Kryptographie

- Eine Einführung in die Verschlüsselung -

Grundlagen der Kryptographie und allgemeine Verschlüsselungsverfahren

Expertise matters

Inhalt

1 Zusammenfassung 2
2 Was ist Kryptographie? 2
3 Attacken auf verschlüsselte E-Mails 3
4 Mathematische Grundlagen der Kryptographie 5
5 Verschlüsslungsverfahren 6
5.1 Symmetrische Verfahren - Secret-Key-Verfahren 6
5.1.1 Allgemein 6
5.1.2 Klassische Chiffriersysteme 7
5.1.3 Strom- und Blockchiffren 8
5.1.4 Symmetrische Algorithmen 9
5.2 Asymmetrische Verfahren – Public-Key-Verfahren 11
5.2.1 Allgemein 11
5.2.2 Asymmetrische Verschlüsselungsalgorithmen 13
5.3 Hybridverfahren 14
6 Protokolle und Funktionen 15
6.1 Protokolle zum Schlüsselaustausch 15
6.2 Hashfunktionen 16
6.2.1 Konstruktionsprinzipien und Algorithmen 17
6.2.2 Algorithmen 17
6.3 Authentifizierungscodes (MAC) 18
6.4 Digitale Signaturen 19
6.4.1 Signaturverfahren 20

1 Zusammenfassung

Der Umfang der Kommunikation per E-Mail ist in den letzten Jahren stark gewachsen und wächst immer weiter. Mittlerweile gibt es kaum noch ein Unternehmen, welches E-Mail für die Abwicklung ihrer Geschäftsprozesse sowohl innerhalb des Unternehmens als auch mit externen Geschäftspartnern nicht verwendet. Neben schnellen Reaktionszeiten, ständiger Erreichbarkeit und kostengünstiger Kommunikation stehen die Fragen nach der Sicherheit von E-Mails im Vordergrund, u. A. die Sicherstellung der E-Mail-Vertraulichkeit. Viele Unternehmen setzen deshalb auf E-Mail-Sicherheitslösungen, die auch die Verschlüsselung von E-Mails beinhalten. Die iQ.Suite von GBS Software bietet eine umfassende und vollständige richtlinienbasierte Lösung mit dem Modul iQ.Suite Crypt Pro sowie eine webbasierte Lösung für Empfänger ohne eigene Verschlüsselungslösung – iQ.Suite WebCrypt Pro/Live.

Die vorliegende Dokumentation behandelt die Grundlagen der Kryptographie und gibt einen Überblick über die verschiedenen grundsätzlichen Verschlüsselungsmöglichkeiten. Näheres zur Public-Key Infrastruktur (PKI) und die Implementierung in iQ.Suite Crypt Pro finden Sie in weiteren Whitepapern, die Sie sich auf unserer Whitepaper-Downloadseite herunterladen können.

2 Was ist Kryptographie?

Du mußt verstehn!
Aus Eins mach Zehn
und Zwei laß gehn,
und Drei mach gleich,
so bist du reich.

Verlier die Vier!
Aus Fünf und Sechs,
so sagt die Hex'
mach Sieben und Acht,
so ist´s vollbracht:

und Neun ist Eins,
und Zehn ist keins.
Das ist das Hexen-Einmaleins.

Goethe, Faust I

Das ist geheimnisvoll, magisch und kryptisch? Zumindest ist es verschlüsselt. Ob Goethe sein Hexeneinmaleins aber tatsächlich im mathematischen Sinn erdachte und welches System bei der Entschlüsselung angewendet werden muss, ist unter Experten noch heute umstritten, auch wenn schon einige magische Quadrate dabei entdeckt wurden.

Kryptographie, oft auch als Kryptologie bezeichnet, ist die (mathematische) Wissenschaft, die sich mit Methoden und Verfahren zur Ver- und Entschlüsselung von Daten beschäftigt ein einfaches und klassisches Beispiel für ein kryptographisches Verfahren ist die so genannte CAESAR-Chiffre. Das Verfahren ist einfach:

Ein Text wird verschlüsselt, indem der Buchstabe A durch den Buchstaben D ersetzt wird, B durch den Buchstaben E, C durch den Buchstaben F und so weiter. Der Schlüssel ist dabei die Anzahl der Stellen, um die die Buchstaben im Alphabet verschoben werden, d.h. in diesem Fall 3. Aus GROUP würde dann durch Verschlüsseln JURZS.

Ein kryptographisches Verfahren gilt als sicher, wenn es trotz Kenntnis des Verfahrens schwierig ist, eine verschlüsselte Nachricht ohne Kenntnis des Schlüssels zu entschlüsseln. Schwierig heißt in der Praxis, dass die Entschlüsselung in vertretbaren Rahmen nicht durchführbar ist. Die CAESAR-Chiffre ist deshalb kein sicheres Verfahren, da es durch einfaches Ausprobieren relativ leicht ist, den Klartext zu finden.

3 Attacken auf verschlüsselte E-Mails

Bei jedem Kryptosystem versucht der potenzielle Angreifer natürlich, alle Schwachpunkte so gut wie möglich auszunutzen. Als Kryptoanalyse bezeichnet man die Lehre, ohne Kenntnis des Schlüssels an die geheimen Daten zu gelangen. Angreifer werden deshalb auch als Kryptoanalytiker bezeichnet, da die gleichen grundlegenden Verfahren zur „legalen“ Kryptoanalyse (Bewertung der kryptographischen Stärke) und „illegalen“ Kryptoanalyse (unbefugte Entschlüsselung von Daten, um die ursprüngliche Information zurück zu gewinnen) verwendet werden.

Bei den weiter unten besprochenen, heute üblichen Verschlüsselungsverfahren tauschen zwei Kommunikationspartner Daten über einen unsicheren Kanal aus. Für die Sicherheitsbetrachtungen wird davon ausgegangen, dass einem potenziellen Angreifer alle Informationen über das verwendete System (bis auf den eigentlichen Schlüssel) zur Verfügung stehen, so dass er unbeschränkten Zugriff auf die Kommunikation hat.

Digitale Signaturen ermöglichen die Prüfung, ob ein Dokument tatsächlich vom angegebenen Absender stammt und können dadurch z.B. gewährleisten, dass man immer mit demselben Kommunikationspartner kommuniziert. Dies verschafft aber keine Gewähr über die tatsächliche Identität. Diese Gewähr erhält man u.A. dadurch, dass man sich in einem Trustcenter persönlich anmelden muss.

Angreifer unterscheidet man nach:

Passiver Angreifer - hört den Übertragungskanal mit dem Ziel ab, Nachrichten zu entschlüsseln

Aktiver Angreifer - ist auch in der Lage, Übertragungen zu manipulieren.

Der Angreifer hat folgende Möglichkeiten gegen ein Chiffriersystem:

Ciphertext-Only-Angriff (nur Chiffretext): die schwächste aller Voraussetzungen. Der Angreifer hat nur Chiffretext und keine Informationen, die über die generell angenommenen Informationen hinausgehen.

Known-Plaintext-Angriff (bekannter Klartext): Der Angreifer besitzt eine bestimmte Anzahl Chiffretext-Klartext-Paare.

Chosen-Plaintext-Angriff (frei wählbarer Klartext): Der Angreifer ist in der Lage, frei gewählten Nachrichten zu chiffrieren, d.h. er hat Zugriff auf das Chiffriergerät. Diese auf den ersten Blick unrealistische erscheinende Möglichkeit des Angriffs ist die Regel bei Public-Key-Verschlüsselung (siehe Asymmetrische Verfahren – Public-Key-Verfahren.

