

# CypherMatrix

## Hashfunktion

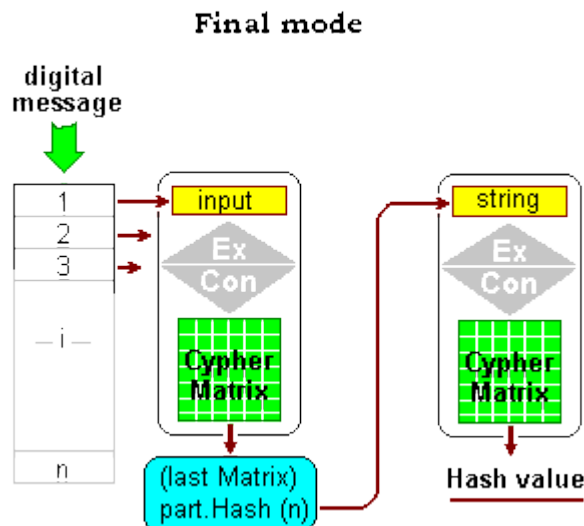
Aufgabe einer Hashfunktion ist die eindeutige Kennzeichnung digitaler Informationen als unveränderliches Merkmal:

Texte (Nachrichten), Zeichnungen, digitale Bilder, digitale Musik, Programme, Messergebnisse und weitere digital gespeicherte Informationen.

Im Verlauf der Forschung digitaler Sachverhalte wurden viele Hashfunktionen entwickelt, die ihre Aufgabe auch mehr oder weniger gut erfüllen. Als Problem bleibt jedoch, dass die Verfahren auch sicher genug sein müssen (eindeutig und unangreifbar). Die bisher verwendeten Verfahren der SHA-Familie sind in letzter Zeit durch spezielle Angriffe gefährdet. NIST [#1] führt daher zur Zeit einen internationalen Wettbewerb durch, in dem am Ende das am besten geeignete Verfahren gekürt werden soll.

Die bereits bekannt gewordenen Kandidaten [#2] sind im Wesentlichen nach einem Schema gestaltet, das weitgehend dem Aufbau bisheriger Hashfunktionen entspricht. Um mit der [CypherMatrix Hashfunktion](#) zu vergleichen, soll im Folgenden eine Gegenüberstellung der beiden Bereiche vorgenommen werden.

Die CypherMatrix Hashfunktion schematisiert:



Der grundsätzliche Unterschied zeigt sich bereits in den Grundelementen: Sowohl die aktuellen Verfahren als auch die Kandidaten im NIST Wettbewerb verwenden „**Bits**“ als kleinste Arbeitseinheit während CypherMatrix nur **Bytes (8 Bits)** als Zahlen verarbeitet (Zahlensysteme von Basis 2 bis zur Basis 256). Dementsprechend ist CypherMatrix im Grunde genommen nur ein rein mathematisches Verfahren.

<b>Hashfunktionen (allgem.)</b>	<b>CypherMatrix Hashfunktion</b>
<b>Grundelemente</b>	
<b>Bits</b>	<b>Bytes</b>
<b>Zusatzfunktionen</b>	
IVs, SALT, padding	keine
<b>Arbeitsschritte, Sequenzen</b> (message digest)	
224, 256, 384, 512, 1024 bits (teilweise: variabel)	kontinuierlich, unbegrenzt (optimal: 16 bis 256 Bytes)
<b>Interne Beschaffenheit</b> (internal states)	
Feistel network keys, S-boxes, constants, shifting, rotation, mixing, XOR swapping, permutations	positionsgewichtet Hash-Konstante C Expansion, Kontraktion (Einwegfunktionen)
<b>Kompressionsfunktion</b>	
„Merkle-Damgård“ block ciphers, AES-based Threefish based	keine
<b>Ausgabefunktion</b>	
output function truncated to output fixed length	CypherMatrix: $GF(16^2)$ dreifache Permutation Zahlensystem zur Basis 62 (9 bis 11 Ziffern)
<b>Anwendungen</b> (applications)	
hashing, MAC, randomizing, PRNG, digital signatures, authentication, encryptions,	Hashfunktion, Zufallsfolgen digitale Signaturen, RNG, Authentifizierung, Verschlüsselungen

## Arbeitsschritte

Der Datenverlauf in der CypherMatrix Hashfunktion wird am Beispiel des oben im ersten Abschnitt gespeicherten Textes - „Aufgabe einer Hashfunktion ..... gekürt werden soll“ - (Beispiel.txt: 761 Bytes) mit verschiedenen langen Hashsequenzen dargestellt.

### A. Hashsequenz = 47 Bytes

Parameter:   Length hash sequence [16 – 256]:   **47**  
                  Number base output [36 – 82]:   **62**  
                  Expansion factor [35 – 255]:   **77**  
                  User code number [1 – 99]:   **1**

Das Programm **CMHash.exe** errechnet folgende Ergebnisse:

Serial Matrix Values in Beispiel.txt

-----  
1035642307326 IERtZug 1  
1106193454635 JTSV0DT 2  
833575103911 EfsqBYV 3  
969677730798 H4RgwBy 4  
1050660534744 IUqGWvA 5  
888134198519 FdRAaPP 6  
1086284186419 J7j7k4B 7  
965941727932 H0Mr3BE 8  
944484830070 Gcwk5rS 9  
1017091570096 HuCSZjU 10  
756727658835 DK088Zv 11  
988458928109 HOwivZF 12  
1011808639945 HoQvwGf 13  
787794925675 DrudDsx 14  
979370393303 HF1eMUB 15  
1023442680251 I18HBc3 16  
2379871530 2b3hlg 17

