



eine Anleitung um
Geocaching Mysteries lösen
zu lernen

aus Ninas Schmierblo(g)ck
Autor: Nina Geiger
Copyright: © 2014 Nina Geiger

www.justchaos.de
<http://www.justchaos.de/blog/>

Version 1.0
Stand: 23.12.2014

Weitergabe dieses PDF in unveränderter Form ist erwünscht und gern gesehen!

Inhaltsverzeichnis

1. - Einleitung.....	3
2. - Analysen von Listing, Rätsel und Codes	4
2.1 Listinganalyse.....	4
2.2 Geocachingspezifische Kryptoanalyse	7
3. Zahlen, Zahlen, Zahlen,.....	10
3.1: Einleitung, Fremdsprachen und das Zahlensystem der Maya.....	10
3.2 Binär-, Oktal- und Hexadezimalzahlen	11
3.3 Binärcodes	13
4 Sprache und Schrift.....	17
4.1 - Sprache und Schrift (Teil 1)	17
4.2 - Sprache und Schrift (Teil 2)	19
5. - Dateianalyse: Bilder - Musik - Video	22
5.1 Bilderanalyse - Teil 1: Einleitung und optisches	22
5.2 Bilderanalyse - Teil 2: Technische Bilderverstecke.....	24
5.3 Musikdateianalyse.....	30
6. Verschlüsselungen.....	32
6.1.1 Monoalphabetische Substitution.....	32
6.1.2 - Verschlüsselungen - Geheimtexte manuell entschlüsseln.....	35
6.2 -Polyalphabetische Verschlüsselung	39
7 Logikrätsel	44
8 - Weiteres	45
8.1 - Barcodes	45
8.2 - Noten und Notenverschlüsselungen.....	50
8.3 – Farben	52
8.4 – Steganographie	54
8.5 Esoterische Programmiersprachen	55
9 - Tools und Links	59
9.1 - Links und Codelisten	59
9.2 - Geocaching-Tools für Zuhause und unterwegs	63
Anhang: Verschlüsselungstabellen.....	65

1. - Einleitung

Einleitung - Warum schreib ich dies?

Ich mag [Geocachen](#). Eines der Dinge, die mir hierbei am besten gefallen, sind die Rätsel. Seien es die Mystery-Caches, für die man schon Zuhause Rätsel eine Koordinate errätseln muss oder Station eines Multis, also eines Mehrstationencaches, bei denen ein Rätsel am Weitergehen hindert. Manche dieser Rätsel folgen bestimmten wiederkehrenden Mustern. Vieles ist zwar unterschiedlich verpackt, ähnelt sich in den Grundzügen dann irgendwann aber doch. Seien es die Verschlüsselungen (z.B. [binär oder Morse](#)) oder die Techniken, in Bildern oder Musikdateien weitere Informationen zu verbergen.

Jeder, der einen derartigen Mystery selber lösen will, muss erstmal herausfinden wie so etwas funktioniert und was überhaupt alles möglich sein kann. Und Jeder, der so etwas schon einmal gehört, gelesen oder angewandt hat, muss sich in dem betreffenden Moment auch wieder daran erinnern.

Für derartige Situationen und Personen hab ich diese Blogrubrik angefangen und werde sie im Lauf der Zeit immer wieder erweitern.

Ihr findet diese Blog-Idee verwerflich? Ein Mysterylösungsblog für all die Abschreiber, Mysterykoordinatentauscher, Telefonjokernutzer und ähnliches Gesocks? Ihr müsst ja noch immer alles selber lösen und findet dieses Geschriebene daher furchtbar unfair? Nein, das ist es gar nicht, fast gar nicht. Alles was hier so steht, findet sich auch im Netz auf diversen Foren, Blogs, bei Twitter, Facebook und über diverse Suchmaschinen, und wird spätestens auf Events wie die berühmten Panini-Bilder getauscht. Übrigens genau wie man häufig auch detaillierte Lösungsansätze zu speziellen Caches findet. Oder auch gleich die Final-Koordinaten.

Ich werde hier keine einzelnen Caches vorstellen und anhand derer die Lösungswege beschreiben. Und auch mit dem Wissen und der Nutzung der hier erklärten Ideen und Strategien ist das Lösen eines Mysterys noch immer Hand- und vor allem Kopfarbeit. Man muss die Mystery-Idee selber verstehen und umsetzen, die Rätsel sind häufig ineinander verschachtelt, man muss die Koordinaten erkennen, wenn sie vor einem liegen. Und man muss, wie bei allem im Leben, viel üben. Eine Menge abstruser Gedanken verfolgen, wobei die Lösung oft viel einfacher ist, als die eigenen Gedanken. Viele Mysteries lassen sich sowieso nicht in ein Lösungsschema pressen, jeder einzelne Cacheowner hat andere, teils wirre, teils geniale, oft einmalige Ideen, auf welche Art man Koordinaten, Hinweise oder Text verstecken kann. Und bei vielen Mysteries ist der Lösungsweg sowieso klar und es erfordert nur etwas Mühe, ihn zu erarbeiten. Durch Webrecherche, durch das zeitraubende Ausfüllen bestimmter Matrizen, durch das Lösen von Logikrätseln, für die es kein Online-Lösetool im großen, bösen Internet zu geben scheint. Oder durch das Finden eines anderen Caches oder Travelbugs.

Somit darf nun jeder der hoffte, nach der Lektüre dieses Blogs jeden Mystery mit wenigen Klicks von der Karte putzen zu können, enttäuscht die Webseite schließen. Nur weil man weiß wie ein Sudoku funktioniert, ist man noch lange nicht in der Lage, ein solches auch zu lösen. Von einem schnellen Lösen noch gar nicht geschrieben.

In dem Sinne wünsche ich euch viel Spaß und hoffentlich geistreiche Erkenntnisse auf meinem Blog. - Nina

2. - Analysen von Listing, Rätsel und Codes

2.1 Listinganalyse

Womit fängt man an, wenn man einen Mystery lösen möchte? Natürlich mit dem Listing und seiner Analyse. Daher hier nun kurz, wie und womit man beginnen kann, wenn der Ansatz des Rätsels nicht sofort erkennbar ist.

1. Der **Titel**. Klingt trivial, aber sehr oft ist dieser ein mehr als nur dezenter Hinweis auf die Art, wie man an die Lösung kommen kann. Manchmal direkt, manchmal verschlüsselt (dazu später mehr), manchmal als Anagramm (es gibt viele, mal mehr und mal weniger nützliche Anagramm-Generatoren im Internet). Manchmal muss man den Titel erst mal einer Suchmaschine vor die Füße werfen, um seinen tieferen Sinn zu verstehen.

2. Der **Listingtext**. Wenn es überhaupt einen gibt. Wie oft hab ich erst beim zweiten, dritten oder vierzehnten Mal lesen die hier schwarz auf weiß aufgeschriebenen Hinweise überhaupt erst als solche wahrgenommen. Häufig ist aber auch mit weißer Schrift auf weißem Grund Text zwischen dem sichtbaren Listingtext verborgen. Dies kann man sichtbar machen, in dem man den gesamten Text markiert (meist mit strg-A). Detaillierte Worte zur Analyse von Text folgen im Kapitel "Sprache und Schrift".

3. **Bilder**. Gibt es welche im Listing? Oder der Image-Gallery? Gibt es ein Hintergrundbild? Sind hinter den sichtbaren Bildern weitere/andere verlinkt (zeigt der Browser vielleicht unten in der Info-Zeile an, sonst sieht man es im Quelltext, siehe weiter unten in diesem Blogbeitrag). Ist der Bildname ein sprechender, oder nur das Zahlen-Buchstaben-Kuddelmuddel, was z.B. geocaching.com benutzt, wenn man dort die Bilder hoch lädt? Unterscheiden sich optisch ähnliche Bilder? Liegen die Bilder auf einem anderen Webpace? Kann man hier über den Webbrowser in das Verzeichnis oder ein weiter oben liegendes gucken? Hätte jemand zum Beispiel dieses Bild im Listing verwendet:

http://www.justchaos.de/img/7grad/IMG_2392.JPG

würde ich schauen, ob ich unter

<http://www.justchaos.de/img/7grad/>

oder <http://www.justchaos.de/img/>

oder eben <http://www.justchaos.de/>

etwas hilfreiches zu sehen bekomme. Hilft dies alles nicht, müssen die Bilder möglicherweise näher analysiert werden, hierzu mehr im Kapitel "Bilderanalyse".

Sehr hilfreich sind Bildersuchmaschinen wie z.B. <http://tineye.com/> oder die Google-Bildersuche. Werft diesen mal die im Cache verlinkten Bilder vor und schaut, ob sie diese kennen und mit welchen Schlagworten oder Informationen diese verbunden sind.

4. **Hint**. Gibt es einen? Nutzt er auch? Oder ist er nur ein Findehelfer für draußen? Bei der Gelegenheit: ich würde mir wünschen, daß alle Hints auch wirklich welche wären. Dies ist kein Zwangsfeld für Fülltext, man bekommt den Cache auch freigeschaltet, wenn man keinen angibt. Der leider viel zu häufig benutzte Blödsinn ("hier ist nix", "wer das liest ist doof" und ähnliches) ist

nervig, sinnlos und schon seit etwa einer Millionen Mal Verwendung auch nicht mal mehr ein bisschen witzig.

5. Der **Quelltext**. Selbst ohne tief greifende html-Kenntnisse kann man hier leicht verborgene Hinweise, versteckte Links, unbeachtete Tooltips (der Text, der erschienen kann, wenn man mit der Maus über einen Link fährt), oder das Ziel der nervigen kleinen Mouseovers (wenn man mit der Maus über ein Bild fahren muss um an einem bestimmten, winzigen Punkt einen Link zu bekommen) erkennen.

Am Beispiel des Firefox und einem geocaching.com-Listing: Markiere den Text zwischen den Notes und dem Hint, klicke mit der rechten Maustaste darauf und wähle "View Selection Source" ("Auswahlquelltext anzeigen"). Es öffnet sich ein weiteres Fenster, in dem Quelltext blau markiert ist. Findet sich hier etwas ungewöhnliches? Links zum Beispiel beginnen mit `< a href="` und enthalten den physikalischen Link sowie den Linktext. Sehen beide aus, wie eine URL, unterscheiden sich aber, sollte man mal beide besuchen.

Bilder beginnen mit `< img src="`). Innerhalb dieser spitzen Klammern könnte hinter `"alt = "` versteckter Text " etwas verborgen worden sein, was aber, je nach Browser, auch schon im Listing sichtbar sein könnte.

Text, der nicht auf der Webseite angezeigt werden soll, kann wie folgt auskommentiert werden `< ! -- hier ist der "Geheimtext" -- ! > .`

Ich musste bei den HTML-Code-Beispielen leider zusätzliche Leerzeichen einfügen, sonst hätte das Blog diese als html-Code interpretiert und nicht hier als Text dargestellt.

6. Die **Waypoints**. Wer guckt schon auf die Waypoints? Diese können durchaus interessant sein. Haben sie einen Beschreibungstext? Liegen die Koordinaten in der Nähe des Myterypunktes? Wenn sie komplett woanders sind, könnte es vielleicht schon reichen, nur die Minuten der Koordinaten (die letzten 3 Ziffern von Nord und Ost) mit denen des Mystery-Fragezeichens zu vertauschen? Ich habe auch schon Caches gesehen, bei denen es eine wahre Flut von Waypoints gab, die dann, geschickt miteinander verbunden, ein X auf der Karte gemalt haben oder einen Text bzw. Ziffern auf einer Karte ergaben.

7. Der **GC-Code**. Bei Geocaching.com haben die Caches eindeutige Namen, die mit GC anfangen. Das, was dahinter folgt, ist zwar vom Cacheowner nicht zu beeinflussen, aber nutzen kann er es natürlich trotzdem. Möglicherweise als Schlüssel um eine andere Information im Listing zu verbergen? Oder simpel als Passwort für ein verschlüsseltes Archiv oder eine Webseite?

8. Das **Legedatum**: Stimmt es ungefähr mit dem Veröffentlichungsdatum überein? Wenn nicht, ist es ein recht deutlicher Hinweis, daß hier eine Information verborgen worden ist. Aber auch sonst kann man über das Datum etwas verschlüsseln, sei es als Passwort, oder weil man z.B. die Zahl des Tages (Monat, Jahr) nehmen, und mit der Ziffer die Worte (Absatz, Zeile, Wort, Buchstabe, â€¦) im Text abzählen muss.

9. Der **Geochecker**: Gibt es einen irgend gearteten Geochecker? Dann versäumt nicht, diesen mal anzuklicken. Möglicherweise verbirgt sich hier noch ein weiterer Hinweis.

10. **Trackables:** Wird zum Finden des Caches vielleicht ein Travelbug/Coin benötigt? Schaut mal in die Liste der "View past Trackables" - ist hier als erstes vom Owner ein TB/Coin hinterlegt und gleich wieder herausgenommen worden? Stimmt die Liste der Cachefinder "zufällig" mit denen überein, die diesen Trackable "discover" haben? Dann beginnt hier die Jagd ;)

11. **Codelexikon:** Sind Informationen im Listing, ihr könnt sie aber nicht zuordnen? Dann legt mal ein Codelexikon (z.B. von www.geocaching-franken.de) daneben und schaut, an welcher Stelle euch vielleicht ein Licht aufgeht.

12. **Related Web Page:** Seit irgendeinem Seitenupdate ist es möglich, einen externen Link mit dem Listing zu verknüpfen. Der wird oben im Kopf des Listings unterhalb der D-T-Sterne mit angegeben (und von mir sehr gern übersehen...).

13. **Weiteres:** Es nutzt alles nix? Dann werft einer Suchmaschine den Cache und all seine Informationen vor die Füße. Vergesst nicht, dem Owner etwas hinterher zu stalken, auf seine Profilseite zu gucken, ggf. weiteren Links zu folgen (Mailadresse? Homepage? Facebook o.äh.). In archivierte Caches, vielleicht hat er noch eine Notiz dort hinterlassen. Sucht nach einem Geocaching-Fake-Account oder TravelBug, dessen Name sich irgendwie aus dem Listing ergibt. Passen eure Gedankengänge überhaupt zur Difficult-Wertung? Allzu häufig übersieht man ja das Naheliegendste und stochert unnötig in der Tiefe.

Und als letzter Tipp auf dieser Seite: fehlt vielleicht der klassische Satz, daß die Mysterykoordinate frei gewählt worden ist und nichts zur Sache tut? Dann liegt die gesuchte Koordinate vielleicht genau da, wo auch das Fragezeichen platziert worden ist?

2.2 Geocachingspezifische Kryptoanalyse

Oder auch: Und was mache ich jetzt hiermit???

Ihr hab das Rätsel gefunden und steht nun verwirrt vor einem Berg aus Zahlen, Buchstaben, Bildern, Zeichen oder ähnlichem?

Dagegen hilft vielleicht dieser Versuch strukturiert darzustellen, was das Gefundene sein könnte. Bitte nehmt es mir nicht übel, wenn ich nicht jeden Begriff mit einem Link hinterlege. Oft gibt es gar nicht DIE Seite, die weiterhilft. Aber meistens hilft zumindest das Stichwort.

Als erstes solltet ihr analysieren, was ihr habt. Buchstaben? Zahlen? Nur bestimmte? Wie viele? Gibt es Gruppierungen? Eine Zweiteilung für Nord und Ost? Entspricht die Anzahl von Ziffern, Zahlen oder Gruppen der üblichen Anzahl von Zeichen einer Koordinate (13-21 Zeichen, je nach Schreibweise)? Könnten es ausgeschriebene Zahlworte ergeben (4-6 Zeichen lang)? Könnte der Anfang z.B. N52 sein? Bzw. N und E, wenn sich die beiden Koordinatenteile eindeutig identifizieren lassen. Suchst Du überhaupt Koordinaten? Oder eine Peilung? Etwas ganz anderes?

Funktioniert "reverse Engineering"? Zu gucken, welche Koordinate aus dem Verschlüsselten in etwa herauskommen müsste und mit dem Code zu vergleichen.

Hilft alles noch nicht? Dann einmal gegenchecken:

a) Chiffriert mit Passwort:

Ihr habt einen offensichtlich chiffrierten Text und vielleicht sogar ein Passwort, wisst aber nicht, womit es entschlüsselt werden kann? Hier eine Liste von gebräuchlichen Chiffrierungen, die ein Passwort benutzen:

- ADFG(V)X (Chiffre enthält nur genau diese Buchstaben bzw. 5 bzw. 6 unterschiedliche)
- Alberti (Zwei Schlüsselworte!)
- AMSCO (Schlüssel besteht nur aus Ziffern!)
- Autokey
- Beaufort-Chiffre
- Bifid-Chiffre
- Four-Square-Chiffre (Zwei Schlüssel!)
- Gronsfeld-Chiffre (Schlüssel besteht aus Ziffern!)
- Kamasutra
- Larrabee-Chiffre
- Polybius (ergibt zweistellige Zahlenketten)
- Nihilisten (Erweiterung von Polybius)
- Playfair
- Porta-Chiffre
- Transposition
- Vigenere (wird am Häufigsten verwendet)

b) binär

Ihr habt etwas gefunden, was zwei (manchmal 3 fürs Leerzeichen) verschiedene Zustände hat? 0 und 1. Da oder nicht da. Weiß oder schwarz. Zwei verschiedene Bilder. Farben. Töne, etc. Meine Seite über [Binärcodes](#) hilft euch hoffentlich weiter.

c) 7-Segment-Anzeige

Ihr habt Zahlen von 1-7 oder Buchstaben von a bis g? Oder doch zumindest so viele Zahlen oder Buchstaben, also sieben verschiedene Zustände? Diese sind in Blöcke aufgeteilt, wobei kein Block länger als sieben Zeichen ist und kein Zeichen im Block doppelt vorkommt? Die Minimalgröße eines Blocks beträgt zwei verschiedene Zeichen? Dann ist es die 7-Segmentanzeige, wie z.B. auf digitalen Uhren. Die Zahlen von null bis neun als 7-Segmentblock: abcdef bc abdeg abcdg bcfg acdfg acdefg abc (oder abcf) abcdefg abcdfg

d) Rechts-Links-Oben-Unten

Ähnlich wie 7-Segment malt diese Variante letztlich auch das gleiche. Also Zeichen. Allerdings hier von einem Ausgangspunkt aus in die jeweilige Richtung rechts, links, oben oder unten. Natürlich könnten die vier Buchstaben RLOU anders heißen. Aber es wären vier verschiedene. Und die Mindestanzahl eines Blocks ist wieder zwei! Die maximale etwa acht (+/- 1, je nach Zeichenschreibart). Wiederholungen sind möglich, aber selten.

e) das Periodensystem

Ihr habt Zahlen bis 118? Oder Buchstaben, die keinem Wort und keiner üblichen Verschlüsselung anzugehören scheinen? Möglicherweise vor allem Buchstaben wie h, he, li, be, b, c, n, o, f? Oder Blöcke von maximal zwei Buchstaben? Dann schaut mal auf das [Periodensystem!](#)

f) **Code-Sonne**

Ihr habt Dreiergruppen von maximal vier verschiedenen Buchstaben? Dann schaut euch mal die [genetische Code-Sonne](#) an.

g) **Zahlen und Zahlensysteme**

Ihr habt Zahlen von 0-9 und Buchstaben bis f? Dann ist es das Hexadezimalsystem. Die Buchstaben gehen weiter als bis f? Oder die Zahlen nicht mal bis 9? Dann könnte es ein anderes Zahlensystem sein. Üblich ist binär (0 und 1), oktal (bis 8), hexadezimal (16) und das Duodezimalsystem (12). Aber alles andere ist ebenfalls möglich. Einfach mal mit einem [Umrechner](#) spielen. So lässt sich sogar ganzer Text in Zahlen umwandeln. Das Wort Nina in Basis 32 ergibt z.B. die Dezimalzahl: 1097254.

- Habt ihr einen Mix aus großen und kleinen Buchstaben, Zahlen und +/-Zeichen, also: A-Z, a-z, 0-9, + / , so dürfte ein Base64-Umrechner weiterhelfen.
- Ihr habt Zahlen von 1-26? Dann sind es vermutlich einfach nur die Buchstabenwerte des Alphabets.
- Habt Ihr Zahlen so etwa ab ca. 50 bis 120? Dann schaut mal auf eine Ascii-Tabelle. N wie Nord ist übrigens 78, E für Ost 69.
- Ihr habt zweistellige Zahlen, die ausschließlich aus den Ziffern 1-5 bestehen? Polybius oder Klopfcode?
- Die Zahlen sind von 2-9, ggf. 0 und kommen oft doppelt und dreifach vor? Vanity bzw. Handytastatur!
- Apropos Handy: Ihr habt eine Zahlenreihe, die einfach keine Koordinate ergeben will? Vielleicht ist es eine Telefonnummer? Oder ein anderes [Koordinatensystem](#)?

h) **Buchstaben**

- Ihr hab die Buchstaben I, V, X, L, C, D vor euch liegen? Das sind Römische Zahlen.
- Ihr hab die Buchstaben m, p und f gefunden? Dann ist es Kenny-Speak.
- Jede Menge As und Bs? Siehe [Binär](#), das ist die Bacon-Chiffre.
- Buchstabensuppe? Mit relativ normaler Buchstabenhäufigkeit? Ist es ein Anagramm? Oder eine Buchstabenverschiebung wie der Lattenzaun oder [Transpositionsverfahren](#)?
- nur 5 bzw. 6 verschiedene Buchstaben? ADFG(V)X!

i) **Weiteres**

- Zahlen und Buchstaben gemixt, die auf nix obiges zu passen scheinen? Vielleicht sind es GC-Codes oder Nummern von Trackables?
- Zahlen und Buchstaben gemixt, immer eine Zahl und ein Buchstabe? Vielleicht eine Matrix? Füllt mal ein Tabellenkalkulationsblatt an den jeweiligen Positionen mit einem X.

Du hast immer noch keine Ahnung? Wirf es einer Suchmaschine vor!

Und lies hier den Rest vom Blog ;).

(to be continued...)

3. Zahlen, Zahlen, Zahlen,...

3.1 : Einleitung, Fremdsprachen und das Zahlensystem der Maya

Zahlen ist in aller Regel ja das, wonach wir suchen. Koordinaten. Ein Pärchen, Nord und Ost. Manchmal auch eine Peilung, wobei man hier dann noch einen Punkt braucht, von dem aus gepeilt werden muss. Gern die Koordinate, an der das Listing platziert wurde.

Ist eine Koordinate versteckt, dann entweder in Gänze und der bei uns üblichen Grad-Dezimalminutenschreibweise (z.B. N 52° 12.345 E 009° 59.876). Oder als Teilmenge davon, also nur die letzten 3 Ziffern von Nord und Ost (wobei dann, wenn es nicht anders angegeben worden ist, die Listingkoordinate die fehlenden Koordinatenteile beinhaltet), oder gern auch die letzten 5 Ziffern.

Wer die Rätselnden verwirren möchte, nutzt auch gerne andere Koordinatenformate, z.B. die Dezimaldarstellung. Wodurch aus dem obigen Beispiel nun diese Schreibweise werden würde: 52.20575 9.997933. In Grad, Minuten und Sekunden ausgedrückt: N52° 12' 20.7" E9° 59' 52.56" . Taugt prima zur Verwirrung, wenn der geübte Mysterylöser seinen Blickwinkel zu sehr auf die vermeintliche Koordinate in üblicher Schreibweise eingestellt hat. Umrechner zu den Koordinatenformaten hierzu finden sich im Internet und in nahezu jeder größeren Geocaching-App für unterwegs.

