CARL VON OSSIETZKY universität OLDENBURG

Fakultät II: Informatik, Wirtschafts- und Rechtswissenschaften
Department für Informatik

Lineare und differentielle Kryptoanalyse von Substitutions-/Permutations-Netzen

Bachelorarbeit

Name:	Manuel Gieseking
Straße:	Ofener Straße 39
Wohnort:	26121 Oldenburg
E-Mail:	manuel.gieseking@informatik.uni-oldenburg.de

Studiengang:	Fach-Bachelor Informatik
Erstgutachterin:	PD Dr. Elke Wilkeit
Zweitgutachter:	Dr. Hans Fleischhack
Datum:	26. Dezember 2010

Inhaltsverzeichnis

1 Einleitung

Der schon seit Jahrhunderten bestehende unerbittliche Kampf zwischen Kryptographen und Kryptoanalytikern um geheime Informationen ist auch in unserem heutigen digitalen Zeitalter aktueller denn je. Wo die Kryptographen versuchen, Informationen auf jedwede Weise zu verstecken, arbeiten die Kryptoanalytiker an deren Entdeckung und Entschlüsselung.

Aktuell scheinen die Kryptographen einen geringen Vorsprung gegenüber den Kryptoanalytikern zu besitzen, da für einige Verfahren und unter Verwendung genügend großer Schlüssel die Rechenkapazität heutiger Computer nicht ausreicht, um mit den aktuell bekannten Algorithmen an die verschlüsselten Informationen zu gelangen. Dieser Vorsprung darf jedoch nicht zur Stagnation der kryptographischen Forschung führen, denn wie uns die Geschichte lehrte, kann das Vertrauen in eine nicht bewiesen sichere Verschlüsselung verheerende Folgen nach sich ziehen; sei es die Hinrichtung von Maria Stuart im Jahre 1587, die darauf vertraut hatte, dass ihre Briefe nicht entschlüsselt werden konnten [Sin08], oder der Eintritt der Amerikaner in den ersten Weltkrieg, wozu ebenfalls eine entschlüsselte Botschaft der Deutschen an die Mexikaner – das Zimmermann-Telegramm – ausschlaggebend gewesen ist [Sin07]. Diese Beispiele zeigen, welche bedeutende Rolle Kryptographie und Kryptoanalyse in der Geschichte gespielt haben und damit ist ersichtlich, welch hohe Relevanz diese Bereiche erst recht in unserem heutigen digitalen Informationszeitalter haben und haben werden.

Diese Arbeit befasst sich mit zwei kryptologischen Analyseverfahren – der linearen und der differentiellen Kryptoanalyse –, die für Angriffe auf iterierte Block-Chiffren, wozu auch der Data Encryption Standard (DES) und der heutzutage vielfach verwendete Advanced Encryption Standard (AES) gehören, genutzt werden können. Im Besonderen werden in dieser Arbeit Substitutions-/Permutations-Netze betrachtet, die spezielle iterierte Block-Chiffren darstellen, zu denen auch der AES gehört. Das Hauptaugenmerk bei dieser Betrachtung liegt auf einer einfachen und verständlichen Darstellung der beiden kryptoanalytischen Verfahren, für deren Verwirklichung ein Werkzeug zur Erstellung von kleinen und gut angreifbaren Substitutions-/Permutations-Netzen im Rahmen dieser Arbeit entwickelt wurde, welches in Kapitel „Implementierung“ vorgestellt wird. Dieses Werkzeug ermöglichte auch eine signifikante Verbesserung des in [Hey02] vorgestellten und in vielen Werken (zum Beispiel [Sti06]) referenzierten linearen Angriffs auf ein Beispiel-Substitutions-/Permutations-Netz (SPN).

Bevor die Analyseverfahren in den Kapiteln „Lineare Kryptoanalyse“ und „Differentielle Kryptoanalyse“ vorgestellt und jeweils anhand eines Beispiels verdeutlicht werden, wer-

den in dem Kapitel „Kryptographie“ die Verschlüsselungsverfahren und im Speziellen die SPNe und ihre Eigenschaften vorgestellt.

Die für die Analyseverfahren benötigten mathematischen Grundlagen und Zusammenhänge werden in gesammelter Form in dem Kapitel „Mathematischer Hintergrund“ dargestellt, bevor abschließend im Kapitel „Fazit und Ausblick“ ein Resümee über die Arbeit und ein Ausblick auf weitere Möglichkeiten der Kryptoanalyse, basierend auf den vorgestellten Verfahren, dargestellt werden.

2 Kryptographie

Die Kryptographie (griech.: kryptós – „verborgen“, gráphein – „schreiben“) ist diejenige Teilwissenschaft der Kryptologie (griech.: kryptós – „verborgen“, lógos – „Wort“,„Lehre“, „Sinn“), welche ausschließlich das Verschlüsseln von Informationen umfasst. Wo früher häufig beide Begriffe synonym verwendet wurden, wird heutzutage die Kryptologie, die Wissenschaft des Ver- *und* Entschlüsselns, in die Teilbereiche Kryptographie und Kryptoanalyse aufgeteilt, wobei letztere ausschließlich die Entschlüsselung von Informationen beinhaltet, auf die in Kapitel 4 „Kryptoanalyse“ weiter eingegangen wird [Wil08].

Die Motivation bezüglich des Verschlüsselns, also des Verbergens von Informationen einer Nachricht, ist darin begründet, dass zwei Personen, meist Alice und Bob genannt, ermöglicht werden soll, geheim zu kommunizieren, sodass kein Außenstehender, häufig als Eve (engl.: *eavesdropper* = *Lauscherin*) oder Mallet (engl.: *malicious* = *bösartig*, *drohend*) bezeichnet, eine Möglichkeit hat, an Informationen der Nachrichten zu gelangen. Ob diese Informationen Texte, Bilder, Videos oder irgendetwas anderes darstellen, ist für den Angriff meist irrelevant [Sti06].

Im Folgenden werden überblickend einige Möglichkeiten, mit denen versucht wird, die Geheimhaltung einer Nachricht zu gewährleisten, dargestellt. Insbesondere wird in Abschnitt 2.2.6 ein Verfahren zur Verschlüsselung von Daten vorgestellt, welches in Kapitel 4 angegriffen werden soll. Die dafür verwendete und die in der Arbeit benötigte Terminologie wird in Abschnitt 2.1 zuerst eingeführt, bevor in Abschnitt 2.2 „Substitutions-/Permutations-Netz“ abschließend die in dieser Arbeit betrachtete Chiffre vorgestellt wird.

2.1 Grundlegende Begrifflichkeiten

Unter dem **Verschlüsseln** (auch: **Chiffrieren**) einer Nachricht wird das Verändern eines Textes zum Zwecke des Verbergens von Informationen mit Hilfe eines Algorithmus (auch: **Verschlüsselungsfunktion**) und eines **Schlüssels**[1] verstanden. Die Nachricht, die bei diesem Vorgang verschlüsselt wurde, wird als **Klartext**[2] bezeichnet. Das Ziel dieses Verfahrens ist es, die Nachricht für Außenstehende zu verbergen und nur Eingeweihten zugänglich zu machen. Der daraus resultierende, veränderte Text wird auch als **Geheimtext**[3] bezeichnet. Der Vorgang, der sozusagen den umgekehrten Weg beschreibt, also einen Geheimtext mit Hilfe eines Algorithmus und eines Schlüssels wieder zu dem Klartext verändert, wird dementsprechend **entschlüsseln** (auch: **dechiffrieren**) genannt. Der Begriff

[1] Engl.: key.
[2] Engl.: plaintext.
[3] Engl.: ciphertext.

„Text“ bedeutet in diesem Zusammenhang nicht zwangsläufig einen sprachlich verständlichen Text, sondern bezeichnet jedwede Art von Daten, die verschlüsselt beziehungsweise entschlüsselt werden sollen.

Der bei diesem Verfahren verwendete **Schlüssel** stellt die Information dar, die benötigt wird, um den Text zu verbergen oder wieder zugänglich zu machen; dies kann im einfachen Fall zum Beispiel lediglich ein Passwort oder auch ein physischer Schlüssel sein, falls die Information zum Beispiel in eine Kiste gelegt wird, die mit einem Schloss verschlossen wird. Die für diese Arbeit relevante Art von Schlüsseln und deren Verwendung werden in Abschnitt 2.2.4 „Keyscheduling“ vorgestellt.

Ein Verfahren, welches diese beiden Vorgänge beinhaltet, wird auch kryptographisches System, kurz **Kryptosystem**, **Verschlüsselungsverfahren** oder auch **Chiffre** genannt. Ein solches System benötigt somit einige Formalia und Eigenschaften, die im Folgenden aufgeführt werden:

Definition 2.1 (Chiffre [Sti06]) *Sei $\mathcal{P}$ eine endliche Menge von möglichen Klartexten, $\mathcal{C}$ eine endliche Menge von möglichen Geheimtexten und $\mathcal{K}$ eine endliche Menge von möglichen Schlüsseln. Außerdem seien $\mathcal{E}$ und $\mathcal{D}$ Familien von Abbildungen, wobei mit $k, k' \in \mathcal{K}$ gilt:*

$$\forall\, e_k \in \mathcal{E}\ \exists\, d_{k'} \in \mathcal{D},\ \textit{mit}\ e_k : \mathcal{P} \to \mathcal{C}\ \textit{und}\ d_{k'} : \mathcal{C} \to \mathcal{P} :\ d_{k'}(e_k(x)) = x\ \textit{für alle}\ x \in \mathcal{P}.$$

*Das heißt, jeder verschlüsselte Text kann auch wieder eindeutig entschlüsselt werden. Dabei meinen die e_k beziehungsweise $d_{k'}$ Abbildungen, die mit Hilfe eines Schlüssels k beziehungsweise k' arbeiten. Damit ist e_k nur eine verkürzte Schreibweise für $e(k, \cdot)$, wobei $e : \mathcal{K} \times \mathcal{P} \to \mathcal{C}$ und e_k auch **Verschlüsselungsfunktion** und $d_{k'}$ dementsprechend **Entschlüsselungsfunktion** genannt werden. Mit der oben beschriebenen Eigenschaft ist e_k zwangsläufig eine injektive und $d_{k'}$ eine surjektive Funktion. Solch ein 5-Tupel $(\mathcal{P}, \mathcal{C}, \mathcal{K}, \mathcal{E}, \mathcal{D})$ wird als **Chiffre** oder auch Kryptosystem bezeichnet.* ∘

Klartexte und Geheimtexte werden meistens über ein **Alphabet** definiert, wobei mit Alphabet eine endliche Menge von Symbolen gemeint ist. Darunter fallen zum Beispiel unser gewöhnliches Alphabet, welches aus den einzelnen Buchstaben besteht, oder auch das Alphabet $\mathbb{Z}_2 = \{0, 1\}$, welches in dieser Arbeit nahezu ausschließlich verwandt wird.

Als **Zeichen** werden die einzelnen Elemente des Alphabets bezeichnet, sodass es in dem Alphabet $\mathbb{Z}_2$ lediglich die 0 und die 1 als Zeichen gibt. Werden nun einzelne Zeichen aneinandergefügt, dann wird von **Zeichenketten**, **Wörtern** oder auch **Blöcken** gesprochen, wobei ein einzelnes Zeichen ein Spezialfall eines Blockes ist, nämlich des Blockes mit der

Länge Eins. Die **Länge eines Blockes** wird als die Anzahl der Zeichen, die den Block bilden, definiert, wobei die Zeichen nicht zwangsläufig unterschiedlich sein müssen. Somit hat der Block 0000 die Länge vier, weil er aus vier Zeichen besteht, obwohl nur ein Symbol auftaucht. Wörter der Länge Null sind ebenfalls zugelassen und werden mit ε bezeichnet. Also ist die Länge eine Abbildung $len : A^n \to \mathbb{N},\ x_0 \dots x_{n-1} \mapsto n$, mit einem Alphabet A, $n \in \mathbb{N}$ und $A^0 = \{\varepsilon\}$.

Als Schreibweise wird vereinbart, dass die einzelnen Zeichen eines Wortes x der Länge $l \in \mathbb{N}$ mit x_0 bis x_{l-1} angesprochen werden. Auch wird das Aneinanderfügen von Zeichen wie auch von Wörtern (auch: **Konkatenation**) formal lediglich ohne Operator als Aneinanderschreiben definiert, sodass $x = x_0 x_1 \dots x_{l-2} x_{l-1}$ geschrieben werden kann. Verkürzend kann für ein Wort $x \in A^l$ mit einem Alphabet A lediglich a^l geschrieben werden, falls $\forall i \in \{0, \dots, l-1\} : x_i = a$, mit $a \in A$.

Die **Umkehrfunktion** (auch: **Inverse**) f^{-1} einer Funktion f bezeichnet diejenige Funktion, welche die Ausführung von f rückgängig macht, das heißt, es gilt $f^{-1}(f(x)) = x$ für alle x. Eine Funktion, für die eine Inverse existiert, wird **invertierbar** genannt.

Die **komponentenweise Addition** $\oplus$ zweier Wörter x, y der Länge $n \in \mathbb{N}$ über dem Alphabet $\mathbb{Z}_2$ (kurz: $x, y \in \mathbb{Z}_2^n$) wird in dieser Arbeit häufig verwandt und ist definiert als $x \oplus y := (x_0 + y_0 \ mod\ 2)(x_1 + y_1 \ mod\ 2) \ \dots \ (x_{n-1} + y_{n-1} \ mod\ 2)$. Wobei mit der dafür verwendeten Addition modulo 2, $\mathbb{Z}_2$ eine Gruppe ist und somit insbesondere $(x \oplus y) \in \mathbb{Z}_2^n$ gilt. Aus der Definition geht direkt hervor, dass die komponentenweise Addition zu sich selbst invers ist, was sich in vielen Kryptosystemen zunutze gemacht wird.

Eine Abbildung $f : \mathbb{Z}_2^{l_1} \to \mathbb{Z}_2^{l_2}$ $(l_1, l_2 \in \mathbb{N})$ heißt **linear** bezüglich der komponentenweisen Addition, falls $f(x \oplus y) = f(x) \oplus f(y)$ für $x, y \in \mathbb{Z}_2^{l_1}$ gilt. Dies ist die Definition der Linearität in Vektorräumen, jedoch kann die Homogenität als Bedingung weggelassen werden, da diese, wie im Folgenden zu sehen, durch die Additivität schon für alle Abbildungen erfüllt ist. Mit $\odot$ ist die komponentenweise Skalarmultiplikation, ganz analog zu der komponentenweise Addition, gemeint:

Sei $x \in \mathbb{Z}_2^{l_1}$ *und* $a \in \mathbb{Z}_2$:

Fall $a = 0$: $0 \odot f(x) = 0^{l_1} = f(0^{l_1}) \oplus f(0^{l_1}) = f(0^{l_1} \oplus 0^{l_1}) = f(0^{l_1}) = f(0 \odot x)$

Fall $a = 1$: $1 \odot f(x) = f(x) = f(1 \odot x)$.

2.2 Substitutions-/Permutations-Netz

Ein SPN ist eine iterierte Block-Chiffre, die zu den symmetrischen Verschlüsselungsverfahren gehört. Diese Begrifflichkeiten und die daraus resultierenden Eigenschaften werden

in Abschnitt 2.2.1 vorgestellt, bevor die einzelnen Bestandteile dieses Kryptosystems – die Substitution (Abschnitt 2.2.2), die Permutation (Abschnitt 2.2.3) und das Verfahren zum Erstellen der Rundenschlüssel (Abschnitt 2.2.4) – eingeführt werden. Abschließend wird in Abschnitt 2.2.5 eine formale Definition eines SPNes gegeben, und diese in Abschnitt 2.2.6 anhand eines Beispieles verdeutlicht.

2.2.1 Kategorisierung und Eigenschaften

Im Allgemeinen wird in der Kryptographie zwischen zwei Sorten von Verschlüsselungsverfahren unterschieden: zum einen den symmetrischen und zum anderen den asymmetrischen Chiffren[4], wobei letztere in dieser Arbeit nicht genauer behandelt werden, da sie nicht durch die in Kapitel 4 vorgestellten Verfahren gebrochen werden können.

Die **symmetrischen Chiffren** arbeiten mit einem geheimen (auch: privaten) Schlüssel (Private-Key-Verschlüsselung) zur Ver- und Entschlüsselung der Daten, welcher jedem der Kommunikationspartner bekannt sein und ansonsten geheim gehalten werden muss (siehe Abbildung 1). Bei den Angriffsverfahren in Kapitel 4 wird versucht, eben diesen Schlüssel beziehungsweise Teile dieses Schlüssels zu erlangen.

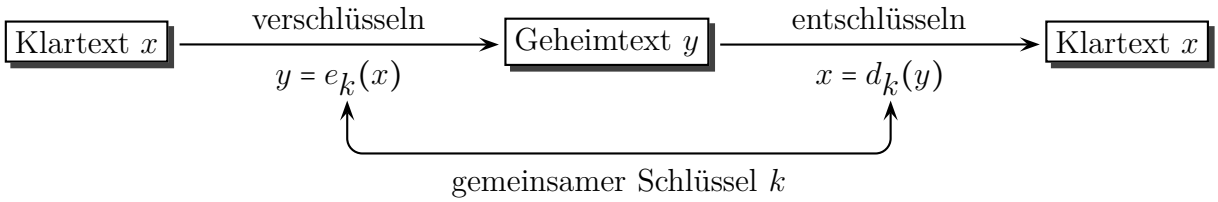

Abbildung 1: Symmetrische Chiffre [Wil08]

Eine **Block-Chiffre** ist ein symmetrisches Verschlüsselungsverfahren, welches vor der Verschlüsselung beziehungsweise Entschlüsselung der Daten diese zuerst in Blöcke mit fester Länge aufteilt:

> **Definition 2.2 (Block-Chiffre)** *Eine Chiffre Block* $= (\mathcal{P}, \mathcal{C}, \mathcal{K}, \mathcal{E}, \mathcal{D})$, *mit* $\mathcal{P} = \bigcup_{i=1}^{n} A^i$ $(n \in \mathbb{N} \setminus \{0\})$ *eine endliche Menge von Klartexten über einem Alphabet* A, $\mathcal{C} = \bigcup_{i=1}^{m} B^i$ $(m \in \mathbb{N} \setminus \{0\})$ *eine endliche Menge von Geheimtexten über einem Alphabet* B *und* $\mathcal{K}$ *eine endliche Menge von Schlüsseln, nennt sich* ***Block-Chiffre*** *mit der Eingangs-Blocklänge* $l_{in} \in \mathbb{N} \setminus \{0\}$ *und Ausgangs-Blocklänge* $l_{out} \in \mathbb{N} \setminus \{0\}$, *mit folgenden Funktionen* $(K \in \mathcal{K})$*:*

[4] Der bekannteste Vertreter für asymmetrische Verschlüsselung ist **RSA**, welcher im Jahre 1977 von Ronald L. **R**ivest, Adi **S**hamir und Leonard **A**dleman entwickelt wurde und auf der Idee der Public-Key-Kryptographie von Whitfield Diffie und Martin Hellman basiert [Wikc].

Padding: *Sei X ein Alphabet, $l \in \mathbb{N} \setminus \{0\}$ und $y_l : \bigcup_{i=1}^{l} X^i \to \mathbb{N},\ x \mapsto l - len(x) - 1$, dann:*

$$pad_l : \bigcup_{i=1}^{l} X^i \to X^l,\ x \mapsto \begin{cases} x & \text{für } len(x) = l \\ x1 \underbrace{0 \dots 0}_{y_l(x)-mal} & sonst \end{cases}.$$

Verschlüsseln: *Sei $E_K : A^{l_{in}} \to B^{l_{out}}$ eine Verschlüsselungsfunktion, dann verschlüsselt Block mit Hilfe von $e_K \in \mathcal{E}$:*

$$e_K : \mathcal{P} \to \mathcal{C},$$
$$p \mapsto E_K(p_0 \dots p_{l_{in}-1}) E_K(p_{l_{in}} \dots p_{2 \cdot l_{in}-1}) \dots E_K(pad_{l_{in}}(p_{len(p)-l_{in}} \dots p_{len(p)-1})).$$

Entschlüsseln: *Sei $D_K : B^{l_{out}} \to A^{l_{in}}$ die zu E_K gehörige Entschlüsselungsfunktion, dann entschlüsselt Block mit Hilfe von $d_K \in \mathcal{D}$:*

$$d_K : \mathcal{C} \to \mathcal{P}$$
$$c \mapsto D_K(c_0 \dots c_{l_{out}-1}) D_K(c_{l_{out}} \dots c_{2 \cdot l_{out}-1}) \dots$$
$$\dots D_K(pad_{l_{out}}(c_{len(c)-l_{out}} \dots c_{len(c)-1})).$$ ∘

Im Folgenden werden bei der Definition von Block-Chiffren lediglich die Vorschrift für einen Block angegeben und implizit diese Definition für die Verarbeitung mehrerer Blöcke verwendet.

