**Kryptoanalyse der Enigma-Maschine durch eine Software-Nachbildung der Turing-Welchman-Bombe**

# Turing-Welchman-Bombe

Emanuel Schäffer

26. November 2024

RWU–University of Applied Sciences
Prof. Dipl.-Math. Ekkehard Löhmann

# Inhaltsverzeichnis

KAPITEL 1

# Einleitung

Diese Projektarbeit beschäftigt sich mit der Kryptoanalyse der Enigma-Maschine. Im Speziellen wird hier auf die Kryptoanalyse mit der von Alan Turing und Gordon Welchman entwickelten „Turing-Welchman-Bombe“ eingegangen. Die Turing-Welchman-Bombe baut auf die Arbeit von Marian Rejewski und seiner „Bomba“ auf, welche der Turing-Welchman-Bombe ihren Namen gab. Dieses Verfahren zur Kryptoanalyse spielte eine wesentliche Rolle im Zweiten Weltkrieg und trug nach der Meinung vieler Historiker maßgeblich dazu bei, den Krieg zu verkürzen und rettete somit zahlreiche Menschenleben.

KAPITEL 2

# Die Enigma-Maschine

## 2.1 Einführung

Um die Funktionsweise der Turing-Welchman-Bombe und ihrer Software-Nachbildung zu verstehen, muss zuerst die Enigma-Maschine verstanden werden. Im Folgenden sei ein Überblick über die Enigma-Maschine gegeben. Die Enigma-Maschine ist eine Rotor-Chiffrier-maschine, die 1918 von Arthur Scherbius zum Patent angemeldet wurde und hauptsächlich im Zweiten Weltkrieg zum Einsatz kam. Aufgrund der Sicherheitsanforderungen der deutschen Wehrmacht wurde die kommerziell erwerbliche Enigma-Maschine modifiziert. In Abb. 2.1 ist eine solche modifizierte Enigma-Maschine zu sehen. Die von der Wehrmacht eingesetzten Enigma-Maschinen verfügten zunächst über drei Walzen (Ziffer 13). Neuerungen waren das Steckerbrett (siehe Abb. 2.1 front) und gegen Ende des Krieges eine zusätzliche, vierte Walze. Adaptiert wurde die Umkehrwalze (Ziffer 20). Hier wird, wenn nicht ausdrücklich erwähnt, ausschließlich die Enigma M3 mit drei Walzen betrachtet.

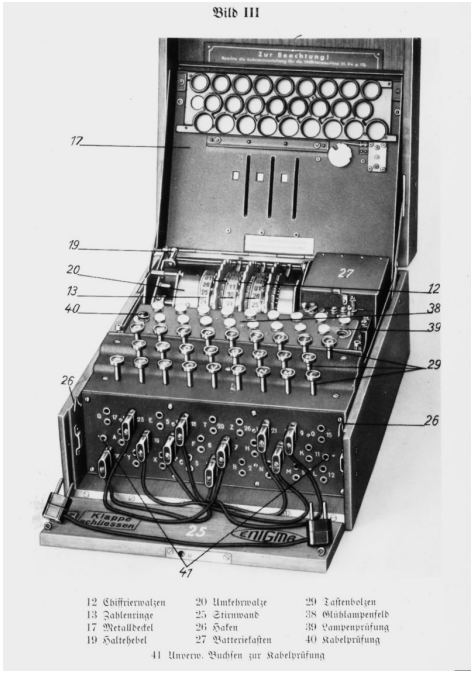



Abbildung 2.1: Die Enigma-Maschine (Foto Deutsches Museum München)

## 2.2 Funktionsweise

### 2.2.1 Walzen

Jede der von der Enigma-Maschine verwendeten Walzen besitzt eine interne Verdrahtung, welche eine monoalphabetische Substitution durchführt. Das bedeutet, dass jeder Buchstabe auf genau einen anderen Buchstaben abgebildet wird. In Abb. 2.2 ist die Verdrahtung der Walze I zu sehen.

Abbildung 2.2: Verdrahtung Walze I

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

Diese Verdrahtung ist starr und individuell für jede Walze. Eine Enigma-Walze hat 26 Eingangs- und 26 Ausgangs-Kontakte. Zudem kann eine Walze 26 Ausgangspositionen annehmen, jeweils repräsentativ für das Alphabet. Wird nun an den Eingangskontakt „A“ Spannung angelegt, so wird dieser Buchstabe durch die interne Verdrahtung zu einem „E“ permutiert.

Um die Enigma-Maschine in Betrieb zu nehmen, müssen drei von acht möglichen Walzen ausgewählt werden. Diese drei Walzen werden in Reihe geschaltet. Die rechte Walze wird bei jedem Tastendruck um eine Position weitergerückt. Hat diese Walze eine komplette Rotation vollzogen, rückt die Walze links neben ihr um eine Position weiter.[1] Die 26 Eingangs-Kontakte der rechten Walze werden durch 26 Kontakte der Enigma-Maschine versorgt, welche mit der Tastatur verbunden sind. Die Enigma-Maschinen Kontakte sind starr und bewegen sich nicht. Wird nun ein „A“ betätigt, während sich die Walze in Stellung „A“ befindet, wird diese zuerst rotiert und dann durchlaufen. Der Strom nimmt somit den „B“ Pfad. Das Resultat dieser in Reihe geschalteten Walzen ist eine polyalphabetische Substitution.

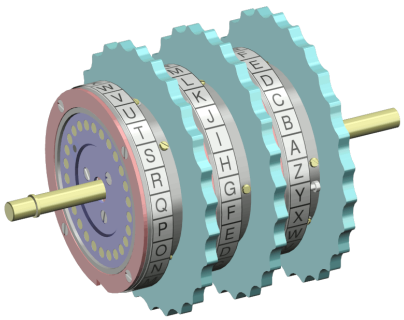


Abbildung 2.3: Enigma-Walzen[1]

Die Walzenstellung sagt aus, in welcher Ausgangsposition sich die Walzen befinden. Diese kann durch ein Sichtfenster vom Bediener abgelesen werden. Eine weitere Einstellmöglichkeit der Walzen ist die sogenannte Ringstellung. Sie verändert die Relation der sichtbaren Buchstaben zu der internen Verdrahtung und bewegt die Übertragskerbe, die festlegt, wann sich die Walze links von der aktuellen bewegt.

[1] Eine Analogie hierfür ist das Verhalten eines mechanischen Kilometerzählers oder das Verhalten von Sekunden-, Minuten- und Studenzeigern einer Uhr.

### 2.2.2 Umkehrwalze

Der originalen Patentschrift[2] von 1918 ist zu entnehmen, dass sehr frühe Enigma-Maschinen nicht involutorisch wirkten. Konkret bedeutet das für sehr frühe Enigma-Maschinen, dass diese zur Dechiffrierung einer Nachricht, die zuvor von einer Enigma-Maschine chiffriert wurde, einen speziellen Modus benötigen. Um die chiffrierte Nachricht zu dechiffrieren, muss ein Hebel umgelegt werden und die Walzen müssen in Ausgangsstellung gebracht werden. Nun wird Strom nicht von rechts nach links, sondern von links nach rechts durch die Walzen geleitet.