Soviel zu dem, was wir suchen. Jetzt dazu, wie wir es finden können. Die Kapitel hierzu werden sich zwangsläufig ständig überschneiden, also bitte nicht wundern. Somit beginnt dieser Bereich mit einem Thema, was hervorragend auch in den Bereich der "Sprache und Schrift" gepasst hätte:

Zahlen lassen sich schön verstecken, in dem man sie in fremden Sprachen darstellt. One, two, three werden wir ja alle noch frei übersetzt bekommen, aber was ist mit aon, dā , trā- oder mā't, hai, ba? Glücklicherweise gibt es ja Suchmaschinen, die uns aus solchen Schwierigkeiten heraushelfen. Gemeiner wird es da schon, wenn der Owner sich die Mühe gemacht hat, und seine eigene Zählsprache entwickelt hat. Sofern diese nur aus den ersten 10 Ziffern besteht, hat man oft noch eine gewisse Chance, alleine über Logik, die Koordinate, wo ungefähr die Dose liegen müsste und Ausschlußverfahren weiter zu kommen. Siehe auch, später mal hier im Blog, im Bereich der Logik- und Symbolrätsel.

Es ist hilfreich den Gedanken im Hinterkopf behalten, daß ein Mystery-Fragezeichen nur 2-3 Km von seinem Versteck (oder dem Startpunkt des Caches) liegen darf. Der gesuchte Ort also die gesuchte Koordinate lässt so schon mal ein wenig einkreisen.

Neben den Ziffern können natürlich auch 10er, 100er oder 1.000er Zahlworte in Fremdsprachen benutzt werden. Diese finden sich oft schon nicht mehr ganz so einfach über Wikipedia und co und erfordern im schlimmsten Fall etwas Kombinationsgabe. Herauszufinden in welcher Sprache man sich befindet und wie dort größer 10 gezählt wird. Oft ist es ja wie im Deutschen, wo der Aufbau der Zahlwörter sich auf eine logische Art ähnelt (zumindest hinter der zwölf, übrigens ein Überbleibsel aus der Zeit, als das Dezimalsystem nicht die einzige Zählweise war und man sich

gerne Zählsystemen bis 12, bis zum Dutzend bediente.). Zwanzig, einundzwanzig, zweiundzwanzig, dreißig, einunddreißig, zweiunddreißig. Auch ein Ausländer, der der Deutschen Sprache nicht mächtig ist, wird hier die Ähnlichkeiten erkennen und möglicherweise Schlußfolgern können, daß zweiundfünfzig in lateinischen Zahlen eben 52 ist. caoga bedeutet fünfzig auf irisch, femti im norwegischen, cinqante bei den Franzosen und in allen romanischen Sprachen sehr ähnlich. Die 3 zum Beispiel ist egal ob nun dree, drie, tre, tri, thrie, tres oder trais geschrieben, in fast allen bei Wikipedia aufgeführten Beispielen für Zahlwörter (http://de.wikipedia.org/wiki/Zahlen_in_unterschiedlichen_Sprachen) leicht zu identifizieren.

Was aber, wenn wir die reinen Zahlen und Zahlenwörter verlassen, wie wir sie kennen und uns in andere Stellenwertsysteme begeben. Wenn wir nicht mehr wie in unserem Dezimalsystem bis 10 zählen, was hervorragend mit der Anzahl unserer Finger harmoniert, sondern bis zwanzig, wie im Vigesimalssystem. Eine bei Cachern häufig benutzte Variante davon sind die Maya-Ziffern:

Vielleicht zählten die Maya mit Fingern und Zehen. Auf jeden Fall teilten sie in vier Blöcke zu je fünf Ziffern, wobei ein Punkt 1 zählt und ein Strich 5. Alles war hübsch sortiert, die Striche, also die 5er unten, die Punkte, also die Einer oben. So kommt man prima bis 19. Alle Zahlen, die höher waren, wurden einfach höher, also oben drüber über die Zahlenblöcke bis 19 geschrieben. Sowas nennt man Stellenwertsystem, wobei der zweite Block von unten mit 20 multipliziert wird, der darüber dann mit 400, dann kommt 8000, ...

3.2 Binär-, Oktal- und Hexadezimalzahlen

"Es gibt 10 Sorten von Menschen. Die, die Binärcode verstehen und die, die es nicht verstehen."

Ein anderes, wesentlich bekanntere Stellenwertsystem ist das Binärsystem. Ein Stellenwertsystem mit der Basis zwei. Es gibt nur zwei Zustände, Null und Eins, an oder aus, Punkt oder Strich, da oder weg, wahr oder falsch. Computer arbeiten so, weil Schaltkreise so arbeiten. Weil Strom nur diese beiden Zustände annehmen kann. Er ist da oder er ist weg.

Das gibt gerade beim Verschlüsseln von Botschaften immens viele Möglichkeiten, die oft binär (also auf zwei Zuständen beruhen), aber nicht unbedingt im Dualsystem (also mit Null und Eins) geschrieben sind. Sehr vieles kann zwei Zustände haben. Groß- und Kleinbuchstaben zum Beispiel. Striche und Punkte (auch völlig "unbinäres", wie z.B. Morse). Etwas kann wahr oder falsch sein, zum Beispiel im Text versteckt. Was diese Form des Zahlenversteckens zu den mit am Häufigsten, aber auch mit am kreativsten genutzten Geocaching-Verschlüsselungsarten macht.

Die Dechiffrierung von Dualzahlen, also Binärcode aus Null und Eins, ist sehr einfach. Dualzahlen werden hintereinander weg geschrieben. Die erste Stelle ganz rechts zählt 2^0 , also eins (in unserem Dezimalsystem), sofern sie mit einer 1 (Strom da) besetzt worden ist. Steht dort eine 0 zählt sie auch Null, also nix. Die nächste Stelle, links neben der ersten, hat den Wert 2^1 , also zwei. Ist sie mit einer 0 besetzt, hat sie den Wert Null, ist sie mit einer 1 besetzt, besitzt sie - dezimal - den Wert zwei. Weiter geht es mit der dritten Stelle (2^2), einer, also vier, dann kommt die acht (2^3), und so geht es dann immer so weiter:

Derartige Dualzahlen lassen sich zwar leicht in unser gewohntes Dezimalsystem übertragen, zumindest bis zu einer gewissen Länge auch im Kopf, haben aber den Nachteil, daß sie extrem lang werden können. Eine in der Form dargestellte Koordinate, beispielsweise 52 45 123, einfach als große Zahl zusammengeschrieben, hat in Nullen und Einsen ausgedrückt schon eine beeindruckende Länge:

10100000000100011000011

0	1	2	3	4
5	6	7	8	9
10	11	12	13	14
15	16	17	18	19

Aus diesem Grund ist man im Bereich der Datenverarbeitung auf die Hexadezimaldarstellung gekommen. Hexa = aus dem griechischen für die 6, dezimal aus dem Lateinischen für 10. Also ein gemischtes Stellenwertsystem auf der Basis 16, wobei neben den Ziffern von 0 bis 9 auch die Buchstaben A bis F benutzt werden. Man zählt hier ganz normal mit den Dezimalziffern bis 9 und nutzt dann, wenn es im Dezimalsystem einen Sprung auf die nächste Stellenwertebene (zehn) gibt, statt diesen die Buchstaben. A (hexadezimal) ist somit eine 10 (dezimal), F eine 15. Erst bei der 16 gibt

es einen Sprung auf die zweite "Dimension", eine zweite Stelle, nur das diese nicht zehn wert ist, wie in unserem Dezimalsystem, sondern eben 16.

	Zahlensystem der Maya		
8000er			
400er			
20er			
1er			
Zahl	127	723	12016

So weit, so klar? Auswendig lernen muss man das aber natürlich nicht, es gibt diverse

Umwandlungstools im Internet und sogar der sonst eher verpönte Windows-Taschenrechner schafft die Umrechnung von Hex/Dual/Dezimal und sogar Oktal in der Programmieransicht (im Taschenrechnermenü unter Ansicht â† Programmierer). Ja, jetzt hab ich mich als Windowsnutzer geoutet, aber ich bin mir sicher, andere Betriebssysteme haben ähnliche Taschenrechner dabei. Oder Smartphones die Apps.

Die eben erwähnte Oktaldarstellung ist ein weiteres Stellenwertsystem, diesmal zur Basis 8. Es wird von 0 bis 7 gezählt, dann folgt ein Stellenwertsprung auf die 10, die dezimal acht wert ist!

oktal	0	1	2	3	4	5	6	7	10	11	12	13	14	15	16	17	20
dezimal	0	1	2	3	4	5	6	7	8	9	10	11	12	13	14	15	16
binär (dual)	0	1	10	11	100	101	110	111	1000	1001	1010	1011	1100	1101	1110	1111	10000
hexadezimal	0	1	2	3	4	5	6	7	8	9	A	B	C	D	E	F	10

Zusammenfassung :

- Findet sich in einem Rätsel 0+1 oder eine irgend geartete "binäre", zweistufige Darstellung? Vielleicht Binärcode?

- Codes, die Ziffern von 0-9 und Buchstaben von A-F beinhalten, könnten hexadezimal verschlüsselt sein.
- Zahlen, die nur Ziffern von 0-7 enthalten, "riechen" stark nach oktal...

Das Dualsystem					
dezimal	16	8	4	2	1
0+0+0+0+1=1	0	0	0	0	1
0+0+0+2+0=2	0	0	0	1	0
0+0+0+2+1=3	0	0	0	1	1
0+0+4+0+0=4	0	0	1	0	0
0+0+4+0+1=5	0	0	1	0	1
0+0+4+2+0=6	0	0	1	1	0
16+0+4+0+1=21	1	0	1	0	1
16+8+4+2+1=31	1	1	1	1	1

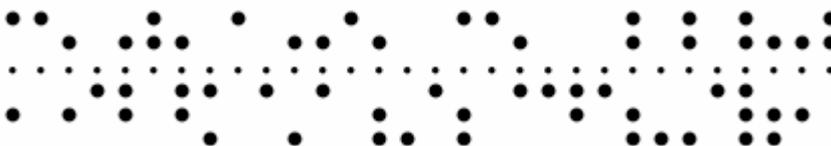
3.3 Binärcodes

Beim Mystery Enträtseln begegnen einem ziemlich häufig Nullen und Einsen. Oder etwas, was verschlüsselt auf zwei Zustände herunter gebrochen werden kann: zwei verschiedene Farben oder Töne, etwas ist wahr oder falsch, etwas ist da oder weg, lang oder kurz. Oder zum Beispiel GroSsBuchStaben, wo sie nicht hingehören (GroSsBuchStaben könnte, wenn das Großgeschriebene die 1 symbolisiert, die Binärzahl 100101001000000 ergeben).

Und dann steht man vor den Nullen und Einsen, die keineswegs immer nur Dualzahlen sein müssen. Das Beispiel oben ergäbe immerhin eine 19008,

was da, wo ich wohne, eine schlüssige Nordkoordinate ergibt- wenn man sich vorn noch ein 52° herandenkt und nach der 19 einen Punkt setzt. Die Verschlüsselung kann aber auch an einem anderen Binär-Code erfolgt sein, als dem der Dualzahlen. Ich bemühe mich hier, einige der üblicheren Binärcodes vorzustellen.

Dual	Dezimal	Hexadezimal
1	1	1
10	2	2
11	3	3
100	4	4
101	5	5
1010	10	A
1011	11	B
1100	12	C
1101	13	D
1110	14	E
1111	15	F
10000	16	10
10100	17	11
100100	42	2A
1101111	111	6F



Hilfreich bei der Suche nach dem passenden Binärcode, ist die Länge der Binärzahl. Ist sie 5 Stellen lang, oder durch 5 teilbar, könnte sie Baudot bzw. Baudot -Murray-Code sein (ITA-1 und ITA-2, bzw. CCITT-2). Dieser stammt aus der Zeit der Telegraphie und ist im Original geteilt in einen Bereich mit 2 Bits und einen mit 3 Bits. Hiermit lassen sich Zahlen und Buchstaben, sowie einige

Sonderzeichen darstellen, Häufig finden sich in den Listings derartiger Caches Bilder von Lochstreifen, die Baudot-Code darstellen. Die größte Schwierigkeit hier ist es, herauszufinden, von welcher Seite des Lochstreifens gelesen wird und welcher Baudot/ITA benutzt worden ist.

Buchstabe	Code	Buchstabe	Code	Buchstabe	Code
A	aaaaa	I, J	abaaa	R	baaaa
B	aaaab	K	abaab	S	baaab
C	aaaba	L	ababa	T	baaba
D	aaabb	M	ababb	U, V	baabb
E	aabaa	N	abbaa	W	babaa
F	aabab	O	abbab	X	babab
G	aabba	P	abbba	Y	babba
H	aabbb	Q	abbbb	Z	babbb

Nicht so richtig Nullen und Einsen, sondern As und Bs benutzte der Herr Bacon für seine [Bacon-Chiffre](#). Er hat einfach jedem Buchstaben im Alphabet einen fünfstelligen Code, bestehend aus den beiden Buchstaben A und B, zugewiesen. Das Wort *Mysterie* in Bacon-Chiffre lautet demnach: ababb babba baaab baaba aabaa baaa babba

Diese Codierung kann durchaus unauffällig im Text versteckt sein. Man könnte, auch wenn es etwas Mühe kostet, einen Text zusammenschreiben, bei dem die vorhandenen Buchstaben A und B Bacons Chiffre ergeben. Oder wieder die Groß- und kLeinSchReibUNG benutzen, und den GROSSGESCHRIEBENEN Buchstaben zum Beispiel das A zuweisen, den kleinen das B. Die Buchstaben könnten auch teilweise fett oder kursiv geschrieben werden, um die zwei Zustände, hier a oder b darzustellen. Möglichkeiten gibt es genug, und Mystery-Owner auch trickreich genug, fast unendlich viele davon zu erfinden ;).

Binary	Oct	Dec	Hex	Glyph	Binary	Oct	Dec	Hex	Glyph	Binary	Oct	Dec	Hex	Glyph
010 0000	040	32	20	☺	100 0000	100	64	40	@	110 0000	140	96	60	`
010 0001	041	33	21	!	100 0001	101	65	41	A	110 0001	141	97	61	a
010 0010	042	34	22	"	100 0010	102	66	42	B	110 0010	142	98	62	b
010 0011	043	35	23	#	100 0011	103	67	43	C	110 0011	143	99	63	c
010 0100	044	36	24	\$	100 0100	104	68	44	D	110 0100	144	100	64	d
010 0101	045	37	25	%	100 0101	105	69	45	E	110 0101	145	101	65	e
010 0110	046	38	26	&	100 0110	106	70	46	F	110 0110	146	102	66	f
010 0111	047	39	27	'	100 0111	107	71	47	G	110 0111	147	103	67	g
010 1000	050	40	28	(100 1000	110	72	48	H	110 1000	150	104	68	h
010 1001	051	41	29)	100 1001	111	73	49	I	110 1001	151	105	69	i
010 1010	052	42	2A	*	100 1010	112	74	4A	J	110 1010	152	106	6A	j
010 1011	053	43	2B	+	100 1011	113	75	4B	K	110 1011	153	107	6B	k
010 1100	054	44	2C	,	100 1100	114	76	4C	L	110 1100	154	108	6C	l
010 1101	055	45	2D	-	100 1101	115	77	4D	M	110 1101	155	109	6D	m
010 1110	056	46	2E	.	100 1110	116	78	4E	N	110 1110	156	110	6E	n

ASCII ist wohl der bekannteste Vertreter der binär-Codes, von den Dualzahlen mal abgesehen. Die "American Standard Code for Information Interchange" ist eine 7-Bit-Zeichenkodierung, die schon 1963 als Standard veröffentlicht wurde. Sie kann 128 Zeichen darstellen (das lateinische Alphabet, arabische Ziffern, einige Satzzeichen) und wird von einer unzähligen Masse von Geräten und Programmen verstanden und unterstützt. Das überzählige achte Bit (ein Ascii-Zeichen wird üblicherweise in einem Byte gespeichert, hätte also Platz für acht Bits) wird entweder als Prüfziffer missbraucht, oder um länderspezifische ASCII-Zeichensätze darzustellen. Die deutschen Umlaute zum Beispiel. Mit dem ASCII-Zeichensatz kann man hervorragend Caches verschlüsseln, zumal man ihn nicht nur direkt in binär, sondern auch in HEX oder Oktal darstellen kann.

BCD (Binary Coded Decimal) ist ein weiterer, bekannter Binär-Code. Es gibt ihn als 4- und als 6-Bit-Code. Der 4er ist ein 8-4-2-1-Code. Also ein numerischer Code, dessen Wertigkeit dem bekannten Dualsystem entspricht. Es lassen sich Ziffern von 0-9 darstellen. Theoretisch könnte man 16 statt 10 Dezimalzahlen mit einem derartigen "Halbbyte" darstellen, man hat sich aber drauf geeinigt, dass man derartiges per BCD nicht tut. Ein Byte sind ja 8 Bits, also 8 Stellen, der 4-Bit-BCD-Code ist somit ein halbes Byte lang. Man nennt dieses Halbbyte auch Nibble. Diesem bin ich bei einem Cache in freier Wildbahn tatsächlich auch schon begegnet.

BCD gibt es auch als 6-Bit-Code. Dieser kann neben Ziffern auch Buchstaben und einige Sonderzeichen darstellen.

Ziffer	codiert
1	0 0 0 0 0 0 0 0 1
2	0 0 0 0 0 0 0 1 1

```

3      0 0 0 0 0 0 0 1 1 1
4      0 0 0 0 0 0 1 1 1 1
5      0 0 0 0 0 1 1 1 1 1
6      0 0 0 0 1 1 1 1 1 1
7      0 0 0 1 1 1 1 1 1 1
8      0 0 1 1 1 1 1 1 1 1

```

Bit-Nr.	4	3	2	1
Wertigkeit	2^3 8	2^2 4	2^1 2	2^0 1
0				
1				
2				
3				
4				
5				
6				
7				
8				
9				

```

9      0 1 1 1 1 1 1 1 1 1
0 (=10) 1 1 1 1 1 1 1 1 1 1

```

Der BCD-Zählcode ist 10 Bits lang, wobei jedes Bit die Wertigkeit 1 im Dezimalsystem hat. Man bekommt die gesuchte Zahl durch plumpes addieren der Einsen. Dieses simple addieren der Einsen, anstatt kompliziert nach Codierungen zu suchen, ist etwas, was man beim Mystery-Entschlüsseln auch mal probieren könnte.

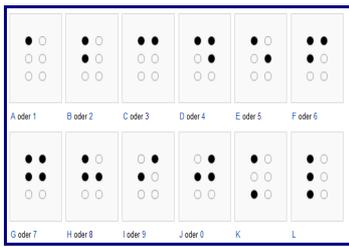
Selten aber möglich ist auch der 1-aus-n-Code, bei dem es 10 bits gibt, von denen 9 immer 0 sind und die 1 als Zähler dient. Steht sie auf der 7. Position von rechts (0001000000), steht sie für eine dezimale 6, steht sie auf der 2. Position von rechts (0000000010), ist sie Eins wert, ganz rechts (0000000001) ist es eine dezimale Null.

Weitere 4-Bit-Sonderformen: der [Aiken-Code](#), bei dem die vierte Stelle von links nicht 8 wert ist, wie bei 4-Bit-BCD, sondern 2. Es ist also ein 2-4-2-1-Code. Oder der Gray-Code <http://de.wikipedia.org/wiki/Gray-Code>, den es auch als 2-Bit, 3-Bit, 4-Bit, 5-Bit und 6-Bit-Code gibt. Er zeichnet sich dadurch aus, das benachbarte Codewörter sich nur in einer einzigen dualen Ziffer unterscheiden und ist entwickelt worden, um Ablesefehler zu minimieren.

[Braille](#), die Blindenschrift mit den kleinen Punkten, ist ebenfalls binär, also hat zwei Zustände. Es gibt 'nen Hubbel oder es gibt keinen Hubbel. Hiermit haben mich schon mehrere Mysteries geärgert, weil ich viel zu spät und von all den Computercodierungen geblendet auf das im Format 3-hoch-2-breit geguckt hab um an das Braille-System zu denken.

Morse ist auch ein Kandidat, der eigentlich nicht so wirklich und dann wieder doch binären Codes. Es gibt zwei Zustände, nur heißen die bei Morse eigentlich lang und kurz, Punkt und Strich. Prinzipiell ist es hier aber das gleiche, irgendwo finden wir die binäre Verschlüsselung und der Schlüssel hierzu könnte das Morsealphabet sein. Allerdings hat Morse einen gravierenden Nachteil, der beim Erkennen helfen könnte: die einzelnen Zeichen sind unterschiedlich lang. Häufig benutzte Buchstaben, wie zum Beispiel das E haben kurze Morsezeichen (ein .), lange Morsezeichen haben fünf Striche und Punkte. Um Morse sicher übersetzen zu können, benötigt die verschlüsselte

Botschaft Leerzeichen zwischen den einzelnen Zeichen. Ziffern hingegen sind bei Morse immer fünf Morsezeichen lang und dank ihres symmetrischen Aufbaus sogar relativ leicht zu erlernen.



Ziffer	Morse
0	--- ---
1	. --- ---
2	. . --- ---
3	. . . ---
4 -
5
6	-
7	- - . . .
8	- - - . .
9	- - - - .

4 Sprache und Schrift

4.1 - Sprache und Schrift (Teil 1)

Die Kapitel [Zahlen](#) und Sprache und Schrift verschwimmen ein bisschen, das lässt sich aufgrund der Thematik aber ebenso schwer trennen wie vermeiden. Der Schwerpunkt hier liegt ein bisschen eher in mehr oder minder sinnvollem Text, der verborgenes in sich trägt.

Buchstabenwerte

Den Anfang machen aber trotzdem erst einmal nackte Buchstaben, die letztlich ja fast immer irgendwie Zahlen ergeben sollen. Am häufigsten und einfachsten sind einfach nur die Buchstabenwerte gesucht, also die Position, die der Buchstabe im Alphabet hat. A=1, B=2, C=3, ... Das Wort Nina hat somit die Buchstabenwerte 14-9-14-1. Manchmal muss man das Alphabet auch bei 0 beginnend durchzählen, dann wäre es eben 13-8-13-0. Ggf. auch rückwärts gezählt, von 0 oder 1. Hiermit kann man nun so allerlei anstellen. Manchmal braucht man nun die Summe der Buchstabenwerte (38 bei der üblichen, bei eins beginnenden Zählung). Manchmal braucht man die Quersumme (die Ziffern einer Zahl addiert), mal von den einzelnen Buchstabenwerten, mal von der Gesamtsumme. Mal die einfache Quersumme (Ziffern nur addiert), mal die iterierte (so lange die Quersumme aus dem Ergebnis der Quersummenbildung errechnen, bis das Ergebnis einstellig ist). Manchmal muss man die ermittelten Ziffern miteinander Multiplizieren oder andere Rechenoperationen anstellen, die sich hoffentlich aus dem Listing ergeben. Manchmal braucht man nur von bestimmten Buchstaben die Werte, zum Beispiel von allen Anfangsbuchstaben oder allen großgeschriebenen, kursivgeschriebenen, in einer bestimmten Farbe geschriebenen, ...

