Vom Bit zum CBC

Crashkurs - Digitale Verschlüsselungstechnik

ein Vortrag von Marius Schwarz

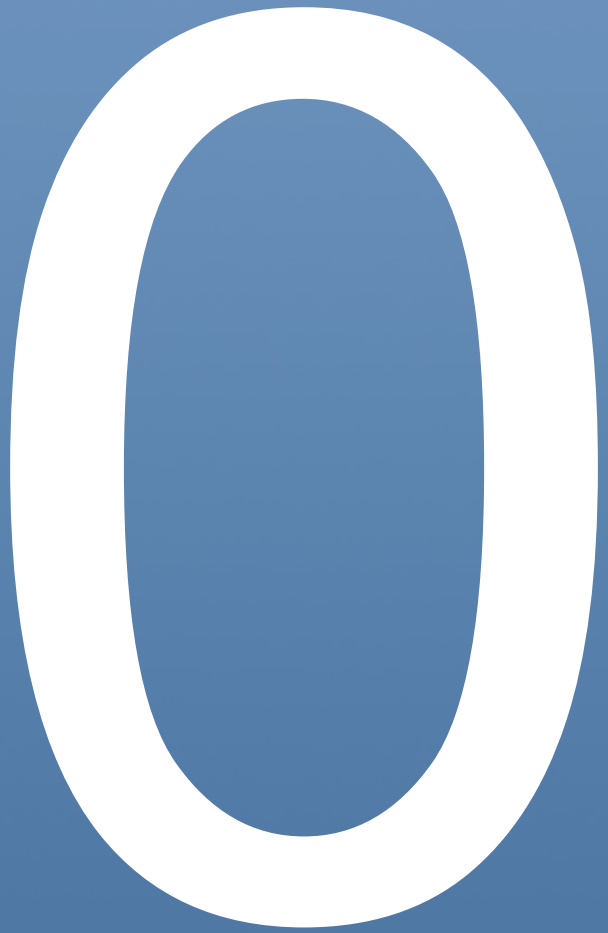
im Rahmen des KP 2018

Vom Bit zum CBC

0

Vom Bit zum CBC

Am Anfang war die Null ...



Vom Bit zum CBC

1

… dann kam die … 1

Vom Bit zum CBC

Damit war das „Bit“ geboren.

Ein *Bit* definiert sich als die kleinste Informationseinheit.

Vom Bit zum CBC

Analoges Beispiel:

Der Zustand eines Lichtschalters ist „an“ oder „aus“ .

Vom Bit zum CBC

Digitale Version:

Der Zustand eines Lichtschalters ist „1“ oder „0“ .

Vom Bit zum CBC

Aus 8 Bits wird 1 Byte

Vom Bit zum CBC

Aus 8 Bits wird 1 Byte

01010101

Vom Bit zum CBC

Aus 8 Bits wird 1 Byte

01010101

Jede Stelle eines Bytes,
hat eine andere Wertigkeit

Vom Bit zum CBC

Was ist eine Wertigkeit ?

die dezimale Zahl 203 besteht aus 3 Ziffern
und bedeutet eigentlich :

2 * 100 + 0 * 10 + 3 * 1

oder anders ausgedrückt

2 * 10^2 + 0 * 10^1 + 3 * 10^0

Vom Bit zum CBC

Warum ?

Weil wir zehn Ziffern haben :

0,1,2,3,4,5,6,7,8,9

Jede Stelle also auf der Anzahl der Möglichkeiten basiert => 10

Deswegen auch der Name Dezimalsystem

Vom Bit zum CBC

Dezimalsystem

„Jede Stelle einer Zahl, von Rechts nach Links, verzehnfacht den Wert der Stelle!“

Wir haben es mit einer Potenzreihe zu tun:

…. 10^3 + 10^2 + 10^1 + 10^0

allgemein : { Xn * 10^n } (+n → -n)

(mit einem Komma nach n=0 ;))

Vom Bit zum CBC

Fragen ?

Vom Bit zum CBC

Binärsystem

„Jede Stelle einer Zahl, von Rechts nach Links, verdoppelt den Wert der Stelle!“

Wir haben es mit einer Potenzreihe zu tun:

…. 2^3 + 2^2 + 2^1 + 2^0

allgemein : { Xn * 2^n } (n→0)

mit X immer „1“ oder „0“.

