Brute-Force-Hacking: mathematische Grundlagen

1. Untersuchung I: Annäherung

Zunächst wollen wir uns der Mathematik mit einer einfachen **Untersuchung** nähern. Dazu ist folgende Tabelle gegeben, die bereits teilweise ausgefüllt ist.

a) Vervollständige die folgende Tabelle.

Zeichen-anzahl	Passwort-länge	mögliche Passwörter	Passwort-anzahl
1 (A)	1	A	1
	2	AA	1
	3	AAA	1
2 (A, B)	1	A; B	2
	2	AA; AB; BA; BB	
	3		
3 (A, B, C)	1		
	2		
	3		

2. Verallgemeinerung mittels einer Formel

Wie wir wissen, ist im Hinblick auf die *Passwortsicherheit* entscheidend, wie viele Passwörter (Passwortanzahl) jeweils erstellt werden können. Je mehr es sind, desto besser, weil so durch die „rohe Gewalt“ der Brute-Force-Methode mehr Passwörter getestet werden müssen, was wiederum mehr Zeit kostet.

Die obige Tabelle lässt sich natürlich nicht unendlich lange fortführen. Es ist erforderlich, die mathematische Regelhaftigkeit zu erkennen, die in der Passwortanzahl steckt.

a) Erstelle daher eine **Formel**, mit Hilfe der man aus der Zeichenanzahl und Passwortlänge die Passwortanzahl *berechnen* kann.

Tipp: Es kann hilfreich sein, ergänzend die Passwortanzahl für die Zeichenanzahl 2 und Passwortlänge 4 zu notieren.

Passwortanzahl = _____

3. Wissenschaftliche Schreibweise I: Grundlagen

Bevor wir den nächsten Berechnungsschritt gehen, widmen wir uns der mathematischen Schreibweise für sehr große Zahlen. Der Sinn liegt darin, dass sehr große (oder auch sehr kleine) Zahlen, bestehend aus vielen Ziffern, schwer lesbar und vergleichbar sind. Daher hat sich eine einheitliche, leicht verständliche Schreibweise herauskristallisiert: die **wissenschaftliche Schreibweise** (engl. „scientific notation“). Aufgebaut ist diese wie folgt:

<eine Vorkommastelle (1-9, ohne 0)> **,** <beliebig viele Nachkommastellen> • $10^{\text{<Hochzahl>}}$

Beispiele:

$3{,}141159 \cdot 10^{5}$
$2{,}718 \cdot 10^{-3}$

Zur Erinnerung: Potenzen

Eine Potenz ist wie folgt definiert: $\text{Basis}^{\text{Exponent}}$, z. B. $5^3 = 5 \cdot 5 \cdot 5 = 125$.

Dabei wird die Basis Exponent-mal mit sich selbst multipliziert (im Beispiel: die 5 3-mal mit sich selbst). Die Potenz 5^3 darf nicht verwechselt werden mit dem Produkt 3 • 5, da dies eine Addition darstellt: 5 + 5 + 5 = 15. Anstelle Exponent sagt man auch Hochzahl. Potenzen zur Basis 2 nennt man Zweierpotenzen (kommen häufig in der Computertechnik vor), zur Basis 3 Dreierpotenzen, …, zur Basis 10 Zehnerpotenzen (s. u.) usw.

Auf diese Art lassen sich alle Zahlen darstellen, ggf. gerundet auf eine bestimmte Anzahl von Nachkommastellen. Die Multiplikation mit einer Zehnerpotenz am Ende (• 10^x) bedeutet dabei eine Verschiebung des Kommas um x Stellen nach *rechts**, wenn x *positiv* ist – und x Stellen nach *links*, wenn x *negativ* ist. Beispiel:

$1{,}0 \cdot 10^{1} = 10$ (Komma von 1,0 um 1 Stelle nach rechts)
$1{,}0 \cdot 10^{-1} = 0{,}1$ (Komma von 1,0 um 1 Stelle nach links)

$1{,}0 \cdot 10^{6} = 1000000$ (Komma von 1,0 um 6 Stellen nach rechts)
$1{,}0 \cdot 10^{-6} = 0{,}000001$ (Komma von 1,0 um 6 Stellen nach links)

$1{,}0 \cdot \underline{10^{0}} = 1{,}0 \cdot \underline{1} = 1$ (Es ist festgelegt: Eine Zahl hoch 0 ergibt *immer* 1!)

*** Anmerkung**

Für positive x gilt: 10^x bedeutet eine 1 mit x Nullen dahinter, z. B. $10^3 = 1000$.

