swiss physical society

Nr. 66, Januar 2022

SPG Mitteilungen
Communications de la SSP

Auszug - Extrait

Physik Anekdoten und persönliche Erinnerungen (24)

Datenverschlüsselung: Von Bürgis Logarithmus zur digitalen Enigma

Bernhard Braunecker

Physik Anekdoten und persönliche Erinnerungen (24)

Datenverschlüsselung: Von Bürgis Logarithmus zur digitalen Enigma

Bernhard Braunecker

Die SPG beteiligt sich alljährlich organisatorisch an den Jost Bürgi Symposien in Lichtensteig im Kanton St. Gallen, dem Geburtsort Bürgis (1552 – 1632). Man entdeckt im Laufe der Jahre immer mehr erstaunliche Eigenschaften dieses Zeitgenossen von Johannes Kepler und Tycho Brahe, mit denen Bürgi um 1600 herum in Prag zusammenarbeitete [1]. Im Gegensatz zu ihnen erlangte Bürgi nicht die historische Aufmerksamkeit, die er nach heutigem Wissen verdient hätte. So ist auch eine seiner Grosserfindungen, der Logarithmus, der heutigen Jugend kaum mehr geläufig; dies ganz anders als früher, als der Gebrauch von Rechenschiebern noch alltäglich war. In einem Referat am JB Symposium 2021 zeigten wir, dass dabei der Logarithmus höchst aktuell ist, da er sich bestens zur Verschlüsselung grosser Datenströme eignet. Allerdings muss man ihn mit Konzepten der Restklassenmathematik vereinen. Dann kann man das geniale Konzept der legendären Codiermaschine Enigma, bekannt aus dem 2. Weltkrieg, in neuer, nunmehr rein digitaler Form für schnelle und sichere Datencodierung im Alltag einsetzen [2].

Bei der mechanischen Ausführung der historischen Enigma zur Verschlüsselung von Texten aus einem Alphabet von 26 Grossbuchstaben wurden drei bis vier Codescheiben hintereinander auf einer gemeinsamen Drehachse eingebaut, wobei jede Scheibe an ihrer Vorder- und Hinterseite 26 im Uhrzeigersinn angeordnete Stromkontakte trug. Diese Kontakte waren in scheinbar zufälliger und geheim zu haltender Weise paarweise elektrisch miteinander verbunden. Drückte man eine der 26 Eingabetasten, wurde ein elektrisches Signal durch die Codescheibenanordnung geschickt und brachte eine der 26 Anzeigelampen zum Aufleuchten. Der Einbau und die Voreinstellung der Drehscheiben waren zeitraubend und gerade in kritischen Situationen fehleranfällig, hingegen erfolgte die eigentliche Verschlüsselung eines Zeichens sehr schnell und sehr sicher. Die allerdings grosse inhärente Schwäche, dass die mechanischen Einstellparameter der zu synchronisierenden Enigmas auf beiden Seiten in unverschlüsselter Papierform mitgeführt werden mussten, führte in der Folge jedoch zur Abkehr von der an und für sich leistungsstarken Verschlüsselungsmethode.

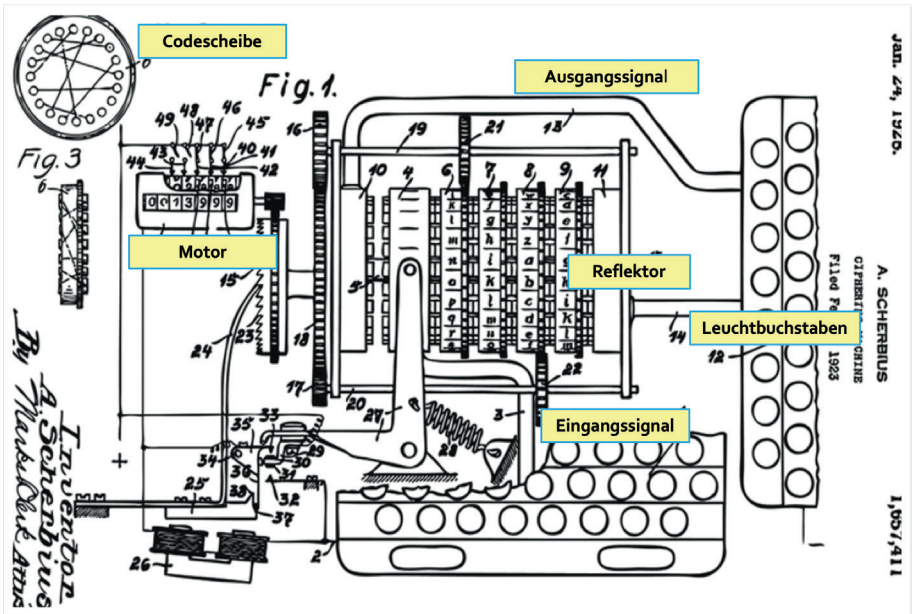
In digitaler Form ersetzt man die Drehscheibenverdrahtung durch Permutationsvektoren des gewählten Alphabets, und die nach jeder Zeicheneingabe vorzunehmenden Codescheibendrehungen durch entsprechende mathematische Shift-Operationen. Die sich ergebenden Permutationsvektoren der Gesamtanordnung kann man nun für eine gewisse Anzahl an Zeicheneingaben vorausberechnen und in einer Look-Up Tabelle LUT ablegen, so dass die anschliessend vorzunehmende eigentliche Verschlüsselung eines Textes oder einer Datenmenge ohne grossen Rechenaufwand sehr schnell, also mit der Taktfrequenz des Prozessors innerhalb von Bruchteilen von Mikrosekunden ablaufen kann.

Das erwähnte Risiko bei der Synchronisierung der Einstellparameter kann man in der digitalen Form eliminieren, indem für jeden Einstellparameter jede Seite eine zufällige und nur ihr allein bekannte Geheimzahl wählt. Diese Geheimzahlen werden mit Hilfe der Restklassenmathematik in neue Zahlen umgewandelt, die öffentlich übertragen werden können. Beide Parteien können aus ihnen mittels ihrer Geheimzahlen dieselbe Einstellzahl extrahieren, während dies für den unbefugten Lauscher praktisch unmöglich ist. Sie wissen also vorab bei der Wahl ihrer spontanen Geheimzahlen nicht, welche Einstellwerte sich ergeben werden; sie wissen nur, dass sie beide dieselben Werte erhalten werden. Das wird im zitierten Bericht ausführlich erläutert und mit Beispielen belegt.

Man kann also parallel zur momentanen Datenübertragung in Bruchteilen von Sekunden die nächste virtuelle Einstellung beider Enigmas, also die Anzahl der Codescheiben, deren Codierung, ihre Drehwinkelvoreinstellungen und die Grösse der Winkelschritte durch Austausch von nur wenigen und spontan gewählten Geheimzahlen ändern, und kann somit bereits die Look-Up Tabelle für das nächste Datenpaket vorbereiten. Vor diesem Feuerwerk an permanenten Änderungen wird im Alltag der genervte Lauscher schnell kapitulieren.

Beispiel einer digitalen Enigma

Sie soll Texte aus einem Alphabet von 127 alphanumerischen Zeichen verschlüsseln. Die Enigma bestehe aus

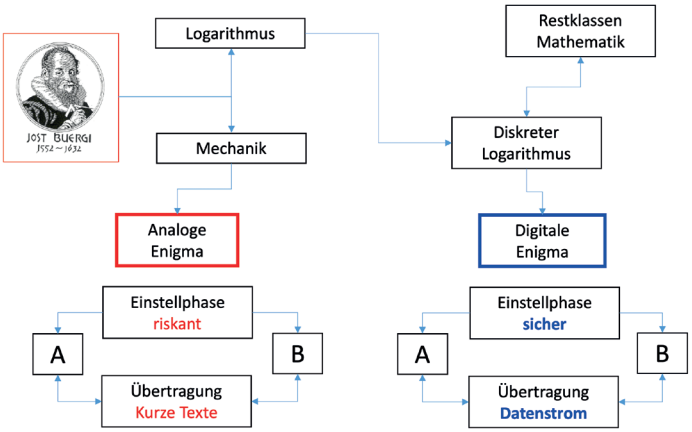


Enigma Patent 1923. Bild: Gemeinfrei

1 siehe den Artikel von Peter Ullrich in den *SPG Mitteilungen* Nr. 65, Seite 14.

2 *https://www.jostbuergi.com/bibliothek/mathematik/*

10 virtuellen Drehscheiben und einer Reflektorscheibe. Die Textcodierung erfolgt somit im doppelten Durchgang wie bei der klassischen Enigma. Man benötigt zur Erzeugung des Permutationsvektors einer Drehscheibe in unserem Fall 7 Geheimzahlen. Hinzu kommen zwei weitere Zahlen für die Drehwinkelvoreinstellung und die Winkelschrittweite. Bei 10 Scheiben inklusive des nicht-rotierenden Reflektors müssten somit lediglich 97 Geheimzahlen auf jeder Seite generiert und öffentlich übertragen werden. Damit wird der erste Permutationsvektor der Länge 127 der LUT erzeugt. Die weiteren LUT Zeilen ergeben sich durch die individuell verschiedenen, aber beiden Seiten bekannten Rotationen der 10 Codescheiben. Die Spaltenlänge der LUT ist frei wählbar.



Es soll folgender Probetext verschlüsselt werden: **ABC:0123456789?XYZ!** In unserem Beispiel ergab sich als erste Zeile der LUT der Permutationsvektor als

22	3	2	64	51	93	16	41	92	40
100	105	74	84	82	7	121	57	115	101
43	1	68	76	45	69	107	116	98	65
49	62	96	114	113	53	77	94	124	10
8	104	21	50	25	81	56	97	31	44
5	125	36	70	95	47	18	120	80	110
75	32	123	4	30	117	111	23	26	54
109	118	78	13	61	24	37	73	106	59
46	15	90	14	89	119	103	112	85	83
126	9	6	38	55	33	48	29	99	11
20	127	87	42	12	79	27	122	71	60
67	88	35	34	19	28	66	72	86	58
17	108	63	39	52	91	102			

Das erste Zeichen im Probetext, der Buchstabe **A** ist in unserem Alphabet als Nr. 34 indiziert und man entnimmt dem Vektor an dieser Stelle den Index 114, also das Zeichen **'**. Wegen des Reflektors muss an der Stelle 114 der Index 34 stehen. Die zweite Zeile der LUT ergab sich als

5	61	25	110	1	84	72	23	48	26
16	82	88	41	34	11	117	69	121	68
106	118	8	78	3	10	105	38	58	115
32	31	55	15	119	57	104	28	50	59
14	98	53	124	101	97	108	9	116	39
90	66	43	103	33	123	36	29	40	81
2	112	86	127	74	52	70	20	18	67
71	7	125	65	122	93	109	24	111	95
60	12	92	6	126	63	99	13	102	51
96	83	76	120	80	91	46	42	87	114
45	89	54	37	27	21	113	47	77	4
79	62	107	100	30	49	17	22	35	94
19	75	56	44	73	85	64			

Für das zweite Zeichen im Probetext, den Buchstaben **B**, der im Alphabet als Nr. 35 indiziert ist, findet man den Index 119, also das Zeichen **-** und umgekehrt.