Chosen-Ciphertext-Angriff (frei wählbarer Chiffretext): Der Angreifer hat die Möglichkeit, einen von ihm gewählten Chiffretext zu dechiffrieren.

Ein Angreifer hat folgende prinzipielle Methoden des Angriffs zur Verfügung:

Brute-Force-Methoden

Das sind generische Methoden, um den geheimen Schlüssel zu bestimmen. Dabei werden systematisch alle Möglichkeiten ausprobiert. Bei diesem so genannten Brute-Force-Angriff werden nacheinander alle Schlüssel zur Verschlüsselung eingesetzt und man ist erfolgreich, falls der erhaltene Klartext einen Sinn ergibt. Der Erfolg eines solchen Angriffs hängt von der Länge des Schlüssels und von der Rechenleistung der eingesetzten Computer ab.

Verschlüsselungssysteme, bei denen der zur Verfügung stehende Schlüsselraum nicht groß genug ist, sind unzureichend. Deshalb sollte die Anzahl der Operationen zur Berechnung des Schlüssels mindestens in der Größenordnung von 10^{25} liegen.

Bei Hashfunktionen [1] , die schlüsselunabhängig sind, versucht man mit einer Geburtstagsattacke[2] das Verfahren zu knacken. Es handelt sich dabei um einen Angriff auf die Kollisionsresistenz: Man versucht, zwei beliebige Nachrichten M und M' zu finden, deren Hashwerte h(M) und h(M') übereinstimmen. Eine Hashfunktion sollte Ausgaben von 160 Bit Länge erzeugen, damit bei einer Attacke 2^{80} Operationen durchgeführt werden müssen. Das entspricht der Sicherheit eines 80-Bit-Schlüssels bei symmetrischen Verfahren.

Bei Public-Key-Verfahren (siehe Asymmetrische Verfahren – Public-Key-Verfahren) ist die Form des Angriffs nicht eindeutig. Bei diesen Verfahren wird versucht, die zugrunde liegenden Probleme aus der Mathematik zu knacken. Man spricht auch von „harten" Problemen. Nach

[1] Hashfunktion: Eine Funktion, die für eine beliebige Eingabe einen Funktionswert (den Hashwert) mit fest vorgegebener Länge berechnet. Diese Funktionen werden dazu benutzt, das elektronische Gegenstück eines Fingerabdrucks zu erzeugen.

[2] Der Name beruht auf dem statistischen Phänomen, dass bereits in einer sehr kleinen Gruppe wenigstens zwei Personen mit 50%iger Wahrscheinlichkeit am gleichen Tag Geburtstag haben

dem heutigen Erkenntnisstand sollte z.B. beim RSA-Algorithmus eine Schlüssellänge von 1024 Bit verwendet werden

Statistische Methoden

Diese Verfahren machen sich die statistische Struktur des Klartextes zu nutzen. Die Verfahren analysieren die Häufigkeit von Zeichen und Zeichengruppen des Chiffrates und vergleichen Sie mit der des Klartextes.

Analytische Methoden

Sind speziell auf ein Chiffriersystem zugeschnitten und versuchen systemimmanente Schwachstellen auszunutzen. Ziel ist es, ein Analyseverfahren zu finden, das die Berechnung der Schlüssel aus (Klartext und) Chiffrat erlaubt.

Für symmetrische Verschlüsselungsverfahren kann zwischen folgenden analytischen Methoden unterschieden werden:

Differentielle Kryptoanalyse – Es werden zwei Chiffretexte betrachtet, deren zugehörige Klartexte gewisse Differenzen aufweisen. Durch Verändern des Klartextes wird versucht, Informationen über den Zusammenhang zwischen Chiffretext und Klartext zu erlangen.

Lineare Kryptoanalyse – Es handelt sich um eine Attacke, die nur den Klartext benötigt. Es wird versucht, einfache („lineare“) Abhängigkeiten zwischen den Bits des Klartextes und des Chiffretextes zu entdecken und auszunutzen, um damit Informationen über den Schlüssel zu erhalten.

Moderne Algorithmen achten darauf, sich gegen diese beiden Arten von Angriffen zu schützen.

4 Mathematische Grundlagen der Kryptographie

Die Basis der Kryptographie ist die modulare Rechnung. Modulares Rechnen ist aus der Schule bekannt. „Klein Hänschen soll um 12:00 Uhr zu Hause sein und verspätet sich um 13 Stunden – um welche Zeit ist er nach Haus gekommen?“ Bei dieser „Uhrenarithmetik“ gibt es keine 25 Uhr, sondern es ist dann wieder 1 Uhr. Man rechnet 25 modulo 24 und erhält 1. Anders ausgedrückt ergibt sich

$$25 \equiv 1 \bmod 24$$

In der Zahlentheorie heißen zwei ganze Zahlen *a* und *b* kongruent modulo *m* (wobei *m* eine positive ganze Zahl ist), wenn *m* die Differenz von (*a*-*b*) teilt. Zwei Zahlen sind also kongruent modulo einer Zahl *m*, wenn sie bei der Division durch *m* denselben Rest ergeben.

Man bezeichnet die Menge aller zu *a* (modulo *m*) kongruenten ganzen Zahlen als die Restklasse von *a* modulo *m. m* wird als Modulus bezeichnet.

Für einen Modulus *m* wird die Menge der ganzen Zahlen {0, 1, 2, …, n-1} mit Z_m bezeichnet. Zusammen mit der Addition modulo *m* formt diese Menge eine mathematische Struktur, die formal Gruppe genannt wird.

Die mathematisch exakte Definition ist:

Eine Gruppe G ist ein mathematisch abstraktes Objekt, das sich aus einer Menge G und einer Operation ° zusammensetzt, die auf einem Paar von Elementen aus der Menge definiert ist. Für eine Gruppe gelten eine Reihe von Eigenschaften und Definitionen.

Die Vorteile der modularen Rechnung sind u. A., dass die Länge der zu betrachtenden Zahlen durch die Länge des Modulus begrenzt wird. Besonders die in der Kryptographie weit verbreitete Operation der Exponentiation oder Potenzierung von Elementen profitiert von diesem Zustand. Für die Verwendung innerhalb der Kryptographie ist der Einsatz einer so genannten Langzahlarithmetik notwendig, d.h. eine Arithmetik, die in der Lage ist, mit Zahlen zu rechnen, die den üblichen Darstellungsbereich von Zahlen auf Computern übersteigen.

Mathematische Probleme, die speziell in der asymmetrischen Kryptographie Anwendung finden, sind:

das Knapsack-Problem – Man hat einen Rucksack (Knapsack), welcher exakt eine vorgegebene Masse fassen kann, sowie eine unbestimmte, große Anzahl von Objekten mit verschiedenen Massen. Die Frage ist, welche Objekte man auswählen muss, um den Rucksack optimal zu füllen.

das Problem der Faktorisierung – Beim Problem des Faktorisierens (IF -Integer Factorization) geht es um die Tatsache, dass sich zwei große Primzahlen (in der Größenordnung von über 100 Dezimalstellen) relativ problemlos multiplizieren lassen, die Umkehrung jedoch, also die Bestimmung der Primfaktoren bei einem vorgegebenen Produkt, ungleich schwieriger ist. Der RSA-Algorithmus als das bekannteste Public-Key-Verfahren basiert auf diesem Problem.