Da das daraus resultierende Gesamtgewicht von rund 50kg unakzeptabel für den Feldeinsatz war und der zusätzlich benötigte Mechanismus fehleranfällig erschien, wurde die Umkehrwalze oder auch der Reflektor von der Wehrmacht adaptiert. Die Umkehrwalze sorgt dafür, dass mit der gleichen Einstellung und „Modus“ ein beliebiger Text sowohl chiffriert als auch dechiffriert werden kann. Wie in Abb. 2.4 zu erkennen, besitzt die Umkehrwalze nicht 52, sondern nur 26 Kontakte. Sie „wirft“ das Signal zurück und schickt dieses ein weiteres Mal, in entgegengesetzter Richtung durch die Walzen. Es werden Buchstabenpaare gebildet, welche die Umkehrwalze kommutativ wirken lassen und die Enigma-Maschine involutorisch machen. Somit wurde sich die Komplikation und das Gewicht des Dechiffrierungs-Modus gespart.

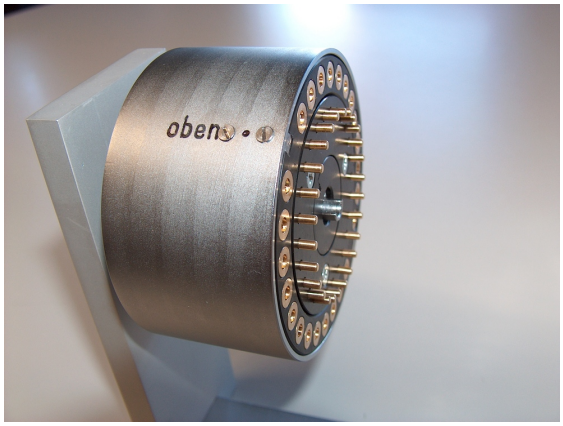



Abbildung 2.4: Umkehrwalze-D Replika[3]

Doch leider ist dieser geniale Einfall die wohl größte Sicherheitslücke der Enigma. Aufgrund der Buchstabenpaare wird niemals ein Buchstabe auf sich selber abgebildet. Das Ergebnis ist eine sogenannte fixpunktfreie Permutation. Kein Element behält somit seine Anfangsposition. Da $\forall x \in Alphabet,\ E_K(x) \neq x$ bleiben nur noch wenige Positionen übrig, an welchen sich ein bekannter Funkspruch-Abschnitt (Known-plaintext) befinden kann.

### 2.2.3 Steckerbrett

Da ein Schlüsselraum von 686.518.560 Möglichkeiten[2] der Wehrmacht nicht genügte, wurde der Enigma-Maschine das Steckerbrett hinzugefügt. Das Steckerbrett wirkt kommutativ und substituiert zwei Buchstaben. Es wird jeweils vor und nach dem Walzensatz traversiert. Die Anzahl von 150.738.274.937.250[4] zusätzlichen Möglichkeiten durch das

[2]Diese Berechnung berücksichtigt die Anomalie des Fortschaltmechanismus.[4]

Steckerbrett erscheint gewaltig, jedoch werden all diese Möglichkeiten von der Turing-Welchman-Bombe überwunden und spielen für die Sicherheit der Maschine de facto keine Rolle.

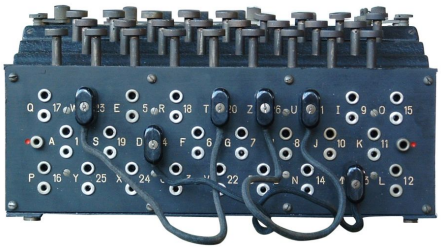


Abbildung 2.5: Enigma-Steckerbrett[5]

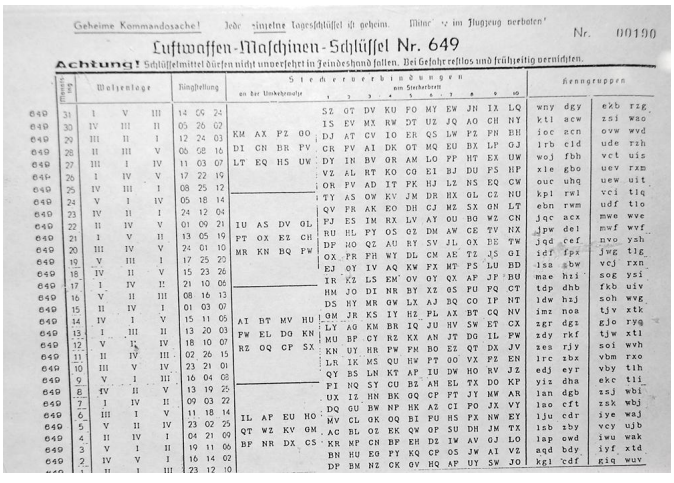
Die Permutation der Buchstaben lässt sich durch folgende Formel beschreiben:

**Permutation der Buchstaben**

$$E_K : A \longmapsto A, \quad E_K(x) := S \circ W_1^{-1} \circ W_2^{-1} \circ W_3^{-1} \circ U \circ W_3 \circ W_2 \circ W_1 \circ S(x)$$

## 2.3 Übertragung der Nachrichten

Bei der Enigma-Maschine wurde jeden Tag ein Tagesschlüssel eingestellt, welcher durch ein Code-Buch vorgegeben war. Dieser Tagesschlüssel bestand darin, welche drei der acht Walzen der Schlüssler in welcher Reihenfolge einzusetzen hatte, welche Ringstellung und welche Steckerbrett-Verbindungen für den Tag gültig waren. Von 13 möglichen Steckerbrett-Verbindungen wurden meistens 10 vorgegeben. In Abb. 2.6 ist ein Code-Buch-Auszug für eine Enigma-Maschine mit einer veränderlichen Umkehrwalze (UKW-D) zu sehen.



Geheime Kommandosache! Jede einzelne Tagesschlüssel ist geheim. Filmen im Flugzeug verboten!

Nr. 00190

Luftwaffen-Maschinen-Schlüssel Nr. 649

Achtung! Schlüsselmittel dürfen nicht unversehrt in Feindeshand fallen. Bei Gefahr restlos und frühzeitig vernichten.