Tipp

Teste diese und eigens ausgedachte Aufgaben mit dem Taschenrechner (nicht Handy, da die Apps den mathematischen Ausdruck häufig nicht adäquat darstellen) aus!

Damit ergeben sich für die obige Schreibweise folgende Zahlen:

$3{,}141159 \cdot 10^{5} = 314115{,}9$ (Komma von 3,141159 um 5 Stellen nach rechts)
$2{,}718 \cdot 10^{-3} = 0{,}002718$ (Komma von 2,718 um 3 Stelen nach links)

Hinweis

Mit negativen Zahlen funktioniert es ebenso: $-2{,}718 \cdot 10^{-3} = -0{,}002718$

a) Schreibe ohne die Verwendung von Zehnerpotenzen.

$1{,}23456789 \cdot 10^{8} =$ _____ $1{,}23456789 \cdot 10^{9} =$ _____

$4{,}24242 \cdot 10^{-4} =$ _____ $4{,}24242 \cdot 10^{-1} =$ _____

b) Gib in wissenschaftlicher Schreibweise an.

13543 = _____ 4200000 = _____ 0,000175 = _____

4. Wissenschaftliche Schreibweise II: Größenordnung erkennen

Dank dieser Schreibweise ist man nun in der Lage, mit einem Blick die **Größenordnung** der vorliegenden Zahl zu erfassen. Dabei spielt es weniger eine Rolle, ob vor dem Komma eine 0, 1, 2, 3 … oder 9 stehen. Vielmehr interessiert die Hochzahl bei der Zehnerpotenz. Betrachten wir im Folgenden der Einfachheit halber nur positive Hochzahlen.

a) Vervollständige die folgende Tabelle bis 10^{12} (Genaueres zu den SI-Präfixen siehe online unter https://bit.ly/3yPkHqG und https://bit.ly/2fYm3mZ).

Beispiel	Hoch-zahl	Ergebnis	Größen-ordnung	SI-Präfix
$1{,}0 \cdot 10^0 = 1{,}0 \cdot 1$	0	1	Eins	–
$1{,}0 \cdot 10^1 = 1{,}0 \cdot 10$	1	10	Zehn	Deka
$1{,}0 \cdot 10^2 = 1{,}0 \cdot 100$	2	100	Hundert	Hekto
$1{,}0 \cdot 10^3 = 1{,}0 \cdot 1.000$	3	1.000	Tausend	Kilo
$1{,}0 \cdot 10^4 = 1{,}0 \cdot 10.000$	4	10.000	(Zehn-)Tausend	–
$1{,}0 \cdot 10^5 = 1{,}0 \cdot 100.000$	5	100.000	(Hundert-)Tausend	–
$1{,}0 \cdot 10^6 = 1{,}0 \cdot 1.000.000$	6	1.000.000	Million	Mega
$1{,}0 \cdot 10^7 = 1{,}0 \cdot 10.000.000$	7	10.000.000	(Zehn-)Millionen	–

Es fällt auf, dass ab der Bezeichnung „Tausend“ (10^3) immer in **3er-Hochzahlschritten** ein **neuer Begriff** verwendet wird, der die Größenordnung kennzeichnet:

10^3: Tausend 10^6: Million 10^9: Milliarde 10^{12}: Billion etc.

b) Recherchiere im Internet, mit welchen weiteren Begriffen die Größenordnungen bis 10^{30} beschrieben werden (inkl. SI-Präfixe, soweit möglich).

10^{15}: _____ 10^{18}: _____ 10^{21}: _____

10^{24}: _____ 10^{27}: _____ 10^{30}: _____

5. Untersuchung II: realistische Szenarien

Nun wird es Zeit für den nächsten Schritt. Da die zu Anfang gewählten Zeichenanzahlen und Passwortlängen aus Sicherheitsgründen für den praktischen Gebrauch nicht zu empfehlen sind, wollen wir im Folgenden **realistische Szenarien** betrachten.

a) Vervollständige die folgende Tabelle mit Hilfe der Formel aus Aufgabe 2. Notiere das Ergebnis der Passwortanzahl auf eine Nachkommastelle gerundet in der wissenschaftlichen Schreibweise.

