Oskar-Maria-Graf-Gymnasium
Neufahrn b. Freising

Abiturjahrgang
2019

S E M I N A R A R B E I T

Rahmenthema des Wissenschaftspropädeutischen Seminars:

Logik und Schaltkreise

Leitfach:

Informatik

Thema der Arbeit:

Erstellung von Anwendungsbeispielen und Implementierung

Visueller Kryptografie

Verfasser: Leonhard Floh

Kursleiter: StR Baumer

Abgabetermin: *06. November 2018*

Bewertung	Note	Notenstufe in Worten	Punkte		Punkte
schriftliche Arbeit				x 3	
Abschlusspräsentation				x 1	
				Summe:	
				Gesamtleistung nach § 29 (7) GSO = Summe:2 (gerundet)	

Datum und Unterschrift der Kursleiterin bzw. des Kursleiters

1. Kryptografie Damals und Heute

Der Begriff Kryptografie setzt sich zusammen aus den altgriechischen Wörtern „krypto“, welches als „geheim, verborgen“ übersetzt werden kann und „grafie“, welches schreiben bedeutet. Es geht hier um das Verfassen von Nachrichten, die nicht von jedem, sondern nur vom gewollten Empfänger verstanden werden sollen. Die Praktik des Verschlüsselns von Nachrichten existiert bereits seit über 5000 Jahren. Damals waren die Verfahren nicht so komplex wie heute, statt mit Computern, welche komplizierte Algorithmen nutzen, verschlüsselte man mit Hilfe des Vertauschens von Buchstaben oder anderen simplen Prozessen. Das dies der Fall war weiß man inzwischen durch verschiedene Funde, zum Beispiel eine Tontafel aus dem Jahr 1500 v. Chr., auf der ein Kochrezept stand, dessen Buchstaben verändert und vertauscht wurden. Das wohl bekannteste Verschlüsselungsverfahren der Geschichte ist die Cäsar Chiffre. Sie folgte der in der Kryptografie universalen Methode der Substitution, dem Ersetzen von Buchstaben durch neue Symbole oder, wie es hier der Fall war, durch andere Buchstaben. [1]

Das Verfahren folgt einem sehr einfachen Algorithmus. Jeder Buchstabe hat als Wert seine Position im Alphabet. Der Unverschlüsselte Buchstabe (UB) wird um eine konstante Anzahl an stellen (K) im Alphabet (A) verschoben, woraus sich der Verschlüsselte Buchstabe (VB) ergibt: UB + K = VB. Wenn UB + K > A, also die Anzahl von Buchstaben im Alphabet übersteigt, wird vom Ergebnis A subtrahiert um VB zu erhalten. Zur Entschlüsselung der Nachricht wird das Prinzip umgedreht, folglich für VB – K > 0; VB – K = UB; und für VB – K < 0; VB – K + A = UB. Für die damalige Zeit war dies ein sicheres Verfahren, da ohne Entschlüsselungshilfen man nur im Wissen der Konstanten den Originalbuchstaben herausfinden konnte. Zur Veranschaulichung: Cäsar benutzte damals stets die Konstante 3 um geheime Nachrichten zu verschlüsseln, dass heißt für folgende Buchstaben gilt:

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

D E F G H I J K L M N O P Q R S T U V W X Y Z A B C [2]

Abwandlungen dieses Verfahrens wurden und werden häufig angewandt, daraus entstanden neue Gedanken zur Entschlüsselung ohne Konstante. Im 15 Jh. erfand Leon Battista Alberti die Chiffrierscheibe. Diese besteht aus zwei mit Buchstaben bedruckten Kreisen, welche individuell verschiebbar sind. Das Verschlüsseln und

Entschlüsseln wurde stark vereinfacht, da durch Probieren die Konstante schnell herausgefunden werden konnte. [1]

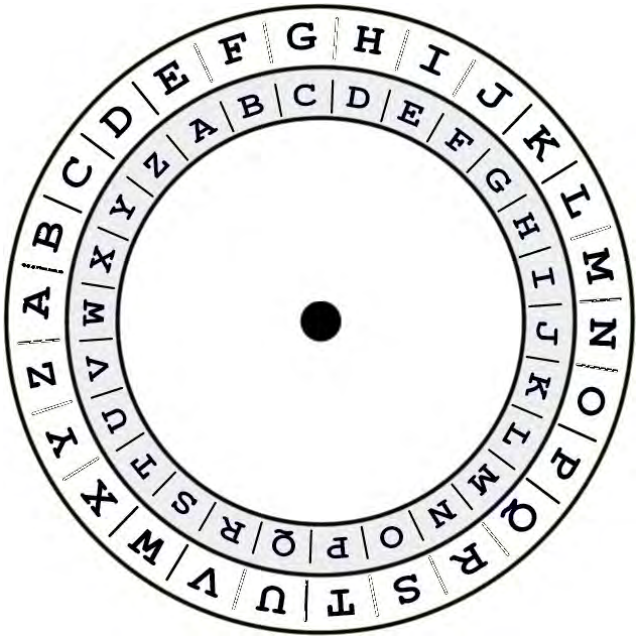


Abb. 1: Chiffrierscheibe

Die Bedeutsamkeit von Kryptografie und Kryptoanalyse kann man exemplarisch an ihrem großen militärischen Nutzen feststellen. Der Höhepunkt dessen fand sich im 2. Weltkrieg. Zu dieser Zeit begann die intensive Anwendung von Maschinen zur Verschlüsselung. Die bekannteste ist die „Engima“ eine Verschlüsselungsmaschine der Deutschen. Das erfolgreiche Erstellen der „Turing Bombe“ (Entschlüsselungs-maschine), die explizit für die von der „Engima“ verschlüsselten Texte durch amerikanische Kryptoanalytiker zur Entschlüsselung entwickelt wurde, war in den Augen heutiger Forscher möglicherweise kriegsentscheidend für die Alliierten, da die entschlüsselten Nachrichten die Positionen der deutschen U-Boote enthüllten. [3]