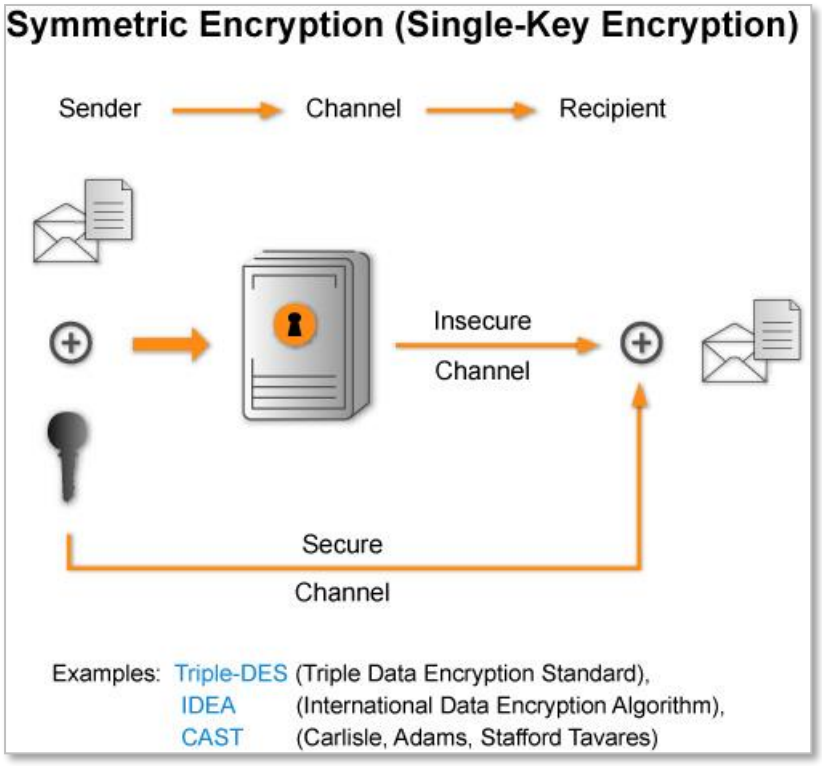
das Problem des diskreten Algorithmus – Bei dem Problem des diskreten Algorithmus (DL) versucht man die Umkehrung der Exponentiation zu berechnen, d.h. zu einem gegebenen $y = g^x$, wobei g allgemein bekannt und x „groß“ ist, den Wert $x = \log_a y$ zu berechnen.

5 Verschlüsslungsverfahren

5.1 Symmetrische Verfahren - Secret-Key-Verfahren

5.1.1 Allgemein

Symmetrische Verschlüsselungsverfahren sind die Verfahren, die zum Ver- und Entschlüsseln jeweils den gleichen Schlüssel, den privaten Schlüssel (Secret-Key), verwenden. Die schon erwähnte CAESAR-Chiffre ist ein symmetrisches Verfahren. Mit dem Besitz dieses Schlüssels ist es möglich, Nachrichten zu ver- und zu entschlüsseln. Das Problem bei diesen Verfahren ist, den Schlüssel auf einem sicheren Weg zwischen den Kommunikationspartnern zu vereinbaren und auszutauschen.



Dieses Verfahren kommt z.B. zur Anwendung, wenn Informationen mehreren Benutzern parallel (zeitgleich) zugänglich gemacht werden sollen. Bei einer "Eins-zu-Eins"-Kommunikation haben diese Verfahren einen erheblichen Verwaltungsaufwand. Um sicherzustellen, dass ein Benutzer A mit einer Anzahl von n Benutzern kommunizieren kann (es sind *n+1* Benutzer beteiligt, der Benutzer A kann an *n* Personen, diese aber nur an A senden) und zusätzlich sichergestellt ist, dass nur diese beiden Personen Daten austauschen können, nicht aber Dritte die ausgetauschten Daten entschlüsseln können, müssen *n* Schlüssel verfügbar sein. Das bedeutet, dass jeder einzelne Benutzer eine große Anzahl von Schlüsseln verwalten muss, und zwar genau so viele Schlüssel wie Benutzer existieren, mit denen er kommunizieren möchte.

Dehnt man die Kommunikation nun auf n Teilnehmer untereinander aus (jeder kann mit jedem kommunizieren) und will sicherstellen, dass diese n Teilnehmer jeweils untereinander Mails austauschen können, ohne dass irgendein anderer diese Mails lesen kann, so sind beim symmetrischen Verschlüsselungsverfahren *n* mal *(n-1) / 2* Schlüssel notwendig. Beispielsweise sind das 499 500 Schlüssel für 1000 Benutzer.

5.1.2 Klassische Chiffriersysteme

Im klassischen Sinn werden alle Chiffriersysteme auf zwei einfache Grundprinzipien zurückgeführt. Es sind die Transposition und die Substitution.

Bei der Transposition werden die Zeichen eines Klartextes nach einem vorgegebenen Schema vertauscht (Permutation des Klartextes). Im Gegensatz dazu werden bei der Substitution die Zeichen

oder Zeichenketten durch andere Zeichen(-ketten) ersetzt. Durch Wiederholen und abwechselnde nacheinander folgende Ausführung von Transposition und Substitution erhält man eine Produktchiffre.

Die Substitutionssysteme lassen sich wie folgt klassifizieren:

Monoalphabetische Substitution: Jedes Zeichen bzw. jede Zeichenkette des Klartextalphabetes wird durch ein fest zugeordnetes Zeichen bzw. eine Zeichenkette über einem Chiffrealphabet B ersetzt. Die schon genannte Caesar-Chiffre ist ein Beispiel für solch ein System.

Polyalphabetische Substitution: Jedes Zeichen bzw. jede Zeichenkette des Klartextalphabetes wird durch ein fest zugeordnetes Zeichen bzw. eine Zeichenkette über das Chiffrealphabet B1,.., Bn ersetzt. Ein Beispiel für solch ein Substitutions-Verfahren ist die Vernam-Chiffre (Frankreich, 1917). Eine Sonderform der Vernam-Chiffre ist der One-Time-Pad. Als Schlüssel wird dabei eine Zufallsfolge verwendet, die nur einmal verwendet wird. Wird dabei eine „echte Zufallsfolge“ verwendet, der Schlüssel also beispielsweise durch das Werfen einer Münze oder das Ziehen von Zahlen erzeugt, so ist das System absolut, d.h. informationstheoretisch beweisbar, sicher.

Monographische Substitution: Hier werden Einzelzeichen ersetzt. Die Caesar-Chiffre ist ein Beispiel für eine monoalphabetische, monographische Substitution.

Polygraphische Substitution: Hier werden jeweils Zeichenketten ersetzt.

5.1.3 Strom- und Blockchiffren

Längere Nachrichten müssen in der Regel vor der Chiffrierung aufgespaltet werden, da ein Chiffreverfahren normalerweise einen Input fester Länge benötigt. Man teilt symmetrische Algorithmen in zwei Kategorien:

1. Stromalgorithmen oder Stromchiffren – Der Klartet wird zeichenweise bearbeitet. Dabei kann ein Zeichen z.B. ein Bit oder ein Byte sein. Bei Hardwareimplementierungen sind diese Verfahren schneller als blockorientierte Verfahren, da die bit- bzw. byteweise Verarbeitung einer Hardwarelösung besser angepasst ist.
2. Blockalgorithmen oder Blockchiffren - Der Klartext wird in Bitgruppen, die Blöcke genannt werden, bearbeitet. Anwendung finden diese Algorithmen normalerweise bei softwarebasierten Systemen. Es ist jedoch sinnvoll, dass auch bei Blockchiffren die Chiffrierung von der Bearbeitung vorhergehender Blöcke abhängig ist. Damit werden identische Klartextblöcke auf unterschiedliche Chiffretexte abgebildet und einem potenziellen Angreifer wird es schwerer gemacht.