Zeichenraum	Zeichenanzahl	Passwortlänge	Passwortanzahl	Größenordnung
Ziffern	10 (0 bis 9)	5	10^5	(Hundert-) Tausend
		10	10^{10}	(Zehn-) Milliarden
		15	10^{15}	
Kleinbuchstaben (ohne Eszett / scharfes S und ohne Umlaute)		5	$1{,}2 \cdot 10^7$	
		10		
		15		
Kleinbuchstaben, Großbuchstaben (ohne Eszett / scharfes S und ohne Umlaute)		5		
		10		
		15		
Ziffern, Kleinbuchstaben, Großbuchstaben (ohne Eszett / scharfes S und ohne Umlaute)		5		
		10		
		15		

b) Wenn wir die Tabelle nun genauer ansehen, können wir einige Auffälligkeiten erkennen:

- Im größten obigen Zeichenraum (Ziffern, Kleinbuchstaben und Großbuchstaben) reicht die Passwortanzahl bis in den unglaublichen **(Hundert-)Quadrillionen-Bereich** – das entspricht einer „1“ mit 26 Nullen. Der Faktor vor der Zehnerpotenz (z. B. 7,7 im letzten Rechenbeispiel) ist bei dieser Größenordnung sogar vernachlässigbar.

- Dieses Ergebnis ist umso erstaunlicher, als dass wir für obige Untersuchung den Zeichenraum der _____, die gemäß allgemeinen Sicherheitsvorgaben für Passwörter ebenfalls verwendet werden sollen, noch nicht berücksichtigt haben.

- Die Größenordnung der Passwortanzahl bei nur 5 Zeichen Passwortlänge bewegt sich höchstens im Millionen-Bereich. Bei 15 Zeichen ist der Sprung ausgehend von Billiarden über Trilliarden bis hin zu Quadrillionen wesentlich größer. Das bedeutet im Hinblick auf die Passwortsicherheit (kreuze die richtige Antwort an):

  ☐ Die Passwortlänge ist relevanter (wichtiger) als der Zeichenraum.

  ☐ Der Zeichenraum ist relevanter (wichtiger) als die Passwortlänge.

  Dass dies mathematisch so sein muss, sieht man auch schon an der Formel zur Berechnung der Passwortanzahl. Erläutere.

  _____

  _____

  _____

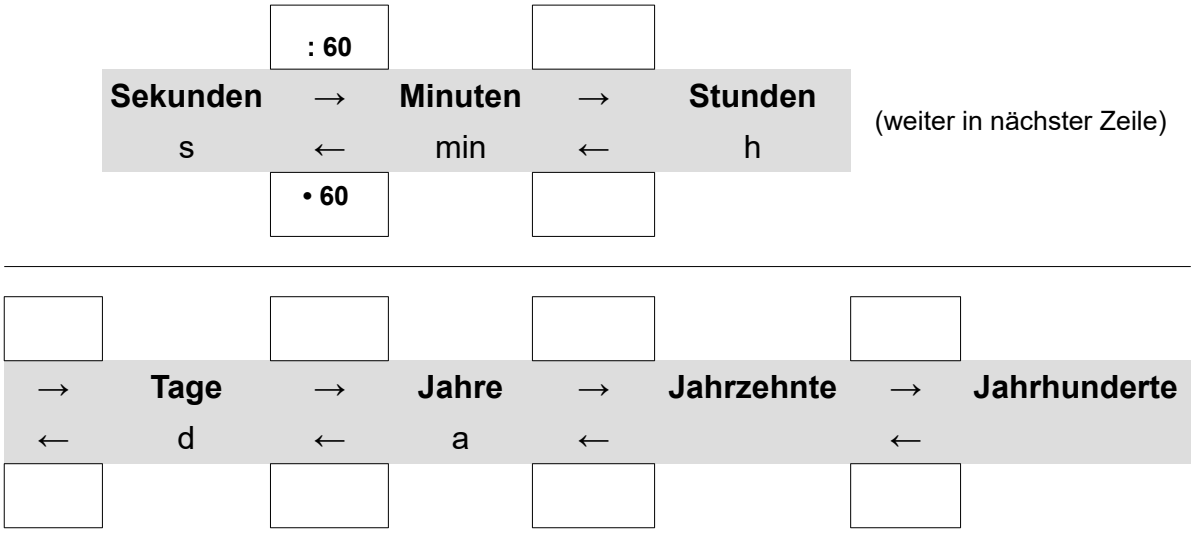
- Fallen Dir noch weitere Dinge auf? Notiere sie auf der Rückseite dieses Blattes.

6. Zeitumrechnung

Nachdem wir nun wissen, welche Größenordnung die Passwortanzahl jeweils erreicht, ist es allmählich an der Zeit zu untersuchen, wie lange man mit der Brute-Force-Methode maximal benötigt, um ein Passwort zu „knacken“. Da es sich hierbei um sehr unterschiedliche Zeitspannen handeln wird, vergegenwärtigen wir uns zunächst die **Umrechnungsfaktoren für Zeiteinheiten**.

a) Vervollständige die folgende Tabelle.

Siehe nächste Seite.