| Tag | Walzenlage | Ringstellung | Steckerverbindungen an der Umkehrwalze | Steckerverbindungen am Steckerbrett | Kenngruppen |
|---|---|---|---|---|---|
| 649 | 31 I V III | 14 06 24 | KM AX PZ OO | SZ GT DV KU FO MY EW JH IX LQ | wny dgy ekb rzg |
| 649 | 30 IV III II | 05 26 02 | DI CN BR FV | IS EV MX RW DT UZ JQ AO CH NY | ktl acw zsi wzo |
| 649 | 29 III II I | 12 24 03 | LT EQ HS UW | DJ AT CV IO ER QS LW PZ FN BH | joc zcn ovw wvd |
| 649 | 28 II III V | 06 08 16 | | CR FV AI DK GT MQ EU BX LP GJ | lrb cld ude rzh |
| 649 | 27 III I IV | 11 03 07 | | DY IN BV OR AM LO FP HT EX UW | woj fbh vct uis |
| 649 | 26 I IV V | 17 22 19 | | VZ AL RT KO CO EI BJ DU FS HP | xle gbo uev rxm |
| 649 | 25 IV III I | 08 25 12 | | OR FV AD IT FK HJ LZ NS EQ CW | ouc uhq uew uit |
| 649 | 24 V I IV | 05 18 14 | | TY AS OW KV JM DR HX GL CZ NU | kpi rwl vci tlq |
| 649 | 23 IV II I | 24 12 04 | | QV FR AK KO DH GJ MZ SX GN LT | ebn rwn udf tlo |
| 649 | 22 II IV V | 01 09 21 | IU AS DV GL | PJ ES IM RX LV AY OU BO WZ CN | jqc acx mwe wve |
| 649 | 21 I V II | 13 05 19 | PT OX EZ CH | HU HL FY OS QZ DM AW CE TV NX | jpw del mwf wvf |
| 649 | 20 III IV V | 24 01 10 | MR KN BQ FW | DF HO QZ AU RY SV JL OX BE TW | jqd cef nvo ysh |
| 649 | 19 V III I | 17 25 23 | | OX FR FH WY DL CM AE TZ JS GI | idf fpx jwg tlg |
| 649 | 18 IV II V | 15 23 26 | | EJ QY IV AQ KW FX MT PS LU BD | isa gbw vcj rxn |
| 649 | 17 I IV II | 21 10 06 | | IR KZ LS EM OV OY QX AF JP BU | mae hzi sqg ysi |
| 649 | 16 V II III | 08 16 13 | | HM JO DI NR BY XZ QS FU FQ CT | tdp dhb fkb uiv |
| 649 | 15 II IV I | 01 03 07 | | DS RY MR OW LX AJ BQ CO IP NT | ldw hzj soh wvg |
| 649 | 14 IV I V | 15 11 05 | AI BT MV HU | QM JR KS IY HZ FL AX BT GQ NV | imz noa tjv xtk |
| 649 | 13 I III II | 13 20 03 | FW EL DQ KN | LY AQ KM BR IQ JU HV SW ET CX | zgr dgz gjo ryq |
| 649 | 12 V I IV | 18 10 07 | RZ OQ CP SX | MU BP CY RZ KX AN JT DG IL PW | zdy rkf tjw xtl |
| 649 | 11 II IV III | 02 26 15 | | KN UY HR FW FM BO EZ QT DX JV | zea rjy sol wvh |
| 649 | 10 III V IV | 23 21 01 | | LR IK MS QU HW PT OO VX PZ EN | irc zbx vbm rxo |
| 649 | 9 V I III | 16 04 08 | | QY BS LN KY AP IU DW HO RV JZ | edj eyr vby tih |
| 649 | 8 IV II V | 13 19 25 | | FI NQ SY GU BZ AH EL TX DO KP | yiz dha ekc tli |
| 649 | 7 I IV II | 09 03 22 | | UX IZ HN BK QQ CP FT JY NW AR | ian dgb zsj wbi |
| 649 | 6 III I V | 11 18 14 | IL AF EU HO | DQ GU BW NP HK AZ CI PO JX VY | iao cft zsk wbj |
| 649 | 5 V II IV | 23 02 25 | QT WZ KV GM | MV CL GK QQ BI FU HS FX NW EY | iju cdr jye waj |
| 649 | 4 II IV I | 04 21 09 | BF NR DX CS | AC BL OZ EK QW GP SU DH JM TX | iab zby vcy ujb |
| 649 | 3 V I II | 19 11 06 | | KN MP CH BP EH DZ IW AV GJ LO | iap owd iwu wak |
| 649 | 2 IV V I | 16 14 02 | | BN HU EQ FY XQ CP OS JW AI VZ | aqd bdy iyf xtd |
| 649 | 1 II I III | 23 12 10 | | DP BM NZ GK QV HQ AP UY SW JO | kgl cdf giq wuv |

Abbildung 2.6: Code-Buch-Auszug[6]

Nach 1938 musste der Schlüssler sich eine eigene Grundstellung der Walzen überlegen, mit welchem er den sogenannten Spruchschlüssel verschlüsselte. Nun überlegte sich der Schlüssler einen „zufälligen“ Spruchschlüssel, mit dem der Text chiffriert wurde.[3] Dieser Spruchschlüssel gibt die Walzenstellung für die folgende Nachricht an. Der „zufällige“ Spruchschlüssel wurde mit der Grundstellung verschlüsselt und ergab zusammen mit der Grundstellung und anderen Zusatzinformationen den „Spruchkopf“. Dieser Spruchkopf wurde dann im Klartext an den Empfänger übertragen. Da jedoch die Ringstellung die Relationen der sichtbaren Buchstaben zu der internen Verdrahtung ändert, war die Information der Grundstellung und des Spruchschlüssels für die Alliierten nicht wirklich brisant. Der Schlüssler gab den zu verschlüsselnden Text nach bestimmten Regeln ein[7]: Eigennamen wurden verdoppelt, Satzzeichen wie ein Punkt wurden durch ein X ersetzt, Uhrzeiten wurden ausgeschrieben und viele mehr.

Der Empfänger musste nun auf seiner Enigma-Maschine den Tagesschlüssel einstellen und die Walzen in die über den Spruchkopf mitgeteilte Grundstellung bringen. Nun entschlüsselte er mit dieser Einstellung den Spruchschlüssel. Wenn die Walzenstellung mitgeteilt durch den entschlüsselten Spruchschlüssel auf der Enigma-Maschine eingestellt wurde, konnte die eigentliche Nachricht dechiffriert werden. In Abschnitt 6.1 ist der Vorgang einer Nachrichtenübertragung zu sehen. Hierbei stellt $K_S$ den Spruchschlüssel und $K_T$ den Tageschlüssel dar.

Hierbei sei angemerkt, dass sich das Verfahren der Nachrichtenübertragung über die Kriegsjahre mehrfach änderte. So war die Grundstellung der Walzen vor 1938 noch Teil des Code-Buchs.

[3] In Wahrheit wählten die Schlüssler oft den gleichen Schlüssel, der meist persönliche Informationen wie zum Beispiel den Namen der Freundin enthielt.

KAPITEL 3

# Die Turing-Welchman-Bombe

## 3.1 Einführung

Da frühe Verfahren zur Kryptoanalyse der Enigma-Maschine, wie zum Beispiel der „Zyklometer“ oder die „Bomba“, durch die Einführung einer neuen Umkehrwalze (UKW-B), neuen Walzen und Änderung des Schlüsselverfahrens unbrauchbar gemacht wurden, musste ein neues Verfahren zur Kryptoanalyse der Enigma-Maschine von den Alliierten entwickelt werden. Der Durchbruch gelang, wie auch schon bei Marian Rejewski und seiner Bomba und Zyklometer, einem, beziehungsweise zwei Mathematikern. Alan Turing und Gordon Welchman waren die Hauptverantwortlichen für die Entwicklung der „Turing-Welchman-Bombe“. Dieses Verfahren basiert ähnlich wie der Zyklometer auf „Zyklen“. Jedoch wurde hier nicht die Verdopplung des Spruchschlüssels im Spruchkopf ausgenutzt, sondern Zyklen zwischen einem an einer bestimmten Stelle im Geheimtext vermuteten Klartext (Crib) und dem Geheimtext bestimmt. Die Turing-Welchman-Bombe testet immer eine Hypothese einer Steckerbrett-Verbindung und probierte mit drei von acht möglichen Walzen alle Walzenstellungen durch. Auch wenn diese Hypothese sich als nicht korrekt erwies, findet die Turing-Welchman-Bombe bei korrekter Walzenlage durch Reductio ad absurdum trotzdem die gültigen Steckerbrett-Verbindungen. Ziel war es die abgefangene Nachricht und ultimativ den Tagesschlüssel zu knacken. Aufgrund der Einfachheit wird im Folgenden der Begriff „Bombe“ anstelle von „Turing-Welchman-Bombe“ verwendet.

