

Bit- und Byte-Technik in der Kryptographie

(Ernst Erich Schnoor)

Kryptographie ist Schreiben und Lesen von geheimen Informationen mit arteigenen Methoden (Verschlüsselung). Seit Jahrtausenden haben Menschen die Zeichen ihrer Sprache so verändert, das nur der berechtigte Empfänger in der Lage war, die Information zu entziffern. Dabei bedienten sie sich fast ausschließlich ganzer Zeichen ihres jeweiligen Alphabets. Zwei Methoden haben sich mit der Zeit entwickelt: "Substitution" und "Transposition".(#1). Bei der Substitution wird das Zeichen an gleicher Stelle durch ein anderes Zeichen ersetzt. Die Transposition bringt das Zeichen an eine andere Stelle der verschlüsselten Nachricht. In beiden Fällen jedoch bleibt das **Element** der Information gleich: das veränderbare **Zeichen**.

Als die Computer kamen, hat sich diese Betrachtung geändert. Computer arbeiten im Zahlensystem zur Basis 2 (Binärtechnik). Als Folge der elektronischen Technik können sie nur zwei Zustände unterscheiden: "vorhanden" (eine Spannung) oder "nicht vorhanden" (keine Spannung), d.h. in Ziffern des Zahlensystems zur Basis 2: "**eins**" oder "**null**". Dieser Zustand wird allgemein als "**Bit**" definiert. Stellt ein Vergleich keinen Unterschied fest, dann wird entweder "null" mit "null" oder "eins" mit "eins" verglichen und der Unterschied ist "null", anderenfalls "eins" (XOR-Verknüpfung). Das Bit stellt somit nur einen Zustand dar und kann daher nicht Träger einer Information sein. Es kann höchstens als „**flag**“ dienen.

Eine Information ist mehrschichtig, sie besteht aus mindestens zwei Bits. Allgemein: eine Folge (**m**) bestimmter Bytes (**a_i**) mit der Länge (**n**). Um die Folge als Sachverhalt zu analysieren, muss sie systematisiert (skaliert) werden. Dazu wird jedem Byte **a** ein Index (**i**) zugeordnet und alle **n** Bytes werden in sachgerechter Weise miteinander verknüpft. Als Index bietet sich das erweiterte ASCII-System an.

$$m = a_1 + a_2 + a_3 + \dots + a_i + \dots + a_n$$

(Der einzelne Wert für "a_i" wird um (+1) erhöht da sonst ASCII-null (0) nicht berücksichtigt wird)

$$m = \sum_{i=1}^n (a_i + 1)$$

Um die einzelnen Bytes **a(i)** innerhalb der Folge zu unterscheiden, müssen weitere Merkmale hinzukommen, da anderenfalls keine eindeutigen Ergebnisse erzielt werden.

Mit Besinnung auf **Renè Descartes** (1596 – 1650) wissen wir, dass jeder Sachverhalt - soweit er in seinen Dimensionen skalierbar ist – durch seine Koordinaten für **Gegenstand**, **Ort** und **Zeit** (kartesisches Koordinatensystem) eindeutig bestimmt werden kann (#2). Wir definieren:

Sachverhalt = (m) digitale Information der Länge (n)
 Gegenstand = a(i) Element der Information, Zeichen, Byte
 Ort = p(i) Position von a(i) innerhalb der Information
 Zeit = t(i) Zeitpunkt von a(i) innerhalb der Information

Damit sich die einzelnen Zeichen unterscheiden wird jedes Byte mit seinem Ort p(i) multipliziert, d.h. **positionsgewichtet**. Die Zeit ist allerdings nur dann von Bedeutung, wenn zwischen den einzelnen Bytes und der Prozessorfrequenz eine variable Funktion besteht. Normalerweise ist die Verbindung jedoch konstant und wir können t = 1 setzen. Um einen Wert für die Folge m zu erhalten, verknüpfen wir die Dimensionswerte für Gegenstand, Ort und Zeit durch Multiplikation und addieren jedes Ergebnis zu einem eindeutigen Bestimmungswert H(k):

$$H(k) = \sum_{i=1}^n (a_i + 1) * p_i * t_i \quad t_i = 1$$

Weitere Einzelheiten zur Systematisierung der digitalen Information werden im Artikel ["Kryptographische Basisfunktion in Byte-Technik"](#) erläutert.

Die vom Computer generierten Ausgaben als Bitfolgen sind technisch unbegrenzt. Um mit Informationen systematisch zu arbeiten, werden die Bitfolgen skaliert und in bestimmte Abschnitte (Bytes) geteilt. Mit ihrer Systematisierung entsteht dann eine „**Welt der Bytes**“.