Wir unterscheiden symmetrische und asymmetrische Kryptografie Verfahren. Symmetrisch bedeutet, dass mit demselben Schlüssel ver- und entschlüsselt wird. Eines der ersten modernen symmetrischen Verfahren war das „DES (Data Encryption Standard)“. Durch die Entwicklung von immer rechenstärkeren PCs verlor das Verfahren seine Sicherheit. Mittels der Brute Force Methode (dem Probieren aller vorhandenen Möglichkeiten), konnten die Passwörter oder Codes sehr schnell entschlüsselt werden, da nur wenige Millionen Kombinationen existieren. Deswegen

wurde sie durch AES (Advanced Encryption Standard) als US-Standard ersetzt, welches weitaus mehr Kombinationen bietet. [1] [4] [5]

Beim asymmetrischen Verfahren werden zwei Schlüsselteile benutzt. Einen Öffentlichen zum Verschlüsseln und einen Privaten, der der Entschlüsselung dient. Dadurch muss nur der öffentliche Schlüsselteil übertragen werden um eine verschlüsselte Antwort zu erhalten, die dann der private Schlüsselteil lesbar macht. Zur Sicherheit wird zusätzlich eine Einwegsfunktion mit Falltür benutzt, deren Einwegseigenschaft den Rückschluss durch den öffentlichen Schlüssel auf den privaten Schlüssel verhindert. [5]

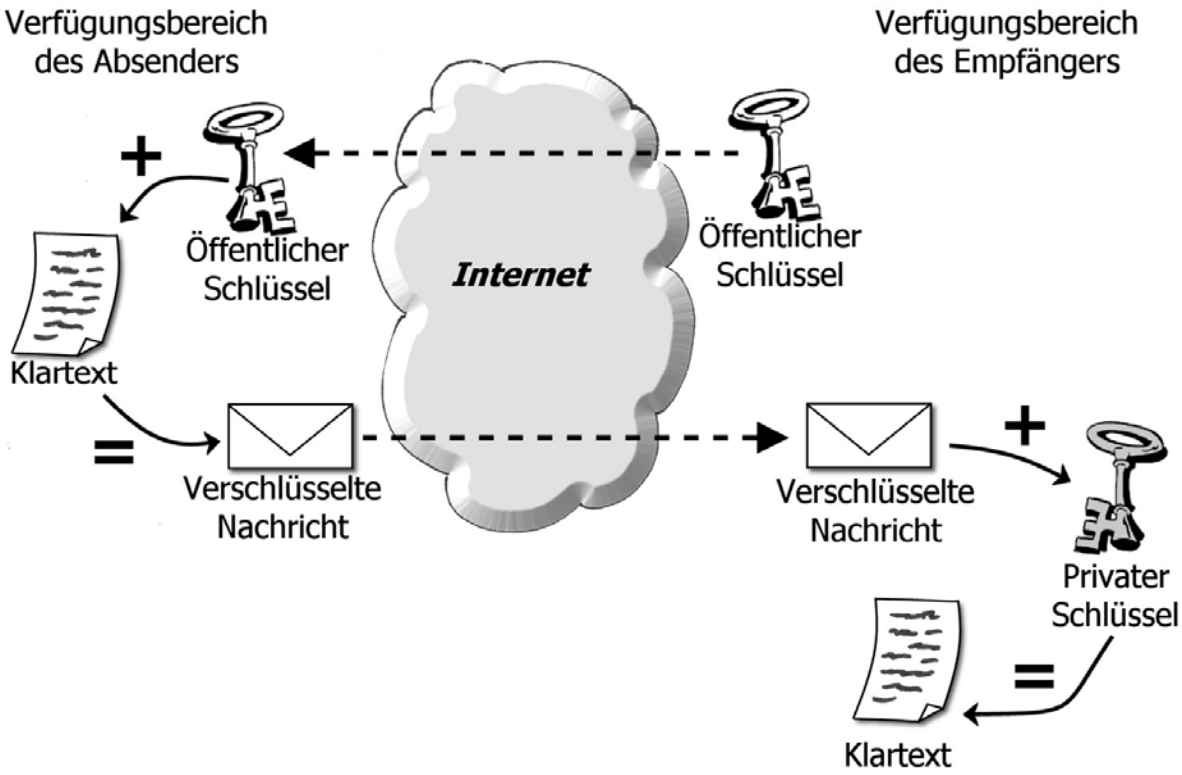
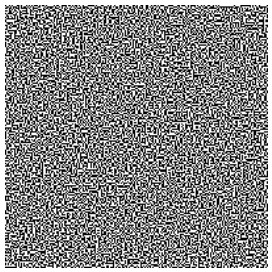


Abb. 2: Asymmetrische Verschlüsselung

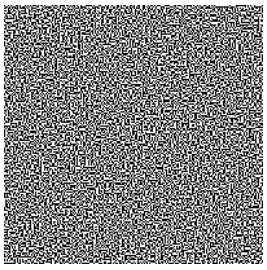
2. Visuelle Kryptografie

Die visuelle Kryptografie ist ein Konzept des “secret sharings”, entworfen von Moni Naor und Adi Shamir und basiert auf der Verschlüsselung von Bildern durch Pixelsubstitutionen. Hierbei liegt die zu verschlüsselnde Nachricht in einem Schwarzweißbild vor. Für jeden zu verschlüsselnden Pixel werden auf „n“ weiteren Folien 4 neue Teilpixel erzeugt. Beim Übereinanderlegen der neuen Folien lässt sich das Original Bild ablesen. Wichtig: ein schwarzer Pixel dominiert über einen weißen Pixel, dadurch lässt sich ein weißer Teilpixel beim Übereinanderlegen der Folien „n“ schwärzen, jedoch ein schwarzer nicht weiß färben. Die Besonderheit des Verfahrens liegt im einfachen Verständnis und der leichten Anwendung. Es wird bei Vorlage der Folien „n“ keine Berechnung benötigt um das Ursprungsbild zu erkennen, was das Fehlerrisiko deutlich minimiert und Zeit spart. [6]