Für die 19 Zeichen des Probetextes ergaben sich dann aus den ersten 19 Zeilenvektoren der LUT die Verschlüsselungsindizes

114	119	117	32	74	120	115	106	82	66
97	35	20	91	85	120	13	79	54	

was der zu übertragenden Buchstabenfolge **'–"?i–'‰qa€B3zt–,nU** entspräche. Es mag verwirren, dass das Zeichen **–** (Alphabet Index 120) zweimal im codierten Text vorkommt (an den Stellen 6 und 16), einmal als Verschlüsselung des Zeichens **1** und dann des Zeichens **X**. Dies ist eine Folge der Drehscheibenrotation. Es wäre sogar denkbar, wenn auch unwahrscheinlich, dass die codierte Botschaft nur aus lauter gleichen Symbolen bestehen würde.

Das Gebiet der modernen Datenverschlüsselung dürfte junge Leute sehr interessieren, da sie nicht nur wie bislang defensiv, also zum Schutz von Informationen eingesetzt werden kann, sondern sie sich auch proaktiv für neue Geschäftsmodelle wie *Geo-Fencing* anbietet [3]. Die Enigma Grundidee der intelligenten Voreinstellung, der permanenten Codeveränderung und der schnellen eigentlichen Datenverschlüsselung wird auch bei der digitalen Version beibehalten. Der Ersatz der Mechanik durch Mikroprozessoren erlaubt problemlos eine nach Belieben wählbare Verschlüsselungssicherheit, so dass sich in der Alltagspraxis der Aufwand zur Entschlüsselung meist nicht lohnt.

Hervorzuheben sei noch das soziale Moment dieser Art der Verschlüsselung, denn sie erfordert den permanenten Informationsaustausch und Datenabgleich zwischen beiden Parteien im gegenseitigen Benehmen. Dieser moralische Aspekt zusammen mit der technischen Herausforderung bei der Programmierung macht das Thema einer digitalen Enigma ideal für Maturaarbeiten.

3 *Geo-Fencing*: Der Betrieb einer geleasten Bau- oder Landwirtschaftsmaschine erfolgt innerhalb eines Areals, dessen Grenzkoordinaten mittels Satellitennavigation bestimmt und der online verbundenen Zentrale übermittelt werden. Diese schätzt die zu erwartenden Betriebskosten ab und schaltet die Software frei, wenn der Kunde den Betrag akzeptiert. Es ist verständlich, dass der subtile Datenaustausch zwischen der Zentrale und den Maschinen vor Missbrauch geschützt werden muss.

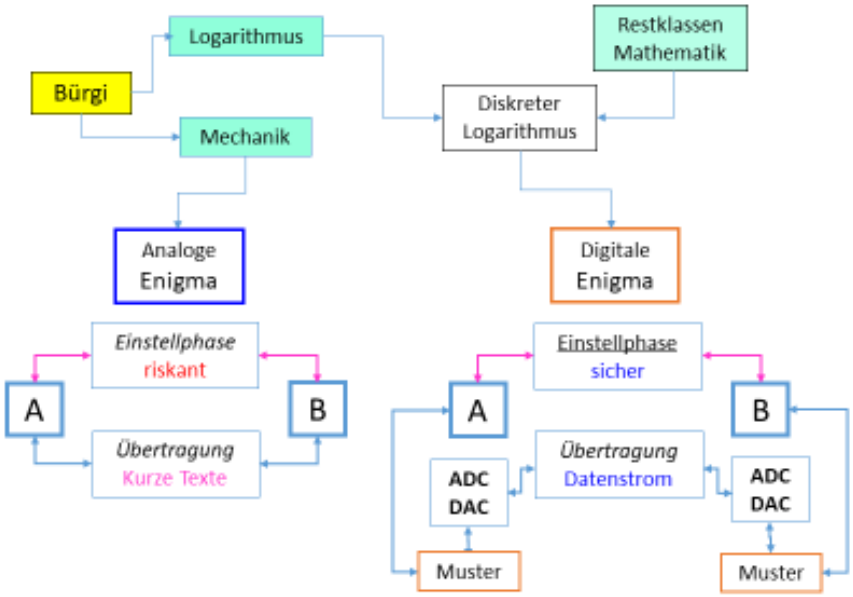
Datenverschlüsselung: Von Bürgis Logarithmus zur digitalen Enigma

Bernhard **Braunecker,** Rebstein

Abstract: Die Kombination des Logarithmus Bürgischer Art mit der Methode der Restklassenmathematik eignet sich gut zur Verschlüsselung schneller Datenströme. Damit kann man auch die Schwächen der legendären Enigma überwinden und sie in eine leistungsstarke digitale Version überführen.

The Bürgi Logarithm when combined with residual mathematics is suitable for modern encryption of fast data streams. The genious concept of the Enigma machine known from the 2nd World War is reconsidered by inserting this modification into a digital Enigma variant.

Übersicht: Die Abbildung zeigt den Aufbau unserer im Folgenden zu erläuternden Betrachtungen. Ausgehend von der präzisen Mechanik der Uhren und Apparate von Jost Bürgi (1552 – 1632) gelangen wir zur klassischen Enigma. Trotz ihres genialen Verschlüsselungskonzepts war sie anfällig auf Lauschangriffe, da ihre synchron vorzunehmende Einstellung bei Sender und Empfänger über nichtcodierte Logbücher erfolgte. Dieses Risiko wäre zu verhindern gewesen, wenn man die Bürgische Erfindung des Logarithmus eingesetzt hätte, allerdings in Verbindung mit der seit fast 2000 Jahren bekannten Restklassenmathematik. Nur hätte das Rechenleistungen erfordert, die erst heutzutage zur Verfügung stehen. Dennoch ist das Enigma Konzept eine Neubetrachtung wert, da es erlaubt, nicht nur Textbuchstaben, sondern ganze Bildmuster mit hoher Datenrate und gesichert zu übertragen.



Für Bürgianer ist es oft schwierig, der heutigen Jugend Bürgis Logarithmus näher zu bringen. Seine frühere Sichtbarkeit wegen des alltäglichen Gebrauchs von Rechenschiebern ist nicht mehr gegeben. Deshalb war auch eine Motivation bei dieser Studie, das Augenmerk auf die Kombination von Logarithmus *und* Datenverschlüsselung zu richten. In der obigen Abbildung liessen sich dann eventuell für Schülerarbeiten geeignete Schnittstellen finden, um damit junge Leute nicht nur für heutige Fragestellungen zu sensibilisieren, sondern auch um sie für Geschichte zu begeistern.

Inhalt

1 EINLEITUNG

Die Verschlüsselung von Daten war in der Vergangenheit meist militärisch bedingt, während heutzutage der Schutz privater Personendaten, aber vermehrt auch der von Maschinen an Bedeutung gewinnt. Wenn immer mehr Apparate vernetzt und durch Algorithmen gesteuert werden, dann häufen sich leider auch destruktive Cyberattacken mit unter Umständen schlimmen Folgen. Allerdings bietet die moderne Datenvernetzung auch Möglichkeiten für neuartige digitale Geschäftsmodelle, bei denen man die Verschlüsselung von Daten nicht nur defensiv einsetzt, sondern um aktiv vorteilhafte Bedingungen für die Kunden zu schaffen (siehe 3.4).

Während man heute dank leistungsstarker Digitalrechner die Datenverschlüsselung algorithmisch vornehmen kann, waren es in der Vergangenheit mehr einfachere Konzepte, die deswegen auf grosse Zahlen setzen mussten. Da boten sich Potenzen zur Codierung und Logarithmen à la Bürgi zur Decodierung an. Um zum Beispiel die geheime Zahl A zu übermitteln, wäre die verschickte und öffentlich zugängige Botschaft $C = p^A$ im Falle von A = 17 und p = 13 bereits eine riesige Zahl mit 18 Zehnerpotenzen, die unbefugte Lauscher von vornherein entmutigen sollte. Nur hätte sich irgendwann das Verschlüsselungskonzept der Potenzierung herumgesprochen, und dann wäre die dem Eindringling anfangs noch unbekannte Zahl p = 13 durch leichtes Probieren schnell erkennbar, da nur sie bei der Decodierung eine ganze Zahl der Nachricht A liefert. Dennoch bleibt der logarithmische Ansatz weiterhin sinnvoll, aber er wäre stark zu modifizieren, wie im Folgenden gezeigt wird.

1.1 DIE PRÄZISION BÜRGISCHER INSTRUMENTE

Als zu Beginn der Neuzeit die Menge der zu übermittelnden und somit zu schützenden Daten stark anstieg, mussten geeignete mechanische Codiermaschinen gebaut werden, bei denen interne Teile sich präzise und verlässlich drehten. Bei deren Auslegung konnte man auf viele historische Arbeiten zurückgreifen, und hier sind besonders auch die feinmechanischen Konstruktionen der Bürgischen Uhren und Himmelsgloben zu nennen. Bei den Bürgi Apparaten beeindruckt uns heute noch, wie es ihm gelang, zum Beispiel bei Zahnrädern die komplizierte Verzahnungsform so zu optimieren und sie zusammen mit der Achslagerung so präzise zu fertigen, dass die Drehbewegungen eines Getriebes ruckfrei mit minimalem Drehmomentverlust abliefen. Nur so konnten seine Uhren bereits verlässlich die Sekunde anzeigen, selbst wenn ungünstige Umgebungsbedingungen wie Stösse, Vibrationen, Feuchte- und Temperaturschwankungen vorlagen.

