Christian-Albrechts-Universität zu Kiel
Institut für Informatik und Praktische Mathematik
Prof. Dr. W.-P. de Roever
Martin Steffen, Andreas Niemann, Melanie Zessack

Softwarepraktikum: Enigma

Sommersemester 2003 **Handout 10** 25. Mai 2003

Handout 10: Angriffe auf die Engima (1. Entwurf)

Ausgabetermin: 25. Mai 2003

Dieser Zettel enthält ein paar genauere Spezifikationen, was in der dritten Teilaufgabe des Semesters, der Implementierung der Angriffe auf die Enigma, abgeliefert werden soll. Anders als in den ersten beiden Teilen ist die Aufgabenstellung nun in vier *Alternativen* unterteilt, von denen jede Gruppe sich für einen entscheiden soll.

Die vier Angriffe sind, was den Aufwand betrifft, ungefähr gleich, doch liegen die Schwerpunkte jeweils etwas anders (Effizienz, Datenstrukturen, Visualisierung ...). Wir streben an, daß von jedem Angriff ungefähr gleich viele Implemetierung umgesetzt werden. Die Verteilung der Angriffe erfolgt nach dem Prinzip "wer zuerst kommt, mahlt zuerst".

Da aber die Umsetzung und auch die Visualierung eine relativ große Bandbreite erlauben, legen wir in diesem Abschnitt besonderen Wert darauf, daß der Umfang und der Entwurf nach *Rücksprache* mit den Betreuern angegangen wird.

In allen 4 Angriffen sind, neben der *Korrektheit* der Angriffe, auch die Darstellung auf dem Bildschirm und die Benutzerführung wichtig. Idealerweise sollten man als Benutzer durch die Interaktion mit der Oberfläche intuitiv verstehen können, was der Gedanke des Angriffes ist, Grundwissen über die Enigma natürlich vorrausgesetzt.

1 Streifenmethode

1.1 Allgemein

Wie in der Vorlesung geschildert, richtet sich die Streifenmethode —auch *méthode de bâton, cliques on the rods,* oder Isomorphieangriff— gegen eine Enigma *ohne Steckerbrett.* Sie nutzt die Achillesferse der Abbildung der Enigma —keine Fixpunkte, Selbstinversität— und die bauartbedingte Tatsache, daß sich der schnelldrehende Rotor und der Reflektor an den gegenüberliegenden Enden der Enigma befinden.

Es handelt sich um einen Angriff mit *bekannten Klartext.* Bekannt sind, wie bei allen Angriffen in diesem Praktikum, Auswahl und Verdrahtungen der Rotoren. *Ziel* des Angriffes in seiner *Grundversion* ist die Bestimmung der *Identität des schnellen Rotors und seine Kernposition.*

In der Grundversion des Angriffes wird angenommen, daß sich alleine der schnelldrehende Rotor bewegt und daß *kein Übertrag* auftritt. Demzufolge weist der Rest R' der Enigma, also

der Teil ohne die schnelle Walze R_0, eine feste Substitution realisiert, die die Charakteristika der Enigma-Abbildung aufweist:

$$p\ R_0\ R' = c\ R_0 \tag{1}$$

Die Gleichung besagt, daß $p\ R_0$ und $c\ R_0$ über eine feste, fixpunktfreie, selbstinverse Permutation zusammenhängen. Demzufolge muß man den gegeben Klartext p und den Schlüsseltext c bei verschiedene Anfangsstellungen für den schnellen Rotor verschlüsseln und in einem Ausschlußverfahren schauen, ob sich daraus Widersprüche ergeben.

Der Angriff kann verfeinert werden, wenn man berücksichtigt, daß bei der Verschlüsselung ein *Übertrag* auftreten kann. Wenn das passiert, zerfällt der Text (der Streifen) in zwei Teilen, deren Grenze man nicht kennt. Wenn man dies berücksichtigt, so treten weniger Widersprüche auf. Verletzungen der "Isomorphie" sind dann auf eine Weiterschalten von R' zurückzuführen.

Das Verfahren für die erste Walze ist unter anderem in [DK85, S. 101–104] geschildert.

Die Bestimmung der weiteren Walzen kann über *Tabellen* geschehen, die abgesucht werden. Eine modernere Darstellung würde die erste Phase auch für den Rest der Enigma anwenden bis hin zur Bestimmung der Umkehrwalze.