Stellenwert

Ist ein Buchstabe anders, als die anderen (Farbe, Stil), kann auch seine Stelle im Wort der Schlüssel sein. Ist also im Wort "Buchstabe" zum Beispiel das *h* kursiv, könnte die Ziffer 4 gesucht sein. Sticht mehr als nur ein Buchstabe heraus, könnte mal wieder eine [Binärcodierung](#) gesucht sein. Dieses "Testwort" ergäbe zum Beispiel, wenn man die fettgeschriebenen Buchstaben als 1 annimmt und die normalen als 0: 0110 0110. Wäre es eine Binärzahl, ergäbe es dezimal 102. Da dieses "Binärwort" praktischerweise aus 8 Buchstaben, also auch aus acht Bits (demnach einem Byte) besteht, wäre auch [ASCII](#) ein toller Kandidat. Da ist 01100110 / 102 mit dem Buchstaben "f" belegt. Welches- und hier schließt sich ein Mysterykreis - dem Buchstabenwert 6 entspricht.

Manchmal liegt des Texträtsels Lösung auch einfach in der Anzahl der Buchstaben eines oder aller Wörter. Oder die Anzahl der Silben. Konsonanten. Vokale. Vielleicht muss man nur bestimmte Buchstaben zählen? Oder welche, die bestimmte Eigenschaften haben. Zum Beispiel Striche nach oben, wie das l und das h. Oder nach unten reichen, wie das p und das g. Vielleicht braucht man die Anzahl von runden Bereichen, wie im O oder im d. Solch Buchstaben- oder auch Zahlenbereiche, die sich ausmalen lassen, waren bereits mehrfach der Schlüssel des einen und anderen Mysteriums, den ich von der Karte habe tilgen dürfen. Vielleicht muss man derartiges auch wieder einer [Binärcodierung](#) gegenüberstellen. Hat man auf eine solche Art eine große Anzahl von Zahlen ermittelt, wäre es auch denkbar, daß man die gefundenen Zahlen einfach nur als gerade und ungerade Zahlen betrachten und demgemäß in einen Binärcode übersetzen muss.

Man könnte auch die Anzahl der Buchstaben bis zum nächsten Leerzeichen oder Interpunktionszeichen zählen. Oder nur den jeweils ersten, oder dritten oder meinethalben auch achten Buchstaben jedes Wortes zur Entschlüsselung benutzen (um das zu probieren empfiehlt es sich, die einzelnen Wörter untereinander zu schreiben).

Apropos erster Buchstabe: ein gewisser, aber längst verstorbener John Laird McCaffrey hat in Montreal diese wunderhübsche Grabinschrift:

"John
Free your body and soul
Unfold your powerful wings
Climb up the highest mountains
Kick your feet up in the air
You may now live forever
Or return to this earth
Unless you feel good where you are!
â€”Missed by your friends"

Seine Freunde scheinen keine sonderlich guten gewesen zu sein, wenn man die Anfangsbuchstaben der einzelnen Zeilen von oben nach unten liest.

Zahlen als Wörter

Allerdings hat diese Grabinschrift keine Koordinate ergeben, wir suchen aber immer noch nach Ziffern. Oder suchen wir doch Wörter? Natürlich kann man Zahlen auch in Buchstaben ausdrücken und somit verschlüsselte Texte ordentlich in die Länge ziehen, die damit zwangsläufig schwieriger zu entschlüsseln werden. Eine 0-9er Reihe habe ich mit etwas Logik oft sehr einfach und am eigentlich Rätsel vorbei gelöst. Vorausgesetzt, die die gesuchte Zahl ist die komplette Koordinate, die ja in den ersten 3-4 Stellen vorne bekannt ist (durch die Koordinate, an der das ? gelistet ist - der Cache selber darf laut Richtlinien von geocaching.com seit einigen Jahren nur 2-3 Kilometer von diesem Punkt entfernt sein). Handelt es sich aber um Buchstaben als Text, um "eins", "zwei", "drei" oder noch schlimmer "zweiundfünfzig" oder "dreiundzwanzig", wird es für den Rätselnden viel schwieriger.

Derartig ausgeschriebene Ziffern lassen sich auch prima in völlig unverdächtigen Texten unterbringen. ZWEIFelnd beobACHTete ich die AktiVIERung der prACHTvollen VerEINscoin. Gut, um da jetzt einen hübschen Mysterytext draus zu basteln, brauchts wohl noch etwas Feinschliff, aber um das Prinzip darzulegen reicht mein Beispiel hoffentlich grad so eben noch aus.

Verwürfeln

Findet sich nur Buchstabensalat? Vielleicht ist dieser nur Rückwärts geschrieben und/oder die Lücken zwischen den Wörtern sind nicht da, wo das Auge sie gern zum Lesen hätte? Möglicherweise steht aber auch der erste Buchstabe für ein Wort der Nord- der zweite für eines der Ostkoordinate, der nächste dann wieder für Nord usw.. Dieser Code hier: "znwuelilunnudlflünnefuznisgeacchhstuunnddizrweainszsiiggderienihzuenhdnert" ergibt, derartig enttündelt: zweiundfünfzigachtundzwanzigdreizehn und nullnullneunsechsdreissigehundert.

Der Lattenzaun- oder Jägerzaun-Chiffre funktioniert ähnlich, nur das hier der zu verschlüsselnde Text diagonal nach unten und dann wieder nach oben geschrieben wird und am Ende zeilenweise verwendet wird. Aus zweiundfünfzigachtundzwanzigdreizehn wird dieser Gartenzaun

```
z   d   g   n   z   i
w  n f  i a  u d  n i  e z
 e u  n z  c t  z a  g r  e n
  i   f   h   w   d   h
```

und damit dieser verschlüsselte Text: zdgnziwnfiaudniezeunzctzagrenifhwdh.

Ähnlich funktioniert es mit dem "Pflügen". Der zu verschlüsselnde Text wird normal aufgeschrieben, aber ein Raster von X Zeichen pro Zeile. Anschließend wird dieser mehrzeilige Text Zeilenweise von oben nach unten, und dann, wenn man unten angekommen ist, von unten nach oben, genommen. Quasi wie beim Pflügen eines Feldes. Man hätte es auch "Rasenmähen" nennen können ;).

Einfacher zu ent- und verschlüsseln ist der Trick, bei dem man auf der Tastatur immer nur den jeweilig links oder rechts, oben oder unten vom abgedruckten Buchstaben drückt. Aus zweiundfünfzig wird, rechts verschoben, ueroimfg+mguoh .

Handytastaturcode (Vanity-Code)

Vielleicht ist es aber auch nur die gute, alte (und von mir viel zu oft vergessenen) Handytastatur? Hier gibt es ja zum Beispiel für die Ziffer 2 a, b und c, die 3 ist mit d, e und f belegt. Der Buchstabenwust Kcbadnz steht so dann für die hübsche Nordkoordinate: 52 22.369 . Diese Form der Verschlüsselung passt in beide Richtungen. Hat man mal einen Zifferncode, der keine 1en aufweist, aber häufige Ziffernwiederholungen, schadet es sicher nichts, mal eben das Handy zu Rate zu ziehen. 69997777833777999 â†? ergibt das Wort Mystery.

Für fortgeschrittene Daumentipper gibt es noch die T9-Texterkennungsvariante. Man nutzt den Vorschlagsalgorithmus der Handys. Das ist aber eher schlecht weil selten eindeutig, das interne Wörterbuch kann vom Nutzer erweitert sein und damit Fehlvorschläge bringen und nicht jedes Handy nutzt T9 (Motorola zum Beispiel kocht ein eigenes Texterkennungssüppchen). Trotzdem wird diese Verschlüsselung manchmal benutzt.

4.2 - Sprache und Schrift (Teil 2)

Nachdem im vorherigen Artikel dieses Kapitels, [Sprache und Schrift \(Teil 1\)](#), schon einige eher einfache Möglichkeiten zur Verschlüsselung von Koordinaten im Text gezeigt worden sind, geht es nun weiter.

Römische Zahlschrift

Ein Chronogramm ist ein Satz oder Satzteil, üblicherweise in lateinischer Sprache, in dem die vorhandenen Buchstaben, die römische Zahlensymbole sind (I, V, X, L, C, D, M), eine Jahreszahl (meist das Baujahr des Gebäudes) ergeben. Häufig wird dabei die übliche, römische Subtraktionsregel außer acht gelassen und die Zahlenwerte einfach addiert.

Nutzt man die Subtraktionsregel, soll damit vermieden werden, daß mehr als drei gleiche Zeichen nebeneinander geschrieben werden, so daß das nächstgrößere Zeichen gesetzt und das jeweils kleinere davor geschrieben, und daher von der größeren Zahl subtrahiert werden muss. Klingt komplizierter als es ist. Normalerweise werden Römische Zahlen von groß nach klein notiert. 2013 ist in römischen Zahlen ausgedrückt: MMXIII M=1000, X=10, I=1. Alles hübsch der Größe nach sortiert aufgeschrieben, also wird hier in jedem Fall nur addiert. $M=1000 + M=1000 + X=10 + I=1 + I=1 + I=1 = 2013$.

I=1, II=2, III=3. IIII wären vier mal ein Zeichen, und wird daher in der Subtraktionsschreibweise IV ausgedrückt. Das eine, kleine Zeichen vor dem großen = subtrahieren! IV bedeutet also 4, VI somit 6.

Natürlich gibt es ne Menge Ausnahmen, außerdem ist die Subtraktionsregel auch mehr eine Richtlinie, kein Gesetz. Und beim Chronogramm wird sowieso überwiegend einfach nur addiert, und sogar die Reihenfolge der römischen Zahlensymbole oft ignoriert.

Der letzte Absatz enthält die die römische Zahl

ALICIMAAMADMIDIAILACMIICLIIDIMCMMIDIIDACADDIDADIILDMICALMLII, was dezimal 65193 ergibt.

Bei vereinfachter Umrechnung (ohne A und ohne Subtraktionsregeln) ergibt sich: 14921 (einfach errechnet durch die Internetseite [Numerologie](#)).

Üblicherweise werden in Mysteries nur die römischen Zahlzeichen bis 1000 benutzt, also: I=1, V=5, X=10, L=50, C=100, D=500, M=1000.

Soviel zu den alten Römern. Aber was machen wir mit einem Text, in dem der eine oder die andere Deutsche Stadt genannt wird? Möglicherweise lohnt es sich ja, nach dessen *Postleitzahl* oder *Vorwahlen* zu gucken? Oder suchen wir vielleicht [Flughafencodes](#), [Bahnhofsnummern](#), [Autobahnen](#) oder [Ländercodes](#)?

Rechtschreibfehler und andere Unterscheidungen

Gibt es viele Rechtschreibfehler? Auch wenn heutzutage nicht mehr in jedem Cachemobil ein Duden dabei zu sein scheint, kann eine Fehlerhäufung auch als Verschlüsselung funktionieren. Als Schlüssel könnte hier Wörterzählen funktionieren, also zum Beispiel in jedem Satz bis zum falsch geschriebenen zählen. Vielleicht muss man die falschen und richtigen Wörter als 0 und 1 in einen [Binärcode](#) übersetzen? Vielleicht sind aber auch die Korrekturen als Buchstabenwerte der Schlüssel? Alternativ die falschen Buchstaben in Buchstabenwerte (A=1, B=2,...) übersetzt? Ebenso könnten bestimmte Buchstaben kursiv oder fettgedruckt sein, wie bei einer alten Schreibmaschine leicht nach oben oder unten verrutscht, mit einem winzigen "Farbklebs" verschmiert oder (der Blick in den Quelltext zeigt es!) in einer ganz leicht anderen Farbe geschrieben? Oder steht die Lösung einfach weiß auf weiß zwischen dem Listingtext, der nur der Ablenkung dient? Einfach mit der Maus das Listing markieren enttarnt derartiges.

Codewörter / Jargon-Code

Codewörter waren schon früher gern genommene aber auch leicht zu durchschauende Verschlüsselungen. Manchmal muss man allerdings zweimal hinschauen, um sie zu enttarnen. So kann man zwar Zahlwörter wie ACHTung noch leicht der 8 zuordnen, aber das "nicht autorisiert" für die Zahl 401 stehen könnte ([http-Fehlermeldungen](#)), ist schon von etwas weiter weg daher geholt. Im Zweifel hilft eine Suchmaschine oder eine Seite wie [Code-Knacker](#), um merkwürdige Begrifflichkeiten zu enttarnen. Wer ein wenig mehr über derartige [Jargon-Codes](#) lesen möchte, dem kann ich Klaus Schmech und seine Bücher wärmstens ans Herz legen. Sie sind spannend, witzig und informativ, quasi die Überraschungseier unter derartiger deutschsprachiger Literatur.

Auch unter Nichtfunkern gibt es eine recht bekannte Sammlung von Codewörtern, die Buchstaben nicht unbedingt verschlüsseln, immerhin fangen sie mit diesen an. Trotzdem und der Vollständigkeit halber: die [Buchstabiertafel](#). Anton, Berta, Cesar, Dora, Emil...

Noch 'n Gedicht!

Ist der Listingtext ein bekannter Text, vielleicht sogar ein Lied oder ein Gedicht? Dann unterscheidet es sich möglicherweise in winzigen Details vom Original und die Unterschiede sind (als Buchstabenwerte oder Stellenwert) der Schlüssel?

Manchmal findet man auch unter einem irgend gearteten Text Zahlensalat wie 4-3-1, 7-3-3, 8-8-8. Meist bedeutet das einfach nur, daß man im x-ten Absatz das x-te Wort und davon den x-ten Buchstaben benutzen soll. Und davon dann - wie fast immer - natürlich dessen Buchstabenwert.

Schreibt der Cache-Owner in merkwürdigen Hieroglyphen, so wie: ""\$%Â&\$/() ? Dann schaut doch mal etwas genauer auf eure Tastatur (sofern ihr noch mit einer Standardtastatur herum surft), vor allem auf die Zeile mit den Ziffern.

Buchstabensalat

Ist es absoluter Buchstabensalat, dann könnte es sich um [Base64](#) handeln. Dies dient der Kodierung von 8-Bit-Binärdaten in ASCII-Zeichen und enthält Klein- und Großbuchstaben, sowie die Ziffern 0-9 und die Zeichen + und /. Ein naher Verwandter ist [Base85](#), welches zusätzlich noch diverse Sonderzeichen enthält. Gibt es Groß- und Kleinbuchstaben, Ziffern von 0-9 und die Zeichen + und - dürfte es sich um [UUencode](#) handeln.

Absoluter Buchstabensalat kann auch nur zur Verwirrung dienen. Manchmal lohnt sich konzentriertes "Lesen" dieser Buchstabensuppe, um aus ihrem tiefsten Inneren noch Sinn entnehmen zu können.

5. - Dateianalyse: Bilder - Musik - Video

5.1 Bilderanalyse - Teil 1: Einleitung und optisches

In Bildern lassen sich auf erschreckend viele Arten Informationen verbergen. Und auch die Bilder selber sind gar nicht immer auf Anhieb zu finden. Ein komplett weißes Hintergrundbild zum Beispiel muss man erst mal aktiv suchen, um es als vom Mystery-Ersteller integriertes Bild wahrzunehmen.

Befinden sich Bilder im Listing, sollte man immer, am sichersten per Quelltext, nachschauen, ob nicht noch weitere Bilder hinter diesem sichtbaren verlinkt worden sind. Man kann einen Link hinter ein Bild legen oder ein weiteres Bild. Gern auch das gleiche Bild in größerer Auflösung. Möglicherweise soll das größere Bild aber auch nur den Anschein wecken, genau gleich zu sein und enthält in Wirklichkeit den gesuchten Hinweis?

Etwas nervig sind die Bilder, bei denen nur ein winziger Bildpunkt mit einem Link versehen ist, was den Rätselnden dazu bringt, Ewigkeiten mit der Maus hin und her zu fahren, bis er diesen gefunden hat. Schneller erledigt man die Suche über den Quelltext. Hierzu beispielsweise mit dem Firefox den Bereich mit dem Bild markieren, mit der rechten Maustaste auf das Listing klicken, View Selection Source / Quelltext anzeigen klicken und nun den blau hervorgehobenen Bereich untersuchen. Ein Bild beginnt im HTML-Code mit `<img src = " dargestellt, Links mit href = "`.

Folgen dem Bild Eintragungen wie: `< area shape = " rect" coords="0,0,603,105 " href = "` befinden sich diese Verweis-sensitive Grafiken (Image Maps) hinter dem Bild. Die Links kann man nun über das Quelltextfenster direkt öffnen oder kopieren.

Liegen die verlinkten Bilder auf einem anderen Webpace, sind also nicht bei Geocaching.com hochgeladen, sind sie gleich ein Stückchen verdächtiger. Wo die Bilder im Internet liegen sieht man zum Beispiel im Firefox, in dem man mit der rechten Maustaste auf das Bild klickt und "view Image" oder "Bild anzeigen" wählt. Der Pfad, in dem das Bild zu finden ist, steht nun oben in der Adresszeile. Alternativ lässt sich diese Information auch über Eigenschaften (auch in anderen Browsern, sogar im Internet Explorer 10) anzeigen.

Wenn man Bilder bei geocaching.com oder Bilderhostern wie z.B. Imagehack hoch lädt, kann man den Dateinamen nicht beeinflussen. Anders bei eigenem Webpace. Findet man gar keinen Einstieg in das Rätsel, könnte hier dann auch der Bildname von Bedeutung sein. Rot13? Hex? Base64? Irgendwas, was man mit google entzaubern kann?

Liegt das Bild auf einem privaten Webpace, wäre es auch möglich, dass sich das im Listing verlinkte Bild zu bestimmten Uhrzeiten verändert, dann also ein ganz anderes zu finden ist. Gibt es im Listing oder im Bild einen Hinweis darauf, dann sollte man zur passenden Uhrzeit mal die ganze Seite neu laden. Dies am Sichersten, in dem man den in den Webbrowsern normalerweise verwendete Cache (gemeint ist hier der Zwischenspeicher vom Browser, keine Dosensuche ;)) umgeht, sonst würde ein verändertes Bild gar nicht neu geladen werden. Drückt man (bei allen

typischen Windowsbrowser) die Taste Strg und F5 wird der Cache umgangen, die gesamte Seite neu geladen und ein eventuell verändertes Bild auch angezeigt.

Optisches Versteckspiel

Die einfachste Art, in einem Bild weitere Informationen zu verstecken, ist es, diese deutlich sichtbar einfach drauf zu schreiben. Hat dieses Bild eine sehr großen Auflösung und wird daher auf dem Bildschirm stark verkleinert dargestellt, kann derartige Beschriftung - je nach Untergrund - fast unsichtbar sein. Sehr große Bilder sollten daher immer ganz besonders unter die - wortwörtliche - Lupe genommen werden, also in einem Bildbetrachter geöffnet und vergrößert angeschaut werden. Mein Lieblingsbildbetrachter ist seit einer digitalen Ewigkeit die Freeware [IrfanView](#). Ebenfalls aus meiner kleinen Weltsicht empfehlenswert ist [xnview](#). Von diesen beiden abgesehen gibt es aber eine Hülle und Fülle an weiteren, nützlichen Tools, die als Bildbetrachter gute Dienste leisten.

Eine durchaus klassische, optische Methode, Informationen in Bildern zu verbergen, die schon lange vor Geocachingspielereien genutzt worden, ist, ist das Anbringen von kleinen Details in einem Bild, die sich, wenn man den Code kennt, entschlüsseln lassen. Ein Beispiel ist [dieses Bild mit den Grashalmen am Bachlauf](#) (wegen unklaren Besitzverhältnissen an den Rechten hab ich es nur verlinkt und nicht selbst im Blog dargestellt).

Die kurzen und langen Grashalme sind Morsecode und ergeben den Text:

"Compliments of CSPA MA to our chief Col. Harold R.Shaw on his visit to San Antonio May 11th 1945."

Einen Gruß an den damaligen Chef der US Zensurbehörde, die sich viel Mühe gegeben hat, die Post auf Auffälligkeiten zu untersuchen. Überliefert sind so nette Geschichten wie die, dass man bei einer Lieferung von Uhren alle Zeiger verstellt hat, da man fürchtete, in den voreingestellte Uhrzeit wäre einen Code verborgen. In einem anderen Brief fand man ein Strickmuster und lieferte diesen erst aus, nachdem eine Mitarbeiterin dieses nachstrickte und somit bewies, daß es wirklich einen Pullover ergibt. Niemand geringerer als Charles Dickens hat eben diese Verschlüsselungsform in einem seiner Romane, "Eine Geschichte aus zwei Städten" benutzt und damit vermutlich auch erfunden. Umgekehrt wurden im zweiten Weltkrieg von deutschen Agenten in England Pullover verschickt, die, beim Geheimdienst wieder aufgeribbelt, Fäden ergaben, in denen in bestimmten Abständen angebrachte Knoten einen Code enthielten.

Punktchiffres sind historisch belegt und wurden schon vor 400 Jahren benutzt. Hierbei wird ein Achsenkreuz benutzt, wobei eine Achse die Buchstaben und eine die Reihenfolge dieser Buchstaben im Text ist. Als Punkte können dann zum Beispiel Sterne wie in Blaise de Vigeneres Buch "Traicte des Chiffres, ou Secretes Manires d Ecrire" oder Bienen wie in der französischen Zeitschrift Spirou benutzt und mit ausreichend Tarnung zu einem Bild verknüpft werden.

Man kann von den Geheimdienstmethoden der Zeit und die beiden Weltkriege und den kalten Krieg viel lernen, waren computerbezogene Verschlüsselungen ja noch unbekannt und mussten Informationen gut verborgen und verschlüsselt übertragen werden. So ist es geheimdiensttechnisch klassisch, bestimmte Buchstaben oder Wörter eines Textes unauffällig zu markieren. Mit einem fast unsichtbaren Punkt über bestimmten Zeichen oder einem verrutschten Schreibmaschinenbuchstaben, einer fast zufällig angebrachten Verschmierung auf dem Papier. Irgendwas, was Teile des Textes von anderen unterscheidet. Das Verstecken von Informationen in irgend gearteten Containern (zum Beispiel Bildern oder Texten) nennt man [Steganographie](#). Eine

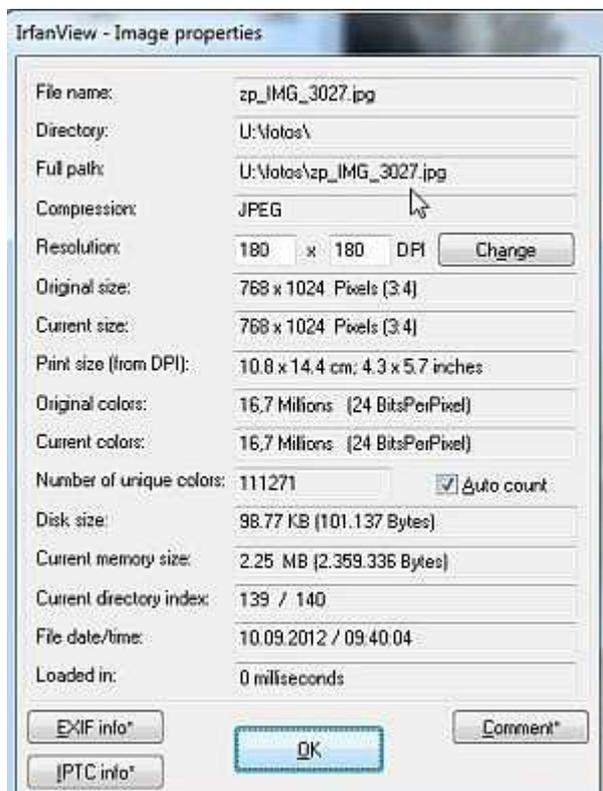
Unterart davon, das Verstecken in Details dieser Texte und Bilder hat den schönen Namen Semagramm.

