Bernd Borchert

Univ. Tübingen WS 13/14

Vorlesung

Kryptographie

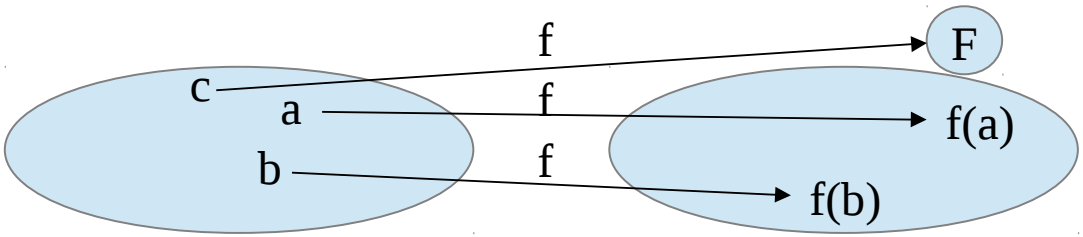
Teil 11

Einwegfunktionen, Geheimnisteilung, Steganographie

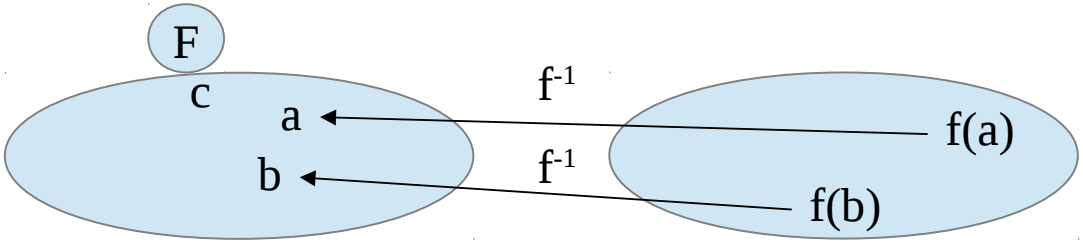
7.2.14

P-Zeit berechenbare Funktionen

FP ist die Klasse aller Funktionen f : {0,1}* → {0,1}* ∪ {F} , die sich in P-Zeit berechnen lasse, Beispiel Sortierung, Addition, Multiplikation, etc. F ist „Fehlerwert“.



Eine Funktion f aus FP heißt *injektiv* (one-one) falls jeder Wert f(x) (von F abgesehen) nur einmal angenommen wird, d.h. es gibt kein anderes y mit f(y)=f(x). Damit gibt es eine Umkehrfunktion f^{-1}: {0,1}* → {0,1}* ∪ {F}, die f(x) auf x abbildet und alle Nicht-f-Bilder auf den Fehlerwert F.



Eine Funktion f aus FP heißt *nicht-stark-verkürzend*, wenn es ein k gibt, so dass $|f(x)|^k > |x|$ für alle x.

Einwegfunktionen

Eine injektive, nicht-stark-verkürzende Funktion aus FP heißt *Einwegfunktion*, wenn f^{-1} nicht in FP ist.

Kandidaten für Einwegfunktionen (leider unbewiesen):

1. $f(<p,q>) = pq$, mit $p<q$ Primzahlen (in Binärdarstellung).

Bem.: seit 2001 gibt es einen P-Zeit Algorithmus für Primalität. In alten Kyptologie-Büchern findet man deshalb noch Zertifikate der Primalität von p als Teil der f Eingabe.

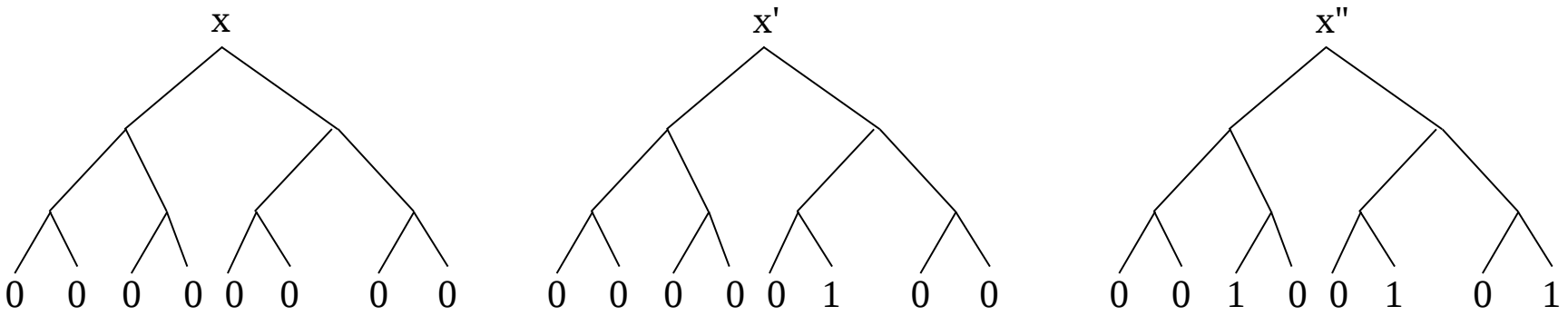
2. $f(<p,r,x>) = <p,r,r^x \bmod p>$ mit p Sophie-Germain Primzahl und $1 < r,x < p-1$.