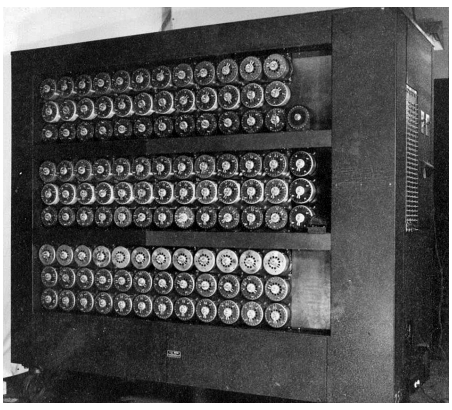


Abbildung 3.1: Eine Turing-Welchman-Bombe in Blechley Park[8]

## 3.2 Funktionsweise

### 3.2.1 Vorbereitungen

#### Crib

Hier wird der Anglizismus „Crib“ verwendet, da dieser Begriff keine richtige Deutsche Übersetzung hat.

Ein Crib ist ein Klartextfragment, welches an einer bestimmten Stelle im Geheimtext vermutet wird. Die deutsche Wehrmacht verwendete in den gesendeten Nachrichten häufig Floskeln. Ein Beispiel hierfür ist: „Das Oberkommando der Wehrmacht gibt bekannt“. Nun musste das Crib positioniert werden. Wie in Abschnitt 2.2.2 erklärt, ist es nicht möglich, dass ein Buchstabe auf sich selbst abgebildet wird. Wurde eine mögliche Position gefunden, konnten die Mitarbeiter von Blechley Park anfangen, das sogenannte Menü zu bauen. Die Abb. 3.2 zeigt solch eine Positionierung.

Abbildung 3.2: Positionierung des Cribs

| B | H | N | C | X | S | E | Q | K | O | B | I | I | O | D | W | F | B | T | Z | G | C | Y | E | H | Q | Q | J |
|---|---|---|---|---|---|---|---|---|---|---|---|---|---|---|---|---|---|---|---|---|---|---|---|---|---|---|---|
| O | B | E | R | K | O | M | M | A | N | D | O | D | E | R | W | E | H | R | M | A | C | H | T | | | | |
| | O | B | E | R | K | O | M | M | A | N | D | O | D | E | R | W | E | H | R | M | A | C | H | T | | | |
| | | O | B | E | R | K | O | M | M | A | N | D | O | D | E | R | W | E | H | R | M | A | C | H | T | | |
| | | | O | B | E | R | K | O | M | M | A | N | D | O | D | E | R | W | E | H | R | M | A | C | H | T | |
| | | | | O | B | E | R | K | O | M | M | A | N | D | O | D | E | R | W | E | H | R | M | A | C | H | T |

In Blechley Park waren zudem viele Sprachexperten beschäftigt, die darauf spezialisiert waren, solche Cribs zu erstellen. Dies erforderte sowohl sehr gute Kenntnisse über die Deutsche Sprache als auch sehr gute Kenntnisse über die in Abschnitt 2.3 beschriebenen Regeln zu Funkspruch-Verschlüsselung. Zudem mussten Eigenheiten und die „Schreibfäule“ der Funker beachtet werden. Ein Crib wurde unter der Vermutung benutzt, dass während der kompletten Eingabezeit des Abschnittes nur die rechte Walze (schnelle) Walze rotiert hat. Aus diesem Grund darf ein Crib nicht länger als 25 Buchstaben sein, da sonst gewiss ein Übertrag auf die mittlere Enigma-Walze stattfand. Die Bombe vernachlässigt also den Übertragzeitpunkt der Enigma-Walzen, wodurch die Ringstellung keine weitere Rolle spielt. Um das Risiko zu minimieren, dass ein Übertrag stattgefunden hat, wurden meist Cribs mit einer Länge von ungefähr 13 Buchstaben verwendet. In unserem Fall wäre also das Crib „OBERKOMMANDODERWEHRMACHT“ zu lang. Da das Crib jedoch keinen linguistischen Sinn ergeben muss, könnte man dies einfach zu „OBERKOMMANDODER“ kürzen.

Eine Strategie der Alliierten für die Erzeugung von Cribs war zum Beispiel das sogenannte „Gardening“. Damit ist das bewusste Provozieren von Funksprüchen, die einen bestimmten Klartext enthalten gemeint. Eine Strategie war es, Seeminen in Flüsse, Häfen oder Seegebiete abzuwerfen. Dafür musste ein Funker-Trupp in der Nähe des Ereignisses sein, welcher nicht verletzt werden durfte. Ein mögliches Ziel war hier zum Beispiel ein Seegebiet in der Nähe eines U-Boots, da diese stets eine Enigma-Maschine an Bord hatten. Nun beinhaltete der kurz darauf folgende Funkspruch mit einer hohen Sicherheit das Wort: „Minen“.

### Menü

Nun werden Buchstaben-Tupel zwischen Crib und Geheimtext gebildet. An einem Beispiel von „WETTERVORHERSAGE“ und „SNMKGGSTZZUGARLV“ wären solche Tupel: `W:S`, `E:N` et cetera. Nun werden die Tupel mit passenden Buchstaben aneinander gesetzt. In dem obigen Beispiel wären die Tupel unter anderem: `W:S` − `S:V` − `V:E`. Der daraus resultierende Graph wird „Menü“ genannt. Er gibt die Einstellungen der Bombe vor.

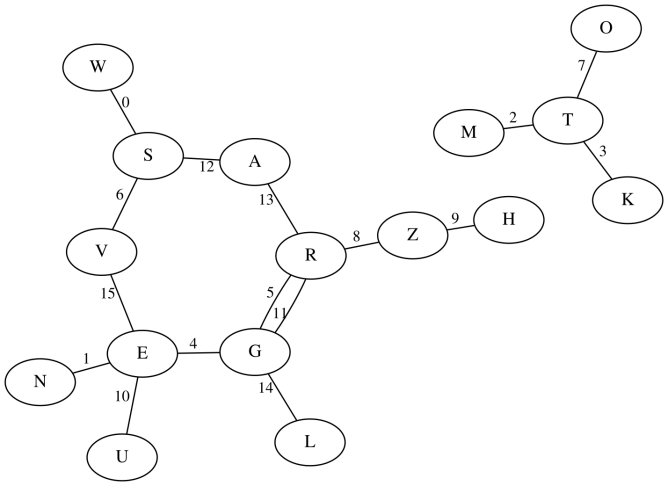



Abbildung 3.3: Crib-Geheimtext Menü

Wenn sich wie in Abb. 3.3 ein Zyklus bildet, ist dies eine brauchbare Crib-Geheimtext Kombination. Alan Turing machte die Beobachtung, dass die Steckerbrett-Verbindungen der Enigma-Maschine keinen Einfluss auf den Verlauf des Menüs haben. Dies ist durch die Eigenschaft des Steckerbrettes der Enigma-Maschine gegeben, welche eine monoalphabetische Substitution durchführt, die sich über den kompletten Chiffrierungs-Prozess nicht ändert.

Die Zahlen auf den Kanten geben hierbei die Position des Tupels innerhalb der gefundenen Crib-Geheimtext Kombination an. Tupel wie 0, 1, 8, 9, 10 und 14 werden hier als „Ausleger“ bezeichnet, da diese nicht aktiv zum Zyklus beitragen, aber trotzdem von Relevanz bei der Kryptoanalyse sind. Aus Gründen, die später erläutert werden, sind Tupel-Kombinationen wie 5 und 11 äußerst effektiv für die Kryptoanalyse. Tupel, die keinen Zyklus bilden, können ignoriert werden. Somit sind die Tupel 2, 3 und 7 für den weiteren Verlauf der Kryptoanalyse irrelevant.