Hat man das Verschlüsselte gefunden, oder weist das Bild sowieso sehr deutlich darauf hin, wo es die für die Koordinatenermittlung benötigten Informationen enthält, muss man diese ja "nur noch" entschlüsseln. Am Einfachsten, in dem man erst mal die bekanntesten Techniken und Codelisten dagegen hält. Wie viel unterschiedliche Codezeichen gibt es denn? Zwei? Dann vielleicht Dualzahlen, Morse, Braille, (siehe [Binärcodes](#)). Oder muss man etwas bestimmtes Recherchieren und hierbei benutzen? Dann ergibt das Listing, der Titel oder der Hint hoffentlich einen kleinen Stubser in die richtige Richtung.

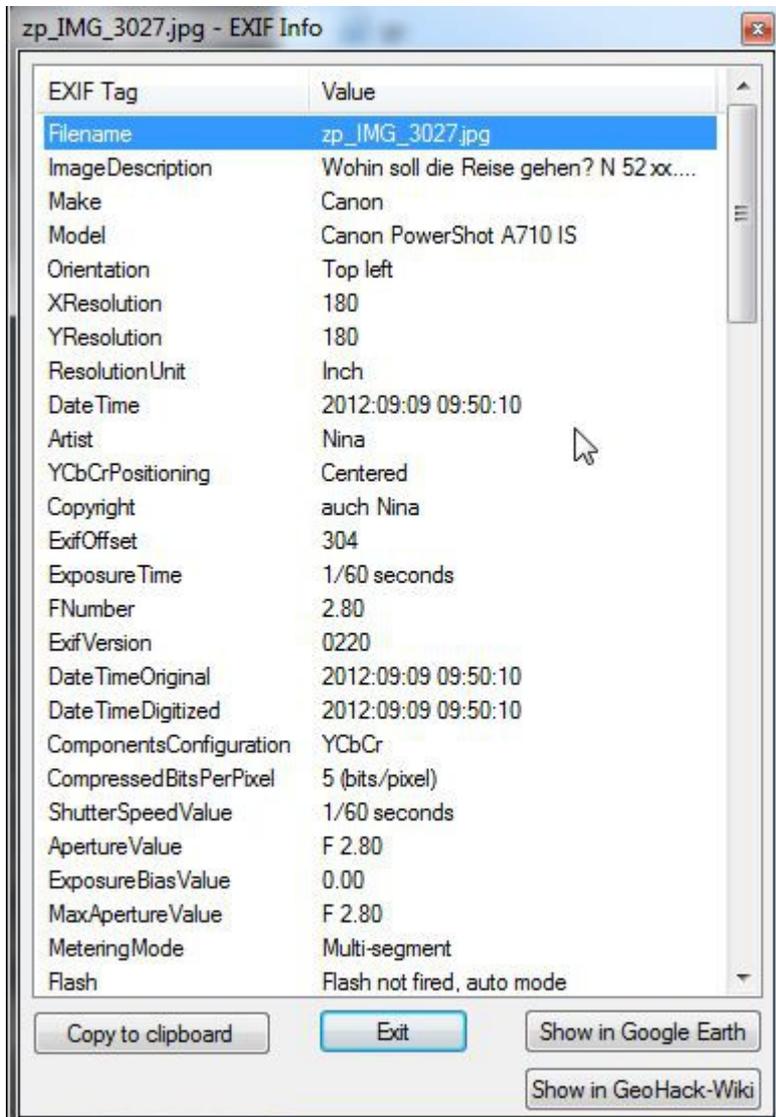
Hat man nichts gefunden, kann man es mit dem folgenden Beitrag, der [technischen Bilderanalyse](#) versuchen

5.2 Bilderanalyse - Teil 2: Technische Bilderverstecke

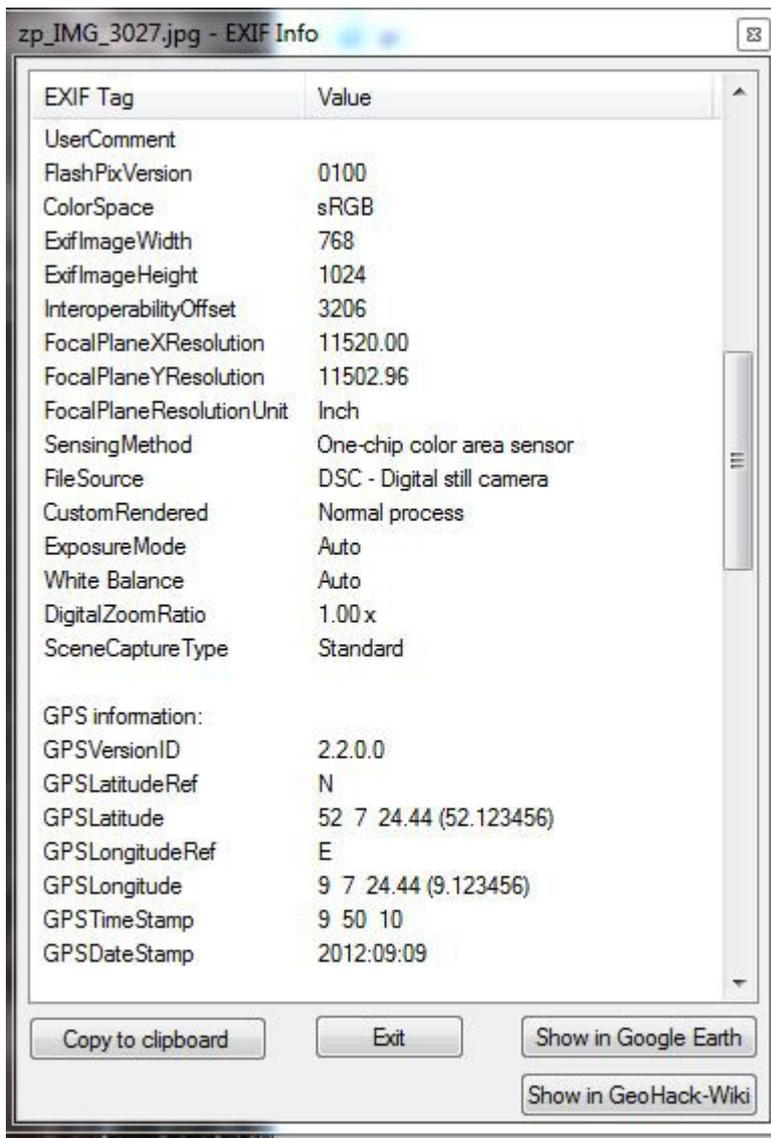
Neben dem rein optischen Varianten lassen sich mit digitalen Bildern auf technische Weise eine Menge Scheußlichkeiten anstellen. Bei Geocaching-Mysteries am Schlimmsten finde ich computergestützte [Steganografie](#). So schön es ist, unsichtbar und fast unknackbar mit kostenlos erhältlichen Programmen fast alle erdenklichen Dinge in Bildern zu verstecken, ist es für den Mysterylöser eher mühsam, da man zum Entschlüsseln genau die Software benutzen muss, mit der auch verschlüsselt worden ist. Weiß man welche es ist, muss man möglicherweise nur noch ein Passwort ermitteln und ist eine Runde weiter. Kennt man die Software nicht, ist es ein mühsames, langweiliges Durchprobieren der üblichen Verdächtigen.



Es gibt aber viel schönere und einfachere Methoden, Informationen in Bildern zu verstecken und wiederzufinden. So könnte einfach die Bildgröße, Höhe und Breite in Pixeln, die fehlende Nord- und Ostkoordinate sein (oder doch wenigstens deren letzte drei Ziffern). Alternativ könnte auch die Stelle eines relevanten Pixels die Koordinate anzeigen.



Ist das Bild ein JPEG (die Endung des Bildes .jpg), kann es EXIF-Zusatzinformationen beinhalten. Gedacht sind diese für Autorenvermerke, Kommentare oder Daten zu Kameras und Fotoeinstellungen. Neuere Fotoapparate enthalten oft schon einen GPS-Empfänger und speichern die Koordinate, an der das Bild gemacht wurde, gleich mit. Smartphones können dies ebenfalls. Und so mancher Mystery verrät seine Finalkoordinaten so schon durch unachtsame Geocacher, die Fotos mit derart interessanten Zusatzinformationen unwissentlich in ihren Logs mit hochladen.

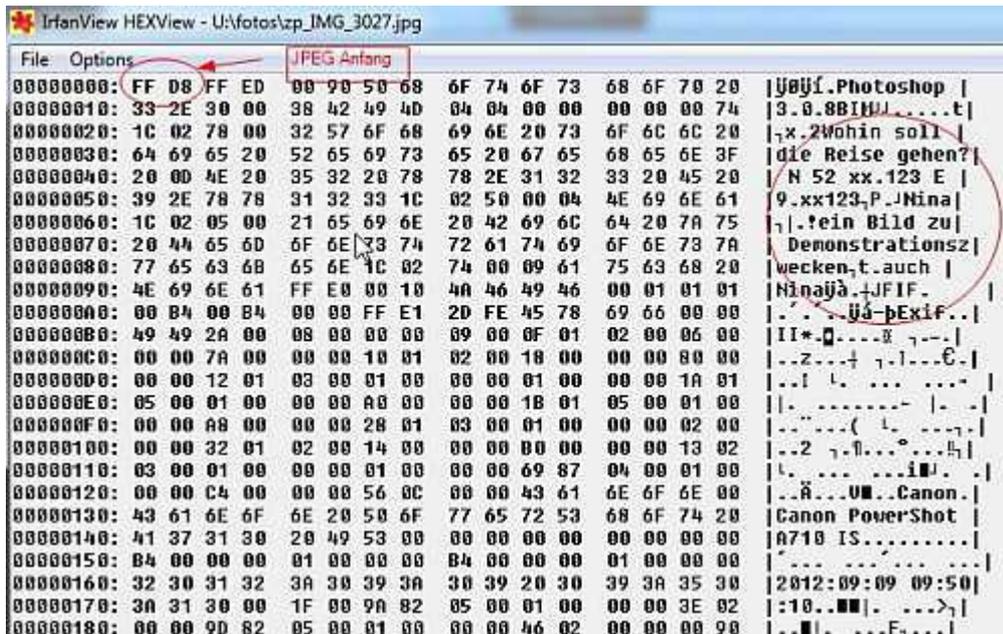


Für das Sichtbarmachen dieser Zusatzinformationen (EXIF, IPTC, Comments) kann man Bildbetrachter benutzen (das erwähnte IrfanView oder xnview) oder das Internet bemühen ([Jeffreys Exif viewer](#)). Bildbearbeitungsprogramme zeigen dieses ebenfalls an. Mein Favorit hierbei ist die Freeware Gimp, die ihrem großen Bruder Photoshop zumindest in meinen semi-professionellen Anwendungsgebieten in nichts nachsteht. Wer Firefox nutzt, kann auch auf einige entsprechende Add-ons zurückgreifen, welche das Herunterladen des Bildes möglicherweise überflüssig machen. Bei meinen Tests funktionierten diese aber eher nur mäßig gut.



Findet sich in den Bildereigenschaften nichts sinnvolles, sollte man sich das Bild mal in einem Hexeditor anschauen. Im IrfanView kann man dies mit View -> Show hex view bzw. durch das Drücken der Taste F3. Im IrfanView werden dann in einem Extrafenster links die Hexadezimaldaten angezeigt, rechts der "Klartext", der bei einem Bild natürlich weitestgehend nicht menschenlesbar ist.

Ein normaler Hexeditor (wie z.B. HxD) tut selbiges natürlich ebenfalls. In der Hexansicht kann man nun vielleicht Informationen oder auch Manipulationen entdecken, die man dem Bild selber nicht ansieht. Das Format jpg ist zwar grundsätzlich an feste Regeln gebunden, nur kann man diese ziemlich beugen und das Bild wird noch immer fehlerfrei angezeigt. So lässt sich an einigen Stellen (im oberen Bereich und ganz unten) Text verbergen, den der geneigte Leser nur in der Hexansicht zu lesen bekommt. Beim Geocaching gern benutzt ist das Anhängen eines Zips, also einer gepackten Datei, an das jpeg-Bild. Öffnet man dieses jpg in einem Zip-Programm (z.B. Winrar), öffnet sich das Archiv und man kann den Inhalt entpacken. Manchmal hat der Owner des Mysterys noch ein Passwort eingebaut, was sich dann hoffentlich aus dem Listing ergibt (vielleicht Cachetitel, GC-Code oder Ownername?).



Ob einem Jpeg-Bild etwas angehängt worden ist, lässt sich über die Hex-Ansicht leicht überprüfen. Laut den [JPEG-Spezifikationen](#) muss ein JPEG-Bild mit dem Hexadezimal-Wert FF D8 beginnen und mit FF D9 enden. Endet es anders, ist es manipuliert. Etwas ist wie das Gezippte ist angehängt worden, oder möglicherweise einfach Text am Ende eingefügt. Um diesem Geheimnis auf die Spur zu kommen, muss man - wenn die Infos nicht schon direkt ablesbar sind - mit einem Hex-Editor alles, was nach dem eigentlichen Ende des JPEG-Bildes, also dem Hex FF D9 , abschneiden und in eine neue Datei packen (das FF D9 kann, beabsichtigt oder nicht, mehrfach vorkommen! Dann müssen eventuell alle möglichen "Schnittvarianten" ausprobiert werden). Dieses neue Dokument nun unter einem passenden Namen speichern und mit sinnvollen Programmen öffnen. Es könnte sich ja ebenfalls um ein Bild handeln, dann wäre ein Bildbetrachter angebracht. Vielleicht auch ein Texteditor. Ein Video- oder Musikplayer vielleicht? Mag möglicherweise ein PDF-Reader Inhalte anzeigen?

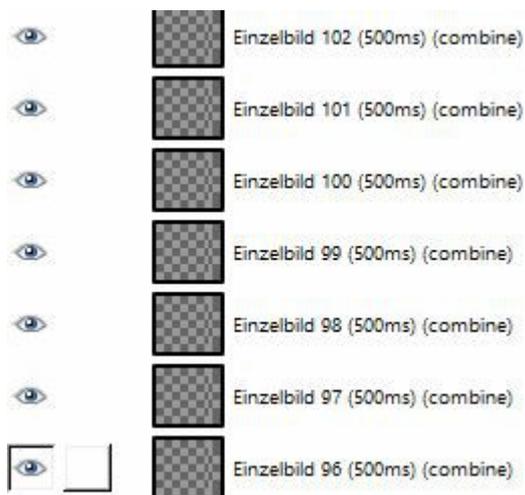
Auch an PNG-Dateien lassen sich Daten anhängen. In HEX betrachtet fängt ein PNG mit den HEX-Zahlen 89 50 4E 47 an und endet mit 49 45 4E 44 AE 42 60 82 . Gibt es nach der Endung noch weiteren Hexcode, würde ich den per HEX-Editor mal abschneiden, abspeichern und gucken, was es wohl sein könnte. Sollte es wieder mit 89 50 4E 47 beginnen, ist es ein neues PNG. Alternativ kann man unbekannte Dateien mit den typischen Programmen (Bildbetrachter, Audioanalyser, Notepad, o.äh.) zu öffnen versuchen. Oder das Internet bemühen und die ersten HEX-Zeichen einer Suchmaschine geben. Die allermeisten Dateiformate haben spezielle Startsequenzen.

Eine weitere Entzauberung fieser Bilderrätsel kann man über die Farb-/Kontrast-/Sättigungs- und Gammaregler einer geeigneten Bildbetrachtungs- oder -bearbeitungssoftware durchführen. Einfach mal die vorhandenen Regler (im IrfanView unter Image / Color correction zu finden, im Gimp gibt es verschiedene Fenster unter dem Menüpunkt Farben) hin und her schieben. Besonders den Gammaregler auch gern komplett mal an beide Enden bewegen, so manches Mal taucht dann, wie von Zauberhand, etwas im Bild auf, was vorher definitiv unsichtbar gewesen ist. Nun... eigentlich nicht ganz unsichtbar, je nach Bildart und Farbfüllung wäre man diesem Trick auch auf die Schliche gekommen, wenn man die Bearbeitungsfunktion (zB. Im IrfanView Edit, Paint Dialog) benutzt und

mit diesem schicken Eimerchensymbol  mal Farbe über diverse, optisch einfarbige Bereiche kippt (Toleranzschwelle auf so klein wie möglich setzen). Dieser Gegenzauber hilft aber nur bei

Bildern mit wenig Details und großflächig einfarbigen Stellen. Alternativ kann man auch mit dem "Zauberstab" der Bildbearbeitungssoftware und einer Empfindlichkeit von "0" oder "1" versuchen, alle exakt gleichfarbigen Stellen zu markieren und erhält so die, die sich vielleicht nur marginal und optisch unsichtbar unterscheiden.

Hat man ein Bild vorliegen, was nur aus zwei Farben besteht oder besonders auffällige Bildbereiche besitzt, könnten die Farbwerte der Schlüssel sein. Mit Gimp und der Pipette im Werkzeugfenster kann man den Farbwert "aufnehmen" und mit einem Doppelklick auf das Quadrat, welches nun im Werkzeugfenster nun die Farbe angenommen hat, erfährt man Details zu ihr. Zum Beispiel die RGB-Farbwerte, bei denen Rot, Gelb und Blau jeweils mit einer Zahl dargestellt werden (R 52/G 12/B 33) oder der Hexadezimalwert der Farbe, wie #4fad91. Oh wie hübsch, dezimal ist das 5221777, eine Nordkoordinate! ;). Was man alles mit Farben in Geocaches anstellen kann, habe ich [hier](#) versucht zu erklären.



Hat man eine GIF-Datei vorliegen (endet mit .gif), kann diese aus mehreren Teilen bestehen, die nach einer vorbestimmten Zeit wie ein Daumenkino ablaufen. Das kann so schnell sein, dass es wie ein solches aussieht, oder so langsam, dass man vermutlich gar nicht mehr hinschaut, wenn das Bild endlich mal wechselt. Daher sollte man GIFs immer auseinander nehmen. Im Gimp geöffnet erkennt man dann im Ebenenfenster die verschiedenen Ebenen, in xnview kann man unter Ansicht -> Frame zwischen den einzelnen hin und herspringen und sieht deren Anzahl und die eigene Position unten links in der Statusleiste. [Jeffreys Exif viewer](#) zeigt alle Einzelbilder auch online an.

GIFs und PNGs können Informationen beinhalten, die man vor einem weißem Hintergrund nicht sieht (der sowohl bei geocaching.com als auch bei den üblichen Bildbetrachtern üblich ist). Vor einem anders farbigen erkennt man vielleicht, was vorher nicht zu sehen war.

Handelt es sich um ein bekanntes Bild oder ist das Bild vielleicht zwei Mal im Listing, lohnt es sich, nach Unterschieden zu suchen. Sollten die nicht optisch offenkundig sein, kann man in einem Bildbearbeitungsprogramm, welches Ebenen beherrscht (gimp zum Beispiel mal wieder als Vertreter der Freeware-Fraktion oder natürlich der fast-alles-Könner Photoshop), diese beiden Bilder als solche übereinander legen. Dann das oben liegende Bild durchscheinend (transparent) machen, vielleicht noch etwas an den Kontrast- und Transparenzreglern spielen oder den Modus auf Abziehen stellen und gucken, ob der Zauber wirkt.

Kann man auf dem Bild nichts erkennen außer einer wirren "Tapetengrafik" eines Designers unter LSD, könnte es sich um ein [Stereogramm](#) handeln. Mit ausreichend Training, etwas schielen und den passenden Augen lassen sich von vielen Menschen nach einer Weile Dinge hier drin sehen, die ich leider nicht erkennen kann. Ich nehme, wenn ich gerade keinen passenden "Schieler" in der Nähe habe, Software, die mir die verborgene Information im Stereogramm anzeigt. Derer gibt es inzwischen eine wahre Flut, so dass ich mir eine Empfehlung lieber spare. Wer gut mit Bildbearbeitungssoftware umgehen kann, dem mag auch diese hierfür reichen: das Bild in eine neue Ebene kopieren, auf Differenzmodus stellen (in Gimp im Ebenenfenster, Modus, Abziehen). Nun hat man ein schwarzes Bild und kann die neue Ebene so lange hin- und her schieben, bis sichtbar wird, was versteckt sein wollte. Die Verschiebung kann gut und gern 50 Pixel betragen.

Mein vorerst letzter Bildertipp, den ich fast täglich nutze: [tineye](#) und (meist noch ergiebiger) die Google-Bildersuche! Will man bestimmte Informationen zu einem bestimmten Bild, weiß man nicht, wer oder was hier gerade abgebildet ist, diese beiden Dienste leisten großartige Arbeit! Tineye gibt es als Firefox-Plugin, mein absolutes Lieblingsplugin. Dafür kann die google-Bildersuche mehr finden, tineye findet nur, was genau den gleichen Ausschnitt hat, wie das gesuchte Bild. Die google Bildersuche benutzt sich für mich am einfachsten, in dem ich die komplette URL des Bildes bei Google ins Suchfenster packe und im folgenden Fenster oben auf "Passende Bilder finden Sie mit der Bildersuche" klicke.

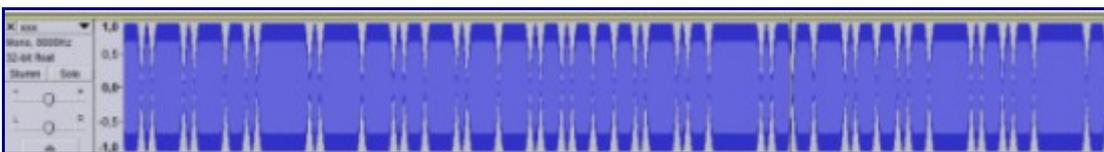
5.3 Musikdateianalyse

Ab und an gibt's im Listing auch was auf oder für die Ohren. Auch in derartigen Medien lassen sich auf unverschämt vielerlei Arten Koordinaten und weiterführende Hinweise verbergen.

Am Einfachsten in den Metadaten, die sich in den meisten Musikplayern, bei Windows über den Date Explorer und Eigenschaften (Rechtsklick auf die Datei) oder auch in einem Hex-Editor (z.B. [HxD](#)) zeigen. Hier lassen sich ähnlich wie bei den exif-Informationen in JPGs Kommentare, Autoren und vieles mehr integrieren. Und ähnlich wie bei den exif-Informationen kann man nicht alles über den einfachen Weg von Date Explorer oder Musikplayer auslesen. Es empfiehlt sich immer mit mehreren Programmen nach zuschauen, am Besten auch über einen Audioeditor (z.B. [Audacity](#)). Hiermit kann man dann gleich weitere Schritte erledigen, so diese nötig werden sollten.

Die versteckten Informationen könnten sich ergeben, in dem man die Datei langsamer abspielt. Oder schneller. Oder rückwärts. Man könnte sie entauschen. All dies lässt sich in Audacity über das Menü „Effekte“ steuern. Man könnte auch die beiden Stereo-Kanäle jeweils einzeln abspielen. Oder die Datei an bestimmten Stellen auseinander schneiden und wieder zusammensetzen.

Immer hilfreich ist ein Blick auf die Wellenform des akustischen Signals. Manchmal kann man dort direkt [Binärcodes](#) ablesen und Morse lässt sich mit einem Blick hier drauf wesentlich einfacher abschreiben, als wenn man nur den Tönen lauschen müsste.