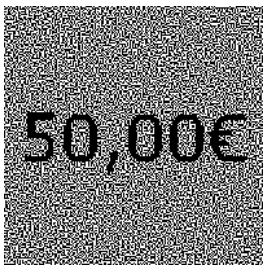
Beispiel mit 2 Folien:



Folie 1



Folie 2



Folie 1 und Folie 2 übereinander

Abb. 3: Visuelle Kryptografie

2.1 „2 aus 2“- secret sharing Verfahren

Das „2 aus 2“- secret sharing Verfahren ist das einfachste Verfahren der Visuellen Kryptografie. Hier wird ein zu verschlüsselndes Bild auf 2 Folien aufgeteilt. Es ist darauf zu achten, dass keine der beiden Folien einzeln Rückschlüsse auf das Originalbild erlauben. Jeder Pixel wird in 4 Teilpixel unterteilt. Auf der ersten Folie werden zufällig 2 sich diagonal gegenüberliegende Teilpixel, der 4 Teilpixel, geschwärzt. Die zweite Folie ergänzt, sollte der Original Pixel schwarz sein, die schwarzen Teilpixel komplementär zu Folie 1 und beim Übereinanderlegen ergibt sich ein schwarzer Pixel. Sollte der Original Pixel weiß sein ist die zweite Folie identisch zu Folie 1 und beim Übereinanderlegen ergeben sich 2 weiße und 2 schwarze Teilpixel, welche in diesem Verfahren als weiß gewertet werden. Das Ganze lässt sich in einer boolischen Matrix darstellen **M^f($m_{i,j}$)**, deren Größe ist abhängig von **Folienanzahl x Teilpixel**. Folienanzahl **(i)** entspricht der Zeile, Teilpixel Anzahl **(j)** entspricht der Spalte. **f∈{0, 1}** und steht für die Farbe der Pixel. Bei **weißen (f = 0)** bzw. bei **schwarzen (f = 1).** Wenn der j-te Teilpixel der i-ten Folie schwarz ist gilt für **($m_{i,j}$) = 1,** ist er weiß gilt **($m_{i,j}$) = 0.** [7]

Es gibt 2 Möglichkeiten zur Verschlüsselung eines weißen Originalpixels:

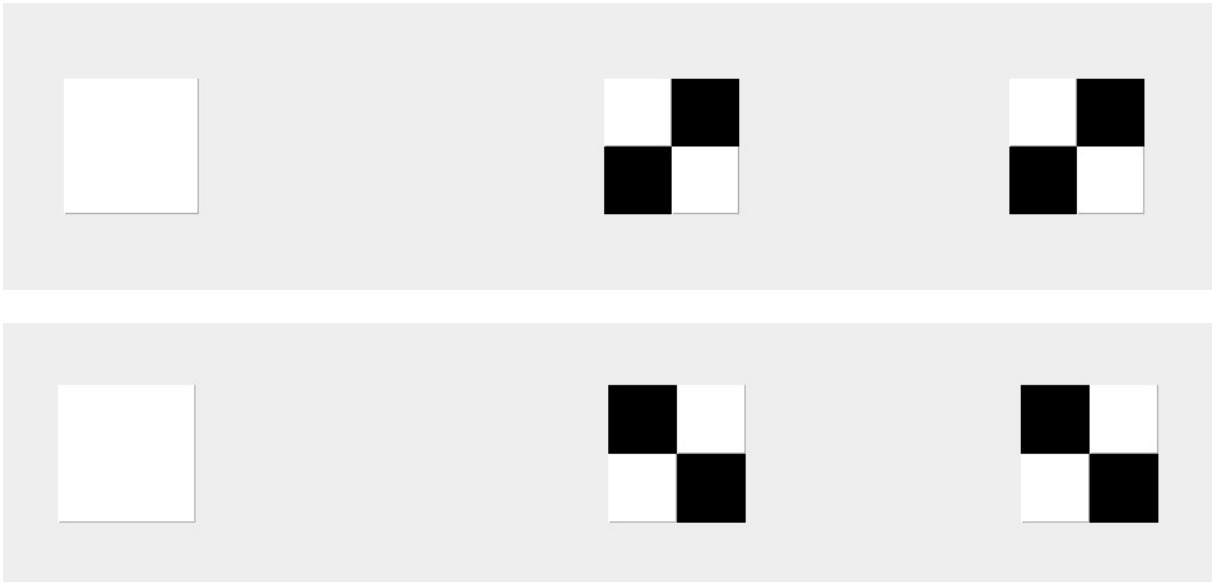


Abb. 4a: „2 aus 2“ secret sharing Verfahren ein Pixel weiß

Als Matrix wäre dies M^0 = $\begin{matrix} 0\,1\,1\,0 \\ 0\,1\,1\,0 \end{matrix}$ oder M^0 = $\begin{matrix} 1\,0\,0\,1 \\ 1\,0\,0\,1 \end{matrix}$

Die 2 Möglichkeiten zur Verschlüsselung eines schwarzen Originalpixels:

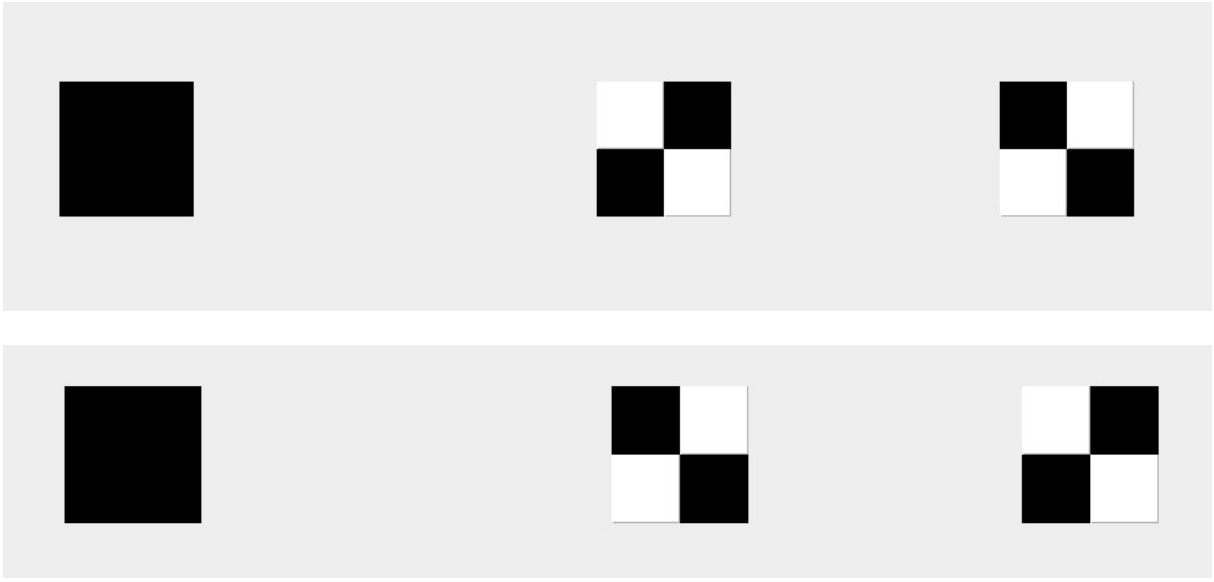


Abb. 4b: „2 aus 2“ secret sharing Verfahren ein Pixel schwarz

Als Matrix wäre dies M^1 = $\begin{matrix} 0\ 1\ 1\ 0 \\ 1\ 0\ 0\ 1 \end{matrix}$ oder M^1 = $\begin{matrix} 1\ 0\ 0\ 1 \\ 0\ 1\ 1\ 0 \end{matrix}$

Folienbeispiel für ein Bild mit 4 Pixeln:

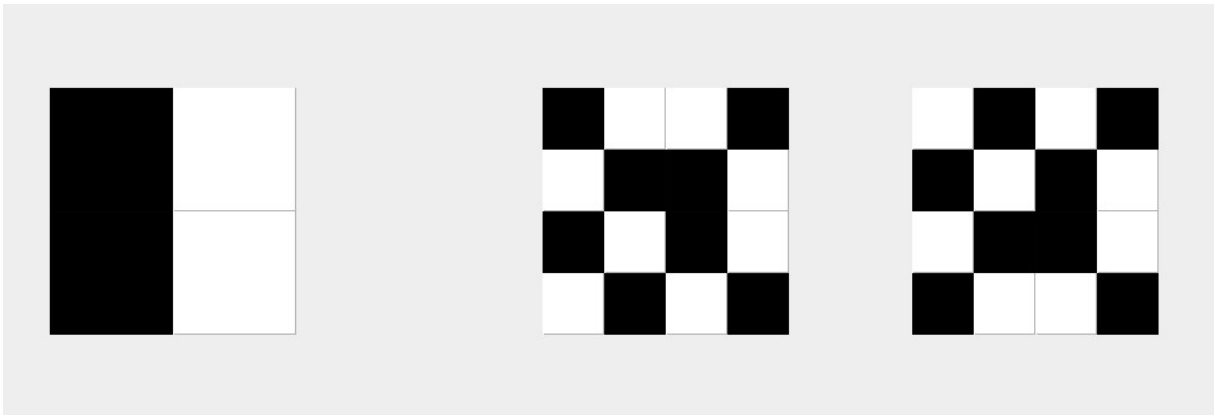


Abb. 5: „2 aus 2“ secret sharing Verfahren vier Pixel

2.2 Neues (maxrandom) Prinzip zur Verschlüsselung

Um mehr Variationen der Bilder zu erzeugen und diese leichter erkennbar zu machen, werden verschiedene Grau Stufen eingeführt. Diese ergeben sich durch eine gewisse Anzahl an schwarz gefärbten Teilpixeln. Man hat im Originalbild nicht mehr nur die Farben weiß und schwarz, das aus 2 schwarzen bzw. 4 schwarzen Pixeln zusammen gesetzt wird sondern pro schwarzem Pixel eine genaue Helligkeitsstufe. So ergeben sich 5 mögliche Helligkeitsstufen. Für **s ∈ {0, 1, 2, 3, 4}** wenn s die Anzahl schwarzer Teilpixel nach dem Übereinanderlegen beider Folien ist, gilt **s = 0;** ist die Farbe weiß. Dadurch, dass weiß als komplett weiß und nicht aus 2 weißen und 2 schwarzen Teilpixeln besteht, sollen Bilder besser erkennbar sein. Je größer s ist, desto dunkelgrauer wird der Pixel bis **s = 4;** dann ist die Farbe schwarz.

Das Ziel ist die Anzahl der Verschlüsselungsmöglichkeiten zu vergrößern, und damit die Sicherheit zu erhöhen, hierzu werden möglichst viele Teilpixel zufällig bestimmt.

Auf der Folie 1 werden, sollte s = 4 sein, zufällig 1-3 Teilpixel geschwärzt um die maximale Anzahl an zufälligen Möglichkeiten zu erhalten. Weder 4 noch 0 sind möglich, da s = 4 der einzige s-Wert >=4 ist. Sollten auf Folie 1 oder Folie 2 4 Teilpixel schwarz sein, ist ein Rückschluss möglich. Bei den darauf folgenden Abstufungen werden auf Folie 1 die Anzahl Teilpixel (F1) von 0 – s zufällig geschwärzt um auch hier die maximale Anzahl an zufälligen Möglichkeiten zu erhalten.