1.2 MODERNE TECHNISCHE ANWENDUNGEN VON DREHSCHEIBEN

Bei den Uhren haben die Zahnräder die Aufgabe, die getakteten, schnellen Schwingungsbewegungen eines Oszillators, der Unruh, geeignet zu untersetzen und ihre Anzahl als Drehwinkel eines Zeigers anzugeben. Bei den Himmelsgloben werden durch Federantrieb Getriebezahnräder bewegt, um die Bahnen stellarer Objekte in sphärischen Koordinaten zu visualisieren. Die Zahnräder übertragen also bei den Uhren nur die Kenntnis über den Schwingungszustand der Unruh und bei den Globen nur die Kenntnis über die relative Lage von Objekten auf der Himmelskugel. Sie dienen in beiden Fällen passiv der Weitergabe von Information. Das schöpft nicht das Potential von Getrieben aus, denn man kann die einzelnen Drehteile zusätzlich mit Funktionalitäten versehen, so dass das Zahnradkollektiv aktiv eine bestimmte Aufgabe erfüllt. Belegt man zum Beispiel die auf einer gemeinsamen Achse sich drehenden Zahnräder mit elektrischen Kontakten, kann man damit ein Rechenwerk zur Datenverschlüsselung bauen, was im Folgenden erläutert werden soll. Bestückt man die Drehscheiben mit Segmenten

neuartiger Magnetmaterialien, kommt man zu modernen Scheibenmotoren von sehr hoher Leistungsdichte (**Abb. 1**).[1] Lässt man Zahnräder, die optische Bauteile wie Prismen oder Hologramme tragen, mit hoher Geschwindigkeit rotieren, ergeben sich schnelle Trepanier-Scanning Systeme, um Laserstrahlen räumlich und winkelmässig je dreidimensional abzulenken. Damit können Materialoberflächen schneller, genauer und flexibler als mit Bohrer oder Fräser bearbeitet werden. (**Abb. 2)** Allerdings sind die Anforderungen an die Scannerkonstruktion wegen der auf Mikrometer genauen und reproduzierbaren Führung des Laserstrahls auf dem Material sehr hoch. Bei Rotationsgeschwindigkeiten bis zu 30‘000 Umdrehungen pro Minute ergeben sich abhängig von der gewählten Bahnkurve des Laserstrahls auf der Oberfläche sehr starke Winkelbeschleunigungen der optischen Drehteile, die zu hoher Belastung der Achslager führen würden, wenn man sie nicht aufwendig konstruktiv abfangen würde.

Abb. 1: Moderner Scheibenmotor mit axialem Magnetfluss, Durchmesser ≈ 5 inch

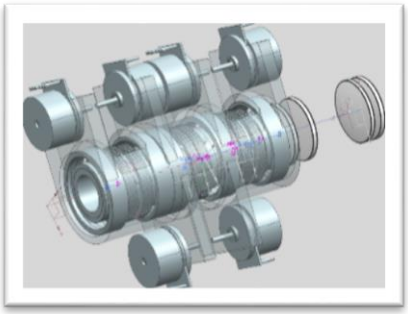


Abb. 2: Sechs Motoren drehen Hologramme zur ultraschnellen Ablenkung eines Laserstrahls in sechs Dimensionen. Bild: Braunecker Engineering GmbH

2 ENIGMA (HISTORISCH)

Unter den historischen mechanischen Verschlüsselungsmaschinen ragt zweifelsohne die Enigma heraus, nicht nur wegen ihres genialen Konzepts, sondern auch wegen ihrer wichtigen Rolle im 2. Weltkrieg. Erfunden hatte sie bereits anfangs des 20. Jahrhunderts Arthur Scherbius (1878 -1929), der sie ursprünglich für den zivilen Postverkehr auslegte und der 1918 einen ersten Patentschutz erlangte **(Abb. 3**).[2]

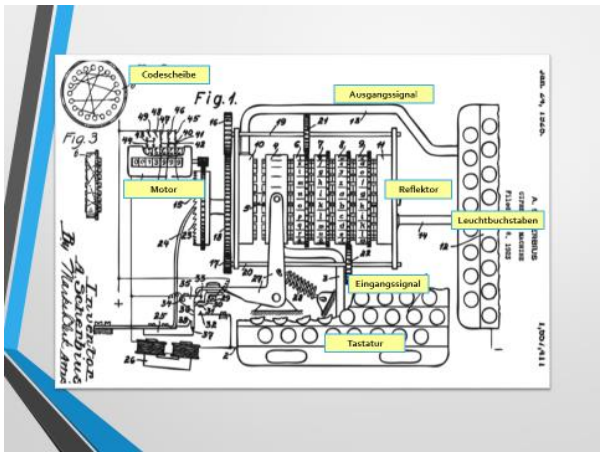


Abb. 3: Enigma Patent 1923. Bild: Gemeinfrei

Im Wesentlichen handelt es sich um eine Art von Schreibmaschine mit einer Eingabetastatur für 26 Grossbuchstaben und einer entsprechenden Ausgabeeinheit von 26 Leuchtlampen. Dazu gibt es etwa sechs Codescheiben, wobei jede Scheibe auf ihrer Vorder- und Hinterfläche 26 im Uhrzeigersinn angeordnete Stromkontakte trägt, die jedoch in scheinbar zufälliger Weise paarweise elektrisch miteinander verbunden sind.

Man setzt nun drei oder vier von ihnen auf derselben Drehachse hintereinander so ein, dass über die Stifte elektrischer Kontakt zwischen den Scheiben besteht. Drückt man eine der 26 Buchstabentasten, so wird eine bestimmte, scheinbar willkürliche Stromverbindung durch eine stationäre erste Codescheibe (Transmitter) und die drei oder vier Drehscheiben geschaltet, anschliessend an einer stationären Scheibe (Reflektor) umgeleitet und wieder durch die Codescheiben, nur diesmal an anderer Stelle, zurückgeführt und als Leuchtbuchstabe

[1] Magnax AXF225**,** Yokeless Axial Flux PM Motor, Gewicht: 16 kg, Leistung: 200 kW = 272 PS, https://www.magnax.com/magnax-blog/axial-flux-motors-and-generators-shrink-size-weight

[2] https://de.wikipedia.org/wiki/Enigma_(Maschine)

angezeigt. Nach jeder Buchstabeneingabe drehen sich die Codescheiben um bestimmte Winkelschritte, so dass jedes Mal ein anderer Codebuchstabe aufleuchtet, selbst wenn man immer dieselbe Eingangstaste drückt. Man kann nun wählen, welche Codescheiben und in welcher Reihenfolge sie eingebaut werden, wie ihre Winkelvoreinstellungen sind und wie zusätzlich die Ein- und Ausgangszeichen noch durch Kontaktstöpsel willkürlich umgeschaltet werden können.

Zieht man all die gebotenen technischen Möglichkeiten in Betracht, kommt man auf 10^{114} Einstellvarianten, was auch heutzutage noch als praktisch unlösbar gilt. Nur müssten die Einstellparameter beiden Seiten bekannt sein, was damals durch Ausgabe eines hochgeheimen Sonder-Maschinenschlüssels SMS in Papierform geschah. Fiel nur ein Exemplar in unbefugte Hände, konnten unter Umständen weitere Enigma Kontakte gefährdet sein.[3] Dieser schwerwiegende Nachteil führte später zur Abkehr vom Enigma Konzept, dabei könnte er durch Einführung des diskreten Logarithmus überwunden werden.

2.1 Diskreter Logarithmus und Restklassenmathematik

Der eingangs erwähnte Verschlüsselungsansatz $C = p^A$ übersteigt, wenn p und A sehr grosse Zahlen sind, schnell die Rechengenauigkeit normaler Digitalcomputer. Es ist daher sinnvoll, die Codeoperation im Rahmen der Restklassenmathematik durch die Modulo Operation (mod) als $C = p^A \bmod(N) = n_{Zyk}\, N$ + Rest auszuführen, wobei n_{Zyk} die grösstmögliche ganze Zahl ist, die den Rest minimiert. So bekommt man für C = 190 Sekunden und N = 60 Sekunden n_{Zyk} = 3 plus 10 Sekunden Rest. Dieser Ansatz lässt sich mechanisch mit drehenden Zahnrädern illustrieren und konstruktiv verwirklichen. Man gewinnt zwei entscheidende Vorteile: erstens liegen die numerischen Werte von C nunmehr im Genauigkeitsbereich handelsüblicher digitaler Rechner und erlauben weiterführende Rechenoperationen, und zweitens macht die Modulo Operation die Extraktion der Geheimbotschaft A aus C beliebig schwierig. Im **Anhang 5.1** werden einige Grundregeln ganzzahliger Rechenoperationen wie die Bildung von Summen, Produkten und Polynomen am Beispiel eines Systems aus den vier Restklassen {3, 4, 5, 7} erläutert. Das entspräche mechanisch einem Getriebe aus vier Zahnrädern mit Durchmesser 3, 4, 5 und 7 in beliebigen Einheiten, das mittels einer Kurbel stufenweise gedreht wird, um Rechenoperationen auszuführen.

2.2 Schneller Algorithmus

Da im Verschlüsselungsansatz $C = p^A \bmod(N)$ grosse Zahlenwerte für p, A und N benutzt werden sollen, sind in der Praxis effiziente Rechenmethoden gefragt.

Es sollen als Beispiel die zu übermittelnde Nachrichtenzahlen A als $C = 431^A \bmod(527)$ übertragen werden, wobei A als 12 Bit Zahl darstellbar ist. Als erstes werden die infrage kommenden Residuen von {431^(2^0), 431^(2^1), 431^(2^2), 431^(2^3), … , 431^(2^{11})} mod(527) berechnet und abgespeichert (**Abb. 4**).