Die folgenden Übungsaufgaben sollen helfen, dieses Wissen anzuwenden.

b) Wandle in die angegebene Einheit um. Notiere das Ergebnis auf 3 Nachkommastellen gerundet in wissenschaftlicher Schreibweise.

42,13 s ≈ _____ min ≈ _____ h

≈ _____ d ≈ _____ a

42,13 a ≈ _____ d ≈ _____ h

≈ _____ min ≈ _____ s

7. Brute-Force-Rechenzeit I: Berechnung

Kommen wir abschließend zur „Krönung“ dieses Skriptums: die Berechnung, wie lange man braucht, um ein bestimmtes Passwort zu „knacken“.

Sofern man keine näheren Informationen zum gesuchten Passwort hat, wodurch es vielleicht schneller gefunden (erraten, berechnet) werden könnte, kann man auf die sog. **Brute-Force-Methode** zurückgreifen. Dabei handelt es sich um eine relativ „stupide“ Vorgehensweise: Es werden *alle* möglichen Passwortkombinationen der Reihe nach (systematisch) ausprobiert. Die Rechenzeit, die man dafür benötigt, ist die Zeit, die man *maximal* braucht, um das Passwort herauszufinden. „Maximal“ deswegen, weil das Passwort beim systematischen Ausprobieren zufällig schon früher gefunden werden kann (i. d. R. sogar wird!), da aller Voraussicht nach ja nicht ausgerechnet das *letzte* auszuprobierende Pass-

wort das gesuchte ist. Das bedeutet: Braucht man mit Brute Force bspw. maximal 10 Stunden, bis ein Passwort gefunden wird, dann wurden nach 10 Stunden *alle* Kombinationen durchprobiert. Das gesuchte Passwort kann allerdings „irgendwo mittendrin“ (sogar am Anfang der Überprüfung) liegen. Mathematisch betrachtet findet man das Passwort durchschnittlich nach der Hälfte der maximalen Zeit, im obigen Beispiel mit 10 Stunden also durchschnittlich nach

_____ Stunden. Kurzum: Brute Force führt (leider) *immer* zum

Erfolg – es ist nur eine Frage der _____ . Doch diese Frage ist aus Sicht der Passwortsicherheit (zum Glück) entscheidend!

Doch mit welcher Formel ermittelt man die Brute-Force-Rechenzeit? Basierend auf der Formel von 2a) konnten wir die Tabelle in 5a) zur Passwortanzahl in wissenschaftlicher Schreibweise füllen. Diese Werte dienen uns nun als Grundlage für die Brute-Force-Rechenzeit. Es ist einleuchtend, dass wenn wir bspw. eine Passwortanzahl von 60 haben und mit einer Rechengeschwindigkeit von zwei Passwörtern pro Sekunde überprüfen können, dass wir 60 : 2 = 30 Sekunden benötigen. Bei einer Verdopplung unserer Rechengeschwindigkeit würden wir nur noch

_____ Sekunden brauchen. Spätestens jetzt erkennt man die der Berechnung zugrunde liegende Formel:

Zeit = Passwortanzahl : Rechengeschwindigkeit

oder genauer:
Maximal benötigte Zeit = Passwortanzahl : Passwortüberprüfungen pro Zeiteinheit

Anstelle von „Passwortanzahl“ oder „Passwortüberprüfungen“ spricht man in der Informatik auch schlichtweg von „Schlüsseln“ oder engl. „keys“. Entsprechend würde die (sehr) ausführliche Rechnung *mit Einheiten* für unser obiges Beispiel mit 60 Passwörtern und einer Rechengeschwindigkeit von zwei Passwörtern pro Sekunde wie folgt aussehen:

$$\text{Zeit} = 60\ \text{keys} : 2\ \text{keys/s} = \frac{60\ keys}{2\ \frac{keys}{s}} = \frac{60\ keys}{1} \cdot \frac{1}{2}\ \frac{s}{keys} = \frac{30 \cdot 1\ s}{1 \cdot 1} = 30\ \text{s}$$

(1) (2) (3) (4) (5)

Zur Erklärung: Von Schritt (1) zu (2) werden das Divisionszeichen „:“ und der Schrägstrich „/“ zu ordentlichen Bruchstrichen umgeschrieben. Es entsteht ein Doppelbruch. Von Schritt (2) zu (3) wird die Rechenregel zur Division von Brüchen angewendet: Man dividiert durch einen Bruch ($2\frac{keys}{s}$), indem man mit seinem Kehrwert multipliziert. Somit werden in Schritt (3) die 2 zu $\frac{1}{2}$ und die Einheit $\frac{keys}{s}$ zu $\frac{s}{keys}$. Von Schritt (3) zu (4) wird über Kreuz gekürzt: 60 und 2 jeweils mit 2, sowie die Einheiten (!) miteinander; dabei erkennt

man sehr schön, wie sich „keys“ und „keys“ komplett wegkürzen, so dass schlussendlich nur noch die Zeiteinheit (hier: Sekunden) übrig bleibt.