Die Verkettung aufeinander folgender Blöcke wird auch als kryptographischer Modus bezeichnet. Die verschiedenen Varianten sind das electronic codebook (ECB), cipher block chaining (CBC), cipher feedback (CFB) oder output feedback (OFB).

Die Erzeugung des geheimen Schlüssels erfolgt durch eine per Zufallsgenerator ausgewählte Bitfolge, die als Schlüssel fungiert.

5.1.4 Symmetrische Algorithmen

DES (Data Encryption Standard)

DES ist der Klassiker der kryptographischen Verfahren. Er wurde aufgrund einer Ausschreibung für einen einheitlichen Verschlüsselungsalgorithmus durch das National Bureau of Standards (NBS, heute NIST) durch die IBM entwickelt und 1976 vorgestellt. Alle 5 Jahre zertifiziert das NIST den DES-Algorithmus. Die letzte Zertifizierung fand 1999 statt, jedoch unter Verwendung der DES-Variante Triple-DES, da DES nicht mehr den heutigen Sicherheitsanforderungen genügt.

DES ist eine symmetrische Blockchiffre, die Daten in Blöcken von 64 Bit verschlüsselt. Der Algorithmus erhält als Eingabe einen Block von 64-Bit-Klartext und liefert als Ausgabe einen 64-Bit-Chiffretext. Die Schlüssellänge beträgt 56 Bit und jedes achte Bit dient einer Paritätsprüfung, so dass der Schlüssel als gewöhnliche 64-Bit-Zahl ausgedrückt wird.

DES arbeitet auf einem 64-Bit-Block des Klartextes. Nach einer Eingangspermutation wird dieser Block in eine jeweils 32 Bit lange rechte und linke Hälfte zerlegt. Jetzt folgen 16 Runden identischer Operationen, in denen die Daten mit dem Schlüssel kombiniert werden. Nach der sechzehnten Runde werden rechte und linke Hälfte zusammengefügt. Eine Schlusspermutation, die zur Eingangspermutation invers ist, schließt den Algorithmus ab.

Der Schlüsselraum für DES umfasst nur 256 verschiedene Schlüssel, von denen einige als schwach bekannt sind. Mit der linearen Kryptoanalyse ist es möglich, eine Known-Plaintext-Attacke3 mit 247 bekannten Klartexten zu unternehmen.

Durch die Methode der differentiellen Kryptoanalyse wird der Komplexitätsgrad von 256 auf 247 reduziert. DES ist mit 56-Bit effektiver Schlüssellänge als nicht mehr sicher einzustufen.

Triple-DES

Der Triple-DES ist eine Variante des DES, dabei wird der DES-Algorithmus dreimal mit verschiedenen Schlüsseln angewendet. Dabei wird ausgenutzt, dass DES keine mathematische Gruppe ist, so dass man beim Anwenden der drei Schlüssel in einem anderen Schlüsselraum landet. Die Dreifachverschlüsselung mit zwei Schlüsseln funktioniert so:

Ein Block wird erst mit dem ersten Schlüssel chiffriert, anschließend mit dem zweiten dechiffriert und dann wieder mit dem ersten chiffriert. Dieses Verfahren wird als encrypt-decrypt-encrypt (EDE)-Modus bezeichnet, welcher zur Verbesserung von DES für die Standards X9.17 und ISO 8732 modifiziert wurde. Triple-DES findet u.a. bei der Berechnung der neuen EC-PIN und beim Homebanking-Standard (HBCI) Anwendung.

AES

Das National Institut of Standards and Technology (NIST) will einen DES-Nachfolger als neuen Standard für symmetrische Verschlüsselungsverfahren festlegen, den Advanced Encryption Standard (AES).

---

[3] Angreifer kennt Klartext und zugehörigen Chiffretext oder mehrere solche Paare und versucht den verwendeten Schlüssel herauszufinden

AES muss folgende Bedingungen erfüllen:

- Ein symmetrisches Kryptographieverfahren
- Eine Blockchiffre
- Eine Blockgröße von 128 Bit
- Eine Schlüsselgröße von 128, 192 und 256 Bit

Die Finalisten der dritten und letzten Runde sind MARS, RC6, Rijndael, Serpent, Twofish

CAST

CAST ist nach seinen Entwicklern Carlisle Adams und Stafford Tavares benannt und wurde am 23.04.1996 zum Patent angemeldet. Die symmetrische Blockchiffre CAST ist ein Feistel-Netzwerk[4]. CAST hat eine Blocklänge von 64-Bit-Blockchiffre und eine Schlüssellänge von 40 - 128 Bit. Der Vorteil gegenüber DES ist das Fehlen schwacher Schlüssel.

CAST wird in PGPhone verwendet. Auch Northern Telecom, IBM, Tandem und Microsoft verwenden CAST in ihren Produkten. CAST ist die Standardchiffre in PGP.

Blowfish

Der bekannte Kryptograph Bruce Schneider entwickelte das Verfahren und stellte es im Dezember 1993 vor. Blowfish ist ein symmetrisches Verschlüsselungsverfahren mit einem Feistel-Netzwerk. Die Blocklänge ist eine 64-Bit-Blockchiffre, mit einer Schlüssellänge von 8 - 448 Bit. Bisher ist kein für die Praxis relevanter Angriff bekannt. Blowfish ist wegen bestimmter Modifikationen und der freien Verfügbarkeit ohne Lizenzgebühren weit verbreitet. Der Algorithmus ist wegen des hohen Speicherplatzbedarfes ungeeignet für Chipkarten. Ebenfalls ungeeignet ist Blowfish für Anwendungen, bei denen der Schlüssel häufig gewechselt werden muss. Blowfish wird in den beiden Programme für abhörsicheres Telefonieren PGPhone und Nautilus verwendet.

IDEA (International Data Encryption Algorithm)

Der IDEA-Algorithmus wurde Anfang der 90-ziger Jahre von der ETH Zürich entwickelt. Die Schweizer Firma ASCOM hat auf IDEA ein Patent angemeldet.

Wegen der Größe der Schlüssellänge ist IDEA immun gegen Brute-Force-Angriffe und gegen die ebenso gefährliche Kryptoanalyse. Bekannt wurde IDEA als Algorithmus im Verschlüsselungsprogramm PGP.

[4] Ein nicht unwesentliches Problem bei der Chiffrierung von Daten ist, dass die Verschlüsselungsfunktion umkehrbar sein muss, um eine korrekte Entschlüsselung von Texten zu ermöglichen. Bei Feistel-Chiffren wird diese Anforderung durch ein bestimmtes Design sichergestellt.

5.2 Asymmetrische Verfahren – Public-Key-Verfahren

5.2.1 Allgemein

Asymmetrische Verfahren bieten effektive Möglichkeiten zur Datensicherung. Beim asymmetrischen Verschlüsselungsverfahren sind zwei Schlüssel beteiligt. Beide Schlüssel sind mathematisch voneinander abhängig, jedoch kann keiner dieser beiden Schlüssel aus dem anderen berechnet werden (mit vertretbarem Aufwand).

Es kann mit beiden Schlüsseln verschlüsselt werden, jedoch nur mit dem jeweils anderen Schlüssel ist eine Entschlüsselung möglich. Dieses Verfahren nennt man auch das Verfahren des "öffentlichen und privaten Schlüssels".

Diese Eigenschaften ermöglichen es, einen der beiden Schlüssel zu veröffentlichen – den öffentlichen Schlüssel, genannt Public-Key. Der private Schlüssel (Privat-Key) wird an einem sicheren Ort geheim gehalten. Solche Verfahren werden Public-Key-Verfahren genannt. Es besteht Unabhängigkeit von der Sicherheit oder Unsicherheit des verwendeten Übertragungsweges. Jeder, der dieses Verfahren verwendet, besitzt ein eigenes Schlüsselpaar, bestehend aus einem privaten (geheimen) und einem zugehörigen öffentlichen Schlüssel.