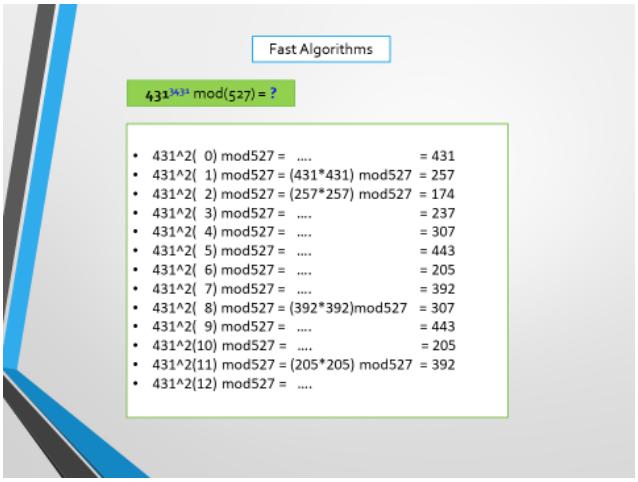


Abb. 4: Abgespeicherte Residuen für Zweierpotenzen

[3] Dies geschah vermutlich am 9. Mai 1941, als das deutsche U-Boot U 110 vor Schottland von der Royal Navy manövrierunfähig geschossen wurde, und die Alliierten aus dem von der Besatzung verlassenen U-Boot sowohl die Enigma M3 wie auch die Codeliste bergen konnten. https://de.wikipedia.org/wiki/U_110_(Kriegsmarine)

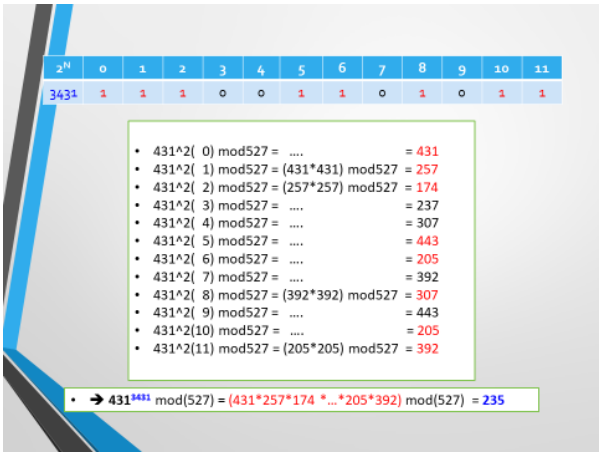


Abb. 5: Berechnung der öffentlichen Botschaft C = 235 für die geheime Nachricht A = 3431

Dabei nutzt man aus, dass $x^{(2^{n+1})} \bmod(Z) = [x^{(2^n)} \bmod(Z)]^2$, das heisst man multipliziert nur zwei kleine Zahlen. Liegt die zu übermittelnde Nachricht A vor, werden aus der Datentabelle nur die Residuen entnommen, bei denen das 12-Bit Wort von A gleich 1 ist. Nur diese Werte werden miteinander multipliziert und Modulo Z genommen.

Dies ist in **Abb. 5** für A = 3431 erläutert, und man erhält $C = 431^{3431} \bmod(527) = 235$.

2.3 Sichere Übertragung der Einstellparameter

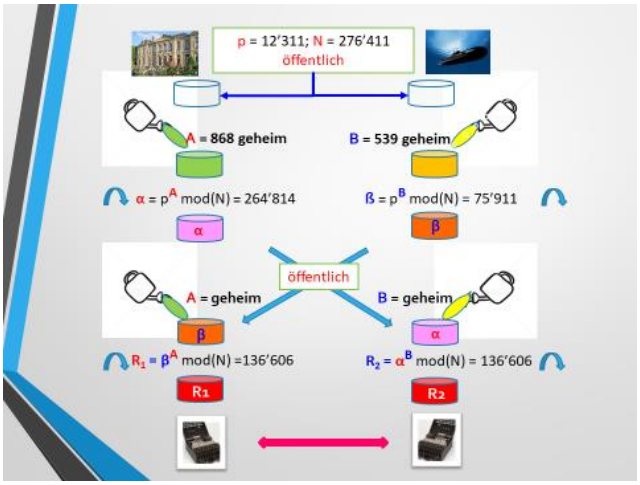


Abb. 6: Übertragung des Einstellparameters R

Der geschützte Datenaustausch zwischen zwei Parteien benötigt dieselben Enigma Einstellparameter auf beiden Seiten, wobei unerheblich ist, welche.

Das Vorgehen wird im Beispiel von **Abb. 6** erläutert: Beide Seiten wollen eine bestimmte Codescheibe ihrer Enigma mit 96 Winkeleinstellungen gemeinsam um einen Winkel φ vorverdrehen. Sie einigen sich im Voraus auf Zahlenwerte von p = 12 311 und N = 276 411, die auch veröffentlich werden dürfen. Partei 1 wählt nun die nur ihr allein bekannte Geheimzahl A = 868 und bildet die öffentlich zugängige Nachricht $\alpha = p^A \bmod(N) = 264\,814$. Partei 2 wählt als Geheimzahl B = 539 und veröffentlicht $\beta = p^B \bmod(N) = 75\,911$. Partei 1 berechnet $R_1 = \beta^A \bmod(N) = 136\,606$ und Partei 2 berechnet $R_2 = \alpha^B \bmod(N) = 136\,606$, also beide erhalten denselben Wert R. Beide Parteien würden somit ihre Enigma Codescheibe um denselben Drehwinkel $\varphi = (2\pi/96) * a$ mit $a = R \bmod(96) = 94$ voreinstellen.

Sie wissen also stets im Voraus, dass sie dieselben Einstelldaten aus dem öffentlich zugänglichen Funkverkehr entnehmen und folglich die gleichen Maschineneinstellungen vornehmen können, aber sie wissen nicht welche. Es ist naheliegend, dass man als zusätzliche Erschwernis für die unbefugten Lauscher als erstes neue, nun allerdings nicht mehr öffentliche Daten p und N erzeugt.

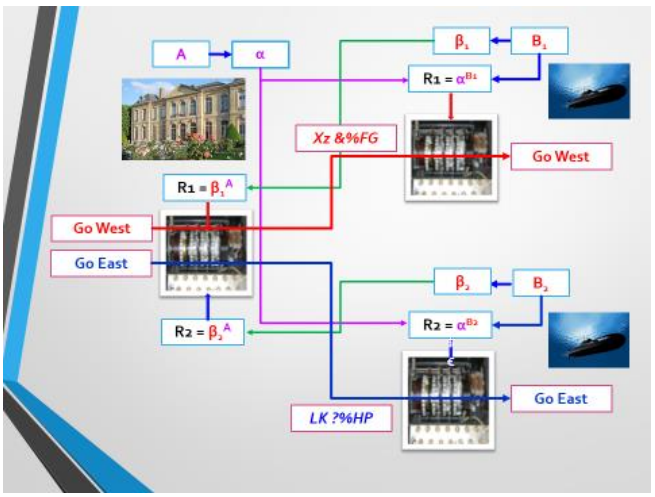


Abb. 7: Datenaustausch: Zentrale (links) und Rover (rechts)

2.4 TIMESHARING MODUS

In **Abb. 7** wird illustriert, wie ein sicherer Datenaustausch zwischen der Zentrale und mehreren Parteien stattfinden könnte. Partei 1 (links) sei die Zentrale, Hauptquartier etc., genannt Base, die verschiedene Nachrichten an ihre Filialen, U-Boote etc., genannt Rover, senden muss. Wiederum seien die Zahlen p und N allen bekannt, auch den Lauschern. Die Zentrale wählt die nur ihr allein bekannte Geheimzahl A, bildet $\alpha = p^A \bmod(N)$ und sendet es per Funk in alle Welt. Die einzelnen Rover wählen die nur ihnen bekannte Geheimzahl B, bilden $\beta = p^B \bmod(N)$ und senden es zu einer bestimmten Uhrzeit an die Zentrale. Zu diesem Zeitpunkt werden beide Enigmas entsprechend eingestellt, und es wird der eigentliche Geheimbefehl als Text übermittelt und vom Rover beantwortet. Anschliessend an Rover 1 kommt im Timesharing Verfahren Rover 2 an die Reihe. Da die zentrale Enigma für Rover 2 umgestellt werden muss, kann Rover 1 nicht den an Rover 2 gesendeten Befehl entschlüsseln, ein zusätzlicher Sicherheitsgewinn.

3 ENIGMA (DIGITAL)

3.1 PUBLIC KEY VERFAHREN, DIFFIE-HELLMAN-MERKLE TRANSFORMATION

Die heutzutage in der Praxis eingesetzten sogenannten Public Key Verschlüsselungen sind sehr ähnlich zu der hier beschriebenen Enigma Verbesserungsvariante (2.3) aufgebaut. Im Wesentlichen merken sich beide Parteien je zwei geheim zu haltende riesige Primzahlen, aus denen dann sowohl geheime wie öffentliche Zahlen gebildet werden, mit denen die eigentliche Nachricht im Modulo Verfahren verschlüsselt wird.[4] [5] [6] Der Vorteil ist, dass die Vorübertragung von Einstelldaten entfällt; nachteilig ist jedoch, dass jede Botschaft am Sende- wie am Empfangsort berechnet werden muss. Das geschieht zwar schnell, ist aber dennoch mit Zeitaufwand verbunden. Die Frage stellt sich, ob es eine Variante zwischen dem Enigma Prinzip und den Public Key Verfahren gäbe, und welches ihre Vorteile wären?

3.2 DIGITALE SIMULATION DER HISTORISCHEN ENIGMA

Heutzutage erfolgt die Datencodierung nicht mehr mechanisch, sondern nur noch elektronisch digital mit signifikant kürzeren Übertragungszeiten. Die **Abb. 8** zeigt zwei von uns in Matlab durchgeführte Verschlüsselungen des Textes *Alle Menschen werden Brueder* für eine simulierte Enigma mit 96 alphanumerischen Zeichen. Sie bestand aus einer stationären Eingangscodescheibe, vier rotierenden Codescheiben und einem stationären Codereflektor. Bei der zweiten Simulation wurde das Kollektiv der

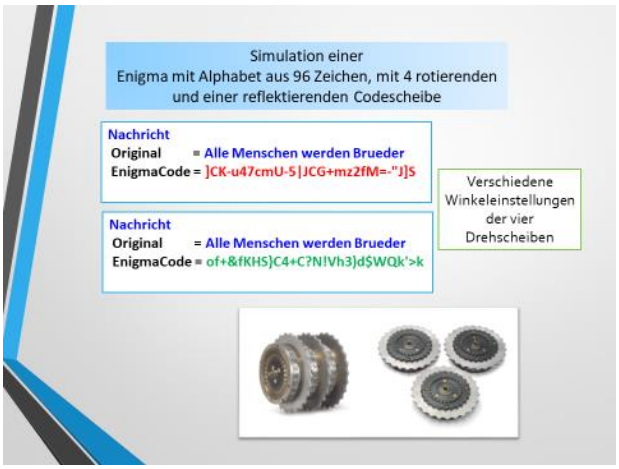


Abb. 8: Simulation derselben Botschaft mit zwei verschiedenen Enigma Einstellungen

[4] The Mathematics of Public-Key Cryptography, M E Hellman, Scientific American Inc, August 1979, p 146 ff, http://www.personal.psu.edu/tcr2/311w/Hellman1979.pdf

[5] Privacy and Authentication: An Introduction to Cryptography, W Diffie and M E Hellman, Proceedings of the IEEE, Vol. 67, No. 3, March 1979, p 397 ff

[6] New Directions in Cryptography, W Diffie and M E Hellman, IEEE Transactions on Information Theory, Vol. IT- 22, No.6, November 1976, p 644 ff

vier Drehscheiben gegenüber der ersten um einen Winkelschritt weitergedreht, und man erhält wie ersichtlich eine zur ersten völlig unkorrelierte Codefolge.