Selbstredend, dass moderne Computer aufgrund ihrer enormen Rechengeschwindigkeit in der Lage sind, wesentlich schnellere Berechnungen – und damit deutlich mehr Passwortüberprüfungen pro Sekunde – durchzuführen. (Computer-)Benchmarks zeigen, dass ein moderner Einzelplatz-PC über 2 Milliarden Passwörter pro Sekunde generieren kann (vgl. https://www.1pw.de/brute-force.php, Stand 2016). Neuere Angaben im Internet sprechen von bis zu 5 Milliarden Passwörtern pro Sekunde, im Zusammenschluss eines sog. GPU-Clusters von gar bis zu 500 Milliarden Passwörtern pro Sekunde (Stand 2018).

a) Recherchiere im Internet oder in der Bibliothek, was man unter einem „(Computer-) Benchmark“ versteht. Notiere eine einfache Definition (inkl. Quellenangabe!).

_____

_____

_____

Der einfacheren mathematischen Handhabbarkeit wegen wollen wir für die folgenden Berechnungen von einem „runden“ Wert, nämlich *100 Milliarden Passwörter pro Sekunde*, ausgehen.

b) Vervollständige die folgenden vier Tabellen gemäß dieser Annahme zur Rechengeschwindigkeit. Gib Werte kleiner als 100.000stel oder größer als 100.000 in wissenschaftlicher Schreibweise an und runde sinnvoll. Mit einem Strich „–“ markierte Zellen müssen nicht berechnet werden, weil die Zeiten hier unverhältnismäßig klein bzw. groß werden.

Tipp: Baue auf der Tabelle in 5a) auf.

(1) Zeichenraum: Ziffern (Zeichenanzahl: 10)

Passwort-länge	**Passwort-anzahl**	**Anzahl der maximal* benötigten Zeit**					
		in Sekunden	in Minuten	in Stunden	in Tagen	in Jahren	in Jahr-hunderten
1			–	–	–	–	–
5	10^5		–	–	–	–	–
10	10^{10}		–	–	–	–	–

15	10^{15}					–	–
20							–

* Zur Erinnerung: Die Zahlen stellen die *maximale* Zeit dar, d. h. es wird davon ausgegangen, dass das richtige Passwort (Schlüssel) erst mit der *letzten* Kombinationsmöglichkeit gefunden wird (realistisch: im Durchschnitt nach der Hälfte der Zeit, s. o.).

(2) Zeichenraum: Kleinbuchstaben (Zeichenanzahl: 26)

Passwort-länge	**Passwort-anzahl**	**Anzahl der maximal* benötigten Zeit**					
		in Sekunden	in Minuten	in Stunden	in Tagen	in Jahren	in Jahr-hunderten
1			–	–	–	–	–
5	$1{,}2 \cdot 10^{7}$		–	–	–	–	–
10					–	–	–
15	$1{,}7 \cdot 10^{21}$	$1{,}7 \cdot 10^{11}$					
20							

(3) Zeichenraum: Klein- und Großbuchstaben (Zeichenanzahl: 52)

Passwort-länge	**Passwort-anzahl**	**Anzahl der maximal* benötigten Zeit**					
		in Sekunden	in Minuten	in Stunden	in Tagen	in Jahren	in Jahr-hunderten
1			–	–	–	–	–
5			–	–	–	–	–
10						–	–
15	$5{,}5 \cdot 10^{25}$	$5{,}5 \cdot 10^{14}$					
20							

Siehe nächste Seite.

(4) Zeichenraum: Ziffern, Klein- und Großbuchstaben (Zeichenanzahl: 62)

Passwort-länge	Passwort-anzahl	Anzahl der maximal* benötigten Zeit					
		in Sekunden	in Minuten	in Stunden	in Tagen	in Jahren	in Jahr-hunderten
1			–	–	–	–	–
5			–	–	–	–	–
10							–
15	$7{,}7 \cdot 10^{26}$	$7{,}7 \cdot 10^{15}$					
20							

