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

Turing-Bombe

Emanuel Schäffer

24. Oktober 2024

RWU–University of Applied Sciences

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-Bombe" eingegangen. Die "Turing-Bombe" baut auf die Arbeit von Marian Rejewski und seiner "Bomba" auf, welcher der "Turing-Bombe" ihren Namen gab. Dieses Verfahren zur Kryptoanalyse spielte eine wesentliche Rolle im 2.Weltkrieg und trug nach der Meinung vieler Historiker maßgeblich dazu bei, den Krieg zu verkürzen.

KAPITEL 2

Die Enigma

2.1 Einführung

Um die Funktionsweise der Turing-Bombe und ihrer Software-Nachbildung zu verstehen, muss zuerst die Enigma-Maschine verstanden werden. Nun sei ein Überblick über die Enigma-Maschine gegeben. Die Enigma ist eine Rotor-Chiffriermaschine, 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 modifiziert. In Abb. 2.1 ist eine solche modifizierte Enigma zusehen. Die von der Wehrmacht modifizierte Enigma verfügte zunächst über drei Walzen (Ziffer 13) und zusätzlich eine Umkehrwalze (Ziffer 20). Neuerungen waren das Steckerbrett (siehe Abb. 2.1 front) und gegen Ende des Krieges eine zusätzliche, vierte Walze. Ich werde mich, wenn nicht ausdrücklich erwähnt, ausschließlich mit der Enigma M3 mit drei Walzen beschränken.

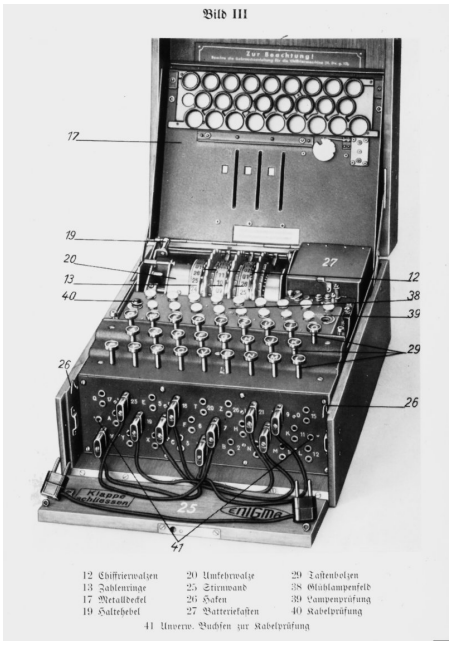

Abbildung 2.1: Die Enigma Maschine (Foto Deutsches Museum München)

2.2 Funktionsweise

2.2.1 Walzen

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

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 war starr und individuell für jede Walze. Eine Enigma-Walze hatte 26 Eingangs- und 26 Ausgangskontakte. Wurde nun an den Eingangskontakt “A” Spannung angelegt, so wurde dieser Buchstabe durch die interne Verdrahtung zu einem “E” permutiert.

Um die Enigma in Betrieb zu nehmen, müssen drei von acht möglichen Walzen ausgewählt werden. Diese drei Walzen wurden in Reihe geschaltet. Die rechte Walze wurde bei jedem Tastendruck um eine Position weitergerückt. Hat diese Walze eine komplette Rotation vollzogen, rückte die Walze links neben ihr um eine Position weiter[1]. Das Resultat dieser in Reihe geschalteten Rotoren ist eine polyalphabetische Substitution.

