Er verkaufte die Dokumente an die Franzosen, die sie an die Polen gaben. Dadurch gelang es dann einem Mann namens Marian Rejewski die Enigma zu knacken. Er entwickelte die sogenannte Bombe, welche ein mechanisches Gerät war, das die Entschlüsselung der Enigma vereinfachte. Aber als die deutschen Ingenieure die Enigma noch mehr verfeinerten, war es dann aus für Rejewski. Er hatte keine Chance mehr. Deshalb gab Langer, der Vorgesetzte von Rejewski, den Engländern und Franzosen Dokumente und Bomben, im Kampf für das Knacken der Enigma. Als man dann in England neue Hoffnung schöpfte die Enigma knacken zu können, gründete man die „Government Code and Cipher School“ für die man jetzt auch viele Mathematiker rekrutierte. Einer der größten Kryptanalytiker dort war Alan Turing. Durch die Entdeckung von Cribs, bekannten Klartextstellen, die im Chiffretext stehen, war es schließlich möglich, die Enigma wieder zu knacken. Allerdings galt dies nicht für die Spezialversion der Marine.
Denn bei ihr gab es weniger mögliche Anhaltspunkte für Cribs.
Erst als man Schlüsselbücher der Marine sowie einige ihrer Enigmas erbeuten konnte, war es möglich, sie wieder knacken. Da man in der Lage war, die deutsche Enigma zu knacken, war es dann den Alliierten auch

möglich die Verschlüsselung der verbündeten Gegner Italien und Japan zu analysieren, sowie offen zu legen. Einer der Kryptoanalytiker, Stuart Milner-Barry, meinte, dass es noch nie einen Krieg gab, bei dem eine Seite ständig alle Geheimmeldungen des Gegners las.
Nach dem Krieg wurde Bletchley Park, wie man die „Government Code and Cipher School“ oft nannte, geschlossen. Sämtliche Dokumente und Bomben wurden vernichtet, und die Mitarbeiter zum Stillschweigen verpflichtet. Viele von ihnen wurden nach dem Krieg angefeindet, da sie nicht an der Front für ihr Vaterland gekämpft hatten. Erst in den 70er Jahren wurde die Arbeit von Bletchley Park veröffentlicht, und viele Mitarbeiter kamen endlich zu den verdienten Ehren. Für Alan Turing kam das jedoch zu spät. Nachdem seine Homosexualität enthüllt war, wurde er öffentlich gedemütigt und beging mit 42 Jahren Selbstmord.

2.5 Die Navajo Code Sprecher

Eine andere Möglichkeit, Nachrichten zu verschlüsseln, fand man in Amerika im Krieg gegen Japan. Es stellte sich heraus, dass elektronische Verschlüsselung ziemlich sicher war, aber sie war quälend unhandlich und langsam. Besonders auf den hart umkämpften Inseln im Pazifik war diese elektromechanische Verschlüsselung ungeeignet. So verfiel ein Ingenieur namens Philip Johnston auf die Möglichkeit, eine der Sprachen der amerikanischen Ureinwohner als Verschlüsselung zu benutzen. Er kannte die Sprache der Navajo und wußte wie undurchdringlich diese für Sprachunwissende war. So schlug er es der Regierung vor, die nach ein paar Tests das Konzept übernahm. Der Patriotismus der Navajos ging soweit, dass sich sogar Minderjährige in die Armee einschrieben, um als Navajo Code Sprecher fungieren zu können. Es entwickelte sich allerdings ein Problem. Technische Begriffe oder andere fremde Wörter mußten in die Navajosprache übersetzt werden, oder es mußte in ihrer Sprache zumindest eine ungefähre Entsprechung gefunden werden. Deshalb wurden Codebücher angelegt, in denen Worte wie „Jagdflugzeug“ als „Kolibri“ übersetzt wurde. Mit der Zeit verbreitete sich der Ruf der Codesprecher immer mehr und der Krieg machte die Navajos zu Helden. Nach dem Krieg wurde es wie bei BletchleyPark geheimgehalten, dass die Navajo als Codesprecher eingesetzt wurden. Erst 1986 wurde die Geheimhaltung aufgegeben, und die Navajos bekamen die würdige Ehrung.

2.6 PGP

Das erste mal, als die Kryptographie nicht gegen einen nationalen Feind, sondern zum Schutz der Privatsphäre benutzt wurde, beschreibt gleichzeitig die Entstehungsgeschichte von PGP (Pretty Good Privacy). Mittels RSA und DES war es mittlerweile möglich, eine Nachricht so zu verschlüsseln, dass sie kein anderer knacken konnte. Allerdings war RSA von RSA Data Security patentiert und wurde fast nur von Firmen genutzt. Phil Zimmermann allerdings, ein Mann der früher für die nukleare Abrüstung kämpfte, glaubte,

dass jeder das Recht hätte, seine Privatsphäre durch RSA geschützt zu sehen.
Deshalb schuf er die Software „Pretty Good Privacy“, kurz PGP. Als er mit der Programmierung der Software fertig war, ergaben sich zwei Probleme. Zuerst mußte Zimmermann hoffen, dass RSA DATA Security ihm eine kostenlose Lizenz gab, da RSA ihr Produkt war. Außerdem wurde 1991 ein Gesetzentwurf zu Verbrechensbekämpfung dem Senat vorgelegt. Dieser enthielt folgende Klausel:
„ Der Kongreß hält es für erforderlich, dass die Anbieter elektronischer Kommunikationsdienste und die Hersteller elektronischer Kommunikationsgeräte garantieren ihre Kommunikationssysteme so auszustatten, dass Klartextinhalte von Telefongesprächen, Datenübertragungen und anderen Mitteilungen den staatlichen Behörden zu Verfügung gestellt werden können, sofern hierzu eine gesetzliche Erlaubnis vorliegt.“([3], S. 363)
Es wurde nämlich befürchtet dass man durch neue Kommunikationtechniken kriminelle Machenschaften schwerer abhören kann. Darunter fiel auch, dass starke Verschlüsselung nicht benutzt werden sollte. Da Zimmermann nun fürchtete, dass PGP verboten werden würde, ließ er es von einem Freund im Usenet zum freien Download speichern. Als aber RSA DATA Security Zimmermann keine Lizenz gewährte, fingen die Probleme an. Bald tauchten Bundesbeamte als Ermittler auf, die ihn dem Export von Rüstungsgütern beschuldigten, da starke Verschlüsselung wie PGP bzw. RSA als Waffe deklariert ist, und diese dürfe nicht ohne Erlaubnis des Außenministeriums verbreitet werden. Die nächsten 3 Jahre lief dann ein Verfahren gegen Zimmermann, dass aber fallengelassen wurde, da es zu spät war die Verbreitung des Produktes zu vereiteln. Zudem hatte Zimmermann die Institution MIT-Press im Rücken, und bei einer Verurteilung würde das Mißtrauen der Bürgerrechtler nur weiter verstärkt werden. Nachdem Zimmermann PGP dann an Network Association verkauft hatte, erledigte sich auch die Patentfrage.

3 Grundlagen der Kryptographie

3.1 Gebiete

Die Gebiete, in welchen die Kryptographie angewandt wird, sind vielfältig. Zunächst einmal ist die Verschlüsselung von Daten, sowie das Brechen von Algorithmen bzw. verschlüsselten Nachrichten vorherrschend. Mittlerweile ist noch die Integration von Nachrichten, Verbindlichkeit sowie die Benutzererkennung hinzugekommen.
Bei der Verschlüsselung von Daten oder Nachrichten soll gewährleistet werden, daß nur die Person die Nachricht lesen kann, die auch den entsprechenden Schlüssel besitzt.
Beim Brechen einer verschlüsselten Nachricht handelt es sich genau um den umgekehrten Vorgang. Hier soll man ohne den Schlüssel, und vielleicht sogar ohne Kenntnis des Algorithmus, die verschlüsselte Nachricht entziffern und damit verstehen können. Wenn man versucht den Algorithmus zu brechen, sucht man nach einer Schwäche, womit sich die Entschlüsselung

von Nachrichten automatisieren läßt. Das Brechen von Nachrichten/Algorithmen nennt man auch Kryptoanalyse oder Kryptanalyse.

Die Integration von Nachrichten bedeutet
1. dass die Nachricht auf dem Weg vom Sender zum Empfänger nicht verfälscht wird.
2. dass die Nachricht nur von dem Absender stammt, von dem sie zu erwarten ist.

Verbindlichkeit bedeutet, dass der Sender später nicht leugnen kann, dass er die Nachricht geschickt hat.
Das Wort Benutzererkennung heißt, daß man sich als echt ausweisen kann, es ist also ein Integrationbeweis für Benutzer. Man sieht hier schon, daß diese neueren Gebiete relativ nah beieinander liegen.
In der Theorie mögen sich diese vier Gebiete ziemlich klein vorkommen, sie werden aber durch die zunehmende Vernetzung unserer Welt immer wichtiger.