1.2 Umfang

- Es soll die *Eingabe* des Klartextes und des Schlüsseltextes interaktiv und von Datei aus möglich sein. Beim Lesen von Datei soll folgendes Format eingehalten werden. Der Text besteht aus 2 Zeile, die erste Zeile ist der Klartext, die zweite Zeile der Schlüsseltext.
- Die Texte sollen in beiden Fällen auf "Plausibilität" (Länge des Texte, Alphabet ...) getestet werden und bei Verstößen geeignet reagiert werden.
- Die einfache sowie die komplexere Lösung, die Übertrage berücksichtigt, soll implementiert werden. Ebenso sollen die restlichen Walzen bestimmt werden.
- Ein Schwerpunkt der Aufgabe ist die *Visualisierung*.
  - Es soll die verbleibenden Menge an konsistenten Positionen ausgegeben werden
  - Daneben sollen die *Widersprüche* geeignet dargestellt werden, eventuell erst auf Anfrage des Benutzers. Ebenso soll dargestellt werden, wo mögliche Überträge stattgefunden haben.
  - die Darstellung könnte sich an "Streifen" der Streifenmethode orientieren
  - Dies kann mit Hilfe eines sogenannte *Auswahlbaumes* dargestellt werden. Wenn eine Walze ausgeschlossen wird, was durch ein Blatt im Auswahlbaum gekennzeichnet ist, soll der Grund für den Ausschluß angezeigt werden.

2 Rejewski-Angriff

2.1 Allgemein

Der Rejewski-Angriff richtet sich gegen das *Indikatorsystem* in der einfachen Variante mit fester Grundstellung für die Übertragung der Spruchschlüssel. Was auf diesem Zettel als Rejewski-Angriff bezeichnet ist, zerfällt im Grunde in zwei verwandte Angriffe gegen das

Indikatorsystem. Beide Teile werten den gesamten Indikatorverkehr eines Tages aus und nutzen die feste Verbindung die zwischen den Buchstaben an erster und vierter, zweiter und fünfter, sowie dritter und sechster Stelle der Indikatoren aus, d.h., die Permutationen

$$\pi_0 = \varrho_0^{-1} \circ \varrho_3 \qquad \pi_1 = \varrho_1^{-1} \circ \varrho_4 \qquad \pi_2 = \varrho_2^{-1} \circ \varrho_5$$

genauer gesagt, deren Zyklenstruktur sowie mathematische Eigenschaften der Komposition bestimmter Permutationen (siehe vorlesung).

Die beiden Teile des Angriffes unterscheiden sich danach, ob die *Verdrahtung* der Rotoren bekannt sein muß; beide funktionieren auch bei vorhandenem Steckerbrett.

Der erste Angriff kommt ohne Kenntnis der Verdrahtung der Walzen aus, benutzt darüber hinaus aber Vorlieben der Verschlüssler nach bestimmten Spruchschlüsseln. Der zweite hingegen verwendet die Verdrahtung und bestimmt über die charakteristische Zyklenstruktur mögliche Walzenlagen und Kernpositionen.

2.2 Umfang

- Teil der Aufgabe ist die Erzeugung der *Indikatorverkehrs*. Da der Verkehr recht umfangreich ist, soll dies *automatisch generierbar* sein. Dabei sollen Möglichkeiten vorgesehen sein, Spezialschlüssel ("`AAA`", "`QWE`" oder Ähnliches) gehäuft generieren zu lassen oder per Hand gehäuft einzutragen, d.h., die Liste sollte modifizierbar sein.

  Die zur Generierung des Indikatorverkehrs notwendige Enigma soll konfigurierbar sein, d.h., die Grundstellung soll einstellbar sein.[1]

- Bei der Überprüfung, ob ein wahrscheinlicher Schlüssel mit dem Verkehr übereinstimmt, sollen die *Widersprüche* dargestellt werden

- die Zykelstruktur der π_i soll dargestellt werden

Zygalski-Roste

2.3 Allgemein

Der Angriff der Zygalski-Roste, auch als *Jeffrey-sheets* bezeichnet, richtet sich wie der Rejewski-Angriff gegen das Indikatorsystem, jedoch in der schwierigeren Variante bei der die Grundstellung vom Verschlüsseler frei gewählt wurden. Bekannt sind die Verdrahtung der Walzen, nicht jedoch der Zustand der Enigma. Ziel des Angriffs ist die Walzenlage und die Kernposition.