Die Folie 2 ergänzt die Folie 1 um auf s zu kommen. S-F1 Teilpixel werden komplementär zu Folie 1 schwarz gefärbt um die gewünschte Färbung s zu erreichen. Ist s erreicht, werden die noch nicht geschwärzten Teilpixel auch auf Folie 2 weiß gelassen um s nicht zu überschreiten. Um auch hier eine erhöhte Anzahl an zufälligen Möglichkeiten zu erhalten, werden schwarze Teilpixel der Folie 1 auf der Folie 2 zufällig schwarz oder weiß gefärbt. Es ist zu beachten, dass bei s=4 auch auf der Folie 2 nicht alle 4 Teilpixel schwarz sind. Deshalb sind bei diesem zufälligen Fall alle nicht komplementär geschwärzten Teilpixel nachträglich weiß gefärbt.

3. Implementierungen in Java

Für die Implementierung in Java wird das Programm BlueJ benutzt mit den vorgegebenen Klassen:

- Zeichnung
- StaticTools
- Rechteck

Mittels der Klasse Zeichnung wird das Fenster und der Frame erzeugt. Durch die Klasse Rechteck zeichnen sich die Rechtecke im Frame selbst. Die Klasse Rechteck ermöglicht zusätzlich die Größe und die Position derer festzulegen. Die Klasse StaticTools vereinfacht das Färben von Rechtecken, indem statt Farbparametern deutsche Farbnamen benutzt werden.

3.1 Implementierung 2er Bilder nur mit weiß-grau-schwarzen Pixel

Um die Bilder zu Verschlüsseln werden Originale benötigt. Es werden 2 Bilder, ein Dinobild und ein Smiley, erstellt. Dies geschieht in der neu erschaffenen Klasse Bild. Zunächst werden 32 x 32 Rechteck Attribute in einer Matrix erstellt, jedes dieser Rechtecke ist 5 Le hoch und 5 Le breit. Diese 32 x 32 Rechtecke werden beim Aufrufen der Klasse Bild mittels zweier for-Schleifen gezeichnet. Die eine for-Schleife zählt die Zeilen (j) und die andere die Spalten (i).

In den individuellen Bild Methoden Dinobild() und Smiley(), werden den Rechtecken in 4er Blöcken indirekt durch die Variable x [265] über mehrere for-Schleifen (i) ein Wert zugewiesen x[i]$\in$ {0, 1, 2, 3, 4}.

Zunächst werden allen Rechteckblöcken der Wert x[i] = 4 zugewiesen. Erst bei der graphischen Umsetzung werden die Rechteckblöcke, die andere Helligkeitsstufen erhalten sollen mit x[i]$\in$ {0, 1, 2, 3,} überschrieben.

Bei der Methode Smiley() wird zunächst ein Kreis erstellt, der die Farbe hellgrau erhalten wird. Dabei wird x[i] = 1 bei den nötigen Feldern gesetzt. Da der Kreis aus Quadraten besteht, wird den Randbezirken der x[i] Wert 3 zugewiesen um das Objekt runder erscheinen zu lassen. Den Rechteckblöcken für Mund und Augen werden x[i]=4 zugewiesen.

Die Methode Dinobild() erzeugt ein detailreicheres Bild. Die Grundgestalt des Dinos wird dunkler gehalten mit x[i] = 2. Die Details, Augen und Zähne, werden mit x[i] = 0 bzw. x[i] = 1 gesetzt. Die Schuppen erhalten die Färbung x[i] = 3.

Um aus den Zahlenwerten etwas Visuelles darzustellen wird die Methode einfachFärben() in die Klasse Bild implementiert. Zunächst wird die Variable p eingeführt und die for-Schleifen (int j=1; j<=31; j=j+2) und (int i=1; i<=31; i=i+2) hinzugefügt. p wird am Ende der inneren for-Schleife (int i=1; i<=31; i=i+2) hochgezählt. Die zwei for-Schleifen gehen, jede zweite Spalte und zweite Zeile durch. Danach gehen die for-Schleifen alle x[p]- Werte in Verbindung mit if-Befehlen durch um die Rechteckblöcke entsprechend des x-Wertes mit der aus StaticTools vorgegebenen Methode setzeFarbe() zu färben. Da durch p immer nur das obere linke Rechteck eines 4er Blocks aufgerufen wird müssen das rechte, das untere und das diagonal unten Rechte mit p gleich gefärbt werden. Problem bei Klasse StaticTools: Mangel an Grau Variationen, deswegen wird für x[p]=2 und x[p]=3 dieselbe Färbung verwendet.

Der Anwender wählt eine der beiden Bild Methoden aus und benutzt die Methode einfachFärben() um ein Bild zu erzeugen, welches erkennbar ist und später verschlüsselt werden kann.

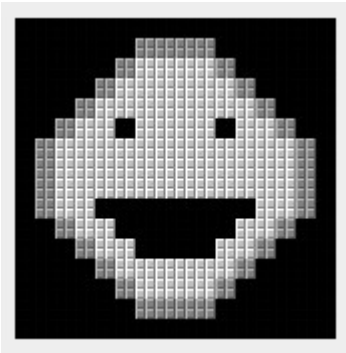


Abb. 6a: 32 x 32 Rechtecke nach Ausführen der Methoden Smiley() und einfachFärben()

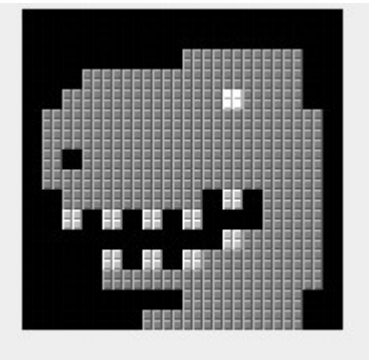


Abb. 6b: 32 x 32 Rechtecke nach Ausführen der Methoden Dinobild() und einfachFärben()

