



Falko Dressler Tobias Limmer

## Netzwerksicherheit WS 2007/08

### 2. Übungsblatt

13.11.2007



Diese Übung gliedert sich in zwei Teile: Im ersten Teil wird eine vereinfachte Version einer Rotor-Schlüsselmaschine vom Typ Enigma programmiert. Im zweiten Teil wird diese Version erweitert, um einen Brute-Force-Angriff auf einen verschlüsselten Text durchzuführen.

1. Unsere vereinfachte Enigma besteht aus 3 fest montierten Rotoren mit jeweils 26 Vertauschungen. Die Rotoren und der Reflektor haben eine feste Belegung (die folgenden Angaben befinden sich auch in Textform auf der Webseite der Vorlesung).

- Rotor I

ABCDEFGHIJKLMN OPQRSTUVWXYZ  
BDFHJLCPRTXVZNYEIWGAKMUSQO

- Rotor II

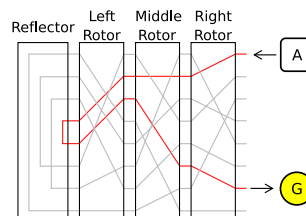
ABCDEFGHIJKLMN OPQRSTUVWXYZ  
AJDKSIRUXBLHWTCQGZNPYFVOE

- Rotor III

ABCDEFGHIJKLMN OPQRSTUVWXYZ  
EKMFLGDQVZNTOWYHXUSPAIBRCJ

- Reflektor

(AY) (BR) (CU) (DH) (EQ) (FS) (GL) (IP) (JX) (KN) (MO) (TZ) (VW)



Die drei Rotoren werden wie bei einem Uhrwerk weiterschaltet: Hat der erste Rotor 26 Bewegungen hinter sich, so wird der Zweite um eine Position weiterschaltet. Der Dritte wird wiederum nach 26 Bewegungen des zweiten Rotors bewegt. Dabei können alle Rotoren eine beliebige vorgegebene Anfangsposition einnehmen.

Der Weg eines Signals ist wie folgt: Bei Drücken einer Taste wird es durch die drei Rotoren geleitet, danach vom Reflektor wieder durch die Rotoren in umgekehrter Reihenfolge zur Lampe zurückgeschickt. Dabei ist der Reflektor wie ein starrer Rotor aufgebaut und reflektiert das Signal zurück an die Rotoren. Der Ablauf ist also wie folgt:

Taste → Rotor I → Rotor II → Rotor III → Reflektor → Rotor III → Rotor II →  
Rotor I → Lampe

Es ist zu beachten, dass ein Rotor vor dem Reflektor von „links nach rechts“ passiert wird und nach dem Reflektor dieser invertiert benutzt wird, also von „rechts nach links“. Ein Beispiel mit den vorgegebenen Rotoren in Anfangsstellung:

C-F→F-I→I-V→(VW)→W-N→N-T→T-J

**Aufgabenstellung:** Programmieren Sie eine vereinfachte Enigma mit den obigen Angaben. Es muss zu Beginn der Chiffrierung der Stand der Rotoren eingegeben werden können. Ein Stand „A“ bedeutet, dass der Rotor sich im Ausgangszustand befindet. Ein Stand „B“ bedeutet, dass der Rotor um eines gedreht wurde. Testen Sie Ihre Implementierung, indem Sie Texte verschlüsseln und entschlüsseln. Dokumentieren Sie diese Vorgänge in der mit abzugebenden Textdatei.

2. Erweitern Sie das in der vorigen Aufgabe erstellte Programm zur Simulation einer Enigma um die Fähigkeit, einen *probable Plaintext*-Angriff auf einen verschlüsselten Text durchzuführen. Dazu wird ein im Originaltext vermutetes Wort (ein sogenannter „Crib“) vorgegeben und dann mit Hilfe der Brute-Force-Methode versucht, den restlichen Text zu entschlüsseln.

Sie können von folgenden Annahmen ausgehen:

- Sie kennen den Aufbau der Rotoren (siehe vorige Aufgabe)
- Sie wissen, dass das Uboot von Wasserbomben angegriffen wurde und diese in verschlüsselten Nachrichten oft mit WABOS abgekürzt wurden.

Mit dem bekannten Wort können Sie die Stellung der bekannten Rotoren herausfinden und den Rest des Textes entschlüsseln.

- (a) In einem ersten Schritt kennen Sie die Anordnung der Rotoren. Somit müssen Sie nur die Stellung der einzelnen Rotoren durch die Brute-Force-Methode herausfinden.
- (b) Nun ist die Anordnung der Rotoren (I, II, III) frei wählbar. Ihr Angriff muss somit alle Möglichkeiten überprüfen. Entschlüsseln Sie einen selbst gewählten Beispieltext, welcher das Wort WABOS enthält.
- (c) Die Enigma bildet keinen Buchstaben des Klartextes auf den identischen Buchstaben im verschlüsselten Text ab. Kann mit diesem Hinweis die Entschlüsselung beschleunigt werden? Warum?

#### Technische Hinweise:

- Die Aufgaben sind in C/C++ oder Java zu erstellen.
- Der Code muss auf dem Betriebssystem Debian/Linux im CIP-Pool der Informatik mit der dort installierten Version des GNU C/C++- oder Java-Compilers kompilieren und lauffähig sein.
- Es ist ein Makefile oder ein ausführbares Skript zu bereitzustellen, welches temporäre Daten löschen (Target „clean“) und eine Binärdatei (Target „all“) erstellen kann.
- Es ist eine Dokumentation im ASCII-Format zu erstellen, welche einen typischen Programmstart dokumentiert und über Besonderheiten des Programms berichtet.

Abgabe der Lösungen per Mail an [tobias.limmer@informatik.uni-erlangen.de](mailto:tobias.limmer@informatik.uni-erlangen.de) bis Montag, den 19.11.2007 als Archiv im Format `.tar.(gz|bz2)` oder `.zip`.