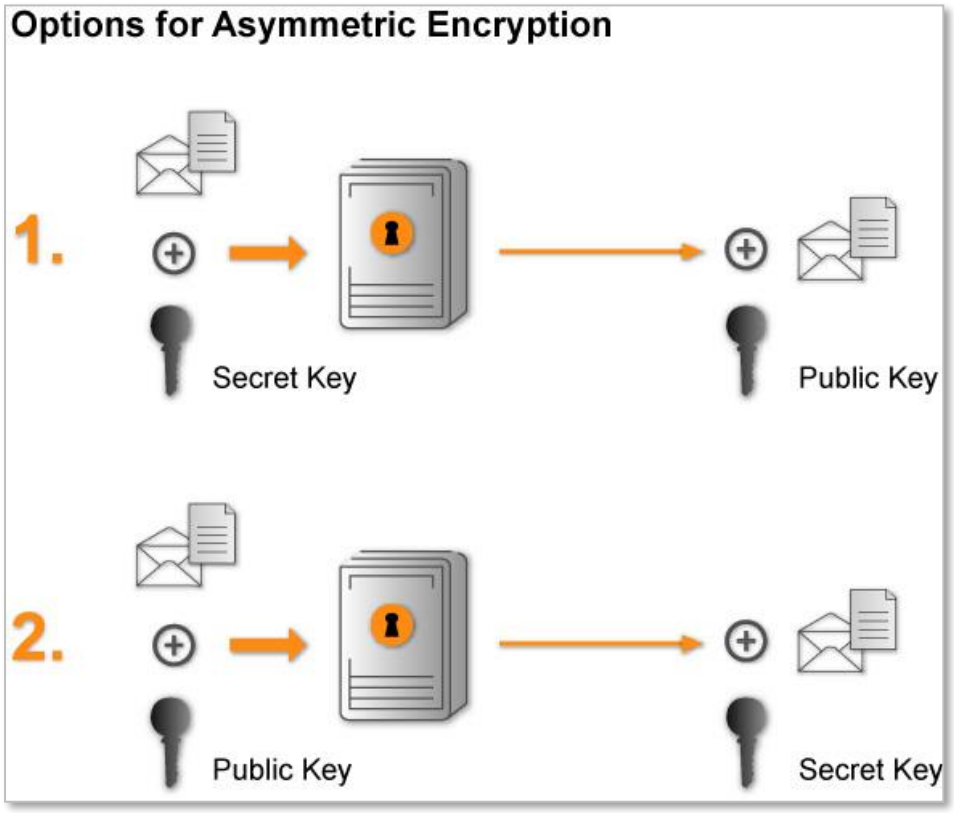
Soll ein Teilnehmer eine verschlüsselte Nachricht erhalten, so wird dessen öffentlicher Schlüssel zum Verschlüsseln verwendet. Der Empfänger ist der einzige, der die Nachricht mit seinem privaten Schlüssel entschlüsseln kann. Integrität und Authentizität, d.h. der Beweis über den Besitz eines bestimmten Schlüssels, wird durch Verschlüsseln einer Nachricht mit dem eigenen privaten Schlüssel erreicht. Jeder kann durch Verwenden des öffentlichen Schlüssels feststellen, ob die Nachricht tatsächlich von dem vermeintlichen Absender stammt.

Misslingt die Entschlüsselung, dann wurde entweder die Nachricht unterwegs manipuliert (Verletzung der Daten-Integrität) oder der Absender ist nicht der, für den er sich ausgibt (Verletzung der Authentizität). Durch Verschlüsseln einer Nachricht mit dem eigenen privaten Schlüssel wird das elektronische Analogon einer Unterschrift erreicht – die so genannte digitale Signatur.

Dies hat zur Folge, dass beliebige Teilnehmer Daten zwar verschlüsseln können, die Entschlüsselung jedoch nur bei demjenigen stattfinden kann, der das passende Pendant hat. Normalerweise ist das genau eine einzige Person, nämlich der Empfänger (der üblicherweise auch den öffentlichen Schlüssel zur Verfügung gestellt hat).

Für den Fall, dass *n* Personen miteinander (jeder mit jedem) kommunizieren wollen, so gibt es genau *n* asymmetrische Schlüsselpaare. Von diesen genau *n x 2* Schlüsseln werden genau *n* Schlüssel, nämlich die "öffentlichen Schlüssel" veröffentlicht.

Der große Vorteil dieses Verfahrens besteht darin, dass der öffentliche Schlüssel jedes Teilnehmers innerhalb eines Netzwerks nur ein einziges Mal verfügbar gemacht werden muss. Alle Teilnehmer sollten dann darauf Zugriff haben, damit sie in der Lage sind, für den Besitzer des jeweiligen öffentlichen Schlüssels Informationen zu verschlüsseln und diesem zu schicken.



Auch neu hinzukommende Teilnehmer können dieser Person sofort verschlüsselte Informationen senden, ohne dass der Empfänger irgendetwas veranlassen muss. Eine verschlüsselte Mail kann also einen Empfänger sogar dann erreichen, wenn der Empfänger den Absender überhaupt nicht kennt.

Im Fall des symmetrischen Verschlüsselungsverfahrens könnten nur Benutzer, die einen Schlüssel ausgetauscht haben, sich gegenseitig codierte Meldungen zukommen lassen.

In der Praxis sind Public-Key Algorithmen kein Ersatz für symmetrische Algorithmen, da sie um Größenordnungen langsamer sind.

Deshalb werden sie z.B. zur Chiffrierung von Schlüsseln für symmetrische Verfahren verwendet. Diese Verfahren werden dann hybride Verfahren genannt (siehe Hybridverfahren).

5.2.2 Asymmetrische Verschlüsselungsalgorithmen

Es gibt eine Vielzahl von asymmetrischen Verschlüsselungsalgorithmen. Einige werden kurz vorgestellt.

RSA

RSA ist das bekannteste asymmetrische Verschlüsselungsverfahren. Das Verfahren wurde 1976 von den Forschern Ron Rivest, Adi Shamir und Leonard Adleman vorgestellt und nach ihnen benannt.

RSA ist ein asymmetrischer Algorithmus, der auf dem IF-Problem5 beruht. Zur Erzeugung der beiden Schlüssel wählt man zufällig zwei große Primzahlen p und q, die in etwa gleich lang sein sollten, um die maximale Sicherheit zu gewährleisten. Man berechnet n = p x q.

Dann wählt man zufällig einen Chiffrierschlüssel *e* mit *e* relativ prim zu (p-1)(q-1). Anschließend wird der Dechiffrierschlüssel d berechnet.

Der öffentliche Schlüssel besteht dann aus *e* und *n* und der private Schlüssel aus *d*. Die Verschlüsselung hat dann die Form

$$c \equiv m^e \bmod n$$

und die Entschlüsselung hat die Form

$$m \equiv c^d \bmod n.$$

Dazu wird die Nachricht m in Blöcke mi zerlegt, die kleiner als n sind. Dann besteht die verschlüsselte Nachricht c aus Nachrichtenblöcken ci gleicher Größe.

Es sollte beachtet werden, dass die Parameter p, q und e so „zufällig wie möglich“ gewählt und p und q geheim gehalten werden sollten, damit das Verfahren sicher ist.