### 3.2.2 Scrambler

Wie auch die Enigma-Maschine hat auch die Bombe „Walzen“. Jedoch haben diese nicht 52, sondern 104 Kontakte, da es erforderlich war, diese miteinander zu verbinden. Die Walzen der Bombe werden oft Scrambler genannt.

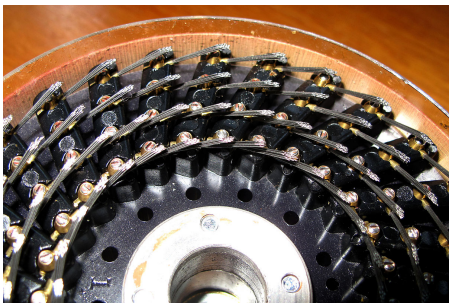


Abbildung 3.4: Die Kontakte der Scrambler[9]

Die meisten Bomben bestanden aus dreimal zwölf Scramblersätzen. Zwölf Scramblersätze ergeben eine „Chain“. Ein Scramblersatz bestand aus drei Scrambler und simulierte eine Enigma-Maschine. Der Grund, warum zwölf „Enigmas“ parallel verwendet werden ist, da somit eine komplette Ausgangsstellung der Walzen für das jeweilige Crib in einem Arbeitsgang überprüft werden kann. Der unterste Scrambler eines Scramblersatzes repräsentiert die rechte, schnelle Walze einer Enigma-Maschine. Der mittlere und oberste Scrambler ist repräsentativ für die mittlere und linke Walze der Enigma-Maschine. Da die Scrambler keine Übertragskerbe besitzen, bewegt sich der nächste Scrambler immer nach eine vollen Rotation des aktuellen Scramblers, unabhängig von der Startposition.

Wurde nun durch die Mitarbeiter von Blechley-Park ein passendes Menü bestimmt, konnten die Scrambler in ihre Ausgangsstellung gebracht werden. Hierfür muss in dem Zyklus eine „Route“ bestimmt werden. Für den Zyklus in Abb. 3.3 könnte die Route lauten: `W` $\rightarrow$ `S` $\rightarrow$ `A` $\rightarrow$ `R` $\rightarrow$ `G` $\rightarrow$ `E` $\rightarrow$ `V` $\rightarrow$ `S`.

Die Zahlen auf den Kanten werden als „Offsets“ für die untersten Walzen benutzt. In unserem Beispiel sind also die Ausgangsstellungen der Scrambler also `AAA`, `AAL` und so weiter. Sollen jetzt auch noch Ausleger oder andere Tupel-Kombinationen wie 5 und 11 eingebunden werden, so können diese einfach an passender Stelle eingefügt werden: `AAD, AAE, AAK`. Die Gesamtlänge der Elemente des Zyklus darf nicht zwölf überschreiten.

### 3.2.3 Terminal

Auf der Rückseite der Bombe befinden sich dreimal 26 Terminal-Kontakte. Ein 26er-Terminal-Satz ist jeweils einer Chain zugeordnet. Jeder Kontakt in einem Terminal-Satz repräsentiert jeweils einen Buchstaben im Alphabet. Jeder der 26 Kontakte hat 26 kleinere Kontakte, die wieder das Alphabet repräsentieren. Wird nun im $A$-Terminal an den $e$-Kontakt Spannung angelegt, so wird die Hypothese getestet, dass der Geheimtext mit der Steckerbrett-Verbindung $A \Leftrightarrow E$ verschlüsselt wurde.

### 3.2.4 In und Outs

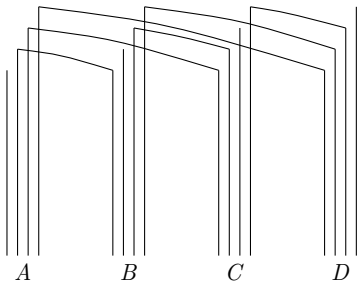
Auf der Rückseite befinden sich Kontakte, die mit „In“ und „Out“ gekennzeichnet sind. Wieder drei mal zwölf In- und Out-Paare, jeweils für die Scramblersätze. Nun werden die Terminals mit den In und Outs verbunden. Da der gewählte Routen-Startpunkt bei

„W“ liegt, wird der erste Scramblersatz auf das Offset `AAA` eingestellt. Darauf wird das Terminal $W$ mit dem ersten In-Kontakt verbunden. Der nächste Routen-Punkt „S“ wird durch einen „Brücken-Konnektor“ dem ersten Out mit dem zweiten In und dem Terminal $S$ verbunden.

### 3.2.5 Diagonalbrett

Gordon Welchman fiel auf, dass bei frühen Bomben nicht die Kommutativität des Steckerbretts beachtet wurde. Ist $A$ mit $B$ gesteckert, so muss auch $B$ mit $A$ gesteckert sein. Das Diagonalbrett stellte diese kommutativen Verbindungen her. Für die Bombe heißt dies, dass wenn im $A$-Terminal an den $b$-Kontakt Spannung angelegt wird, so auch der $a$-Kontakt im $B$-Terminal aktiv wird. Dies trug maßgeblich zur Effizienz der Bombe bei. Der Kontakt gleich dem Terminal-Buchstaben wird „Self-Steckered“ genannt und stellt keine Steckerbrett-Verbindung dar.

Abbildung 3.5: Diagonalbrett Verbindungen




### 3.2.6 Commons

Sollen nun auch noch Ausleger oder andere Graphen-Konstrukte miteinbezogen werden, reicht ein In- und Out-Kontakt nicht aus. Hierfür gibt es dreimal sechs Common-Kontaktblöcke à fünf Kontakte. Commons-Kontakte sind blockweise mit CO1, CO2 et cetera gekennzeichnet. Sechs Blöcke sind einer Chain zugeordnet. Die Kontakte eines Blocks sind miteinander verbunden. Somit ist es möglich, den Out-Kontakt des Scramblersatzes, der dem „E“ in unserem Zyklus entspricht, mit Terminal $V$ und $N$ und den jeweiligen Ins der Scramblersätze zu verbinden. In Abb. 3.6 sind die Terminals, Commons und In und Outs mit den „Brücken-Konnektoren“ zu sehen.

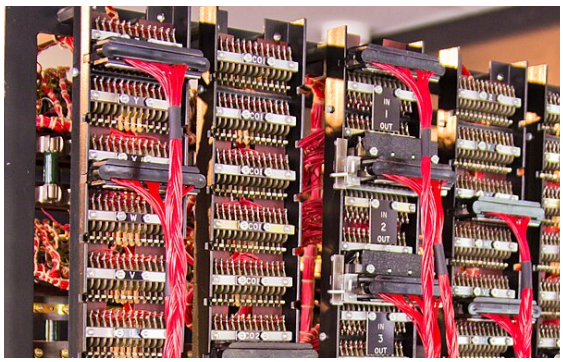
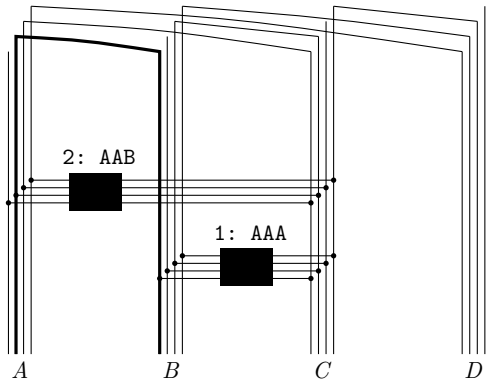


Abbildung 3.6: Rückansicht der Bombe[10]

### 3.2.7 Test-Register

Um eine Steckerbrett-Verbindungs Hypothese zu testen, muss ein Test-Register bestimmt werden. Dies sollte ein Buchstabe im Menü sein, der sehr viele Verbindungen hat. In dem Fall von Abb. 3.3 wäre der Buchstabe „G“ geeignet. Nun muss ein Test-Buchstabe bestimmt werden. Fällt die Wahl zum Beispiel auf $A$, so wird die Hypothese der Steckerbrett-Verbindung $A \Leftrightarrow G$ getestet. Es wird vermutet, dass während des Chiffrierungs-Prozesses die Steckerbrett-Verbindung $A \Leftrightarrow G$ benutzt wurde.