3. AES, im Feedback Modus mit dem Schlüssel k selber als mehrfacher Teil der Eingabe.

Die bekannten Hashfunktionen (SHA1 etc.) sind übrigens keine Einwegfunktionen, denn jeder Hashwert hat im Allgemeinen sehr viele Urbilder, denn die Hashwerte sind ja alle von kurzer Länge (bei SHA1 z.B. immer 160 bit).

P, NP und UP

P ist die Klasse aller in deterministischer P-Zeit berechenbaren Sprachen. (Sprache = Teilmenge von {0,1}* = Abbildung von {0,1}* → {0,1}).
NP ist die Klasse aller von einer nichtdeterministischen P-Zeit Maschine berechenbaren Sprachen L. D.h., es gilt x in L genau dann, wenn bei Eingabe x im Verzweigungsbaum (Verzweigung durch Raten) mindestens ein Pfad der Berechnung akzeptiert. In folgendem Beispiel ist x nicht in L, x' und x" aber wohl.



UP ist die Menge der Sprachen, für die es eine NP-Maschine gibt, die aber für alle Eingaben x garantiert höchstens einen akzeptierenden Berechnungspfad hat (also der linke und mittlere Fall im Beispiel oben, so was wie rechts darf für keine Eingabe passieren), und die dann ein x akzeptiert, falls es einen akzeptierenden Pfad (also genau einen) im Berechnungsbaum gibt. Im mittleren Beispiel wird x' akzeptiert, x im linken nicht.

Nach Definition ist also UP Teilmenge von NP. P ist natürlich eine Teilmenge von UP (nie raten, d.h. Baum=Wurzel). Insgesamt gilt also:

P ist Teilmenge von UP ist Teilmenge von NP.

Ob NP gleich P ist, ist nicht bekannt. Wir betrachten die Frage: Ist UP gleich P?

UP vs. P

Behauptung: UP ungleich P gdw. es Einwegfunktionen gibt.

Beweis ←
Sei f eine Einwegfunktion. Definiere L_f= {<x,y> | es gibt z < x und f(z) = y}. Behauptung: L_f ist in UP, aber nicht in P:

- L_f ist in UP: Der nondet. Algorithmus rät bei Eingabe <x,y> systematisch alle z, die Urwert von y sein könnten (wegen nicht-stark-verkürzend Eigenschaft nur exponentiell viele z < $|y|^k$) und akzeptiert wenn f(z) = y. Weil es für f nur max. einen Urwert z von y gibt, ist diese Maschine eine UP Maschine, die also genau die Sprache L_f akzeptiert.
- L_f ist nicht in P: Sei A der P-Zeit Algorithmus für L_f. Damit konstruieren wir folgende FP Maschine für f^{-1}, Idee: binäre Suche. Gegeben ein y, berechnen wir mit A, ob $<1^{|y|^{k-1}},y>$ gilt (k wieder der Wert aus der nicht-stark-verkürzend Eigenschaft). Wenn ja, dann fragen wir mit A, ob auch $<1^{|y|^{k-2}},y>$ gilt, etc. bis wir die Länge l des Urwerts gefunden haben. Innerhalb dieser Länge fragen wir weiter, ob A $<01^{l-1},y>$ akzeptiert. Wenn ja, dann fragen wir weiter mit $<001^{l-2},y>$, ansonsten mit $<101^{l-2},y>$, usw. Am Schluss haben wir den Urwert von y gefunden. D.h., f^{-1} ist in P-Zeit berechenbar. Also ist f keine Einwegfunktion. Das war aber vorausgesetzt. Also ist die Annahme, dass L_f in P ist, falsch.

L_f beweist damit, dass UP echt größer ist als P, unter der Annahme, dass es Einwegfunktionen gibt.

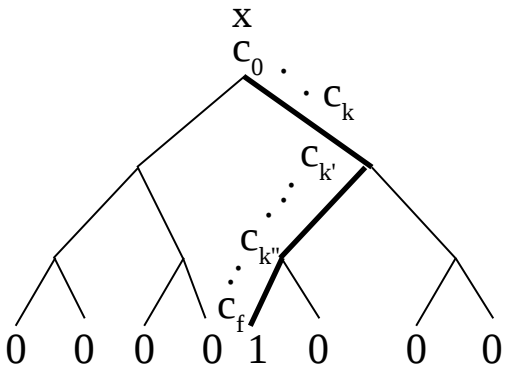
Schlechte Nachricht also: man wird nur schwer bei den Kandidaten für Einwegfunktionen nachweisen können, dass sie tatsächlich welche sind, denn dann hätte man P ungleich NP bewiesen.

UP vs. P

Behauptung: UP ungleich P gdw. es Einwegfunktionen gibt.

Beweis →
Sei L ein Sprache in UP, aber nicht in P, und sei U die L definierende nichtdeterministische Maschine. Sei y = $<c_0,...,c_f>$ eine Darstellung einer Folge von Konfigurationen einer Berechnung von U. Dann sei f(y) = x falls y bei Eingabe x die Folge der Konfigurationen von U auf dem akzeptierenden Pfad darstellt, für alle anderen y sei f(y) der Fehlerwert F.
Behauptung: f ist eine Einwegfunktion.