Angeregt durch die Binärtechnik haben die zur Weiterentwicklung der Kryptographie berufenen Wissenschaftler sich vordergründig auf die "**Bits**" gestürzt und dabei der Systematik der "**Bytes**" nicht die ihr im Grunde zukommende Bedeutung beigemessen. "**Coding base 64**" und **Ron Rivest's RC4** scheinen noch die wenigen Anwendungen zu sein, die sich im Prinzip auf Bytes gründen.

So wie der Zahlenstrahl in verschiedene Zahlensysteme (von zur Basis 2 bis zur Basis 256) aufgeteilt wird, können auch Bitfolgen in bestimmte Segmente – **Bytes** - geteilt und dann systematisch geordnet werden. Abschnitte mit 4 Bits beispielsweise sind ein "nibble" und Segmente mit 6 Bits ergeben ein Alphabet von 6x6 Zeichen, wie bereits im Verfahren "coding base 64" (#3) realisiert. Fast alle anderen in der Kryptographie entwickelten Verfahren verwenden Abschnitte von 8 Bits. In der Informatik gilt der Grundsatz: **1 Byte = 8 Bits**.

Diese Definition ist offensichtlich einseitig und zu eng. Beispielsweise im Vergleich mit der **Zahlentheorie** würde das bedeuten: **Zahl = 10 Ziffern**. Dies gilt natürlich nur im Zahlensystem zur Basis 10. Im Hezadezimalsystem umfasst eine Zahl bekanntlich 16 Ziffern und im Zahlensystem zur Basis 62 insgesamt 62 Ziffern. Die gebräuchliche Definition eines Bytes umfasst nur einen Teil der Möglichkeiten. Alle uniformen Bit-Sequenzen können als eigene Einheiten definiert werden: neben 8-bit Sequenzen beispielsweise auch 5-bit, 7-bit, 9-bit, 10-bit und 12-bit Einheiten.

Dass heute überwiegend 8-bit Sequenzen verwendet werden, hat natürlich ihre Ursache in der Technik der Computer und der Anwendung des klassischen Alphabets und seiner Erweiterung auf 256 Zeichen. Bei genauer Betrachtung der Bitfolgen und ihrer Aufteilung in Abschnitte ergibt ein Strukturvergleich mit Begriffen der **Zahlentheorie** folgendes:

Ziffern	-->	Bits
Zahlen, Elemente	-->	Abschnitt, Zeichen, Bytes
Strecke	-->	Segment, Block
geordnete Mengen	-->	Arrays, Zeichenvorräte, Alphabete
Betrag	-->	Index, Wert

Das Bit entspricht der Ziffer, das Byte der Zahl, die geordnete Menge dem Zeichenvorrat und Alphabet. Es erscheint daher sinnvoll, in Anlehnung an die Zahlentheorie entsprechende Begriffe auch in der Kryptographie zu verwenden. Dem Begriff "Zahlensystem zur Basis .." kann der Begriff "Bytesystem zur Basis .." gegenüber gestellt werden: z.B. "**Bytesystem zur Basis 8**". Aus dieser Betrachtung ergeben sich dann folgende Entsprechungen:

Bitfolgen von 2-bit	=	Bytesystem zur Basis 2	=	2^2 Bytes	=	4 Bytes
7-bit	=	Bytesystem zur Basis 7	=	2^7 Bytes	=	128 Bytes
8-bit	=	Bytesystem zur Basis 8	=	2^8 Bytes	=	256 Bytes
10-bit	=	Bytesystem zur Basis 10	=	2^{10} Bytes	=	1024 Bytes
12-bit	=	Bytesystem zur Basis 12	=	2^{12} Bytes	=	4096 Bytes
16-bit	=	Bytesystem zur Basis 16	=	2^{16} Bytes	=	65536 Bytes
32-bit	=	Bytesystem zur Basis 32	=	2^{32} Bytes	=	4294967296 Bytes

Die Menge der Bytes eines Bytesystems bilden jeweils das zugehörige Alphabet. Das Alphabet zur Basis 6 umfasst dementsprechend 64 Bytes (Zeichen), das Alphabet zur Basis 7 hat 128 Zeichen, das Alphabet zur Basis 8 hat 256 Zeichen und das Alphabet zur Basis 12 insgesamt 4096 Zeichen.