3.2 Terminologie, Grundlagen und Protokolle

In der Kryptographie werden Verfahren und Verschlüsselungen durch Protokolle beschrieben. Die Protokolle besteht aus einer Folge von Aktionen, an welcher mindestens zwei oder mehr Parteien beteiligt sind. Eine Partei kann ein Mensch ein Computer oder etwas anderes sein. Zusätzlich muß ein Protokoll ein am besten klar definiertes Ziel haben.
Für das Erstellen und für die Durchführung von Protokollen gibt es mehrere Regeln. Allen Parteien muß das Protokoll und seine Aktionen im voraus bekannt sein. Auch darf im Protokoll kein Mißverständnis auftreten, denn jede Aktion ist klar definiert. Nicht zuletzt müssen sich alle Parteien an das Protokoll halten, welches auch für jede denkbare Situation eine bestimmte Aktion ausführt. Man sieht hier, dass das Wort Protokoll in vielen Punkten mit dem Algorithmus sympathisiert. Es ist allerdings allgemeiner gefaßt und braucht nicht notwendigerweise einen Computer.
Deshalb kann man sogar sagen dass ein kryptographisches Protokoll eine Spezialisierung ist, da es kryptographische Verfahren einsetzt, um Geheimhaltung zu bieten, sowie Vertrauen zwischen den Partnern zu schaffen.
Es ist viel schwieriger die Sicherheit eines Protokolls zu beweisen, als es zu brechen, denn ein Protokoll ist die Abstraktion eines Vorgangs, den wir im täglichen Leben vielleicht total unbewußt einsetzen.
In der Kryptographie ist es üblich bei der Festlegung der Parteien bestimmte Namen zu vergeben welche eine Bedeutung haben.

Person	Funktion
Alice	1.Beteiligte an allen Protokollen
Bob	2.Beteiligter an allen Protokollen
Carol	3.Beteiligte an großen Protokollen
Dave	4.Beteiligte an großen Protokollen
Eve	Lauscherin(eng: eavesdropper)
Mallory	Aktiver Angreifer auf das Protokoll (eng: malicious)
Trent	Vertrauenswürdiger Vermittler

Dies sind alle Personen die irgendwie an den Protokollen beteiligt sind, ob erlaubt oder nicht, ob aktiv oder nicht. Ein vertrauenswürdiger Vermittler ist eine unvoreingenommene dritte Partei, die das Protokoll durchführt, ohne das man Angst haben muß, dass diese Partei das Protokoll kompromittiert. Im wirklichen Leben sind Vermittler oft Anwälte, Notare oder Banken.
Ein Beispiel mit Alice, Bob und einer Bank
Der gewünschte Vorgang ist, dass Alice von Bob ein Auto kaufen will.

1. Alice unterschreibt einen Scheck und übergibt ihn der Bank.
2. Nachdem versichert ist, dass der Betrag auf Alice's Scheck auch wirklich vorhanden ist, beglaubigt die Bank den Scheck und gibt ihn Alice wieder.
3. Bob gibt Alice den Fahrzeugbrief, nachdem er sich von der Beglaubigung der Bank überzeugt hat und bekommt den Scheck.
4. Bob reicht den Scheck ein und bekommt das Geld.

Oft werden in Protokollen mit Vermittlern diese untergliedert bzw. als eigene Protokolle aufgeführt. Die Beglaubigung der Bank kann z.B. als eigenes Protokoll gewertet werden.
Natürlich stellt sich jetzt die Frage, wie man solche klar definierten Protokolle überhaupt erfolgreich angreifen kann. Dies liegt dann manchmal daran dass die Protokolle nicht sorgfältig genug erstellt werden.
Es gibt verschiedene Arten um ein Protokoll zu attackieren. Zunächst einmal die naheliegende Methode es zu belauschen (passiver Angriff). Hierfür ist Eve zuständig. Aktive Angreifer (die in der Theorie Mallory genannt werden) versuchen hingegen den Verlauf oder das Protokoll selbst zu ändern. Ihre Maßnahmen sind zum Beispiel, sich als andere Personen auszugeben, Informationen zu manipulieren, zu löschen oder zu erstellen bis hin zu der radikalsten Maßnahme, den Kommunikationskanal ganz zu unterbrechen. Ein aktiver Angreifer verändert also im Gegensatz zum passiven Angreifer das Protokoll. Es ist schwierig, die Sicherheit eines Protokolls zu gewährleisten, wenn die Mehrheit der Beteiligten eines Protokolls Betrüger sind. Gegen passive Angreifer sollte allerdings jedes Protokoll geschützt sein, sonst ist es praktisch wertlos.

Bei kryptographischen Protokollen werden Nachrichten oft verschlüsselt. Dabei nennt man E() die Verschlüsselungsfunktion, K den Schlüssel, M den Klartext, sowie C den Chiffretext. Dieser Vorgang sieht dann so aus

$$E_k(M) = C$$

In mathematischen Arbeiten kann es dann auch statt C ein C' geben, dies ist aber in der Kryptographie unüblich. Zur Entschlüsselung ist dann die Formel

$$D_k(C) = M$$

gültig. Bei symmetrischen Kryptosystemen sind die Funktionen E() und D() oft gleichwertig, deshalb schreibt man dann auch oft P() für beide Funktionen.
Ein Algorithmus ist die technische Beschreibung eines Protokolls.

Hier noch eine Anmerkung zu der elektronischen Verschlüsselung, der Verschlüsselung mit dem Computer. Generell ist es bei der technischen Beschreibung eines Algorithmus von Vorteil, gut mit der Funktionsweise eines Computers vertraut zu sein. Ich setze dieses Wissen um die Funktionsweise vorraus, da es den Rahmen sprengen würde, das zu erklären.
Was ich kurz anreißen möchte ist das Bit. Das Bit ist die Grundeinheit des Computers. Es beschreibt einen An- und einen Auszustand, also die Werte 0 und 1. Durch die Kombination von sehr vielen Bits, kann man Werte verarbeiten und speichern, und somit letztlich Texte verschlüsseln.

3.3 Kryptosysteme

Es gibt viele verschiedene Kryptosysteme zu verschiedenen Zwecken, oft ist auch die Bezeichnung dafür anders. Gebräuchlich sind zum Beispiel Algorithmus, Verschlüsselungsfunktion oder Verfahren.
Der symmetrische Algorithmus ist der häufigste und älteste. Er bekommt zur Ver- und Entschlüsselung einer Nachricht das gleiche Paßwort und benutzt das gleiche Verfahren.
Asymmetrische Algorithmen gibt es erst seit ca. 1980. Sie beruhen zumeist auf dem mathematischem Verfahren der Einwegfunktion. Das bedeutet, dass sie mit dem öffentlichem Schlüssel k verschlüsselt werden und mit dem privatem Schlüssel S entschlüsselt werden. Das wichtige ist hierbei, dass es schwierig bis gar nicht möglich ist aus dem Schlüssel K den Schlüssel S zu berechnen. Das ist auch das Verfahren der PublicKeyKryptographie. Diese ist viel praktischer als symmetrische Kryptographie, da sich hier das Problem der Schlüsselverteilung erledigt.

Asymmetrische Kryptographie wird zum Verschlüsseln von Daten zur Benutzerauthentifizierung verwendet, sowie für digitale Signaturen, welche die Echtheit eines Dokuments bestätigen.

$$E_k(M) = C \qquad D_S(C) = M$$

Schließlich gibt es noch die Hashfunktionen. Eine Hashfunktion wird in der Kryptographie MessageDigest, MDC, kryptographische Prüfsumme oder schlicht und einfach Fingerabdruck genannt. Hashfunktionen gibt es schon seit längerem. Sie ist nichts anderes als eine meist mathematisch definierte Funktion, die einen Eingabestring unbestimmter Länge erhält, mit ihr die definierten Aktionen vornimmt und schließlich auf einen kürzeren Ausgabestring fester Länge umwandelt. Wichtig sind bei Hashfunktionen zwei Dinge, erst einmal dass das Verfahren öffentlich ist, und dass es am

besten kollisionsfrei ist. Kollisionsfrei bedeutet, dass es für zwei verschiedene Eingabetexte keinen gleichen Hashwert gibt.