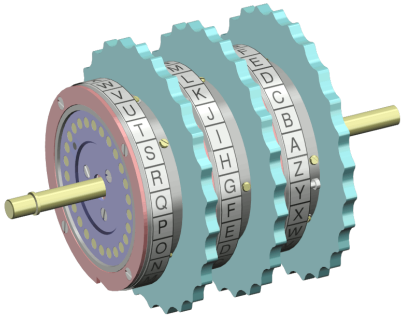

Abbildung 2.3: Enigma-Walzen[1]

Die Walzenstellung sagt aus in welcher Ausgangsposition die Walzen sich befinden. Diese kann durch ein Sichtfenster von dem Bediener abgelesen werden. Eine weitere Einstellmöglichkeit der Walzen ist die sogenannte Ringstellung. Sie veränderte die Relation der sichtbaren Buchstaben zu der internen Verdrahtung und bewegte die Übertragskerbe, die festlegte, wann sich die Walze links von der aktuellen bewegt. Da die Ringstellung bei dem Entschlüsselungsverfahren der Turing-Bombe keine weitere Rolle spielt, wird hier nicht weiter darauf eingegangen.

[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 es zu entnehmen, dass sehr frühe Enigmas nicht involutorisch wirkten. Das bedeutet ganz allgemein, dass $dec \neq enc$ ist. Es wird also zur Dechiffrierung eine andere Funktion als zur Chiffrierung benötigt. Konkret für sehr frühe Enigmas bedeutet dass, diese zur Dechiffrierung von einem mit der Enigma zuvor chiffrierten Textes einen speziellen Modus benötigte. Um die von einer Enigma chiffrierten Nachricht zu dechiffrieren musste ein Hebel umgelegt werden und die Walzen mussten in Ausgangsstellung gebracht werden. Nun wurde Strom nicht von rechts nach links sondern von links nach rechts durch die Rotoren geleitet.

Da dass 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 eingeführt. Die Umkehrwalze sorgt dafür, dass $dec = enc$ ist, sprich man mit der gleichen Einstellung einen beliebigen Text sowohl chiffrieren als auch dechiffrieren kann. Somit wurde sich die Komplikation und das Gewicht des dechiffrier-Modus gespart.

Abbildung 2.4: Enigma-Umkehrwalze[3]

Die Umkehrwalze “wirft” das Signal zurück und schickt dieses ein weiteres mal, in entgegengesetzter durch die Walzen. Doch leider ist dieser geniale Einfall die wohl größte Sicherheitslücke der Enigma. Aufgrund der Umkehrwalze wird niemals ein Buchstabe auf sich selber abgebildet. Aufgrund $A \not\rightarrow A$ bleiben nur noch sehr wenige Positionen übrig, an welchen sich ein bekannter Funkspruch-Abschnitt (Known-plaintext) befinden kann.

2.2.3 Steckerbrett

2.3 Übertragung der Nachrichten

Bei der Enigma wurde jeden Tag eine Tagesschlüssel eingestellt, welcher durch ein Code-Buch vorgegeben war. Dieser Tagesschlüssel bestand darin, welche drei der fünf Walzen der Schlüssler in welcher Reihenfolge einzusetzen hatte. Ferner musste er die Walzenstellung einstellen. Sie bestimmt die Ausgangsstellung der Walzen. Diese konnte durch ein Sichtfenster abgelesen werden.

Zusätzlich musste der Schlüssler die Ringstellung des Tages einstellen. Die Ringstellung veränderte die Relation der sichtbaren Buchstaben zu der internen Verdrahtung und bewegte die Übertragskerbe (siehe **??**), die festlegte, wann sich die Walze links von der aktuellen bewegt.

Wenn der Schlüssler nun eine Taste auf der Tastatur betätigte, floss Strom durch das Steckerbrett, welches eine Substitution durchführt. Danach floss der Strom durch den Walzensatz, in die Umkehrwalze und nochmals in entgegengesetzter Richtung durch den Walzensatz. Jeder dieser Walzen führte eine monoalphabetische Substitution durch. Nach dem der Strom den Walzensatz verlassen hatte, floss der Strom nochmals durch das Steckerbrett und erleuchtete letztendlich eine Lampe.

Als letzte Einstellung musste das Steckerbrett verdrahtet werden. Das Steckerbrett vertauschte zwei Buchstaben miteinander. Von 13 möglichen Steckerverbindungen wurden meistens 10 vorgegeben.