Vom Bit zum CBC

Binärsystem

2^0 = 1
2^1 = 2
2^2 = 4
2^3 = 8
2^4 = 16
2^5 = 32
2^6 = 64
2^7 = 128
2^8 = 256

Vom Bit zum CBC

Binärsystem

Beispiel:

76543210
01010101

$0*2^7+1*2^6+0*2^5+1*2^4+0*2^3+1*2^2+0*2^1+1*2^0$

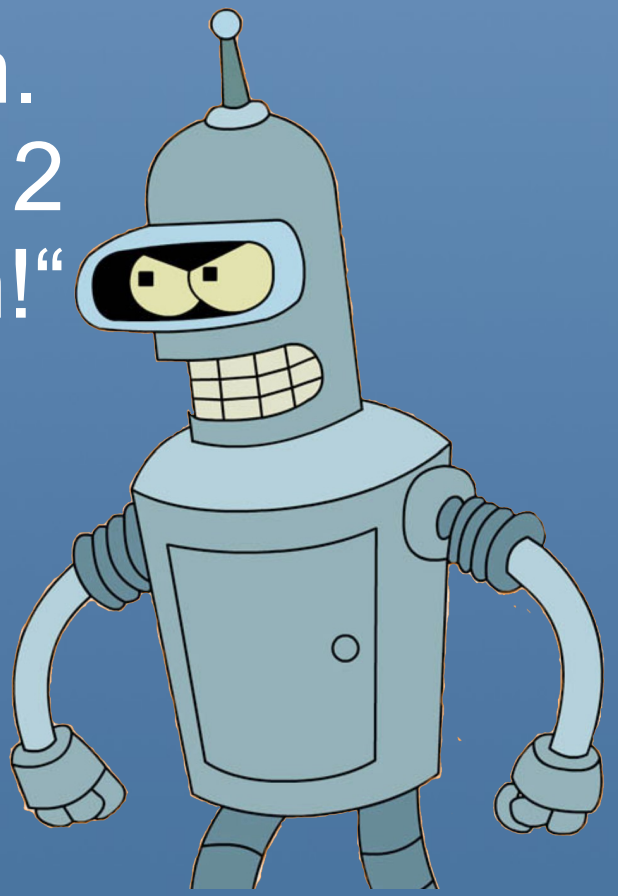
Dezimal : 64 + 16 + 4 + 1 => 85

Vom Bit zum CBC

Binärsystem

Merke:

„Ich hatte einen Albtraum.
Ich denke, ich habe eine 2
gesehen!“



Vom Bit zum CBC – Zeichensatz

„American Standard Code for Information Interchange“
„DIN Norm Zeichensatz de-latin1“
„UTF-8“, „UTF-16“, „UTF-32“

sind in Kurzform Zeichensätze:

Wie man Buchstaben und Zeichen in Zahlen umwandelt!

Auflösung: Jedem Buchstaben, der Ziffer und jedem Sonderzeichen, daß man haben will wird eine Zahl zugeordnet.

Man beachte den sprachlichen Unterschied Ziffer ↔ Zahl

Vom Bit zum CBC – Zeichensatz

„Eine Zahl besteht aus Ziffern.“

„Eine Ziffer ist keine Zahl, sondern ein Symbol.“

„Ein Zeichensatz wandelt Symbole in Zahlen um.“

Beispiel:

„A“ => 65 „B“ = 66, „C“ = 67 usw.

Der Vater aller Zeichensätze heißt ASCII.

Vom Bit zum CBC – Zeichensatz

„Die Zeichenkodierung ASCII definiert 128 Zeichen, bestehend aus 33 nicht druckbaren sowie 95 druckbaren Zeichen. Druckbare Zeichen sind, beginnend mit dem Leerzeichen:
!"#$%&'()*+,-./0123456789:;<=>?
@ABCDEFGHIJKLMNOPQRSTUVWXYZ[\]^_
`abcdefghijklmnopqrstuvwxyz{|}~ „

die 33 nicht Druckzeichen sind Steuerbefehle:

10 => LF => LineFeed => Zeilenumbruch
13 => CR => CarriageReturn => Wagenrücklauf

Vom Bit zum CBC – Zeichensatz

Merke:

„Es gibt eine Tabelle in der stehen alle Symbole mit Ihren Zahlenwerten.“

„In einem Computer werden nur die Zahlenwerte benutzt. Sowas wie Buchstaben kennt ein Computer gar nicht.“

„Das man einen Text „Lesen“ kann liegt daran, daß es Zahlenwert-in-Symbolumwandlerprogramme gibt.“

„Jedes Symbol hat einen festen Zahlenwert (Platz) im Zeichensatz.“

„Jede Zahl besteht aus Einsen und Nullen.“

Vom Bit zum CBC - Binärsystem

algebraische: Addition

```
  01010101
+ 00100001
------------
  01110110
```

Vom Bit zum CBC - Binärsystem

algebraisch: Addition

```
 01010101
+00100001
----------
 01110110
+00100001
----------
 10010111
```

Vom Bit zum CBC - Binärsystem

logisch: UND

```
 01010101
&00100001
----------
 00000001
```

Vom Bit zum CBC - Binärsystem

logisch: UND

```
 01010101
&00100001
----------
 00000001
&00100001
----------
 00000001
```

Vom Bit zum CBC - Binärsystem

logisch: ODER

```
 01010101
|00100001
----------
 01110101
```

Vom Bit zum CBC - Binärsystem

logisch: ODER

```
  01010101
| 00100001
------------
  01110101
| 00100001
------------
  01110101
```

Vom Bit zum CBC - Binärsystem

EXCLUSIVE-ODER

```
 01010101
^00100001
----------
 01110100
```

Vom Bit zum CBC - Binärsystem

EXCLUSIVE-ODER

```
 01010101
^00100001
-----------
 01110100
^00100001
-----------
 01010101
```

Vom Bit zum CBC - Binärsystem

EXCLUSIVE-ODER (xor)


```
 01010101
^00100001
----------
 01110100
^00100001
----------
 01010101
```

Vom Bit zum CBC

Schlagzeile:

Grundlagen der Symmetrischen

Verschlüsselung entdeckt!

Vom Bit zum CBC – Symmetrische Verschlüsselung

A ^ B => C ^ B => A

A xor B => C xor B => A

Vom Bit zum CBC – Symmetrische Verschlüsselung

A ^ B => C ^ B => A

Wenn man A als Date betrachtet,
B als Schlüssel,
dann ist C das verschlüsselte Ergebnis.

Kennt man die „Formel“ und den Schlüssel,kann man die Date wieder herstellen => Entschlüsseln.

Vom Bit zum CBC – Symmetrische Verschlüsselung

Natürlich ist ein XOR und ein Ein-Bit Schlüssel, keine sichere Verschlüsselung.

Vom Bit zum CBC – Symmetrische Verschlüsselung

Was wäre, wenn der Schlüssel 4096 Bits lang wäre und die Bitfolge (fast) zufällig generiert wurde?

Vom Bit zum CBC – Symmetrische Verschlüsselung

Was wäre, wenn der Schlüssel 4096 Bits lang wäre und die Bitfolge (fast) zufällig generiert wurde?

Besser :)

Vom Bit zum CBC – Symmetrische Verschlüsselung

$$An \wedge Bn = Cn$$

Jetzt hat jede Stelle seinen eigenen Schlüssel und wenn der Schlüssel länger ist, als die zu verschlüsselnde Bitfolge, dann ist alles gut.

Vom Bit zum CBC – Symmetrische Verschlüsselung

„ABC“

„41 42 43“

Vom Bit zum CBC – Symmetrische Verschlüsselung

„ABC“

„41 42 43“

„01000001 01000010 01000011“

Umwandlung von ASCII in Binärzahlen

Vom Bit zum CBC – Symmetrische Verschlüsselung

Lernbeispiel: Hexdezimalzahlen

echo "ABC" | hexdump

00000000 41 42 43 0a

0x0a = 10d = LineFeed (Zeilenumbruch von echo)

Vom Bit zum CBC – Symmetrische Verschlüsselung

Lernbeispiel: Hexdezimalzahlen

0x00 = 0000 0000	0x08 = 0000 1000
0x01 = 0000 0001	0x09 = 0000 1001
0x02 = 0000 0010	0x0A = 0000 1010
0x03 = 0000 0011	0x0B = 0000 1011
0x04 = 0000 0100	0x0C = 0000 1100
0x05 = 0000 0101	0x0D = 0000 1101
0x06 = 0000 0110	0x0E = 0000 1110
0x07 = 0000 0111	0x0F = 0000 1111