Zwei besondere Block-Chiffren, die die Bestandteile eines SPNes bilden, sind zum einen die **S-Chiffre**, welche Zeichen oder ganze Blöcke ersetzt und auf die genauer in Abschnitt 2.2.2 eingegangen wird, und zum anderen die **Transpositions-Chiffre**, welche die Position der Zeichen verändert und in Abschnitt 2.2.3 erklärt wird. Beide Chiffren arbeiten jeden Block des Klartextes unabhängig voneinander und mit demselben Schlüssel ab. Häufig werden diese beiden Verfahren kombiniert, um die Komplexität des Kryptosystems zu erhöhen und damit die Kryptoanalyse zu erschweren [Sha49].

Eine Block-Chiffre nennt sich **iterierte Block-Chiffre**, wenn eine Verschlüsselungsfunktion mehrmals hintereinander auf den Klartext beziehungsweise auf die daraus resultierende Zeichenkette angewandt wird. Jede dieser Ausführungen wird als **Runde** und die in jeder Runde gleich bleibende Funktion als **Rundenfunktion** bezeichnet. Dabei wird in jeder dieser Ausführungen ein neuer, aus dem Schlüssel erstellter, **Rundenschlüssel** (Abschnitt 2.2.4) genutzt, um die Runde sicherer zu machen. Bei der Wahl der Rundenfunktion ist neben vielen weiteren Sicherheitsaspekten zu beachten, dass ein einfaches Hintereinanderausführen von Chiffren im Allgemeinen keine Verstärkung des Kryptosys-

tems bedeutet, sondern zum Beispiel erst das Zusammenspiel von Substitutionen und Transpositionen die Iteration zu einem starken Hilfsmittel macht [Sha49]. Eine iterierte Block-Chiffre, deren Rundenfunktion ausschließlich Substitutionen und Transpositionen beinhaltet, wird auch SPN genannt.

Eine formale Definition eines SPNes ist in Abschnitt 2.2.5 nachzulesen, wobei vorerst die eben erwähnten und für ein SPN benötigten Bestandteile, wie der geheime Schlüssel und ein Verfahren, um aus diesem Schlüssel Rundenschlüssel zu erstellen (Keyscheduling), wie auch die für die Rundenfunktion benötigte S- und die Transpositions-Chiffre, vorgestellt werden.

2.2.2 Substitution

Bei einer **Substitution** handelt es sich um ein Ersetzen von einzelnen Zeichen oder ganzen Blöcken eines Textes. Formal ist eine Substitution s eine Abbildung der Art:

$$s : A^l \to B^m,$$

wobei A und B Alphabete und $l, m \in \mathbb{N} \setminus \{0\}$ sind. Eine Folge von Substitutionen $S = (s_0, s_1, \ldots, s_{n-1})$ mit $n \in \mathbb{N} \setminus \{0\}$ zerlegt den Block $x \in A^L, L \in \mathbb{N}$ in n Teilworte $x_{<0>}, x_{<1>}, \ldots, x_{<n-1>}$. Die Länge l_i dieser $x_{<i>}$ $(0 \leq i < n)$ hängt von der Blockgröße ab, die die zugehörige Substitution s_i verarbeiten kann und es gilt $L = \sum_{i=0}^{n-1} l_i$. Daraufhin wird jedes dieser Teilworte mit der entsprechenden Substitution s_i $(0 \leq i < n)$ verarbeitet und die daraus resultierenden Wörter werden durch Konkatenation zu dem Geheimtext-Block $y \in B^M, M \in \mathbb{N}$ zusammengefügt (siehe Abbildung 2). Damit ist S eine Abbildung der Art $S : A^L \to B^M$ und ein Klartext $z \in A^\alpha$, $\alpha \in \mathbb{N}$ wird verschlüsselt, indem z in Blöcke der Länge L zerlegt wird, diese Blöcke mit S abgebildet werden und der Geheimtext durch Konkatenation der Bilder wieder zusammengesetzt wird. Mit invertierbaren Substitutionen s_i $(0 \leq i < n)$ ist S eine **S-Chiffre**:

Definition 2.3 (S-Chiffre) *Eine Block-Chiffre $S = (\mathcal{P}, \mathcal{C}, \mathcal{K}, \mathcal{E}, \mathcal{D})$, mit $\mathcal{P} = A^n$ $(n \in \mathbb{N} \setminus \{0\})$ eine endliche Menge von Klartexten über einem Alphabet A und $\mathcal{C} = B^m$ $(m \in \mathbb{N} \setminus \{0\})$ eine endliche Menge von Geheimtexten über einem Alphabet B, wird **S-Chiffre** genannt und besitzt folgende Eigenschaften:*

Schlüssel: *Sei $t \in \mathbb{N}$:*

$$\mathcal{K} = \bigcup_{t=0}^{n} \{(s_0, s_1, \ldots, s_t) \mid s_i : A^{l_{in_i}} \to B^{l_{out_i}}\ l_{in_i}, l_{out_i} \in \mathbb{N}\ (0 \leq i \leq t),\ \textit{invertierbare Substitution und}\ \sum_{i=0}^{t} l_{in_i} = n\ \textit{und}\ \sum_{i=0}^{t} l_{out_i} = m\}.$$

Sei $K \in \mathcal{K}$, mit $K = (s_0, s_1, \ldots, s_t)$ $(t \in \mathbb{N})$:

Verschlüsseln: *Sei $e_K \in \mathcal{E}$:*

$$e_K : A^n \to B^m,$$
$$p \mapsto s_0(p_0 \ldots p_{l_{in_0}-1}) s_1(p_{l_{in_0}} \ldots p_{l_{in_1}-1}) \ldots s_t(p_{l_{in_{t-1}}} \ldots p_{l_{in_t}-1}).$$

Entschlüsseln: *Sei $d_K \in \mathcal{C}$:*

$$d_K : B^m \to A^n,$$
$$c \mapsto s_0^{-1}(c_0 \ldots c_{l_{out_0}-1}) s_1^{-1}(c_{l_{out_0}} \ldots c_{l_{out_1}-1}) \ldots s_t^{-1}(c_{l_{out_{t-1}}} \ldots c_{l_{out_t}-1}). \quad \circ$$

Falls S lediglich aus einer Substitution s_0 besteht, dann wird S auch eine **Substitutions-Chiffre** genannt.

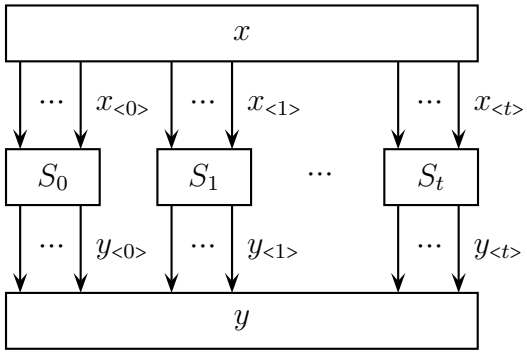

Abbildung 2: Substitutions-Chiffre S

Die einzelnen Substitionen s_i $(0 \leq i \leq t)$ werden im Folgenden auch **S-Boxen** genannt und mit S_i bezeichnet [Sti06]. Diese Arbeit konzentriert sich ausschließlich auf S-Boxen mit den Eigenschaften $A = B = \{0,1\} = \mathbb{Z}_2$ und $l, m > 1$. Substitutionen mit $l, m > 1$ werden auch **polygraphische Substitutionen** genannt [Koh04].

2.2.3 Transposition

Eine **Transpositions-Chiffre** verändert die Position der einzelnen Zeichen innerhalb einer Zeichenkette; dieser Vorgang wird auch als **Index-Permutation** bezeichnet. Dabei ist eine **Permutation** π eine bijektive Abbildung einer endlichen Menge in sich selbst:

$$\pi : \{0, \ldots, k-1\} \to \{0, \ldots, k-1\}$$

mit $k \in \mathbb{N} \setminus \{0\}$.

Mit Hilfe einer Permutation π verändert eine **Transposition** T die Indizes der einzelnen Zeichen eines Blockes $x \in A^k$ mit $k \in \mathbb{N} \setminus \{0\}$:

$$T : A^k \to A^k$$
$$x_0 x_1 \dots x_{k-1} \mapsto x_{\pi(0)} x_{\pi(1)} \dots x_{\pi(k-1)},$$

wobei x_i den i-ten Buchstaben der Zeichenkette x bezeichnet und A für ein Alphabet steht.

Eine Transpositions-Chiffre teilt einen Klartext in Blöcke mit einer festen Länge k auf und bildet diese daraufhin unabhängig voneinander mit derselben Transposition T ab. Der Geheimtext ist anschließend die Konkatenation der Bilder dieser Blöcke:

Definition 2.4 (Transpositions-Chiffre) *Eine Block-Chiffre $T = (\mathcal{P}, \mathcal{C}, \mathcal{K}, \mathcal{E}, \mathcal{D})$, mit $\mathcal{P} = \mathcal{C} = A^n$ $(n \in \mathbb{N} \setminus \{0\})$ eine endliche Menge von Klartexten beziehungsweise Geheimtexten über einem Alphabet A, wird* ***Transpositions-Chiffre*** *genannt und besitzt folgende Eigenschaften:*

Schlüssel: $\mathcal{K} = \{\pi : \{0, 1, \dots, n-1\} \to \{0, 1, \dots, n-1\} \mid \pi \text{ Permutation}\}$.

Sei $K \in \mathcal{K}$:

Verschlüsseln: *Sei $e_K \in \mathcal{E}$:*

$$e_K : A^n \to A^n,\ p \mapsto p_{K(0)} p_{K(1)} \dots p_{K(n-1)}.$$

Entschlüsseln: *Sei $d_K \in \mathcal{D}$:*

$$d_K : A^n \to A^n,\ p \mapsto p_{K^{-1}(0)} p_{K^{-1}(1)} \dots p_{K^{-1}(n-1)}.$$ ∘

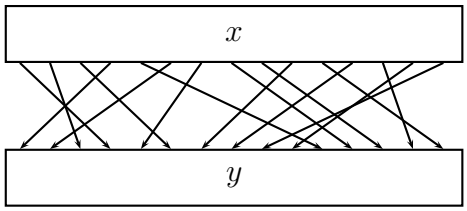

Abbildung 3: Transpositions-Chiffre T

Häufig werden in der Kryptologie die Begrifflichkeiten Permutation und Transposition synonym verwandt und wie in Abbildung 3 visualisiert.

2.2.4 Keyscheduling

Ein Verfahren κ, welches aus einem Schlüssel K – im Folgenden auch **Hauptschlüssel** genannt – mehrere unterschiedliche Schlüssel K_i $(0 \leq i < n)$ (auch: **Rundenschlüssel**) erstellt, wird **Keyscheduling** genannt und nun beispielhaft an einem sehr einfachen Verfahren in diesem Abschnitt vorgestellt. Also ist κ eine Abbildung der Art:

$$\kappa : K \to (K^0, \ldots, K^{n-1}).$$

Bei dem folgenden Keyscheduling handelt es sich um ein extrem schwaches und für die Praxis nicht empfehlenswertes Verfahren, welches jedoch leicht zu erklären und schnell zu implementieren ist, dabei die Idee solcher Verfahren gut vermittelt. Dieses Keyscheduling wurde auch in dem Werkzeug, welches in Kapitel 5 beschrieben wird, implementiert.

Im Folgenden wird angenommen, dass die Schlüssel, Klartexte und Geheimtexte binär codiert vorliegen, das heißt sie werden eindeutig durch eine Zeichenkette ausschließlich aus Nullen und Einsen bestehend repräsentiert, sodass nicht mehr zwischen dem Schlüssel und dessen Repräsentation unterschieden werden muss. Dieses Keyscheduling basiert auf der Idee, dass die Rundenschlüssel lediglich Teilzeichenketten des Hauptschlüssels sind und nacheinander mit einem Versatz $j \in \mathbb{N} \setminus \{0\}$ aus dem Hauptschlüssel entnommen werden:

Definition 2.5 (Simple-Keyscheduling) *Ein Verfahren*

$$\kappa_{simple} : K \to (K^0, \ldots, K^{n-1}),$$

mit

- *dem geheimen Hauptschlüssel* $K \in \mathbb{Z}_2^m$, $m \in \mathbb{N} \setminus \{0\}$,
- *der Anzahl der Rundenschlüssel* $n \in \mathbb{N} \setminus \{0\}$,
- *den* n *Rundenschlüsseln* $K^i \in \mathbb{Z}_2^l$ *für* $0 \leq i < n$ *und* $l \in \{1, 2, \ldots, m-1\}$,
- *dem Versatz* $j \in \{1, 2, \ldots, l-1\}$,

wobei $m \leq l + (n-1) \cdot j \leq m + \lfloor \frac{l}{2} \rfloor$, *erstellt mit* $x := (n-1) \cdot j + l$ *und*

$$K' := pad_x(K)$$

folgende n Rundenschlüssel:

$$K^i := K'_{i \cdot j} \dots K'_{i \cdot j + l - 1}, \text{ für } 0 \leq i < n$$

und wird im Folgenden ***Simple-Keyscheduling*** *genannt.* ∘

Die Einschränkung $m \leq l + (n-1) \cdot j$ bewirkt, dass jedes Zeichen des Hauptschlüssels in diesem Verfahren benutzt wird und mindestens in einem Rundenschlüssel auftaucht. Am besten ist selbstverständlich, wenn genau alle Zeichen des Hauptschlüssel verwendet werden, da dies jedoch eine sehr große Einschränkung wäre, ist die Grenze $l+(n-1)\cdot j \leq m+\lfloor \frac{l}{2} \rfloor$ gegeben, sodass maximal die Hälfte eines Rundenschlüssels aus Zeichen besteht, die nicht in dem Hauptschlüssel vorhanden sind. Im Normalfall wird das in diesem Verfahren auf den Schlüssel angewendete Padding auf Klartexte beziehungsweise Geheimtexte angewandt, falls die Blockgröße des Textes beim Ver- beziehungsweise Entschlüsseln nicht mit der Blockgröße, die die Chiffre verarbeiten kann, übereinstimmt. Bei einem Schlüssel macht solch ein Verfahren die Chiffre extrem unsicher, da direkt auf einige Schlüsselzeichen geschlossen werden kann; für das Werkzeug wurde jedoch ein Verfahren benötigt, welches sehr flexibel auf unterschiedliche SPNe angewandt werden kann und auf Grund dessen wurde auf diese sehr einfache Weise das Keyscheduling erweitert. Beispielhaft wird dieses Verfahren im Folgenden auf einen konkreten Hauptschlüssel angewandt, und die resultierenden Rundenschlüssel werden in Abschnitt 2.2.6 benutzt.

Beispiel 2.1 (Simple-Keyscheduling) *Der Versatz beträgt $j = 3$ und als geheimer Schlüssel wurde*

$$K = 011\ 110\ 111\ 101\ 110$$

gewählt. Damit ergeben sich mit $n = 4$ und κ_{simple} folgende Rundenschlüssel:

$$K^0 = 011\ 110$$
$$K^1 = 110\ 111$$
$$K^2 = 111\ 101$$
$$K^3 = 101\ 110.$$

Also anschaulich:

$$K = \underbrace{\underbrace{011\ 110}_{K^0}\ \underbrace{111}_{K^1}\ \underbrace{101}_{K^2}\ \underbrace{110}_{K^3}}.$$

*

2.2.5 Definition eines Substitutions-/Permutations-Netzes

In diesem Abschnitt werden die einzelnen Verfahren der letzten Abschnitte verbunden, und mit deren Hilfe wird ein SPN formal definiert. Hilfreich zum Verständnis der Definition ist die Abbildung 7 (S. 21), welche ein Beispiel-SPN visualisiert.

Es ist leicht nachzuvollziehen, dass für eine S-Chiffre S durch $S \circ S = S'$ wieder eine S-Chiffre S' erzeugt wird und ebenso gilt für eine Transpositions-Chiffre T, dass mit $T \circ T = T'$ die Abbildung T' erneut eine Transpositions-Chiffre darstellt. Somit wird durch ein mehrfaches Hintereinanderausführen einer dieser Chiffren keine Verstärkung des Kryptosystems erreicht [Sha49]. Die Kombination beider Chiffren und deren Hintereinanderausführung $((T \circ S) \circ (T \circ S))$ liefert jedoch im Allgemeinen keine Chiffre der Form $T' \circ S'$, sondern erstellt ein komplexeres Kryptosystem. Genau in dieser Eigenschaft, welche in Lemma 2.2 gezeigt wird, liegt die Stärke des im Folgenden eingeführten SPNes.

Definition 2.6 (SPN [Wil08]) *Ein 6-Tupel SPN=$(\mathbb{Z}_2^l, K, \kappa, S, T, n)$ mit*

- *der Anzahl der Runden $n \in \mathbb{N} \setminus \{0\}$,*
- *der Blocklänge $l \in \mathbb{N} \setminus \{0\}$,*
- *dem geheimen Hauptschlüssel $K \in \mathbb{Z}_2^q$ mit $q \in \mathbb{N} \setminus \{0\}$ und einem Keyscheduling $\kappa : K \to (K^0, \ldots, K^n)$ zum Erstellen der $n+1$ Rundenschlüssel $K^i \in \mathbb{Z}_2^l$ für $0 \leq i \leq n$,*
- *den S-Boxen $S_j : \mathbb{Z}_2^{l_{in_j}} \to \mathbb{Z}_2^{l_{out_j}}$, für $0 \leq j \leq r$, mit $r \in \mathbb{N}$, $l_{in_j}, l_{out_j} \in \mathbb{N} \setminus \{0\}$ und $\sum_{j=0}^{r} l_{in_j} = \sum_{j=0}^{r} l_{out_j} = l$,*
- *einer Folge von Substitutionen $S = (S_0, \ldots, S_r)$, mit $S : \mathbb{Z}_2^l \to \mathbb{Z}_2^l$,*
- *der Transposition $T : \mathbb{Z}_2^l \to \mathbb{Z}_2^l$,*

*wird **Substitutions-/Permutations-Netz** über dem Alphabet $\mathbb{Z}_2$ genannt, mit der Verschlüsselungsfunktion e_K:*

Verschlüsseln:

$$e_K = \oplus_{K^n} \circ \underbrace{S \circ \oplus_{K^{n-1}}}_{n.\ \text{Runde}} \circ \underbrace{T \circ S \circ \oplus_{K^{n-2}}}_{(n-1).\ \text{Runde}} \circ \cdots \circ \underbrace{T \circ S \circ \oplus_{K^1}}_{2.\ \text{Runde}} \circ \underbrace{T \circ S \circ \oplus_{K^0}}_{1.\ \text{Runde}},$$

*wobei $\circ$ die **Komposition von Abbildungen**, das heißt das Hintereinanderausführen von Funktionen bedeutet $((f \circ g)(x) := f(g(x)))$ und $\oplus_{K^i}$ definiert als $\oplus_{K^i}(x) :=$*

$x \oplus K^i$, also die Funktion, die zur Eingabe x komponentenweise den Schlüssel K^i addiert $(0 \leq i \leq n)$, bezeichnet. ∘

Auffallend ist, dass in der letzten Runde statt einer Transposition eine Kombination mit einem Rundenschlüssel durchgeführt wird. Der Grund hierfür ist, dass häufig die Substitutionen invertierbar gewählt werden und somit ohne die abschließende Kombination mit dem Rundenschlüssel ein Angreifer mit Kenntnis des Algorithmus mit Hilfe der inversen Funktion den Geheimtext zum Teil entschlüsseln kann und somit die letzte Runde hinfällig werden würde. Auch startet das SPN mit der Addition eines Rundenschlüssels, damit ein Angreifer ohne Kenntnis des Schlüssels keine partielle Verschlüsselung durchführen kann; dieses Einkesseln mit Schlüsseloperationen wird auch **whitening** genannt [Sti06]. Dass in der letzten Runde nicht einfach nur ein Rundenschlüssel addiert wurde, sondern die letzte Transposition auch noch weggefallen ist, ist darin begründet, dass es dadurch möglich wird, falls die S-Boxen auch invertierbar sind, mit einer Modifikation des Keyscheduling denselben Algorithmus zur Verschlüsselung, als auch zur Entschlüsselung zu nutzen. Bevor dieser Zusammenhang gezeigt wird, wird für den Fall, dass die S-Boxen invertierbar sind, die Art des Dechiffrierens vorgestellt, die von Hinten Schritt für Schritt mit der inversen Funktion jede Aktion rückgängig macht.