2.4 Übertragung der Nachrichten

Nachdem alle Einstellung getroffen waren, überlegte sich der Schlüssler einen „zufälligen“ Spruchschlüssel, mit dem der Text verschlüsselt wurde.[2] Dieser Spruchschlüssel gab die Walzenstellung für die folgende Nachricht an. Der „zufällige“ Spruchschlüssel wurde mit dem Tagesschlüssel verschlüsselt und ergab zusammen mit anderen Zusatzinformationen den „Spruchkopf“. Der Schlüssler gab nun den zu verschlüsselnden Text nach bestimmten Regeln ein. Eine Eigenschaft der Enigma ist, dass sie selbstinvers ist. Das bedeutet, dass mit der gleichen Einstellung, mit der ein Text verschlüsselt wurde, dieser auch wieder entschlüsselt werden kann.

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

KAPITEL 3

Die Turing-Bombe

3.1 Algorithmus Bombe

Algorithm 1 Bombe Algorithmus

procedure BOMBE($p_0 \dots p_{n-1}$: [Char], $c_0 \dots c_{n-1}$: [Char])
  for all *rotors* ∈ **permut(rotor order)**, *pos* ∈ [**AAA** … **ZZZ**] **do**
    plugs: Char → {Char}
    plugs(p_0) ∪ = {**'A'**}
    while *plugs* changing **do**
      for all $i \in$ [**0…n–1**] **do**
        plugs(c_i) ∪ = $\bigcup_{p \in \text{plugs}(p_i)}$ encrypt(*rotors*, *p*, *pos*+*i*)
        plugs(p_i) ∪ = $\bigcup_{p \in \text{plugs}(c_i)}$ encrypt(*rotors*, *p*, *pos*+*i*)
      end for
    end while
    if ∀ $S \in$ **cod(plugs)**: #S < #Char **then**
      report(*pos*, *plugs*)
    end if
  end for
end procedure

3.2 Cycle Finding Algorithm

```
typedef struct {
   char first;
   char second;
} Tuple;
```

KAPITEL 4

Häufigkeitsanalyse

4.1 Koinzidenzindex

Zunächst benötigen wir ein geeignetes Maß, um den Grad der Annäherung eines teilweise dechiffrierten Klartextes an authentisches Deutsch abzuschätzen. Dafür verwenden wir den Koinzidenzindex:

Koinzidenzindex

$$IC = \frac{\sum_{i=A}^{Z} f_i(f_i - 1)}{N(N-1)}$$

Sei f_i die Häufigkeit des Buchstabens in Abhängigkeit von i und N die Gesamtanzahl der Buchstaben. Wir Summieren also die Anzahl der Buchstabenpaare auf, und teilen diese durch die Anzahl der allgemein möglichen Buchstabenpaare. Der Koinzidenzindex ist also ein Maß für die Buchstabenverteilung. Für ein Text bestehend aus zufälligen Buchstaben beträgt der $IC \approx 0.038$; wobei er für Deutsch ≈ 0.076 beträgt.

Quellen

[1] W. Commons. „Enigma rotor set.“ File: Enigma-rotor-set.png. (7. Dez. 2004), Adresse: https://en.wikipedia.org/wiki/File:Enigma_rotor_set.png (besucht am 22. 10. 2024).

[2] A. Scherbius, „Chiffriermaschine,“ dt. Pat. DE416219C1, Accessed: 22 Oct 2024, 8. Juni 1925. Adresse: https://www.cdvandt.org/Enigma%20DE416219C1.pdf.

[3] W. Commons. „EnigmaReflektor.“ File: EnigmaReflector.jpg. (9. Feb. 2007), Adresse: https://commons.wikimedia.org/wiki/File:EnigmaReflector.jpg (besucht am 23. 10. 2024).