

CryptIt:

Einfaches Beispiel einer symmetrischen Vinerge-Verschlüsselung. Dabei werden die Buchstaben ueber einen Offset verschoben. Dabei gibt ein Schluesselwort (Key) die jeweiligen Offsets an.

Die Verschiebung erfolgt nach dem folgenden Alphabet:

Offset	Entsprechendes Alphabet
1	ABCDEFGHIJKLMNOPQRSTUVWXYZ
2	BCDEFGHIJKLMNOPQRSTUVWXYZA
3	CDEFGHIJKLMNOPQRSTUVWXYZAB
4	DEFGHIJKLMNOPQRSTUVWXYZABC
5	EFGHIJKLMNOPQRSTUVWXYZABCD
6	FGHIJKLMNOPQRSTUVWXYZABCDE
7	GHIJKLMNOPQRSTUVWXYZABCDEF
8	HJKLMNOPQRSTUVWXYZABCDEFG
9	IJKLMNOPQRSTUVWXYZABCDEFGH
10	JKLMNOPQRSTUVWXYZABCDEFGHI
11	KLMNOPQRSTUVWXYZABCDEFGHIJ
12	LMNOPQRSTUVWXYZABCDEFGHIJK
13	MNOPQRSTUVWXYZABCDEFGHIJKL
14	NOPQRSTUVWXYZABCDEFGHIJKLM
15	OPQRSTUVWXYZABCDEFGHIJKLMN
16	PQRSTUVWXYZABCDEFGHIJKLMNO
17	QRSTUVWXYZABCDEFGHIJKLMNOP
18	RSTUVWXYZABCDEFGHIJKLMNOPQ
19	STUVWXYZABCDEFGHIJKLMNOPQR
20	TUVWXYZABCDEFGHIJKLMNOPQRS
21	UVWXYZABCDEFGHIJKLMNOPQRST
22	VWXYZABCDEFGHIJKLMNOPQRSTU
23	WXYZABCDEFGHIJKLMNOPQRSTUV
24	XYZABCDEFGHIJKLMNOPQRSTUVW
25	YZABCDEFGHIJKLMNOPQRSTUVWX
26	ZABCDEFGHIJKLMNOPQRSTUVWXY

Dabei wird nach folgendem Algorithmus verfahren:

Jeder einzelne Buchstabe wird um einen angegebenen Offset verschoben:

Offset		Buchstabe vor		Buchstabe nach
		dem Verschieben		dem Verschieben
1		a		b
2		a		c
3		a		d
usw.				

tab.1: Offsetermittlung

Das Verschieben uebernimmt dabei die Funktion MoveForward

```

char MoveForward(char cChar)
{
    if (cChar == 'z')
        cChar = 'a';
    else
        cChar++;

    return cChar;
}

```

Der um 1 verschobene Buchstabe wird als Rueckgabewert uebergeben.

In der Funktion Crypt wird nun die Verschluesselung durchgefuehrt.

Der per Parameter uebergebene Text wird mit dem Key-Wort verschluesst. Zunaechst wird ueber den Text das Schluesselwort gesetzt, so dass der folgende Ausdruck entsteht:

z.b.: Text = "ichbineintest", Keyword="schluessel"

```

    schluesselsch
    ichbineintest

```

Die Differenz des ASCII-Codes vom Text und dem darueberliegenden Schluesselwort ergibt nun den Offset (siehe Tab.1) ermittelt, das Zeichen wird ueber den Offset verschoben.

Die programmtechnische Loesung hat die folgende Form:

```

int Crypt(char *pcText, char *pcKey, char *pcCrypt)
{
    int    i1, i2, i3, iText, iKey, iDiff=0, iVal;
    char   cChar;

    iText = strlen(pcText);
    iKey = strlen(pcKey);

    i2=0; i3=0;
    for (i1=0; i1<iText; i1++)
    {
        iVal = pcKey[i3];
        if (pcKey[i3]=='a')
            iDiff = 0;
        else
            iDiff = iVal - 97;

        for (i2=0; i2<iDiff; i2++)
            pcCrypt[i1] = MoveForward(pcText[i1]);

        if (i3 == (iKey-1))
            i3 = 0;
        else

```

```

        i3++;
    }
    pcCrypt[iText] = '\0';

    return 1;
}

```

Hierbei wird zunaechst die Schluessellaenge und die Textlaenge ermittelt.

```

iText = strlen(pcText);
iKey = strlen(pcKey);

```

Nun wird jeder Buchstabe um den Offset verschoben, der fuer das jeweilige Zeichen ermittelt wird:

```

for (i1=0; i1<iText; i1++)
{
    iVal = pcKey[i3];
    if (pcKey[i3]=='a')
        iDiff = 0;
    else
        iDiff = iVal - 97;

    pcCrypt[i1] = pcText[i1];
    for (i2=0; i2<iDiff; i2++)
        pcCrypt[i1] = MoveForward(pcCrypt[i1]);

    if (i3 == (iKey-1))
        i3 = 0;
    else
        i3++;
}

```

Die Verschiebung erfolgt mit der bereits beschriebenen Funktion MoveForward. Bei der Entschluesselung wird der entsprechende Algorithmus rueckwaerts durchgefuehrt.

Der Code fuer die Funktionen MoveBack und Decrypt hat die Form:

```

char MoveBack(char cChar)
{
    if (cChar == 'a')
        cChar = 'z';
    else
        cChar--;

    return cChar;
}

```

und

```

int Decrypt(char *pcCrypt, char *pcKey, char *pcText)
{
    int i1, i2, i3, iCrypt, iKey, iVal=0, iDiff;

    iCrypt = strlen(pcCrypt);
    iKey = strlen(pcKey);

    i2=0; i3=0;
    for (i1=0; i1<iCrypt; i1++)
    {
        iVal = pcKey[i3];
        if (pcKey[i3]=='a')
            iDiff = 0;
        else
            iDiff = iVal - 97;
        pcText[i1] = pcCrypt[i1];
        for (i2=0; i2<iDiff; i2++)
            pcText[i1] = MoveBack(pcText[i1]);

        if (i3==(iKey-1))
            i3 = 0;
        else
            i3++;
    }
    pcText[iCrypt]='\0';
    return 1;
}

```

So, anzumerken bleibt, dass beide Seiten das keyword kennen muessen (und es sollte halt kein anderer kennen). Diese Verschluesselung ist nicht besonders sicher, Charles Baggage (hatte die Idee mit der Differenzenmaschine, mit der wir immer noch arbeiten und Computer nennen) hat diesen Algorithmus schon vor langer Zeit geknackt. Das Keqword sollte immer recht lang gewaehlt werden.

Ausserdem hatte ich keine Lust mehr, eine Abfrage fuer Leerzeichen einzubauen.

Nun ja, das sollte der interessierte Coder allerdings sicherlich selbst hinkriegen. Oder vielleicht mache ich mir ja mal die Arbeit.

Der Quellcode kann auch auf jeden C++ Compiler compiliert werden, ich habe das zwar probiert, allerdings noch nicht zu Downloaden klargemacht.

Okay, das solls erstmal gewesen sein. Fragen?

email: sir_kimmi@gmx.de
hp: www.sir-kimmi.de

Bis dann, Kimmi