Da es sich, wie schon erwähnt, bei der Permutation um eine invertierbare Abbildung handelt, lässt sich die Transposition invertieren. Auch die komponentenweise Addition stellt offenkundig eine invertierbare Abbildung dar und ist sogar zu sich selbst invers. Ebenfalls werden die S-Boxen häufig invertierbar gewählt, sodass in diesem Fall die Entschlüsselungsfunktion wie folgt definiert werden kann.

Bemerkung 2.1 *Sei SPN=$(\mathbb{Z}_2^l, K, \kappa, S, T, n)$ ein Substitutions-/Permutations-Netz mit einer S-Chiffre S und Rundenschlüsseln $(K^0, \ldots, K^n)$, dann gilt für die Entschlüsselungsfunktion:*

Entschlüsseln:

$$\begin{aligned} d_K = \oplus_{K^0} \circ S^{-1} \circ T^{-1} \circ \oplus_{K^1} \circ S^{-1} \circ T^{-1} \circ \cdots \\ \cdots \circ \oplus_{K^{n-2}} \circ S^{-1} \circ T^{-1} \circ \oplus_{K^{n-1}} \circ S^{-1} \circ \oplus_{K^n}, \end{aligned} \tag{1}$$

wobei S^{-1} und T^{-1} die Umkehrfunktionen von S beziehungsweise T bezeichnen. Damit sich die Schlüssel gegenseitig wieder auflösen, müssen sie in umgekehrter Reihenfolge benutzt werden. Damit ist ein SPN eine Chiffre im Sinne der Definition 2.1. ♣

Weil es für die Implementierung von Chiffren sowohl in Hardware als auch in Software vorteilhaft ist, lediglich einen Algorithmus zum Ver- und Entschlüsseln benutzen zu können, wird diese Möglichkeit im Falle invertierbarer S-Boxen im Folgenden dargestellt.

Lemma 2.1 *Sei SPN=$(\mathbb{Z}_2^l, K, \kappa, S, T, n)$ ein Substitutions-/Permutations-Netz mit einer S-Chiffre S, Rundenschlüsseln $(K^0, \ldots, K^n)$ und Verschlüsselungsfunktion e_K. Des Weiteren seien $x, y \in \mathbb{Z}_2^l$, sodass $e_K(x) = y$. Dann lässt sich mit e_K der Geheimtext entschlüsseln, wenn für die Funktionen S und T die inversen Funktionen S^{-1} und T^{-1} und für die Rundenschlüssel statt $(K^0, \ldots, K^n)$ die Rundenschlüssel $(K^n, T^{-1}(K^{n-1}), \ldots, T^{-1}(K^1), K^0)$ gewählt werden. Also gilt:*

$$\begin{aligned} x = &(\oplus_{K^0} \circ S^{-1} \circ \oplus_{T^{-1}(K^1)} \circ T^{-1} \circ S^{-1} \circ \oplus_{T^{-1}(K^2)} \circ \cdots \\ &\cdots \circ T^{-1} \circ S^{-1} \circ \oplus_{T^{-1}(K^{n-1})} \circ T^{-1} \circ S^{-1} \circ \oplus_{K^n})(y). \end{aligned}$$

Das heißt, das SPN kann mit demselben Algorithmus ver- und entschlüsseln und damit kann die Funktion d_K aus Bemerkung 2.1 ersetzt werden, da derselbe Effekt bereits mit der Verschlüsselungsfunktion e_K, nur mit unterschiedlichen Parametern, erreicht werden kann. ∘

Beweis:
Sei SPN=$(\mathbb{Z}_2^l, K, \kappa, S, T, n)$ ein Substitutions-/Permutations-Netz mit der Verschlüsselungsfunktion e_K, S eine S-Chiffre und den Rundenschlüsseln $(K^0, \ldots, K^n)$. Seien weiterhin $x, y \in \mathbb{Z}_2^l$ mit $e_K(x) = y$, dann ist also zu zeigen:

$$\begin{aligned} x = &(\oplus_{K^0} \circ S^{-1} \circ \oplus_{T^{-1}(K^1)} \circ T^{-1} \circ S^{-1} \circ \oplus_{T^{-1}(K^2)} \circ \cdots \\ &\cdots \circ T^{-1} \circ S^{-1} \circ \oplus_{T^{-1}(K^{n-1})} \circ T^{-1} \circ S^{-1} \circ \oplus_{K^n})(y). \end{aligned}$$

Wir wissen $x = d_K(y)$ mit der Entschlüsselungsfunktion d_K aus der Bemerkung 2.1, darum genügt es zu zeigen, dass:

$$\begin{aligned} d_K(y) = &(\oplus_{K^0} \circ S^{-1} \circ \oplus_{T^{-1}(K^1)} \circ T^{-1} \circ S^{-1} \circ \oplus_{T^{-1}(K^2)} \circ \cdots \\ &\cdots \circ T^{-1} \circ S^{-1} \circ \oplus_{T^{-1}(K^{n-1})} \circ T^{-1} \circ S^{-1} \circ \oplus_{K^n})(y). \end{aligned} \tag{2}$$

Werden nun diese beiden Terme miteinander verglichen, fällt direkt auf, dass sie sich lediglich pro Runde an einer Stelle unterscheiden; wo es $T^{-1} \circ \oplus_{K^i}$ in der Definition von d_K (Gleichung (1)) heißt, wird in Gleichung (2) die Abbildung $\oplus_{T^{-1}(K^i)} \circ T^{-1}$ verwendet $(0 < i < n)$. Also bleibt nur noch zu zeigen:

$$(T^{-1} \circ \oplus_{K^i})(a) = (\oplus_{T^{-1}(K^i)} \circ T^{-1})(a)$$

für alle $a \in \mathbb{Z}_2^l$ und für $(0 < i < n)$. Durch geringfügige Umformung der Gleichung fällt auf, dass sie lediglich von der Linearität von T bezüglich der komponentenweisen Addition

abhängt:

$$\begin{aligned} &\textit{Sei } a \in \mathbb{Z}_2^l: \\ &\qquad (T^{-1} \circ \oplus_{K^i})(a) \quad = \quad T^{-1}(\oplus_{K^i}(a)) = T^{-1}(a \oplus K^i) \\ &\qquad\qquad\qquad\quad \overset{lin.}{=} \quad T^{-1}(a) \oplus T^{-1}(K^i) \\ &\qquad\qquad\qquad\quad = \quad (\oplus_{T^{-1}(K^i)} \circ T^{-1})(a). \end{aligned} \tag{3}$$

Dass aber eine Transposition T immer linear bezüglich der komponentenweisen Addition ist, ist offensichtlich, da, wenn π die Permutation ist, die T benutzt, dann gilt mit Hilfe der Definitionen von T und $\oplus$ für $a, b \in \mathbb{Z}_2^l$:

$$\begin{aligned} T((a_0 \dots a_{l-1}) \oplus (b_0 \dots b_{l-1})) &\overset{Def.\oplus}{=} T((a_0 \oplus b_0) \dots (a_{l-1} \oplus b_{l-1})) \\ &\overset{Def.T}{=} (a_{\pi(0)} \oplus b_{\pi(0)}) \dots (a_{\pi(l-1)} \oplus b_{\pi(l-1)}) \\ &\overset{Def.\oplus}{=} (a_{\pi(0)} \dots a_{\pi(l-1)}) \oplus (b_{\pi(0)} \dots b_{\pi(l-1)}) \\ &\overset{Def.T}{=} T(a_0 \dots a_{l-1}) \oplus T(b_0 \dots b_{l-1}). \end{aligned} \tag{4}$$

Damit ist zum einen gezeigt, dass eine Transposition linear bezüglich der komponentenweisen Addition ist, aber zum anderen ist mit dieser Linearität und Gleichung (3) auch die Behauptung bewiesen. ■

In der Kryptographie stellt die Linearität von Chiffren im Allgemeinen keine wünschenswerte Eigenschaft dar, denn, wären zusätzlich zu den Transpositionen auch die S-Boxen linear, so ließe sich der Rundenschlüssel K jeder Runde wie folgt berechnen:

$$\begin{aligned} T(S(x \oplus K)) = y &\overset{lin.}{\rightsquigarrow} T(S(x)) \oplus T(S(K)) = y \\ &\overset{T(S(x))\oplus}{\rightsquigarrow} T(S(K)) = T(S(x)) \oplus y \\ &\overset{T^{-1},S^{-1}}{\rightsquigarrow} K = S^{-1}(T^{-1}(T(S(x)) \oplus y)) \\ &\overset{lin.}{\rightsquigarrow} K = x \oplus S^{-1}(T^{-1}(y) \end{aligned} \tag{5}$$

und damit wäre die Chiffre sehr leicht angreifbar. Da es sich bei S-Boxen um Ersetzungen handelt, ist leicht nachzuvollziehen (Beispiel 2.2), dass sich eine S-Box $s : \mathbb{Z}_2^l \to \mathbb{Z}_2^l$, mit $l \in \mathbb{N} \setminus \{0\}$ in der Art wählen lässt, dass die Gleichung $s(x \oplus y) = s(x) \oplus s(y)$ mit $x, y \in \mathbb{Z}_2^l$ nicht erfüllt und somit die S-Box nicht linear ist. Aufgrund dessen wird in Abschnitt 4.2 „Lineare Kryptoanalyse“ versucht, sich den S-Boxen linear anzunähern, um eine ähnliche Gleichung wie die obige für die Schlüssel zu ermitteln.

Beispiel 2.2 *Sei*

$$\begin{aligned} s : \mathbb{Z}_2^2 &\to \mathbb{Z}_2^2 \\ 00 &\mapsto 01 \\ 01 &\mapsto 10 \\ 10 &\mapsto 11 \\ 11 &\mapsto 00, \end{aligned}$$

dann gilt

$$s(00 \oplus 01) = s(01) = 10 \neq 11 = 01 \oplus 10 = s(00) \oplus s(01).$$

Also ist dieses Minimalbeispiel ein Nachweis dafür, dass S-Boxen im Allgemeinen nicht linear sind. ∗

Eine weitere Eigenschaft der S-Boxen, die eine Kryptoanalyse erschwert, findet sich in deren Umgang mit dem Klartext. Denn ebenso leicht lässt sich eine S-Box finden, die allein durch das Verändern eines Zeichen des Eingangstextes, die Veränderung von einem Großteil bis hin zu allen Zeichen des Ausgabetextes bewirkt. Schon die S-Box aus Beispiel 2.2 bewirkt durch Veränderung eines Eingabezeichens die Veränderung von mindestens 50% der Ausgabezeichen. Dies liegt erneut darin begründet, dass es sich bei S-Boxen um Ersetzungen handelt.

Auch wenn eine Folge von Substitutionen sich wieder mit einer einzigen Substitution darstellen lässt, hat die Aufteilung in mehrere kleine Substitutionen den Vorteil, dass sie einen geringeren Speicherbedarf benötigt als die Speicherung einer einzigen großen Substitution [Sti06]. Häufig werden die S-Boxen in der Software als Tabelle – wie in Abbildung 5(a) (S. 20) visualisiert – implementiert, sodass die S-Box aus Beispiel 2.2 zur Speicherung $2 \cdot 2^2$ Bits benötigt, also werden für vier unterschiedliche S-Boxen dieser Größe 2^5 Bits Speicher verbraucht. Eine einzige S-Box, die diese vier S-Boxen vereint, benötigt jedoch $8 \cdot 2^8 = 2^{11}$ Bits Speicher.

Auch wenn die Transposition, wie in Gleichung (5) gesehen, auf Grund ihrer Linearität eine Schwäche aufweist, hat eine gut gewählte Transposition den Vorteil, dass sie die Zeichen eines Textes sehr gut über die Blockgrenzen der einzelnen S-Boxen hinaus verteilt und damit Muster des Klartextes weit über den Geheimtext verstreut. Dies hat den Effekt, dass die Struktur des Klartext soweit durcheinander gebracht wird, dass eine statistische Analyse erschwert wird. Ist dies der Fall, so wird auch, nach dem von Claude Shannon eingeführten Gütekriterium für Chiffren, von **Diffusion** gesprochen [Sha49].

Wie eingangs erwähnt, sollte ein SPN die Eigenschaft haben, dass die Komplexität der

Chiffre mit jeder Runde steigt; dass dies möglich ist, wird durch folgendes Lemma sichergestellt:

Lemma 2.2 *Sei* $3 < l \in \mathbb{N} \setminus \{0\}$ *und* $n \in \mathbb{N}$*, dann existieren eine Folge von Substitutionen* $S = (S_0, \ldots, S_n)$ *mit* $S : \mathbb{Z}_2^l \to \mathbb{Z}_2^l$ *und* $T : \mathbb{Z}_2^l \to \mathbb{Z}_2^l$ *Transposition, sodass es keine Folge von Substitutionen* $S' = (S'_0, \ldots, S'_n)$ *gibt, mit* $S' : \mathbb{Z}_2^l \to \mathbb{Z}_2^l$ *deren einzelnen* S'_i *dieselbe Eingabe- und Ausgabelänge wie auch die* S_i *besitzen* $(0 \leq i \leq n)$ *und keine Transposition* $T' : \mathbb{Z}_2^l \to \mathbb{Z}_2^l$ *gibt, sodass* $\forall m \in \mathbb{Z}_2^l$*:*

$$(T \circ S \circ T \circ S)(m) = (T' \circ S')(m). \tag{6}$$

∘

Beweis:
Fall $l = 4$*:* *Sei* $S = (S_1, S_2)$ *eine Folge von Substitutionen mit*

$$\begin{aligned} S_1 : \mathbb{Z}_2^2 &\to \mathbb{Z}_2^2 \\ 00 &\mapsto 00 \\ 01 &\mapsto 10 \\ 10 &\mapsto 01 \\ 11 &\mapsto 11 \end{aligned} \qquad \begin{aligned} S_2 : \mathbb{Z}_2^2 &\to \mathbb{Z}_2^2 \\ 00 &\mapsto 00 \\ 01 &\mapsto 11 \\ 10 &\mapsto 01 \\ 11 &\mapsto 10 \end{aligned}$$

und sei weiterhin $T : \mathbb{Z}_2^4 \to \mathbb{Z}_2^4$*,* $x_0x_1x_2x_3 \mapsto x_3x_0x_1x_2$ *eine Transposition. Dann bildet* $T \circ S \circ T \circ S$ *wie in folgender Tabelle beschrieben einige Klartexte* $m \in \mathbb{Z}_2^4$ *ab:*

idx	m	$m_1 = S(m)$	$m_2 = T(m_1)$	$m_3 = S(m_2)$	$m_4 = T(m_3)$
0	0000	0000	0000	0000	0000
1	0011	0010	0001	0011	1001
2	1111	1110	0111	1010	0101
3	1100	1100	0110	1001	1100

Abbildung 4: Abbildung einiger Klartexte mit $T \circ S \circ T \circ S$

Annahme: *Es existieren* S' *und* T' *mit* $S' = (S'_1, S'_2)$ *eine Folge von Substitutionen mit* $S'_1 : \mathbb{Z}_2^2 \to \mathbb{Z}_2^2$ *und* $S'_2 : \mathbb{Z}_2^2 \to \mathbb{Z}_2^2$ *und* $T' : \mathbb{Z}_2^4 \to \mathbb{Z}_2^4$ *eine Transposition, sodass* $\forall m \in \mathbb{Z}_2^4$*:*

$$(T \circ S \circ T \circ S)(m) = (T' \circ S')(m).$$

Da die Transposition T' *lediglich die Zeichen permutiert, lassen sich aus Abbildung 4 direkt einige Voraussetzungen für* S'_1 *und* S'_2 *ableiten:*

Aus Zeile 0 *folgt* $S_1'(00) = 00$ *und* $S_2'(00) = 00$ *und damit folgt aus Zeile* 1*, da die beiden Nullen der letzten Spalte schon durch* S_1' *erzeugt worden sind, dass* $S_2'(11) = 11$*. Ebenso werden in Zeile* 2 *die Einsen der letzten Spalte schon von* S_2' *erzeugt und somit gilt* $S_1'(11) = 00$*. Damit folgt aus Zeile* 3*, da die Nullen von* S_1' *erzeugt wurden, dass* $S_2'(00) = 11$ *sein muss. Zusammen soll also gelten, dass* $S_2'(00) = 00$ *und* $S_2'(00) = 11$*, dies ist aber ein Widerspruch, da damit* S_2' *keine Abbildung wäre und damit auch keine S-Box. Somit ist die Annahme falsch und die Behauptung für den Fall* $l = 4$ *bewiesen.*

Fall $l > 4$*: Sei* $S = (S_1, S_2, S_3)$ *eine Folge von Substitutionen mit* S_1, S_2 *wie im Fall* $l = 4$ *und* $S_3 : \mathbb{Z}_2^{l-4} \to \mathbb{Z}_2^{l-4}$ *die Identitätsabbildung. Sei ferner* $T : \mathbb{Z}_2^l \to \mathbb{Z}_2^l$*,* $x_0 \cdots x_{l-1} \mapsto x_3 x_0 x_1 x_2 x_4 \cdots x_{l-1}$ *eine Transposition. Damit lässt sich mit den mit Nullen erweiterten Klartexten und analogen Argumenten des Falles* $l = 4$ *der Widerspruch erneut herleiten und somit die Behauptung beweisen.* ■

Auch wenn gerade für den Beweis des Falles $l > 4$ starke Restriktionen an die Transposition und Substitution gestellt wurden, sind diese ausschließlich eine Notwendigkeit für den allgemeinen Beweis; im Speziellen lassen sich für jede beliebige Blocklänge direkt nahezu beliebige Beispiele finden, welche die geforderte Eigenschaft erfüllen.

Die Notwendigkeit dieser gerade bewiesenen Eigenschaft von S und T liegt darin begründet, dass, wenn sich S' und T' finden ließen, sodass die Gleichung (6) erfüllt ist, abgesehen von der Schlüsseladdition, immer jeweils zwei Runden des SPNes auf eine Runde heruntergebrochen werden könnten und damit das SPN per Induktion mit einer einzigen Runde darstellbar wäre und damit eine Komplexitätssteigerung mit jeder weiteren Runde des SPNes lediglich an der Schlüsseladdition hängen würde.

2.2.6 Beispiel eines Substitutions-/Permutations-Netzes

Dieser Abschnitt befasst sich mit einem konkreten SPN und verdeutlicht die in den vorherigen Abschnitten vorgestellten allgemeinen Verfahren. Ebenso wird dieses Netz für die Analyseverfahren in Kapitel 4 verwendet.

Das Werkzeug, welches in Kapitel 5 vorgestellt wird, liefert beliebige SPNe; dass dieses SPN ausgewählt wurde, liegt vor allem an dessen Größe und relativ guter Angreifbarkeit. Es sollte möglichst klein sein, insbesondere kleiner als das häufige zitierte Beispiel aus [Hey02], sodass die Rechnungen für eine bessere Nachvollziehbarkeit auch noch gut per Hand durchgeführt werden können, dabei jedoch groß genug, damit die Analyseverfahren den kompletten letzten Rundenschlüssel zumindest in einer minimalen Auswahl liefern. Ein etwas größeres Netz ist das gerade schon erwähnte Beispiel aus [Hey02], welches

auf Grund der von mir gefundenen Verbesserung des linearen Angriffs in Kapitel 5 kurz aufgegriffen wird.

Beispiel 2.3 (SPN) *Betrachte das SPN:*

$$SPN_{bsp} = (\mathbb{Z}_2^6, K, \kappa_{simple}, S = (S_1, S_2), T, 3)$$

mit den einzelnen Substitutionen $S_1 : \mathbb{Z}_2^3 \to \mathbb{Z}_2^3$ *und* $S_2 : \mathbb{Z}_2^3 \to \mathbb{Z}_2^3$*, die durch Abbildung 5(a) beziehungsweise Abbildung 5(b) definiert werden. Dabei ist um Platz zu sparen und die Übersicht zu erhöhen eine dezimalcodierte Schreibweise gewählt worden, das bedeutet:* $000 \mapsto 0, 001 \mapsto 1, \ldots, 110 \mapsto 6, 111 \mapsto 7$.

z	0	1	2	3	4	5	6	7
$S_1(z)$	2	5	0	4	6	3	1	7

(a) S-Box S_1

z	0	1	2	3	4	5	6	7
$S_2(z)$	2	5	4	6	1	0	7	3

(b) S-Box S_2

Abbildung 5: Die Substitution S von SPN_{bsp}

Des Weiteren sei die Transposition T durch die Index-Permutation π *definiert, welche in Abbildung 6 angegeben ist.*

z	1	2	3	4	5	6
$\pi(z)$	5	1	2	4	6	3

Abbildung 6: Die Permutation π von SPN_{bsp}

und sei schlussendlich der geheime Hauptschlüssel

$$K = 011\ 110\ 111\ 101\ 110$$

und damit derselbe, der auch schon im Beispiel 2.1 mit dem Simple-Keyscheduling betrachtet wurde. Im Folgenden werden dieselben Parameter wie in dem gerade schon erwähnten Beispiel für dieses Verfahren gewählt, sodass die Rundenschlüssel erneut

$$K^0 = 011\ 110$$
$$K^1 = 110\ 111$$
$$K^2 = 111\ 101$$
$$K^3 = 101\ 110$$

sind.