Da wir die Verdrahtung aller sechs Einzelscheiben und die Voreinstellungswinkel der vier Drehscheiben kannten, liess sich für jeden der 96 Drehwinkel des Vierer-Scheibenkollektivs die Permutation der 96 Eingangszeichen vorab berechnen und in einer 96 x 96 grossen Look-Up Tabelle (LUT) ablegen. Der Zeitgewinn bei der eigentlichen Textverschlüsselung war beträchtlich, da man keine Rechenoperationen mehr durchführen musste, sondern lediglich in der LUT nachsah. Die Ver- und Entschlüsselung eines Zeichens geschahen daher im Wesentlichen mit der Taktfrequenz des Prozessors innerhalb von Bruchteilen von Mikrosekunden.

3.3 Digitales Konzept

Die Enigma war seit Anbeginn zweistufig aufgebaut: in einer Vorstufe wurden von beiden Seiten gemäss der ihnen vorliegenden Information die Maschinen eingestellt; danach in der Hauptstufe begann die eigentliche Ver- und Entschlüsselung von Texten. Da damals hauptsächlich militärische Texte von Hand und oft in kritischen Situationen eingegeben werden mussten, war mit Bedienungsfehlern in beiden Phasen zu rechnen. Es galt also die Einstellung und die Durchführung möglichst zu vereinfachen. Da die Texte aus nur wenigen kurzen Sätzen bestanden, war naheliegend, immer nur ein Zeichen in ein anderes Zeichen aus demselben Alphabet umzuwandeln, dieses auszulesen und erst danach das nächste Zeichen einzugeben. Ebenfalls musste man mit möglichst wenigen Codescheiben auskommen, sei es aus technischen Gründen, aber auch um möglichst wenige mechanische Einstellungen auf beiden Seiten vornehmen zu müssen. Die trotz dieser Vereinfachungen erreichte hohe Datensicherheit der Übertragung wurde durch den genialen Ansatz eines sich drehenden Scheibenkollektivs im Doppeldurchgang zwischen zwei stationären Codescheiben erreicht.

Auch wenn heutzutage die Codierung digital durchgeführt wird, sollte die bewährte Trennung in Einstell- und Betriebsphase beibehalten werden, aber anders gewichtet. Anders als früher stehen bereits in der Einstellphase leistungsstarke Prozessoren zur Verfügung, und sie kann somit rechenaufwändig gestaltet werden. Hingegen sollte der eigentliche Verschlüsselungsvorgang trotz schneller Hardware so einfach wie möglich ausgelegt werden, da man durchaus mit Übertragungsraten der zu verschlüsselnden Datenströme von mehreren 100 Gigabit/s oder in Glasfasern bereits mit 50 Terabit/s rechnen muss. Man muss dann zwar immer ganze Textblöcke zwischenspeichern, aber das sollte nicht noch durch zusätzlichen Rechenaufwand verkompliziert werden. Das in 3.2 angedeutete LUT-Konzept dürfte dafür die schnellste Variante sein.

3.3.1 Konfiguration

Die digitale Enigma für ein Alphabet von n Textzeichen bestehe weiterhin aus K Codescheiben und einem Reflektor. Die Anzahl K kann sehr gross gewählt werden. Die synchrone Codierung aller K Scheiben wird gemäss 2.3 vorgenommen, ebenso, welche der K Scheiben sich drehen sollten, welche Voreinstellwinkel sie bekämen, und wie gross die Winkelschritte für jede Scheibe wären. All diese Einstelldaten könnten in Übermittlungspausen innerhalb von Millisekunden verändert werden und würden zu stets neuen Code-LUTs auf beiden Seiten führen.

3.3.2 Beispiel

Im Anhang 5.3.2 wird quantitativ das Konzept für ein Test-Alphabet aus 31 Textzeichen und für 6 Codescheiben plus Reflektor als siebte Scheibe beschrieben. Statt der Verdrahtung einer Codescheibe wird die Permutation der 31 Eingangskontakte berechnet. Die permutierten Werte sind wiederum die Eingangswerte für die anschliessende nächste virtuelle Codescheibe. Im Test sind die Scheiben 1, 3, 5 und der Reflektor stationär, während sich die Scheiben 2, 4, 6 nach jeder Zeicheneingabe um einen Winkelschritt weiterdrehen.

3.3.3 Vor- und Nachteile des Konzepts

Die Anordnung kann während des Betriebs problemlos auf beiden Seiten und ohne grösseren Zeitaufwand auf andere n und K Werte und auf andere Codierungen umgestellt werden. Die freie Wahl der *Seeds* als eine Art der Winkelvoreinstellung gibt zusätzliche Freiheitsgrade für die Verschlüsselung. Die virtuelle Codescheibendrehung ist weiterhin wichtig, um im verschlüsselten Text Redundanzen wie das gehäufte Auftreten bestimmter Zeichen nicht widerzuspiegeln. Ebenso ist ein virtueller Reflektor von Vorteil, da beide, Sender und Empfänger, die verschlüsselte Nachricht unmittelbar vor dem Aussenden, bzw. nach Erhalt noch auf ihre Korrektheit überprüfen können, um sicher zu gehen.

Neben diesen Vorteilen muss man auch die Nachteile sehen: So zeigt sich rückblickend, dass die damals aus technischen Gründen vorgenommene gemeinsame Drehung der Codescheiben ebenso wie der Gebrauch der Reflektorscheibe die Komplexität der Verschlüsselungen zwar in gewünschter Weise so sehr erhöhten, dass eine unbefugte Entschlüsselung mit den damaligen Methoden spontan nicht möglich war. Aber es war vermutlich auch schon damals klar, dass systematische Eingriffe wie die Drehung eines Kollektivs oder ein doppelter Durchgang durch die Codescheiben wegen des Reflektors in einer Codiermaschine eigentlich nichts verloren hätten. Sie bieten ähnliche Angriffsstellen wie in den Texten stets wiederkehrende Schlüsselworte. Die bereits angedeuteten Möglichkeiten, die Scheiben individuell verschieden zu drehen und auch den Reflektor nicht permanent einzusetzen, entspannen das Problem erheblich. Nur sollte man sich im Klaren sein, dass jede wie auch immer konfigurierte Enigma mit entsprechend grossem Rechenaufwand geknackt werden kann; ob sich allerdings der Aufwand lohnt, bleibt in jedem Einzelfall abzuklären.

3.4 Anwendung: Geo-Fencing

Eine moderne Anwendung für Datenverschlüsselung ist das Markieren von Strassen, Sport- und Flugplätzen, Parkarealen, Hafenanlagen etc. Dazu setzt man immer mehr halb oder voll automatisch fahrende Linemarker ein, bei denen die Farbspritzdüse auf einer senkrecht zur Fahrtrichtung beweglichen Spindel montiert ist (**Abb. 9**). Die Spindel wird motorisch bewegt, um Linien beliebiger Kurvengeometrie zu markieren, aber auch um Fahrfehler und Geländeirregularitäten auszugleichen. Die dazu nötigen Regelsignale werden mit einer Genauigkeit von ±1 cm in Echtzeit von Satelliten geliefert. Da diese Maschinen teuer sind, ist folgendes Geschäftsmodell attraktiv: Der Kunde fährt mit der von ihm geleasten Maschine zu einigen wenigen Punkten, die das zu markierende Feld eindeutig in Lage, Ausrichtung und Grösse charakterisieren. Die mittels Satellitennavigation bestimmten Punktkoordinaten werden der on-line verbundenen Zentrale übermittelt, die für die gewünschte Feldfigur den Farbverbrauch abschätzt und die zu erwartenden Kosten angibt. Wenn der Kunde diesen Betrag akzeptiert, wird die Software innerhalb der Feldkoordinaten (Geo-Fencing) freigeschaltet. Der Kunde zahlt also nur für die Farbmenge, die er benötigt. Es ist verständlich, dass der subtile Datenaustausch der Zentrale mit Linemarkern in aller Welt bestens gesichert sein muss. [7] [8]

Abb. 9: Satelliten gesteuerter Linemarker

4 Zusammenfassung

Unsere Motivation zur Betrachtung der Datenverschlüsselung kam durch die Beschäftigung mit dem Werk Bürgis zustande. Beeindruckt von der Präzision und Zuverlässigkeit seiner Uhren und Apparate wollten wir den mechanischen Drehteilen mehr Funktionalität zuteilen. Wir kamen so zur klassischen

[7] https://lidarmag.com/issue/volume-10-issue-06/

[8] https://www.sps.ch/artikel/diverse-artikel/cyberphysische-systeme-und-autonome-mobilitaet

Enigma, bei der die Zahnräder mit speziell verdrahteten Stromkontakten bestückt waren, und so eine nahezu unbegrenzte Anzahl an Verschlüsselungen von Texten ermöglichten. Es mussten jedoch Art und Einstelldaten der Codescheiben dem Empfänger bekannt sein, und hier bot sich zur Risikominimierung eine weitere Bürgi Erfindung an, der Logarithmus. Er müsste allerdings modifiziert werden durch Einbezug der Restklassenmathematik. Sie wurde bereits im 3. Jahrhundert unserer Zeitrechnung in einer Arbeit des chinesischen Mathematikers Sun Zi erwähnt [9] und ausführlich von Qin Jiushao im Jahr 1247 beschrieben. [10] Ihre Betrachtungen gehen vermutlich auch auf periodische mechanische Systeme zurück, so dass wir über diesen historischen Exkurs wieder zu unserer ersten Bürgi Motivation, den Zahnrädern, zurückkämen.

Der beschriebene zweistufige Ablaufprozess der klassischen Enigma, bestehend aus einer von beiden Seiten gemeinsam durchzuführenden Einstellphase und anschliessender sehr schneller Zeichencodierung sollte auch bei einer modernen digitalen Variante beibehalten werden. Die früher riskante Übertragung der Einstellparameter kann dank vorhandener Rechenkapazität mit einem Minimum an Daten nunmehr sicher und flexibel vorgenommen werden. Die bereits früher sehr schnell durchgeführte Zeichencodierung durch einen Stromimpuls kann auch heutzutage durch Auslesen des Codezeichens aus einer vorberechneten Look-Up Tabelle sehr zeiteffizient vorgenommen werden, eine Notwendigkeit angesichts hoher Datenraten.