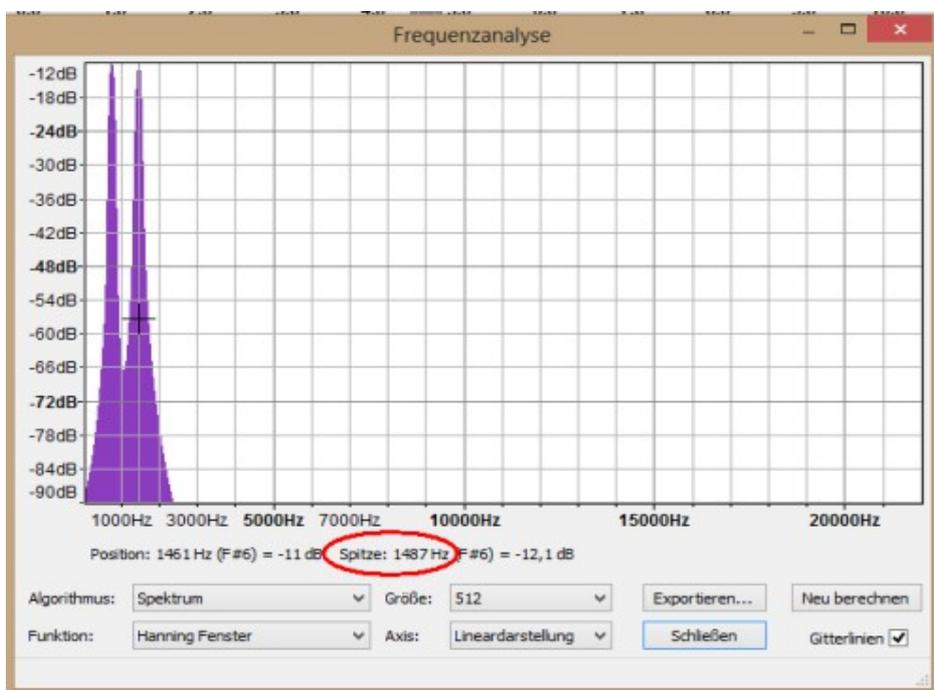


Ebenfalls möglich ist es, die BPM, die Beats per Minute, also das Tempo des Stückes, zur Koordinatenfindung zu nutzen. Diese lassen sich simpel zählen (eine Minute lang den Takt mitschreiben), über eine Unzahl von kleinen Freewareprogrammen auslesen oder über diese [Webseite](#) mit der Maus "zusammenklicken".

Sollten euch Töne unterkommen, die der Rufnummernübermittlung eines Tastentelefon ähnlich klingen, habt ihr es vermutlich mit [DTMF](#) (Dual-Tone-multi-frequenzy) zu tun. Mehrfrequenzverfahren bedeutet, jeder dieser Töne hat zwei Frequenzen. Anhand dieser beiden Frequenzen lassen sich über eine Tabelle die gesuchten Tasten ermitteln:

	1209 Hz	1336 Hz	1477 Hz	1633 Hz
697 Hz	1	2	3	a
770 Hz	4	5	6	b
852 Hz	7	8	9	c
941 Hz	*	0	#	d

Finden sich zum Beispiel die Frequenzen 1336 und 770 Hz, ist die Taste 5 gedrückt worden. Mit Audacity lassen sich die Frequenzen ermitteln, in dem man den gewünschten Bereich der Sounddatei im Wellenprofil markiert und dann oben im Menü „Analyse“ und „Frequenzanalyse“ wählt. Nun einfach mit der Maus über die beiden „Türme“ gehen und unter dem Graphen ablesen, was dieser für einen Spitzenwert hat. Wer diese Stelle etwas einfacher treffen möchte, kann vorher noch bei „Axis“ auf „Log. Darstellung“ klicken



Wer es einfacher haben möchte, kann eine solche Sounddatei auch über eine [Webseite](#) auslesen oder diversen Apps für's Handy (beispielsweise dem "PhoneTone_Extractor") vorwerfen. Wobei man Koordinaten über Frequenzen auch ohne offensichtliche Telefontöne in einer Sounddatei verstecken kann. Ein Blick auf die verwendeten Frequenzen schadet sicher nie.

Brachte dies alles noch keine Erkenntnis, handelt es sich vielleicht um [software-steganographisch](#) verstecktes? Dann bleibt wohl nur das Ausprobieren der hierfür in Frage kommenden Stegaographie-Programme, z.B. [mp3stego](#), [steghide](#), [OpenPuff](#), [Stealth Files 4.0](#) oder [MP3Stegz](#). Als mögliche Passwörter eignen sich wie so oft der Cachename, der Owner, der GC-Code oder irgendwas aus dem Listing, was hervorsteicht.

Und wie fast immer ist das hier erklärte nur ein winziger Teil der Möglichkeiten, wie man in Sounddateien Informationen verbergen kann.

6. Verschlüsselungen

6.1.1 Monoalphabetische Substitution

Monoalphabetische Substitution meint eine Verschlüsselung, bei der jeder Buchstabe oder jede Buchstabengruppe durch genau einen Buchstaben, eine Buchstabengruppe oder ein Zeichen ersetzt wird. Es gibt also genau ein Schlüsselalphabet.

Der Vorteil hier liegt in der Einfachheit des Ver- und Entschlüsselns. Der Nachteil in der Möglichkeit der Häufigkeitsanalyse und des "logischen" Entschlüsselns. Je länger der verschlüsselte Text ist, um so einfacher ist es zu raten, welcher häufig vorkommende Buchstabe des Geheimtextes wohl dem zum Beispiel im Deutschen am meisten verwendeten E entspricht. Tabellen zur [Buchstabenhäufigkeit](#) und Tools im Internet ([Crypt-Online](#) oder [kas-bc.de](#) oder zum Herunterladen der [Code-Brecher](#)) machen dies sehr einfach. Und je simpler ein Text aufgebaut ist, je wahrscheinlicher bestimmte Wörter (Nord, Ost, Cache, ausgeschriebene Zahlen, Punkt, Grad), Phrasen oder Wortteile vorkommen, um so einfacher lassen sich monoalphabetische Chiffres auch von Laien mit einem Zettel, einem Stift und etwas Zeit entschlüsseln.

Cesar-Chiffre

Der erste, bis heute bekannte Nutzer einer monoalphabetischen "Geheimschrift" war Julius Caesar, der einfach das Alphabet um drei Stellen verschoben hat. Die 3 entspricht dem Buchstabenwert von dem C aus Caesar. Aus dem A wurde so also ein C, aus dem B ein D, aus dem C ein E. Aus dem Wort *Kryptologie* wird somit das Kaudawelsch: *Nubswrorjlh*.

Rot13, Rot5, RotX

Natürlich kann man auch jede andere der 25 möglichen Alphabetverschiebungen nehmen. Diese werden meist ROT für Rotation abgekürzt. Weiterhin lassen sich auch Zahlen und Sonderzeichen hinzunehmen, wobei dann besser irgendwie definiert sein sollte, welche Reihenfolge dem Klartextalphabet zugrunde liegt. ROT5 nur mit mit Ziffern ist mir beim Cachen aber schon häufiger begegnet. Aus 1 wird somit 6 oder, auf der 10er-Achse gespiegelt, aus 1 wird 9, aus 2 wird 8, aus 3 wird 7,...

Es gibt immens viele Webseiten, die einem das manuelle Entschlüsseln dieser Rotationschiffres abnehmen. Sogar welche, auf denen alle 25 Alphabet-Möglichkeiten mit einem Klick dargestellt werden. Sehr hilfreich, wenn man nicht weiß, um wie viele Buchstaben das Alphabet denn nun verschoben worden ist.

Ein solcher Verschiebechiffre als monoalphabetischen Substitutionschiffres ist gleich doppelt schön für denjenigen, der es entschlüsseln möchte, da man, wenn man erstmal zwei Buchstaben sicher entschlüsselt hat, die anderen 24 gleich mitgeliefert bekommt. Nichts desto trotz galt er noch Jahrhunderte nach Caesar als hinlänglich sicher und wird bis heute gern benutzt. Allerdings weitestgehend nur noch um Geschriebenes nicht auf den ersten Blick lesbar zu machen. Im Falle von Geocaching-Hints mit dem beliebten ROT13 ein lobenswerter "Entspoiler".

Möchte man die Entschlüsselung von monoalphabetischen Substitutionen wenigstens ein bisschen erschweren, tut man gut daran, die verräterischen Leerzeichen und Satzzeichen, aus denen sich typische Wort- oder Satzanfänge oder Endungen erraten lassen, möglichst zu entfernen und vielleicht, um einen schwierigeren Verschlüsselungsansatz vorzutäuschen, den verschlüsselten Buchstabensalat noch in hübsche 5-er-Buchstabengruppen aufteilen. Gegen Häufigkeitsanalysen hilft das zwar auch nicht, aber zumindest verwirrt es kurzzeitig den Entschlüsselnden ;).

Atbasch

Fast so einfach, wie ein Verschiebechiffre ist [Atbasch](#), wobei hier das Alphabet symmetrisch "gespiegelt" wird. Aus A wird Z, aus B wird Y, aus C wird X, usw. Atbasch stammt aus dem hebräischen, daher auch ihr Name, der aus den ersten beiden (Aleph und Beth) und den beiden letzten Buchstaben (Taw und Schin) des hebräischen Alphabets besteht.

Verwürfelte Alphabete mit Schlüsselwörtern

Natürlich kann man auch jede andere Alphabetsverwürfelung benutzen, welche dann immerhin den Vorteil hat, nicht durch reines Verschieben entschlüsselt zu werden, sondern etwas mehr Kopf- und Hand bzw. Rechenarbeit bedeutet. Da eine Entschlüsselung derartig verwürfelter Alphabete immer bedeutet, daß der Empfänger einer solchen Nachricht Kenntnis über das Verschlüsselungsalphabet haben muss, dieses aber natürlich nicht mitgeliefert werden darf (höchstens auf einem anderen Wege), liegt es nahe, zum Erzeugen des Geheimalphabets Schlüsselwörter zu benutzen. Das funktioniert auch bei einfachen, monoalphabetischen Substitutionen. Legt man sich zum Beispiel auf das Schlüsselwort SCHMIERBLOG fest, entfernt nun alle doppelt vorhandenen Buchstaben (praktischerweise sind in SCHMIERBLOG derer nicht vorhanden) und stellt Sie nun dem zu erzeugenden Geheimalphabet vorne an. Buchstaben, die im normalen Alphabet abzüglich der schon mit SCHMIERBLOG verwendeten nun noch übrig sind, werden hinten angehängt.

Und so wird aus dem normalen Alphabet:
ABCDEFGHIJKLMNOPQRSTUVWXYZ

Das Geheimalphabet "Schmierblog"
SCHMIERBLOGADFJKNPQTUVWXYZ

Mit diesem Geheimalphabet wird nun aus der *NINA* eine verschlüsselte *FLFS*.

Je länger das Schlüsselwort ist, um so besser, da weniger Buchstaben mit sich selbst "verschlüsselt" werden müssen. im Beispiel Schmierblog bleiben alle Buchstaben ab dem T sie selbst. Es ist daher keineswegs unüblich, den hinteren Teil des Schlüsselalphabets noch einmal umzudrehen, also nach dem Schlüsselwort (SCHMIERBLOG) mit dem Z das Alphabet aufzufüllen.

Statt
SCHMIERBLOGADFJKNPQTUVWXYZ

erhält man:
SCHMIERBLOGZYXWVUTQPNKJFDA

Die multiplikative Substitution

Auch eine Variante der monoalphabetischen Substitution, bei der das Alphabet durchgewürfelt statt nur verschoben wird, ist die [multiplikative Substitution](#). Hierbei wird jedem Buchstaben des Klartextalphabetes gemäß seiner Position im Alphabet die entsprechende natürliche Zahl zugeordnet (A=0, B=1,...). Multipliziert man den Wert eines jeden Klartextbuchstaben mit einer frei wählbaren Zahl und ersetzt diese Zahl nun wieder mit dem Buchstaben des Alphabets (A=0, B=1,...), entsteht ein neues Geheimentalphabet.

Nehme ich die 7 als Multiplikator, erhalte ich folgendes Geheimalphabet:

ABCDEFGHIJKLMNOPQRSTUVWXYZ
AHOVCJQXELSZGNUBIPWDKRYFMT

Wobei das A aus A=0, 0 mit 7 multipliziert = immer noch 0, also das A ein A bleibt (bei der Zählweise A=0 wird dies immer der Fall sein).

Das B mit dem Wert 1 multipliziert mit 7 ergibt die 7, welches dem Buchstabenwert von H entspricht.

Das F mit dem Wert 5 multipliziert mit 7 ergibt 35. Bisschen zu viel für das 26-Zeichen-Alphabet, daher rechnet man 35 modulo 26. Es ergibt sich ein Rest von 9, was wiederum dem J entspricht.

Playfair

Ebenfalls mit einem Schlüsselwort arbeitet die [Playfair-Verschlüsselung](#). Keine reine monoalphabetische Substitution, sondern eine "bigraphische, monoalphabetische". Soll heißen, hierbei wird jedes Buchstabenpaar des zu verschlüsselnden Textes durch ein anderes Buchstabenpaar ersetzt. Dafür wird das Alphabet in ein 5*5er Raster gelegt (I=J, sonst passt es nicht), das Schlüsselwort (um doppelte Buchstaben bereinigt) vorn angestellt und mit dem Rest des Alphabets aufgefüllt. Zum Verschlüsseln werden die Buchstaben nun in diesem Quadrat nach zwei Regeln vertauscht:

1. liegen die zu verschlüsselnden Buchstabenpaare in einer Zeile oder einer Spalte, wird der jeweils nächste (untere oder rechte) Buchstabe benutzt
2. liegen die zu verschlüsselnden Buchstabenpaare in unterschiedlichen Zeilen oder Spalten, nimmt man den Buchstaben in der selben Zeile aber der Spalte des jeweils anderen Klartextbuchstaben.

Begegnet euch also mal ein Code wie dieser:

UE QP XY XK KE EN BC RV HL

Probiert mal euer Glück mit diesem Quadrat

K	R	Y	P	T
O	G	A	I	E
B	C	D	F	H
L	M	N	Q	S
U	V	W	X	Z

Oder auf einer Seite wie [Crypt-Tool](#):

Das verwendete Schlüsselwort lautet: Kryptographie.

Aber auch die Playfair-Verschlüsselung, immerhin schon eine schwieriger als eine einfache, monoalphabetische Methode, ist noch relativ leicht zu knacken. Auch wenn bei ihr Buchstaben

nicht immer die gleichen ergeben, sind doch die selben Buchstabenpaare immer durch die selben Chiffrebuchstaben verschlüsselt.

Albertis Chiffrierscheibe

Das sich monoalphabetisch Verschlüsseltes leicht entschlüsseln lässt, wusste man schon vor etwa 400 Jahren und so hat der Herr [Alberti](#) den vernünftigen Einfall gehabt, man könne statt eines einzigen Schlüsselalphabets mehrere benutzen und zwischen diesen nach einer bestimmten Anzahl von Buchstaben oder Wörtern wechseln. Das macht eine Häufigkeitsanalyse nicht unmöglich aber, da man erstmal den Schlüssel für den Alphabetswechsel benötigt, doch zumindest schwieriger. Und damit man diesen Alphabetswechsel schnell vollziehen kann, gab es eine praktische [Chiffrierscheibe](#) von ihm. Wie recht der Herr Alberti mit dem Alphabetswechsel hatte, sieht man daran, daß auch die "Königin" unter den Verschlüsselungsmaschinen, die [Enigma](#), genau nach diesem Prinzip arbeitet. Sie wechselt allerdings sogar nach jedem Buchstaben das Schlüsselalphabet.

6.1.2 - Verschlüsselungen - Geheimtexte manuell entschlüsseln

Einfache, monoalphabetische Verschlüsselungen, bei dem jedem Buchstaben des Alphabets einfach ein anderer Buchstabe, ein anderes Zeichen oder eine Zahlenkette zugewiesen wird, lassen sich in relativ kurzer Zeit Mithilfe von Zettel, Papier und einer Tabelle für [Buchstabenhäufigkeiten](#) der jeweiligen Sprache sowie ein wenig Zeit entschlüsseln.

Grundsätzlich gilt: je länger der Geheimtext ist, um so einfacher ist es, ihm über Buchstabenhäufigkeiten, Worthäufigkeiten, Wortendungen und sprachliches Geschick auf die Schliche zu kommen.

Hierfür muss man als erstes die Zeichen des Geheimtextes zählen und nach Häufigkeit sortieren. Hat man (je nach Textlänge) etwa 20-27 unterschiedliche Zeichen, hat man mit ziemlicher Sicherheit ein einfach verschlüsseltes Alphabet (26 Zeichen) plus Leerzeichen und möglicherweise noch ein oder zwei Interpunktionszeichen (Punkt, Komma und für die Geocacher vielleicht noch ein Gradzeichen). Sind es ungefähr 55 Zeichen, sind möglicherweise Groß- und Kleinschreibung verwendet worden, gegebenenfalls auch deutsche Umlaute. Kommen noch etwa 10 Zeichen oben drauf, könnten sich auch noch Ziffern im Text befinden.

Natürlich gibt es im Internet willige Helfer, die einem das Zählen und Sortieren und in vielen Fällen sogar noch das Entschlüsseln abnehmen. Eine Häufigkeitsverteilung kann man sehr hübsch bei [cryptool-online](#) und [kas-bc.de](#) erledigen lassen.

Auf der [Cryptool-Seite](#) findet sich, wie auch im [Wikipedia-Artikel](#), die Tabelle mit Buchstabenhäufigkeiten der deutschen und englischen Sprache. Ebenfalls findet sich dort das „Häufigkeitsgebirge“, was sehr hilfreich ist, wenn man Rotations-Chiffren optisch erkennen möchte. Also die Form von Buchstabenverschlüsselungen, bei denen das Alphabet nur um x Stellen verschoben wird. Bei Caesar und seinem Code waren es 3 (aus einem "A" wird ein "C", aus einem "B" ein "D", ...), heute wird sehr häufig die 13 benutzt (ROT13), was den Charme hat, daß man mit einem weiteren Sprung um 13 Zeichen im Alphabet wieder beim Ausgangstext angekommen ist. Entschlüsseln und Verschlüsseln sind somit auf gleichem Wege möglich. Wobei zu betonen ist, daß eine Rotationschiffre keineswegs eine irgend geartete Verschlüsselung ist, also nie benutzt werden

sollte, um wirklich geheimzuhaltende Informationen zu verschleiern. Es ist eher eine Spielerei, bei der ein Text nicht sofort lesbar ist.

Ergibt die Zeichenverteilung ähnliche „Ausschläge“, wie die Buchstabenverteilung in den Tabellen? Es sollten ein bis zwei Zeichen sehr häufig vorkommen, das Leerzeichen (sofern es überhaupt verschlüsselt worden ist) ist üblicherweise das häufigste Zeichen. Auf dieses kann aber auch verzichtet werden, dann sind die Wörter nicht mehr so leicht lesbar. Dem Leerzeichen dicht auf den Fersen ist der Buchstabe „E“, der ungefähr 17% der Buchstaben in durchschnittlichen Deutschen Texten besetzt.

Sogar in meiner sehr kleinen Beispielverschlüsselung stimmt die relative Häufigkeit des „E“'s.

Der Geheimtext:

tuxjklaktlfckokotyfckojxkobokxlaktl

hat zwar nur 12 verschiedene Buchstaben, das liegt aber daran, daß er so kurz geraten ist. Der häufigste Buchstabe ist das „K“ mit ungefähr 20%. Nehmen wir an, daß dies das E ist, haben wir vielleicht schon ein Fünftel des Geheimtextes entschlüsselt und vor allem ein Ansatzpunkt für sprachliches Geschick und typische Buchstabenverbindungen oder Wortendungen.

So gibt es neben den Tabellen für die Buchstabenhäufigkeiten auch welche mit den häufigsten Buchstabenendungen. Hier führen: „en, em, es, el und er, st, ing, sam, bar, lich, ung, heit, keit“.

Auch interessant sind die häufigsten Bigramme, also zusammen auftretende Buchstabenpaare: „en, er, ch, ck, (wobei c alleine fast nie vorkommt), te, de, nd, ei, ie, in, es“. Und Trigramme (die drei am Häufigsten aufeinander folgenden Buchstaben): „ein, ich, nde, die, und, der, che, end, gen, sch“.

Einen weiteren Blick sollte man auf die im Deutschen am [Häufigsten verwendeten Wörter](#) werfen. Diese Hitliste führen „der, die, und, in, den, von, zu, das, mit, sich, des, auf, für, ist und im“ an. Für Geocacher verändert sich diese Hitliste vermutlich ein wenig, wodurch die Wörter „Nord, Ost, Grad, Cache, Koordinaten, Dose, suchen, Versteck“ sowie die ausgeschriebenen Ziffern: „eins, zwei, drei, vier, fuenf/fünf, sechs, sieben, acht, neun und null“ weiter nach oben rutschen.

Übrigens gibt es im Deutschen eigentlich keine Ein-Buchstaben-Wörter – was deutsche von englischen Texten stark unterscheidet.

Zum manuellen Entschlüsseln einfacher Geheimtexte nutze ich einen Texteditor (das kostenlose Notepad++). Genauso gut funktioniert ein beliebiges Textverarbeitungsprogramm, wobei als Schriftart eine gewählt werden muss, deren Buchstaben feste Breiten haben (zum Beispiel Courier New oder monospace). So lassen sich Geheimtext und der Entschlüsselungsversuch direkt untereinander platzieren.

Im Beispiel von eben:

tuxjklaktlfckokotyfckojxkobokxlaktl

Davon ausgehend, daß der Text ohne Leerzeichen verschlüsselt worden ist (da es nur einen sehr häufigen Buchstaben gibt) und der häufigste tatsächlich das „E“ ist, schreibe ich diese Erkenntnis unter den Geheimtext.

```
tuxjlahtlfcokotyfcokjxkobokxlahtl
-----e-----e-----e-----e-----e-----e-----
```

Lesbar ist das leider noch nicht. Aber vielleicht funktioniert hier ja das, was schon vielen historischen Geheimtexten das Genick gebrochen hat: vielleicht kann man hier ja raten, wie der Text anfängt, oder welche Wörter drin enthalten sind. Noch im zweiten Weltkrieg sind viele, eigentlich fast sichere Chiffres geknackt worden, weil typische Grußfloskeln, die immer gleichen Phrasen und leicht zu erratende Wörter verwendet worden sind.

In unserem Fall, einem für Geocaching typischen Verschlüsselung, gehen wir mal davon aus, dass es sich hierbei um eine Koordinatenangabe handelt. Diese beginnt üblicherweise mit Nord oder N. Das „T“, der erste Buchstabe des Geheimtextes macht 12% von diesem aus, was mit der üblichen statistischen Häufigkeit von etwa 10% vom „N“ gut zusammen passt. Probieren wir es aus:

```
tuxjlahtlfcokotyfcokjxkobokxlahtl
n-----en--e-e-n--e--e--e--en-
```

Naja, wirklich lesbar ist es noch nicht, also weiter. Aber wie? Man könnte jetzt weitere Buchstaben raten. Das zweit häufigste im Geheimtext ist ein „O“, an dritter Stelle steht das „L“. Nehmen wir die Buchstabenhäufigkeitstabellen, sind die Buchstaben E N I S R und A am Häufigsten. Somit dürfte O und L einer von denen sein. Da „E“ und „N“ vermutlich schon gefunden wurden, fehlt ja nur noch I, S, R und A.