Die Verschlüsselungsfunktion ergibt sich damit auf Grund der Definition 2.6 zu:

Verschlüsseln:

$$e_K = \oplus_{K^3} \circ \underbrace{S \circ \oplus_{K^2}}_{3.\ Runde} \circ \underbrace{T \circ S \circ \oplus_{K^1}}_{2.\ Runde} \circ \underbrace{T \circ S \circ \oplus_{K^0}}_{1.\ Runde}.$$

Da die S-Boxen in diesem Beispiel alle invertierbar sind, lässt sich beispielsweise mit Bemerkung 2.1 entschlüsseln:

Entschlüsseln:

$$d_K = \oplus_{K^0} \circ S^{-1} \circ T^{-1} \circ \oplus_{K^1} \circ S^{-1} \circ T^{-1} \circ \oplus_{K^2} \circ S^{-1} \circ \oplus_{K^3}$$

und damit ist das Beispiel-SPN vollständig definiert. *

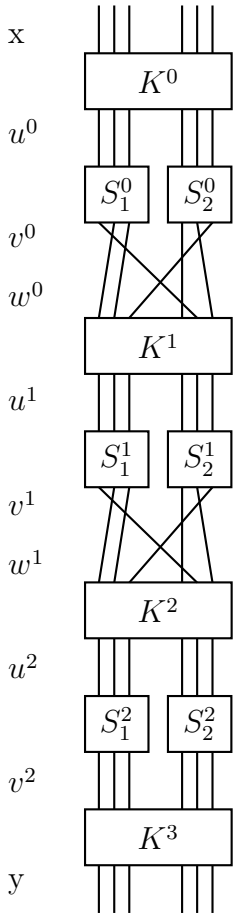

Abbildung 7: Visualisierung von SPN_{bsp}

Am einfachsten lassen sich solche Verfahren anhand einer Graphik nachvollziehen; für dieses Beispiel wurde dies in der Abbildung 7 getan. Des Weiteren wird abschließend ein Klartext mit diesem SPN verschlüsselt, um die genaue Vorgehensweise beispielhaft darzustellen. Die Bezeichnungen der einzelnen Zeilen orientieren sich an Abbildung 7.

Beispiel 2.4 (Verschlüsseln) *Sei $x \coloneqq 001\ 100$ der mit SPN_{bsp} zu verschlüsselnde Klartext, dann ergeben sich, mit den Bezeichnungen angelehnt an Abbildung 7, folgende Werte für die Verschlüsselung:*

$$\begin{aligned}
x &= 001\ 100 \\
K^0 &= 011\ 110 \\
u^0 = x \oplus K^0 &= 010\ 010 \\
v^0 = S(u^0) &= 000\ 100 \\
w^0 = T(v^0) &= 000\ 100 \\
K^1 &= 110\ 111 \\
u^1 = w^0 \oplus K^1 &= 110\ 011 \\
v^1 = S(u^1) &= 001\ 110 \\
w^1 = T(v^1) &= 010\ 101 \\
K^2 &= 111\ 101 \\
u^2 = w^1 \oplus K^2 &= 101\ 000 \\
v^2 = S(u^2) &= 011\ 010 \\
K^3 &= 101\ 110.
\end{aligned}$$

Damit ergibt sich

$$y = v^2 \oplus K^3 = 110\ 100$$

als Geheimtext. *

3 Mathematischer Hintergrund

In diesem Kapitel werden die mathematischen Grundlagen vorgestellt, die für das Verständnis des Kapitels 4 „Kryptoanalyse“ notwendig sind. Dabei sind die Beweise dieses Kapitels lediglich aus Interesse und der Vollständigkeit halber aufgeführt, sind jedoch zum Verständnis der Analyseverfahren des Kapitels 4 nicht von besonderer Bedeutung.

Die stochastischen Inhalte dieses Kapitels können zum Beispiel in [BH08] nachgelesen werden, ebenso sind die algebraischen Inhalte zum Beispiel wiederzufinden in [Bos05].

Definition 3.1 (Wahrscheinlichkeitsmaß, Wahrscheinlichkeitsraum) *Ein **Wahrscheinlichkeitsmaß** auf einer Menge Ω ist eine Mengenfunktion:*

$$P : \mathscr{P}(\Omega) \to [0,1]$$

mit

(1) $P[\Omega] = 1$

(2) $P[\biguplus_{i\in I} A_i] = \sum_{i\in I} P[A_i]$ *für* $(A_i)_{i\in I}$ *und* I *Indexmenge.*

Wobei mit $\mathscr{P}$ die Potenzmenge, also die Menge aller Teilmengen ($\mathscr{P}(X) := \{U \mid U \subseteq X\}$) gemeint ist und $\biguplus$ die Vereinigung paarweise disjunkter Mengen ($A, B \subseteq \Omega$ disjunkt $:\Leftrightarrow A \cap B = \varnothing$ wobei $\varnothing$ die leere Menge darstellt) bezeichnet.

*P wird auch vereinfacht nur **Wahrscheinlichkeit** genannt und das Paar (Ω, P) als **Wahrscheinlichkeitsraum** bezeichnet. Im Falle, dass Ω eine höchstens abzählbare Menge ist, nennt sich (Ω, P) **diskreter Wahrscheinlichkeitsraum**.*

*Falls $A \subseteq \Omega$, dann wird A als **Ereignis** und $a \in A$ als **Elementarereignis** bezeichnet und vereinfacht wird für alle Elementarereignisse $a \in A \subseteq \Omega$ die Wahrscheinlichkeit $P[a] := P[\{a\}]$ definiert.* ∘

Diese Definitionen ermöglichen es, weitere elementare Eigenschaften einzuführen.

Definition 3.2 (unabhängige Ereignisse, bedingte Wahrscheinlichkeit) *Sei (Ω, P) ein Wahrscheinlichkeitsraum.*

*Zwei Ereignisse $A, B \subseteq \Omega$ heißen (stochastisch) **unabhängig** genau dann wenn:*

$$P[A \cap B] = P[A] \cdot P[B]$$

gilt. Im Fall $P[B] > 0$ definiere

$$P[A \mid B] := \frac{P[A \cap B]}{P[B]}$$

*als **bedingte Wahrscheinlichkeit** (sprich: die Wahrscheinlichkeit von A unter der Bedingung B).*

Damit gilt offensichtlich für $P[B] > 0$:

$$A, B \subseteq \Omega \text{ unabhängig } \rightsquigarrow P[A \mid B] = P[A]. \qquad \circ$$

Auf diesen Grundlagen lässt sich der Begriff der Zufallsvariablen einführen und mit der folgenden Definition kann P anschließend auch als **Wahrscheinlichkeitsverteilung** einer Zufallsvariablen aufgefasst werden.

Definition 3.3 (Zufallsvariable, Wahrscheinlichkeitsverteilung) *Sei (Ω, P) ein Wahrscheinlichkeitsraum. Eine Abbildung*

$$X : \Omega \to \mathbb{R}$$

*heißt **Zufallsvariable** über $\mathbb{R}$. Falls Ω höchstens abzählbar ist, dann heißt X auch eine **diskrete Zufallsvariable**.*

Für $z \in X(\Omega) := \{X(\omega) \mid \omega \in \Omega\}$ betrachte:

$$P[X = z] := P[\{\omega \in \Omega \mid X(\omega) = z\}]$$

*und nenne das Wahrscheinlichkeitsmaß P für diese speziellen Mengen auch die **Verteilung** der Zufallsvariablen X, auch **Wahrscheinlichkeitsverteilung** genannt. $P[X = a] = p$ besagt somit, dass die Zufallsvariable X mit der Wahrscheinlichkeit $p \in [0, 1]$ den Wert $a \in X(\Omega)$ annimmt.*

Sei zusätzlich Y eine diskrete Zufallsvariable über Ω und $b \in Y(\Omega)$, dann ist die Wahrscheinlichkeit, dass X den Wert a und Y den Wert b annimmt, wie folgt definiert:

$$P[X = a, Y = b] := P[\{\omega \in \Omega \mid X(\omega) = a\} \cap \{\omega \in \Omega \mid Y(\omega) = b\}]. \qquad \circ$$

Aus diesen Definitionen folgt direkt folgendes Lemma über unabhängige Zufallsvariablen.

Lemma 3.1 (unabhängige Zufallsvariablen) *Seien (Ω, P) ein Wahrscheinlichkeitsraum, X, Y zwei diskrete Zufallsvariablen über Ω und $a \in X(\Omega)$ und $b \in Y(\Omega)$, dann gilt für $P[Y = b] > 0$:*

$$\begin{aligned} X, Y \text{ unabhängig} &\rightsquigarrow P[X = a, Y = b] = P[X = a] \cdot P[Y = b] \\ &\rightsquigarrow P[X = a \mid Y = b] = P[X = a]. \end{aligned} \qquad \circ$$

Mit Definition 3.3 lässt sich eine weitere Bezeichnung einführen, welche im darauf folgenden Lemma benutzt wird. Die folgenden Definitionen und Lemmata basieren auf den Resultaten aus [Sti06].

Definition 3.4 (Bias) *Sei* $(\Omega = \{0,1\}, P)$ *ein Wahrscheinlichkeitsraum, dann bezeichnet der* ***Bias*** ϵ *einer Zufallsvariablen* X *über* Ω*, auch* ***systematischer Fehler*** *genannt, die Entfernung der Wahrscheinlichkeit, dass* X *den Wert* 0 *annimmt zu der Wahrscheinlichkeit des „fairen", zufälligen Ereignisses, also eines Ereignisses mit der Wahrscheinlichkeit* $\frac{1}{2}$*. Formal gesprochen:*

$$\epsilon(X) := \epsilon := P[X = 0] - \frac{1}{2}. \qquad \circ$$

Zusammen mit der Definition 3.3 liefert diese Definition folgendes Resultat über den Zusammenhang zwischen dem Bias und der Wahrscheinlichkeit.

Lemma 3.2 *Sei* $(\Omega = \{0,1\}, P)$ *ein Wahrscheinlichkeitsraum und* X *eine Zufallsvariable über* Ω*, mit dem zugehörigen Bias* ϵ*, dann gilt:*

$$-\frac{1}{2} \leq \epsilon \leq \frac{1}{2}, \tag{7}$$

$$P[X = 0] = \frac{1}{2} + \epsilon, \tag{8}$$

$$P[X = 1] = \frac{1}{2} - \epsilon. \tag{9}$$

$\circ$

Wird nun zur Vereinfachung der Schreibweise in einem Wahrscheinlichkeitsraum (Ω, P) für eine Zufallsvariable X_i über $\Omega = \{0,1\}$, mit $i \in I$, I Indexmenge, die Wahrscheinlichkeit $P[X_i = 0] =: p_i$ und somit $P[X_i = 1] = 1 - p_i$ definiert, dann lässt sich folgendes Lemma zeigen.

Lemma 3.3 *Sei* $(\Omega = \{0,1\}, P)$ *ein Wahrscheinlichkeitsraum und seien* X_i *und* X_j *zwei unabhängige Zufallsvariablen über* Ω*, mit* $i, j \in I$*,* I *Indexmenge und* $i \neq j$*, dann gilt:*

$$\begin{aligned}
P[X_i = 0, X_j = 0] &= p_i p_j \\
P[X_i = 0, X_j = 1] &= p_i(1 - p_j) \\
P[X_i = 1, X_j = 0] &= (1 - p_i)p_j \\
P[X_i = 1, X_j = 1] &= (1 - p_i)(1 - p_j).
\end{aligned} \qquad \circ$$

Beweis:
Aufgrund der Analogie der anderen Gleichungen wird lediglich die erste Gleichung bewiesen.

Sei $(\Omega = \{0,1\}, P)$ Wahrscheinlichkeitsraum und seien X_i und X_j zwei unabhängige Zufallsvariablen über Ω, mit $i, j \in I$, I Indexmenge und $i \neq j$. Betrachte:

$$\begin{aligned} P[X_i = 0, X_j = 0] &\overset{Def.3.3}{=} P[\{\omega \in \Omega \mid X_i(\omega) = 0\} \cap \{\omega \in \Omega \mid X_j(\omega) = 0\}] \\ &\overset{\substack{Def.3.2 \\ X_i, Y_i \text{ unabh.}}}{=} P[\{\omega \in \Omega \mid X_i(\omega) = 0\}] \cdot P[\{\omega \in \Omega \mid X_j(\omega) = 0\}] \\ &\overset{Def.3.3}{=} P[X_i = 0] \cdot P[X_j = 0] = p_i p_j. \end{aligned}$$

Damit ist die erste Gleichung gezeigt und mit denselben Hilfsmitteln lassen sich auch die anderen Gleichungen analog beweisen. ■

Zur Verbesserung der Lesbarkeit definiere für zwei diskrete Zufallsvariablen X, Y über $\{0,1\}$ den binären Operator $\oplus$ durch $X \oplus Y := X + Y \ mod\ 2$ und damit für eine Familie von Zufallsvariablen $(X_i)_{i \in \{1,\dots,n\}}$ über $\Omega = \{0,1\}$, $n \in \mathbb{N} \setminus \{0\}$ und einen Wahrscheinlichkeitsraum (Ω, P) die Wahrscheinlichkeit $P[X_1 \oplus \cdots \oplus X_n = 0] =: p_{1,\dots,n}$. Dabei gilt auf Grund von Definition 3.3:

$$\begin{aligned} P[X_1 \oplus \cdots \oplus X_n = 0] &\overset{Def.3.3}{=} P[\{\omega \in \Omega \mid (X_1 \oplus \cdots \oplus X_n)(\omega) = 0\}] \\ &= P[\{\omega \in \Omega \mid X_1(\omega) + \cdots + X_n(\omega)\ mod\ 2 = 0\}] \end{aligned}$$

und somit ergibt sich folgendes Lemma:

Lemma 3.4 *Sei $(\Omega = \{0,1\}, P)$ ein Wahrscheinlichkeitsraum und seien X_i und X_j zwei unabhängige Zufallsvariablen über Ω, mit $i, j \in I$, I Indexmenge, dann gilt mit Lemma 3.3:*

$$\begin{aligned} a)\ P[X_i \oplus X_j = 0] &= P[X_i = 0, X_j = 0] + P[X_i = 1, X_j = 1] \\ &= p_i p_j + (1 - p_i)(1 - p_j) \qquad (10) \\ b)\ P[X_i \oplus X_j = 1] &= P[X_i = 0, X_j = 1] + P[X_i = 1, X_j = 0] \\ &= p_i(1 - p_j) + (1 - p_i)p_j \qquad (11) \\ c)\ P[X_i \oplus X_j = 0] &= p_i p_j + (1 - p_i)(1 - p_j) \\ &= p_i p_j + 1 - p_j - p_i + p_i p_j \\ &= 1 - (p_i(1 - p_j) + (1 - p_i)p_j) \\ &= 1 - P[X_i \oplus X_j = 1]. \qquad (12) \end{aligned}$$

∘

Damit sind alle Voraussetzungen für folgendes Lemma gegeben, welches für die lineare Kryptoanalyse in Abschnitt 4.2 benötigt wird.

Lemma 3.5 (Piling-Up-Lemma) *Sei $(\Omega = \{0,1\}, P)$ ein Wahrscheinlichkeitsraum und sei $(X_i)_{i\in\{1,\dots n\}}$, $n \in \mathbb{N} \setminus \{0\}$, eine Familie von paarweise unabhängigen Zufallsvariablen über Ω mit dem jeweils zugehörigen Bias ϵ_i, dann gilt für den Bias von $X_1 \oplus \cdots \oplus X_n$:*

$$\epsilon(X_1 \oplus \cdots \oplus X_n) = 2^{n-1} \prod_{i=1}^{n} \epsilon_i.$$

∘

Beweis (per Induktion über n):
Sei $(\Omega = \{0,1\}, P)$ ein Wahrscheinlichkeitsraum und sei $(X_i)_{i\in\{1,\dots n\}}$, $n \in \mathbb{N} \setminus \{0\}$, eine Familie von paarweise unabhängigen Zufallsvariablen über Ω, mit dem jeweils zugehörigen Bias ϵ_i.

(IA): Sei $i = 1$:

$$\epsilon(X_1) = 2^{1-1}\epsilon_1 = \epsilon_1.$$

(IV): Es gelte für ein beliebiges, aber festes $n \in \mathbb{N}$:

$$\epsilon(X_1 \oplus \cdots \oplus X_n) = 2^{n-1} \prod_{i=1}^{n} \epsilon_i.$$

(IS): $n \mapsto n+1$:

$$\begin{aligned}
\epsilon(X_1 \oplus \cdots \oplus X_{n+1}) &\overset{Def.\ 3.4}{=} P[X_1 \oplus \cdots \oplus X_{n+1} = 0] - \tfrac{1}{2} \\
&= P[(X_1 \oplus \cdots \oplus X_n) \oplus X_{n+1} = 0] - \tfrac{1}{2} \\
&\overset{(10)}{=} p_{1,\dots,n}p_{n+1} + (1 - p_{1,\dots,n})(1 - p_{n+1}) - \tfrac{1}{2} \\
&= p_{1,\dots,n}p_{n+1} + 1 - p_{n+1} - p_{1,\dots,n} + p_{1,\dots,n}p_{n+1} - \tfrac{1}{2} \\
&\overset{(8)}{=} 2p_{1,\dots,n}(\tfrac{1}{2} + \epsilon_{n+1}) + 1 - \tfrac{1}{2} - \epsilon_{n+1} - p_{1,\dots,n} - \tfrac{1}{2} \\
&= p_{1,\dots,n} + 2p_{1,\dots,n}\epsilon_{n+1} - \epsilon_{n+1} - p_{1,\dots,n} \\
&= 2\epsilon_{n+1}(p_{1,\dots,n} - \tfrac{1}{2}) \overset{Def.\ Bias}{=} 2\epsilon_{n+1}\epsilon(X_1 \oplus \cdots \oplus X_n) \\
&\overset{(IV)}{=} 2\epsilon_{n+1}2^{n-1}\textstyle\prod_{i=1}^{n}\epsilon_i \\
&= 2^n \textstyle\prod_{i=1}^{n+1}\epsilon_i.
\end{aligned}$$

Somit wurde mit Hilfe der vollständigen Induktion das Lemma 3.5 für alle $n \in \mathbb{N} \setminus \{0\}$ bewiesen. ■

Aus diesem Lemma folgt direkt ein Korollar, welches ebenfalls für die lineare Kryptoanalyse benötigt wird:

Korollar 3.1 *Sei* $(\Omega = \{0,1\}, P)$ *ein Wahrscheinlichkeitsraum und sei* $(X_i)_{i\in\{1,\dots n\}}$, $n \in \mathbb{N} \setminus \{0\}$, *wie im Piling-Up-Lemma, eine Familie von paarweise unabhängigen Zufallsvariablen über* Ω *mit dem jeweils zugehörigen Bias* ϵ_i*, dann gilt:*

$$\epsilon_j = 0 \text{ für ein } j \in \{1,\dots,n\} \rightsquigarrow \epsilon(X_1 \oplus \cdots \oplus X_n) = 0. \qquad \diamondsuit$$

Beweis:
Folgt direkt aus dem Piling-Up-Lemma. ■

Bei der Betrachtung eines Bias für Linearkombinationen der Art $X_1 \oplus \cdots \oplus X_n = 1 \rightsquigarrow X_1 \oplus \cdots \oplus X_n \oplus 1 = 0$ für eine Familie $(X_i)_{i\in\{1,\dots n\}}$, $n \in \mathbb{N} \setminus \{0\}$ von paarweise unabhängigen Zufallsvariablen über $\mathbb{Z}_2$ hilft folgendes Korollar:

Korollar 3.2 *Sei* $(\Omega = \{0,1\}, P)$ *ein Wahrscheinlichkeitsraum und sei* $(X_i)_{i\in\{1,\dots n\}}$, $n \in \mathbb{N} \setminus \{0\}$, *wie im Piling-Up-Lemma, eine Familie von paarweise unabhängigen Zufallsvariablen über* Ω *mit dem jeweils zugehörigen Bias* ϵ_i*, dann gilt:*

$$\epsilon(X_1 \oplus \cdots \oplus X_n \oplus 1) = -\epsilon(X_1 \oplus \cdots \oplus X_n). \qquad \diamondsuit$$

Beweis:
Sei $(\Omega = \{0,1\}, P)$ *ein Wahrscheinlichkeitsraum und sei* $(X_i)_{i\in\{1,\dots n\}}$, $n \in \mathbb{N} \setminus \{0\}$, *eine Familie von paarweise unabhängigen Zufallsvariablen über* Ω *mit dem jeweils zugehörigen Bias* ϵ_i*, dann gilt:*

$$\begin{aligned}
\epsilon(X_1 \oplus \cdots \oplus X_n \oplus 1) &\overset{\text{Lem. 3.5}}{=} 2^n \textstyle\prod_{i=1}^n \epsilon_i \epsilon(1) \\
&\overset{\text{Def. 3.4}}{=} 2^n \textstyle\prod_{i=1}^n \epsilon_i (P[1=0] - \frac{1}{2}) \\
&= -2^n \tfrac{1}{2} \textstyle\prod_{i=1}^n \epsilon_i \\
&= -2^{n-1} \textstyle\prod_{i=1}^n \epsilon_i \\
&= -\epsilon(X_1 \oplus \cdots \oplus X_n).
\end{aligned}$$

Also ist die Behauptung gezeigt. ■

4 Kryptoanalyse

Die Kryptoanalyse versucht auf einem beliebigen Weg an geheime Informationen des Kryptosystems zu gelangen, wobei zu beachten ist, dass der Verschlüsselungsalgorithmus im Allgemeinen nicht zu den geheimen Informationen gehört, was schon im Jahre 1883 eine der von Auguste Kerckhoff formulierten Maxime darstellte (**Kerckhoffs Prinzip**) [Ker83]. Im Besonderen versuchen die beiden in den folgenden Kapiteln vorgestellten Verfahren, Teile des geheimen Schlüssels zu berechnen.