8. Brute-Force-Rechenzeit II: Auswertung

Vergleichen und bewerten wir nun die Ergebnisse (Tipp: Es ist lohnenswert, sich zuvor nochmal die Größenordnungen aus Kapitel 4 zu vergegenwärtigen). Wir wollen dies aus Sicht der Passwortsicherheit tun, d. h. eine Brute-Force-Rechenzeit ist dann „gut“, wenn sie möglichst lange dauert – und dementsprechend „schlecht“, wenn sie von nur kurzer Dauer ist. Das führt uns unweigerlich zu der Frage, welche Brute-Force-Rechenzeit als *minimale* Dauer ausreichend ist, um von „gut“ bzw. von einem „sicheren“ Passwort zu sprechen.

a) Folgendes **Wort-Case-Szenario** [≈ schlechtester oder ungünstigster Fall, ähnlich zu „GAU“]: Ein mit einem Passwort verschlüsselter Datenträger von dir mit hochsensiblen persönlichen und beruflichen Daten ist von einer unbekannten Person mit böswilliger Absicht gestohlen worden. Sollte der Dieb deinen Datenträger entschlüsselt bekommen, droht die Preisgabe der Daten im Internet, was eine Katastrophe für dich bedeuten würde, mit schwerwiegenden und langfristigen Folgen.

Welche *minimale* Brute-Force-Rechenzeit zum Knacken des Datenträger-Passworts würdest du als „sicher“ erachten? Begründe.

_____

_____

_____

Eine allgemeine Antwort auf diese Frage zu finden ist schwierig. Eine Möglichkeit wäre,

dass eine Brute-Force-Rechenzeit dann als ausreichend betrachtet werden kann, wenn sie **länger als die durchschnittliche Lebensdauer eines Menschen** ist. Damit der Zufall zum Auffinden des korrekten Passworts so gering wie möglich gehalten wird, könnte man diese Zeitspanne mit dem **„Sicherheitsfaktor“ 1.000** multiplizieren.

Beispiel: Sollte eine Brute-Force-Rechenzeit zu einem Passwort 420 Jahre betragen, müsste dieser Wert durch 1.000 dividiert werden. Das Ergebnis wäre mit 0,42 Jahre, also weniger als einem halben Jahr, zu unsicher.

- Zurück zur Auswertung: Betrachten wir zunächst die **erste Tabelle**, die einen Zeichenraum von 10 Ziffern beschreibt. Passwörter mit einer Länge bis 10 Zeichen sind völlig

  indiskutabel, da sie in weniger als einer _____ geknackt werden. Auch bei

  15 Zeichen Länge benötigt man nur knapp _____ Stunden. Mit 20 Zeichen

  erreicht man immerhin eine Rechenzeit in Höhe von rund 32 _____, was aber nach wie vor nicht ausreicht.

  Erstes Zwischenfazit: Ein Passwort sollte nicht nur aus Ziffern bestehen.

- Kommen wir zur **zweiten Tabelle** mit einem Zeichenraum von 26 Kleinbuchstaben.

  Passwörter der Länge 10 können in weniger als einer _____ berechnet werden, was nicht akzeptabel ist. Ein großer Sprung ergibt sich bei einer Zeichenlänge

  von 15 in Form von ungefähr 539 _____. Dividiert man diesen Wert jedoch durch den oben angenommen „Sicherheitsfaktor“ 1.000, bleibt nur etwas mehr als ein

  _____ Jahr übrig – zu wenig. Bei 20 Zeichen Länge wird es interessant:

  Ohne „Sicherheitsfaktor“ ergeben sich nun rund 6,3 _____ Jahre, die divi-

  diert durch 1.000 immer noch erst in maximal 6,3 _____ Jahren berechnet werden können.

  Zweites Zwischenfazit: Im Zeichenraum der Kleinbuchstaben ist ein 15-Zeichen-Passwort zu kurz. Dem entgegen zeigt sich bei einer Länge von 20 Zeichen in unseren Tabellen erstmals eine hinreichend starke Passwortsicherheit.

- Weiter geht es mit der **dritten Tabelle** und einem Zeichenraum von 52 Klein- und Großbuchstaben. Auch hier werden Passwörter mit einer Länge von bis zu 10 Zeichen

  schnell „geknackt“, nämlich in höchstens ca. _____ Tagen. Bei einer Zei-

  chenlänge von 15 sieht es anders aus: Rund 17 _____ Jahre vergehen bis zur Berechnung aller Passwörter, was nach Abzug unseres „Sicherheitsfaktors“ immer

noch etwa _____ Jahren entspricht. Eindeutig ist die Sache auch bei 20 Zeichen: Mit einer Zehnerpotenz von 10^{15} bewegen wir uns im Bereich von unfassbaren rund 6,7 _____ Jahren, wobei selbst eine Division durch 1.000 keine Rolle mehr spielt.