Man könnte auch Wörter raten. Fängt der Geheimtext wirklich mit Nord an? Dann wäre das „U“ um Geheimtext ein „O“ und noch hilfreicher wäre das „X“, welches im Geheimtext gleich drei Mal vor kommt, und einem „R“ entspräche.

Der Text endet mit en und einem weiteren, noch unbekanntem Buchstaben. Was könnte hier eine plausible Endung sein? Ist es eine ausgeschriebene Ziffer? Welche endet denn mit en und einem weiteren Buchstaben? Dann wäre fuenf oder fünf ein passender Kandidat. Hat der Verschlüsseler die ue-Schreibweise gewählt um keine deutschen Umlaute zu chiffrieren, müsste, wenn es wirklich die fuenf ist, der fünfte Buchstabe von hinten dem letzten entsprechen. Bingo! „lahtl“ sind die letzten fünf Buchstaben. Und das bedeutet mit Sicherheit „fuenf“.

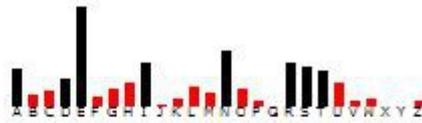
Wir könnten aber auch den Ansatz des Häufigkeitsgebirges wählen. Falls es sich bei dem Geheimtext nur um eine ROT-Verschiebung (Alphabetsverschiebung um x Stellen) handelt, sollten Auffälligkeiten hier vielleicht sogar mit so wenigen Buchstaben schon sichtbar sein.

Normales Alphabet

Statistische Daten:

Varianz: 14.50603

StdAbw: 3.80868



Vorliegendes Alphabet

Statistische Daten:

Varianz: 29.79054

StdAbw: 5.45807



Und tatsächlich, die großen Balken scheinen sich in ähnlichem Abstand oben und unten zu wiederholen. Der E, I und N im normalen Alphabet könnten den Balken K, O und T im Geheimtext entsprechen. Dieses „Häufigkeitsgebirge“ wäre wesentlich aussagekräftiger, wäre der Geheimtext länger. Aber auch bei dem kurzen Schnipsel könnte es reichen und wir sehen eine Verschiebung um 6 Buchstaben. Dies ist auch der Vorschlag, den cryptool-online uns hier machen würde, wenn wir auf den passenden Knopf "ROT-Check" klicken würden.

Und eigentlich hätte man dies schon ein paar Schritte vorher ausprobieren können, da der Buchstabe „E“ (im Geheimtext „K“) und „N“ (im Geheimtext „T“) ja schon erraten worden sind. Beide sind um 6 Buchstaben verschoben, somit kann man immerhin schon hoffen, dass alle Buchstaben um 6 verschoben worden sind.

Aber egal welchen Weg wir wählen, mit ein bisschen Übung braucht es nur Minuten, um aus dem Geheimtext

tuxjlaktlfckokotyfckojxkobokxlaktl

den Originaltext

nordfuenfzweieinszweidreivierfuenf

zu erhalten.

Auch wenn das Alphabet komplett verwürfelt worden sind (anstatt um x Stellen zu verschieben), dauert die Entschlüsselung nur etwas länger. Hilfreich ist immer der einfache Ansatz, über Häufigkeitsanalysen das Leerzeichen und das „E“ zu identifizieren. In längeren Texten ist auch immer nach Punkt und Komma zu suchen, die nie an einem Wortanfang stehen aber immer von einem Leerzeichen verfolgt werden. Als nächstes sollte man versuchen, die kurzen Wörter zu entschlüsseln (der, die, das, und, in, im, ...) und nach identischen Textpassagen suchen, die gleiche Worte oder gleiche Wortteile bedeuten. Gleiche Geheimtextzeichen hintereinander sind auch ein schöner Ansatzpunkt, da im Deutschen nur bestimmte Buchstaben doppelt auftauchen und diese oft von gleichen oder ähnlichen Buchstaben umschlossen werden. Doppelte Konsonanten haben immer Vokale, doppelte Vokale immer Konsonanten um sich herum. Und natürlich sollte man immer nach typischen Grußformeln (Lieber Cacher, ...) und Abschiedsworten (Viel Spaß bei der Suche) schauen.

Funktioniert der Ansatz mit der Häufigkeitsanalyse nicht, sticht also kein Zeichen des Geheimtextes auffällig heraus, dann ist es keine monoalphabetische Verschlüsselung sondern möglicherweise eine wiederholte Buchstabenverschiebung, bei der alle x Zeichen das Schlüsselalphabet gewechselt worden ist. Auch das lässt sich mit etwas Mühe per Hand entschlüsseln, ist aber definitiv schon wesentlich anstrengender. Wichtig ist hier über gleiche Geheimtextstellen herauszufinden, nach wie vielen Buchstaben das Alphabet - und wie oft - wechselt.

Ein weiterführender Link für das Decodieren von Geheimtexten mittels einer Tabellenkalkulation und dem Umrechnen und Vergleichen von Ascii-Werten findet sich auf dem [Mathebord](#).

6.2 -Polyalphabetische Verschlüsselung

Die Verschlüsselungsmaschine Enigma



Die [Enigma](#) (vom griechischen Wort "ainigma" = Rätsel) ist der unangefochtene Star unter den Verschlüsselungsmaschinen des zweiten Weltkrieges. Nicht weil sie die Beste, sondern weil sie die mit Abstand bekanntest ist. Wobei sie wirklich gut und beinahe nicht zu knacken gewesen ist.

Rein technisch ist Enigma eine Rotor-Schlüsselmaschine, bei der jeder Buchstabe des Klartextes mit einem anderen Schlüsselalphabet verschlüsselt wird. Sie beherrscht also die polyalphabetische Substitution, die damit dank der maschineller Unterstützung ab ca. 1920 einfach und – theoretisch - für Jedermann sicher anwendbar war. Relativ sicher zumindest, sofern man bestimmte Regeln beachtete und auch nur, solange es noch keine Maschinen gab, die beim Entschlüsseln helfen.

Arthur Scherbius hat die Enigma erfunden. Etwa zeitgleich mit ihm sind in anderen Ländern ähnliche Geräte patentiert worden. Anfangs war sie für zivile Zwecke konzipiert worden. Da der erste Weltkrieg aber einen deutlichen Mangel an sicheren und dennoch einfach zu bedienenden Verschlüsselungsmethoden deutlich gemacht hatte, zeigte das Militär recht bald Interesse an den Geräten des Herrn Scherbius. Dieser konnte den Siegeszug seiner Erfindung leider gar nicht mehr genießen, er starb 1929 an den Folgen eines Verkehrsunfalls (mit einem Pferdefuhrwerk!). Dafür blieb es ihm auch erspart mitzerleben, dass sein Baby eben doch nur FAST unknackbar gewesen ist. Letztlich schreiben viele Historiker, dass Enigma kriegsentscheidend gewesen ist. Vor allem des Umstandes wegen, daß in den letzten Kriegsjahren viele deutsche Funkprüche vom Feind mitgelesen werden konnten. Dank dem Geschick der Alliierten, was die Nutzung dieser Informationen anging, blieben die Deutschen bis zum Kriegsende von der Sicherheit ihrer Verschlüsselung überzeugt. Ein in meinen Augen sehr spannendes Stück Zeitgeschichte, was wohl auch den Reiz der Enigma als Rätselinhalt erklären dürfte.

Ein bisschen zur Technik

Es gab im Laufe der Jahre eine Fülle an Enigmas. Die bekanntesten (und beim Geocachen auch am

häufigsten benutzten) sind die Enigma I, Enigma M3 und die Enigma M4. Die Zahlen drei und vier beziehen sich hier auch auf die Anzahl der Walzen, die sich mit jedem verschlüsselten Buchstaben um eine Position weiter drehen und jeweils eine neues Verschlüsselungsalphabet erzeugen. Die M4 war somit sicherer als ihre kleine Schwester mit nur drei Walzen und wurde bei der Marine zur Kommunikation mit den U-Booten benutzt.



Neben den Walzen (austauschbar, bis zu 8 verschiedene normale plus Beta und Gamma als vierte Walze bei der M4) hatten die hier relevanten Enigmas noch eine Umkehrwalze (UKW-B oder C), außerdem eine Tastatur, ein Lampenfeld, was den ver-/bzw. entschlüsselten Buchstaben darstellte und ein Steckbrett, mit dem die Buchstaben noch einmal paarweise verwürfelt worden sind; wodurch die Sicherheit der Verschlüsselung noch einmal enorm verstärkt worden ist.

Die genaueren technischen Details spare ich mir hier, da es schon mehr als genug wirklich gute Erklärungen, Bau- und Schaltpläne sowie mathematische Betrachtungen zu Schlüsselstärken und Entschlüsselungsalgorithmen im großen, weiten Web gibt.

Ich helfe dafür hoffentlich in ausreichender Kürze dabei, mit den im (Geocaching-) Rätsel vorhandenen Informationen den verschlüsselten Text in Klartext zu übersetzen.

Die Enigma entschlüsseln nur echte Freaks „zu Fuß“, die letzten dürften es vor etwa 60 Jahren getan haben. Weniger tapfere Naturen bedienen sich heutzutage der dafür reichlich vorhandenen Software. Empfehlen kann ich den Download von Dirk Rijmenants hervorragender [Enigma-Simulation](#). Ebenfalls empfehlenswert sind diese online-Varianten der [Sternenhimmelsstürmer](#) und [Enigmaco](#). Und vermutlich viele weitere, die ich nicht näher kennengelernt habe.

Und wie geht das nun?

Enigma kann nur Großbuchstaben, keine Ziffern oder Satzzeichen verschlüsseln. Letztere wurden einfach durch ein X ersetzt, Ziffern ausgeschrieben. Eigennamen wurden üblicherweise verdoppelt und mit X umschlossen. Außerdem das „ch“ durch den Buchstaben Q ersetzt. Anschließend wurde der Text in Fünfergruppen dargestellt und nun verschlüsselt.

Dazu benötigte man den Tagesschlüssel, der die Grundstellung der Enigma beinhaltete. Also die Angabe, welche der Walzen benutzt wird (die sogenannte Walzenlage I, II, III, ... sowie Gamma und Beta, falls es eine M4 war), welche Umkehrwalze (UKW-B oder -C), die Grundstellung des inneren Rings dieser Walzen (Ringstellung, Walzenstellung, entweder in Buchstabenwerten A=1, B=2, ... oder in Buchstaben ausgedrückt), sowie die Buchstabenvertauschungen (Steckerverbindungen) vom Steckbrett.

Zum Entschlüsseln werdet ihr irgendwo diese Informationen finden, die seinerzeit in der Realität und heutzutage, wenn wir damit spielen, gerne auf Monatsblättern, von unten nach oben aufsteigend, dargestellt werden. Von unten nach oben, damit man den Schlüssel vom Vortag abscheiden und wegwerfen kann. Gedruckt wurden diese Blätter häufig auf Löschpapier, damit man sie bei Bedarf einfach vernichten kann.

Beispiel-Tagesschlüssel (Enigma I):

Tag	UKW	Walzenlage	Ringstellung	Steckerverbindungen	Kenngruppe
31	B	III I IV	16 05 09	AB CD EF GH IJ KL MN OP QR ST	ABC SDF
30	C	II V I	15 11 10	BN DZ EP FX GT HW IY OU QV RS	YXC QWE
29	C	I IV III	12 18 22	CY EL FH GS IJ KQ MW PV RZ TU	ZUI JKL

Es gab viele verschiedene Tagesschlüssel, je nach Anwendungsgebiet bzw. Empfängergruppe. Damit der Entschlüsselnde sich sicher sein konnte, dass die Nachricht für ihn bestimmt war und er sie in Klartext umwandeln konnte, gab es häufig beim Tagesschlüssel noch eine (ebenfalls täglich wechselnde) Kenngruppe. Diese wurde dann der verschlüsselten Nachricht unverschlüsselt vorne angestellt.

Das alles wirkt zwar schon ausreichend wirr, reichte als Sicherheit aber noch nicht, da mit dieser Methode eine große Anzahl von Funksprüchen eines Tages auf die gleiche Art verschlüsselt worden wären. Also musste der Verschlüsselnde sich häufig noch die äußere Walzenstellung (den Spruchschlüssel) selber ausdenken und ihn mit einer ebenfalls selbst ausgedachten Buchstabenfolge verschlüsseln.

Die Enigma stellte er dann (innen) in die Grundstellung des Tages laut Tagesschlüssel, wählte die drei (oder vier) Buchstaben des selbst ausgedachten Spruchschlüssels an den äußeren Walzenstellungen und tippte seine drei ausgedachten Buchstaben. Die Verschlüsselungsmaschine verschlüsselte diese und lieferte als Antwort drei andere Buchstaben. Dieser so verschlüsselte Spruchschlüssel wird zusammen mit der gewählten Grundstellung der Enigma Nachricht – unverschlüsselt – vorangestellt.

Eine korrekt verschlüsselte Enigma-Nachricht hatte im Kopf die Uhrzeit (als vierstellige Zahl, z.B. 1130 für halb zwölf Uhr), die Zeichenlänge des Funkspruches, die Grundstellung sowie den damit verschlüsselten Spruchschlüssel. Anschließend folgte die verschlüsselte Nachricht in Fünfergruppen, gegebenenfalls mit der vorangestellten dreistelligen Kenngruppe, die um zwei Füllbuchstaben zu einer üblichen Fünfergruppe aufgestockt worden ist.

Ein Beispiel zum Nachspielen und Verstehen

1. Verschlüsseln

Nimmt man als Beispiel den oberen Tag 29 und stellt die Enigma M3 auf folgende Grundstellung innen (der Bereich, der sich nur täglich einmal änderte):

UKW-C

Walzen Nummer *I IV III*

Ringstellung der Walzen *12 18 22*

Steckerverbindungen *CY EL FH GS IJ KQ MW PV RZ TU*

Außen (der Bereich, den der Verschlüsselnde sich selber jedesmal neu ausdenken musste) die Grundstellung *PLR* und tippt nun den ausgedachten Spruchschlüssel: *NVD*

Die Enigma gibt als Antwort ein *WGT* zurück.

Der somit als *WGT* verschlüsselten Spruchschlüssel wird dem Empfänger zusammen mit der zufällig gewählten Grundstellung *PLR* im Kopf des Nachricht mitgeteilt.

Verschlüsselt wird Text des Funkspruches dann mit dem gewählten Spruchschlüssel, hier also *NVD*, auf den die Walzen (außen) eingestellt und der Text eingegeben wird. In meinem Beispiel lautet dieser:

Hier wäre ich ohne Geocaching nie hingekommen!

Also in Enigma-Schreibweise:

HIERW AEREI CHOHN EGEOC ACHIN GNIEH INGEK OMMEN X

Verschlüsselt schaut er so aus:

JOPVV QKJZS FNXNJ RUMXT NLQGQ RPEPJ HTLGI SKWLT Z

Das ergäbe nun mit einem korrekten Schlüsselkopf folgenden Funkspruch:

2333 55 PLR WGT

YZUIX JOPVV QKJZS FNXNJ RUMXT NLQGQ RPEPJ HTLGI SKWLT Z

2. Entschlüsseln

Der Empfänger guckt nun zuerst ob er die Kenngruppe (irgendwo in den ersten 5 Buchstaben) in seinem Tagesschlüssel hat. *ZUI* gibt es dort, also kann er die Nachricht entschlüsseln. Er stellt die Enigma passend ein

C I V III 12 18 22 C Y E L F H G S I J K Q M W P V R Z T U

und erhält den noch fehlenden Spruchschlüssel, in dem er die Walzen auf *PLR* stellt und *WGT* aus dem Kopf der Nachricht tippt. Die Enigma antwortet mit dem unverschlüsselten Spruchschlüssel *NVD*. Dieser (*NVD*) wird nun auf der (äußeren) Walzenlage eingestellt und der verschlüsselte Text (ohne die ersten fünf Zeichen mit der Kenngruppe) eingegeben.

JOPVV QKJZS FNXNJ RUMXT NLQGQ RPEPJ HTLGI SKWLT Z

Und schwuppdiewupp (*hüstel*) erhält man die unverschlüsselte Variante zurück:

HIERW AEREI CHOHN EGEOC ACHIN GNIEH INGEK OMMEN X

Und wer bis hier erstmal etwas verwirrt ist, kann sich beruhigt zurück lehnen: das geht wohl den meisten so. Am Besten spielt man dies Szenario tatsächlich einfach einmal nach und tröstet sich anschließend mit dem Gedanken, daß viele Enigma-Caches gar nicht die komplizierte Variante mit einem verschlüsselten und im Kopf der Nachricht mitgesendeten Spruchschlüssel arbeiten, sondern sie verschlüsseln plump ihren Text und geben dann die ganzen Daten inklusive Grundstellung heraus. Da muss man dann nur die Enigma dementsprechend einstellen, den verschlüsselten Text hineinwerfen und erhält die entschlüsselte Nachricht zurück.

Ebenfalls einwerfen muss ich aber noch den Hinweis, dass sich im Laufe der (Kriegs-)Jahre der Umgang mit der Enigma, den Verschlüsselungen und ihren Regeln, gern auch abhängig von den

jeweiligen Nutzergruppen und Enigmaversionen, geändert hat. Somit ist das hier dargestellte noch nicht ganz die einzige Wahrheit, aber wenn man das Grundprinzip erstmal verstanden hat, ist der größte Schritt in Richtung Entschlüsselung sicher schon getan!

Nochmal, nochmal?

Wer jetzt noch etwas üben möchte, dem hab ich etwas vorbereitet ;) :

Tagesschlüssel Kriegsmarine, M4 vom 17.6.2013

UKW B

Walzenlage: *Gamma, VIII, V, VI*

Ringstellung: *15, 5, 16, 12*

KENNGRUPPEN: *liu aer vpu*

AN BO DQ FS GT HU IV LY ER

Funkspruch

1141 183 ASDFJHGF

XAER QRHG REYX BXWB MTBY VDWF BGOB XWVW TPEX EKTZ ZLTC OFWS BQEJ

UNLQ ZTMT ELIO FSHM HXHU WSZP EXHQ XMHN ZDJA ERZD WJBD DCJD UFLH

WCQR EIXA PHPR QLAH OUAK VDEE FUCF YHGD PKPC GBRJ URXJ TIV

Viel Erfolg! :)

7 Logikrätsel

(noch in Bearbeitung)

8 - Weiteres

8.1 - Barcodes

Strichcodes kennen wir alle. Schon seit einigen Jahrzehnten prangen die Balkencodes auf allen Waren um an der piependen Kasse zu verraten, zu welcher Ware sie gehören. Mit der Verbreitung von Smartphones bekamen wir mit den quadratischen, schwarz-weißen Data-Matrix-Kästchen einen weiteren Barcode in unserem Alltag, der inzwischen allerorts auf Werbetafeln und in Zeitungen weiterführende Links und Informationen verspricht (aber dieses Versprechen aber leider oft nicht einlöst - meist verpassen die Ersteller dieser Codes es, mehr als nur einen simplen Link zu hinterlegen).

Neben diesen beiden bekannten Barcode-Varianten gibt es noch eine Fülle weiterer, vor denen natürlich auch die technikgebeisterten Cacher nicht halt machen um ihre Informationen möglichst menschenunlesbar anzubringen. Wirklich verschlüsselt sind diese Inhalte aber nicht, nur braucht man in aller Regel eine passende Software auf Smartphone oder PC um den Strichinformationen menschenlesbare zu entlocken. Natürlich geht dies meist auch mit Hirn und Wissen, wenn man nur weiß, wie ein bestimmter Strichcode aufgebaut ist und welchen man gerade zu entschlüsseln versucht. Die Codeliste, aus der Code 39 aufgebaut ist, kann man bei [Wikipedia - Code 39](#) bewundern. Diesen gibts auch als ttf-Schriftart.

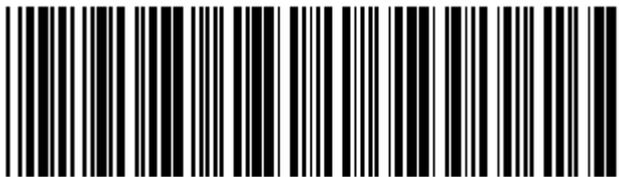
Hat man das weltweite Netz zur Verfügung, erschlägt es einen beinahe mit dem Angebot von Webseite, um zumindest die Standardcodes zu übersetzen. Wird es etwas komplizierter, der Code schlechter lesbar, nutze ich gern die Software [bcTester](#). Unterwegs hab ich die Erfahrung gemacht, daß mehrere Barcodeleser im Smartphone die bessere Wahl sind. Mehrere Smartphones die Beste. Im schlimmsten Fall kann man das Gefundene auch abfotografieren und mit einem weiteren Gerät - hoffentlich - lesen und entschlüsseln.

Entstanden sind Barcodes aus der Anforderung, maschinenlesbare und möglichst eindeutige Informationen anbringen zu können. Aus diesem Grund enthalten die bekannten EAN-Warencodes Prüfsummen, eine Fehllesung ist somit fast ausgeschlossen. Ansonsten enthält dieser Code nur Ziffern, dargestellt in unterschiedlich breiten Strichen und Lücken. Andere Codes können auch Ascii-Zeichen oder erweiterte Zeichensätze darstellen. Am meisten Inhalt bekommt man in die zweidimensionalen Codes, die Datamatrixen.

Hier eine kleine und keineswegs vollständige Übersicht bekannterer Barcodes:

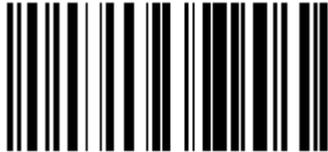


"EAN - Numerischer Code 0-9, Striche und Lücken enthalten Information, 8 oder 13 Zeichen "



1234567890

Code 39: Alphanumerischer Code, 0 - 9, 26 Buchstaben, 7 Sonderzeichen. Jedes Zeichen besteht aus 9 Elementen (5 Strichen und 4 Lücken, 3 breit, 6 schmal).



1234567890

Code 128: voller ASCII-Zeichensatz mit Hilfe von 3 Zeichensätzen, die über ein Startzeichen gewählt werden. Besteht aus 11 Zeichen, aufgeteilt in 3 Striche und 3 Lücken.



1234567890

Code 2/5 Interleaved: Numerischer Code 0-9, besteht aus breiten und schmalen Strichen und Lücken.



012345678905

Deutsche Post Identcode: Numerischer Code, 0-9, 12 Zeichen lang, Prüfziffer.



Code Postnet: Code des US Post Office, numerischer Code, 0-9 darstellbar. Prüfsumme, Start und Stoppzeichen.



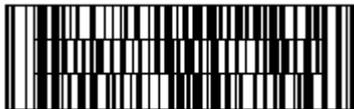
Code Royalmail: enthält Ziffern 0-9 und 26 lateinischen Buchstaben. Code "kix" sieht diesem hier sehr ähnlich.



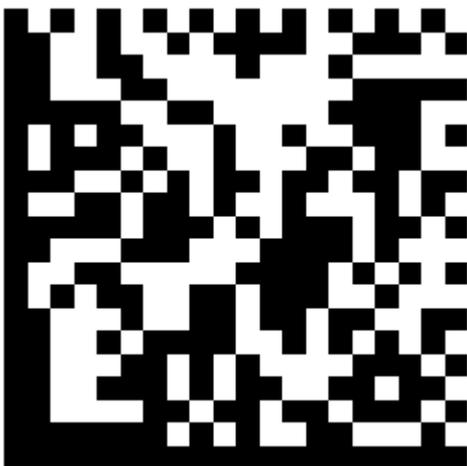
Zielcode: wird von der Deutschen Post benutzt.