Es gibt mannigfaltige Gründe, die zu dieser Maxime führten; so ist es zum Beispiel ungleich komplizierter, einen kompromittierten Algorithmus auszutauschen, als einen kompromittierten Schlüssel. Ebenso ist es allgemein schwieriger, einen Algorithmus geheimzuhalten, als einen Schlüssel, wie auch der Versuch der US-Regierung zeigt, die erfolglos probierte, die Veröffentlichung und Verbreitung von RSA zu unterbinden [Wil08].

Dieses Kapitel führt zunächst in Abschnitt 4.1 die Terminologie und einige allgemeine Eigenschaften der Kryptoanalyse ein, bevor in Abschnitt 4.2 die lineare Kryptoanalyse und in Abschnitt 4.3 die differentielle Kryptoanalyse vorgestellt und auf das in Abschnitt 2.2.6 vorgestellte Beispiel angewandt werden.

4.1 Grundlegende Begrifflichkeiten

Unter **Brute-Force** beziehungsweise der **Brute-Force-Suche** wird das Ausprobieren aller potentiellen Lösungen zum Finden der korrekten Lösung eines Problems verstanden. Dieses Verfahren ist zwar theoretisch dazu geeignet, die korrekte Lösung zu finden, aber in vielen Fällen reicht die Rechenkapazität nicht aus, um zu einer Lösung zu gelangen, da mit steigender Komplexität der Probleme die Anzahl der potentiellen Lösungen exponentiell ansteigt [Wika].

Unter dem **Brechen** einer Chiffre wird nicht ausschließlich, wie vielleicht anzunehmen ist, das Finden einer Möglichkeit für den Angreifer, den Klartext einer Nachricht lediglich durch den Besitz des Geheimtextes zu erlangen, verstanden, sondern in der Kryptologie bezeichnet das Brechen einer Chiffre allgemeiner das Auffinden einer Schwäche des Kryptosystems, welche mit einer geringeren Komplexität als mit Brute-Force ausgenutzt werden kann [Sch00]. Zum Beispiel reicht es, um vom Brechen einer Chiffre sprechen zu können, falls statt des gesamten Schlüsselraums nur noch eine Teilmenge ausprobiert werden muss, selbst wenn diese Menge immer noch nicht in praktikabler Zeit mit der heutigen Rechenleistung getestet werden kann.

Von einem **Angriff** (engl.: attack) wird gesprochen, wenn eine Schwäche des Kryptosystems ausgenutzt wird, um an geheime Informationen der Chiffre zu gelangen. Mit welchen

Voraussetzungen diese Informationen erhalten werden, lässt sich in einige Kategorien aufteilen. Im Folgenden werden lediglich, nach dem Schwierigkeitsgrad absteigend sortiert, diejenigen Kategorien aufgeführt, welche in dieser Arbeit Erwähnung finden [Knu94].

Angriff mit bekanntem Geheimtext[5]**:** Der Angreifer besitzt eine Menge von Geheimtexten, zum Beispiel durch das Abhören eines Nachrichtenkanals.

Angriff mit bekanntem Klartext[6]**:** Der Angreifer besitzt eine Menge von Klartext-Geheimtext-Paaren. Das heißt, er kennt eine endliche Anzahl von Klartexten mit den zugehörigen Geheimtexten, hat jedoch keinen Einfluss auf die Art der Texte.

Angriff mit gewähltem Klartext[7]**:** Der Angreifer besitzt, wie bei dem Angriff mit bekanntem Klartext, eine Menge von Klartext-Geheimtext-Paaren, allerdings mit dem Unterschied, dass er sich bei diesem Angriff eine Menge von Klartexten wählt und für diese auf irgendeine Art und Weise die zugehörigen Geheimtexte erhält.

Als eine **Linearkombination** wird in dieser Arbeit ein Term der Art

$$a_0 \cdot X_0 \oplus a_1 \cdot X_1 \oplus \cdots \oplus a_{n-1} \cdot X_{n-1} \oplus a_n \cdot X_n$$

mit $n \in \mathbb{N}$, $a_i \in \mathbb{Z}_2$ und X_i Zufallsvariable über $\mathbb{Z}_2$ $(0 \leq i \leq n)$ bezeichnet.

Die **Differenz** zwischen zwei Wörtern $x, y \in \mathbb{Z}_2^l$ $(l \in \mathbb{N})$ wird in dieser Arbeit als $\Delta(x, y) := x \oplus y$ definiert und damit die Menge $\Delta(x') := \{(x, x^*) \mid \Delta(x, x^*) = x' \; mit \; x, x^* \in \mathbb{Z}_2^l \; (l \in \mathbb{N})\}$ für $x' \in \mathbb{Z}_2^l$ $(l \in \mathbb{N})$ eingeführt.

Als die **Eingabedifferenz** x' von zwei Wörtern $x, x^* \in \mathbb{Z}_2^m$ einer S-Box $S : Z_2^m \to Z_2^n$ $(n, m \in \mathbb{N})$ wird das Wort $x' = x \oplus x^*$ bezeichnet, wobei x und x^* aus dem Definitionsbereich von S sind. Dementsprechend wird das Wort $y' = S(x) \oplus S(x^*)$ **Ausgabedifferenz** von x und x^* der S-Box S genannt. Wenn für den weiteren Verlauf nicht relevant ist, aus welchen Bestandteilen (x, x^*) die Differenz x' zusammengesetzt ist, sondern lediglich, dass sie sich aus Wörtern aus dem Definitionsbereich von S zusammensetzt, dann wird x' auch als eine **Eingabedifferenz** von S bezeichnet. Ebenso lässt sich der Begriff der **Ausgabedifferenz** vereinfachen.

4.2 Lineare Kryptoanalyse

Bei der linearen Kryptoanalyse handelt es sich um einen Angriff mit bekanntem Klartext, der, wie in [Mat94] zu sehen ist, in bestimmten Situationen auch für einen Angriff mit

[5] Engl.: ciphertext only attack.
[6] Engl.: known plaintext attack.
[7] Engl.: chosen plaintext attack.

bekanntem Geheimtext genutzt werden kann. Nachdem Mitsuru Matsui schon im Jahre 1992 zusammen mit Atsuhiro Yamagishi einen ähnlichen Angriff auf FEAL dokumentiert hatte [MA93], veröffentlichte er im Jahre 1993 den Angriff auf DES [Mat94], der heute als lineare Kryptoanalyse bekannt und im Prinzip auf jedwede iterierte Block-Chiffre, insbesondere auf SPNe, anwendbar ist [Sti06].

Im Folgenden werden zuerst die Art des Angriffs allgemein vorgestellt und daraufhin das Verfahren anhand eines Beispiel-Angriffs auf das SPN aus Abschnitt 2.2.6 verdeutlicht.

4.2.1 Basis-Angriff

Der Grundgedanke der linearen Kryptoanalyse basiert auf dem Versuch, für eine iterierte Block-Chiffre der Länge $l \in \mathbb{N}$, einen stochastischen, linearen Zusammenhang von m Zeichen eines Klartextes $M \in \mathbb{Z}_2^l$, sowie c Zeichen eines Geheimtextes $C \in \mathbb{Z}_2^l$ und k Zeichen eines Schlüssels $K \in \mathbb{Z}_2^q$ mit $q, m, c, k \in \mathbb{N}$ zu finden, der mit einer Wahrscheinlichkeit $p \neq \frac{1}{2}$ zutrifft. Im weiteren Verlauf werden die einzelnen Zeichen eines Textes als Zufallsvariablen über $\mathbb{Z}_2$ aufgefasst. Gesucht ist also eine Kombination der Form:

$$M_{i_0} \oplus M_{i_1} \oplus \cdots \oplus M_{i_{m-1}} \oplus C_{j_0} \oplus C_{j_1} \oplus \cdots \oplus C_{j_{c-1}} = K_{h_0} \oplus K_{h_1} \oplus \cdots \oplus K_{h_{k-1}} \tag{13}$$

mit $0 \leq i_\alpha, j_\beta < l$ ($0 \leq \alpha < m$ und $0 \leq \beta < c$) und $0 \leq h_\gamma < q$ ($0 \leq \gamma < k$), wobei der Bias aus Definition 3.4 (S. 25) eine Aussage über die Güte solch einer Linearkombination tätigt; denn je weiter die Wahrscheinlichkeit

$$p = P[\underbrace{M_{i_0} \oplus M_{i_1} \oplus \cdots \oplus M_{i_{m-1}} \oplus C_{j_0} \oplus C_{j_1} \oplus \cdots \oplus C_{j_{c-1}} \oplus K_{h_0} \oplus K_{h_1} \oplus \cdots \oplus K_{h_{k-1}}}_{=:G} = 0],$$

von der Wahrscheinlichkeit $\frac{1}{2}$ des zufälligen Verhaltens entfernt ist, desto weniger zufällig ist der Zusammenhang dieser Zeichen. Für $p > \frac{1}{2}$ folgt dies direkt aus der Gleichung (13) und falls $p < \frac{1}{2}$ ist, dann ergibt sich die Gegenwahrscheinlichkeit $q = P[G = 1] > \frac{1}{2}$ und damit sollte bei einer Menge von Klartext-Geheimtext-Paaren, die mit demselben Schlüssel verarbeitet wurden, für mehr als die Hälfte dieser Paare die Gleichung $G = 1$ erfüllt sein. Bei einer gegen diese Art Angriff sicheren Chiffre sollte also die Wahrscheinlichkeit jeder solcher Linearkombinationen bei $\frac{1}{2}$ liegen, damit durch den Besitz von beliebigen Klartext- und Geheimtextzeichen auf diese Weise kein Rückschluss auf den Schlüssel beziehungsweise auf Schlüsselzeichen gezogen werden kann.

Ist eine Gleichung der Art (13) gefunden, so lässt sich diese mit zwei unterschiedlichen Methoden ausnutzen, die jeweils eine Menge von mit demselben Schlüssel verschlüsselten Klartext-Geheimtext-Paaren, benötigen (Angriff mit bekanntem Klartext). Zuerst wird

die einfachere Methode vorgestellt, welche keine Teilentschlüsselung des Systems benutzt und somit weniger rechenintensiv ist, dafür jedoch auch nur eine Aussage $K_{h_0} \oplus K_{h_1} \oplus \cdots \oplus K_{h_{k-1}} = 0$ oder $K_{h_0} \oplus K_{h_1} \oplus \cdots \oplus K_{h_{k-1}} = 1$ liefert.

Lemma 4.1 (Methode 1 [Mat94]) *Sei $\mathcal{N}$ die Menge aller verfügbaren mit demselben Schlüssel verschlüsselten Klartext-Geheimtext-Paare und $\mathcal{T} \subseteq \mathcal{N}$ die Menge der Klartext-Geheimtext-Paare, für die die linke Seite der Gleichung (13) gleich Null ist. Sei des Weiteren $T = |\mathcal{T}|$, $N = |\mathcal{N}|$ und p die Wahrscheinlichkeit, dass die Linearkombination der relevanten Klartext-, Geheimtext- und Schlüsselzeichen gleich Null ist (Gleichung (13)). Dann nehme für*

$$T > \tfrac{N}{2}\text{:} \qquad K_{h_0} \oplus K_{h_1} \oplus \cdots \oplus K_{h_{k-1}} = \begin{cases} 0 & \text{für } p > \frac{1}{2} \\ 1 & \text{für } p < \frac{1}{2} \end{cases}$$

$$T <= \tfrac{N}{2}\text{:} \qquad K_{h_0} \oplus K_{h_1} \oplus \cdots \oplus K_{h_{k-1}} = \begin{cases} 1 & \text{für } p > \frac{1}{2} \\ 0 & \text{für } p < \frac{1}{2} \end{cases}$$

an. ∘

Das heißt also, wenn $M_{i_0} \oplus M_{i_1} \oplus \cdots \oplus M_{i_{m-1}} \oplus C_{j_0} \oplus C_{j_1} \oplus \cdots \oplus C_{j_{c-1}} = 0$ aus Gleichung (13) für den Großteil der verfügbaren Klartext-Geheimtext-Paare gilt, dann lässt sich annehmen, dass die Gleichung im Allgemeinen gilt und somit die Linearkombination der Schlüsselzeichen wahrscheinlich gleich Null ist, wenn die Wahrscheinlichkeit p für das Eintreten der Gleichung (13) hoch ist – also größer als $\frac{1}{2}$ –. Im Fall $p < \frac{1}{2}$ tritt, wie oben erwähnt, der Umkehrschluss ein. Dann ist die Wahrscheinlichkeit für $M_{i_0} \oplus M_{i_1} \oplus \cdots \oplus M_{i_{m-1}} \oplus C_{j_0} \oplus C_{j_1} \oplus \cdots \oplus C_{j_{c-1}} \oplus K_{h_0} \oplus K_{h_1} \oplus \cdots \oplus K_{h_{k-1}} = 1$ hoch und damit folgt direkt durch Einsetzen, dass die Linearkombination der Schlüsselzeichen wahrscheinlich Eins ist. Der zweite Teil des Lemmas ist analog zu begründen. Dabei ist leicht einzusehen, dass die Wahrscheinlichkeit der Korrektheit der Annahme steigt, wenn sich die Anzahl der Klartext-Geheimtext-Paare N erhöht, denn wenn die Stichprobengröße wächst, so steigt auch die Aussagekraft der Behauptung. Ebenso verhält es sich mit dem Bias: wenn dieser betragsmäßig größer wird, erhöht sich dadurch die Wahrscheinlichkeit für die Korrektheit der Gleichung und damit auch für die Richtigkeit der Annahme [Mat94].

Die zweite Methode ist eine Weiterentwicklung der ersten und folglich etwas komplexer, liefert dafür aber ganze Blöcke an Schlüsselzeichen, bis hin zu einen kompletten, letzten Rundenschlüssel. Die Idee dieser Methode basiert darauf, dass eine iterierte Block-Chiffre mit n Runden auf eine iterierte Block-Chiffre mit $n-1$ Runden heruntergebrochen wird, indem die letzte Runde als entschlüsselt angenommen wird. Damit ist für diese Methode

eine Gleichung der Art:

$$M_{i_0} \oplus M_{i_1} \oplus \cdots \oplus M_{i_{m-1}} \oplus U^{n-1}_{g_0} \oplus U^{n-1}_{g_1} \oplus \cdots \oplus U^{n-1}_{g_{a-1}} = K_{h_0} \oplus K_{h_1} \oplus \cdots \oplus K_{h_{k-1}} \tag{14}$$

gesucht, wobei die $U^{n-1}_{g_\alpha}$ mit $0 \leq g_\alpha < l$ $(0 \leq \alpha < a)$ Zufallsvariablen über $\mathbb{Z}_2$ sind und die durch das partielle Entschlüsseln erhaltenen und für die Linearkombination relevanten $a \in \mathbb{N}$ Zeichen vor der letzten Runde bezeichnen. Bei einem SPN wären dies also die Zeichen vor der letzten Substitution (siehe Abbildung 7). Die Klartext- und Schlüsselzeichen (M_{i_β} mit $0 \leq i_\beta < l$ $(0 \leq \beta < m)$ und K_{h_γ} mit $0 \leq h_\gamma < q$ $(0 \leq \gamma < k)$) sind dabei immer noch wie in Gleichung (13) definiert.

Um zu diesen $U^{n-1}_{g_\alpha}$ mit $0 \leq g_\alpha < l$ $(0 \leq \alpha < a)$ zu gelangen, werden die Zeichen aller potentiellen Schlüssel $k \in \mathbb{Z}_2^l$ des letzten Rundenschlüssels ausprobiert, die für die partielle Entschlüsselung benötigt werden. Wie viele und welche Schlüsselzeichen benötigt werden, ist von der Art der Chiffre abhängig. Bei DES zum Beispiel wird ein gesamter Rundenschlüssel benötigt, um mit einer teilweisen Entschlüsselung an einzelne Zeichen vor der letzten Runde zu gelangen. Im Gegensatz dazu reichen bei den SPNen, die in dieser Arbeit betrachtet werden, die Blöcke von Schlüsselzeichen aus, welche mit einem Ausgang einer S-Box verknüpft sind, an deren Eingang eines der $U^{n-1}_{g_\alpha}$ mit $0 \leq g_\alpha < l$ $(0 \leq \alpha < a)$ anliegt.

Sind also alle relevanten Zeichen der Gleichung (14) vorhanden, dann lässt sich dieser Zusammenhang wie folgt ausnutzen:

Lemma 4.2 (Methode 2 [Mat94]) *Sei $K^n_{(i)} \in \mathbb{Z}_2^l$ $(0 \leq i < 2^l$ und l Länge der Rundenschlüssel$)$ einer der 2^l potentiellen letzten Rundenschlüssel einer iterierten Block-Chiffre und sei weiterhin $(K^n_{(i)})_J$ mit $J \subseteq \{0, 1, \ldots, l-1\}$ ein Wort aus den für die partielle Entschlüsselung der letzten Runde relevanten Schlüsselzeichen des i-ten potentiellen Schlüssels der letzten Runde. Zusätzlich sei $T_i \in \mathbb{N}$ die Anzahl der Klartexte, für die, die mit Hilfe der letzten Rundeschlüsselzeichen $(K^n_{(i)})_J$ erstellte, linke Seite der Gleichnung (14) gleich Null ist und sei schließlich $N \in \mathbb{N}$ die Anzahl von Klartexten, die zur Verfügung stehen, K_{h_γ} $(0 \leq \gamma < k)$, wie in Gleichung (13), k Zeichen des Hauptschlüssels K und p die Wahrscheinlichkeit, dass Gleichung (14) gilt. Dann lässt sich mit $T_{max} := max\{T_i \mid 0 \leq i < 2^l\}$ und $T_{min} := min\{T_i \mid 0 \leq i < 2^l\}$ folgendes annehmen:*

Für $|T_{max} - \frac{N}{2}| > |T_{min} - \frac{N}{2}|$:

- *Die zu T_{max} gehörigen Schlüsselzeichen sind die gesuchten.*
- $K_{h_0} \oplus K_{h_1} \oplus \cdots \oplus K_{h_{k-1}} = \begin{cases} 0 & für\ p > \frac{1}{2} \\ 1 & für\ p < \frac{1}{2} \end{cases}$.

