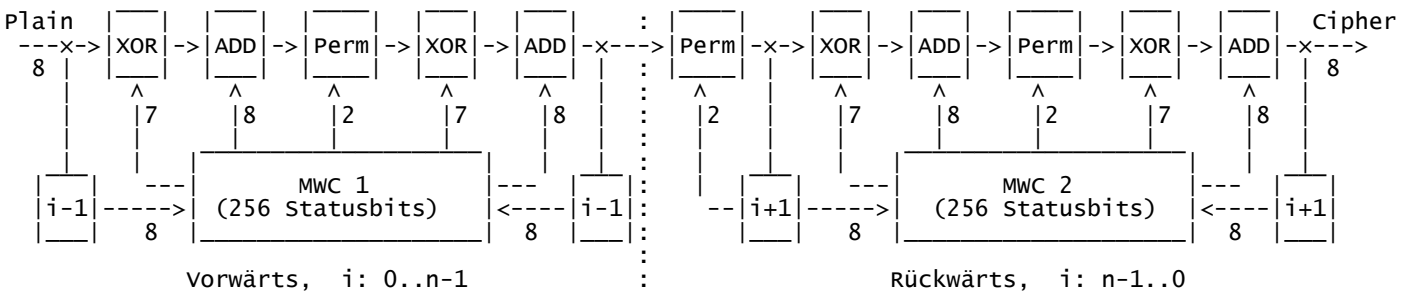
7KE-Verschüsselung




```
 Perm: Permutation durch eine von vier Abbildungs-Tabellen, x = PTab[t][x], t: 0..3
 ADD:  8-Bit-Addition (Modulo 256)
 XOR:  8-Bit-Exklusiv-Oder
 i-1:  Zwischenspeicher/Verzögerungsglied vorwärts, bei i=0 Startwerte aus dem Schlüssel
 i+1:  Zwischenspeicher/Verzögerungsglied rückwärts, bei i=n-1 Startwerte aus dem Schlüssel
 MWC:  2 Vierfach-Pseudo-Zufallsgeneratoren, Periode je 2.6e76,  Multiply With Carry
       t=a*x+c+e, x=t%b, c=(int)(t/b), b=1<<32, e: 16-Bit-Einkopplung der Verzögerungsglieder
       a=A[k], x=X[k], c=C[k], k: 0..3 bei MWC 1 und 4..7 bei MWC 2, iterierend

Sichere Primzahlen: a*b-1 und a*b/2-1 sind Primzahlen, mit a=A[k] und b=2^32,
A[8]={4164903690,4204114314,4187999619,4198054089,4210396968,4197302403,4178097609,4201298934};
Damit ist die Periode eines MWC-Pseudozufallsgenerators = a*b/2-1
Kombination von 4 Generatoren -> Periode pro 32-Bit-Wert ungefähr 4*(a*b/2-1)^4 = 2.6*10^76
(genauer 4 * Produkt aus vier Primzahlen A[k]/2-1, gilt auch wenn e ungleich 0 aber konstant ist)
Test z.B.: https://www.wolframalpha.com/input/?i=Is+(4164903690+*+2%5E31+-+1)+prime%3F
```

7 KBit Encryption (7KE):
Dieses Verschlüsselungsverfahren verwendet beliebige eindeutige und umkehrbare Abbildungen (auch bijektiv oder eineindeutig) zwischen zwei 8-Bit-Werten (bijektive Selbstabbildung oder Permutation - nicht zu verwechseln mit Transposition).
Dabei gibt es 256! (Fakultät von 256) mögliche Abbildungen, das sind fast 10^{507} oder 2^{1684} Abbildungen (zum Vergleich, nur 256 von all diesen Abbildungen beschreiben eine XOR-Operation).
Zur Erstellung einer Abbildungstabelle sind 1684 Bit erforderlich (GammaLog(1+256)/log(2)=1684).
Von all den 10^{507} möglichen Abbildungen werden völlig willkürlich vier herausgegriffen, und zwar abhängig vom Schlüssel.
Dafür kommen 4 * 1684 = 6736 Bits vom insgesamt 7280 Bit großen Schlüssel zum Einsatz.
Durch Modifikation und Verkettung entstehen daraus $266 = 7 * 10^{19}$ unterschiedliche Abbildungen, das reicht, um selbst bei 64 Exabyte (64 Millionen Terrabyte) jedes Byte mit einer einzigartigen Abbildung kodieren zu können, so dass statistische Analysen von vorn herein zum Scheitern verurteilt sind.

Welches Byte im Datenblock mit welcher Abbildung kodiert wird, ist abhängig vom Schlüssel, von den unverschlüsselten und den verschlüsselten Daten und zwei Pseudozufallsgeneratoren mit jeweils 256 Bit großem "Seed" (Vierfach-MWC-Zufallsgeneratoren, Status je 256 Bit, Periode je etwa $2.6*10^{76}$).
Deren Initialisierungswerte (Seeds) sind ebenfalls Teil des Schlüssels, so dass sich zusammen mit 32 Initialisierungsbits eine Schlüsselgröße von 4 * 1684 + 2 * 256 + 32 = 7280 Bit ergibt.
Diese werden aus einem beliebig großen Passwort durch wiederholte Anwendung der Hash-Operation SHA-256 gewonnen, s. a. CalcSha256.

Die Blockgröße ist beliebig, die Daten eines Blocks müssen aber komplett im Speicher stehen.
Damit die Änderung nur eines Bits im Block ein vollkommen unterschiedliches Ergebnis zur Folge hat, wird jeder Block vorwärts und rückwärts durchlaufen mit jeweils unterschiedlichen Parametern (daher auch zwei Pseudozufallsgeneratoren, je einer für vorwärts und rückwärts).
Sind die Daten zu groß, um sie komplett in dem Speicher zu laden, kann man sie in Blöcke unterteilen.
Dann sorgt der Parameter dStartBlock dafür, dass jeder Block auf andere Weise verschlüsselt wird.
Der Befehl CryptFile verwendet bei Dateien ab einer Größe von 4096 Bytes eine Blockgröße von 4096.
Die Geschwindigkeit ist zumindest bei kleinen Blockgrößen vergleichbar mit AES-256 (CryptAES).

Kommandozeilenprogramm 7KE.exe mit Quellcode → https://www.dropbox.com/s/bv9jejxx2qsa8ur/Crypt7KE.zip?dl=1