Der Angriff kann als eine Verallgemeinerung der Rejewski-Methode verstanden werden, die die charakteristische Zyklenstruktur eine Walzenlage und Kernposition der Enigma ausnutzte. Er verwendet nämlich die einzige Information die man bei *variabler Grundstellung* aus den Nachrichtenverkehr über die Zyklenstruktur noch bekommen kann, nämlich die *Existenz eine Fixpunktes.*. Doppelt-auftretende Buchstaben an erste und vierter Stelle des verschlüsselten Spruchschlüssels[2] werden Weibchen oder *females* genannt, genauer gesagt 1-4 females; analog gibt es noch 2-5 und 3-6 females. In Unkenntnis der Ringstellung, gibt die in den ersten drei Zeichen übertragene Grundstellung die Kernposition nicht preis und insbesondere kann man

[1] Das war ja ohnehin im zweiten Teil gefordert.

[2] also an vierter und siebter Position des gesamten, neunbuchstabigen Indikators.

nicht ablesen, welches die Position ist, die gegebenenfalls einem Female entspricht. Da sich jedoch die zur Entschlüsselung interessante Kernposition von der veröffentlichten Grundstellung nur die relative *und im gesamten Verkehr feste* Ringstellung unterscheidet, kann man durch „Herausrechnen“ der Grundstellung zumindest ein *Muster* an aufgetretenen Females bestimmen.

Durch Abgleich des aus dem Indikatorverkehrs festgestellten female-Muster mit dem kompletten Katalog, in dem für alle Positionen vermerkt ist, ob das Alphabet einen Fixpunkt besitzt oder nicht, d.h., ob eine female möglich ist oder nicht, kann man, bei genügend großer Anzahl von Weibchens, die Position meist eindeutig feststellen, indem man nur alle Möglichkeiten durchprobiert.

Die Zygalski-Roste stellen eine „Lochblattimplementierung“ dieses Gedankens dar, bei der der Mustervergleich durch Übernanderlegen und Verschieben der Blätter, also durch eine „Parallelisierung”, manuell praktikabel wird.

Die einfache Variante, wie hier beschrieben und wie auch historisch angewandt, ignoriert die Tatsache, daß Überträge stattfinden.

Material findet sich über die Code&Ciphers-Seite [Cod03]

2.4 Umfang

- man implementiere beide Varianten: das Original und das Komplexere, welches die Übertragspunkte berücksichtigt.
- Die Visualisierung sollte die Originalhandhabung der Lochkarten deutlich machen. D.h., es bietet sich eine (quasi) Vervierfachung der $26 * 26$ Matrix pro Blatt an. Die Alternative, wie sie beispielsweise von der Code&ciphers-Seite gewählt wurde, macht das Verschieben nicht sehr klar.
- Man soll Varianz und Mittelwert der Löcheranzahl der Blätter ausrechnen und darstellen.

3 Turing-Welchmann Angriff

3.1 Allgemein

Die Methode von Turing und Welchman, implementiert in der *Britischen Bombe* mit dem Diagnoalbrett von Welchman, ist ein Klartextangriff der das Indikatorsystem ignoriert[3] und sich direkt gegen den Rumpf der Nachricht richtet. Der Angriff ist eine Methode “roher Gewalt”, insofern er Walzenlagen- und Positionen durchspielt um zu testen, ob sie den gegebenen Klartext in den ebenfalls gegebenen Schlüsseltext überführt. Der entscheidende Punkt und der Beitrag von Turing und Welchman dabei ist zu vermeiden, auch alle Konfigurationen des *Steckerbretts* durchzutesten; die Anzahl der Kombinationen wäre untragbar hoch.

Hier der entscheidende Gedanke: sei eine Walzenlage und -stellung gegeben, von der bei unbekanntem Steckerbrett überprüft werden soll, ob sie mit dem gegebenen paar aus Klartext und Chiffretext vereinbar ist. Anstelle nun der Reihe nach die astronomische Anzahl von Steckverbindung ebenso durchzuspielen, geht man schlauer vor: man nutzt aus, daß die

[3] Die Polnische Bombe, die mit der britischen außer dem Namen und der Tatsache der Mechanisierung des Entschlüsselns nicht viel gemein hat, richtete sich gegen die Indikatoren.