3.2 Implementierung der „2aus2“ Verschlüsselung

Für die Verschlüsselung auf 2 Folien werden 2 neue 32 x 32 Rechteck Felder mit denselben Maßen gebraucht (V1[ ] [ ]) und (V2[ ] [ ]). Beide benötigen pro Pixel einen Zahlenwert ob er schwarz oder weiß ist (v1[ ] [ ]) bzw. v2([ ] [ ]). v1,2[ ] [ ] ∈ {1, 2}. Auch hier werden die Variable p und di zwei for-Schleifen in derselben Funktion benutz wie bei der Implementierung der Methode einfachFärben(). Da im 2 aus 2 Verfahren keine Graustufen möglich sind werden Stufen 3 und 4 zu Schwarz und Stufen 0,1,2 zu weiß zusammen gefasst. Dann gibt es einen If Befehl oder x[p] unter schwarz oder weiß fällt. Es wird v[j] [i] zufällig mit (int)(Math.random()*2+1) zufällig auf 1 oder 2 gesetzt und somit V[j] [i] weiß oder schwarz gefärbt. Abhängig davon werden die anderen Pixel des 4er Blocks gefärbt diagonal rechts gleich und die anderen beiden Komplementär. Wichtig hierbei ist dass ihnen auch die richtigen Zahlenwerte zugeordnet werden. Der zugehörige 4er Block der zweiten Folie V2[j] [i] wird, sollte x[p] unter schwarz fallen dann komplementär zu Folie 1 gefärbt ansonsten identisch. Die Zahlenwerte v1,2[ ] [ ] werden entsprechend gesetzt.

Durch Aufrufen der Methode ZweiAusZweiVerfahren() wird das Original, welches nur aus von Dinobild() oder Smiley() erzeugten Werten besteht, auf die zwei 32 x 32 Felder aufgeteilt und somit verschlüsselt. Dazu muss nicht vorher mit einfachFärben() gefärbt werden.

Die beiden Folien werden auf den 32 x 32 Feldern des Originals als Ursprungsbild erkennbar, wenn man die Methode Übereinanderlegen() aufruft. Diese überprüft ob v1[j] [i] oder v2[j] [i] = 2 (also schwarz) sind, dann erscheint im Original der entsprechende Pixel schwarz, ansonsten weiß. Dies wird dann für die restlichen drei Felder v1[j+1] [i], v1[j] [i+1] und v1[j+1] [i+1] wiederholt.

3.3 Implementierung des maxrandom Prinzips

Die Implementierung dieses Prinzips ist wesentlich aufwendiger. Grundlegend wird das Prinzip der for-Schleifen und der Variablen p übernommen. Hier gibt es 5 verschiedene Stufen gegen vorher 2. Daher wird jede einzeln für sich betrachtet. Somit besteht die Methode innerhalb der for-Schleifen aus 5 großen If’s die erst einmal die Helligkeitsstufe abfragen. Analog dem oben beschriebenen Prinzips des größtmöglichen Zufalls, werden die Werte und Farben zugeordnet.

Es wird nur der kompliziertere Teil erklärt, da die Erläuterung jedes Teiles dieser Methode die zulässige Seitenzahl dieser Arbeit überschreiten wird.

```
if (v1[j][i+1] ==1&& v1[j][i] + v1[j+1][i] + v1[j][i+1] + v1[j+1][i+1] ==6
|| v2[j][i]==2 &&v1[j][i] ==1&& v1[j][i] + v1[j+1][i] + v1[j][i+1] + v1[j+1][i+1] >=5
||v2[j][i]+v2[j+1][i] ==4
|| v2[j+1][i]==2 &&v1[j+1][i] ==1&& v1[j][i] + v1[j+1][i] + v1[j][i+1] + v1[j+1][i+1] >=5  )
{  v2[j][i+1]=1;  V2[j][i+1].setzeFarbe("weiss");}

else{

if (v1[j][i+1] ==1&& v1[j][i] + v1[j+1][i] + v1[j][i+1] + v1[j+1][i+1] + v2[j][i]+ v2[j+1][i] ==6 )
      {  v2[j][i+1]=2;  V2[j][i+1].setzeFarbe("schwarz");}else{
  v2[j][i+1]=(int)(Math.random()*2+1);
      if(v2[j][i+1]==2) (V2[j][i+1].setzeFarbe("schwarz");}
      else{V2[j][i+1].setzeFarbe("weiss");}} }
```

Abb. 7: Auszug aus dem Quellcode zur Implementierung des maxrandom Prinzips

Dieser findet sich im if für die Stufe [p] = 2 bei der Ergänzung von v1[j] [i+1] durch v2[j] [i+1]. Da x[p] = 2, dürfen maximal zwei Pixel schwarz sein. Unter den zwei Bedingungen, dass v1[j][i+1] = 1 und alle v1[ ] [ ] in Summe 6 ergeben (also v1[ ] [ ] zwei Teilpixel mit Wert 2 enthält) folgt: v2[j] [i+1] muss 1 sein, um keinen komplementär zu v1[j] [i+1] Teilpixel mit dem Wert 2 zu erzeugen.

Auch muss v2[j] [i+1] 1 sein, wenn v1[ ] [ ] in Summe 5 ergeben und v1[j][i] = 1oder v1[j+1][i] =1 bereits durch v2[j][i] bzw. v2[j+1][i] komplementär mit 2 ergänzt wurde. Auch hierfür ist es Bedingung, dass v1[j] [i] = 1 ist. Die letzte Möglichkeit, dass v2[j] [i+1] = 1 sein muss, ist, wenn v2[j][i] und v2[j+1][i] beide 2 sind, da dann auch hier bereits 2 Teilpixel schwarz sind. Wenn dies nicht der Fall ist, wird zufällig entschieden ob v2[j] [i+1] 1 oder 2 ist. Wenn alle bisher bestimmten v1[ ] [ ] und v2[ ] [ ] 1 sind, dann muss v2[j][i+1] = 2 sein. Ansonsten können die durch x[p]=2 schwarzen Teilpixel nicht mehr erreicht werden.

Nach dem Aufruf der Methode maxrandom() zur Verschlüsselung ist es möglich die Methode Übereinanderlegen() aufzurufen um die beiden Folie übereinander, also entschlüsselt auf dem 32 x 32 Feld des Originals zu sehen.

