Dr. Johannes Mayer
Axel Blumenstock
Christian Ehrhardt
Abteilung Angewandte Informationsverarbeitung

Allgemeine Informatik III / Systemnahe Software I (WS 2005/2006)

Abgabetermin: 22. November 2005

Versteckte Botschaften (10 Punkte)

Wer vermag schon zu sagen, wie sicher ein Verschlüsselungsverfahren ist – nicht nur aktuell, sondern in aller Zukunft? Da bietet sich an, eine (verschlüsselte) Botschaft so zu übertragen, dass ein potentieller Angreifer überhaupt nicht auf die Idee kommt, dass es da etwas zum Angreifen gäbe.

Dies ist das Gebiet der *Steganographie* (etwa: „verborgenes Schreiben"). Zu den klassischen Beispielen zählen die Zitronentinte, die erst durch Erwärmen des Blattes wieder sichtbar wird, oder (in der Antike) das Tätowieren einer Nachricht auf den glattrasierten Schädel eines Sklaven, der, sobald sein Haar nachgewachsen war, zum Empfänger geschickt werden konnte.

Glücklicherweise stehen uns heute einfachere Methoden zur Verfügung. Hier wollen wir nun einen (unverschlüsselten) Text in ein Bild einbetten, so dass man ihm die verborgene Botschaft nicht ansieht, und hernach wieder extrahieren.

Als Bildformat wählen wir dieses Mal *portable graymaps* (pgm), die mit den bereits bekannten *portable bitmaps* eng verwandt sind. Eine pgm-Datei (im Rohformat) beginnt mit der Kennung „P5", es folgen (als Dezimalzahlen, jeweils durch Whitespace getrennt) die Breite w und Höhe h in Pixeln, sowie der maximale Grauwert d eines Pixels. Wir können hier davon ausgehen, dass d maximal 255 beträgt. Abgesehen davon, dass Sie diesen Wert natürlich wieder in den Header des Ausgabebildes schreiben müssen, können Sie ihn hier ignorieren.

Nach einem einzelnen, weiteren Whitespace-Zeichen (newline) folgen $w \times h$ Zeichen, wobei jedes Zeichen für ein Pixel steht und einen Wert zwischen 0 und d annehmen kann. 0 bedeutet „schwarz", d „weiß", und alles dazwischen sind die entsprechenden Graustufen.

Eine gültige pgm-Datei könnte also wie folgt beginnen:

```
P5
400 400
255
òñïíÜ#-ÀÅÃÀÀÂÁÀ¿¿¿ ...
```

Wenn wir nun in jedem Pixel (bzw. Bildbyte) das jeweils niedrigstwertige Bit durch ein Bit unserer Nachricht ersetzen, ändern wir den Grauwert von (statistisch) der Hälfte der Pixel um $1 / (d+1)$, bei $d = 255$ also um 0.39 % – dies ist vom menschlichen Auge praktisch nicht wahrnehmbar, bzw. es geht im normalen Bildrauschen unter. Tabelle 1 illustriert die Vorgehensweise.

Bytefolge im Ursprungsbild	Zeichen 'Z' = 0x5a	Bytefolge im Ausgabebild
0x57 = 0101 011**1**	**0**	0101 011**0** = 0x56
0x9f = 1001 111**1**	**1**	1001 111**1** = 0x9f
0x02 = 0000 001**0**	**0**	0000 001**0** = 0x02
0xaa = 1010 101**0**	**1**	1010 101**1** = 0xab
0x4d = 0100 110**1**	**1**	0100 110**1** = 0x4d
0xe2 = 1110 001**0**	**0**	1110 001**0** = 0xe2
0x30 = 0011 000**0**	**1**	0011 000**1** = 0x31
0x91 = 1001 000**1**	**0**	1001 000**0** = 0x90

Tabelle 1: Das Zeichen 'Z' = 0x5a = 01011010 wird in den jeweils niedrigsten Bits von acht Bildbytes untergebracht.

(a) Schreiben Sie ein Programm `stegano-write`, welches ein pgm-Bild an der Standardeingabe entgegennimmt und ein solches über die Standardausgabe ausgibt. Die Botschaft soll als Zeichenkettenkonstante im Quelltext des Programms festgelegt werden (siehe unten bzw. im Beispielprogramm). Arbeiten Sie diese Botschaft von links nach rechts ab, in jedem Zeichen beginnend mit dem höchstwertigen Bit. Diese Bitfolge schreiben Sie in das jeweils niedrigstwertige Bit der Bildbytefolge, beginnend mit dem ersten.

Das letzte Zeichen, das Sie so einbetten, soll das Nullzeichen (`'\0'`) sein, damit Sie beim Auslesen das Ende der Botschaft wieder erkennen können. Alle nachfolgenden Bildbytes reichen Sie unverändert durch. Ist die Botschaft so lang, dass sie nicht komplett im Bild untergebracht werden kann, soll das Programm eine entsprechende Warnmeldung ausgeben.

(b) Erstellen Sie ferner ein Programm `stegano-read`, das ein pgm-Bild an der Standardeingabe entgegennimmt und die mit `stegano-write` darin versteckte Botschaft wieder an der Standardausgabe ausgibt.

Weitere Hinweise

- Einige pgm-Beispieldateien können Sie sich auf der Vorlesungsseite herunterladen oder beliebig selbst erzeugen, beispielsweise mit den Tools des netpbm-Pakets, oder mit gimp. Unter MS Windows können Sie pgm-Bilder u.a. mit *irfanview* anzeigen.
- Das Programm `stegano-write` enthält die Botschaft als Zeichenkettenkonstante. Eine solche legen Sie wie folgt fest:

```
const char *const MESSAGE = "Eine Botschaft";
```

Danach können Sie mit `MESSAGE[i]` auf das *i*-te Zeichen der Botschaft zugreifen, hier also mit `MESSAGE[0]` auf das `'E'`, mit `MESSAGE[1]` auf das `'i'` usw. Erhöhen Sie den Index *i* so lange, bis `MESSAGE[i] == '\0'` (oder 0) ist – genau dann nämlich haben Sie das Ende der Zeichenkette erreicht.

- Wenn Sie alles richtig gemacht haben, sollten Sie in der Beispieldatei `hund.pgm` eine Botschaft entdecken.

Viel Erfolg!