Für $|T_{max} - \frac{N}{2}| < |T_{min} - \frac{N}{2}|$**:**

- *Die zu T_{min} gehörigen Schlüsselzeichen sind die gesuchten.*
- $K_{h_0} \oplus K_{h_1} \oplus \cdots \oplus K_{h_{k-1}} = \begin{cases} 1 & für\ p > \frac{1}{2} \\ 0 & für\ p < \frac{1}{2} \end{cases}$. ∘

Für die nicht korrekten Schlüsselzeichen wird erwartet, dass die linke Seite der Gleichung (14) für ungefähr die Hälfte $\frac{N}{2}$ der Klartext-Geheimtext-Paare gleich Null ist; denn dies bedeutet, dass sich dieser Teil annähernd zufällig verhält. Da jedoch Gleichung (14) absichtlich so gewählt wurde, dass der Zusammenhang der Zeichen gerade nicht zufällig ist, sind die Schlüsselzeichen, deren Zähler am weitesten von der Hälfte der Paare entfernt ist, die wahrscheinlichsten, denn mit diesen wurde somit wahrscheinlich die letzte Runde richtig entschlüsselt. Da jedoch auch der oben schon so häufig erwähnte Umkehrschluss betrachtet werden muss, müssen auch die Schlüsselzeichen mit der minimalen Anzahl betrachtet werden, denn das bedeutet, dass für einen Großteil der Klartext-Geheimtext-Paare die linke Seite der Gleichung (14) gleich Eins ist und aus diesem Grund ergeben sich erneut die beiden Fälle. Alles weitere erklärt sich analog zur Methode 1. Auf dieser Idee basiert auch der Algorithmus, der in dem Werkzeug, welches in Kapitel 5 beschrieben wird, implementiert wurde.

Bleibt also noch das Problem, Linearkombinationen in der Art von Gleichung (14) zu finden, die einen hohen Bias besitzen. Wie erwähnt sind die einzigen nicht linearen Teile eines SPNes die S-Boxen. Im Fall linearer S-Boxen ließe sich, wie in Abschnitt 2.2.5 gezeigt, leicht eine solche Linearkombination finden. Die lineare Kryptoanalyse versucht diese Tatsache auszunutzen, indem sie die nicht linearen S-Boxen linear annähert und daraufhin diesen Zusammenhang auf das gesamte SPN ausweitet, um zu einer Gleichung der Art (14) zu gelangen.

Diese Annäherung, auch **lineare Approximation** der S-Box genannt, wird im Folgenden vorgestellt.

4.2.2 Lineare Approximation von S-Boxen

Sei

$$S : \mathbb{Z}_2^m \to \mathbb{Z}_2^n,$$

mit $n, m \in \mathbb{N} \setminus \{0\}$, eine nicht lineare S-Box und bezeichne $x = x_0 \dots x_{m-1}$ ein Wort aus dem Definitionsbereich von S und $y = y_0 \dots y_{n-1}$ ein Wort aus der Bildmenge von S. Das Wort x wird auch als **Eingabewort** oder einfach **Eingabe** von S und y dementsprechend als **Ausgabewort** oder **Ausgabe** von S bezeichnet. Dabei ist x zufällig aus $\mathbb{Z}_2^m$ gewählt,

sodass jedes x_i $(0 \leq i < m)$ eine Zufallsvariable X_i über $\mathbb{Z}_2$ mit der Wahrscheinlichkeit $p_i = \frac{1}{2}$, also Bias $\epsilon_i = 0$, definiert. Da x zufällig gewählt wurde, sind alle diese X_i stochastisch unabhängig voneinander. Im Gegensatz dazu sind die durch die einzelnen y_j definierten Zufallsvariablen Y_j $(0 \leq j < n)$ über $\mathbb{Z}_2$ im Allgemeinen nicht stochastisch unabhängig voneinander und auf Grund der Abbildung auch nicht unabhängig von den X_i. Gesucht ist ein Term der Art

$$\underbrace{a_0 \cdot X_0 \oplus a_1 \cdot X_1 \oplus \cdots \oplus a_{m-1} \cdot X_{m-1} \oplus b_0 \cdot Y_0 \oplus b_1 \cdot Y_1 \oplus \cdots \oplus b_{n-1} \cdot Y_{n-1}}_{=:G} \tag{15}$$

mit $a_i, b_i \in \mathbb{Z}_2$ $(0 \leq i < m$ und $0 \leq j < n)$, der mit einer sehr hohen beziehungsweise geringen Wahrscheinlichkeit gleich Null ist. Das Wort $a_0 a_1 \dots a_{m-1}$ wird als **Eingabesumme** und das Wort $b_0 b_1 \dots b_{n-1}$ als **Ausgabesumme** bezeichnet. Der Grund dafür, dass geringe, sowie hohe Wahrscheinlichkeiten für $P[G = 0]$ betrachtet werden, liegt erneut daran, dass $P[G = 0] < \frac{1}{2} \rightsquigarrow P[G = 1] > \frac{1}{2}$. S-Boxen, für die eine solche Linearkombination existiert, können für die lineare Kryptoanalyse benutzt werden, da sie, wie im Folgenden zu sehen, eine Schwäche im Sinne dieses Angriffs aufweisen, welche ausgenutzt werden kann, um eine Gleichung der Art (14) mit einem betragsmäßig hohem Bias zu finden. Die Begrifflichkeit einer Schwäche wird im Folgenden definiert.

Definition 4.1 (lin-schwach) *Eine S-Box $S : \mathbb{Z}_2^m \to \mathbb{Z}_2^n$ mit $n, m \in \mathbb{N} \setminus \{0\}$ heißt **angreifbar im Sinne der linearen Kryptoanalyse**, wenn*

$\exists\, a \in \mathbb{Z}_2^m$ *und* $b \in \mathbb{Z}_2^n$, *mit* $a \neq 0^m$ *und* $b \neq 0^n$:

$$\epsilon(a_0 \cdot X_0 \oplus a_1 \cdot X_1 \oplus \cdots \oplus a_{m-1} \cdot X_{m-1} \oplus b_0 \cdot Y_0 \oplus b_1 \cdot Y_1 \oplus \cdots \oplus b_{n-1} \cdot Y_{n-1}) \neq 0,$$

wobei die $(X_i)_{0 \leq i < m}$ Zufallsvariablen für die Eingabezeichen von S und dementsprechend die $(Y_i)_{0 \leq i < n}$ Zufallsvariablen für die Ausgabezeichen von S bezeichnen.

*Falls eine S-Box in diesem Sinne angreifbar ist, dann nenne diese S-Box auch **lin-schwach**. Je größer $|\epsilon|$, desto lin-schwächer ist die S-Box.* ∘

Das bedeutet: Für eine lin-schwache S-Box S existiert auf Grund der Definition eine Linearkombination aus Eingabe- und Ausgabezeichen von S, die für mehr als die Hälfte aller Eingaben gleich Null beziehungsweise gleich Eins ist. Für diesen Angriff werden, wie im Abschnitt 4.2.1 gesehen, lediglich Aussagen über Linearkombinationen von Zeichen getätigt, sodass für die benötigte Aussage die nicht lineare S-Box S durch diese Linearkombination ersetzt werden kann, da sich diese für einen Großteil der Eingaben von S im Sinne der Linearkombination von Zeichen genau gleich verhält. Solch eine Linearkombination wird auch als **Approximation** beziehungsweise **lineare Approximation** von

S bezeichnet. In der Definition wird der Fall, dass die Eingabesumme beziehungsweise die Ausgabesumme nur aus Nullen besteht, herausgenommen, weil zu viele Informationen verloren gingen, da die gesamte Eingabe oder Ausgabe der S-Box nicht beachtet und damit eine sinnvolle Ausweitung auf das gesamte SPN unmöglich gemacht werden würde.

S-Boxen, die durch eine solche Approximation in den Angriff involviert sind, werden im Folgenden auch **aktive S-Boxen** genannt.

Um zu den Wahrscheinlichkeiten für die Linearkombinationen zu gelangen, werden für alle Eingaben der S-Box die zugehörigen Ausgaben berechnet und daraufhin wird für alle Kombinationen von Eingabe- und Ausgabesumme der Term (15) ausgewertet und nachgeprüft, für wie viele Eingaben dieser gleich Null ist. Diese Anzahlen werden häufig in einer Tabelle aufgelistet, beispielhaft zu sehen in Abbildung 10. Eine solche Tabelle wird **Approximationstabelle** genannt.

Approximationen von S-Boxen lassen sich auf das SPN ausweiten, indem in jeder Runde möglichst gute Approximationen für eine oder mehrere S-Boxen in Abhängigkeit der vorherigen Runde gewählt werden, sodass im Gesamten eine Linearkombination mit möglichst hohem Betrag des Bias und möglichst vielen S-Boxen, die in der letzten Runde involviert sind, entsteht. Dieser Zusammenhang wird in folgender induktiven Bemerkung manifestiert.

Bemerkung 4.1 (Lineare Approximations-Ausweitung) *Sei* $SP = (\mathbb{Z}_2^l, K, \kappa, S = (S_1, \dots, S_t), T, n)$ *ein SPN mit* $l, n, t \in \mathbb{N} \setminus \{0\}$ *und mit* $S_j : \mathbb{Z}_2^{m_j+1} \to \mathbb{Z}_2^{n_j+1}$, $m_j, n_j \in \mathbb{N}$, $j \in \{1, \dots, t\}$.

Für $0 \leq i < n$ *bezeichne* u^i *den Text nach der* $(i+1)$*-ten Rundenschlüsseladdition,* v^i *den Text nach der* $(i+1)$*-ten Ausführung der S-Chiffre* S*,* w^i *den Text nach der* $(i+1)$*-ten Ausführung der Transposition* T *und* $m \in \mathbb{Z}_2^l$ *die Eingabe des SPNes. Diese Bezeichnungen lassen sich anschaulich in dem Beispiel in Abbildung 8 nachvollziehen. Die jeweils zu den einzelnen Zeichen eines Textes zugehörigen Großbuchstaben bezeichnen, genauso wie im vorherigen Abschnitt, die Zufallsvariablen über* $\mathbb{Z}_2$*, welche durch die einzelnen Zeichen definiert sind.*

Seien des Weiteren K *der Hauptschlüssel und* κ *der Rundenschlüssel-Erzeugungs-Algorithmus mit* $\kappa(K) = (K^0, \dots, K^n)$*. Bezeichne außerdem* $a_{<j>}$ $(j \in \{1, \dots, t\})$ *den Block eines Textes* $a \in \mathbb{Z}_2^l$*, den die zugehörige S-Box* S_j *verarbeitet und sei* $J^i \subseteq \{1, \dots, t\}$ *die Menge der Indizes der aktiven S-Boxen der* $(i+1)$*-ten Runde, dann bezeichnet* A^i *eine Approximation der ersten* $i+1$ *Runden von* SP *und lässt sich wie folgt berechnen:*

Anfang: $i = 0$ *Zu jeder aktiven S-Box S_j der 1-ten Runde ($j \in J^0$) wähle eine Linearkombination L_j^0 aus Eingabe- und Ausgabezeichen der S-Box S_j, für die $\epsilon(L_j^0) \neq 0$ ist. Dann gilt:*

$$L_j^0 = (a_j)_0 \cdot U^0_{<j>_0} \oplus \cdots \oplus (a_j)_{m_j} \cdot U^0_{<j>_{m_j}} \oplus (b_j)_0 \cdot V^0_{<j>_0} \oplus \cdots \oplus (b_j)_{n_j} \cdot V^0_{<j>_{n_j}},$$

mit $a_j \in \mathbb{Z}_2^{m_j+1}$, $b_j \in \mathbb{Z}_2^{n_j+1}$.

Es gilt, da für die Approximation des SPNes lediglich die Zufallsvariablen, die in die linearen Approximationen der S-Boxen involviert sind, verfolgt werden müssen, für alle $k \in \{1, \ldots, t\}$:

$$v^0_{<k>} = \begin{cases} a_k & \text{für } k \in J \\ 0^{n_k+1} & \text{sonst} \end{cases}.$$

Die einzelnen $U^0_{<j>_\alpha}$ $(0 \leq \alpha \leq m_j)$ lassen sich auf Grund der Definition des SPNes mit einer Addition von einem Schlüsselzeichen mit einem Eingabezeichen von SPN_{bsp} ersetzen, also gilt für alle $j \in J$:

$$L_j^0 = (a_j)_0 \cdot (M_{<j>_0} \oplus K^0_{<j>_0}) \oplus \cdots \oplus (a_j)_{m_j} \cdot (M_{<j>_{m_j}} \oplus K^0_{<j>_{m_j}}) \oplus \cdots$$
$$\cdots \oplus (b_j)_0 \cdot V^0_{<j>_0} \oplus \cdots \oplus (b_j)_{n_j} \cdot V^0_{<j>_{n_j}}.$$

Damit ist die Approximation der ersten Runde:

$$A^0 = \bigoplus_{j \in J} L_j^0.$$

Voraussetzung: *Für ein festes i mit $0 \leq i < n-1$ sei die Approximation A^i für die ersten $i+1$ Runden von SP berechnet.*

Schritt: $i \mapsto i+1$, $0 < i+1 < n-1$ *Sei A^i die Approximation des SPNes der vorherigen Runden und v^i wie im Fall $i = 0$ berechnet, dann verfolge die durch die Einsen in v^i gekennzeichneten, in die Approximation involvierten Zufallsvariablen durch die Transposition. Das heißt also, die Positionen der in der $(i+1)$-ten Runde involvierten Zufallsvariablen der Ausgabezeichen sind mit der Indexmenge $I = \{j \mid (T(v^i))_j = 1 \text{ mit } 0 \leq j < l)\}$ für die $(i+2)$-te Runde beschrieben. Jedes Zeichen $(T(v^i))_j = w_j^i$ mit $j \in I$, welches durch die Addition mit einem Schlüsselzeichen an einer S-Box der $(i+2)$-ten Runde anliegt, aktiviert diese S-Box für diese Runde. Dann wähle zu jeder aktiven S-Box S_j der $(i+2)$-ten*

Runde ($j \in J^{i+1}$) *Ausgabezeichen der S-Box* S_j, *sodass eine Linearkombination* L_j^{i+1} *aus den durch die* $(i+1)$*-te Runde vorgegebenen Eingabezeichen und den gewählten Ausgabezeichen der S-Box* S_j *entsteht, für die* $\epsilon(L_j^{i+1}) \neq 0$. *Dann gilt:*

$$L_j^{i+1} = w^i_{<j>_0} \cdot U^{i+1}_{<j>_0} \oplus \cdots \oplus w^i_{<j>_{m_j}} \cdot U^{i+1}_{<j>_{m_j}} \oplus (b_j)_0 \cdot V^{i+1}_{<j>_0} \oplus \cdots \oplus (b_j)_{n_j} \cdot V^{i+1}_{<j>_{n_j}},$$

mit $b_j \in \mathbb{Z}_2^{n_j+1}$.

Die einzelnen $U^{i+1}_{<j>_\alpha}$ $(0 \leq \alpha \leq m_j)$ *lassen sich auf Grund der Definition des SPNes mit einer Addition von einem Schlüsselzeichen mit einem Zeichen aus* w^i *ersetzen, also gilt für alle* $j \in J$:

$$L_j^{i+1} = w^i_{<j>_0} \cdot (W^i_{<j>_0} \oplus K^{i+1}_{<j>_0}) \oplus \cdots \oplus w^i_{<j>_0} \cdot (W^i_{<j>_{m_j}} \oplus K^{i+1}_{<j>_{m_j}}) \oplus \cdots$$
$$\cdots \oplus (b_j)_0 \cdot V^{i+1}_{<j>_0} \oplus \cdots \oplus (b_j)_{n_j} \cdot V^{i+1}_{<j>_{n_j}}.$$

Damit ist die Approximation der ersten $i+2$ *Runden von* SP:

$$A^{i+1} = A^i \oplus \bigoplus_{j \in J} L_j^{i+1}.$$

Da jede der in diese Approximation involvierten Zufallsvariablen W^i_α, *mit* $\alpha \in \{0, \ldots, l-1\}$ *gerade die durch Transposition aus den in der Approximation* A^i *involvierten Zufallsvariablen* V^i_β, *mit* $\beta \in \{0, \ldots, l-1\}$ *darstellen, heben sich diese in* A^{i+1} *auf, sodass die Approximation* A^{i+1} *lediglich aus Zufallsvariablen definiert durch Eingabezeichen des SPNes, Schlüsselzeichen und den Ausgaben der in der* $i+2$*-ten Runde aktiven S-Boxen besteht.*

Im Fall $i+1 = n-2$ *werden alle diese* V^{i+1}_β, *mit* $\beta \in \{0, \ldots, l-1\}$ *mit Hilfe desselben Verfahrens durch eine Addition von Zeichen des* n*-ten Rundenschlüssels und Eingabezeichen vor der* n*-ten Substitution ersetzt, sodass die gewünschte Gleichung der Art (14) erstellt wurde, in der lediglich Eingabezeichen des SPNes, Schlüsselzeichen und Zeichen vor der letzten Substitution enthalten sind.* ♣

Leichter ist dieser Zusammenhang anhand des Beispiels in Abschnitt 4.2.3 nachzuvollziehen.

Auch wenn die Linearkombinationen der einzelnen Approximationen der S-Boxen im Allgemeinen nicht stochastisch unabhängig voneinander sind, lässt sich dies für die Praxis annehmen, sodass einige Zusammenhänge für unabhängige Zufallsvariablen benutzt und ausgenutzt werden können und damit der Bias der Approximation des SPNes berechnet

werden kann [Hey02]. Auch wenn die Bewertung der Approximation somit keinem mathematischen Beweis standhält, liefert sie für die Praxis eine genügend gute Approximation, sodass die richtigen Schlüsselzeichen herausbekommen werden [Sti06].

Da der Bias der einzelnen Approximationen der S-Boxen und somit auch deren Umformungen L_j^i, wobei $j \in J^i \subseteq \{1, \ldots, t\}$ die Indizes der aktiven S-Boxen der $(i+1)$-ten Runde bezeichnet, bekannt ist, lässt sich unter Berücksichtigung des letzten Absatzes und mit Hilfe des Piling-Up-Lemmas der Bias der Approximation A des SPNes, also einer Gleichung der Art (14), berechnen:

$$\epsilon(A) \overset{\text{Bem. 4.1}}{=} \epsilon\Big(\bigoplus_{i=0}^{n-2}\Big(\bigoplus_{j\in J^i} L_j^i\Big)\Big) \overset{\text{Lem. 3.5}}{=} 2^{\sum_{i=0}^{n-2}|J^i|-1} \cdot \prod_{i=0}^{n-2}\Big(\prod_{j\in J^i} \epsilon(L_j^i)\Big). \tag{16}$$

Da für die Methode 2 (Lemma 4.2) die linke Seite von Gleichung (14) benötigt wird, bedarf es einer weiteren Umformung:

$$\begin{aligned} \epsilon(A) &\overset{\text{Bem. 4.1}}{=} \epsilon(\underbrace{M_{i_0} \oplus \cdots \oplus M_{i_{m-1}} \oplus U_{g_0}^{n-1} \oplus \cdots \oplus U_{g_{a-1}}^{n-1}}_{=:L} \oplus \underbrace{K_{h_0} \oplus \cdots \oplus K_{h_{k-1}}}_{=:k}) \\ &\overset{\text{Lem. 3.5}}{=} 2^1 \cdot \epsilon(L) \cdot \epsilon(k) \end{aligned} \tag{17}$$

mit Bezeichnungen wie in Gleichung (14). Da die Schlüsselzeichen für ein SPN fix sind, gilt $k = 0$ oder $k = 1$ und somit

$$\epsilon(L) = \begin{cases} \epsilon(A) & \text{für } k = 0 \\ -\epsilon(A) & \text{für } k = 1 \end{cases}.$$

Also ist der Wert, der als Bewertung für die Güte der linearen Approximation des SPNes dient, direkt berechenbar.

Wie diese Approximation der S-Boxen im konkreten Fall berechnet und daraufhin dieser Zusammenhang auf ein SPN ausgeweitet wird, wird im Folgenden an einem Beispiel verdeutlicht und daraufhin verwendet, um an konkrete Schlüsselwerte zu gelangen.

4.2.3 Beispiel-Angriff auf ein Substitutions-/Permutations-Netz

Dieser Abschnitt zeigt einen linearen Angriff nach der Methode 2 (Lemma 4.2) auf das in Beispiel 2.3 vorgestellte Substitutions-/Permutations-Netz SPN_{bsp} und grenzt dabei die Anzahl der möglichen Schlüssel für die letzte Runde von $2^6 = 64$ auf vier Schlüssel ein.

Eine graphische Darstellung des Angriffs ist in Abbildung 8 zu sehen und dient als Unterstützung für das Verständnis der Verfahren. Lediglich die in diesem Angriff aktiven

S-Boxen sind beschriftet; die Pfeile bezeichnen die Zufallsvariablen, die in die lineare Approximation des SPNes involviert sind.