Als letztes Verfahren ist das hybride Kryptosystem anzumerken. Es vereint die Vorteile symmetrischer und asymmetrischer Kryptographie. Wenn man zum Beispiel über einen unsicheren Kommunikationskanal ein streng geheimes Dokument verschicken möchte, geht man auf folgende Weise vor: Bob verschlüsselt mit Alice`s öffentlichem Schlüssel ein Paßwort mit einem asymmetrischen Algorithmus. Mit diesem verschlüsseltem Paßwort verschlüsselt Bob vorher das geheime Dokument mittels einem symmetrischem Algorithmus.
Bob sendet jetzt das verschlüsselte Dokument und das verschlüsselte Paßwort an Alice über den unsicheren Kommunikationskanal. Alice entschlüsselt das verschlüsselte Paßwort für das Dokument mit ihrem geheimen Schlüssel. Danach nimmt sie das entschlüsselte Paßwort und entschlüsselt das geheime Dokument.
Die Vorteile dieses relativ schwierig erscheinenden Protokolls liegt darin, daß es über jede unsichere Verbindung gesendet werden kann, da der öffentliche Schlüssel des Partners bekannt ist. Ein weiter Vorteil liegt darin, daß ein symmetrischer Algorithmus schneller ist als ein asymmetrischer. Dadurch, dass man statt des langen Dokuments das kürzere Paßwort verschlüsselt und mit dem schnellerem symmetrischem Algorithmus das Dokument verschlüsselt, hat man einen Zeit bzw. Geschwindigkeitsvorteil. Nicht zu vergessen ist, dass ein mit dem asymmetrischen Algortihmus verschlüsseltes kürzeres Paßwort auch schwerer zu knacken ist, als ein längerer Text wie das Dokument selbst.

Eine Sonderrolle in der Kryptographie spielt die Steganographie. Sie versteckt eine Nachricht in einem anderem Medium, z.B. in einem Bild. Dies spielt hier jedoch keine Rolle, da es keine kryptographische Sicherheit bietet.

3.4 Grundlegende Angriffsarten auf Verschlüsselungen

Es gibt viele Angriffsarten auf kryptographische Verfahren, aber sie alle bauen auf diesen kryptanalytischen Attacken auf. Sie spezifizieren nicht die Art des Angriffs, sondern was für diese Attacke dem Kryptanalytiker verfügbar ist. Nicht eingeschlossen sind hier Verfahren die auf den Fehlern der Menschen beruhen, im Securitybereich auch oft Social Engineering genannt.

CipherText-Only-Angriff
Der Kryptanalytiker hat eine gewisse Menge an Chiffretext der mit dem gleichen Algorithmus und Schlüssel chiffriert wurde. Schlüssel und Klartext soll aus nur aus dem Geheimtext abgeleitet werden.
Gegeben $C_1 = E_k(P_1), C_2 = E_k(P_2) ... C_i = E_k(P_i)$
Gesucht ist hier dann k, P oder ein Algorithmus um P_i aus $C_{i+1} = E_k(P_{i+1})$ abzuleiten.

KownPlaintext-Angriff
Hier hat der Analytiker nicht nur den Chiffretext, sondern kennt auch den dazugehörigen Klartext.
Gegeben ist der Klartext sowie der verschlüsselte Text. Gesucht ist hier also der Schlüssel oder einen Algorithmus um den Klartext ohne Kenntnis des Schlüssels zu entziffern.

ChosenPlainText-Angriff
Hier kann der Angreifer beliebig viele Nachrichten verschlüsseln, also hat er beliebig viele Klar/Chiffretext Paare. Den Schlüssel kennt er allerdings meist nicht, oder zumindest bei einem PublicKeySystem nur den öffentlichen Schlüssel.

Adaptive-ChosenPlainText-Angriff
Wie der Name schon vermuten läßt, ist das ein Spezialfall des oberen ChosenPlainText-Angriffs. Hier kann er die Erstellung der Klar/Chiffretext Paare so variieren, daß sie in Beziehung zu den vorherigen verschlüsselten Nachrichten stehen.

ChosenCipherText-Angriff
Der Analytiker kann auf Chiffretexte und deren Klartexte zugreifen aber nicht auf das Paßwort.
Eine Kombination aus ChosenPlainText und ChosenCipherText Angriffen nennt man häufig ChosenText-Angriff

Kryptanalyse mit Gewalt
Dies wird gerne als Methode der Geheimdienste bezeichnet, weil der „Kryptanalytiker“ so lange jemanden bedroht oder erpreßt, bis das Opfer den Schlüssel preisgibt.
Bei dem Aufwand heutige Verschlüsselungen zu knacken, ist die Kryptanalyse mit Gewalt sicher der erfolgreichste und einfachste Weg da der Mensch immer die größte Schwachstelle in jedem System ist.

Bruteforce
Bruteforce (eng.: brutale Gewalt) bedeutet nichts anderes als bei einem verschlüsseltem Text und bekanntem Algorithmus alle Schlüssel und Klartexte solange miteinander zu verschlüsseln, bis das Ergebnis mit dem feststehendem Chiffretext identisch ist.

Man-in-the-Middle-Angriff
Dieser Begriff erklärt sich zum Teil von alleine. Während ein passiver Angreifer nur die oben erwähnten Grundformen zum Kryptanalysieren einer Nachricht benutzen kann, stehen dem aktiven Angreifer Mallory viel mehr Möglichkeiten zur Verfügung. Diese Attacke verlauft wie folgt:
Alice sendet Bob ihren öffentlichen Schlüssel, wobei Mallory diesen Schlüssel abfängt und statt dessen seinen öffentlichen Schlüssel an Bob schickt. Dann sendet Bob seinen öffentlichen Schlüssel vermeintlich an Alice, aber Mallory bekommt ihn und schickt wiederum den seinen an Alice. Sendet

Alice eine mit Bobs öffentlichem Schlüssel chiffrierte Nachricht an Bob, fängt Mallory diese ab, und dechiffriert sie mit seinem Schlüssel.
Nun kann Mallory die Nachricht beliebig manipulieren oder Löschen. Danach chiffriert Mallory sie mit Bobs wirklichem öffentlichem Schlüssel und schickt sie an ihn. Dieser meint natürlich die Nachricht sei von Alice und denkt jetzt der Kommunikationskanal sei sicher, obwohl er sogar schon kompromittiert ist.
Dieser Angriff wird auch oft bei Algorithmen durchgeführt, indem man Zwischenergebnisse analysiert und nach Schwächen und Mustern sucht.

3.5 Leitsätze und Maximen für gute Algorithmen

Kerkhoff's Maxime
Diese Regel des Niederländers A. Kerckhoff besagt, daß die Sicherheit eines Verschlüsselungsverfahrens nur von der Geheimhaltung des Schlüssels, und nicht wie heute noch oft praktiziert, von der Geheimhaltung des kryptographischen Algorithmus abhängen darf. Wenn er trotzdem geheimgehalten wird, heißt der Algorithmus eingeschränkt

Ein großer Schlüsselraum
Die Menge aus welcher der Schlüssel gewählt werden kann muß sehr groß sein, damit der Aufwand für einen Bruteforce Angriff sich nicht lohnt bzw. praktisch nicht zu machen ist. In den frühen 80 Jahren war die dafür ausreichende Menge 64 Bit, mittlerweile ist man bei 128 oder sogar 256 Bit angelangt. Allerdings sind Schlüssellängen >= 256 Bit für alle Zeiten sicher. Dies gilt jedoch nur für symmetrische Systeme. Bei asymmetrischen Systemen die auf dem Problem der Faktorisierung großer Zahlen beruhen, müssen es mehr Bits sein.
Die unterste Grenze liegt zwischen 1024 Bit und 4096 Bit. Man nimmt hier deshalb so grosse Schlüssellängen, da man annimmt, dass Regierungsorganisationen wesentlich mehr Ressourcen für Bruteforce Angriffe aufbieten können.

Effizienz
Ein Algorithmus sollte aus einfachen Operationen bestehen, die sich leicht in Hardware und Software implementieren lassen. Gilt dies nicht, kann er nur eingeschränkt verwendet werden, was nicht im Sinne der Kryptographie ist.

3.6 Algorithmen und deren Modes

Bei Algorithmen unterscheidet man hauptsächlich zwischen zwei Arten. Einer ist der Blockalgorithmus. Er nimmt vom Klartext einen Block bzw. Menge mit bestimmter Länge, zur Zeit ist 64 Bit gebräuchlich. Diesen Block verarbeitet er auf bestimmte Weise und gibt dann diesen chiffrierten Block aus. Er wiederholt diesen Vorgang mit der gesamten Nachricht bis alles chiffriert ist. Ist eine Nachricht dabei nicht restlos in 64 Bit Blöcke aufteilbar und es bleibt ein Rest, so wird dieser Rest solange mit Zufallswerten aufgefüllt bis die Blockgrösse erreicht ist. Dieser Vorgang nennt sich Padding.