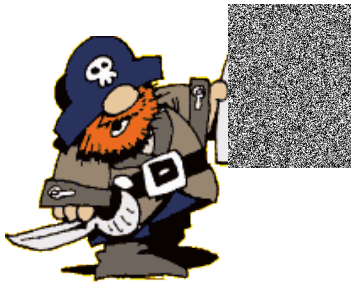
- Offensichtlich ist f in FP, denn die Prüfung, ob ein y eine zu U passende Folge von Konfigurationen ist und ob die letzte Konfiguration akzeptierend ist, ist in P-Zeit machbar, ebenso das Herausfischen von x aus der ersten Konfiguration.
- Außerdem ist f injektiv, denn es gibt maximal einen akzeptierenden Konfigurations-Pfad für einer Berechnung von U auf Eingabe x.
- Und f ist nicht-stark-verkürzend, denn die Länge der Darstellung der Konfigurationsfolge ist nur polynomiell in der Eingabelänge |x|.
- Es bleibt zu zeigen, dass die Umkehrfunktion nicht in FP ist: wenn man aus x die akzeptierende Konfigurationsfolge y in P-Zeit berechnen könnte, dann wäre das schon der P-Zeit Algorithmus für L, denn genau dann wenn y nicht der Fehlerwert ist, gibt es eine akzeptierende Berechnung für Eingabe x. Dieser P-Zeit Algorithmus für L existiert aber gemäß Annahme nicht.



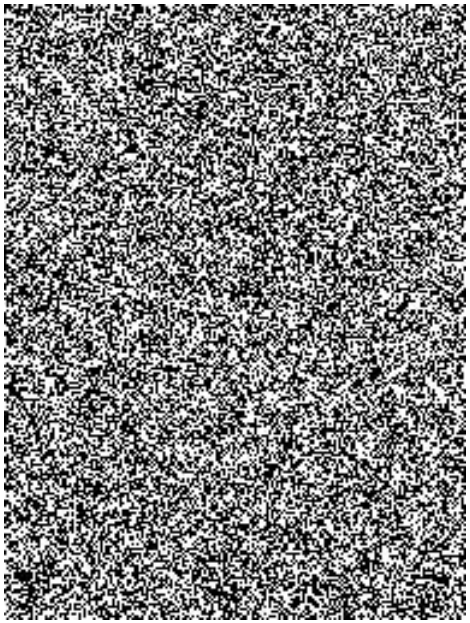
Thema Geheimnisteilung

Am Beispiel Visuelle Kryptographie

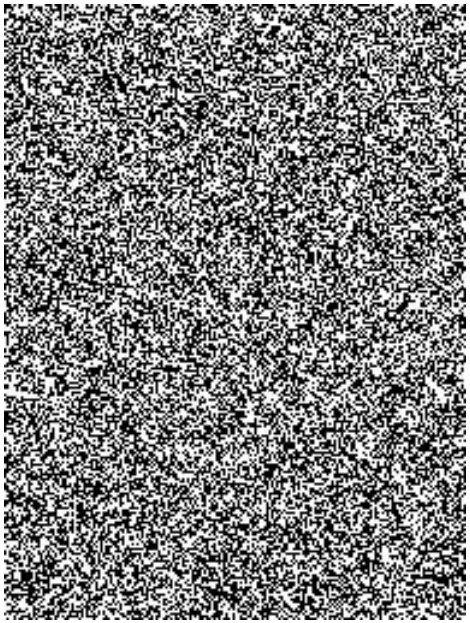
Schatzsuche



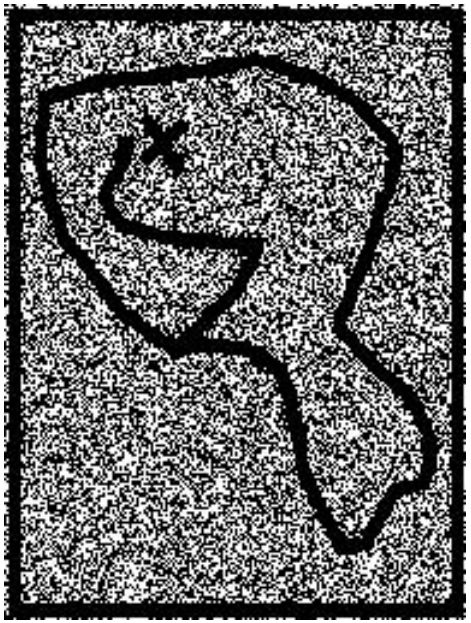
Ein Pirat hat diese Folie:



Ein anderer Pirat hat diese Folie:



Übereinandergelegt ergibt sich:



Eigenschaften:
- eine Folie allein hat keine Information
- beide Folien zusammen ergeben die gesamte Information

Wie funktioniert‘s?

EXOR-Codierung

Zu codierende Folge:	0	0	0	1	1	1	0	0	0
Zufällige Folge:	0	1	0	1	1	0	0	1	1
Differenzfolge:	0	1	0	0	0	1	0	1	1

Die Differenzfolge wird gebildet, indem bei einer 0 in der zu codierenden Folge das zufällige bit kopiert wird, während bei ein 1 in der zu codierenden Folge das zufällige bit negiert wird. Die zu codierende Folge ist also das EXOR der zufälligen und der Differenzfolge - wenn man EXOR als Addition auffasst, stimmt also der Name Differenzfolge.