Das kleinste Bytesystem zur Basis 2 umfasst 4 Elemente (Bytes): 00, 01, 10, 11. Mit dem System kann beispielsweise eine Information über Bewegungsabläufe definiert werden: **11** = geradeaus, **01** nach rechts, **10** nach links und **00** keine Bewegung. Dieses Beispiel zeigt, dass ein Bit noch keine Information tragen kann, es müssen mindestens zwei Bit sein.

Die einzelnen Alphabete sind völlig unabhängig voneinander, Schwierigkeiten bereitet es nur, eigene Symbole (Zeichen) für die jeweiligen Bytes zu definieren. In der heutigen Praxis wird fast ausschließlich der erweiterte ASCII-Zeichensatz verwendet: Alphabet des Bytesystems zur Basis 8. Für das Bytesystem zur Basis 16 gibt es den Unicode (#4). Für alle anderen Basisysteme müssen in Ermangelung eigener Zeichen die erforderlichen Darstellungen ersatzweise aus dem ASCII-Code abgeleitet werden. Für höhere Bytesysteme (z.B. 12-bit) kann ein Zeichen auch aus zwei Zeichen niedriger Bytesysteme zusammengesetzt werden. Bei dieser Vielfalt von Möglichkeiten erscheint es sinnvoll, sie in einer „**Bytetheorie**“ zu systematisieren.

Wie in der Zahlentheorie können auch in der "Bytetheorie" Umwandlungen von einem Bytesystem in ein anderes System vorgenommen werden, es müssen dabei nur die Anzahl der Bits gleich bleiben. Im Falle einfacher Umwandlung beispielsweise ergeben **63 Bytes zur Basis 8** (504 Bits) **72 Bytes im System zur Basis 7** (504 Bits). Bei doppelter Umwandlung werden aus 84 Bytes zur Basis 8 (672 Bits) insgesamt 56 Bytes zur Basis 12 (672 Bits), die ihrerseits in 96 Bytes zur Basis 7 (672 Bits) zurück gewandelt werden.

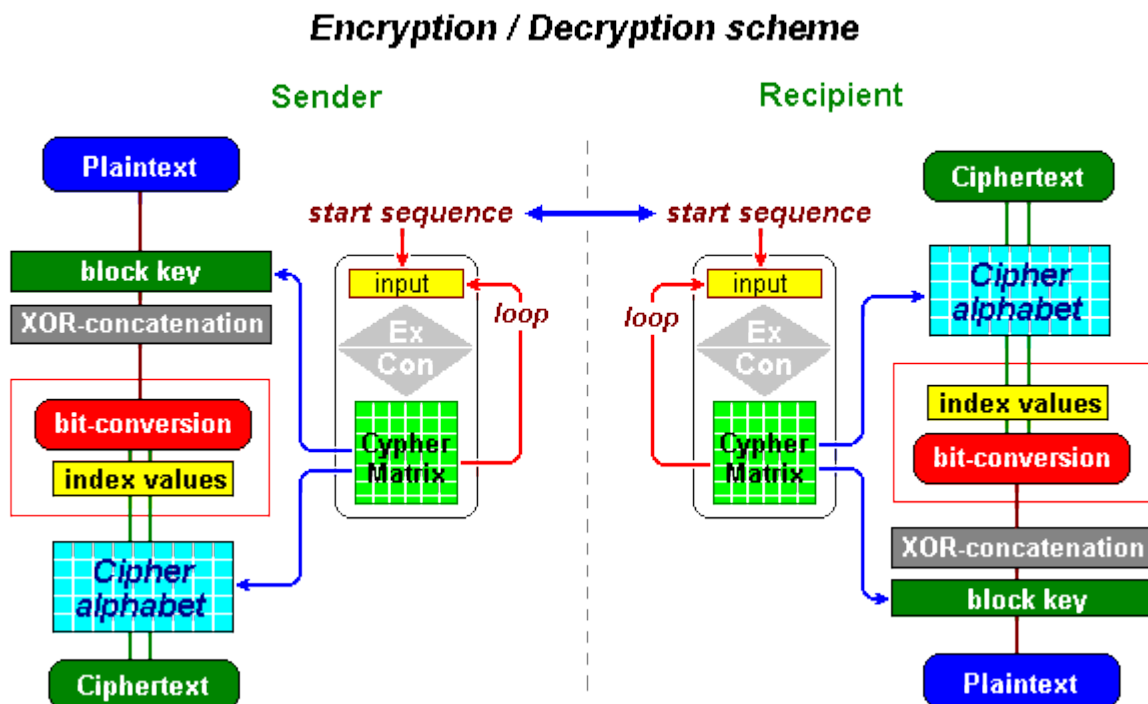
Auf dieser Basis lassen sich eine Vielzahl von Verschlüsselungsverfahren entwickeln. Zum Beispiel: Umwandlungen vom einem Klartext-Alphabet mit 64 Zeichen (Bytesystem zur Basis 6) in das Bytesystem zur Basis 5 mit einem Alphabet von 36 Zeichen.