Die Kryptoanalyse konnte die Sicherheit von RSA weder beweisen noch widerlegen. Die Sicherheit beruht auf der Schwierigkeit, große Zahlen zu faktorisieren. Der öffentliche und private Schlüssel hängen von einem Paar großer Primzahlen ab (mehr als 150 Stellen). Es zeigt sich immer deutlicher, dass die derzeit oftmals eingesetzten 512 Bits RSA-Verschlüsselung nicht mehr sicher sind. Das Problem ist aber, dass sich stärkere RSA-Verschlüsselungen von 2048 Bit oder größer in Smartcards nur schwer oder gar nicht einsetzen lassen.

DLIES

DLIES ist ein asymmetrisches Verfahren zur Chiffrierung, welches aus einem Schlüsselaustauschverfahren und einem MAC[6] (siehe auch Hashfunktionen) besteht und auf

[5] Integer Factoring, kurz IF-Problem: Faktorisierung ist das Problem, für eine gegebene Zahl die Primfaktoren zu bestimmen. Für den Fall n=pxq (d.h. n ist das Produkt zweier Primzahlen p und q) ist dies das dem RSA-Algorithmus zugrunde liegende Problem.

[6] Ein Message Authentication Code (MAC) erweitert eine Nachricht mit Hilfe eines geheimen Schlüssels um spezielle redundante Informationen, welche zusammen mit der Nachricht gespeichert bzw. übertragen werden, um im Nachhinein die Authentifizierung der Nachricht zu ermöglichen, siehe auch Authentifizierungscodes (MAC).

einem Diffie-Hellman-Problem [7] basiert. Die frühere Bezeichnung war DHAES, die dann aufgrund der Kollision der Namen mit dem DES-Nachfolger AES geändert wurde.

Wenn sich zwei Partner Nachrichten zusenden wollen, wird ein gemeinsames Geheimnis g^{uv} vereinbart, welches zusammen mit dem öffentlichen Schlüssel g^u des einen Partners durch eine Hashfunktion[8] komprimiert wird. Dieser Hashwert wird aufgeteilt. Ein Anteil geht in den MAC.

Der andere Anteil wird zusammen mit der Nachricht M durch einen symmetrischen Algorithmus zum Chiffretext SYM(M) verschlüsselt, welcher der zweite Input für den MAC ist. Nach dem Anwenden des MAC bekommt man als Output das TAG. Die Daten, die dann von dem einen Partner an den anderen geschickt werden, bestehen aus dem öffentlichen Schlüssel g^u, dem Chiffretext SYM(M) und dem Tag.

Die Sicherheit von DLIES beruht auf der Schwierigkeit des Diffie-Hellman-Problems und der Annahme, dass der zugrunde liegende symmetrische Algorithmus, die Hashfunktion und der MAC sicher sind.

EC-IES

ES-IES ist eine Variante des DLIES-Schematas, bei der man über elliptische Kurven arbeitet. Er wird im IEEE-Standard beschrieben.

5.3 Hybridverfahren

Hybridverfahren sind eine Kombination aus symmetrischen und asymmetrischen Verfahren. Sie nutzen die Vorteile beider Methoden: die Schnelligkeit symmetrischer Kryptographie und die Sicherheit asymmetrischer Kryptographie. Sie werden deshalb vorzugsweise für große Datenmengen eingesetzt.

Die Nachricht wird mittels eines so genannten Sitzungsschlüssels, der nur einmal verwendet wird, symmetrisch verschlüsselt. Dieser Sitzungsschlüssel wird dann asymmetrisch mit dem öffentlichen Schlüssel des Empfängers verschlüsselt und der verschlüsselte Schlüssel wird der Nachricht angehängt. Der Empfänger kann mit seinem privaten Schlüssel den Schlüssel und anschließend die Nachricht entschlüsseln. Für „lange“ Nachrichten ist dieses Vorgehen effektiver als eine asymmetrische Verschlüsselung der gesamten Nachricht. Da der Sitzungsschlüssel nur ein einziges Mal verwendet wird, ist diese Methode genauso sicher wie eine asymmetrische Verschlüsselung.

[7] Diffie-Hellman (DH): siehe Protokolle zum Schlüsselaustausch.

[8] Hashfunktion: Eine Funktion, die für eine beliebige Eingabe einen Funktionswert (den Hashwert) mit fest vorgegebener Länge berechnet. Diese Funktionen werden dazu benutzt, das elektronische Gegenstück eines Fingerabdrucks zu erzeugen.

6 Protokolle und Funktionen

6.1 Protokolle zum Schlüsselaustausch

Eines der Hauptprobleme bei der symmetrischen Verschlüsselung besteht darin, dass beide Partner über denselben geheimen Schlüssel verfügen müssen. Ein geheimer Schlüssel muss ausgetauscht werden, bevor verschlüsselte Nachrichten verschickt werden können.

Für die Verständigung über einen geheimen Schlüssel, ohne dass ein Dritter Kenntnis von dem Schlüssel erhält, werden folgende Protokolle verwendet:

Schlüsselaustausch nach Diffie-Hellman

Dieses Schlüsselaustausch-Protokoll ist das bekannteste Protokoll. Es eignet sich für die Erzeugung und Verteilung von Schlüsseln, jedoch nicht zur Ver- und Entschlüsselung. Es beruht auf dem Problem des diskreten Algorithmus.

Beide Partner eignen sich auf eine geeignete Gruppe, in der das DL-Problem rechentechnisch nicht beherrschbar ist. Außerdem wird ein festes Gruppenelement g gewählt, das wie die Gruppe selber öffentlich bekannt sein darf:

- Partner A wählt eine zufällige Zahl x, berechnet gx und sendet dies an Partner B
- Partner B wählt eine zufällige Zahl y, berechnet gy und sendet dies an Partner A
- Partner A bildet (gy)x = gyx
- Partner B bildet (gx)y = gxy

Dann ist g^{yx} der gemeinsame, geheime Schlüssel für ein symmetrisches Verfahren.

Dieses Protokoll zählt zu den asymmetrischen Verfahren, da man die Werte g^x und g^y als die öffentlichen, sowie x und y als die zugehörigen privaten Schlüssel der beiden Partner ansehen kann.

Die Sicherheit von Diffie-Hellman hängt von der Sicherheit des DL-Problems[9] in der gewählten Gruppe ab. Bei Implementierungen in Z_P sollte p groß genug gewählt werden (mindestens 128 Bit).

EC-DH

EC-DH ist die Entsprechung des Protokolls für den Schlüsselaustausch von Diffie-Hellman auf elliptischen Kurven.

MQV

Das Menezes-Qu-Vanstone Protokoll für den Schlüsselaustausch ist eine Erweiterung des Diffie-Hellman Protokolls. Das Problem bei Diffie-Hellman besteht darin, dass in jeder Sitzung identische Schlüssel erzeugt werden. Zur Lösung dieses Problems und zur Absicherung gegen

[9] **Diskreter Logarithmus (DL) siehe Mathematische Grundlagen der Kryptographie**

Angriffsmöglichkeiten wie die Man-in-the-Middle-Attack 10 wurden Protokolle wie MQV entwickelt. Vorteil ist die implizierte Authentifizierung der Kommunikationspartner. Außerdem fließen in die Berechnung von Sitzungsschlüsseln Zufallselemente ein, die gewährleisten, dass bei verschiedenen Sitzungen auch verschiedene Schlüssel verwendet werden.

EC-MQV

EC-MQV wird im IEEE-Standard beschrieben. Das ist die Variante des Menezes-Qu-Vanstone Protokolls, bei der man auf elliptischen Kurven arbeitet.

6.2 Hashfunktionen

Im Allgemeinen kann man nicht davon ausgehen, dass die Verschlüsselung einer Nachricht automatisch deren Authentizität oder Integrität garantiert.