- die zufällige Folge und die Differenzfolge sind von einer Zufallsfolge nicht zu unterscheiden, denn (bei fester zu codierender Folge) ist jedes bit unabhängig von den anderen bits mit Wahrscheinlichkeit 50% auf 1 gesetzt

- wenn man die zufällige Folge und die Differenzfolge untereinander schreibt, lässt sich die zu codierende Folge als EXOR herauslesen.

Codierung via bit-weises ODER von komplementären bit-Paaren

Zu codierende Folge:	0	0	0	1	1	1	0	0	0
Zufällige Folge:	01	10	01	10	10	01	01	10	10
Differenzfolge:	01	10	01	01	01	10	01	10	10
OR:	01	10	01	11	11	11	01	10	10

Die Differenzfolge wird gebildet, indem bei einer 0 in der zu codierenden Folge das zufällige komplementäre bit-Paar kopiert wird, während bei einer 1 in der zu codierenden Folge das zufällige komplementäre bit-Paar negiert wird.

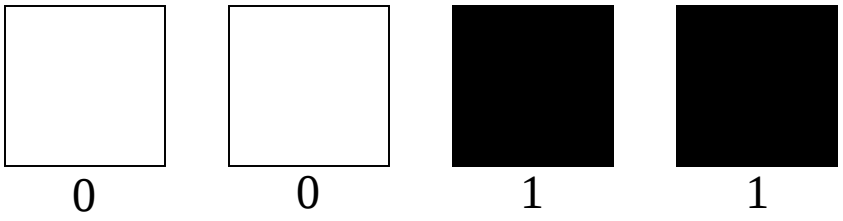
- die zufällige Folge und die Differenzfolge sind von einer Zufallsfolge nicht zu unterscheiden, denn (bei fester zu codierender Folge) ist jedes bit-Paar unabhängig von den anderen bit-Paaren mit Wahrscheinlichkeit 50% auf 10 gesetzt

- wenn man die zufällige Folge und die Differenzfolge untereinander schreibt, lässt sich die zu codierende Folge durch bit-weises ODER herauslesen: sie ist dann 1 wenn das bit-weise ODER zwei 1‘en produziert, und ist 0, wenn das bit-weise ODER nur eine 1 produziert.

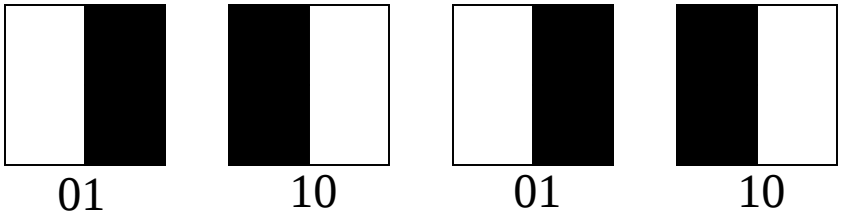
Visuelle Kryptographie

Anstatt Folgen werden jetzt die Pixel eines 2-dimensionale Bildes codiert

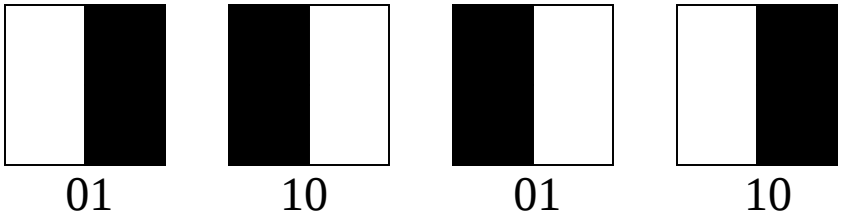
Zu codierendes Pixel:



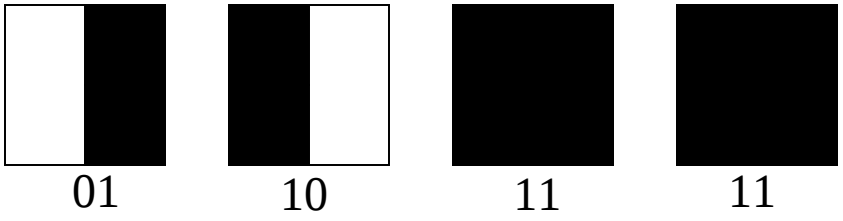
Zufälliges komplementäres Paar: (Folie 1)



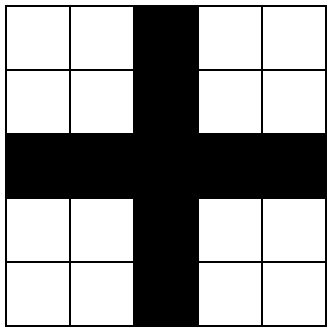
Differenz-Paar: (Folie 2)