0x38 = 0011 1000
0xFA = 1111 1010

„Ein Byte besteht hexadezimal immer aus 2 Stellen.“

Vom Bit zum CBC – Symmetrische Verschlüsselung

Der Satz : "Ich gehe nach Hause." in Hexadezimal:

```
49 63 68 20 67|65 68 65 20 6e|61 63 68 20 48|
61 75 73 65 2e
```

in Binär:

```
01001001 01100111 01101000 00100000 01100111
01100101 01101000 01100101 00100000 01101110
01100001 01100011 01101000 00100000 01001000
01100001 01110101 01110011 01100101 00101110
```

Vom Bit zum CBC – Symmetrische Verschlüsselung

Das Passwort : "Passwort" in Hexadezimal:

```
50 61 73 73 77|6f 72 74
```

in Binär:

```
01010000 01100001 01110011 01110011 01110111
01101111 01110010 01110100
```

Vom Bit zum CBC – Symmetrische Verschlüsselung

Die ersten 5 Zeichen Verschlüsselung: An ^ Bn = Cn

```
   I          c          h                     g
01001001 01100111 01101000 00100000 01100111 A
01010000 01100001 01110011 01110011 01110111 B
-------- -------- -------- -------- --------
00011001 00000110 00011011 01010011 00010000 C
```

```
0        1        2        3        4 ...  n
```

Vom Bit zum CBC – Symmetrische Verschlüsselung

Die ersten 5 Zeichen Verschlüsselung: An ^ Bn = Cn

I	c	h		g	
01001001	01100111	01101000	00100000	01100111	A
01010000	01100001	01110011	01110011	01110111	B
---	---	---	---	---	
00011001	00000110	00011011	01010011	00010000	C
{np}	{np}	{np}	S	{np}	

{np} = nicht druckbares Zeichen

Vom Bit zum CBC – Symmetrische Verschlüsselung

Die ersten 5 Zeichen Verschlüsselung: An ^ Bn = Cn

```
    I          c          h                     g
01001001 01100111 01101000 00100000 01100111 A
01010000 01100001 01110011 01110011 01110111 B
-------- -------- -------- -------- --------
00011001 00000110 00011011 01010011 00010000 C
  {np}     {np}     {np}       S       {np}
```

Die ersten 5 Zeichen Entschlüsselung: Cn ^ Bn = An

```
00011001 00000110 00011011 01010011 00010000 C
01010000 01100001 01110011 01110011 01110111 B
-------- -------- -------- -------- --------
01001001 01100111 01101000 00100000 01100111 A
```

Vom Bit zum CBC – Symmetrische Verschlüsselung

Warum reicht das nicht?

Unser Zeichensatz besteht in der Regel aus Bytes, d.h. ein Zeichen eines Textes ist ein Byte lang. Die Zeichen stehen in dem Zeichensatz an einer festen Stelle einer dicken Tabelle.

Vom Bit zum CBC – Symmetrische Verschlüsselung

Warum reicht das nicht?

Man muß also beim Codeknacken nur Ergebnisse beachten, die an diesen Stellen der Tabelle im Zeichensatz stehen.

Warum? Weil das sonst der Empfänger nicht lesen kann.

Das schränkt die Suche pro Byte immens ein, auch wenn der Schlüssel sehr lang ist.

Vom Bit zum CBC – Symmetrische Verschlüsselung

Ende Teil 1

Fragen ?

Vom Bit zum CBC – F.A.Q. Sektion

„Was passiert, wenn das Passwort kürzer als der zu verschlüsselnde Text ist ?“

Na, dann fängt man mit dem Passwort wieder von vorne an.

Vom Bit zum CBC – F.A.Q. Sektion

„Ok, ich habe meinen Text verschlüsselt. Kann ich jetzt dem Empfänger beides schicken ?“

„Ähm,... Nein !“

Vom Bit zum CBC – F.A.Q. Sektion

„Das Passwort muß man auf einem zweiten sicheren Kanal austauschen, oder ?“

Ja, außer man besucht Teil 2 dieses Kurses:

„Asymmetrischer Schlüsselaustausch“