Abbildung 3.7: Diagonalbrett Verbindungen mit Scrambler Verbindungen




In Abb. 3.7 wird die Hypothese der Steckerbrett-Verbindung $A \Leftrightarrow B$ getestet. Unser Test-Register ist $A$. Es wurde ein Scramblersatz mit der Grundstellung `AAA` konfiguriert und Terminal $B$ mit dem ersten In und Terminal $C$ mit dem ersten Out verbunden. Ein weiterer Scramblersatz wurde auf die Grundstellung `AAB` konfiguriert und dem zweiten In mit dem ersten Out, welcher durch den Brücken-Konnektor auch mit Terminal $C$ verbunden ist, verbunden. Außerdem wurde das zweiten Out mit Terminal $A$ verbunden. Die Tupel in Abb. 3.7 sind `B:C` an erster und `C:A` an zweiter Stelle. Da das Steckerbrett keinen Einfluss auf den Verlauf des Zyklus hat, spielt es keine Rolle, welche Hypothese getestet wird. Nun wird $a$ durch den Scramblersatz permutiert. Ergibt sich durch die Permutation von $a$ ein anderer Buchstabe als $c$, müsste die Steckerbrett-Verbindung signalisiert durch den aktiven Kontakt während der Chiffrierung benutzt worden sein.

Die Bombe hält in zwei Fällen:

1. In dem Test-Register ist nach den Permutationen ein Kontakt aktiv – die Hypothese hat sich bewahrheitet.

2. In dem Test-Register sind nach den Permutationen 25 Kontakte aktiv – die Hypothese ist falsch, aber die nicht aktiven Kontakte in den Terminals geben die „richtige“ Steckerbrett-Verbindung an. (Reductio ad absurdum)

Die anderen Fälle werden von der Bombe ignoriert. Angenommen unser erster Scramblersatz permutiert das $a$ im $B$-Terminal zu einem $c$ und unser zweiter Scramblersatz permutiert dieses $c$ zu einem $d$ im $A$-Terminal, so erzeugt dies ein Widerspruch. Es wurde die Hypothese der Steckerbrett-Verbindung $A \Leftrightarrow B$ aufgestellt, aber damit der dritte Buchstabe im Zyklus bei der aktuellen Walzenlage einem $a$ entsprechen kann, muss zusätzlich $A \Leftrightarrow D$ herrschen. Dies ist eine widersprüchliche Aussage, da Buchstaben nur mit einem anderen verbunden sein können — die Walzen rotieren.

Die Tupel-Kombinationen 5 und 11 in Abb. 3.3 sind besonders effektiv, da sich die Anzahl der aktiven Verbindungen rasch „aufschaukelt“.

KAPITEL 4

# Implementierung der Turing-Welchman-Bombe

## 4.1 Menü Algorithmus

Um eine Turing-Welchman-Bombe in der Programmiersprache C nachzubilden, muss zuerst ein Algorithmus entworfen werden, welcher das Menü durch ein von dem Nutzer vorgegebenes Crib und Geheimtext bildet.

Hierfür werden die einzelne Buchstaben als Knoten mithilfe einer Struktur dargestellt, welche zum einen den Buchstaben und zum anderen einen Vektor mit den anliegenden Auslegern beinhaltet. Die Buchstaben-Tupel wurden ebenfalls als Struktur dargestellt, welche zwei Knoten, die Position im Crib und einen Booleschen Wert beinhaltet, welcher aussagt, ob dieses Tupel zum Zyklus beiträgt.

Quelltext 4.1: Realisierung der Menü Strukturen

```
typedef struct MenuNode MenuNode;
typedef struct CribCipherTuple CribCipherTuple;

struct MenuNode {
    CribCipherTuple *stubs;
    uint8_t num_stubs;
    char letter;
};

struct CribCipherTuple {
    MenuNode first;
    MenuNode second;
    uint8_t position;
    bool visited;
};
```

Die Tupel werden nun in einer „Nachschlagetabelle“ abgelegt. Diese Tabelle hat 26 Stellen, repräsentativ für das Alphabet. Ein jeweiliges Tupel wird sowohl unter dem ersten als auch unter dem zweiten Buchstaben abgelegt. Ein Tupel wie `W:S` ist also unter `W` als auch unter `S` abgelegt. Ein Tiefensuche-Algorithmus, der modifiziert wurde, um weniger „gierig“ zu agieren, schlägt die Tupel in der Tabelle nach und markiert die besuchten Tupel. Das Ergebnis ist eine gemessene, lineare Laufzeit.[1]

Das Menü wird als Vektor von `CribCipherTuple` in einer Struktur mit der Länge abgelegt. Da es erforderlich ist, eine eindeutige „Route“ durch das Menü anzugeben,

[1] Es wurden Crib-Längen bis 26 betrachtet, um zu garantieren, dass die lineare Laufzeit bei einer maximalen Länge von 13 Buchstaben gegeben ist. Der Laufzeit-Graph ist im Anhang enthalten: Abb. 6.5

werden Tupel-Kombination bei den betroffenen Knoten als Ausleger angegeben. Die Bombe kann somit im Falle der „Scramblersatz-Knappheit“ entscheiden, ob sie diesen (nicht notwendigen) Bestandteil des Menüs aufnehmen möchte.

## 4.2 Modellierung der Turing-Welchman-Bombe

### 4.2.1 Scrambler

Wie in Abschnitt 3.2.1 erklärt, vernachlässigt die Bombe den Übertragzeitpunkt der Walzen. Die Software-Implementation berücksichtigt ebenfalls die Übertragskerbe der Rotoren nicht. Die Verdrahtung der Scrambler wird als Vektor abgebildet, welcher wie in Abb. 2.2 die Permutation für jeden Buchstaben speichert. Die Buchstaben werden jedoch nicht in der typischen ASCII-Kodierung gespeichert, sondern auf den Wertebereich 0–25 abgebildet. Mit dem Resultat kann somit direkt der Vektor des nächsten Scramblers indiziert werden. Es muss zudem die Walzenlage mitgeführt werden, welche mit dem zu permutierenden Buchstaben addiert wird, um den Vektor-Index zu bilden. Der Wertebereich wird durch eine Modulo-Operation zwischen 0 und 25 gehalten.

### 4.2.2 Terminal

Die Kontakte der Terminals werden als Struktur dargestellt. Es beinhaltet einen 32 Bit Integer, welcher als Bitvektor mit 26 seiner Stellen die aktiven Kontakte repräsentiert. Die Anzahl der aktiven Kontakte wird ebenfalls als Integer mitgeführt. Da es in Bezug auf die Laufzeit medioker ist, alle 26 Stellen des Bitvektors traversieren zu müssen, um zu überprüfen, welche der 26 Kontakte aktiv sind, wurde ein normaler Vektor eingeführt, der die aktiven Kontakte „dicht“ speichert. Zuletzt wird der Buchstabe/die Nummer des jeweiligen Kontakts in der Struktur gespeichert.