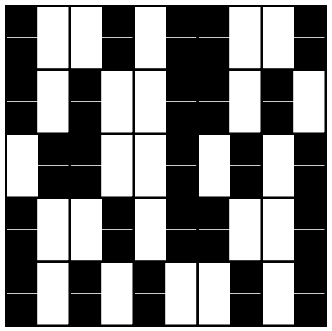
Bit-weises OR: (Folie 1 und 2 zusammen)



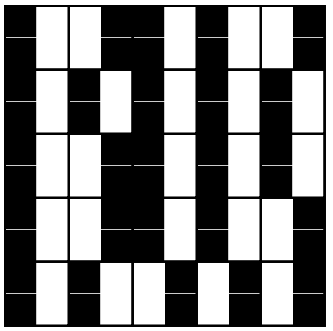
Beispiel Visuelle Kryptographie (2-aus-2 Schema)



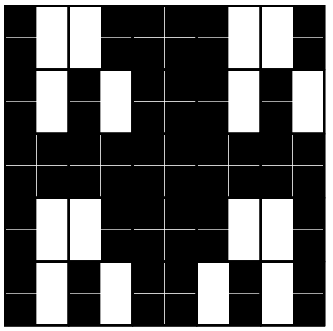
Original
-Bild



Zufalls-
Bild

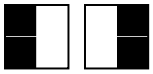
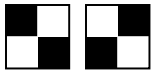


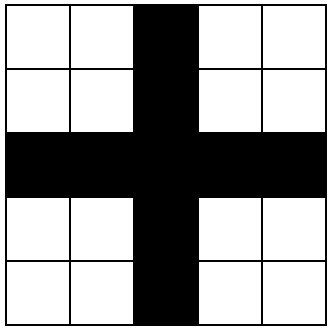
Differenz-
Bild

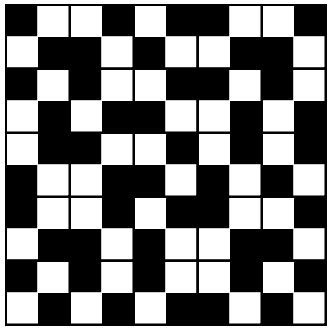


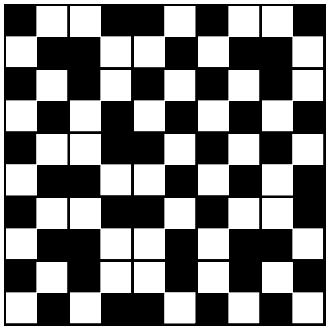
übereinander-
gelegt

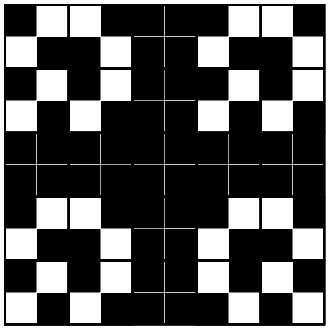
Verbesserung:

Statt  mit  codieren („verwischt“ besser)