Code PDF417: Stapelcode mit starker Fehlerkorrektur. Besteht aus einzelnen Elementen, den "Codewörtern", welche aus je 17 Modulen aufgeteilt in 4 Striche und 4 Lücken bestehen.



Codablock: gestapelter Code39/128. Jede Zeile hat einen Zeilenindikator (gleicher Start) und Prüfsumme.



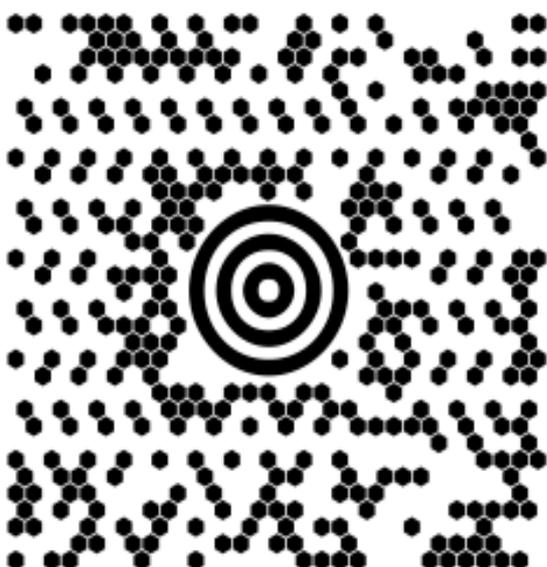
Datamatrix: variable, rechteckige Grösse zwischen 10x10 und 144x144 Zeichen. Enthält den erweiterten Ascii-code. Waagerechte und senkrechte Umrandung beschreibt eine Ecke zur Orientierung des Lesegerätes.



QR-Code: Markierung in 3 von 4 Ecken zur Orientierung des Lesegerätes. Große Fehlertoleranz. Ascii-Zeichen.

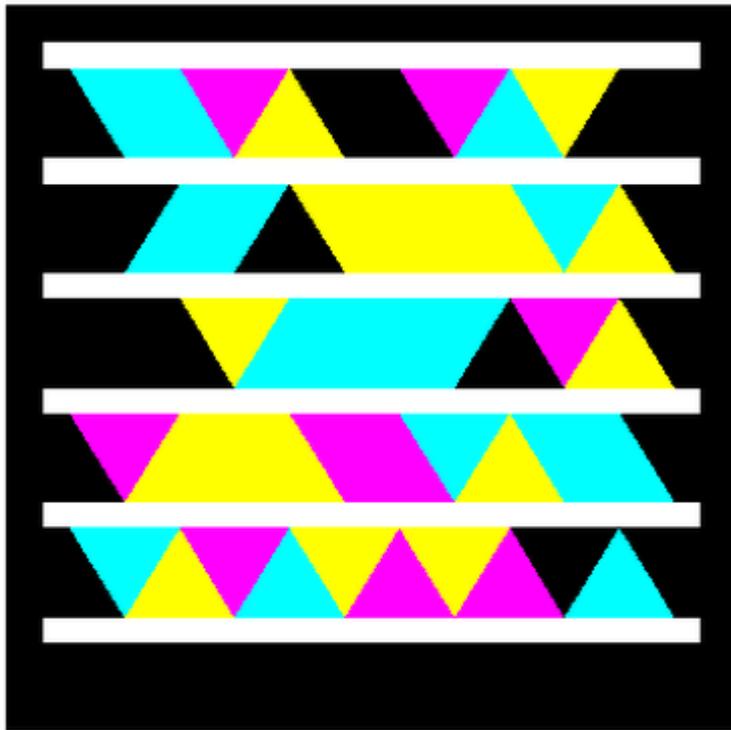


Aztec: Ziffern und Buchstaben, kreisförmig um den Mittelpunkt angeordnet.



Maxicode: sechseckige Zeichen um runden Mittelpunkt. enthält bis zu 93 alphanummerische oder

138 numerische Zeichen. Große Fehlertoleranz. Ähnlich beetag-Code.



Microsoft Tag: einziger farbiger Code, kann sehr verschiedenartig aussehen, auch in Bilder integrierbar. Closed Code, generieren und lesen nur über von Microsoft bereitgestellte Wege <http://tag.microsoft.com/home.aspx>.



Weitere Microsoft-Tag Beispiele

Weitere Codebeispiele:

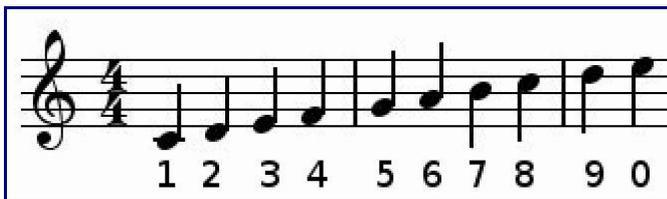
<http://www.logicalconcepts.eu/wDeutsch/autoid/barcodetypen/index.html?navid=21>

<http://www.activebarcode.de/codes/>

Wer gerne selber Barcodes unter Windows erstellen möchte, dem kann ich (Stand: Erstellung dieses Blogbeitrages) den [Zint Barcode Generator](#) empfehlen. Simpel, selbsterklärend und erstellt über 50 verschiedene Barcodes.

8.2 - Noten und Notenverschlüsselungen

Musik ist ein Betätigungsfeld, mit dem sich viele Cacher in den Pausen vom Dosensuchen erholen zu scheinen. Zumindest erkläre ich mir so die gefühlte Häufigkeit von Mysteries, die irgendwelche Noten enthalten. Für mich immer ein kleiner Horror, hab ich diese runden "Bubbel" schon zu Schulzeiten gehasst und erfolgreich mit Verachtung gestraft. Was neben der pädagogischen Unfähigkeit des Lehrpersonals sicher auch daran lag, daß ich in etwa so musikalisch wie ein Kühlschrank bin. Letztlich taugt das aber nicht als Ausrede, die damals entstandenen Wissenslücken musste ich mir nun, wo ich endlich einen sinnvollen Einsatzzweck dafür gefunden habe (Mysteries zu entschlüsseln ;)), mühsam zusammenflicken.



Die meisten Notenmysteries sind ziemlich simpel gestrickt. Irgendwie muss man den Notengrubbel ja Zahlen gegenüberstellen, also hat irgendwer mal damit angefangen, einfach nur durchzuzählen. Die erste Note ist eine 1, die nächsthöhere eine 2 und so weiter.

Praktischerweise ist das ein Noteneinsatz, den sogar ich auf Anhieb durchschauen konnte.

Dummerweise ist diese Zählweise nicht unbedingt "genormt", so daß man vielleicht noch etwas mit den Ziffern hin- und herjonglieren muss. Man könnte auch bei 0 anfangen zu zählen und musikalische Laien wie ich würden auch eher auf den Notenzeilen beginnen zu schreiben und zu zählen anstatt irgendwie im Nichts darunter ;) .



Englisch bzw. Deutsch oben, italienisch unten.

Etwas schwieriger sind die Fälle, in denen die Verschlüsselung auf die Notenbezeichnung fußt. Johann Sebastian Bach hat gerne seinen Namen, also die Noten B-A-C-H in seinen Stücken untergebracht. Andere Künstler taten dies ebenfalls, nur hatten die wenigsten einen von den Buchstaben bzw. Noten her so tauglichen Namen.

Nimmt man für derartige Wortbildungen nun noch die Halbtöne:

-is für um einen halben Ton erhöhte Noten, zum Beispiel cis, mit einem kleinen Doppelkreuz markiert

-es für um einen halben Ton erniedrigte Noten, zum Beispiel des, durch ein kleines b vor der Note gekennzeichnet

kann man schon beinahe sinnhafte Worte bilden. Oder möglicherweise zumindest hübsche Vorlagen für weitere Buchstabenwertberechnungen liefern. Richtig fiese Naturen mischen noch die italienischen Notenbezeichnungen (do-re-mi-fa-sol-la-si) dazwischen. Das braucht dann eine Menge Leseverständnis oder Notenliebe zum Entschlüsseln.

Neben den Notenbezeichnungen und der Höhe auf den Notenlinien unterscheiden sich die Noten auch noch durch ihr Aussehen. Es gibt die hier gezeigten schwarzen Noten mit Stiel dran, das sind Viertelnoten. Haben sie ein Stiel, sind aber nicht schwarz ausgefüllt, handelt es sich um eine halbe Note. Ist die Note komplett schwarz, hat aber noch ein kleines Fähnchen (oder sind mehrere mit einem Strich oben verbunden), gehören diese zu der Gattung der Achtelnoten und fehlt der Stiel



völlig, ist es eine ganze Note. Manchmal sind Noten unten mit einem Bogen verbunden, dann spielt man sie in der Musik zusammenhängend. Für die Mysteryberechnung könnte es bedeuten, dass man diese hier vielleicht addieren muss. Aus den verschiedenen Notenarten kann man sich nun allerlei lustiges ausdenken, um den Mysterylöser zu quälen. Zum

Beispiel könnten nur die ganzen Noten zur Koordinatenberechnung benutzt werden, und dann ihren Zahlenstellenwert wie oben beschrieben. Man könnte auch Mathematik ins Spiel bringen, immerhin bieten die Brüche (Halbnote, Achtelnote) eine gefällige Vorlage dafür (die Notengattungen einzeln zählen und durch ihren "Bruch" teilen?). Sicher hilft es hier mal wieder, wenn man das Notenblatt versucht möglichst logisch zu betrachten, die Anzahl der Noten der üblichen Anzahl der Koordinatenziffern gegenüberstellt. 52° 12.345 und 009° 12.345 sind sieben und acht Ziffern, also 15. Hab ich praktischerweise 15 Noten (oder einzelne Notenarten), weiß ich schon, wo die Lösung steckt und muss nur noch über das "wie" nachdenken.

Wem dies noch nicht zur Entschlüsselung ausreicht, kann das Alphabet durch Noten ersetzen. Francis Poulenc und Giovanni della Porta taten dies im 16. Jahrhundert ([Versteckte Botschaften](#) von Klaus Schmeh, Seite 27 und 28). Als "Französische Notenverschlüsselung" bekannt ist eine ähnliche Variante, bei der die obere Zeile die üblichen Notenbezeichnungen sind (das B ist im Deutschen die Note H) und diese verschiedenen Entsprechungen haben können. Geschickt komponiert, zum Beispiel mit verschiedenen Tonlagen garniert, kann man so lustige Notengeschichten in die Listings bringen.

A	B	C	D	E	F	G
H	I	J	K	L	M	N
O	P	Q	R	S	T	U
V	W	X	Y	Z		

Ebenfalls möglich wäre eine simple Notenhäufigkeit, die dann dem Alphabet gegenüber gestellt wird. Die häufigste Note könnte dann dem A entsprechen, die zweit häufigste dem B. Alternativ könnte man die Notenhäufigkeit der üblichen [Buchstabenhäufigkeit](#) in der deutschen Sprache gegenüberstellen.

Dies hat sogar mal jemand im großen Stil gemacht: [Christiane Licht](#) hat 40.000 Musiktitel dementsprechend untersucht und ich bin mir sicher, irgendwer hat ihr Forschungsergebnis doch bestimmt schon in einem Mystery benutzt, oder?

Der russische Komponist Alexander Nikolajewitsch Skrjabin (*1872 - 1915) verknüpfte bestimmte Tonarten bzw. Töne mit speziellen Farben und schuf damit den [Skrjabin-Code](#).

Da die Fingersätze bei den verschiedenen Instrumenten meist genormt sind (also wo welcher Finger bei welcher Tonart zu liegen hat), könnte auch dies ein trefflicher Lösungsansatz sein, wenn im Listing ein bestimmtes Instrument genannt ist.

Und sollte jemand von euch einen weiteren dieser lustigen Notenverschlüsselungscaches basteln wollen, mit auf [Scorio.com](#) kann man (und sogar ich! ;)) hervorragend einfach Noten malen lassen.

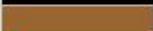
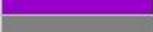
Nachtrag: Noch eine Variante, mit Noten und dem dazugehörigen Liedtext Koordinaten verschlüsseln: man nimmt die Noten und legt sie über den Text - und nimmt die Buchstaben, die von einer Note "getroffen" wurden.

8.3 – Farben

Viele Mysteries verschlüsseln die Koordinaten mit Hilfe von Farben. Hat ja auch gleich den Vorteil, dass es das Listing hübsch bunt macht. ;)

Widerstandsfarbcode

Mit am Häufigsten begegnen einem dabei der [Widerstandsfarbcode](#). Meist ohne, manchmal aber auch mit Berechnung der Stromstärke, die übrig bleibt. Aber auch hierfür gibt es genügend [Rechner](#) im Internet.

schwarz		0
braun		1
rot		2
orange		3
gelb		4
grün		5
blau		6
violett		7
grau		8
weiß		9

In der einfachen Variante entspricht eine Farbe einfach einer Ziffer, die dann, je nach Rätsel, weiterverwendet werden kann.

RAL-Farben

Auch sehr häufig werden die normierten [RAL-Farben](#) benutzt. Entweder als benutzte Farbe oder als benutzter Farbwert (Zitronengelb ist zum Beispiel RAL 1012 und Erdbeerrot RAL 3018). Die Farbwerte gibt es auch auf Englisch.

Hex-Farbcodes

Wenn diese beiden nicht zur Rätsellösung nutzen, dann vielleicht der [HEX-Farbcode](#). Hier ein paar Beispielfarben und ihre Hexadezimalwerte.

	Farbe	Hex-Wert
schwarz	black	#000000
grau	gray	#888888
weiß	white	#FFFFFF
rot	red	#FF0000
dunkelrot	maroon	#800000
grün	lime	#00FF00
dunkelgrün	green	#008000
blau	blue	#0000FF
dunkelblau	navy	#000080
gelb	yellow	#FFFF00

Hex-Wert-Spielereien sind unter Mysteryerstellern sowieso sehr gefragt und praktischerweise kann man Farben auf Webseiten im Hex-Code darstellen. Somit hilft mal wieder ein Blick in den [HTML-Quellcode](#) . Ist dort Text in zwei verschiedenen Farben geschrieben, könnten das vielleicht schon die Nord- und Ostkoordinaten sein.

Die beiden Worte "[Koordinaten](#) " und "[Verschlüsselung](#) " haben im Quelltext diese Farbdefinitionen: font color="#500AEE" und font color="#E6E0E"

Die beiden verwendeten HEX-Zahlen in Dezimalzahlen umgerechnet ergibt 5245678 und 945678 . Also wunderschöne Koordinaten, wenn man nur ein paar Pünktchen und Gradzahlen hinzufügt. N 52° 45.678 und E 9° 45.678 (diese Koordinate ist natürlich nur ein Beispiel. Falls dort oder an anderen, hier im Blo(g)ck genannten Punkten wirklich Dosen liegen, bitte ich um einen freundlichen Hinweis, damit ich (sie loggen???? *hihi* nein! natürlich) diese Beispiele hier abändern kann.

Neben den Farbwerten können die [16 Grundfarben in HTML](#) auch über ihren englischen Namen (z.B. blue oder yellow) angesprochen werden.

RGB-Farben

Die HEX-Darstellung der Farben ist nur eine andere Darstellung der RGB-Farben. RGB bedeutet Red-Green-Blue und beschreibt einen additiven Farbraum; also die Darstellung von Farben durch das Mischen der drei Grundfarben (rot, grün und blau). Dieses in Zahlen von 0 bis 255 ausgedrückt, ergibt die Darstellung im [RGB-Farbraum](#). Gängige Bildbearbeitungsprogramme wie Gimp oder Photoshop zeigen für jeden Bildpixel die RGB-Farben an. Gibt es im Listing zum Beispiel ein Bild mit nur zwei Farben, könnten auch deren RGB-Farbwerte ein Lösungsansatz sein.

	Farbe	Hex-Wert	RGB
schwarz	black	#000000	0,0,0
grau	gray	#888888	136,136,136
weiß	white	#FFFFFF	255,255,255
rot	red	#FF0000	255,0,0
dunkelrot	maroon	#800000	128,0,0
grün	lime	#00FF00	0,255,0
dunkelgrün	green	#008000	0,128,0
blau	blue	#0000FF	0,0,255
dunkelblau	navy	#000080	0,0,128
gelb	yellow	#FFFF00	255,255,0

Und was es sonst noch alles geben könnte

Das waren jetzt die häufigeren Farbensprechungen. Je nach Owner, seinen Präferenzen und dem Thema des

Listings gibt es natürlich noch ungefähr unendlich viele weitere Möglichkeiten. Zum Beispiel der [HKS-Farbfächer](#), die Farben von TÜV-Plaketten, Leuchtdioden, die Wellenlängen von Licht, [Hexahuhe](#), die Farbzahlen der Eddingstifte, Farbfilterfolien und noch wesentlich mehr, welches einzeln zu erwähnen, hier nicht mehr sinnvoll erscheint.

Nur eins noch: sind es vielleicht bis zu 10 verschiedenen Farben im Rätsel, die sich abwechselnd wiederholen? Dann steht möglicherweise eine Farbe für eine Ziffer? Sind es mehr als 20? Dann steht vielleicht eine Farbe für einen Buchstaben? Da hilft möglicherweise (neben Fleiß) das Kapitel über [Buchstabhäufigkeiten und manuellem Entschlüsseln](#).

8.4 – Steganographie

Steganographie bedeutet, eine Information in oder auf etwas so zu verstecken, dass ein nicht Eingeweihter von der Existenz der Information nichts mitbekommt. Als Trägermedium kann so ziemlich alles in Frage kommen, sogar Menschen: in der Antike wurde Sklaven der Kopf geschoren, auf diesem etwas tätowiert und wenn die Haare nachgewachsen waren, wurden sie als

lebende „Datenträger“ losgeschickt. Aber auch so klassische Agentenmethoden wie Geheimtinte oder der doppelte Boden in Paketen oder Koffern, hohle Absätze in Schuhen und die Nutzung von Mikropunkten ist Steganographie. Für Geocacher eher von Bedeutung ist die linguistische Steganographie, also das Verstecken von Text in einem Text (zum Beispiel über Schlüsselwörter mit besonderer Bedeutungen) oder das von Informationen in einem Bild (optisch: zum Beispiel über [Grashalme als Morse](#)) oder computergestützt mit entsprechender Software.

Der Nachteil dieser eigentlich hübschen Methoden: Sender und Empfänger müssen sich über die Art des Verstecke(n)s austauschen. Und das ist auch der große Nachteil, wenn computergestützte Steganographie in (Geocaching-)-Rätseln verwendet wird: der Rätselnde sollte wenigstens grob eine Ahnung haben, ob und wenn mit welchem Tool hier etwas versteckt worden ist, weil das Durchprobieren aller üblichen oder unüblichen Verdächtigen, teils mit (diversen) Passwortmöglichkeiten, ergeben eine fast unendliche und vor allem recht langweilige Suche nach der Koordinate. Schließlich lässt sich etwas was mit einer Steganographie-Software verborgen worden ist, immer auch nur exakt mit dieser wieder zurückholen.

Solltet ihr mal über ein Listing stolpern, bei dem sich einfach kein Weg zu einem Koordinatenversteck ergeben möchte, welches aber eine Datei, meist ein Bild, aber möglicherweise auch ein mp3-File, ein Video oder eine unbekannte Dateart enthält, die der Owner auf einem eigenen Webspace abgelegt hat, dann könnte eine genauere Untersuchung dieser Datei weiterhelfen. Ist diese Datei vom Datenvolumen her größer, als sie typischerweise sein sollte? Dann könnte eine weitere in ihr versteckt sein. Ist es ein Bild und dieses ist ungewöhnlich groß (ich meine hier die Pixelanzahl), könnte optisch etwas verborgen sein, was man vielleicht nur in voller Bildgröße erkennen kann. Ich fand mal in einer hübschen schwarz-weiß-Zeichnung bei voller Auflösung seltsam wirkende Punkte an einer Türzarge. Eine Websuche ergab, daß das Originalbild diese nicht aufwies. Die Lösung war dann simples Abzählen um die Ost- und Nordminuten zu erhalten.

Handelt es sich um ein JPG-Bild mit einer möglicherweise vorhandenen, computergestützten Steganographie, untersuche ich es immer erstmal mit der kleinen, uralten Software „[stegdetect](#)“. Das ist allerdings ein Kommandozeilentool ("stegdetect -t p dateiname.jpg"). Selbst wenn [stegdetect](#) nicht erkennen kann welches Tool hier zum Verstecken benutzt worden ist, so gibt es häufig wenigstens einen Hinweis darauf, ob das JGP überhaupt manipuliert worden ist. Ähnlich deutliche Hinweise hab ich aber auch schon mit anderer Steganographie-Software erhalten. So meldete mal steghide einmal, dass an einer BMP-Datei etwas nicht stimmen würde, die allerdings mit einer ganz anderen Software, nämlich Grafik-Key, verschlüsselt worden ist. Immerhin war ich mir von da an wenigstens sicher, auf der richtigen, nämlich der Steganographie-Spur, zu sein.

An dieser Stelle bleibt mir nicht mehr viel zu schreiben, als das simple (und keineswegs vollständige) Aufzählen von mehr oder minder üblichen Steganographie-Programmen und den Datentypen, die sie verschlüsseln können. Sollte die Software die Möglichkeit von Passwörtern bieten, ist der GC-Code des Listings, der Name des Owners, des Caches oder etwas, was im Text deutlich heraus sticht, ein guter Kandidat. Alternativ: der Dateiname.

Viel Erfolg beim Suchen nach versteckten Informationen zum Beispiel mit:

[stegano.net](#) (JPG, PNG)

[Carmouflage](#) (eher nicht mehr aktuell, läuft in der kostenlosen Windows-Version nur bis Windows XP)

[steghide](#) (Bild- und Audio-Dateien)

[Grafik-Key](#) (BMP)

[steganog](#) (BMP)

[Openstego](#)

[OpenPuff](#) (Bilder, Audio, Video, Flash)

[JPBS](#) (Audio, Video, Bilder, Text)

[Outguess](#) (JPG)

[data-stash](#)

[silent eye](#)

[GpgSX 0.67b](#)

[Stealth Files 4.0](#) (diverse Dateitypen EXE-, DLL-, OCX-, COM-, JPG-, GIF-, ART-, MP3-, AVI-, WAV-, DOC-, BMP- und WMF-Dateien)

[PGE - Pretty Good Envelope](#)

S-Tools 4.0 (scheint es frei im Web nicht mehr zu geben?)

F5 (scheint es frei im Web nicht mehr zu geben?)

[mp3stego](#)

[Snow](#) - versteckt Daten in ASCII-Text, genauer in dessen Leerzeichen

[spammimic](#) verschlüsselt Text in etwas, was wie Spam aussieht

Sogar der momentane Freewareverschlüsselungs-Marktführer [TrueCrypt](#) bietet eine Form der Steganographie. Man kann nicht nur einfach Dateien oder Laufwerke hiermit verschlüsseln, man kann in ihnen einen „hidden-Container“ anlegen, von dessen Existenz man nur erfährt, wenn man das richtige Passwort eingibt. Es gibt eins für den „normalen“ Container und ein weiteres für den steganographisch versteckten „hidden-Container“. Verschlüsselung und Verstecken in einem. Weniger etwas für Geocaching-Mysteries, aber definitiv etwas für Leute, die doch etwas zu verbergen haben.