Hashfunktionen werden häufig für die Überprüfung der Integrität von Daten und MACs für die Sicherstellung der Authentizität eingesetzt.

Eine Hashfunktion komprimiert Daten in einer bestimmten, praktisch schwer umkehrbaren Art. Zur exakten Definition braucht man den Begriff der Einwegfunktion. Als Einwegfunktion wird eine Funktion f bezeichnet, die schnell berechenbar ist, für die es aber praktisch nicht möglich ist, zu einem gegebenen Funktionswert y ein Argument x mit der Eigenschaft f(x)=y zu bestimmen.

Eine kryptographische Hashfunktion ist ein Algorithmus, der zu einer Nachricht einen nicht manipulierbaren Prüfwert („Fingerabdruck“) fester Länge erzeugt. Der einzige Parameter der Hashfunktion ist die Nachricht selber, d.h. es geht kein Schlüssel ein. Ein Beispiel ist die elektronische Unterschrift, bei der anstelle der gesamten Daten nur deren Hashwert dem Signaturprozess unterliegt.

Heute sollte ein Hashwert von mindestens 160 Bit Länge verwendet werden.

Die Anforderungen an eine Hashfunktion sind:

1. Der Hashwert h(M) ist bei gegebenen M „leicht“ zu berechnen.
2. Es ist praktisch nicht möglich, zwei beliebige Nachrichten M und M’ zu finden, deren Hashwerte h(M) und h(M’) übereinstimmen.

Eine Hashfunktion mit diesen Eigenschaften wird als kollisionsfrei oder kollisionsresistent bezeichnet.

Angriffsmöglichkeiten auf Hashfunktionen sind:

1. Zu gegebenem Hashwert sucht man eine Nachricht mit identischem Hashwert.
2. Man sucht zwei Nachrichten mit identischem Hashwert.

Beispielsweise sind bei einem Angriff auf Hashfunktionen mit 160 Bit Hashwertlänge ca. 2^{80} Operationen notwendig, um Kollisionen zu erzeugen.

---

[10] Fortführung des Server-Spoofings. Bei der Man in the Middle Attack steht der Angreifer zwischen zwei miteinander kommunizierenden Rechnern. Von hier aus kann er alle Daten abfangen und verändert (oder gar nicht) an den Adressaten weitergeben, der glaubt, sie vom ursprünglichen Absender zu erhalten.

6.2.1 Konstruktionsprinzipien und Algorithmen

Es gibt verschiedene Vorgehensweisen zur Konstruktion einer Hashfunktion. Oft wird folgender verwendet:

Die Nachricht wird in eine Sequenz gleich großer Blöcke passender Länge aufgeteilt. Bei Bedarf wird der letzte Block mit einem vorgegebenen Muster aufgefüllt (das so genannte Padding).

Der Algorithmus wird mit einem bestimmten, festen Initialisierungswert IV (Initial Value) gestartet.

Die Nachrichtenblöcke werden nacheinander verarbeitet, indem sie als Input für eine Kompressionsfunktion benutzt werden. Der jeweilige Output dieser Funktion ist dann der Initialisierungswert für die nächste Anwendung.

Der Hashwert ist der letzte Output der Kompressionsfunktion.

Die Kompressionsfunktion wird mehrmals angewendet.

Damit ergibt sich eine Klasse von Hashfunktionen, die für 32-Bit-Architekturen entwickelt wurden. Der Vorreiter war MD4, dem MD5, RIPEMD-128, RIPEMD-160, SHA-0 und SHA-1 folgten.

Bei einer 128-Bit-Ausgabe sind ca. 2^{64} Operationen erforderlich, um zwei verschiedene Nachrichten mit einem Hashwert zu konstruieren, bzw. 2^{128}-1 Operationen, um zu einem gegeben Hashwert eine passende Nachricht, d.h. eine mit dem gleichen Hashwert zu bestimmen.

6.2.2 Algorithmen

MD2

Ein von Ron Divest 1989 entwickelte Hashfunktion. MD steht für Message Digest und wurde für eine 8-Bit-Architektur optimiert.

MD4

MD4 wurde 1990 von R.L. Rivest entwickelt. MD4 ist der Vorläufer vieler Hashfunktionen. Von der Anwendung von MD4 ist abzuraten, da potenzielle Angriffsmöglichkeiten bekannt sind und auch schon zum Erfolg geführt haben.

MD5

Die Weiterentwicklung von MD4 wurde von R.L. Rivest 1991 vorgestellt und u.a. in PGP verwendet.

RIPEMD-128

RIPEMD hat ein ähnliches Design wie MD4. Für RIPEMD-128 kann man Kollisionsfunktionen finden und die Geburtstagsattacke ist realisierbar, wenn auch mit hohen Kosten. Deshalb ist von der Anwendung abzuraten.

RIPEMD-160

RIPEMD-160 ist die Weiterentwicklung von RIPEMD-128 mit einem Hashwert von 160 Bit. Dafür sind keine Schwächen bekannt und diese Hashfunktion wird allgemein empfohlen.

SHA-0

Der Secure Hash Algorithm (SHA-0) wurde von der National Security Agency (NSA) entworfen. Entwickelt wurde SHA-0 als Hashalgorithmus für den Signaturalgorithmus DSA.

SHA-1

Weiterentwicklung von SHA-0 und für den Einsatz empfohlen, da keine Schwächen bekannt.

Die folgende Übersicht ist die Zusammenfassung von Hashfunktion, Bitlänge und Anzahl der Runden innerhalb des Algorithmus.

Name	*Bitlänge*	*Runden x Schritte pro Runde*
MD4	128	3 x 16
MD5	128	4 x 16
RIPEMD-128	128	4 x 16 zweimal (parallel)
RIPEMD-160	160	5 x 16 zweimal (parallel)
SHA-0	160	4 x 20
SHA-1	160	4 x20

6.3 Authentifizierungscodes (MAC)

Für die elektronische Authentifizierung von Daten wird eine Nachricht M um spezielle Informationen erweitert, die mittels eines kryptographischen Verfahrens aus M berechnet und zusammen mit der Nachricht gespeichert bzw. übertragen werden. Damit ein Angreifer die angefügte Redundanz nicht gezielt modifizieren kann, muss diese geschützt werden.

Erfolgt die Ermittlung der redundanten Informationen unter Verwendung eines geheimen Schlüssels, so spricht man von einem Message Authentication Code (MAC). Deshalb bezeichnet man MAC auch als Hashfunktion mit einem zusätzlichen geheimen Schlüssel.

Der Nachweis der Integrität einer Nachricht basiert auf der Geheimhaltung bzw. der Integrität des kryptographischen Schlüssels. Der einfachste Fall besteht in der symmetrischen Verschlüsselung des Hashwertes. Der Empfänger muss den Schlüssel kennen und kann mit diesem Schlüssel aber auch andere Nachrichten mit demselben Hashwert erzeugen.

Ein MAC lässt sich aus einer Hashfunktion oder einer symmetrischen Blockchiffre erzeugen.

Es gibt folgende Verfahren für MACs:

MAC basierend auf Hashfunktionen – Der MAC wird erzeugt, indem man Schlüssel, Nachricht und wieder den Schlüssel zusammenfasst und diese der Hashfunktion als Eingabe gibt. Die Sicherheit dieses Verfahrens hängt von der Geheimhaltung des Schlüssels und vom verwendeten Hashverfahren ab.

CBC-MAC
Die Konstruktion dieses MAC erfolgt durch eine sehr einfache Möglichkeit, der Verschlüsselung einer Nachricht mit einem Blockalgorithmus im CBS-Modus.