Ein Blockalgorithmus verwendet Konfusion und Diffusion zur Verschlüsselung. Optimal arbeitet ein Blockalgorithmus wenn es zu jeder der $2^n!$ verschiedenen Blockkonfigurationen einen Schlüssel gibt. Dabei bezeichnet n die Anzahl der Bits eines Blocks.
Nun gibt es neben den Blockalgorithmen auch noch die Stromalgorithmen. Hier wird immer nur Zeichen für Zeichen chiffriert. Es kann ein Byte oder ein Bit sein. Das Zeichen hängt vom verwendeten Alphabet ab.
Wenn eine Stromchiffre ohne Rückkopplung arbeitet, benutzt sie nur Konfusion.
Nun will ich kurz die Techniken Diffusion und Konfusion erläutern, welche aus der Shannontheorie stammen und wesentlich für eine sichere Verschlüsselung sind.
Konfusion bedeutet, daß es den Zusammenhang zwischen Klartext und Chiffretext sowie Paßwort verschleiert. Dazu wird im allgemeinen die Substitution benutzt. Bei der Substitution wird ein Symbol, Zeichen, Byte oder Bit durch ein anderes ersetzt.
Ein *sehr einfaches* Beispiel ist, wenn man jedem Buchstaben im Alphabet durch einen anderen ersetzt

A	B	C	D	E	F	G	H	.....	U	V	W	X	Y	Z
B	D	Z	E	L	J	W	M	.....	I	A	C	F	G	U

Dann wird aus dem Wort „Fuge“ das Wort „Jiwl“. Dadurch ist der Text nicht mehr lesbar. Ein anderes Beispiel dafür ist XOR das in C++ durch das Zeichen ‚^‘ dargestellt wird. Es ist eine Addition modulo 2 und wird auf Bitebene durchgeführt. Zudem ist es ein logischer Operator.
Diffusion verteilt die Redundanz des Klartexts über das Chiffrat. Das heißt die Zeichen sind immer noch die gleichen jedoch aus ihrem Zusammenhang gerissen und es erschwert die statistische Analyse. Die Diffusion wird durch die Transposition (Permutation) der Zeichen erledigt.
Ein ganz einfaches Beispiel wäre es einfach jedes Wort umzukehren.
Hier eine ähnliche Möglichkeit mit dem Wort „Woche“. Hierbei wird auch die ursprüngliche Reihenfolge der Buchstaben als Matrize (1,2,3,4,5) genommen.
Wenn die Buchstaben in der Matrize auf bestimmte Weise vertauscht bzw. permutiert werden, bekommt man eine andere Matrize, z.B. (3,1,4,2,5).
Wenn die Zahlen den Buchstaben jetzt zugeordnet werden, kommt das Wort „cWhoe“ heraus.
Mit der Zeit wurde allerdings klar, das die Transposition ein schwaches Verfahren ist. Es gibt natürlich noch viel mehr Verfahren als Substitution und Transposition aber dies sind grundlegende Techniken, die in eigentlich in Abarten immer wieder auftauchen.
Da einzelne Techniken sich im Altertum wie in der Neuzeit als zu schwach erwiesen, ging man mit der Zeit zu der Kombination verschiedener Algorithmen über.
Zur Erweiterung dieser Algorithmen dachte man sich deshalb die Modes aus. Modes sind bei kryptographischen Algorithmen für unterschiedliche Aufgaben zuständig. Zum einen können sie durch einfache Operationen die Sicherheit des Algorithmus verstärken. Andere Anwendungen wiederum

verlangen, dass Bitfehler im Chiffre oder im Klartext durch den Mode korrigiert werden.

Electronic-Code-Book-Modus (ECB)
Das ist die naheliegendste und oben schon angedeutete Art eine Blockchiffre anzuwenden. Das heißt jeder Klartextblock wird in den entsprechenden Chiffretextblock überführt, bis die gesamte Nachricht chiffriert wurde. Jeder Block ist hier unabhängig von den anderen Blöcken. Dies bedeutet man kann mit dem Verschlüsseln anfangen, wo man will; deshalb ist diese Art wichtig bei wahlfreien Zugriffen z.B. bei Datenbanken. Der Nachteil dadurch aber ist, dass sich bei wiederholenden Nachrichten, welche mit dem gleichen Schlüssel chiffriert wurden, Muster einschleichen. Erkennt ein Angreifer, daß ein bestimmter Chiffretext zu einem bestimmten Klartext gehört, erhält er automatisch Informationen für die nächsten Nachrichten, nämlich statistische Informationen, und er kann ganz unabhängig von der Stärke der Codierung die Nachricht leichter brechen. Die Gefahr des Musters steigt bei Anfang und Enden von Nachrichten natürlich erheblich an, da hier stereotype Anfänge und Enden vorhanden sind.
Z.B. wird in einer Email am Anfang meist „Hi Klaus“ geschrieben. Hier kann der Angreifer schon etwas vermuten, und eine Angriffsfläche für die erfolgreiche Kryptanalyse finden.

Cipher-Block-Chaining (CBC)
Wörtlich übersetzt heißt das Wort „Verknüpfung verschlüsselter Blöcke“. Das ist eine Rückkopplung bei der Blockchiffrierung, bei welcher die Verschlüsselung früherer Blöcke in die Verschlüsselung des aktuellen Block eingeht. Der Klartext wird vor der Verschlüsselung mit dem vorherigen Text XOR-verknuepft. Der vorherige Chiffretext wird in einem Rückkopplungsregister gespeichert, damit er dann zur Verfügung steht.
Die entsprechenden Formeln lauten:
Für die Verschlüsselung: $C_i = E_k(P_i \oplus C_{i-1})$
Für die Entschlüsselung: $P_i = C_{i-1} \oplus D_k(C_i)$

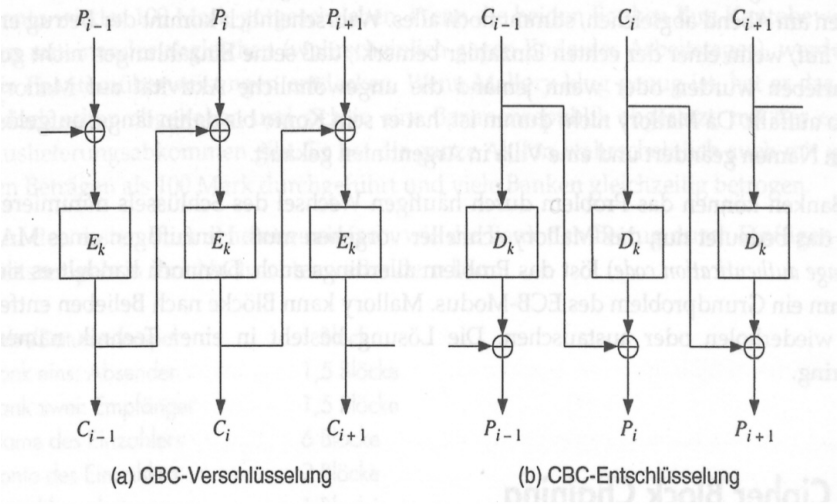


Bild 2: CBC-Diagramm ([1], S.228)

Ein Problem bleibt aber, da der erste Block immer noch im ECB Modus verschlüsselt wird. Um das zu verhindern, führte man den sogenannten IV, den Initialisierungsvektor ein. Er besteht aus einem zufälligem Byteblock, der bloß existiert um den ersten richtigen Klartextblock zu verschleiern. Man muß ihn nicht einmal geheimhalten, da es ein Pseudochiffreblock ist.