(Nachtrag vom 23.12.2014: truecrypt hat inzwischen seine Dienste eingestellt. Man munkelt, daß die US-amerikanische Regierung ein Hintertürchen in diese Software erzwungen hat. Truecrypt ist also nicht mehr sicher .)

Und zum Schluss noch ein Besserwisserhinweis für zukünftige Rätsel-Owner: eigentlich entsprechen computergestützte Steganographie-Rätsel nicht den GC-Richtlinie, da man für die Entschlüsselung eine Software installieren muss.

8.5 Esoterische Programmiersprachen

Nein, wir verlassen jetzt nicht den Bereich der Logik und begeben uns in unseren inneren, spirituellen Erkenntnisraum. Esoterische Programmiersprachen wären als „exotisch“ möglicherweise etwas besser betitelt und beschreiben Programmiersprachen, die nicht für den praktischen Einsatz, aber für Demonstrationszwecke, als akademische Scherze oder aus purer Langeweile (hochbegabter Menschen) entstanden sind.

Für Geocaching-Mysteries ist es erstmal wichtig zu wissen, daß es derartiges gibt, damit man im Fall der Fälle nach diesen und Interpretern (Übersetzern) oder doch wenigstens der Syntax suchen und das Listing so entschlüsseln kann. Hier also ein paar der üblicheren Beispiele und hoffentlich weiterführende Links.

Am häufigsten begegnete mir bislang die Programmiersprache mit dem wundervollen Namen:

„[Brainfuck](#)“.

Brainfuck besteht aus nur acht Befehlen, die jeweils durch ein einziges Zeichen dargestellt werden:

<>+-.,[]. Ein Programm könnte dann zum Beispiel so aussehen:

```
+++++ +++[- >++++ +++++< ]>++++ +++++ +++++ +.<+++ +++++[- >++++ +<]>+ +.+++  
++.<+ ++[-> ---<] >---- .<+++ +[->+ +++<] >+>.< +++++ +++++[->---- ----  
-<]>- .<+++ +++++ [->+ +++++<]>+> .<+++ +[->+ +++<] >.>+ +.++ +.+-  
---.---.< +++[- >++++< ]>++++ +.<+++ +[-> ----< ]>.<+ ++[-> +++<] >+>+>  
+.<+++ +++++ +[-> ----< ]> ---- .<+ +++++ +[->+ +++++ +<]>+ ++++++  
+++++ +.+.< +++++ +[-> ----< ]> ---- ---- .< +++++ +[-> ++++++  
++<]> +++++ +++++. +++++ +.+.<
```

Einen Online-Übersetzer findet ihr zum Beispiel [hier](#). Dieser funktioniert (was für ein hübscher Übergang ;) auch mit der auf Brainfuck basierenden Sprache „Ook“ (Terry-Pratchett-Leser fühlen sich hier sofort wie Zuhause). Ook ist die erste Programmiersprache, die es sich zum Ziel gesetzt hat, von einem durchschnittlichen Orang-Utan verstanden zu werden. Sie besteht nur aus drei Elementen: Ook. Ook? Und Ook!

Beispiel:

```
Ook.  
Ook. Ook! Ook? Ook! Ook! Ook. Ook? Ook. Ook. Ook. Ook. Ook. Ook. Ook. Ook.  
Ook. Ook. Ook. Ook. Ook. Ook. Ook. Ook. Ook? Ook. Ook? Ook! Ook. Ook? Ook.  
Ook. Ook. Ook. Ook! Ook. Ook? Ook. Ook. Ook. Ook. Ook. Ook. Ook. Ook. Ook.  
Ook. Ook. Ook! Ook? Ook! Ook! Ook. Ook? Ook. Ook. Ook. Ook. Ook. Ook. Ook.  
Ook. Ook. Ook. Ook? Ook. Ook? Ook! Ook. Ook? Ook. Ook. Ook. Ook. Ook. Ook.  
Ook. Ook. Ook. Ook. Ook. Ook. Ook. Ook! Ook. Ook? Ook. Ook. Ook. Ook. Ook.  
Ook. Ook! Ook? Ook! Ook! Ook. Ook? Ook. Ook. Ook. Ook. Ook. Ook. Ook? Ook.  
Ook? Ook! Ook. Ook? Ook. Ook. Ook. Ook. Ook. Ook. Ook. Ook. Ook. Ook! Ook. Ook?  
Ook. Ook. Ook. Ook. Ook. Ook. Ook. Ook! Ook? Ook! Ook! Ook. Ook? Ook! Ook!  
Ook! Ook! Ook! Ook! Ook? Ook. Ook? Ook! Ook. Ook? Ook! Ook! Ook! Ook! Ook!  
Ook! Ook! Ook! Ook! Ook. Ook? Ook. Ook. Ook. Ook. Ook. Ook. Ook. Ook! Ook?  
Ook! Ook! Ook. Ook? Ook. Ook. Ook. Ook. Ook. Ook. Ook? Ook. Ook? Ook! Ook.  
Ook? Ook. Ook. Ook. Ook. Ook. Ook. Ook. Ook. Ook! Ook. Ook! Ook! Ook! Ook!  
Ook! Ook! Ook! Ook! Ook! Ook! Ook! Ook! Ook! Ook! Ook! Ook! Ook! Ook!  
Ook. Ook? Ook.
```



Wer mal in einem Mystery auf verwirrend bunte Pixelbildchen stößt, könnte sich vielleicht einmal mit [Piet](#) beschäftigen. Diese Sprache verbirgt ihren Code in bunten Gif-Bildern, die den Bildern des Malers [Piet Mondrian](#) ähnlich sehen.

[Hier](#) gibt's einen Online-Interpreter:

Und hier jede Menge [Beispielprogramme](#), also Bilder:

Wenn ihr mal über einen Text stolpert, der ungewöhnlich seltsame Leerzeichen aufweist (strg-a könnte sie euch zeigen), schaut euch mal „[Whitespace](#)“ an. Hier verbergen sich die relevanten Inhalte in allen vorhandenen Leerzeichen, Tabs und Zeilenumbrüchen! Beispiele findet ihr [hier](#).

Ziemlich witzig finde ich den Ansatz der Programmiersprache „[Chef](#)“ : Programmcode innerhalb eines Kochrezeptes zu übermitteln. Hierbei sind die Zutaten die Variablen, flüssige Zutaten stellen Unicode-Zeichen da, alle sonstigen Zutaten sind Zahlen. Außerdem gibt es Anweisungen wie zum Beispiel Liquify zum Umwandeln und Rührschüsseln oder Backformen zum Speichern von Stacks.

Hallo World in „Chef“ könnte dann so aussehen:

Der Hallo Welt Nachtisch à la Nina:

Ingredients.

72 g haribo colorado

97 gummibären

108 g kinderschokolade

111 cups oil

32 marzipanbrot

87 ml water

101 eggs

116 g bonbons

33 kekse

Method.

Put kekse into the mixing bowl. Put bonbons into the mixing bowl.

Put kinderschokolade into the mixing bowl. Put eggs into the mixing bowl. Put water into

the mixing bowl. Put marzipanbrot into the mixing bowl. Put oil into the mixing

bowl. Put kinderschokolade into the mixing bowl. Put kinderschokolade into the mixing bowl. Put

gummibären into the mixing bowl. Put haribo colorado into the mixing bowl.

Liquify contents of the mixing bowl. Pour contents of the mixing bowl into the baking dish.

Serves 1.

Auch irgendwie besonders ist die [SPL \(Shakespeare Programming Language\)](#) , deren Quelltext aussieht wie ein Theaterstück. Extra für böse (!) Totenbeschwörer gibt es [Zombie](#) , die – irgendwie logisch - Tote beschwört. Und für die krass-coolen Netzjargonliebhaber gäbe es noch den

LOLCODE.

Und noch viel mehr Beispiele für esoterische Programmiersprachen finden sich im großen, weiten Netz:

http://esolangs.org/wiki/Language_list

<http://www.99-bottles-of-beer.net/>

<http://www.dangermouse.net/esoteric/>

Viel Spaß beim Stöbern und "entmystifizieren". Für manche dieser IT-Spielereien wird es keinen Übersetzer geben, da muss man im Zweifel die Syntax verstehen lernen und dem Programm logisch hinter sein Geheimnis kommen.

9 - Tools und Links

9.1 - Links und Codelisten

(zuletzt aktualisiert: 05. November 2014)

Hier findet sich ein Abriss meiner Lesezeichen, die keinen Anspruch auf Vollständigkeit erhebt. Über Hinweise auf weitere Links wäre ich sehr dankbar!

Codelisten

[Bergziege OWL](#)

[Codelisten Geocaching-Franken.de](#)

[MyGeotools/](#)

[Code-Knacker](#) (Codes, Symbolen und Kurzzeichen)

Chiffrierungen

[cryptool-online](#) Chiffren, Kodierungen, Kryptoanalyse - die Besten der Besten, Sir!

[Kryptographiespielplatz](#)

[Secret Code Breaker](#)

Decodierungssammlungen, Ciphertools, diverse Umwandler

[netteleuthe.de](#) verschiedene Geocachingtypische Umwandler

[Happy Security](#) diverse eher klassische Verschlüsselungen

[rumkin.com](#) Cipher Tools

[cacheblogger.de](#) decrypter

[cachebrett.de](#) converter

<http://www.yellowpipe.com/>

[easycalculation](#) sehr viele Umwandler und Umrechner

[thematrixer.net](#) Binäre Umwandler

[paulschou.net](#) Binäre Umwandler

[ascii to hex](#) und mehr

[dasumo.com](#) Binäre Umwandler

[patshaping.de](#) Base64

[mobilefish.com](#) Passworthashes erzeugen

[manuelles chiffrieren](#)

[Häufigkeitsanalyse](#)

[Baudot](#) automatisch in allen Spielarten

[cryptii](#) konvertiert in diverses, von Baudot bis Navajo

[Automatic Vigenere Decoder](#) <-! und der kann das wirklich!

Buchstaben-in-Ziffern:

[rentfort.de](#) buchstaben-in-zahlen-umwandeln

[oliver-rahe.de](#) buchstaben in zahlen umwandeln

[nummerologie](#) quersumme-berrechnen.php inkl. Römisch

Rot

rot13.com

[ROT-Irgendwas](#)

[DecodeRot](#) (alle Varianten auf einen Blick!)

harald.ist.org Codeknacker für Buchstaben-Verschiebe-Codes

[Rot13 / Rot 47](#)

Bilderanalysen online

[regex](#) Online Exif-Viewer mit Thumbnail-Anzeige

[metapicz](#) Exif und mehr

[img-ops](#) weiterführende Links zur Bilderanalyse und Veränderung

<http://fotoforensics.com>

Zahlensysteme umrechnen

[mahoplus](#) Umrechner für Zahlensysteme

www.welt-zeit-uhr.de zahlensysteme/

[convertworld.com /](http://convertworld.com/)

[arndt-bruenner](#) Zahlensysteme.htm

Anagramme

<http://anagramme.spieleck.de/app/neu?0>

<http://wordsmith.org/anagram/index.html>

<http://www.buchstabensalat-knacken.de/>

<http://www.sibiller.de/anagramme/>

Farben

[widerstand-farbcode-rechner-](#)

[Farbcodes von Widerständen](#)

www.bader-frankfurt.de - Widerstandscode

geocaching-sued.de farbcodes

[Farbenrechner](#)

[RAL Farbtabelle](#)

Fibanocci

[Fibonacci - Fingerzahlen](#)

[Fibonacci Zahlenreihe](#)

Bildersuchmaschinen

www.tineye.com DIE Bildersuche

<http://www.revimg.net/>

[googles Bildersuche](#)

Koordinaten, Umrechnung und mehr

[cache-test-dummies](#) koordinatenumrechnung

http://www.gpsvisualizer.com/map_input konvertieren, Kreisabstände

[Wegpunktprojektion](#)

<http://www.mygeoposition.com/>

[Deine Berge - umfangreicher, sehr guter KO-Umrechner](#)

[zwanziger](#) (Wegpunktprojektion, KO konvertieren)

Kartesische und Polarkoordinatenkonvertierer

Maps und Umrechner

Flopps wirklich tolle Karte

gps0

twcc (Umwandlung sehr vieler Koordinatensysteme)

maptools (konvertieren, Radius)

Barcode Online Decoder

<http://www.onlinebarcodereader.com/>

<http://zxing.org/w/decode.aspx>

MHD5/SHA1 Hashes

web-max.ca

<http://md5cracker.org/>

<http://www.md5.cz/>

Schriften

omniglot.com

Science Fiction Schriften

Alte Schriften

Leet-Key

[leet-key/ Firefox-Plugin](#)

[robertecker leet-converter](#)

Enigma

[sternenhimmelstuermer](#)

enigmaco.de

[cryptomuseum](#)

[rijmenants Enigmasim](#) Der m.E. beste Enigmasimulator

Nonogrammlöser

teall.info/nonogram/

griddler.co.uk Solve

[comp auto nonogram](#)

Hieroglyphen

[Gardiner-Liste](#)

philognosie.net

esotherische Programmiersprachen

http://esolangs.org/wiki/Language_list

<http://www.dangermouse.net/esoteric/>

splitbrain.org (ook/Brainfuck)

Töne

[DTMF Töne](#)

[Frequenzbelegung der Telefontastatur](#)

Steganographie-Tools

[stegano.net](#) (JPG, PNG)

[Carmouflage](#) (eher nicht mehr aktuell, läuft in der kostenlosen Windows-Version nur bis Windows XP)

[steghide](#) (Bild- und Audio-Dateien)

[Grafik-Key](#) (BMP)

[steganog](#) (BMP)

[Openstego](#)

[OpenPuff](#) (Bilder, Audio, Video, Flash)

[Outguess](#) (JPG)

[data-stash](#)

[silent eye](#)

[GpgSX 0.67b](#)

[Stealth Files 4.0](#) (diverse Dateitypen EXE-, DLL-, OCX-, COM-, JPG-, GIF-, ART-, MP3-, AVI-, WAV-, DOC-, BMP- und WMF-Dateien)

[PGE - Pretty Good Envelope](#)

[mp3stego](#)

[Snow](#) - versteckt Daten in ASCII-Text, genauer in dessen Leerzeichen

[spammimic](#) verschlüsselt Text in etwas, was wie Spam aussieht

[MP3Stegz](#)

Orte und Länder

[Autobahnatlas](#)

[Bahnhofsnummern](#)

[Flughafencodes](#)

[Länderkennzeichen nach ISO 3166-1](#)

Sonstiges

[Gauß-Weber-Telegrafen](#)

[Ogham](#)

[Astrologische Symbole](#)

[Waldläuferzeichen](#)

[Blowfish](#)

[7-Segment-Anzeige](#)

[Farbumrechnungen](#)

[ICD-Code \(Krankheiten\)](#)

[Verkehrszeichen](#)

[Genetischer Code](#)

[Diverse Maßeinheiten umrechnen](#)

[HTTP-Fehlermeldungen](#)

[Die Zahl PI inklusive Suchfunktion](#)

[Primzahlen](#)

[Gleichungslöser](#)

<http://codeconverter.onlinetoolkit.org/>
[rexswain - Analyse des http-Datenstrom](#)
[Emulator für diverse Programmiersprachen](#)
[ICD-Code, Krankheiten](#)

9.2 - Geocaching-Tools für Zuhause und unterwegs

Diese Auswahl von Tools ist zwar sehr persönlich, enthält also nur genau das, womit ich am Liebsten arbeite und soll auch gar nicht bedeuten, daß es für die jeweiligen Einsatzgebiete nicht viel bessere Lösungen gibt (im Zweifel sind das ja immer die, mit denen man selber am Besten umgehen kann), aber vielleicht hilft es dem einen oder anderen ja doch bei der Frage, was zu installieren sinnvoll sein könnte. Ich bin zwar übrigens mit Kommandozeilen aufgewachsen (LOAD "\$",8,1 *fg*), aber seit viel zu langer Zeit windowsgeschädigt. Daher jetzt hier meine **Standard-Windows-Tools**, die bei etwa 95% der Mysteries zum Einsatz kommen:

- [notepad++](#) Ein Notepad, also ein Texteditor mit diversen Einsatzmöglichkeiten (Freeware)
- [hxd](#) Ein Hexeditor (Freeware)
- [IrfanView](#) Ein Bildbetrachter mit Exif-Ansichten, Hex und Bildbearbeitungsmöglichkeiten (Freeware)
- [gimp](#) Eine sehr mächtige Bildbearbeitungssoftware (Freeware)
- [winrar](#) Ein universal Packprogramm (kostet was, lässt sich aber unbegrenzt als Testversion einsetzen)
- [google earth](#) Gucken, peilen, Streetview und vieles mehr. Bei mir immer offen. (Freeware)
- [audacity](#) Eine Audiotbearbeitungssoftware. Sehr nützlich zum Anzeigen und editieren von Sounddateien. (Freeware)
- [stegdetect](#) Findet häufig Steganographie in JPG-Dateien (Freeware)
- [bctester](#) Erkennt Barcodes in Bildern (Freeware)
- [mopsos](#) Ein Universaltool für diverse GC-Verschlüsselungen und Koordinatenberechnungen (Freeware, braucht Registrierung)
- [Der Enigma Simulator von Dirk Rijmenants](#)

Unterwegs bin ich zur Zeit mit einem Android-Handy, daher sind das hier Tools, die man auf einer solchen Plattform zum Laufen bekommt:

- Quickmark und Neoreader (zum Einlesen von Barcodes, QRcodes etc.)
- Microsofttag (das Tool für die bunten Microsoft-Tags)
- einen beliebigen Sudoku-Solver (gibts ja doch immer mal unterwegs und nicht immer ist Zeit und Ruhe, sie per Hand zu lösen)
- [gcc \(GeoCache Calculator 1.0.5\)](#) Ein Universaltool mit allen möglichen und unmöglichen Verschlüsselungsvarianten und mehr.
- eine Taschenlampenapp (braucht man durchaus häufiger mal, wenn auch nich unbedingt zum Lösen von Mysteries)

Und last but not least: Die Codetabellen von [MyGeoTools](#) als jpgs auf Handy und meinem GPS-Gerät.

In meinem **Browser** (Firefox) nutze ich

- das Firefox-Addon LeetKey zum Konvertieren in/aus Rot13/L337, BASE64, Hex, Bin, Morse etc.

und habe eigentlich ständig die Seiten

- [Rentfort](#) (zum Umwandeln von Buchstaben in Zahlen)
- [Jeffreys Exif viewer](#) und
- [PaulSchous Binary Translator](#) offen.

Anhang: Verschlüsselungstabellen

Bacon-Chiffre

Buchstabe	Code	Buchstabe	Code	Buchstabe	Code
A	aaaaa	I, J	abaaa	R	baaaa
B	aaaab	K	abaab	S	baaab
C	aaaba	L	ababa	T	baaba
D	aaabb	M	ababb	U, V	baabb
E	aabaa	N	abbaa	W	babaa
F	aabab	O	abbab	X	babab
G	aabba	P	abbba	Y	babba
H	aabbb	Q	abbbb	Z	babbb

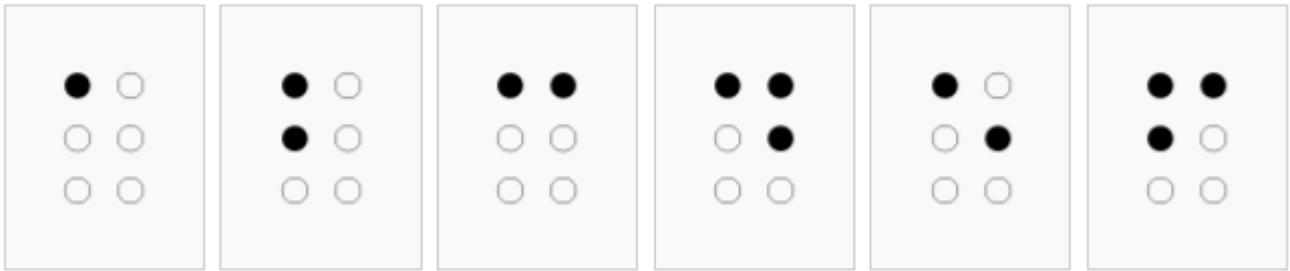
BCD

Bit-Nr.	4	3	2	1
Wertigkeit	2^3 8	2^2 4	2^1 2	2^0 1
Dezimalziffern				
0				
1				■
2			■	
3			■	■
4		■		
5		■	■	
6		■	■	■
7		■	■	■
8	■			
9	■			■

BCD-Zählcode

Ziffer	codiert
1	0 0 0 0 0 0 0 0 0 1
2	0 0 0 0 0 0 0 0 1 1
3	0 0 0 0 0 0 0 1 1 1
4	0 0 0 0 0 0 1 1 1 1
5	0 0 0 0 0 1 1 1 1 1
6	0 0 0 0 1 1 1 1 1 1
7	0 0 0 1 1 1 1 1 1 1
8	0 0 1 1 1 1 1 1 1 1
9	0 1 1 1 1 1 1 1 1 1
0 (=10)	1 1 1 1 1 1 1 1 1 1

Braille



A oder 1

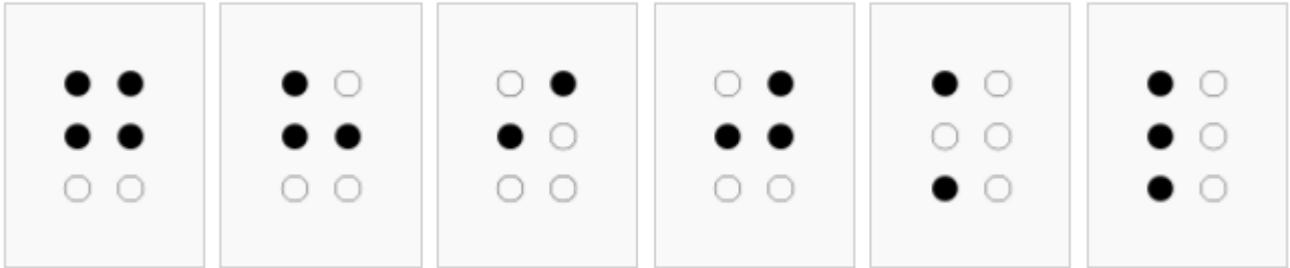
B oder 2

C oder 3

D oder 4

E oder 5

F oder 6



G oder 7

H oder 8

I oder 9

J oder 0

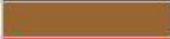
K

L

Morse

Ziffer	Morse
0	-----
1	·-----
2	··-----
3	···-----
4	····----
5	·····
6	-·····
7	-----·
8	-----·
9	-----·

Widerstandsfarbcodes

schwarz		0
braun		1
rot		2
orange		3
gelb		4
grün		5
blau		6
violett		7
grau		8
weiß		9

Noten als Ziffern (eine mögliche Variante)



Zielcode

Wert	Codierung 01247	Codierung 8421
1		
2		
3		
4		
5		
6		
7 ^[2]		
8		
9		
0 ^[3]		

Zahlensystem der Maya

0	1	2	3	4
5	6	7	8	9
10	11	12	13	14
15	16	17	18	19

Zahlensysteme, Dual, Dezimal, Hexadezimal

Dual	Dezimal	Hexadezimal
1	1	1
10	2	2
11	3	3
100	4	4
101	5	5
1010	10	A
1011	11	B
1100	12	C
1101	13	D
1110	14	E
1111	15	F
10000	16	10
10100	17	11
100100	42	2A
1101111	111	6F