-----  
decimal: 15447668742098  
base 16: E0CB0CED7D2  
base 62: **4NxoxIIC**

-----  
Matrix value last cycle:  
decimal: 9135905201329335  
base 16: 20750EB03A14B7  
base 62: **fqEjWivUF**

Final **Hash value** for: BEISPIEL.TXT  
decimal: 9151352870071433  
base 16: 20831B6108EC89  
base 62: **fuchLghFR**

---

## B. Hashsequenz = 64 Bytes

Parameter:	Length hash sequence [16 – 256]:	<b>64</b>
	Number base output [36 – 82]:	<b>62</b>
	Expansion factor [35 – 255]:	<b>77</b>
	User code number [1 – 99]:	<b>1</b>

Das Programm **CMHash.exe** errechnet folgende Ergebnisse:

Serial Matrix Values in Beispiel.txt

---

5135317144137	1SPQZOObB	1
4240610916255	1CeodF8B	2
4658062293460	1K0TygBI	3
5207704644798	1TgRRyIQ	4
4730291412416	1LHK8hYu	5
4296783479944	1De89azw	6
4415576962642	1FjnavWQ	7
4648838326867	1JqPjr6Z	8
4133242430358	1AlcO1Be	9
4366099196483	1Ern9HJz	10
4517376446902	1HWuxCju	11
2661722168482	krO6xhS	12

---

decimal: 53011625422744  
base 16: 3036BB1A2F98  
base 62: **F3IYrrLs**

---

Matrix value last cycle:  
decimal: 9217987451786080  
base 16: 20BFB5F39B9360  
base 62: **gDXpygqJc**

---

Final **Hash value** for: BEISPIEL.TXT  
decimal: 9270999077208824  
base 16: 20EFECAEB5C2F8  
base 62: **gSb8XYhfU**

---

### C. Hashsequenz = 128 bytes

Parameter: Length hash sequence [16 – 256]: **128**  
Number base output [36 – 82]: **62**  
Expansion factor [35 – 255]: **77**  
User code number [1 – 99]: **1**

Das Programm **CMHash.exe** errechnet folgende Ergebnisse:

Serial Matrix Values in Beispiel.txt

-----

148895869216428 gHOVRG0K 1  
158465762316310 izsTNR8Y 2  
143759512328681 eoxwVCav 3  
147486277455189 fsZsDsWb 4  
137507860187648 d2tz5vM0 5  
112316817381538 VtOpjWny 6

-----

decimal: 848432098885794  
base 16: 303A4FDDEC0A2  
base 62: **3sv7cNEQM**

-----

Matrix value last cycle:  
decimal: 9399682672710725  
base 16: 2164F62AB55445  
base 62: **h38gRhHtN**

-----

Final **Hash value** for: BEISPIEL.TXT  
decimal: 10248114771596519  
base 16: 24689B289414E7  
base 62: **kw3o44WJj**

=====

### D. Hashsequenz = 256 Bytes

Parameter: Length hash sequence [16 – 256]: **256**  
Number base output [36 – 82]: **62**  
Expansion factor [35 – 255]: **77**  
User code number [1 – 99]: **1**

Das Programm **CMHash.exe** errechnet folgende Ergebnisse:

## Serial Matrix Values in Beispiel.txt

---

4.94718332115168E+15 Menw4WEY8 1  
4.72044524527918E+15 LcQ5JMkB5 2  
4.01108316364358E+15 IMzMqHGsi 3

---

decimal: 13678711730074427  
base 16: 3098B741A7F33B  
base 62: **10eDOEAFbv**

---

Matrix value last cycle:  
decimal: 9051482939056357  
base 16: 2028469945E0E5  
base 62: **fSGQqdtob**

---

Final **Hash value** for: BEISPIEL.TXT  
decimal: 22730194669130784  
base 16: 50C0FDDAEDD420  
base 62: **1g6Tp4o9QW**

---

---

## Erläuterungen

Die Ausgabefunktion des CypherMatrix Verfahrens stellt die errechneten Hashwerte in Zahlen des Zahlensystems zur **Basis 62** dar.

Das Zahlensystem zur Basis 62 umfasst folgende Ziffern:

**0123456789ABCDEFGHIJKLMNPOQRSTUVWXYZ**

**abcdefghijklmnopqrstuvwxy**

(definiert vom Autor, nicht standardisiert)

Die errechneten Hashwerte können aber auch in allen anderen Zahlensystemen von zur Basis 2 bis zur Basis 256 ausgegeben werden. Mit den Werten lassen sich alle mathematischen Operationen durchführen (Addition, Subtraktion, Multiplikation, Division, MODULO Rechnungen und weitere).

Sie können die obigen Beispiele selbst mit dem DEMO-Programm

### **CMHash.exe**

testen. Es steht unter [Telecypher.net/ZUSENDEN.HTM](http://Telecypher.net/ZUSENDEN.HTM) als Download zur Verfügung.

**Hinweise:**

[#1] NIST: National Institute of Standards and Technology, USA

[#2] Grøstl, LANE, SHAvite-3, Skein, TIB3, ...

siehe: [//ehash.iak.tugraz.at/wiki/The\\_SHA-3\\_Zoo](http://ehash.iak.tugraz.at/wiki/The_SHA-3_Zoo)

München, den 12. Januar 2009

© Ernst Erich Schnoor

---