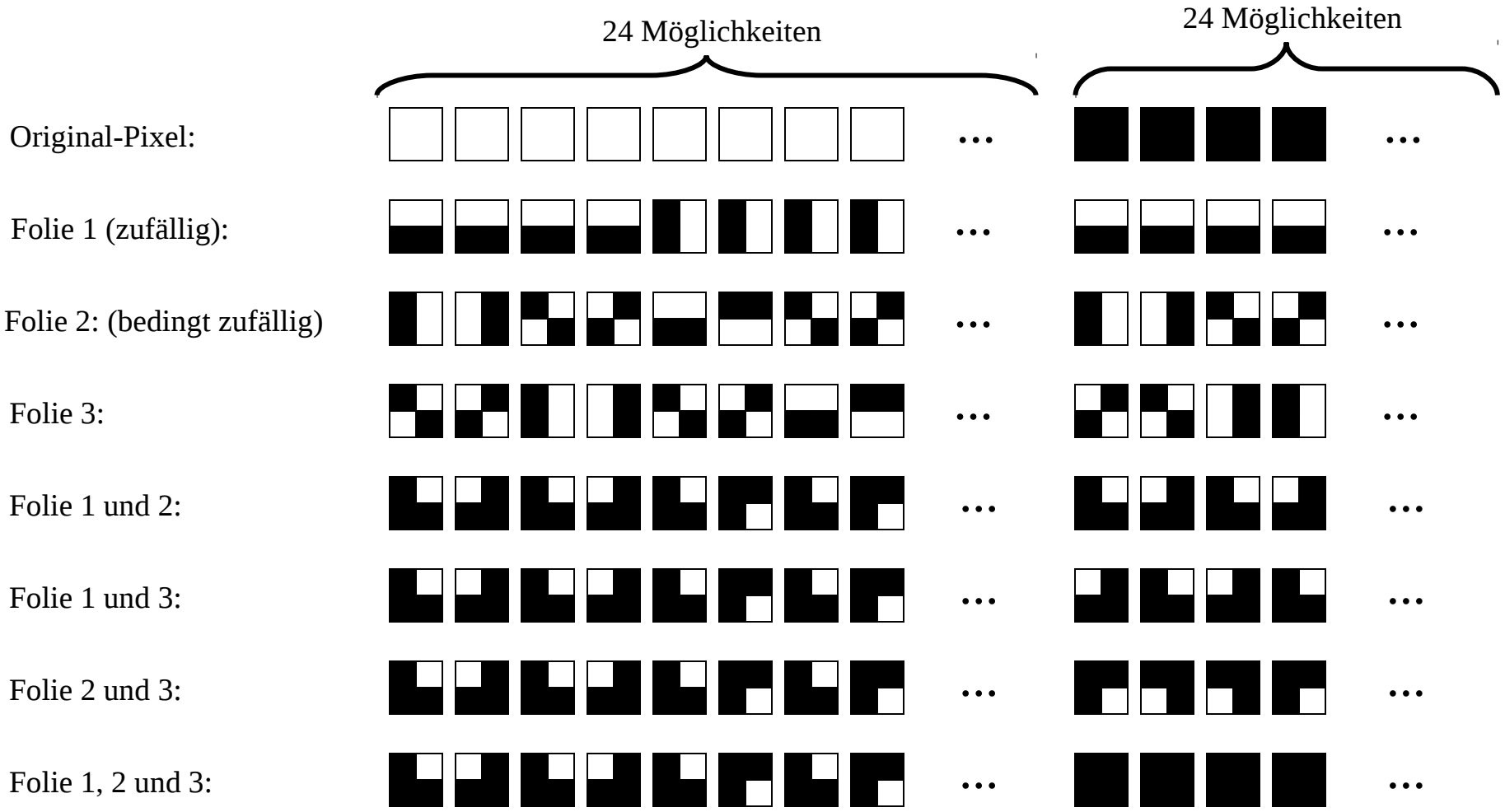
k-aus-n Schema

Bei einem k-aus-n Schema wird das Bild in n verschiedene Folien kodiert: k-1 Folien sollen auch zusammen keinerlei Information liefern, aber k davon sollen zusammen ausreichen, die gesamte Information zu rekonstruieren.

Beispiel 3-aus-3 Schema: Alle drei Piraten müssen ihre Folien zusammenlegen, zwei von Ihnen können ohne den dritten nichts damit anfangen:

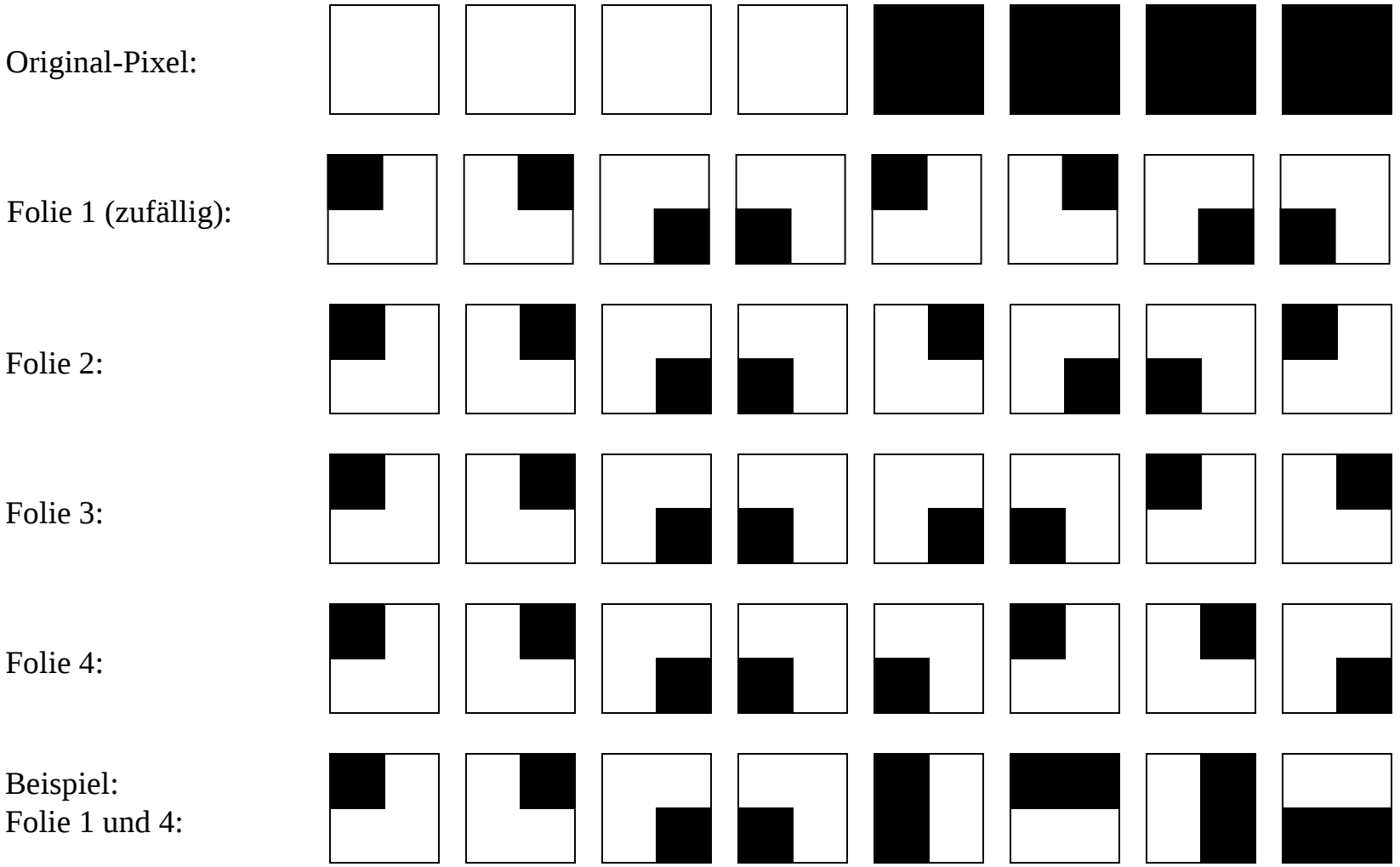
Konstruktion: 3-aus-3 Schema

Bei Folie 1 werden zwei von vier Pixel zufällig ausgewählt. Bei Folie 2 werden wieder zwei der vier Pixel zufällig ausgewählt, wovon allerdings genau eines mit denen aus Folie 1 übereinstimmen muss. Bei Folie 3 kommt das Original-Pixel ins Spiel: wenn das 0 ist, werden die zwei der vier Pixel, die bislang nur einmal gewählt wurden, gewählt, wenn das 1 ist, genau die beiden anderen.



2-aus-n Schema

Bei einem 2-aus-n Schema wird das Bild in n verschiedene Folien kodiert: eine Folie soll keine Information liefern, aber zwei davon sollen zusammen ausreichen, die gesamte Information zu rekonstruieren. Beispiel hier: 2-aus-4 Schema: (durch Wegnehmen von Folie 4 ist das auch ein 2-aus-3 Schema)



Satz. (Naor & Shamir 1994)
Es gibt für alle k,n mit $n \geq k \geq 1$ ein k-aus-n Schema.
Ein n-aus-n Schema braucht dabei 2^{k-1} Pixel für ein Pixel des Originals.

Vorgestellt wurden hier die Fälle 2-aus-3 und 3-aus-3.

Geheimnisteilungsschema n-aus-n auf Wörtern

Ein bit b soll auf n Teilgeheimnisse verteilt werden:

Dazu reicht ein bit b_i für jedes Teilgeheimnis:
n-1 Teilgeheimnisse $b_1, ..., b_{n-1}$ werden zufällig bestimmt. Das letzte Teilgeheimnis b_n wird als EXOR($b_1, ..., b_{n-1}$, b) bestimmt.

Also kann b als EXOR($b_1, ..., b_{n-1}, b_n$) bestimmt werden, aber keine Teilmenge von n-1 Teilgeheimnissen reicht alleine aus.

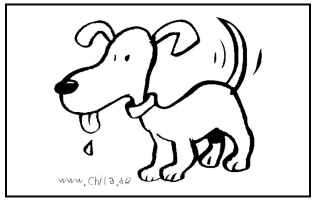
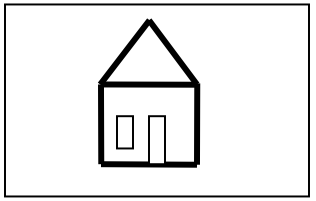
Wörtern mit m bits können also für ein Geheimnisteilungsschema n-aus-n in n Teilgeheimnisse mit je m bits aufgeteilt werden.

Bei der Visuellen Kryptographie aus dem Beispiel sind ein paar mehr bits nötig, weil EXOR da nicht möglich ist.