5 Anhang I

5.1 Mathematische Operationen mit Residuen

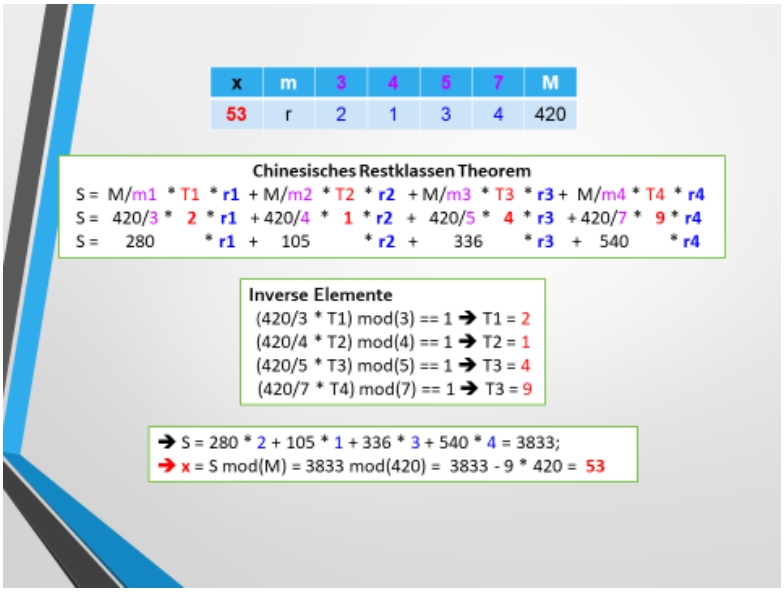
In **Tab. 1** sind für die vier Restklassen m = {3, 4, 5, 7} die Residuen r für x = 11, y = 35, für deren Summe und Produkt, sowie für das Polynom P(x) = $3x^2$ + x +1 an der Stelle x und y aufgetragen. So findet man zum Beispiel für das Polynom an Stelle x in Restklasse 7 den Wert r = P(11) mod(7) = 4.

M = 420	r	m1	m2	m3	m4
		3	4	5	7
x = 11	r_x	2	3	1	4
y = 35	r_y	2	3	0	0
∑ = x+y = 46	r_Σ	1	2	1	4
∏ = x*y = 385	r_Π	1	1	0	0
Polynom P(x) = 3 x^2 + x + 1					
P(x) = 375	r_P	0	3	0	4
P(y) = 3711	r_P	0	3	1	1

Tab. 1: Einige arithmetische Operation im Restklassensystem {3, 4, 5, 7}

[9] https://de.wikipedia.org/wiki/Sun_Zi_(Mathematiker)
[10] https://de.wikipedia.org/wiki/Qin_Jiushao

Die Rücktransformation ins Dezimalsystem kann man mit Hilfe des chinesischen Restklassensatzes vornehmen. Nach **Tab. 2** bekommt man für M = 3*4*5*7 = 420 die folgenden Koeffizienten S1 = 280, S2 = 105, S3 = 336 und S4 = 540. Addiert man sie gewichtet mit den Residuen (r_1, r_2, r_3, r_4) und nimmt die Summe modulo M, erhält man die gesuchte Dezimalzahl.



x	m	3	4	5	7	M
53	r	2	1	3	4	420

Chinesisches Restklassen Theorem

S = M/m1 * T1 * r1 + M/m2 * T2 * r2 + M/m3 * T3 * r3 + M/m4 * T4 * r4
S = 420/3 * 2 * r1 + 420/4 * 1 * r2 + 420/5 * 4 * r3 + 420/7 * 9 * r4
S = 280 * r1 + 105 * r2 + 336 * r3 + 540 * r4

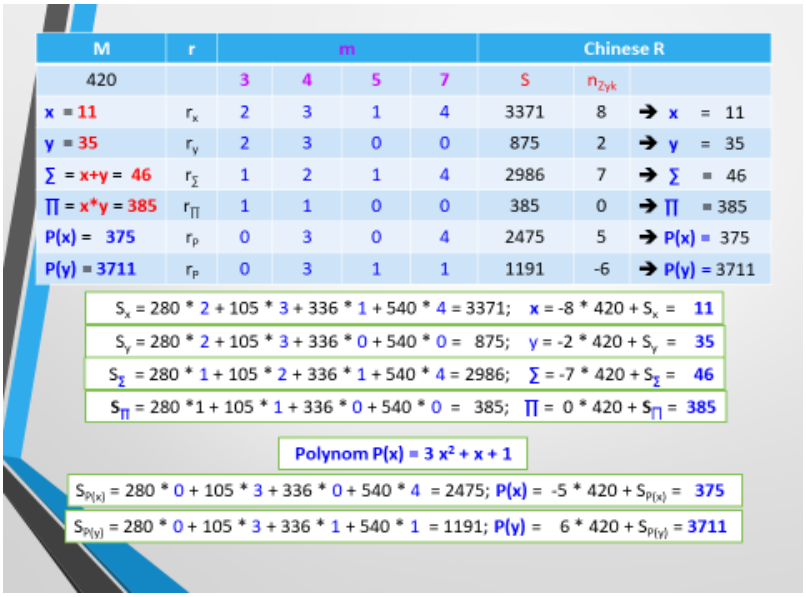
Inverse Elemente

(420/3 * T1) mod(3) == 1 ➔ T1 = 2
(420/4 * T2) mod(4) == 1 ➔ T2 = 1
(420/5 * T3) mod(5) == 1 ➔ T3 = 4
(420/7 * T4) mod(7) == 1 ➔ T3 = 9

➔ S = 280 * 2 + 105 * 1 + 336 * 3 + 540 * 4 = 3833;
➔ x = S mod(M) = 3833 mod(420) = 3833 - 9 * 420 = 53

Tab. 2: Chinesisches Restklassen Theorem für Restklassen {3, 4, 5, 7}, angewendet auf x = 53

Die Rücktransformation wird in **Tab. 3** auf die Werte von Tab. 1 angewendet.



M	r	m				Chinese R		
420		3	4	5	7	S	n_{Zyk}	
x = 11	r_x	2	3	1	4	3371	8	➔ x = 11
y = 35	r_y	2	3	0	0	875	2	➔ y = 35
∑ = x+y = 46	r_Σ	1	2	1	4	2986	7	➔ ∑ = 46
∏ = x*y = 385	r_Π	1	1	0	0	385	0	➔ ∏ = 385
P(x) = 375	r_P	0	3	0	4	2475	5	➔ P(x) = 375
P(y) = 3711	r_P	0	3	1	1	1191	-6	➔ P(y) = 3711

S_x = 280 * 2 + 105 * 3 + 336 * 1 + 540 * 4 = 3371; x = -8 * 420 + S_x = 11
S_y = 280 * 2 + 105 * 3 + 336 * 0 + 540 * 0 = 875; y = -2 * 420 + S_y = 35
S_Σ = 280 * 1 + 105 * 2 + 336 * 1 + 540 * 4 = 2986; ∑ = -7 * 420 + S_Σ = 46
S_Π = 280 *1 + 105 * 1 + 336 * 0 + 540 * 0 = 385; ∏ = 0 * 420 + S_Π = 385

Polynom P(x) = 3 x² + x + 1

$S_{P(x)}$ = 280 * 0 + 105 * 3 + 336 * 0 + 540 * 4 = 2475; P(x) = -5 * 420 + $S_{P(x)}$ = 375
$S_{P(y)}$ = 280 * 0 + 105 * 3 + 336 * 1 + 540 * 1 = 1191; P(y) = 6 * 420 + $S_{P(y)}$ = 3711

Tab. 3: Wiederherstellung der Werte aus Tab. 1. Die Grösse n_{Zyk} gibt die Anzahl der Perioden in M an

5.2 Vor- und Nachteile der Restklassen Mathematik

Die ganzzahligen Rechenoperationen sind eindeutig im Zahlenbereich zwischen 0 und M, im Beispiel zwischen 0 und 420. Fügt man eine weitere Klasse z.B. Modulo(11) hinzu, würde sich der Eindeutigkeitsbereich auf 420*11 = 4620 vergrössern, wobei die bisherigen Residuen r nicht geändert werden

müssten. Allerdings müsste man die Werte des chinesischen Restklassensatzes gemäss **Tab. 2** neu berechnen. Als weitere Vorteile sind zu nennen: die Rechenoperationen werden in jeder Restklasse unabhängig von den anderen ausgeführt. Es gibt nicht wie im Dezimalsystem einen Übertrag in die nächste höhere Klasse. Die Anforderung an die Rechengenauigkeiten sind deutlich geringer als im Dezimalsystem, denn man müsste im Beispiel im Zahlenraum von 0 bis 4520 innerhalb der 11-Restklasse nur auf etwa 10% genau rechnen, in den anderen Klassen dürfte man sogar noch ungenauer rechnen. Allerdings würde jedoch trotz der grossen Fehlertoleranz eine falsche Angabe um nur eine Stelle bereits einen sehr grossen Fehler bei der Rücktransformation ins Dezimalsystem bewirken.

	7	11	13	15	17	19
0	0	0	0	0	0	0
1 322 685	0	1	0	0	0	0
3 730 650	0	0	1	0	0	0

In der Tabelle wird durch die sechs Restklassen {7, 11, 13, 15, 17, 19} ein Eindeutigkeitsbereich von 0 bis 4 849 845 entsprechend 22 Bit aufgespannt. Ändert man bei der Zahl 0 nur einen der Residuenwerte um 1, so erhält man bei der Rekonstruktion mittels des chinesischen Restklassensatzes die sehr weit auseinander liegenden Werte 1 322 685 bzw. 3 730 650. Man kann zwar innerhalb einer Klasse auf nur 5% genau messen oder rechnen, aber nicht schlechter. Diese Tatsache ist oft zu beobachten, wenn mehrere zyklische Prozesse konkurrierend ablaufen. Aus der Laserphysik kommt dafür der Begriff *Mode Hopping* (siehe 6.1).

5.3 AUFBAU EINER DIGITALEN ENIGMA

Im Folgenden soll das Vorgehen für eine digitale Enigma mit K virtuellen Codescheiben für n Eingangszeichen beschrieben werden. Es werde ein ganze Zahl m gewählt, sodass $n = 2^m - 1$ die Grösse des Zeichenalphabets beschreibt. Bei der mechanischen Enigma war n = 26, bei der digitalen kann man mit m = 7 oder 8 einen Satz aus n = 127 oder 255 verschiedenen alphanumerischen Zeichen benutzen.