Cipher-Feedback-Modus (CFB)
Der CFB ist eine selbstsynchronisierende Blockchiffrierung, wobei die Selbstsynchronisierung von den Stromchiffren abgeschaut wurde. Hier kann die Verschlüsselung allerdings erst anfangen, wenn der vollständige Datenblock vorhanden ist. Da im CFB-mode die Daten in Einheiten, sprich in Bits verschlüsselt werden, muß n <= 64 bei n-bit-CFB sein. Am Anfang der Prozedur steht wie bei dem CBC ein Initialisierungsvektor, das Register wird verschlüsselt, und die n Bit die im Register links stehen werden mit der gleichen Anzahl Klartextbits verknüpft.
Das Ergebnis der n-Bit-Chiffre wird übertragen und in das Register ganz rechts verschoben. Links fällt natürlich dann die gleiche Anzahl n Bits aus dem Register heraus. Dann wird die nächste n-Bit Reihe chiffriert. Zum Entschlüsseln wird dieser Vorgang noch einmal umgekehrt. Zur Verdeutlichung des Vorgangs noch ein Schaubild:

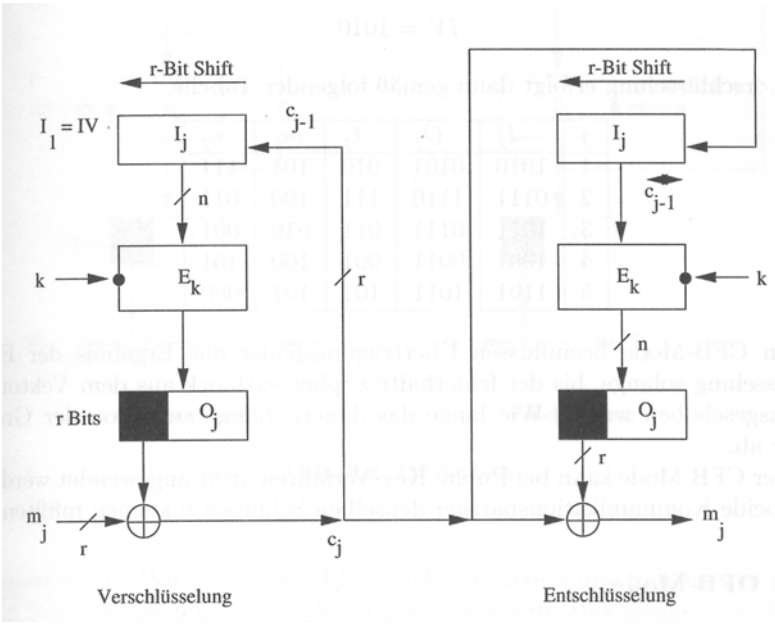


Bild 3 CFB-Diagramm ([2], S.69)

Im Register wird immer der aktuelle Block verschlüsselt. Es gibt bei dem CFB allerdings ein paar feine Unterschiede. Der Initialisierungsvektor des CFB muß im Gegensatz zum IV des CBC jedesmal anders sein, da es sonst dem Kryptanalytiker die Arbeit erleichtert. Die Formel des CFB lautet für die Verschlüsselung $C_i = P_i \oplus E_k(C_{i-1})$ und analog für die Entschlüsselung. Der Vorteil ist, dass ein Fehler im Klartext sich zwar auf den ganzen Chiffretext ausbreitet (gute Diffusion!), sich jedoch beim Entschlüsseln wieder

rückgängig macht, da der fehlerhafte Bitblock irgendwann aus dem Register herausrotiert. Bei einem Fehler im verschlüsseltem Text, wirkt sich allerdings der n-bit-CFB sehr viel drastischer aus. Es sind dann nämlich der aktuell zu entschlüsselnde Block und bedingt durch Register und Blockgröße auf die folgenden m/(n-1) Blöcke aus, mit m als Blocklänge. Diese Eigenschaft kann allerdings zum Angriff „Angriff durch Einfügen „ genutzt werden. Durch die Selbstsynchronisierung erholt sich der CFB von selbst.

Output-Feedback-Mode (OFB)
Der OFB ist dem CFB sehr ähnlich. Er wird benutzt um eine Blockchiffrierung als synchrone Stromchiffrierung durchzuführen. Da allerdings die n Bits für das Schieberegister von dem zu verschlüsselndem Klartext kommen, werden sie vom Schlüssel genommen. Diese Rotation nennt man dann interne Rückkopplung
Ein n-bit-OFB sieht so aus: $C_i = P_i \oplus S_i \quad S_i = E_k(S_{i-1})$ Der Entschlüsselungsvorgang verläuft genauso.
Für den Initialisierungsvektor gilt das gleiche wie bei dem CFB.
Fehlerfortpflanzung im Klar bzw. Chiffretext existiert nicht. Allerdings würde bei Verlust der Synchronisierung das System fast zerstört werden, wenn Schieberegister und Schlüssel „aus dem Takt“ geraten. Noch einmal ein Schaubild zur Verdeutlichung:

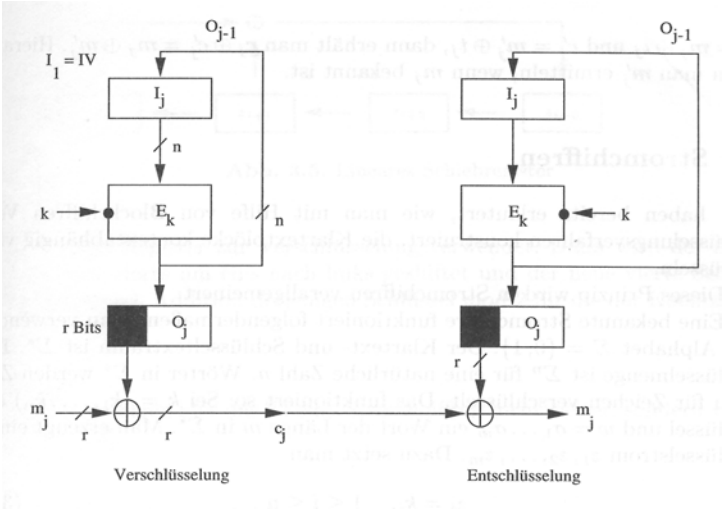


Bild 4 OFB-Diagramm ([2], S.71)

Zum Schluß werden noch ein paar Einzelheiten zu den Modi erwähnt. Es gibt außer diesen vier Modi noch viel mehr z.B. den CounterModus. Allerdings sind es diese die vom NIST, dem Institut für Standardisierung von kryptographischen Algorithmen, als implementierbar gegeben sein müssen. ECB eignet sich nur für zufällige Daten. Bruce Schneier, ein anerkannter KryptographieExperte rät davon ab, den ECB-Modus zur Verschlüsselung von Nachrichten zu benutzen.
CBC, CFB sowie OFB sind für die Verschlüsselung von Klartext gut geeignet CBC ist speziell auf Dateien ausgerichtet, da er die Muster verwischt.

Allerdings zur Verschlüsselung von Daten wie z.B. Telnet eignet sich das 8-Bit-CFB sehr gut, da alle Zeichen einzeln verschlüsselt und verschickt werden. OFB ist durch seine nichtvorhandene Fehlerfortpflanzung für Übertragungen von Videodaten prädestiniert.

3.7 Informationstheorie

Claude E. Shannon gründetete die Informationstheorie. Sie wurde die wichtigste theoretische Grundlage für alle Arbeiten und Geräte, welche auf dem Gebiet der elektronischen Kommunikation bzw. der digitalen Nachrichtenübertragung. Ihre Besonderheit war, dass sie einen Text nicht nach Qualität sondern nach „Quantität“ betrachtet.
Man unterscheidet bei einer Nachricht zwischen irrelevantem, redundantem und restlichem Code. Der Entscheidungsgehalt ist $H_0 = ld(N)$ [Bit pro Zeichen], d.h. der Aufwand der Bildung einer Nachricht bei einem Alphabet von 40 Zeichen ist ld(40) = 5.32 Bit pro Zeichen.
Ein ASCII Text in englischer Sprache hat also durch sein Alphabet 6.7 Bit redundanter Information in jeder Binärdarstellung eines Zeichens. Dadurch hat jedes Bit des Textes 0.84 Bit Redundanz. Diese erkennt man an der Differenz von H_0 - H(X). H(X) ist die Entropie, also der mittlere Entscheidungsgehalt einer Quelle und wird durch $\sum_{k=1}^{N} \frac{1}{N} * ld(N) = \sum_{k=1}^{N} p(x_k) * ld(\frac{1}{p(x_k)})$ definiert, wobei p(x_k) die Auftrittswahrscheinlichkeit für ein Zeichen aus dem Alphabet ist. Da ein ASCII Code nicht nur Buchstaben enthält, werden diese Zahlen auch durch Leerzeichen sowie Interpunktion verändert. Wichtig ist, dass die Redundanz bei wahrscheinlichkeitsmässig gleich verteilten Zeichen aus dem Alphabet gleich Null ist. Im allgemeinen nähert sie sich aber nur 0 an, was aber erstrebenswert ist, da es bei der elektronischen Übermittlung von Daten wichtig ist, möglichst wenig redundanten Code zu übermitteln.
Eine interessante aber für die Erstellung von Kryptosystemen eher unwichtige Erkenntnis ist die Unizitätslaenge bei symmetrischen Kryptosystemen. Sie ist definiert durch H(K)/D, und besagt die Entropie des Systems durch die Redundanz der Sprache. Dies ist allerdings ein unsicheres Ergebnis, welche nur die Länge des Chiffretext abschätzt bei der ein Bruteforce ein Erfolg ist.
Generell ist ein Kryptosystem umso besser, je größer der Wert ist. Daher ist ein Kryptosystem wie das OneTimePad unknackbar, da es eine unendliche Unizitätslaenge hat.