3.4 Umfrage über verschiedene implementierte Bilder

Um die Erkennbarkeit der verschiedenen Folien der unterschiedlichen Verschlüsselungssysteme und verschiedenen Bilder rauszufinden werden 3 Personen gefragt ob sie die Figur im Bild erkennen können, ob Umrisse einer Figur erkennbar sind, oder ob Nichts erkennbar ist. Keine der befragten Personen konnte weder bei der ersten noch bei der zweiten Folie des „2aus2“ Verfahrens etwas erkennen. Beim maxrandom Verfahren wird von 2 der befragten Personen eine kreisförmige Figur in der ersten Folie des Smiley Bildes erkannt. In der zweiten erkannte hingegen keine der Personen etwas. Bei dem Dinobild erkannte auf der ersten Folie keiner der Befragten etwas. Auf der zweiten gaben alle an die Umrisse einer Figur zu erkennen, einer genauer: Umrisse eines Tierkopfes. Beim Übereinanderlegen der durch das „2aus2“ Verfahren erzeugten Folien können alle Personen die jeweilige Figur gut erkennen. Beim Übereinander legen der des durch da maxrandom Verfahrens erzeugten Folien auch, jedoch gaben beim Smiley alle an er sei schlechter zu erkennen als beim „2aus2“ Verfahren. Beim Dinobild variierten die Meinungen zwischen besser, schlechter und gleich. Zur Überprüfung wurden die Personen danach gefragt was sie bei den einfachgefärbten Bildern erkennen. Alle Personen gaben richtig Smiley und Dino an. Für die genauen Daten der Umfrage siehe Anhang (Seiten 16 -19).

4. Fazit

Beim Vergleich des „2aus2“ Verfahren mit dem maxrandom Verfahren hinsichtlich Schwierigkeitsgrad beim Implementieren, Sicherheit der verschlüsselten Folien und Erkennbarkeit des wieder entschlüsselten Bildes, lässt sich generell sagen, dass das alte „2aus2“ Verfahren dem neuen Verfahren in allen Aspekten überlegen ist.

Der Code zur Implementierung des „2aus2“ Verfahrens ist gerade mal knapp 50 Zeilen lang und einfach. Der Code des maxrandom Verfahrens benötigt über 250 Zeilen und ist weitaus komplizierter.

Die verschlüsselten Folien des „2aus2“ Verfahrens sind laut der Umfrage nicht erkennbar, in den verschlüsselten Folien des neuen Prinzips können teilweise Umrisse und Figuren erkannt werden.

Beide Verfahren liefern erkennbare Entschlüsselungen. Die des „2aus2“ Verfahrens sind bei einfachen Bildern, wie dem Smiley, besser erkennbar. Nur bei detailreichen Bildern wie dem Dinobild sind die beiden Verfahren in diesem Punkt gleich gut.

Somit sind die beiden Annahmen des maxrandom Verfahrens, dass eine erhöhte Anzahl an zufälligen Möglichkeiten die Sicherheit erhöht, und dass die Einführung der neuen Helligkeitsstufen die entschlüsselten Bilder besser darstellt, falsch. Das alte „2aus2“ Verfahren liefert gerade durch seine Schlichtheit die bessere Möglichkeit zur Verschlüsselung von Bildern.

A Anhang

A.1 Daten der Umfrage

Verschlüsselte Folie 1 „2 aus 2“ Verfahren:

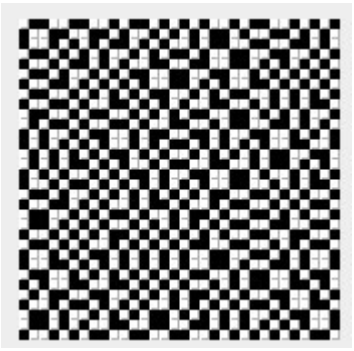


Abb.8a: Smiley Folie 1 „2 aus 2“ Verfahren

Person 1: Nichts erkennbar
Person 2: Nichts erkennbar
Person 3: Nichts erkennbar

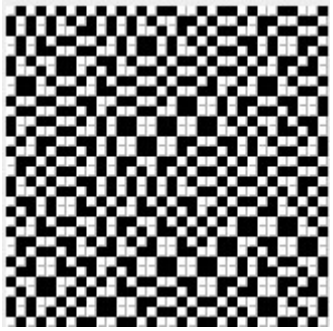


Abb.8b: Dinobild Folie 1 „2 aus 2“ Verfahren

Nichts erkennbar
Nichts erkennbar
Nichts erkennbar

Verschlüsselte Folie 2 „2 aus 2“ Verfahren:

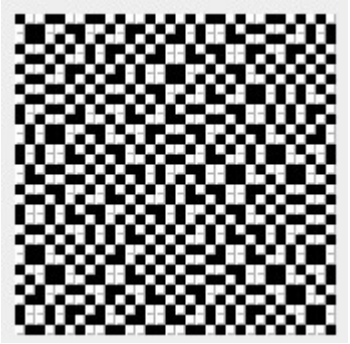


Abb.9a: Smiley „2aus2“ Verfahren Folie 2

Person 1: Nichts erkennbar
Person 2: Nichts erkennbar
Person 3: Nichts erkennbar

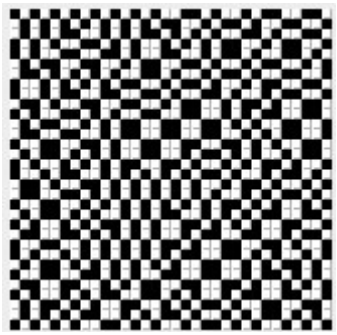


Abb.9b: Dinobild „ 2 aus 2“ Verfahren Folie 2

Nichts erkennbar
Nichts erkennbar
Nichts erkennbar

Verschlüsselte Folie 1 „maxrandom“ Verfahren:

Abb.10a: Smiley maxrandom Folie 1

Person 1: Kreis leicht erkennbar
Person 2: Nichts erkennbar
Person 3: Runde Fläche erkennbar

Abb.10b: Dinobild maxrandom Folie 1

Nichts erkennbar
Nichts erkennbar
Nichts erkennbar

Verschlüsselte Folie 2 „maxrandom“ Verfahren:

Abb.11a: Smiley maxrandom Folie 2

Person 1: Nichts erkennbar
Person 2: Nichts erkennbar
Person 3: Nichts erkennbar

Abb.11b: Dinobild maxrandom Folie2

Umrisse eines Tierkopfes
Umrisse einer Figur
Umrisse einer Figur

Übereinandergelegt „2 aus 2“ Verfahren:

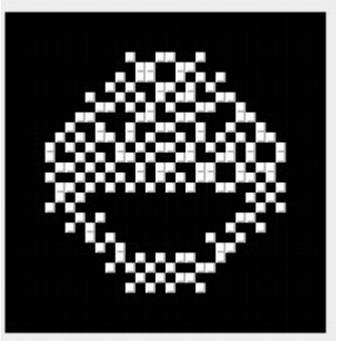


Abb.12a: Smiley „2aus2“ Verfahren übereinandergelegt

Person 1: Smiley erkennbar
Person 2: Smiley erkennbar
Person 3: Smiley erkennbar

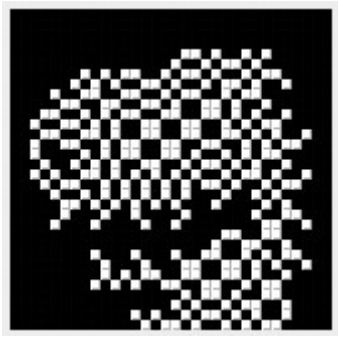


Abb.12b: Dinobild „2aus2“ Verfahren übereinandergelegt

Dino erkennbar
Dino erkennbar
Dinokopf erkennbar

Übereinandergelegt „maxrandom“ Verfahren:

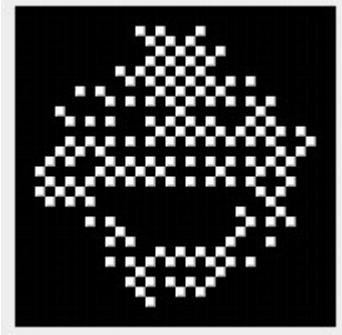


Abb.13a: Smiley maxrandom übereinandergelegt

Person 1: Schlechter erkennbar
Person 2: Schlechter erkennbar
Person 3: Schlechter erkennbar

Abb.13b: Dinobild maxrandom übereinandergelegt

gleich
besser erkennbar
schlechter erkennbar

Einfachgefärbt :

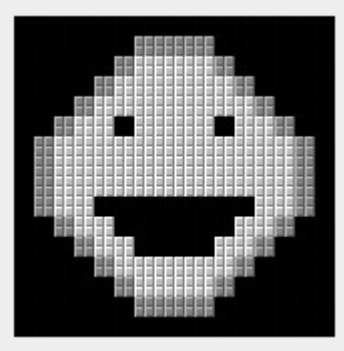


Abb.14a: Smiley Einfachgefärbt

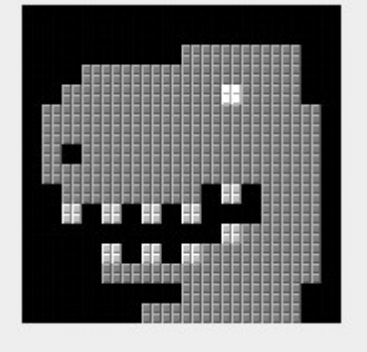


Abb.14b: Dinobild Einfachgefärbt

Person 1: Smiley

Person 2: Smiley

Person 3: Smiley

Dino

Dino

Dino

A.2 Quellcode der Klasse Bild

Abbildungsverzeichnis

Quellenverzeichnis

[1] Oliver Kuhlemann *Kryptografie Index Einleitung*, (zuletzt abgerufen am 28.10.2018). [Online]: http://kryptografie.de/kryptografie/index.htm

[2] Oliver Kuhlemann *Kryptografie Caesarchiffre*, (zuletzt abgerufen am 28.10.2018). [Online]: http://kryptografie.de/kryptografie/chiffre/caesar.htm

[3] Oliver Kuhlemann *Kryptografie Chiffriermaschine Engima*, (zuletzt abgerufen am 28.10.2018). [Online]: http://kryptografie.de/kryptografie/chiffre/enigma.htm

[4] Oliver Kuhlemann *Kryptografie moderne Verfahren aes* (zuletzt abgerufen am 28.10.2018). [Online]: http://kryptografie.de/kryptografie/chiffre/aes.htm

[5] Oliver Kuhlemann *Kryptografie asymmetrische Verfahren*, (zuletzt abgerufen am 28.10.2018). [Online]: http://kryptografie.de/kryptografie/asymmetrisch.htm

[6] Lennart Suhr *Visuelle Kryptografie* 2011 S.3 (zuletzt abgerufen am 29.10.2018). [Online]: https://www.thi.unihannover.de/fileadmin/forschung/arbeiten/suhr-sa.pdf

[7] Lennart Suhr *Visuelle Kryptografie* 2011 S.4-5 (zuletzt abgerufen am 29.10.2018). [Online]: https://www.thi.unihannover.de/fileadmin/forschung/arbeiten/suhr-sa.pdf

Erklärung zur Seminararbeit

Hiermit erkläre ich, dass ich die vorliegende Arbeit selbstständig und ohne fremde Hilfe verfasst und keine anderen als die angegebenen Hilfsmittel verwendet habe. Insbesondere versichere ich, dass ich alle wörtlichen und sinngemäßen Übernahmen aus anderen Werken als solche kenntlich gemacht habe.

Neufahrn, 05.11.2018 .....