5.3.1 Algorithmus zur Codescheibenbelegung

Der paarweisen Verdrahtung der n Eingangskontakte an der Vorderseite einer Codescheibe $\mathbf{z}_{Eingang} = 1 \ldots n$ mit den n Ausgangskontakten an der Rückseite entspricht digital ein Permutationsvektor $\mathbf{z}_{Ausgang} = P(\mathbf{z}_{Eingang})$. Dieser Ausgangsvektor ist wiederum der Eingangsvektor für die nächstfolgende virtuelle Codescheibe.

- Der gesuchte Permutationsvektor der Länge n wird berechnet als P = S-Matrix (C_{Index}). [11]
- *Step 1*: Berechnung der zyklischen S-Matrix der Dimension n x n. Dazu wählt man einen *Seed* $\mathbf{x}_{Seed}$ der Länge m, der die Phase der S-Matrix einstellt. Er entspricht dem Voreinstellungswinkel der mechanischen Codescheibe und ist geheim zu übertragen gemäss 2.3.
- *Step 2:* Berechnung des Koeffizientenvektors der Länge m als $C_{Index} = 1 + (C + n_M) \bmod(n)$ mit $n_M = (n - 1)/2$. C ist der geheime Vektor der Länge m und muss gemäss 2.3 aus den von beiden Seiten übermittelten Zahlen decodiert werden. Beispiel für m = 4 (n = 15, $n_M = 7$): Für C = [3, 7, 12, 13] bekommt man C_{Index} = [11, 15, 5, 6].
- *Step 3:* Bei jeder der n Zeilen der binären S-Matrix wird an den m Spalten mit Index C_{Index} das Matrixelement ausgelesen. Die so erhaltenen m binären Zahlen ergeben den gesuchten Permutationsindex in dualer Schreibweise.

5.3.2 Testbeispiel für m = 5

Wir beschränken uns im Folgenden auf m = 5 und somit auf n = 31 Zeichen. Zur geheimen Übertragung des Seed Vektors x_{Seed} und der Koeffizienten C gemäss 2.3 wählen wir die öffentlichen Zahlen p = 12 311 und N = 276 411. Partei 1 wählt dann jeweils seine Geheimzahl A, Partei 2 entsprechend seine Geheimzahl B und beide Seiten decodieren daraus die Zahl C. Die virtuelle Test-Enigma enthalte sechs

[11] E.E. Fenimore "Large symmetric Pi transformations for Hadamard transforms"
Applied Optics, Vol. 22, No. 6 / 15 March 1983

Codescheiben CS1 bis CS6 und einen Reflektor, wobei die Codescheiben 2, 4 und 6 nach jeder Eingabe gemeinsam um einen Winkelschritt verdreht werden, nicht aber die Codescheiben 1, 2, 5 und der Reflektor.

Codescheibe 1		x_{Seed} = [1 0 0 1 0]
Koeffizienten		**Permutationsvektor P_{CS1}(1 .. 31)**
A	868 345 235 764 14	19 10 24 23 07 11 29 31 28 26 20 01 05 08 27 17 09 30 25 18 15 16 12 22 02 03 06 14 21 04 13 *
B	1868 122 235 899 764	
C	3 4 14 17 6	
Codescheibe 2		x_{Seed} = [1 0 0 1 0]
Koeffizienten		**Permutationsvektor P_{CS2} (1 .. 31)**
A	532 109 632 127 135	03 06 25 11 19 26 13 10 17 30 16 28 20 01 02 04 29 22 05 31 18 24 09 23 07 27 15 14 12 08 21 *
B	674 430 452 610 60	
C	1 15 25 17 18	
….		
….		
Codescheibe 6		x_{Seed} = [1 0 0 1 1]
Koeffizienten		**Permutationsvektor P_{CS6} (1 .. 31)**
A	547 399 416 181 256	30 27 20 21 25 10 14 13 31 23 07 17 26 24 06 29 09 28 05 15 01 12 19 04 03 18 08 16 22 11 02 *
B	164 666 895 517 703	
C	3 30 28 6 27	
Reflektor		
		Permutationsvektor P_{CS6} (1 .. 31)
	Daten sind manuell gewählt	26 20 29 28 18 16 27 19 17 25 24 30 23 21 22 06 09 05 08 02 14 15 13 11 10 01 07 04 03 12 **31** *

Da wir wegen des Erzeugungsalgorithmus in 5.3.1 für jedes ganzzahlige m immer eine ungerade Zahl $n = 2^m - 1$ bekommen, wird ein Kontakt im Reflektor nicht mit einem anderen verdrahtet, sondern führt das Zeichen denselben Pfad durch die Codierscheiben zurück, im Beispiel ist es Kontakt 31.

Aus der Sequenz CS1 ➔ CS2 ➔… ➔ CS6 ➔ Reflektor ➔ CS6 ➔…. ➔ CS1 ergibt sich für die erste Drehwinkeleinstellung $\varphi_1 = 0$ der Permutationsvektor $\mathbf{P_{Winkel\ 1}}$

Total, Drehwinkel $\varphi_1 = 0$	
	Permutationsvektor $P_{Winkel\ 1}$ (1 .. 31)
	27 *02* 15 29 17 11 10 23 24 07 06 14 21 12 03 22 05 20 30 18 13 16 08 09 31 28 **01** 26 04 19 25 *

Wie ersichtlich spiegelt sich der Reflektor in den Permutationswerten wider. So wird das erste Eingangszeichen in das 27ste Ausgangszeichen codiert und das 27ste Eingangszeichen in das erste Zeichen, und so weiter. Lediglich das zweite Eingangszeichen wird in sich zurückgeführt wegen des unverdrahteten Reflektorkontakts 31.

Nach erfolgter Eingabe eines Zeichens drehen sich Codescheiben 2, 4 und 6 um $d\varphi = 360/n$ und es ergibt sich der neue Permutationsvektor $\mathbf{P_{Winkel\ 2}}$.

Total, Drehwinkel $\varphi_2 = \varphi_1 + d\varphi$	
	Permutationsvektor $P_{Winkel\ 2}$ **(1 .. 31)**
	17 26 21 *04* 09 23 08 07 05 22
	20 29 31 24 25 27 **01** 30 28 11
	03 10 06 14 15 02 16 19 12 18 13 *

Die Paarbildung hat sich in gewünschter Weise verändert, da das erste Eingangszeichen nunmehr als 17. Zeichen erscheint und umgekehrt. Das singuläre nicht codierte Zeichen erscheint nunmehr als das Vierte.

Nach n = 31 Umdrehungen fasst man die so erhaltenen Permutationsvektoren $\mathbf{P_{Winkel\ 1}, \ldots, P_{Winkel\ 31}}$ in einer 31 x 31 Look-Up Tabelle zusammen. Damit ist die Einstellphase abgeschlossen und die eigentliche Textverschlüsselung kann beginnen.

6 ANHANG II

6.1 MODE HOPPING

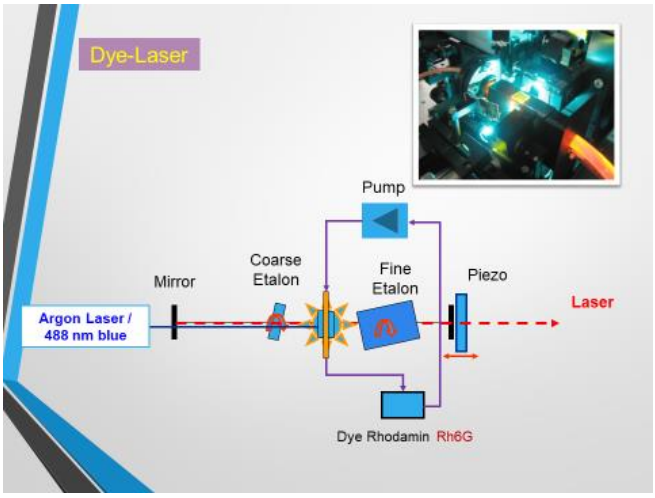


Abb. 10: Prinzipaufbau eines Farbstoff Lasers

Woher kommt der Ausdruck? Ein Farbstoff Laser (Dye Laser) besteht im Wesentlichen aus einer optischen Kavität, gebildet aus zwei Spiegeln **(Abb. 10**). Mittels einer Pumpe wird ein intensiv roter Farbstoff (Rhodamin Rh6G) als dünner Flüssigkeitsfilm quer durch die Kavität gespritzt. Mit einer externen Lichtquelle, einem blauen Argon Laser, wird der rote Farbvorhang zum Fluoreszieren im gelben Spektralbereich gebracht. In der Kavität befinden sich weiter zwei dünne Glasplättchen verschiedener Dicke (Etalons), die leicht verdreht werden können. Jedes dieser drei optischen Elemente definiert für sich stehende Lichtwellen, ähnlich einem Seil, das eingespannt ist zwischen zwei Fixpunkten. Dabei sind die Frequenzabstände der Oberschwingungen umgekehrt proportional zum Spiegelabstand, respektive der Glasdicke der Etalons. Es wird daher nur die Lasermode anschwingen, bei der die drei Frequenzkämme übereinstimmen (in **Abb. 11** die rote gestrichelte Linie). Verschiebt sich nun z.B. der Frequenzkamm der Kavität minim, kann die Resonanzbedingung bei einer weit entfernten Frequenz auftreten (in **Abb. 11** die violette Linie), was man als *Mode Hopping* bezeichnet. Will man bewusst den Laser auf einer anderen Wellenlänge emittieren lassen, muss die Einstellung aller drei Elemente, d.h. die Lage ihrer Frequenzkämme, nach den Regeln der Restklassenmathematik berechnet werden, was regeltechnisch nicht trivial ist.

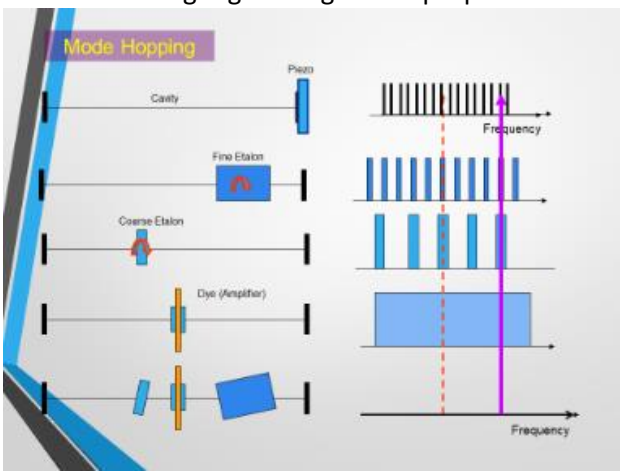


Abb. 11: Der Laser emittiert nur bei der Wellenlänge (Frequenz), bei der die drei Frequenzkämme übereinstimmen (Single Mode)

6.2 ADC / DAC

Ein Analog-zu-Digital Converter (ADC) und sein Gegenpart, ein Digital-zu-Analog Converter (DAC) sind wichtige Module der modernen Messtechnik. So sind die Signalamplituden von Sensoren, die Temperatur- oder Röntgenstrahlen messen, proportional zur Strahlungsintensität, und sie werden als Spannungswerte in Volt oder als Stromwerte in Ampère ausgegeben. Damit sie durch Computer weiter bearbeitet werden können, müssen die analog anfallenden Messwerte mit einer ADC Einheit digitalisiert und nach der Bearbeitung zur Anzeige wieder mit einer DAC Einheit analogisiert werden. Es gibt nun verschiedene Varianten, wobei wir im Folgenden uns auf die ADC beschränken.