3.8 Komplexitätstheorie

Die Komplexitätstheorie handelt von der Schwierigkeit, bestimmte Probleme mit bestimmten Ressourcen in einem festgelegtem Zeitraum zu lösen. Die, für Computer gemachten Algorithmen, haben meistens folgende Parameter: Die Rechenzeit T, den Speicherbedarf S sowie Eingabedaten der Größe n. Die Abschätzung der Berechnungsdauer, sowie die benötigte Kapazität dafür

wird mit der O- bzw. Ω - Notation dargestellt. Ein Algorithmus mit dem Laufzeitaufwand $7n^2+3n$ hat $O(n^2)$ als Ordnung, also gibt der höchste Grad der Variablen die Komplexität des Algorithmus bekannt, denn dies ist die obere Schranke.

Zum Beispiel ist auch die Komplexität einer Binärzahl definiert durch O(log(n)). Wie oben schon ersichtlich, ist die Komplexität von Algorithmen stark abhängig von Rechenzeit und Speicherkapazität.

Es gibt dabei mehrere Einteilungen von Arten von Algorithmen:

Konstanter Algorithmus	Die Komplexität von Berechnung oder Zeit ist unabhängig von n. zb. O(13)
Linearer Algorithmus	Komplexität ist O(n)
Polynomialer Algorithmus	Der Grad von n ist größer 1 z.B. $O(n^3)$, wichtig hier ist das der Grad konstant.
Superpolynomialer Algorithmus	Die Komplexität eines superpolynomialen Algorithmus nähert sich an die Komplexität eines exponentiellen Algorithmus an. Die Basis ist konstant, der Exponent ist eine Funktion die langsamer als linear wächst.
Exponentieller Algorithmus	$O(3^n)$ Die Basis ist konstant, „nur“ der Exponent ist variabel

Die Einteilungen sind geordnet von dem am langsamsten wachsenden Algorithmus bis zu dem schnellsten wachsenden Algorithmus bei gleichem Eingabeparameter.
In der Praxis bedeutet das, dass alles was höhere Komplexität als polynomial besitzt, zu lange dauert um in vernünftiger Zeit berechnet werden zu können. Alle Dechiffrieralgorithmen von Texten ohne Paßwort sollten deshalb mindestens superpolynomiale Laufzeitkomplexität besitzen. Dies läßt sich jedoch nicht so einfach beweisen.
Deshalb wird bei Algorithmen auch mindestens eine Schlüssellänge von 128 Bit verlangt, weil dies bei Bruteforce zu erwähnter superpolynomialer Zeit führt. Zur Erleichterung dieser Komplexitätsprobleme werden in der Informatik verschiedene Klassen geführt.

4 Beschreibung des Data Encryption Standards (DES)

4.1 Geschichte und Entwicklung

1972 wurde von dem NIST (National Institute of Standards) der Anstoß gegeben, einen allgemein gültigen Algorithmus zu schaffen der überall einsetzbar, und sicher ist. Die Anforderungen für diesen kryptographischen Standard lauteten wie folgt:

- Der Algorithmus muß einen hohen Grad an Sicherheit gewährleisten.
- Er muß vollständig spezifiziert und nachvollziehbar sein.
- Die Sicherheit des Algorithmus muß auf dem Schlüssel beruhen, und hängt nicht von der Geheimhaltung des Verfahrens ab.
- Er muß für alle verfügbar und einsehbar sein.
- Kostengünstige Implementierung, auch in elektronische Geräte.
- Der Algorithmus soll effizient und exportierbar sein.

Einige Zeit gab es nichts erwähnenswertes, doch dann tauchte von IBM „Lucifer“ auf. Das Verfahren wurde von einem Team von Kryptanalytikern wie Horst Feistel und Don Coppersmith entwickelt. Die Sachkenntnis, ob dieser Algorithmus sicher sei, hatte bei dem NIST (früheres NBS (National Bureau of Standards)) niemand, deshalb zog man die NSA (National Security Agency) zu Rate. Nachdem die NSA den Algorithmus untersucht und zurückgeschickt hatten, stellte sich heraus, dass, dieser völlig verändert war. Nachdem das Verfahren der differentiellen Kryptanalyse bekannt wurde(siehe 4.4), wurden auch die Entwurfskriterien für die Sboxen veröffentlicht, da diese gegen die der NSA und IBM schon früher bekannten Kryptanalyse optimiert waren. Vorher wie jetzt auch vermutet(e) man allerdings dass die NSA eine Backdoor in das Verfahren einbaute, ansonsten hätten sie nicht erlaubt den Algorithmus standardisieren zu lassen. Ein Mißverständnis zwischen NSA und IBM ist wahrscheinlicher, da die NSA glaubte, der Algorithmus solle nur in Hardware implementiert werden. Dafür spricht, dass der nächste Standard der NSA, der Algorithmus Skipjack jahrelang geheimgehalten wurde. Allerdings kürzte die NSA den Schlüssel von 128 Bit auf 56 Bit, was später noch zurückgekommen wird. Bis 1997 wurde dann DES als Standardverfahren eingesetzt. Dies war das erstemal, dass ein von der NSA als sicher eingestufter Algorithmus zur Analyse freigegeben war, infolge dessen wurde er auch immer wieder analysiert und angewendet.

4.2 Einführung

Als Vorbemerkung, durch die Veröffentlichung des Algorithmus war das Kerckhoffprinzip gegeben, so dass auch wir in der Lage sind, den Algorithmus zu analysieren.
DES ist ein symmetrischer Produktalgorithmus, eine Blockchiffre, spezieller ein Feistelnetzwerk. Es gibt mehrere Arten von Feistelnetzwerken. Aufgrund der großen Theorie, die sich hinter den Feistelnetzwerken verbirgt, wird hier nur die beim DES verwendete Art beschrieben. Die Chiffre verwendet einen 64 Bit langen Schlüssel wobei das letzte Bit eines Bytes allerdings immer zur Fehlerüberprüfung genutzt wird, die effektive Schlüssellänge beträgt also 56 Bit. Daraus ergeben sich $2^{56} \approx 7.2x10^{16}$ Schlüssel. Das Verfahren für Verschlüsselung und Entschlüsselung ist dasselbe, bei der Entschlüsselung wird bloß die Bit Reihenfolge umgedreht. Die Bitlänge des zu verschlüsselnden Blockes beträgt 64 Bit, also ist das Alphabet des Verfahrens $\{0,1\}^{64}$. Das Fundament des DES ist es, abhängig vom Schlüssel, Konfusion und Diffusion in verschiedenem Maße anzuwenden. Ein Vorgang,

in dem dies erreicht wird, heißt „Runde“. Üblicherweise wird DES mit 16 Runden betrieben. Durch die Bauart des Algorithmus, nur einfache arithmetische und logische, also leicht in Hardware implementierbare Operatoren, zu verwenden, eignete er sich hervorragend für den Einsatz in Chipsätzen.
Hier ein Bild über den allgemeinen Verlauf einer Runde mit Schlüsseltransformation in DES.

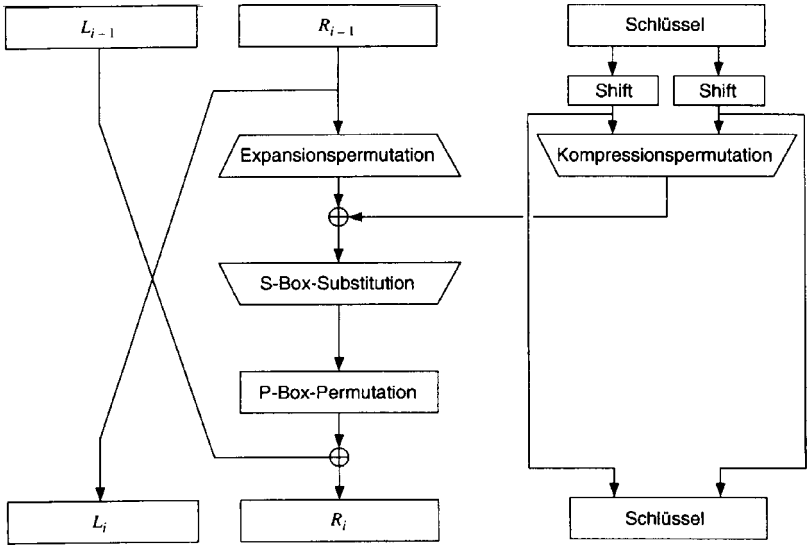


Bild 5: DES-Runde ([1], S.37)

Am Anfang und am Ende stehen zwei schlüsselunabhängige Permutationen, die dazu dienten, die Daten hardwaremässig schneller zu verarbeiten, da sich Bytes dadurch schneller in einen Chip laden lassen. Durch die Unabhängigkeit des Schlüssel sind sie also keine kryptographische Funktion. Die Endpermutation ist invers zur Eingangspermutation. Sie ändern bloß deterministisch die Bitpositionen.