Steganographie

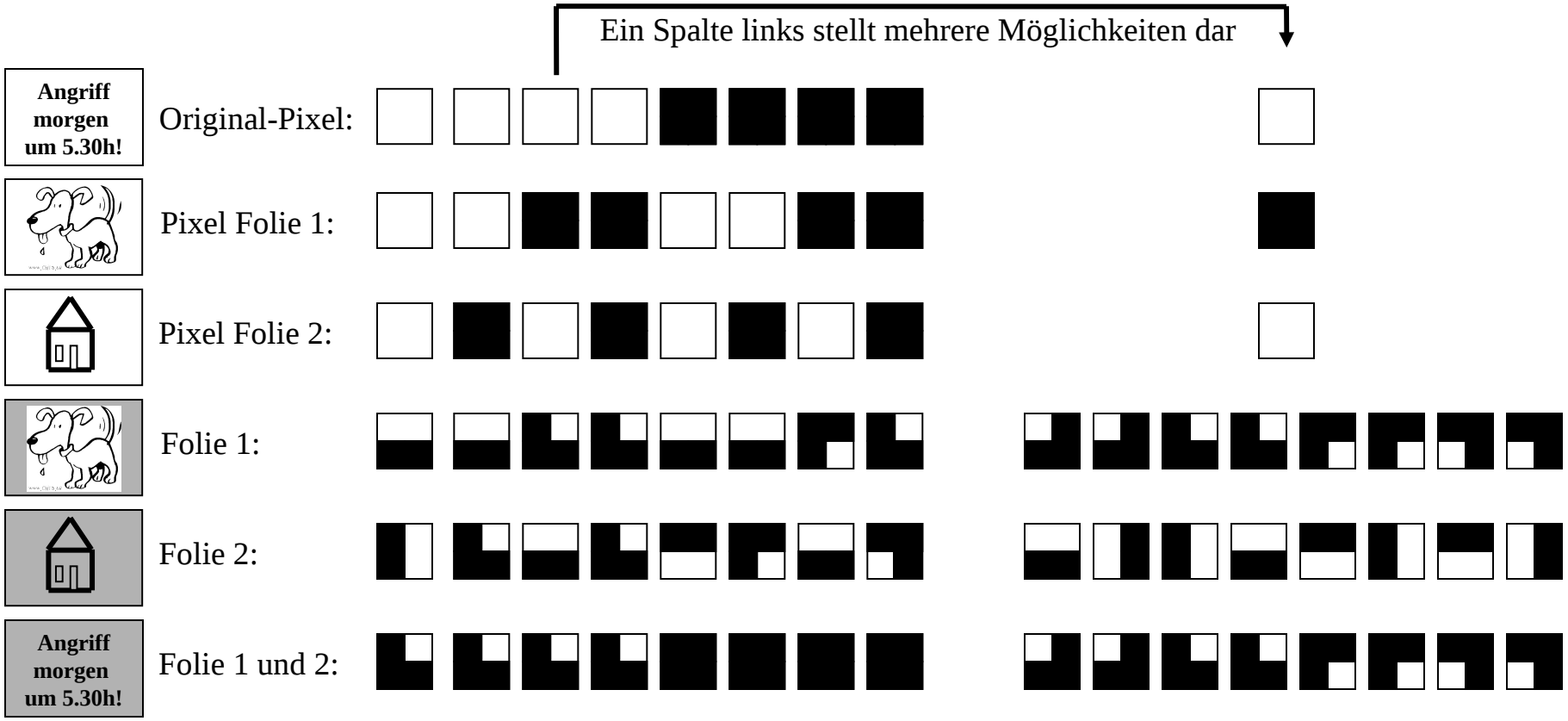
Bei der Steganographie geht es darum, in unauffälligen Nachrichten geheime Nachrichten zu verstecken.

Visuellen Kryptographie kann dazu verwendet werden:

Folie 1		Folie 2		Folie 1 und 2
	+		=	**Angriff morgen um 5.30h!**

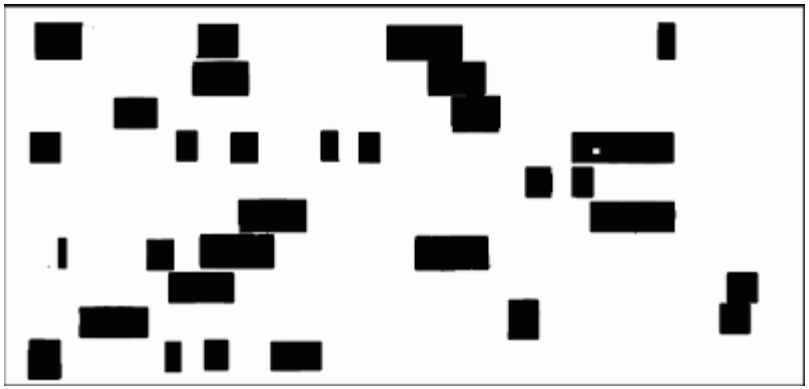
Konstruktion: Steganographie

Bei Folie 1 werden bei einem weißen Pixel zwei von vier Pixel zufällig ausgewählt (die diagonalen Kombinationen werden hier ausgelassen), bei einem schwarzen Pixel werden drei Pixel zufällig ausgewählt. Ebenso bei Folie 2, jedoch so, daß bei einem Original-Pixel 0 bei der Überdeckung ein weißes Pixel übrigbleibt, und bei einem Original-Pixel 1 kein weißes Feld übrig bleibt.



Weitere Beispiele Steganographie

Symmetrisches Geheimnis
(hier ein Lochkarte)



Geschickte Nachricht

CALL ME ANY TIME AT YOUR CONVENIENCE. WE CAN PLAN A DAY TO VISIT THE ZOO WHERE WE WILL SEE THESE ANIMALS – CROCODILES, ELEPHANTS, MONKEYS, AND CAMELS. AGE DOESN'T MATTER. ALL OF THE CHILDREN WILL HAVE FUN AND ENJOY THEMSELVES. WITH OUR DISCOUNT, THE COST WILL ONLY BE $5. WE CAN GET BURGERS TO EAT, THEN RETURN TO OUR HOMES AROUND 5:00. DOES THIS SOUND LIKE A GOOD IDEA?

Geheime Nachricht

```
CA        NY          OUR             E
          A D           THE
     SE                   CR
E       T   M    E S            S AGE
                          HI D
              D EN                WITH
 I      N   THE          L ON
         GER
   MES                       S           A
G       E  ?
```

Computergestützte Steganographie

Ein Originalbild

Das gleiche Bild, aber das unterste Rot-bit (RGB) jedes Pixels ist verändert. Kein Unterschied zu sehen.

Das Schwarzweiss-Bild, das sich aus den untersten Rot-bits ergibt.
(hier könnte ggfs. vorher noch eine Entschlüsselung stattfinden)

Hi Tobias,
bei den Verhandlungen mit Siemens nicht unter 70.000 gehen.
VG Kai