Quelltext 4.2: Realisierung der Terminals

```
typedef struct {
    uint8_t active_cable_connections[ALPHABET_SIZE];
    uint32_t active_bit_vector;
    uint8_t num_active_connections;
    uint8_t contact_num;
} Contact;

typedef struct {
    Contact *contacts[ALPHABET_SIZE];
    Contact *test_register;
} Terminal;
```

Permutiert ein Scrambler einen Buchstaben, wird mithilfe des Bitvektors überprüft, ob dieser Kontakt bereits aktiv ist.

Hierbei gibt es zwei Szenarien:

1. Der Kontakt ist inaktiv: In dem Bitvektor wird die repräsentative Stelle aktiviert, der Buchstabe wird in dem normalen Integer Vektor abgelegt und die Anzahl der aktiven Verbindungen inkrementiert.

2. Der Kontakt ist aktiv: Dieser Buchstabe wird ignoriert.

Die Terminal Struktur beinhaltet einen Vektor mit 26 Kontakte und einen Zeiger auf den Kontakt des Test-Registers. Bei dem Test-Register wird hierbei nicht von einem „normalen“ Kontakt unterschieden.

### 4.2.3 In und Outs

Die In- und Out-Kontakte werden mit dem dazugehörigen Scramblersätzen als Knoten dargestellt. Sämtliche Verbindungen durch Kabel und Brücken-Konnektoren des analogen Originals werden hier durch Zeiger auf Kontakte ersetzt, so auch die In- und Out-Kontakte.

### 4.2.4 Diagonalbrett

Das Diagonalbrett als native Datenstruktur darzustellen, hat sich als ein eher schlechter Einfall herausgestellt. Eine Möglichkeit das Verhalten mit einer Datenstruktur zu imitieren, ist es, dem einzelnen Kontakte der Terminals als Vektor mit Zeiger auf Booleschen Werten abzubilden. Hierbei zeigt zum Beispiel der $a$-Kontakt des $B$-Terminals und der $b$-Kontakt des $A$-Terminals auf den gleichen Booleschen Wert. Somit ist die Verbindung $A \Leftrightarrow B$ abgebildet. Jedoch wäre es hier abermals nötig, den gesamten Vektor zu traversieren, um die aktiven Kontakte herauszufinden. Die entsprechenden Kontakte müssten nun wieder einen normalen Vektor besitzen, in dem die aktiven Kontakte „dicht“ stehen, damit die Scrambler effektiv arbeiten können. Somit wäre es hier abermals notwendig die Kontakte separat zu behandeln.

Aus diesem Grund wurde die Funktion des Diagonalbrettes in einer Funktion gekapselt:

Quelltext 4.3: Realisierung des Diagonalbrettes

```
static void activate_contact(TuringBombe *restrict turing_bombe,
                             const uint8_t first_contact,
                             const uint8_t second_contact)
{
    Contact *primary_contact = turing_bombe->terminal->contacts[
       first_contact];
    Contact *secondary_contact = turing_bombe->terminal->contacts[
       second_contact];

    primary_contact->active_cable_connections[primary_contact->
       num_active_connections] = second_contact;
    secondary_contact->active_cable_connections[secondary_contact
       ->num_active_connections] = first_contact;

    primary_contact->active_bit_vector |= (1 << second_contact);
    secondary_contact->active_bit_vector |= (1 << first_contact);

    primary_contact->num_active_connections++;
    secondary_contact->num_active_connections++;
}
```

Diese Funktion erhält drei Argumente: Einen Zeiger auf eine `TuringBombe` und zwei Integer, repräsentativ für die zwei Terminal-Buchstaben, die aktiviert werden sollen.

### 4.2.5 Commons

Aufgrund der Einfachheit wurden die Commons in dieser Software-Implementation „weg abstrahiert“. Diese wurden durch Zeiger auf den gleichen Kontakt ersetzt. Die Anzahl der Kontakte pro Commons und die Gesamtzahl der verwendeten Commons wurde gezählt, sodass diese nicht die Anzahl des analogen Originals überschreiten.

## 4.3 Algorithmus der Implementation

Im Gegensatz zur „echten“ Bombe, welche eine Steckerbrett-Hypothese für die jeweils 17576 Walzenstellungen einer Walzenlage testet, testet die Software-Nachbildung eine Hypothese für alle Walzenstellungen der 60 möglichen Walzenlagen. Eine modifizierte Breitensuche traversiert die Knoten des aufgespannten Graphen der „Scrambler-Knoten“. Da es aufgrund der Tupel-Kombinationen nicht genügt, einige Knoten nur einmal zu besuchen, und das Diagonalbrett es schwierig macht, die Datenstruktur der Knoten linear zur durchlaufen, wird erneut eine „Nachschlagetabelle“ benötigt. Die Nachschlagetabelle besitzt 26 Stellen, jeweils für 26 Terminal-Kontakte. Die Knoten, die durch den In-Kontakt mit einem jeweiligen Terminal verbunden sind, werden in dieser Tabelle abgelegt. Zuerst werden alle Scrambler-Knoten, die durch den In-Kontakt mit dem Test-Register verbunden sind, in die Warteschlange eingereiht. Danach werden die Commons der jeweiligen Knoten eingereiht und als letztes die Knoten, welche an jene Kontakte anliegen, die durch direkte Permutation der Scrambler oder durch das Diagonalbrett aktiviert wurden. Wird durch die Permutation eines Scramblers in dem mit ihn verbundenen Out-Kontakt kein neuer Kontakt aktiv, so werden die Knoten, welche durch den In-Kontakt mit dem jeweiligen Kontakt verbunden sind, nicht eingereiht. Ist die Warteschlange leer, ergo keine Permutationen mehr möglich, wird das Test-Register überprüft. Ergeben die Permutationen keine gültige Konfiguration, werden alle Kontakte die durch Permutation oder Diagonalbrett aktiviert wurden, deaktiviert. Die Software-Nachbildung hält nach den in Abschnitt 3.2.7 beschriebenen Regeln.