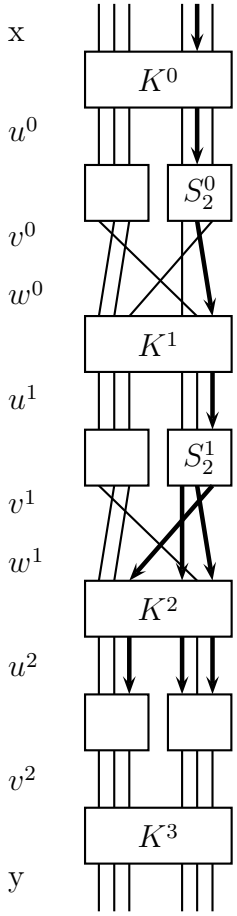

Abbildung 8: Linearer Angriff auf SPN_{bsp}

Da für jedes Netz eine Vielzahl von Approximationen der einzelnen S-Boxen und damit auch eine Vielzahl von Approximations-Ausweitungen existieren, musste eine Entscheidung bezüglich dieser unterschiedlichen Ausweitungen getroffen werden. Die Wahl ist auf die folgende Approximation des SPNes gefallen, da deren Linearkombination zum einen zusammen mit wenigen anderen Approximationen, den betragsmäßig höchsten Bias besitzt und zum anderen, als Vorteil gegenüber den anderen mit betragsmäßig gleich hohem Bias, den gesamten letzten Rundenschlüssel geliefert hat. Auch wenn dieser lediglich in einer Auswahl zusammen mit drei weiteren möglichen Rundenschlüsseln auftaucht, so ist dieses Ergebnis für die weitere Berechnung hilfreicher als drei eindeutige Zeichen des letzten Rundenschlüssels, die bei anderen Approximationen erreichbar sind, da in diesem Fall für den Schlüssel der letzten Runde noch $2^3 = 8$ statt der oben erwähnten vier Möglichkeiten existieren.

Es besteht die Möglichkeit, dass sich Approximationen mit einem betragsmäßig höheren Bias für dieses SPN finden lassen, da eine möglichst gute Approximations-Ausweitung lediglich mit Hilfe einer Heuristik gefunden wurde, jedoch stellt diese Approximation mit einem Bias von $-\frac{1}{4}$ schon eine nahezu perfekte Approximation dar und ist damit ausreichend für die Verdeutlichung der Zusammenhänge. Das Verfahren, welches zu dieser Approximation führte, ist in Kapitel 5 beschrieben.

Wie in Abschnitt 2.2.6 erwähnt, war die Größe von SPN_{bsp} ausschlaggebend für dessen Wahl und daraus folgt auch der große Vorteil dieses Netzes und somit des folgenden Angriffes, da alle vorbereitenden Berechnungen noch sehr gut per Hand berechnet, nachvollzogen und auch vollständig abgedruckt werden können. Um zum Beispiel alle möglichen Linearkombinationen aus Eingabe- und Ausgabesumme für eine S-Box auflisten zu können, bedarf es bei einer S-Box mit drei Zeichen als Eingabe Platz für $2^3 \cdot 2^3 = 64$ Linearkombinationen; bei einer S-Box mit vier Zeichen als Eingabe müssen dafür schon $2^4 \cdot 2^4 = 256$ Linearkombinationen berechnet werden.

Auch wenn keine systematische Untersuchung des folgenden Zusammenhangs durchgeführt wurde, zeigten die ersten Eindrücke, dass der Nachteil solch kleiner S-Boxer darin begründet liegt, dass die berechneten Wahrscheinlichkeiten der Linearkombinationen lediglich auf acht Werten (2^3 Eingabezeichen) basieren, sodass zwar im einzelnen recht hohe Wahrscheinlichkeiten herausbekommen wurden, diese jedoch nicht die gewünschte Aussagekraft besitzen. Daraus folgt, dass der korrekte letzte Rundenschlüssel nicht alleine, sondern mit drei weiteren Schlüsseln die beste Bewertung erhalten hat.

X_0	X_1	X_2	Y_0	Y_1	Y_2	$X_1 \oplus Y_1$	$X_2 \oplus Y_0 \oplus Y_1 \oplus Y_2$
0	0	0	0	1	0	1	1
0	0	1	1	0	1	0	1
0	1	0	1	0	0	1	1
0	1	1	1	1	0	0	1
1	0	0	0	0	1	0	1
1	0	1	0	0	0	0	1
1	1	0	1	1	1	0	1
1	1	1	0	1	1	0	1
Bias:						$\frac{1}{4}$	$-\frac{1}{2}$

Abbildung 9: Einige Linearkombinationen der S-Box S_2 von SPN_{bsp}

Dieser Angriff benutzt lediglich zwei aktive S-Boxen, also werden auch zwei Linearkombinationen von Eingabe- und Ausgabezeichen der S-Boxen benötigt, die diese S-Boxen

gut approximieren. Da lediglich S_2 in die Approximation von SPN_{bsp} involviert ist, reicht es für diese S-Box, zu allen möglichen Eingaben die zugehörigen Ausgaben zu berechnen und daraufhin alle $2^3 \cdot 2^3 = 64$ möglichen Linearkombinationen dieser Zeichen auszuwerten. Die Tabelle aller möglichen Kombinationen für S_1 befindet sich im Anhang im Abschnitt A.1 und für S_2 im Abschnitt A.2. Der relevante Ausschnitt, der die für diesen Angriff benötigten Linearkombinationen enthält, ist in Abbildung 9 zu finden; dabei bezeichnen die $(X_i)_{\{0,1,2\}}$ die Eingabezeichen von S_2 und $(Y_i)_{\{0,1,2\}}$ die zugehörigen Ausgabezeichen.

Die Approximationstabellen der S-Boxen, die die Häufigkeiten, dass die jeweiligen Linearkombinationen gleich Null sind, beinhalten, sind in Abbildung 10 zu sehen, wobei die Eingabesummen dezimalcodiert auf den Zeilen und die Ausgabesummen dezimalcodiert auf den Spalten aufgetragen sind.

$a \setminus b$	0	1	2	3	4	5	6	7
0	8	4	4	4	4	4	4	4
1	4	6	4	2	6	4	6	4
2	4	4	2	2	4	4	2	6
3	4	6	2	4	2	4	4	2
4	4	6	6	4	4	6	2	4
5	4	4	2	6	6	6	4	4
6	4	2	4	2	4	6	4	2
7	4	4	4	4	6	2	2	2

(a) Die Approximationstabelle von S_1

$a \setminus b$	0	1	2	3	4	5	6	7
0	8	4	4	4	4	4	4	4
1	4	4	4	4	4	4	4	0
2	4	4	6	2	6	6	4	4
3	4	4	2	2	6	2	4	4
4	4	6	4	2	2	4	2	4
5	4	6	4	6	6	4	2	4
6	4	2	2	4	4	6	2	4
7	4	6	2	4	4	6	6	4

(b) Die Approximationstabelle von S_2

Abbildung 10: Die Approximationstabellen der S-Boxen von SPN_{bsp}

Das bedeutet, der Wert für die Linearkombination $X_2 \oplus Y_0 \oplus Y_1 \oplus Y_2$ befindet sich in Abbildung 10(b) an der Stelle $a = 1$ und $b = 7$.

Die beiden in Abbildung 9 aufgelisteten Approximationen lassen sich mit Hilfe der Bemerkung 4.1 auf SPN_{bsp} ausweiten, also gilt:

$$\begin{aligned} L_2^0 :=& U_4^0 \oplus V_4^0 = X_4 \oplus K_4^0 \oplus V_4^0 = X_4 \oplus K_4^0 \oplus K_5^1 \oplus U_5^1, \\ L_2^1 :=& U_5^1 \oplus V_3^1 \oplus V_4^1 \oplus V_5^1 \\ =& U_5^1 \oplus U_3^2 \oplus K_3^2 \oplus U_5^2 \oplus K_5^2 \oplus U_2^2 \oplus K_2^2. \end{aligned}$$

Also ist eine lineare Approximation A von SPN_{bsp} ausschließlich der letzten Runde gefunden, die lediglich aus Eingabezeichen, Schlüsselzeichen und Zeichen vor der letzten Runde

besteht, denn nach Bemerkung 4.1:

$$\begin{aligned} A =& L_2^0 \oplus L_2^1 = X_4 \oplus K_4^0 \oplus K_5^1 \oplus U_5^1 \oplus U_5^1 \oplus U_3^2 \oplus K_3^2 \oplus U_5^2 \oplus K_5^2 \oplus U_2^2 \oplus K_2^2 \\ =& X_4 \oplus U_2^2 \oplus U_3^2 \oplus U_5^2 \oplus \underbrace{K_4^0 \oplus K_5^1 \oplus K_2^2 \oplus K_3^2 \oplus K_5^2}_{=:k}. \end{aligned}$$

Der Bias dieser Linearkombination lässt sich leicht mit Hilfe des Piling-Up-Lemmas (Lemma 3.5), wie in Gleichung (16), berechnen:

$$\epsilon(A) = \epsilon(L_2^0 \oplus L_2^1) = 2 \cdot \epsilon(L_2^0) \cdot \epsilon(L_2^1) = 2 \cdot \frac{1}{4} \cdot (-\frac{1}{2}) = -\frac{1}{4}. \tag{18}$$

Nach Gleichung (17) gilt also:

$$\epsilon(X_4 \oplus U_2^2 \oplus U_3^2 \oplus U_5^2) = \begin{cases} -\frac{1}{4} & \text{für } k = 0 \\ \frac{1}{4} & \text{für } k = 1 \end{cases}.$$

Also ist eine Linearkombination, lediglich aus Klartextzeichen, Schlüsselzeichen und Zeichen vor der letzten Substitution bestehend mit einem betragsmäßig hohem Bias gefunden, sodass mit der Methode 2 (Lemma 4.2) das SPN_{bsp} angegriffen werden kann. Dafür werden für alle möglichen $2^6 = 64$ Klartext-Geheimtext-Paare (x, y) jeweils der Geheimtext y mit Hilfe von einem der $2^6 = 64$ potentiellen Rundenschlüsseln der letzte Runde entschlüsselt und für das Ergebnis u^2 überprüft, ob $x_4 \oplus u_2^2 \oplus u_3^2 \oplus u_5^2 = 0$ ist. In diesem Fall wird ein Zähler für den Rundenschlüssel, der für die Entschlüsselung der letzten Runde benutzt wurde, erhöht.

Wird dies für alle 64 Klartext-Geheimtext-Paare durchgeführt, so ergeben sich die in Abbildung 11 aufgelisteten Schlüssel als die wahrscheinlich richtigen.

idx	K^3	Zähler c	$\lvert c - \frac{64}{2} \rvert$
0	000 011	48	16
1	000 110	16	16
2	101 011	16	16
3	101 110	48	16

Abbildung 11: Linearer Angriff: Wahrscheinlichste Schlüsselzeichen von SPN_{bsp}

Nun besteht die Möglichkeit, mit diesen vier Schlüsseln die letzte Runde zu entschlüsseln und dasselbe Verfahren auf die vorletzte Runde anzuwenden. Wegen des sehr einfachen

Keyschedulings sind schon drei Zeichen von K^2 durch die Wahl des letzten Rundenschlüssels festgelegt, was die Möglichkeiten weiter einschränkt. Außerdem ist durch den zweiten Teil des Lemmas 4.2 eine weitere Information bekannt, die in die Wahl für die richtigen Schlüssel einbezogen werden kann. Da wegen Gleichung (18) gilt: $p = P[A = 0] = \frac{1}{4} < \frac{1}{2}$, kann angenommen werden, dass:

$$k = \begin{cases} 1 & \text{für } K^3 = 000\,011 \text{ oder } K^3 = 101\,110 \\ 0 & \text{für } K^3 = 000\,110 \text{ oder } K^3 = 101\,011 \end{cases}.$$

Auf diese Weise lässt sich die zu testende Schlüsselmenge noch weiter eingrenzen, aber schon das vorgestellte Grundverfahren liefert, dass statt $2^{15} = 32768$ Schlüsseln lediglich noch $2^9 + 4 = 516$ Schlüssel getestet werden müssen.

4.3 Differentielle Kryptoanalyse

Bei der differentiellen Kryptoanalyse handelt es sich um einen Angriff mit gewähltem Klartext, der schon vor der linearen Kryptoanalyse entdeckt worden und dieser in vielen Teilen sehr ähnlich ist. Der Hauptunterschied zwischen der von Eli Biham und Adi Shamir im Jahr 1990 publizierten differentiellen Kryptoanalyse [BS90] und der linearen ist, dass die differentielle mit der Differenz zweier Klartexte und der Differenz der zugehörigen Geheimtexte arbeitet und nicht wie die lineare direkt auf den Klartext-Geheimtext-Paaren [Sti06]. Dafür haben beide Verfahren gemein, dass sie auf jedwede iterierte Block-Chiffre angewandt werden können [BS90] und auch sonst wird im weiteren Verlauf die Verwandtschaft dieser Verfahren deutlich.

Im Folgenden wird, wie schon bei der linearen Kryptoanalyse, vorerst die Art des Angriffs allgemein vorgestellt und daraufhin das Verfahren anhand eines Beispiel-Angriffs auf das SPN aus Abschnitt 2.2.6 verdeutlicht.

4.3.1 Basis-Angriff

Die Idee der differentiellen Kryptoanalyse basiert ebenso wie die der linearen Kryptoanalyse darauf, dass eine Schwäche in den S-Boxen aufgedeckt und diese auf das SPN ausgeweitet und ausgenutzt wird. Im Weiteren wird zuerst die mögliche Schwäche einer S-Box aufgezeigt und daraufhin verdeutlicht, wie dieser Zusammenhang auf das SPN ausgeweitet und dieses damit angegriffen werden kann.

Sei

$$S : \mathbb{Z}_2^m \to \mathbb{Z}_2^n,$$

mit $n, m \in \mathbb{N} \setminus \{0\}$ eine S-Box. Diese S-Box weist eine Schwäche auf, wenn für eine Eingabedifferenz $x' \in \mathbb{Z}_2^m$ von S die zu den Paaren aus $\Delta(x')$ zugehörigen Ausgabedifferenzen nicht gleichverteilt sind. Also definiere:

Definition 4.2 (diff-schwach) *Eine S-Box $S : \mathbb{Z}_2^m \to \mathbb{Z}_2^n$ mit $n, m \in \mathbb{N} \setminus \{0\}$ heißt **angreifbar im Sinne der differentiellen Kryptoanalyse**, wenn*

$\exists\, x' \in \mathbb{Z}_2^m$ *und* $\exists\, y' \in \mathbb{Z}_2^n$:

$$N_D(x', y') := |\{(x, x^*) \in \Delta(x') \mid y' = S(x) \oplus S(x^*)\}| > 1.$$

*Falls eine S-Box in diesem Sinne angreifbar ist, dann nenne diese S-Box auch **diff-schwach**. Je größer $N_D(x', y')$, desto diff-schwächer ist die S-Box.* ∘

Es wird von einer Schwäche gesprochen, weil $N_D(x', y') > 1$ bedeutet, dass, wenn eine Differenz x' von zwei Klartextpaaren als Eingabe in die S-Box genommen wird, die zugehörige Differenz der Ausgaben der S-Box nicht zufällig ist, sondern die Wahrscheinlichkeit, dass die Ausgabedifferenz y' ist, bei $R_p(x', y') = \frac{N_D(x', y')}{|\Delta(x')|}$ liegt. Also ergibt sich für diese Eingabedifferenz $x' = x \oplus x^*$, mit $x, x^* \in \mathbb{Z}_2^m$ eine wahrscheinliche Linearität der S-Box

$$S(x') = S(x \oplus x^*) = y' = S(x) \oplus S(x^*)$$

und dieser Zusammenhang lässt sich dann für alle $(x, x^*) \in \Delta(x')$ anstelle der S-Box selbst nehmen.

Folgendes Lemma hilft bei der Berechnung der Wahrscheinlichkeiten R_p:

Lemma 4.3 *Sei $S : \mathbb{Z}_2^m \to \mathbb{Z}_2^n$ mit $n, m \in \mathbb{N} \setminus \{0\}$ eine S-Box und $x' \in \mathbb{Z}_2^m$ eine Eingabedifferenz, dann gilt:*

(i) $\Delta(x') = \{(x, x \oplus x') \mid x \in \mathbb{Z}_2^m\}$,

(ii) $|\Delta(x')| = 2^m$. ∘

Beweis:
Sei $S : \mathbb{Z}_2^m \to \mathbb{Z}_2^n$ mit $n, m \in \mathbb{N} \setminus \{0\}$ eine S-Box und $x' \in \mathbb{Z}_2^m$ eine Eingabedifferenz.

(i)

$$\begin{aligned} \Delta(x') &\overset{Def.}{=} \{(x, x^*) \mid x \oplus x^* = x' \text{ mit } x, x^* \in \mathbb{Z}_2^m\} \\ &\overset{x\oplus}{=} \{(x, x^*) \mid x^* = x \oplus x' \text{ mit } x, x^* \in \mathbb{Z}_2^m\} \\ &= \{(x, x \oplus x') \mid x \in \mathbb{Z}_2^m\}. \end{aligned}$$

(ii) Folgt direkt aus (i).

Damit sind beide Teile des Lemmas bewiesen. ■

Für jede S-Box lassen sich für alle möglichen Eingabedifferenzen die möglichen Bestandteile berechnen (Lemma 4.3 (i)), diese einzeln mit der S-Box abbilden und damit die zugehörigen Ausgabedifferenzen erstellen; beispielhaft für zwei Eingabedifferenzen zu sehen in Abbildung 12. Aus diesen Auflistungen können durch Abzählen direkt die N_D erhalten werden, welche häufig gesammelt und übersichtlich in einer Tabelle eingetragen werden. Solch eine Tabelle wird auch als **Tabelle der Differenzenverteilung** bezeichnet; beispielhaft zu sehen in Abbildung 13. Da in den Zeilen alle Eingabedifferenzen und in den Spalten alle Ausgabedifferenzen eingetragen werden, gilt, dass die Summe der Einträge jeder Zeile gleich der Summe der Einträge jeder Spalte gleich 2^m ist. Des Weiteren müssen alle Einträge gerade ganze Zahlen sein, da die Differenz symmetrisch ist, das heißt $\Delta(x, y) = x \oplus y = y \oplus x = \Delta(y, x)$ für zwei Eingaben oder Ausgaben x, y einer S-Box. Da jedoch eine Eingabedifferenz von Null lediglich für zwei gleiche Eingabewerte erreicht wird, folgt direkt, dass in diesem Fall die zugehörigen Ausgabedifferenzen auch Null sein müssen, da die S-Box eine Abbildung ist. Damit steht im oberen linken Feld der Tabelle immer der Wert 2^m und alle weiteren Felder der ersten Spalte und der ersten Zeile sind auf Grund der eben erwähnten Summe der Elemente mit 0 belegt. Daraus folgt jedoch direkt, dass es eine nicht schwache S-Box, also eine S-Box, deren Ausgabedifferenzen gleichverteilt sind (die Wahrscheinlichkeit für jede Ausgabedifferenz liegt bei $\frac{1}{2^m}$), mathematisch nicht existiert. Jedoch erschweren S-Boxen, deren Ausgabedifferenzen nahezu gleichverteilt sind, die differentielle Krypoanalyse in hohem Maße [Hey02].

Diese Schwäche muss auf das Netz ausgeweitet werden, sodass für eine Eingabedifferenz $x' \in \mathbb{Z}_2^l$ ($l \in \mathbb{N}$) des SPNes die zugehörigen Ausgabedifferenzen nicht zufällig verteilt sind, denn dann lässt sich das Netz, ähnlich wie bei der linearen Kryptoanalyse, angreifen: Angenommen der Angreifer besitzt eine Menge von 4-Tupeln (x, x^*, y, y^*), mit $x \oplus x^* = x'$ und x, x^* sind Klartexte, die mit ein und demselben unbekannten Schlüssel K verschlüsselt wurden und damit zu den zugehörigen Geheimtexten y und y^* gelangt wird, dann wird die letzte Runde für alle diese y und y^* mit Hilfe jedes Wortes, zusammengesetzt aus den relevanten, potentiellen Schlüsselzeichen des letzten Rundenschlüssels, partiell entschlüsselt und die Differenz dieser Werte mit den wahrscheinlichsten Werten, erhalten aus der Ausweitung der Schwäche auf das SPN, verglichen. Falls diese Differenzen übereinstimmen, wird für die Schlüsselzeichen, welche für diese partielle Entschlüsselung genutzt wurden, ein Zähler erhöht. Das Wort mit dem höchsten Zähler wird daraufhin als das Wort mit den richtigen Schlüsselzeichen angenommen.