Drittes Zwischenfazit: Im Vergleich zu nur Kleinbuchstaben sind bei Klein- und Großbuchstaben bereits Passwörter mit einer Länge von 15 Zeichen ausreichend sicher. Noch gravierender ist der Unterschied bei einer Passwortlänge von 20 Zeichen: Hier ergibt sich sogar eine Millionenfach längere Rechenzeit.

- Betrachten wir zuletzt **Tabelle vier** mit insgesamt 62 Ziffern, Klein- und Großbuchstaben, welche eine Art Zusammenführung der ersten und dritten Tabelle darstellt. Während die Passwortlängen 1 und 5 auch hier offensichtlich indiskutabel sind, erreicht man für 10 Zeichen erstmals die Größenordnung _____ – was dennoch unsicher ist. Dafür erfahren wir wie in Tabelle drei einen enormen Sprung bei Passwörtern bestehend aus 15 Zeichen: knapp 240 _____ Jahre werden zum „Knacken“ benötigt. Unter Berücksichtigung des „Sicherheitsfaktors“ entspricht dies noch rund _____ Jahre, was vollkommen ausreichend ist. Extreme Werte erzielen wir wieder mit 20 Zeichen langen Passwörtern: deutlich über 200 _____ Jahre – eine 2 mit 17 Nullen! –, was auch abzüglich des Sicherheitsfaktors weit mehr als genug ist.

  Viertes Zwischenfazit: Natürlich erreicht die Passwortsicherheit in diesem untersuchten Zeichenraum die höchste Qualität. Im Vergleich zu nur Klein- und Großbuchstaben ändert sich hier nichts wesentlich – mit 15 Zeichen Länge ist man auf der sicheren Seite.

9. Schlussbetrachtung (Fazit)

Fassen wir zusammen:

- Ausgehend von unseren Berechnungstabellen sind Passwörter im Zeichenraum der 26 Kleinbuchstaben bei einer Länge von 20 Zeichen sicher (ob dies womöglich auch schon für 17 oder 18 Zeichen gilt („Zwischenlängen“) haben wir nicht betrachtet).

- Bei 52 Klein- und Großbuchstaben befinden wir uns mit einer Länge von bereits 15 Zeichen auf der sicheren Seite (Zehntausende Jahre, inkl. Berücksichtigung des „Sicherheitsfaktors“).

- Bei 62 Ziffern, Klein- und Großbuchstaben ist eine Passwortlänge von 10 Zeichen nach wie vor unsicher. Bei Länge 15 sind die Passwörter aber wieder sicher (Hunderttausende Jahre, inkl. Berücksichtigung des „Sicherheitsfaktors“).

Bei genauerer Analyse der *Tabellen im Vergleich zueinander* fällt etwas Interessantes auf:

- Wenn wir *Passwörter der Länge 10* betrachten, macht es keinen signifikanten Unterschied, ob wir uns im Zeichenraum von 10 Ziffern oder 62 Ziffern, Klein- und Großbuchstaben bewegen. In Zehnerpotenzen gesprochen drückt sich dies so aus, dass im 10er-Zeichenraum $0{,}1 = 10^{-1}$ Sekunden und im 62er-Zeichenraum $8{,}4 \cdot 10^6 \approx 10^6$ Sekunden benötigt werden. Das ist zwar ein Unterschied von 10^7 Sekunden (Faktor 10 Millionen), dennoch führt die (mehr als) Versechsfachung des Zeichenraums zu keiner ausreichenden Sicherheit.

- Wenn wir jedoch *Passwörter der Länge 20* betrachten, sieht die Sache anders aus: Im Zeichenraum von 10 Ziffern erreichen wir mit rund 10^9 Sekunden zwar nach wie vor keine ausreichende Sicherheit. Im Zeichenraum der 62 Ziffern, Klein- und Großbuchstaben sind es hingegen rund $7 \cdot 10^{24} \approx 10^{24}$ Sekunden. Das ist ein Unterschied von 10^{15} Sekunden (Faktor 1 Billiarde) – und damit wesentlich mehr als die Versechsfachung des Zeichenraums bei Passwortlänge 10.

- Mehr noch: Ein Passwort der Länge 20 im Zeichenraum 10 ist mit 10^9 Sekunden „sicherer“ als ein Passwort der Länge 10 (Hälfte der Länge) im Zeichenraum 62 (mehr als Versechsfachung des Zeichenraums), für dessen Berechnung nur 10^6 Sekunden benötigt werden.

