ulm university universität
uulm

Institut für Angewandte Informationsverarbeitung
Dr. Andreas F. Borchert
Stefan Lenz

15. November 2012
Blatt 4

Systemnahe Software I (WS 2012/2013)

Abgabe bis zum 22. November. 2012, 16:00 Uhr

Lernziele:

- Zeichen und Zeichenketten (Characters und Strings)
- Zeigerarithmetik

In diesem Blatt wollen wir die Vigenère-Verschlüsselung betrachten. Es handelt sich dabei um eine Erweiterung der Cäsar-Verschlüsselung. Bei der Cäsar-Verschlüsselung handelt es sich um eine Verschiebung jedes Zeichens im Originaltext (=Klartext) um eine Anzahl von Buchstaben im Alphabet, um den Geheimtext zu erhalten. Die Vigenère-Verschlüsselung benutzt nicht die gleiche Verschiebung für alle Buchstaben, sondern verschiebt abhängig von deren Position im Klartextstrang. Die Verschiebung wird, um sie besser im Gedächtnis zu behalten, in Buchstaben codiert:

A	B	C	D	E	F	G	H	I	J	K	L	M
0	1	2	3	4	5	6	7	8	9	10	11	12

N	O	P	Q	R	S	T	U	V	W	X	Y	Z
13	14	15	16	17	18	19	20	21	22	23	24	25

Wir wollen hier ausschließlich mit Großbuchstaben umgehen und auch alle Leerzeichen aus dem Text entfernen, um möglichst wenig Rückschlüsse auf den Originaltext zu geben. Beispiel mit Schlüssel „SCHLUESSEL“:

Klartext:	`VERRATEESKEINEM`
Schlüssel:	`SCHLUESSELSCHLU`
Chiffre:	`NGYCUXWWWVWKUPG`

Aufgabe 7: Ver- und Entschlüsseln (5 Punkte)

Ihr sollt zwei Funktionen schreiben, die einen übergebenen Text mit einem gegebenen Schlüssel Vigenère-ver- bzw. entschlüsseln können. Die beiden Funktionen können/sollen den

Text direkt bearbeiten (d. h. es muss nichts kopiert werden). Die Signaturen der beiden Funktionen sollen wie folgt aussehen:

```
void encrypt(char* text, const char* key);
void decrypt(char* text, const char* key);
```

Testet eure Funktionen in der `main`-Methode: Lest den Schlüssel und dann zeichenweise den zu verschlüsselnden Text (`getchar`) von der Standardeingabe in einen 5-KB-Character-Puffer (=Array). Alle Zeichen außer Großbuchstaben sollen dabei ignoriert werden. Hört auf, wenn der Puffer voll ist oder das Eingabeende (`EOF`) erreicht ist.

```
submit ss1 7 vigenere.c team
```

Aufgabe 8: Automatisches Entschlüsseln (5 Punkte)

Um die Vigenère-Verschlüsselung zu knacken, kann man wie folgt vorgehen:

a) Führe eine Häufigkeitsanalyse der Buchstaben im Text für jede mögliche Schlüssellänge durch:

   Dies soll die mit der folgenden Funktion umgesetzt werden:

   ```
   void freq_analysis(double p[], const char* t,
                      int strlen_t, int shift, int offset);
   ```

   Das Array `p` kann in der Funktion als ausreichend dimensioniert angenommen werden und soll mit den relativen Häufigkeiten der Buchstaben von A bis Z gefüllt werden, die in dem Text auftauchen, der entsteht, wenn man bei `offset` im Text `t` anfängt und jeden `shift`-ten Buchstaben - von `offset` ab gezählt - nimmt.

   (3 Punkte)

b) Berechne die *Shannon-Entropie* für die Häufigkeitsverteilung. Diese Größe ist ein Maß für die mittlere Informationsdichte eines Textes. Sie berechnet sich durch

$$-\sum_{i=1}^{26} p_i * log_2 p_i.$$

   Falls $p_i = 0$, wird $p_i * log_2 p_i$ mit dem Grenzwert 0 definiert. Signatur der gesuchten Funktion:

   ```
   double shannon_entropy(double p[]);
   ```

   (Es ist für das Beispiel ausreichend, die Entropie nur für den Teiltext mit `offset == 0` zu berechnen.) Die Shannon-Entropie eines Textes in deutscher Sprache ist im Mittel 4,08. Werte die Schlüssellänge als gefunden, wenn die Abweichung davon weniger als 0,1 beträgt oder die Entropie einen geringeren Wert hat.

   Hinweis: Um den Logarithmus zu berechnen, benötigt man die Bibliothek für mathematische Funktionen. Um sie zu verwenden, muss man nicht nur `math.h` einbinden, sondern die Bibliothek auch linken. Dies muss man dem `gcc` mit der zusätzlichen Option `-lm` (hinter dem Namen der Quelldatei!) mitteilen.

   (1 Punkt)

c) Um den Text schließlich zu entschlüsseln, muss man noch die Häufigkeitsanalyse mit verschiedenen Offsets durchführen, um daraus den Schlüssel zu berechnen. Man kann davon ausgehen, dass der häufigste auftretende Buchstabe jeweils dem E entspricht.

   (1 Punkt)

Bei der Lösung dieser Aufgabe kann die Aufgabe 7 natürlich verwendet werden. Das gesuchte Programm soll den `geheim.txt`, der auf der Vorlesungshomepage zu finden ist, von der Standardeingabe lesen und knacken können.

```
submit ss1 8 crack_vigenere.c team
```

Viel Erfolg!