4.3 Das Feistelnetzwerk

Das Feistelnetzwerk ist aus der bildlichen Darstellung gut ersichtlich. Hier noch die mathematische Definition:
$(R_{i-1}, L_{i-1}) = (L_i, R_i \oplus f_{k_i}(L_i))$ mit 1 <= i <= r.
r ist die Anzahl der Runden, L ist die linke Hälfte des Blockes und R die rechte. i gibt den aktuellen Rundenindex an. F ist die interne Rundenfunktion, und $\oplus$ beschreibt den Operator XOR.

4.4 Die interne Blockchiffre F

Am Anfang der Runde wird der Block in eine linke und rechte Hälfte aufgeteilt, mit je 32 Bit. Danach wird der rechte Block mittels der Expansionspermutation auf 48 Bit erweitert, wobei die Bits dann permutiert und wiederholt werden. Erst dadurch ist die XOR-Operation mit dem Rundenschlüssel möglich. Durch die Erweiterung des Blockes kommt der SAC (Strict Avalanche Criterion) auf Deutsch der Lawineneffekt stärker zum Tragen. Dieser sagt aus, dass von einem Eingabebit viele Ausgabebits abhängen. Die Expansionfunktion geht auf folgende Weise vor sich: Hat man den Bitstring $R_1 R_2 .....R_{32}$, wobei jedes R ein Bit darstellt, wird es nach der ExpansionTabelle entsprechend den Positionen die darin angegeben sind so dargestellt: $R_{32} R_1 R_2 R_3 ....R_{32} R_1$. Das ist die Tabelle der Expansionspermutationsbox:

8	9	10	11	12	13
12	13	14	15	16	17
16	17	18	19	20	21
20	21	22	23	24	25
24	25	26	27	28	29
28	29	30	31	32	33
32	33	34	35	36	37
36	37	38	39	40	41

Danach wird der geänderte Block mit der Substitutionsbox, besser bekannt als Sbox, verändert. Über diese Sboxen ist schon sehr viel gerätselt worden. Sie haben aber auch so fundierte mathematische Grundlagen, dass allein darüber schon seitenlange Dokumente geschrieben wurden.
Nun eine Beschreibung des Substitutionsverfahrens. Es gibt 8 Sboxen, der 48-bittige Teilblock wird in 8 Teile der Länge 6 Bit zerlegt, wobei jeder Teil mit einer Sbox verarbeitet wird. Eine Sbox ist eine Tabelle mit 16 Spalten und 4 Zeilen. Jeder Tabelleneintrag enthält einen binären String der Länge 4. Von dem 6 Bit-Eingabe-String A zum 4-Bit-Ausgabe-String B kommt man indem man die Bits A_1, A_6 als Zeilenindex verwendet und $A_2 A_3 A_4 A_5$ wird als Spaltenindex verwendet. Dadurch kommt man nach Abarbeitung aller Sboxen zu einem 32-Bit großen Block.
Danach werden die 32 Bit aus den Sboxen durch die untenstehende Permutationstabelle P vermischt:

16	7	10	21
29	12	28	17
1	15	23	26
5	18	31	20
2	8	24	14
32	27	3	9
19	13	30	6
22	11	4	25

Die Tabelle besagt wie schon oben, dass das 1. Eingabebit danach an 9 Stelle und das 16 Bit jetzt an 1 Stelle steht.

Jetzt wird die Ausgabe mit dem linken Block XOR-verknüpft. Danach wird der bearbeitete rechte Block zum linken Block und umgekehrt. Dann startet eine neue Runde.

4.5 Die Schlüsseltransformation

Aus Paritätsüberbrüfungsgründen wird der Schlüssel von 64 Bit auf 56 Bit reduziert. Dadurch wird sichergestellt, dass der Schlüssel fehlerfrei ist. Dies geschieht in der Kompressionspermutation. In ihr wird der Schlüssel auch in 2 28-Bit Blöcke (C_0, D_0) aufgeteilt. Hier die erste Kompressionspermutation:

57	49	41	33	25	17	9
1	58	50	42	34	26	18
10	2	59	51	43	35	27
19	11	3	60	52	44	36
63	55	47	39	31	23	15
7	62	54	46	38	30	22
14	6	61	53	45	37	29
21	13	5	28	20	12	4

Abhängig von der Rundenanzahl werden die Hälften um 1 oder 2 nach links verschoben. Diese Verschiebung erstellt für jede Runde einen anderen Schlüssel, weil sie eine andere Teilmenge benutzt:

Runde	1	2	3	4	5	6
Verschiebung	1	1	2	2	2	2
Runde	7	8	9	10	11	12
Verschiebung	2	2	1	2	2	2
Runde	13	14	15	16		
Verschiebung	2	2	2	1		

Nach der Verschiebung wird aus den beiden Paaren mittels einer zweiten Kompressionspermutation ein 48 Bit langer Block erzeugt:

14	17	11	24	1	5
3	28	15	6	21	10
23	19	12	4	26	8
16	7	27	20	13	2
41	52	31	37	47	55
30	40	51	45	33	48
44	49	39	56	34	53
46	42	50	36	29	32

Danach beginnt eine neue Runde, und die Beschreibung des Algorithmus endet hier.

4.6 Sicherheitsaspekte zum DES

Um die Besprechung für einen kryptographisch sicheren Algorithmus zu vervollständigen, muß man noch einige Dinge ansprechen, die nicht vom Aufbau her ersichtlich werden. Diese Sicherheit wird durch kryptanalytische

Attacken auf den Algorithmus oder Erörterungen über sein grundsätzliches Design klar.
Da wären zum einen die schwachen Schlüssel zu erwähnen. Schwache Schlüssel sind Schlüssel bei denen entweder als Ergebnis Klartext und Chiffretext identisch sind, oder alle Teilmengen des Schlüssels gleich sind. Dies führt sonst zur leichteren Dechiffrierung des verschlüsselten Textes.
Eine weitere, allerdings sehr theoretische Eigenschaft ist die Komplementäreigenschaft von DES. Zum Beispiel $DES_k(P) = C$. Durch einen komplementären Schlüssel und komplementären Text entsteht ein dazu entsprechender komplementärer Chiffretext, $DES_{k'}(P') = C'$.
Das nächste wichtige ist die Eigenschaft, dass DES keine abgeschlossene Gruppe ist, da es ansonsten für eine Man-in-the-middle-Attack anfällig wäre, da für 2 Schlüssel von ineinander ausgeführten DES Verfahren stets ein dazu äquivalenter 3. Schlüssel existent wäre.
Heutzutage ist die Schlüssellänge von 64, respektive 56 Bit nicht mehr ausreichend, 128 Bit ist heute die Standardlänge.
Nach Aussage verschiedener Experten ist die Rundenzahl 16 für DES sehr gut gewählt. Für diese Anzahl von Runden ist die zu bewältigende Komplexität zu groß, als dass es heutzutage möglich ist, das Ergebnis ohne das Paßwort zu dechiffrieren. Alle Rundenanzahlen davor wären machbar, allerdings schien ab 16 Runden nichts mehr möglich zu sein.

Der letzte aber auch zugleich größter zu behandelnder Sicherheitsaspekt sind gehören die Sboxen. Sie sind der wichtigste und zugleich mysteriöseste Teil in dem ganzen Verschlüsselungsverfahren. Ende der 70er Jahre waren die Konstruktionskriterien für die Sboxen noch nicht bekannt, deshalb wurden viele Spekulationen für diese willkürlich aussehende Zahlentabelle gemacht. Mittlerweile besitzt man hier mehr Sachkenntnis. Im Laufe der Zeit wurden die Konstruktionskriterien für die Boxen bekannt. Auch erkannte man, dass die Tabellen gegen die differentielle Kryptanalyse optimiert waren.
Hier die grundlegenden Kriterien für die Sboxen:

- Jede Sbox hat 6 Eingabe und 4 Ausgabebits. Das war die maximale Größe, die man damals auf Chips unterbringen konnte.
- Kein Ausgabebit einer Sbox sollte zu nahe an einer linearen Funktion der Eingabebits liegen
- Wenn man das ganz linke und das ganz rechte Bit einer Sbox unverändert läßt und die vier mittleren Bits variiert, sollen alle möglichen 4Bit-Ausgaben genau einmal vorkommen.
- Wenn sich zwei Eingabewerte einer Sbox in genau einem Bit unterscheiden, müssen sich die Ausgabewerte in mindestens 2 Bit unterscheiden.
- Wenn sich zwei Eingabewerte einer Sbox in den ersten beiden Bits unterscheiden und in den letzten 2 Bits übereinstimmen, dann dürfen die beiden Ausgabewerte nicht identisch sein.
- Für jede mögliche 6 Bit Differenz (ungleich 0) zwischen 2 Eingabewerten muß gelten: höchstens 32 Paare, die diese Differenz haben, liefern die gleiche Differenz der Ausgabewerte.
- Ein Kriterium das analog zum vorherigen eine Bedingung für drei aktive Sboxen aufstellt

Folgend die Pbox-Kriterien:

- Die 4 Ausgabebits jeder Sbox in Runde i werden so verteilt, dass zwei davon die mittleren Bits der Sboxen in Runde i +1 beeinflussen und die anderen beiden die äußeren Bits.
- Die vier Ausgabebits jeder Sbox beeinflussen sechs verschiedene Sboxen. Es gibt keine zwei Bits, welche die gleiche Sbox beeinflussen.
- Wenn das Ausgabebit einer Sbox ein mittleres Bit einer anderen Sbox beeinflußt, dann darf ein Ausgabebit einer anderen Sbox kein mittleres Bit der ersten Sbox beeinflussen.