KAPITEL 5

# Geschwindigkeit der Software-Nachbildung im Kontext

KAPITEL 6

# Anhang

## 6.1 Übertragung der Nachrichten

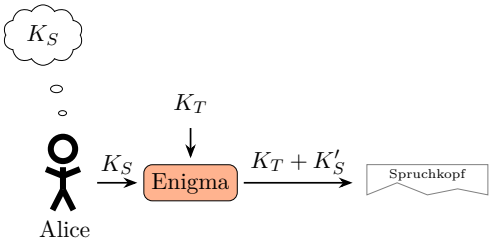



Abbildung 6.1: Erzeugung Spruchkopf

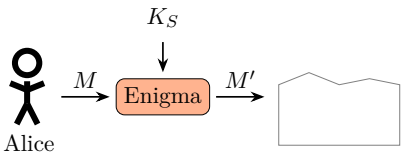



Abbildung 6.2: Chiffrierung Nachricht

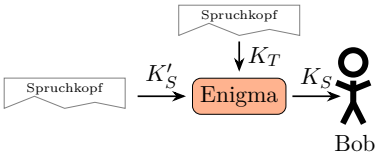



Abbildung 6.3: Dechiffrierung Spruchschlüssel

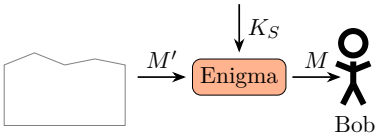



Abbildung 6.4: Dechiffrierung Nachricht

## 6.2 Laufzeit Menü Algorithmus

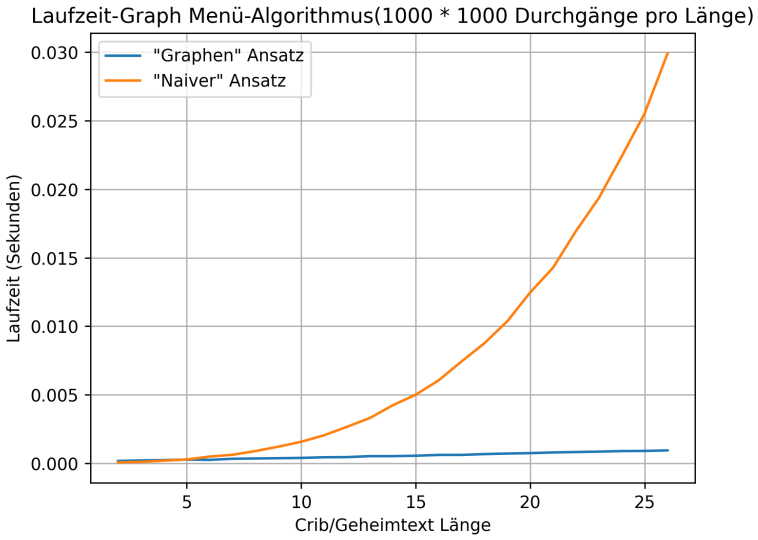



Abbildung 6.5: Laufzeit-Graph Menü-Algorithmus

Es wurden pro Länge 1000 Zufalls Crib und Geheimtexte generiert und diese jeweils mit 1000 Durchgängen getestet.

## 6.3 Glossar

- Blechley-Park: Zentrale militärische Dienststelle, die sich im Zweiten Weltkrieg erfolgreich mit der Entzifferung des deutschen Nachrichtenverkehrs befasste.
- Crib: Klartextfragment, welches an einer bestimmten Stelle im Geheimtext vermutet wurde.
- $D_K$: Die Dechiffrierfunktion.
- $E_K$: Die Chiffrierfunktion.
- Gardening: Das Provozieren von Funksprüchen, welche einen bestimmten Klartext enthalten.
- Menü: Ein Graph aus Crib-Geheimtext Tupeln, der die Einstellungen der Bombe vorgibt.
- Involution: Eine selbstinverse Abbildung, hier auf $dec = enc$ bezogen.
- Schlüssler: Person, die Nachrichten ver- oder entschlüsselt.
- Spruchkopf: Unverschlüsselter, erster Teil einer Nachricht, welcher die Uhrzeit, die Buchstabenanzahl, die Grundstellung und den verschlüsselten Spruchschlüssel beinhaltete.
- Spruchschlüssel: Individueller Schlüssel für einen Funkspruch.

KAPITEL 7

# Quellen

## 7.1 Einzelnachweise


[2] A. Scherbius, „Chiffriermaschine,“ dt. Pat. DE416219C1, Eingereicht: 23. Feb. 1918, Stand: 22. Okt. 2024, 8. Juni 1925. Adresse: https://www.cdvandt.org/Enigma%20DE416219C1.pdf.

[4] Wikipedia, *Enigma (Maschine) — Wikipedia, die freie Enzyklopädie*, [Online; Stand 23. Oktober 2024], 2024. Adresse: https://de.wikipedia.org/w/index.php?title=Enigma_(Maschine)&oldid=247875362.

[7] Oberkommando der Kriegsmarine, *Der Schlüssel M - Verfahren M Allgemein*, 1940. Adresse: https://www.cryptomuseum.com/crypto/enigma/files/schluessel_m.pdf (besucht am 25. 10. 2024).

[11] W. Ertel und E. Löhmann, *Angewandte Kryptographie*, 5. überarbeitete und erweiterte Auflage. Carl Hanser Verlag München, 2018, ISBN: 978-3-446-45468-2.

[12] Wikipedia, *Enigma-Schlüsselprozedur — Wikipedia, die freie Enzyklopädie*, [Online; Stand 14. November 2024], 2024. Adresse: https://de.wikipedia.org/w/index.php?title=Enigma-Schl%C3%BCsselprozedur&oldid=249763718.

[13] Graham Ellsbury, *The Turing Bomb*, 1998. Adresse: http://www.ellsbury.com/bombe1.htm (besucht am 25. 10. 2024).

[14] Virtual Colossus Project, Martin Gillow, *Bombe Technical Information*, 2016. Adresse: https://bombe.virtualcolossus.co.uk/technical.html (besucht am 26. 10. 2024).

[15] Entropia, andi, *Turingbomben*, 2014. Adresse: https://entropia.de/GPN14:Turingbomben (besucht am 26. 10. 2024).

[16] Wikipedia contributors, *Bombe — Wikipedia, The Free Encyclopedia*, https://en.wikipedia.org/w/index.php?title=Bombe&oldid=1236749954, [Online; Stand 26. Oktober 2024], 2024.

[17] Wikipedia, *Letchworth-Enigma — Wikipedia, die freie Enzyklopädie*, [Online; Stand 16. November 2024], 2024. Adresse: https://de.wikipedia.org/w/index.php?title=Letchworth-Enigma&oldid=240861245.

## 7.2 Bildnachweise


[1] Wikimedia Commons, *File:Enigma rotor set.png — Wikimedia Commons, the free media repository*, [Online; Stand 22. Oktober 2024], 2020. Adresse: https://commons.wikimedia.org/w/index.php?title=File:Enigma_rotor_set.png&oldid=509553237.

[3] Ravensburg-Weingarten University, [Online; Stand 22. Oktober 2024], 2011. Adresse: http://www.enigma.hs-weingarten.de/gallery.htm.

[5] Wikimedia Commons, *File:Macchina crittografica elettromeccanica - Museo scienza tecnologia Milano 08808 03.jpg — Wikimedia Commons, the free media repository*, [Online; Stand 24. Oktober 2024], 2023. Adresse: https://commons.wikimedia.org/w/index.php?title=File:Macchina_crittografica_elettromeccanica_-_Museo_scienza_tecnologia_Milano_08808_03.jpg&oldid=732612097.

[6] Wikimedia Commons, *File:Enigma keylist 3 rotor.jpg — Wikimedia Commons, the free media repository*, [Online; Stand 24. Oktober 2024], 2024. Adresse: https://commons.wikimedia.org/w/index.php?title=File:Enigma_keylist_3_rotor.jpg&oldid=864353172.

[8] Wikimedia Commons, *File:Wartime picture of a Bletchley Park Bombe.jpg — Wikimedia Commons, the free media repository*, [Online; Stand 27. Oktober 2024], 2024. Adresse: https://commons.wikimedia.org/w/index.php?title=File:Wartime_picture_of_a_Bletchley_Park_Bombe.jpg&oldid=853558909.

[9] Wikimedia Commons, *File:WireBrushesOnBombeDrum.jpg — Wikimedia Commons, the free media repository*, [Online; Stand 26 Oktober 2024], 2022. Adresse: https://commons.wikimedia.org/w/index.php?title=File:WireBrushesOnBombeDrum.jpg&oldid=704372169.

[10] Wikimedia Commons, *File:Bletchley Park Bombe8.jpg — Wikimedia Commons, the free media repository*, [Online; Stand 27. Oktober 2024], 2023. Adresse: https://commons.wikimedia.org/w/index.php?title=File:Bletchley_Park_Bombe8.jpg&oldid=825628616.