6.2.1 ADC mit Residuen

Dazu wird als erstes in einer ADC der Eingangswert über eine bestimmte Zeitdauer konstant gehalten (*Sample and Hold*). **Beispiel**: im Messbereich Δx_{Range} von 0 bis 100 soll eine Grösse x mit der Genauigkeit von $\partial x = 0.1$ erfasst werden.

- Variante 1: man startet eine Referenzfunktion, deren Wert sich schrittweise um $\Delta x_{Step} = \partial x = 0.1$ erhöht und bildet laufend die Differenz zur Messgrösse x. Ändert diese ihr Vorzeichen, liest man die Anzahl gemachter Schritte n_{Step} aus und bekommt für x den Wert $x = n_{Step} * \Delta x_{Step}$. Diese direkte Methode wird oft in der Praxis für ADC-Module eingesetzt.
- Variante 2: Man kann den Prozess beschleunigen, wenn man die Schrittweite erhöht und beim Vorzeichenwechsel den Differenzwert x_{Rest} mit der Genauigkeit ∂x ermittelt: $x = n_{Step} * \Delta x_{Step} + x_{Rest}$.
- Variante 3: Da in unserem Beispiel der Messbereich Δx_{Range} in 1000 Schritte eingeteilt werden muss, könnte man die Variante 2 auch mit drei, voneinander unabhängigen Funktionen der Schrittweiten 7, 11 und 13 abtasten, denn 7 * 11 * 13 = 1001. Man ermittelt jeweils die Restwerte $x_{Rest}(7)$, $x_{Rest}(11)$ und $x_{Rest}(13)$ und bestimmt den gesuchten Wert x mittels des chinesischen Restklassentheorems. Der Vorteil ist, dass man die Schritte nicht mitzählen muss. So würde man für die Grösse x = 73.1 die Restwerte (3, 5, 3) in Schritten erhalten.

6.2.2 ADC von Bildmustern

Es gibt noch eine in der Praxis oft eingesetzte Variante 4: Sollte die Anzahl der Auflösungspunkte $N = \Delta x_{Range} / \partial x$ eine Zweierpotenz sein wie $2^{10} = 1024$, dann kann man in einem ersten Schritt abfragen, ob der unbekannte Spannungswert in der unteren (➔ 0) oder oberen Hälfte (➔ 1) des Messbereichs liegt, dann in welchem Viertel, Achtel und so weiter. Man bekommt somit das 10-Bit Codewort für den gesuchten Eingangswert, der als Dezimalzahl angezeigt wird.

Diese Methode kann man auch auf die Verschlüsselung von Symbolen, Signaturen, Graphiken, ja von ganzen Bildern anwenden. Als Beispiel soll eine Folge von chinesischen Zeichen gesichert übertragen werden. Entstammen diese einem Katalog aus $N = 2^K$ Zeichen, so ist jedes darin enthaltene Zeichen durch ein K-Bit Wort eindeutig identifizierbar. Der Sender muss nur dieses Bit Wort gesichert übermitteln, worauf der Empfänger im auch ihm bekannten Katalog das Zeichen findet. So bleibt die Frage, wie schnell und effizient kann man das zu übertragende Zeichen im Katalog finden und somit das ihm zugeordnete K-Bit Wort? Hierzu eignet sich bestens die Methode der *Principal Component Filter (PC)*, die der ADC-Variante 4 entspricht. Dabei sucht man K Filter, die, wenn angewandt auf ein Einzelzeichen des Katalogs, das dem Zeichen zugeordnete K-Bit Wort liefern, welches dann anschliessend verschlüsselt wird.

Die K PC-Filter werden wie folgt berechnet: man digitalisiert als erstes jedes der N chinesischen Zeichen in M Rasterpunkte und bekommt so ihre Helligkeitsfunktion $\boldsymbol{I}_{Zeichen\ (1\ldots M)}$. Die so erhaltenen N Helligkeitsfunktionen ordnet man als Zeilen in einer Matrix $\boldsymbol{O}_{Zeichen(M,N)}$ an. Die den N Zeichen zugeordneten Codeworte der Länge K werden in der Matrix $\boldsymbol{I}_{Code\ (N,K)}$ zusammengefasst.

Die Zeichenerkennung geschieht, indem man das Einzelzeichen mit den K Filtern multipliziert und so das ihm entsprechende Codewort erhält. Für das Kollektiv gilt dann

$$\boldsymbol{O}^{+}_{\text{Zeichen}}\,\boldsymbol{F}_{\text{Filter}} = \boldsymbol{I}_{\text{Code}} \qquad (1)$$

wobei die Matrix $\boldsymbol{O}^{+}_{\text{Zeichen}}(N,M)$ die zu $\boldsymbol{O}_{\text{Zeichen}}(M,N)$ transponierte Matrix ist. Da die Katalogzeichen $\boldsymbol{O}_{\text{Zeichen}}$ bekannt sind, ebenso wie die Codewortzuteilung $\boldsymbol{I}_{\text{Code}}$, kann man unter der *Bedingung* $M \geq N$ die Filterfunktionen $\boldsymbol{F}_{\text{Filter}}$ aus (1) berechnen als

$$\boldsymbol{F}_{\text{Filter}}(M,K) = \boldsymbol{O}_{\text{Zeichen}}\,(\boldsymbol{O}^{+}_{\text{Zeichen}}\,\boldsymbol{O}_{\text{Zeichen}})^{-1}\,\boldsymbol{I}_{\text{Code}}. \qquad (2)$$

Als Beispiel zeigen wir in **Abb. 12** einen Katalog aus 2^6 = 64 Kanji Zeichen, mit denen wir in einer früheren Studie das PC-Konzept entwickelt hatten. [12] Zur Berechnung der erforderlichen 6 PC-Filter wurde dem ersten Zeichen das 6-Bit Codewort [0,0,0,0,0,0] zugeteilt, dem 64. Zeichen das Codewort [1,1,1,1,1,1]. Die Zeichen selber wurden bei unserer Studie in 50 x 50 Rasterpunkten registriert, also mit M = 2500 digitalisiert. In dieser Publikation zeigten wir auch, dass die Zeichenerkennung nach (1) innerhalb von Nanosekunden erfolgen kann, wenn sie rein optisch durchgeführt wird. Dazu musste man zuerst aus dem Zeichensatz der Abb. 12 die Objektmatrix $\boldsymbol{O}_{\text{Zeichen}}$ bestimmen und mit ihr gemäss (2) anschliessend die 6 PC-Filterfunktionen $\boldsymbol{F}_{\text{Filter}}$ berechnen. Die Filterfunktionen wurden dann als *Computer Generated Hologram* (CGH) photographisch erzeugt und in die Optik eingesetzt. Kam nun eines der chinesischen Zeichen in das Gesichtsfeld dieses optischen Prozessors, so konnte man in der Sensorebene direkt das 6-Bit Codewort als Hell/Dunkel Signal auslesen. Dieser optische *Bild-Analog-zu-Digital Converter* wird besonders dann interessant, wenn die Kataloge sehr viele und komplizierte Zeichen beinhalten. Beim alljährlich von der japanischen Regierung durchgeführten *Kanji-Kentei Test* [13] müssen 4000 bis 5000 Kanji Zeichen erkannt werden: Mit der PC Methode bräuchte man dazu lediglich 12 PC Filter, die als CGH Kollektiv lithographisch auf einer dünnen Glasplatte aufgebracht und ins optische System eingebaut werden müssten. [14]

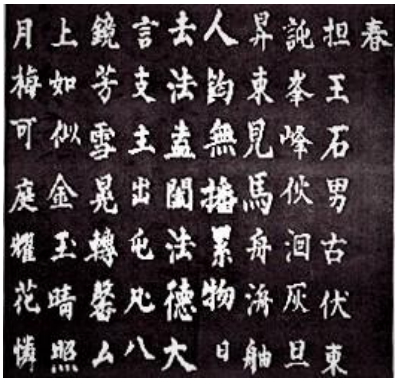


Abb. 12: Katalog aus 64 chinesischen Schriftzeichen

6.2.3 Schlussbemerkung

Für die eigentliche Verschlüsselung bleibt die Aufgabe im Wesentlichen die gleiche, ob man das 8-Bit Codewort eines Zeichens aus einem Alphabet von 256 alphanumerischen Zeichen überträgt, oder das 12-Bit Codewort aus einem Kollektiv von 4096 chinesischen Zeichen. Allerdings bräuchte es bei zu übertragenden Bildmustern schnelle ADC / DAC Vor- bzw. Nachprozesse.

Bernhard Braunecker studierte bis 1972 Kernphysik an der Universität Erlangen-Nürnberg, wechselte danach ins Gebiet der optischen Informationsverarbeitung und forschte an den Universitäten Erlangen und Essen und am IBM Research Lab in San José / USA. 1981 schloss er sich WILD Heerbrugg in der Schweiz an, eine der damals bedeutendsten Optikfirmen weltweit, die heute als Leica Geosystems zum schwedischen Konzern Hexagon gehört. Er leitete bis zu seiner Pensionierung 2006 die Forschungs-und Entwicklungsabteilung für Optik und mathematische Systemmodellierung. Er ist seit 2016 Mitglied des Bürgi Kernteams in Lichtensteig im schweizerischen Toggenburg, dem Geburtsort Bürgis. https://www.jostbuergi.com/

[12] Optical character recognition based on nonredundant correlation measurement,
B. Braunecker, R. Hauck, and A.W. Lohmann, Applied Optics / Vol.18, No.16 / 15 August 1979

[13] https://en.wikipedia.org/wiki/Kanji_kentei

[14] https://www.sps.ch/fileadmin/articles-pdf/2017/Mitteilungen_Bildsensoren.pdf