Der letzte Chiffreblock wird als Tag verwendet. Die Sicherheit des CBS-MAC hängt von der Geheimhaltung des Schlüssels und vom verwendeten symmetrischen Verfahren ab.

6.4 Digitale Signaturen

Die Signierung eines in digitaler Form vorliegenden Dokumentes ist das Äquivalent zu einer eigenhändigen Unterschrift eines auf Papier vorliegenden Dokumentes.

Die Probleme bestehen hauptsächlich aus folgenden Fragestellungen:

1. Ist das Dokument nach Fertigstellung noch verändert worden?
2. Stammt das Dokument vom vorgegebenen Absender?

Die rechtliche Seite ist zu beachten. In Deutschland wurden 1997 das Signaturgesetz und eine ergänzende Signaturverordnung verabschiedet, die unter anderem die rechtlichen Rahmenbedingungen für fälschungssichere digitale Signaturen und für die Wahrung der Rechte der Teilnehmer am elektronischen Rechtsverkehr schaffen.

Auf EU-Ebene gibt es auch Ansätze für Regelungen zu elektronischen Signaturen.

Zum Unterzeichnen von Dokumenten können Public-Key Verfahren verwendet werden. Bei einem asymmetrischen Verschlüsselungsverfahren wird normalerweise mit dem öffentlichen Schlüssel chiffriert und mit dem privaten Schlüssel dechiffriert. Beim digitalen Signieren wird der Gebrauch der Schlüssel umgedreht:

Partner A chiffriert die Daten mit seinem privaten Schlüssel und sendet das unterzeichnete Dokument an Partner B.

Partner B dechiffriert das Dokument mit dem öffentlichen Schlüssel des Partners A, wodurch er die Echtheit der Unterschrift prüft.

Bei großen Dokumenten gibt es jedoch das Problem, das das Verschlüsseln mit einem asymmetrischen Verfahren einige Zeit in Anspruch nimmt. Deshalb wird in der Praxis nicht das ganze Dokument verschlüsselt, sondern nur der Hashwert. Das ist eine ähnliche Situation wie bei den hybriden Verfahren.

Die digitale Signatur funktioniert nun so:

Partner A chiffriert den Hashwert seines Dokumentes mit seinem privaten Schlüssel, womit er das Dokument unterzeichnet.

Partner A sendet das Dokument und den signierten Hashwert an Partner B.

Partner B berechnet den Hashwert des von Partner A gesendeten Dokumentes mit derselben Hashfunktion. Mit dem öffentlichen Schlüssel von Partner A und dem Algorithmus für elektronische Unterschriften dechiffriert er den signierten Hashwert. Stimmt dieser mit dem von ihm generierten Hashwert überein, kann er Vertrauen in die Echtheit des Dokumentes gewinnen.

Damit ist das Problem der Verfälschung, da die Änderungen erkennbar werden, mit dem Anwenden von Hashfunktionen gelöst.

Die Identität einer Person ist durch seinen öffentlichen Schlüssel gewährleistet, falls er durch eine Zertifizierungsinstanz bestätigt ist.

6.4.1 Signaturverfahren

DSA

DSA ist eine Variante der Unterschriftenalgorithmen von Schnorr und ElGamal. DSA basiert auf dem Problem des Diskreten Algorithmus in endlichen Körpern.

DSA benutzt folgende Parameter:

- p Primzahl mit 512 bis 1024 Bit Länge (in 64-Bit-Schritten)
- q Primzahl, 160 Bit langer Faktor von p-1
- g erzeugendes Element der Ordnung q
- x privater Schlüssel, beliebige Zufallszahl kleiner q
- $y = g^x \bmod p$ öffentlicher Schlüssel

Die ersten drei Parameter p, q, g und natürlich der öffentliche Schlüssel y sind öffentlich bekannt und der private Schlüssel x muss geheim gehalten werden.

Weiter sei

- m Nachricht
- H Hashfunktion

Zum Erstellen und Verifizieren von Signaturen beim DSA dienen die wesentlichen Gleichungen:

(1) $r = (g^k \bmod p) \bmod q$ k zufällig gewählt

(2) $sk = H(m) + xr \pmod q$

Die Idee ist, dass der Inhaber der Signatur r und s bestimmen kann. Der Empfänger verifiziert die Signatur, indem er prüft, ob (2) gilt. Die Zufallszahl k ist als temporärer, geheim zu haltender Schlüssel anzusehen und nur mit Kenntnis dieses Geheimnisses ist (2) nach s auflösbar, was für die Erstellung einer korrekten Signatur notwendig ist.

Mit 512 Bit ist DSA nicht stark genug, um Sicherheit zu bieten. Dies ist erst mit 1024 Bit der Fall. Die Sicherheit entspricht der von RSA mit vergleichbaren Parametern. Die Sicherheit von DSA beruht auf zwei verschiedenen, aber verwandten Problemen: Einerseits auf dem allgemeinen DL-Problem in Zp (für welche ähnlich wie beim Faktorisierungsproblem subexponenzielle Angriffsmethoden existieren). Anderseits beruht die Sicherheit von DSA auf dem DL-Problem in der von g erzeugten Untergruppe mit Ordnung q. Dafür benötigen die besten bekannten Angriffe Operationen in der Größenordnung von $\sqrt{q}$.

RSA

Man kann RSA nicht nur als Verschlüsselungsalgorithmus einsetzen, sondern auch zum digitalen Signieren. Der Algorithmus hat eine analoge Form wie bei der Verschlüsselung (siehe auch Asymmetrische Verschlüsselungsalgorithmen). Es sind jedoch die Rollen vertauscht: Mit dem eigenen privaten Schlüssel wird signiert und der Empfänger überprüft die Unterschrift mit dem zugehörigen öffentlichen Schlüssel.

Dabei wird wie allgemein üblich nicht die gesamte Nachricht unterschrieben, sondern nur der Hashwert. Es muss beachtet werden, dass nach Anwenden einer Hashfunktion der zu signierende Wert eine geeignete Länge haben muss. Zum Auffüllen des Hashwertes auf die geforderte Länge verwendet man so genannte Padding-Verfahren.

EC-DSA

EC-DSA ist die Entsprechung von DSA auf elliptischen Kurven. Anstatt auf einer Untergruppe der Ordnung q von Zp zu arbeiten, ist die zugrunde liegende mathematische Struktur eine elliptische Kurve E.

Die Sicherheit dieser kryptographischen Systeme basiert auf der Schwierigkeit des DL-Problems in Gruppen von Punkten auf einer elliptischen Kurve (EC-DLP).

NR

NR ist ein Signaturverfahren, das auf dem DL-Problem basiert. Der ursprüngliche Algorithmus erlaubt das so genannte message recovery (Nachrichten-Wiederherstellung). Die in die Standards eingegangenen Formulierungen dieses Protokolls machen von dieser prinzipiellen Möglichkeit allerdings keinen Gebrauch.

Der Algorithmus wurde von K. Nyberg und R. Rueppel entwickelt, was die Abkürzung NR erklärt. Es handelt sich um eine Variante des ElGamal-Signaturschemas.

EC-NR

Ähnlich wie im Fall des DSA lässt sich das originale Protokoll von Nyberg und Rueppel fast wörtlich auf den Fall der Verwendung von elliptischen Kurven übertragen. Überschrift 1

Über GBS

GROUP Business Software ist führender Anbieter von Lösungen und Services in den Bereichen Messaging Security und Workflow für die IBM und Microsoft Collaboration Plattformen. Weltweit vertrauen mehr als 5.000 Kunden und 4 Millionen Anwender auf die Expertise von GBS. Der Konzern ist in Europa, Nordamerika sowie Asien tätig.

Weitere Informationen unter www.gbs.com