hypothetische Annahme einer Steckerverbindung i.d.R. *Konsequenzen* für andere Buchstabenpaare hat. Da die Steckverbindungen für die Nachricht fest ist und da man für jeden Paar von Klartext/Chiffretext*buchstaben* die Konfiguration der Walzen fixiert hat —man kennt das Fortschalten der Enigma— können sich zunächst *unmittelbar* Widersprüche zur Annahme der zu testenden Steckerverbindung ergeben. Dies allein wäre wenig hilfreich, da zu selten. Neben den direkten Widersprüchen, und das ist der wichtige Punkt, ergeben sich meist als *Folgerung Festlegungen* für andere Steckerpaare, die ihrerseits zu Widersprüchen oder weiteren Konsequenzen führen, die dann wiederum . . . .

Die Eigenschaft der Enigma, die es erlaubt, diese Konsequenzen zu ziehen, ist wieder einmal die Selbstinversität bzw. der "reflektierende" Aufbau der Maschine; insbesondere wird das selbe Steckerbrett symmetrisch sowohl vor der Rotorverschlüsselung also auch danach verwendet. Das Diagnonalbrett nutzt zusätzlich die Selbstinversität des Permutation des Steckerbretts aus, d.h., die Tatsache, daß man die Konsequenzen der Steckerbretts auch "rückwärts" verwenden kann. Dies erlaubt eine weitere Reduktion des Aufwands, da eine gewählte Steckverbindung mehr Konsequenzen besitzt und es somit falsche Steckverbindungen eher ausgesiebt, bzw., daß man mit kürzeren und einfacheren Texten auskommt.[4]

Eine sehr klare Darstellung zu diesem Verfahren findet sich in [Hod83, Kapitel 4].[5]

3.2 Umfang

Da es sich, was die Walzen betrifft, im wesentlichen um ein Verfahren mit roher Gewalt ("brute force"-Angriff) ohne große inhärente algorithmische Eleganz handelt, legen wir auf folgende Punkte besonderes Gewicht:

- Es soll eine *Einzeltest* und ein *Volltest* implementiert werden. Bei dem Einzeltest wird eine gegebene Konfiguration (außer dem Steckerbrett, natürlich) bei gegebenem Klar- und Schlüsseltextpaar auf *Konsistenz* getestet. Das Ergebnis des Tests soll ähnlich wie in [Hod83] dargestellt werden, z.B. sollte bei einer inkonsistenten Konfiguration sichtbar sein, was die Ursache des Widerspruches ist. Der eigentliche Angriff spielt dann Walzenlagen und Walzenpositionen durch. Dabei soll die *Anzahl* der Steckerverbidungen berücksichtigt werden.
- Effizienz: da bei dem vollen Angriff mehr oder minder Enigma-Automaten „durchgenudelt" werden, sollte der Automat der Gruppen, die diesen Angriff wählen, einigermaßen flott implementiert sein.
- Konfigurierung und Einschränkung des Suchraumes: damit der Algorithmus auch mit geringerer Laufzeit getestet werden kann, muß die Enigma in einer Form konfigurierbar sein, daß man bestimmte Teil als *fest* vorgibt, sodaß der Suchraum eingeschränkt ist.
  - Walzenvorrat: 3, 4, oder alle 5.
  - Eintrittswalze: A oder T fixiert, oder beide möglich
  - Fixierung des Reflektors, oder beide möglich.

[4] Was hier logisch als Prüfen auf Konsistenz und "Schlußfolgerungen ziehen" beschrieben ist, wurde in der realen Bombe elektro-galvanisch gelöst, in dem man eine Reihe von modifizierten Enigmas parallelschaltete.

[5] Hodges geht kaum auf die elektromechanische Umsetzung der Bombe ein; dies ist für uns auch nicht so wichtig. Wen es dennoch interessiert, wird auch der Seite [Bom03] fündig. Auch die Darstellung auf [Cod03] ist lesenswert.

Literatur

[Bom03] The reconstruction of the British Bombe. `http://www.jharper.demon.co.uk/bombe1.htm`, 2003.

[Cod03] The Codes & Ciphers homepage. `http://www.codesandciphers.org.uk/`, 2003.

[DK85] Cipher A. Deavours and Louis Kruh. *Machine Cryptography and Modern Cryptanalysis*. IPF. Artech House Books, 1985.

[Hod83] Andrew Hodges. *Alan Turing: The Enigma of Intelligence*. Unwin Paperbacks, 1983.