Diese Liste stammt aus [1], S.341/342

Durch diese Regeln sind die Substitutionsboxen der einzig nichtlineare Teil der Verschlüsselung, d.h. von ihrer Sicherheit hängt im wesentlichen die Sicherheit des Algorithmus ab. Aber erst durch die Kombination mit den Permutationsboxen, vereinigen sie sich zu einer guten Produktchiffrierung weil in den Pboxen dafür gesorgt wird, dass immer wieder ein anderes Bit verschlüsselt wird.
Die Sboxen wurden als Kompressionsalgorithmus konzipiert, da auf einen Input immer ein kleinerer Output entsteht. Dies war nötig, da die Expansion der Pboxen die Daten zwangsläufig vergrößerte.
Die Spalten- bzw. Zeilenanzahl der Sboxen haben auch eine ganz bestimmte Funktion. Eine Sbox ist im Grunde ja nichts anderes als eine LookupTabelle, also eine Art eigenes Alphabet. Daher nennt man eine Sbox mit m Bit Eingabe und n Bit Ausgabe eine m*n Bit Sbox. m wirkt hier aber stärker als n, dass heißt die Anzahl der Ausgabebits sollte möglichst relativ klein sein. Wird n zu groß, dann nähert sich die Sbox einer linearen Funktion an.

Das wichtigste Kriterium, das aus den Gesetzen der Boxen gefolgert werden kann, ist der oben schon erwähnte SAC. SAC fordert, wenn sich ein Eingabebit ändert, dass genau die Hälfte der Ausgabebits sich auch ändert. Genaueres darüber erfährt man beim Studium boolescher Funktionen.
Eine Sache sollte noch erwähnt werden. Es ist offensichtlich dass bei steigender Rundenzahl und Schlüssellänge, und größeren Sboxen die Sicherheit steigt, allerdings wäre der Algorithmus dann nicht mehr so effizient. Die Herausforderung einen sicheren Algorithmus zu erstellen, liegt darin mit möglichst kleinen Laufzeitanforderungen und Speicherplatz zu arbeiten.

Die Sboxentabelle ist in [1] angegeben, deshalb ist es nicht erforderlich, dass sie noch mal angegeben wird.

Wenn man sich genauer über Stärken und Schwächen des Algorithmus informieren will, sollte man sich die kryptanalytischen Analysen differentielle und lineare Kryptanalyse zu Gemüte führen. Die differentielle Kryptanalyse führt mittels Charakteristiken mit einem ChosenPlainText-Angriff bei einem DES mit 13 Runden zum Erfolg, indem man den wahrscheinlichsten Schlüssel ermittelt, mit dem der Chiffretext verschlüsselt wurde.

5 Beschreibung des RSA-Algorithmus

5.1 Geschichte von RSA

RSA war ein patentiertes Produkt von RSA Security Systems. In den letzten Jahren jedoch lief dieses Patent aus, so dass es nun für eigene Zwecke gebraucht werden kann. Dass solch ein asymmetrisches Verschlüsslungsverfahren überhaupt möglich ist, wurde in der Öffentlichkeit erst 1976 von Martin Hellmann und Whitfield Diffie verkündet in dem Artikel „New Directions in Cryptography“. RSA selbst wurde von Ron Rivest, Adi Shamir und Leonard Adleman erfunden.
Damals wurde es im Scientific American unter der Kolumne „Mathematical Games“ veröffentlicht. Dieses Verfahren ist durch zahlentheoretische Beweise mathematisch fundiert bewiesen worden. Es wurde dadurch also viel besser abschätzbar, welche Sicherheit der Algorithmus besitzt. Bei symmetrischen Algorithmen ist dies noch nicht möglich.
Die Unsicherheitsfaktoren sind zahlentheoretisch ungelöste Probleme auf die ich nicht eingehen werde, aber vollständigkeitshalber erwähne: Es sind die diskreten Logarithmen und das Faktorisierungsproblem.

5.2 Mathematische Beschreibung

1. Bob erzeugt zwei zufällige große Primzahlen p und q aus Z
2. Bob multipliziert pq = n
3. Zusätzlich wird eine Zahl e aus Z gewählt wobei gilt:
   1 < e < Phi(n)
   mit Phi(n)= (p-1)(q-1) und der ggT von e und Phi(n)n muß 1 sein.

Ein paar Erläuterungen zu Phi() und dem ggT:
Der ggT ist der größte gemeinsame Teiler zweier Zahlen. Den ggT kennt man aus dem Schulunterricht. Phi() hingegen ist schon etwas mehr abgehoben. Es ist die eulersche Funktion, und beschreibt formal die Ordnung der primen Restklassengruppe modulo m, also Phi(m). Eine prime Restklassengruppe modulo m bildet einen Körper aus Z, die größte Zahl ist m-1 in diesem Körper. Da es ein Körper ist, sind die Verknüpfungen Addition und Multiplikation definiert, sowie das Inverse jeder Zahl. Das Phi() einer Primzahl ist bestimmt durch p-1
Das Phi() einer zusammengesetzten Zahl ist bestimmt durch $(p_1 - 1)(p_2 - 1)(p_3 - 1)...$ Die p's sind hier die Primzahlfaktoren
Jetzt wird eine Zahl d aus Z berechnet wobei 1 < d <Phi(n) mit

$$ed \equiv 1 \bmod(phi(n))$$

Dies ergibt sich aus ggT(e,phi(n)) = 1
Der öffentliche Schlüssel ist also (e,n) und der private (d,n)
e hat dadurch den Namen Verschlüsselungsexponent bekommen und d den Namen Entschlüsselungsexponent.
Das war die Theorie dazu.
Die Implementierung sieht aus wie folgt.
Die Formel für die Verschlüsselung lautet:

$$c_i = m_i^e \bmod n$$

Analog die Entschlüsselung:

$$m_i = c_i^d \bmod n$$

m steht für den Klartextblock, c für den Chiffretextblock, so ist i dann der Index des aktuellen Blocks. m als Zahl kann durch die Binärdarstellung des AsciiTextBlockes gewonnen werden, m muß hierbei allerdings kleiner n sein. Aber gehen wir noch weiter ins Detail.
Das Alphabet hat N Zeichen, wie oben schon erwähnt muß also der Klartextblock mit mindestens einem Zeichen aus N kleiner als n sein, da es sonst Sicherheitsprobleme gibt.
Es gilt also $N^k \leq n$. Die Technik des linken Ausdrucks stammt aus der Kombinatorik, bei einem Wort von k Zeichen kann jedes Zeichen aus dem Alphabet mit Länge N beliebig oft genommen werden.
Daraus können wir gleich die Anzahl der Blöcke bestimmen.

$$k = \lfloor \log_N n \rfloor$$

Nun wird also der Text in k Blöcke aufgeteilt und mit der obigen Verschlüsselungsformel wird jeder Block einzeln chiffriert. Dies entspricht und zwar nicht der allgemeinen ECB-Definition, das wird hier aufgrund mathematischer Probleme nicht zu schwer genommen. Zu erwähnen ist noch, dass RSA im CBC Mode möglich sein soll.
Bei der Dechiffrierung teilt man den Chiffretext einfach in n Blöcke und wendet obige Formel an.
Dieser Algorithmus läßt sich, von vielen diversen Implementierungsfehlern abgesehen, nur durch das Faktorisieren der Zahl n in p und q brechen. Im Internet wird zwar verbreitet, dass es in einer Arbeit einen Tip gibt, dass es möglich wäre, den Algorithmus RSA damit zu brechen. Es werden allerdings dazu keine genaueren Angaben gemacht.