Folgende Kurzübersicht stellt dies anschaulich dar:

Passwort-länge	**Anzahl der maximal benötigten Zeit (in Sek.)**		**Differenz (in Sek.)**
	Zeichenanzahl/-raum: **10**	Zeichenanzahl/-raum: **62**	
10	10^{-1}	10^6	10^7
20	10^9	10^{24}	10^{15}

Gemäß unserer Berechnungsformel zur Passwortanzahl aus Kapitel 2a) gibt es zwei „Stellschrauben“ für die Passwortsicherheit: (1) die Zeichenanzahl (bzw. der Zeichenraum) und (2) die Passwortlänge. In unseren Tabellen zeigt die Verdopplung der Passwortlänge einen wesentlich besseren Sicherheitszugewinn als die (mehr als) Versechsfachung des Zeichenraums. Somit spielt die Passwortlänge offensichtlich eine wichtigere Rolle als die Zeichenanzahl. Dies ist bereits in der Formel ersichtlich: Die Passwortlänge ist die Hochzahl. Ihre Erhöhung sorgt mathematisch gesehen für den größeren Zuwachs als eine Erhöhung der Basis (hier: die Zeichenanzahl) (vgl. Kap. 5).

Für die Passwortsicherheit gilt also in erster Linie: „Es kommt auf die Länge an!“

Wer ein Passwort bestehend aus 15 bis 20 Zeichen im Zeichenraum der 62 Ziffern, Klein- und Großbuchstaben wählt, kann davon ausgehen, dass zum „Knacken“ Millionen bis Billiarden Jahre vergehen. Das bedeutet **Rechnen bis zur nächsten Eiszeit** – und darüber hinaus...

Dabei unberücksichtigt ist eine weitere Forderung an die Wahl eines sicheren Passworts: die Erweiterung des Zeichenraums um **Sonderzeichen**. Diese haben wir hier deswegen unberücksichtigt gelassen, weil nicht klar definiert ist, wie groß genau der (Standard-)Zeichenraum der Sonderzeichen ist. Welche unvorstellbaren Größenordnungen sich hier ergeben, wenn man von nur 30 weiteren Zeichen (z. B. ! " § $ % & / ( ) = ? }] [{ \ + * ~ ' # - _ . : , ; < > und |) ausgeht, kann sich jetzt jeder selbst ausrechnen. ☺

Ein **weiterer Aspekt** kommt für Passwort-Cracker erschwerend hinzu: Gemäß der Tabelle in 5a) müssen zum „Knacken“ eines Passworts der Länge 15 im Zeichenraum der 26 Kleinbuchstaben $1{,}7 \cdot 10^{21} \approx 1{,}7$ Trilliarden Passwörter ausprobiert werden. Dieser Wert beschreibt aber nur die Anzahl der Passwörter mit der exakten Länge 15. Nur wenn der Angreifer weiß, dass das gesuchte Passwort exakt 15 Zeichen umfasst, müsste er (bereits) 1,7 Trilliarden Passwörter ausprobieren. **Meist kennt der Angreifer die Passwortlänge jedoch *nicht***, weil diese i. d. R. variabel vom Nutzer ausgewählt werden kann. Daher muss der Angreifer auch alle Passwörter bestehend aus Kleinbuchstaben der Länge 14 (rund $6{,}5 \cdot 10^{19}$), der Länge 13 (rund $2{,}5 \cdot 10^{18}$), der Länge 12 ($9{,}5 \cdot 10^{16}$) usw. ausprobieren! Dabei summiert sich eine ungeheure Gesamtzahl auf, dem entsprechend auch eine benötigte Zeitdauer.

Wie wir zum Abschluss (hoffentlich) sehen, ist die Theorie zur Passwortsicherheit kein „Voodoo-Zauberwerk“, das nur Experten durchblicken. Mit einfacher Schulmathematik kann man zeigen, dass Passwörter, die aus Ziffern, Klein- und Großbuchstaben sowie Sonderzeichen bestehen und eine Länge von 15 Zeichen oder mehr haben, so **sicher sind**, dass selbst die besten Supercomputer der Welt diese nicht „Knacken“ können – das gilt auch für die futuristischen Science-Fiction-Computer aus Kino und Romanen. Zwar führt die Brute-Force-Methode theoretisch immer zum Erfolg, doch benötigt dies (im entscheidenden praktischen Maße!) eine gewisse Zeit. Man spricht in der Informatik daher auch von einem **theoretisch lösbaren, aber praktisch unlösbaren Problem**. Klar, ein unglaublicher Zufallstreffer, dass bereits das z. B. 42. von 100.000.000.000.000.000.000.000.000.000.000.000 = 10^{35} möglichen Passwörtern (Passwortlänge 20 im Zeichenraum der 62 Ziffern, Klein- und Großbuchstaben, siehe Tabelle 4 oben) das richtige ist, ist prinzipiell möglich. Aber wie wahrscheinlich?