Ein weiteres Beispiel: Im Bytesystem zur Basis 8 werden Klartextblöcke von 63 Bytes Länge zunächst mit gleich langen Bytesequenzen (Block-Schlüssel) XOR-verknüpft. Anschließend werden die resultierenden 8-bit Sequenzen in 7-bit Abschnitte umgewandelt (**bit-conversion**), die ihrerseits als Indizes 72 Chiffre-Zeichen aus einem eigenständigen Alphabet von 128 Zeichen (cipher alphabet im Bytesystem zur Basis 7) holen und zum Geheimtext verbinden .

8-Bit XOR-Sequenzen umgruppiert in 7-Bit-Abschnitte (Bytes im Bytesystem zur Basis 7):

8-Bit: **111011110000111100110101010111110101001110111101000100**
 7-Bit: **111011110000111100110101010111110101001110111101000100**

Die umgruppierten 7-Bit Segmente sind nur Indexwerte (Zeiger) auf Positionen im Array **Chiffre-Alphabet** von 128 Zeichen. Das folgende Schema zeigt das Prinzip:



Es bleibt nur das Problem, woher Schlüsselsequenzen (block key) und eigenständige Alphabete (cipher alphabet) nehmen?

Dieses Problem wird mit einer Basisfunktion gelöst, vom Autor "**CypherMatrix**" Methode genannt. Eine Startsequenz mit optimal 42 Bytes initialisiert die Funktion, diese Eingabe wird in ein höheres Zahlensystem expandiert (z.B. Basis 77), dann wieder mit MODULO 256 zurückgeführt auf Elemente des Bytesystems zur Basis 8 ($2^8 = 256$ Zeichen) und in der CypherMatrix mit 16x16 Zeichen abgebildet. Gesteuert von der Startsequenz werden die zur Verschlüsselung benötigten Schlüsselsequenzen und eigenständigen Alphabete aus der CypherMatrix entnommen. Im Prinzip kann die Basisfunktion auch auf jedes andere Bytesystem angewendet werden.

Verschlüsselungen bestehen fast immer aus "Substitutionen". Das zeigt sich schon darin, dass allen Anwendungen das Bytesystem zur Basis 8 zugrunde liegt, sowohl beim Klartext als auch beim Chiffretext. Es wird ausdrücklich gefordert, dass Klartext und Chiffretext gleich lang sein sollen. Das bedeutet, dass für jedes Klartextzeichen ein bestimmtes Chiffretextzeichen vorhanden ist und dass wegen der gleichen Länge des Chiffretextes das Zeichen auch an der gleichen Position steht wie im Klartext: Kriterium der "Substitution".

Das Arbeiten im Bytesystem zur Basis 8 hat jedoch eine Einschränkung zur Folge: Ein wesentlicher Teil der klassischen Kryptographie, die "Transpositionen", werden nur schwer (oder gar nicht) realisiert, denn dazu wäre die Übertragung der Information von einem Bytesystem in ein anderes erforderlich. Zum Beispiel durch Umwandlungen von 8-bit Sequenzen in 7-bit Sequenzen (**bit-conversion**). Allerdings verlängert sich der Chiffretext dann auch von 7 auf 8 Längeneinheiten. Auf jedes Klartextzeichen kommen $8/7 = 1,143$ Chiffretextzeichen. Gleiche Positionen von Klar- und Chiffretext kann es dabei nicht geben. Lösungen durch Wechsel im Bytesystem bieten sich auch in vielen anderen Situationen an.

Mehr Einzelheiten über Bytes in kryptographischen Lösungen finden sich im Internet unter:

www.telecypher.net

Von dort können auch verschiedene DEMO-Programme geladen und selbst getestet werden.

- (#1) Bauer, Friedrich L., Entzifferte Geheimnisse, Berlin Heidelberg New York, 1995 Sn. 36 und 78
- (#2) Abstraktionsebene für jeden skalierbaren Sachverhalt, abgeleitet aus dem kartesischen Koordinatensystem (nach René Descartes).
- (#3) Coding Base 64
- (#4) Unicode

München, im Januar 2010
Ernst Erich Schnoor