Um die Schwäche auf das SPN ausweiten zu können, müssen die Differenzen über die Transposition und die Schlüsseladdition hinaus verfolgt werden, damit die Eingabedifferenz der nächsten S-Box bekannt ist. Wenn solch eine Verbindung zwischen den einzelnen diff-schwachen S-Boxen gefunden wurde, so wird auch von einem **Weg** der Differenzen durch das SPN gesprochen. Da die Transposition lediglich eine bekannte Umsortierung der Zeichen ist, stellt diese kein Problem dar. Da jedoch die Tabelle der Differenzenverteilung, wie in Lemma 4.4 zu sehen ist, unabhängig davon ist, ob für die Eingabedifferenzen die reinen Eingaben für die S-Box genutzt werden oder die einzelnen Bestandteile der Differenzen vorher noch mit Rundenschlüsselzeichen addiert werden, kann die Schlüsseladdition an dieser Stelle unberücksichtigt bleiben und somit lässt sich der Weg der Differenzen durch das SPN verfolgen. Anschaulich wird dieser Weg an einem Beispiel in Abbildung 15 verdeutlicht, wobei die Pfeile die Einsen der Differenzen kennzeichnen, die in diesem Angriff involviert sind. S-Boxen, von denen solche Pfeilen abgehen, werden wie in der linearen Kryptoanalyse **aktive S-Boxen** genannt. Alle weiteren Zeichen, die nicht direkt in den Angriff involviert sind, werden als Null angenommen.

Folgendes Lemma zeigt, dass die Schlüsselzeichen bei der Betrachtung der Differenzen keinen Einfluss auf die Tabelle der Differenzenverteilung einer S-Box haben.

Lemma 4.4 *Seien $x, x^* \in \mathbb{Z}_2^m$ Eingaben einer S-Box $S : \mathbb{Z}_2^m \to \mathbb{Z}_2^n$, mit einer Differenz $x' := x \oplus x^*$ und sei $k \in \mathbb{Z}_2^m$. Damit definiere $w := x \oplus k$ und $w^* := x^* \oplus k$, mit zugehöriger Differenz $w' := w \oplus w^*$, dann gilt:*

$$x' = x \oplus x^* = x \oplus x^* \oplus \underbrace{k \oplus k}_{=0} = x \oplus k \oplus x^* \oplus k = w \oplus w^* = w'.$$

∘

Es wird also ein Weg durch das SPN gesucht, indem die einzelnen aktiven S-Boxen eine möglichst hohe Wahrscheinlichkeit für eine Ausgabedifferenz zu der durch die S-Box und die Transposition der vorherigen Runde vorgegebene Eingabedifferenz aufweist. Auch wenn diese einzelnen Wahrscheinlichkeiten nicht stochastisch unabhängig voneinander sind, lässt sich dies ebenso wie bei der linearen Kryptoanalyse annehmen, sodass die einzelnen Wahrscheinlichkeiten aufmultipliziert werden können und das Ergebnis in der Praxis ein gutes Gütekriterium für diesen Angriff liefert [Hey02].

Ist solch ein Weg gefunden, dann lässt sich dieser Zusammenhang mit folgendem Lemma ausnutzen:

Lemma 4.5 *Sei $l \in \mathbb{N}$ die Blocklänge des anzugreifenden SPNes und $n \in \mathbb{N}$ die Rundenanzahl, sowie e_K die Verschlüsselungsfunktion. Außerdem sei $K^n_{(i)} \in \mathbb{Z}_2^l$ mit $0 \leq i < 2^l$ einer der 2^l potentiellen letzten Rundenschlüssel und sei weiterhin $(K^n_{(i)})_J$*

mit $J \subseteq \{0, 1 \ldots, l-1\}$ ein Wort aus den für die partielle Entschlüsselung der letzten Runde relevanten Schlüsselzeichen des i-ten Rundenschlüssels. Sei weiterhin $x' \in \mathbb{Z}_2^l$ die Eingabedifferenz, für die mit hoher Wahrscheinlichkeit $(u^{n-1})' \in \mathbb{Z}_2^l$ die Ausgabedifferenz des SPNes vor der letzten Substitution ist und sei außerdem $(x, x^*) \in \Delta(x')$ und damit $y := e_K(x)$ sowie $y^* := e_K(x^*)$, dann bezeichne $u_i^{n-1} \in \mathbb{Z}_2^l$ beziehungsweise $(u_i^{n-1})^* \in \mathbb{Z}_2^l$ das Wort, welches durch die partielle Entschlüsselung der letzten Runde von y beziehungsweise y^* mit Hilfe der potentiellen Schlüsselzeichen $(K_{(i)}^n)_J$ erhalten wurde, wobei $(u_i^{n-1})_j = ((u_i^{n-1})^*)_j = 0$ für $j \in \{0, \ldots, l-1\} \setminus J$. Definiere dazu die Menge $T_i = \{(x, x^*) \in \Delta(x') \mid u_i^{n-1} \oplus (u_i^{n-1})^* = (u_i^{n-1})'\}$, dann nehme an:

Die zu $max\{T_i \mid 0 \leq i < l\}$ gehörigen Schlüsselzeichen sind die gesuchten. ◦

Wenn also eine Eingabedifferenz x' des SPNes gefunden wurde, sodass mit einer hohen Wahrscheinlichkeit die Ausgabedifferenz vor der letzten Substitution $(u^{n-1})'$ ist, dann wird für alle Klartexte x, x^* mit $x \oplus x^* = x'$ die letzte Runde des SPNes partiell entschlüsselt und wenn deren Differenz häufig $(u^{n-1})'$ ergibt, kann davon ausgegangen werden, dass die letzte Runde richtig entschlüsselt wurde und somit die für diese Entschlüsselung benutzten Zeichen des letzten Rundenschlüssels die gesuchten darstellen.

4.3.2 Beispiel-Angriff auf ein Substitutions-/Permutations-Netz

Dieser Abschnitt zeigt einen differentiellen Angriff auf das in Beispiel 2.3 vorgestellte Substitutions-/Permutations-Netz SPN_{bsp} und verdeutlicht damit die im letzten Abschnitt vorgestellten Verfahren. Als Ergebnis dieses Angriffes wird der komplette letzte Rundenschlüssel von SPN_{bsp} geliefert.

Für diesen Angriff müssen zuerst, wie im vorangegangenen Abschnitt gesehen, diff-schwache S-Boxen, das heißt S-Boxen, für die eine Eingabedifferenz existiert gefunden werden, sodass für diese Eingabedifferenz die Ausgabedifferenzen der S-Box nicht gleichverteilt sind, damit daraufhin ein Weg durch das SPN mit Hilfe dieser S-Boxen erstellt werden kann, sodass schließlich eine Eingabedifferenz für SPN_{bsp} existiert, für die die Ausgabedifferenzen des SPNes nicht gleichverteilt sind. Das heißt zuerst müssen die S-Boxen hinsichtlich ihrer Eingabedifferenzen untersucht werden, sodass möglichst eine Eingabedifferenz gefunden wird, für die eine Ausgabedifferenz existiert, die eine hohe Wahrscheinlichkeit für ihr Auftreten besitzt.

Zur Analyse der einzelnen S-Boxen wurde für jede Eingabedifferenz x' jeder S-Box S die Menge $\Delta(x')$ der Paare von Wörtern (x, x^*), deren Differenz x' ist, erstellt und damit alle Ausgabedifferenzen $y' = S(x) \oplus S(x^*)$ berechnet. Eine vollständige Tabelle dieser

Differenzen befindet sich im Anhang im Abschnitt A.3. Da für diesen Angriff lediglich zwei aktive S-Boxen benötigt werden, sind in Abbildung 12(a) die für diesen Angriff relevanten Differenzen der S-Box S_1 und in Abbildung 12(b) die relevanten Differenzen von S_2 zu sehen.

$\Delta(001)$				
x	x^*	y	y^*	y'
000	001	010	101	111
001	000	101	010	111
010	011	000	100	100
011	010	100	000	100
100	101	110	011	101
101	100	011	110	101
110	111	001	111	110
111	110	111	001	110

(a) Die relevanten Differenzen von S_1

$\Delta(010)$				
x	x^*	y	y^*	y'
000	010	010	100	110
001	011	101	110	011
010	000	100	010	110
011	001	110	101	011
100	110	001	111	110
101	111	000	011	011
110	100	111	001	110
111	101	011	000	011

(b) Die relevanten Differenzen von S_2

Abbildung 12: Die relevanten Differenzen der S-Boxen von SPN_{bsp}

Die aus den vollständigen Tabellen (Abbildung 23) direkt abgezählten Werte für die Differenzenverteilung N_D sind in Abbildung 13 zu sehen, wobei die Eingabedifferenzen (a'), aufgetragen in den Zeilen, und die Ausgabedifferenzen (b'), aufgetragen in den Spalten, dezimalcodiert dargestellt werden.

$a' \setminus b'$	0	1	2	3	4	5	6	7
0	8	0	0	0	0	0	0	0
1	0	0	0	0	2	2	2	2
2	0	2	2	0	2	0	0	2
3	0	2	2	0	0	2	2	0
4	0	2	0	2	2	0	2	0
5	0	2	0	2	0	2	0	2
6	0	0	2	2	0	0	2	2
7	0	0	2	2	2	2	0	0

(a) Die Tabelle der Differenzenverteilung von S_1

$a' \setminus b'$	0	1	2	3	4	5	6	7
0	8	0	0	0	0	0	0	0
1	0	2	2	0	2	0	0	2
2	0	0	0	4	0	0	4	0
3	0	2	2	0	2	0	0	2
4	0	0	0	4	0	4	0	0
5	0	2	2	0	2	0	0	2
6	0	0	0	0	0	4	4	0
7	0	2	2	0	2	0	0	2

(b) Die Tabelle der Differenzenverteilung von S_2

Abbildung 13: Die Tabellen der Differenzenverteilung der S-Boxen von SPN_{bsp}

Die Anzahl, wie häufig für die Eingabedifferenz 010 in S-Box S_2 die Ausgabedifferenz 110 erhalten wird, befindet sich also in der Zeile $a' = 2$ und $b' = 6$ der Abbildung 13(b).

S-Box	Wahrscheinlichkeit
S_1^0	$R_p(001, 100) = \frac{2}{8} = \frac{1}{4}$
S_2^1	$R_p(010, 011) = \frac{4}{8} = \frac{1}{2}$

Abbildung 14: Bedingte Wahrscheinlichkeiten R_p der aktiven Differenzen

Damit lassen sich die Wahrscheinlichkeiten $R_p(x', y')$ für das Auftreten einer Ausgabedifferenz y' unter der Bedingung, dass die Eingabedifferenz x' ist, für die in diesem Angriff aktiven S-Boxen berechnen. Als Differenzen für die erste S-Box wähle $a_1' = 001$, $b_1' = 100$ und für S_2 sind die in diesem Angriff genutzten Differenzen $a_2' = 010$ und $b_2' = 011$. Damit ergeben sich die bedingten Wahrscheinlichkeiten für diese Differenzen zu $\frac{1}{4}$ beziehungsweise $\frac{1}{8}$; zu sehen in Abbildung 14.

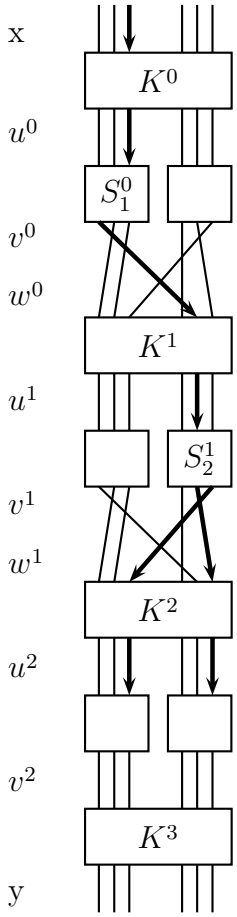

Abbildung 15: Differentieller Angriff auf SPN_{bsp}

Dass nun gerade diese Eingabe- und Ausgabedifferenzen betrachtet wurden, liegt daran, dass sich aus diesen ein Weg (Abbildung 15) durch das SPN für die ersten beiden Runden erstellen lässt, sodass sich für eine Eingabedifferenz des SPNes eine Ausgabedifferenz

angeben lässt, die eine hohe Wahrscheinlichkeit für ihr Auftreten hat. Dieser Weg von Differenzen diff-schwacher S-Boxen durch das SPN wurde mit Hilfe der in Abschnitt 5 vorgestellten Heuristik gefunden.

Die Differenzen a'_1 und b'_2 können also, wie in Abschnitt 4.3.1 erwähnt mit Nullen erweitert werden, sodass die Eingabedifferenz 001 000 für das SPN erhalten wird, für die die Wahrscheinlichkeit, dass die Ausgabedifferenz nach der zweiten Substitution 000 011 ist, wie folgt berechnet werden kann:

$$R_p(001\,000, 000\,011) = \frac{1}{4} \cdot \frac{1}{2} = \frac{1}{8}.$$

Das heißt, mit den Bezeichnungen aus Abbildung 15, dass für eine Eingabedifferenz $x' = 001\,000$ die Wahrscheinlichkeit dafür, dass die Ausgabedifferenz nach der zweiten Substitution $(v^1)' = 000\,011$ ist, bei $\frac{1}{8}$ liegt. Da aber wie schon bei der Ausweitung der Schwäche einer S-Box auf das SPN gezeigt

$$(v^1)' = 000\,011 \nrightarrow (u^2)' = 001\,001$$

gilt, heißt das, dass folgende Schlussfolgerung mit der Wahrscheinlichkeit $\frac{1}{8}$ gilt:

$$x' = 001\,000 \nrightarrow (u^2)' = 001\,001,$$

wobei x' eine Eingabedifferenz von SPN_{bsp} ist und $(u^2)'$ eine Eingabedifferenz der letzten Runde darstellt.

Somit kann mit Hilfe des Lemmas 4.5 der Angriff auf SPN_{bsp} durchgeführt werden. Das heißt, es werden alle $2^6 = 64$ Paare der Menge $\Delta(001\,000)$ erstellt und daraufhin die Menge $T := \{(x, x^*, y, y^*) \mid (x, x^*) \in \Delta(001\,000)$ und y, y^* die zugehörigen Geheimtexte$\}$ berechnet. Die zugehörigen y beziehungsweise y^* müssen bekannt sein, deswegen handelt es sich auch um einen Angriff mit gewähltem Klartext. Nun wird die letzte Runde mit jedem der $2^6 = 64$ potentiellen Schlüssel der letzten Runde für jedes der y und y^* aus T entschlüsselt und überprüft, ob $(u^2) \oplus (u^2)^* = (u^2)'$ ist. Ist dies der Fall, so wird für den potentiellen Schlüssel, der für diese Entschlüsselung genutzt wurde, ein Zähler erhöht. Bei der Anwendung dieses Algorithmus bekommt der richtige Schlüssel der letzten Runde ($K^3 = 101\,110$) im Schnitt nach 46 der 64 4-Tupel aus T den höchsten Zählerwert und gibt diesen nicht mehr ab. Wie im Abschnitt 5 zu sehen ist, ist die Anzahl der benötigten 4-Tupel, um den besten Schlüssel zu identifizieren, noch beeindruckender, wenn die Größe des diff-schwachen SPNes steigt. Auch wenn dieses Ergebnis auf den ersten Blick besser als das der linearen Kryptoanalyse scheint, darf nicht außer Acht gelassen werden, dass es

sich bei diesem Angriff um einen Angriff mit gewähltem Klartext handelt, sodass starke Anforderungen an die Daten gestellt werden, die der Angreifer benötigt.

5 Implementierung

Das im Folgenden beschriebene Werkzeug dient der Erstellung von beliebigen SPNen und bietet die Möglichkeit, diese Netze mit der linearen und der differentiellen Kryptoanalyse zu untersuchen.

Geschrieben wurde das Programm in C++ und das Graphical User Interface (GUI) wurde mit Hilfe von Qt[8] und nach den Richtlinien des Tango Desktop Projektes[9] realisiert, woher auch, in teilweise etwas abgewandelter Form, die Icons stammen. Des Weiteren wurde die Bibliothek Poppler[10] für die Visualisierung von PDF-Dokumenten genutzt.

Abbildung 16 zeigt die Eingabemaske für die Einstellungen eines SPN. Die Permutation und die einzelnen S-Boxen lassen sich zufällig erstellen, permutieren oder auch manuell eingeben; mit „Standardwerte“ lässt sich das SPN aus [Hey02] erstellen.

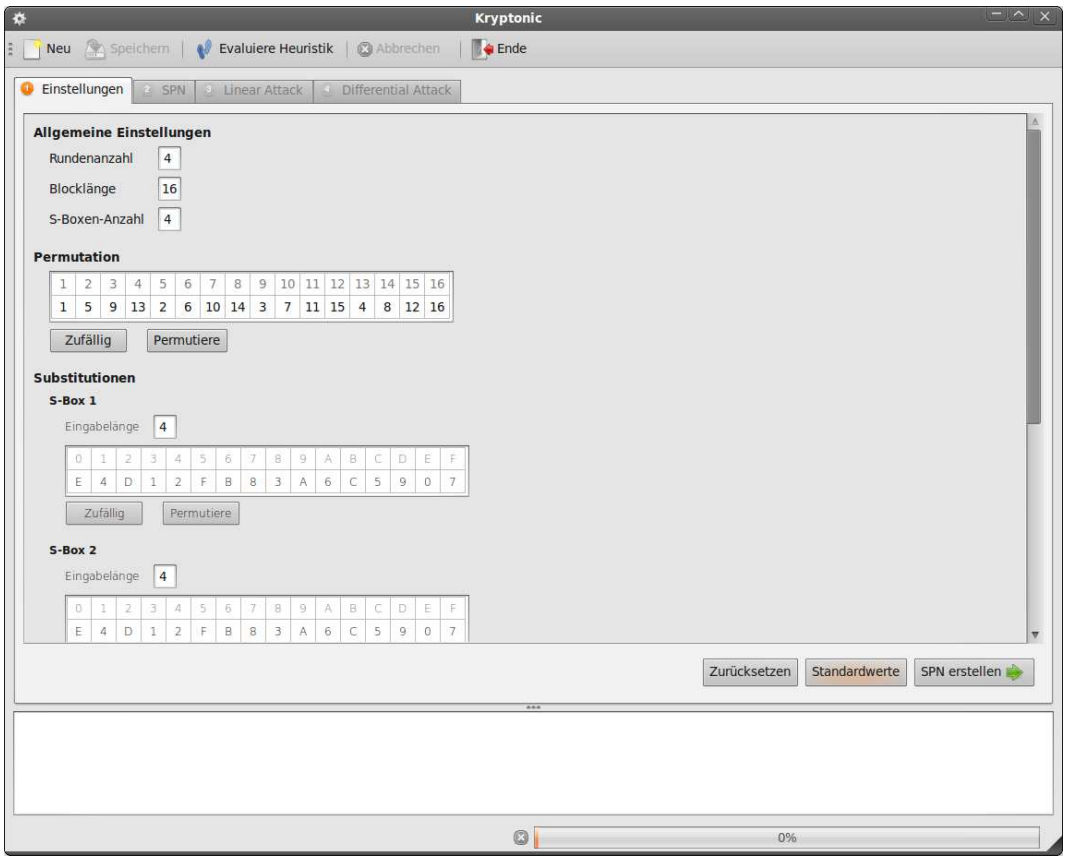

Abbildung 16: Eingabemaske SPN-Einstellungen

Nach dem Erstellen eines SPNes, wird dieses Netz angezeigt und die Möglichkeit geboten,

[8] http://qt.nokia.com/products/.
[9] http://tango.freedesktop.org/Tango_Desktop_Project.
[10] http://poppler.freedesktop.org/.

mit Hilfes dieses Netzes, Zeichenketten $x \in \mathbb{Z}_2^l$ ($l \in \mathbb{N}$) zu ver- beziehungsweise entschlüsseln. Dabei wird beim Erstellen des Netzes ein zufälliger Hauptschlüssel erstellt und mit Hilfe des Simple-Keyscheduling werden die für dieses Netz benötigten Rundenschlüssel berechnet. Für die Ver- beziehungsweise Entschlüsselung wird, sofern benötigt, das in Definition 2.2 vorgestellte Padding benutzt. Beispielhaft ist diese Eingabemaske für das SPN aus [Hey02] in Abbildung 17 zu sehen. Das Textfeld im unteren Bereich des Fensters wird durchgängig als eine Art Ausgabekonsole genutzt, in der alle relevanten Informationen über die gerade ausgeführten Aktionen ausgegeben werden.

In dem Hauptfenster befindet sich ein rudimentärer PDF-Betrachter, in dem ein PDF-Dokument angezeigt wird, welches einschließlich der Latex-Quellen gespeichert werden kann und die Visualisierung des Netzes, eine Beispielverschlüsselung, das Keyscheduling und die einzelnen Abbildungen beinhaltet. Durch einen Klick auf den Betrachter wird das Dokument in einem externen Betrachter (Ghostview[11]) gestartet.

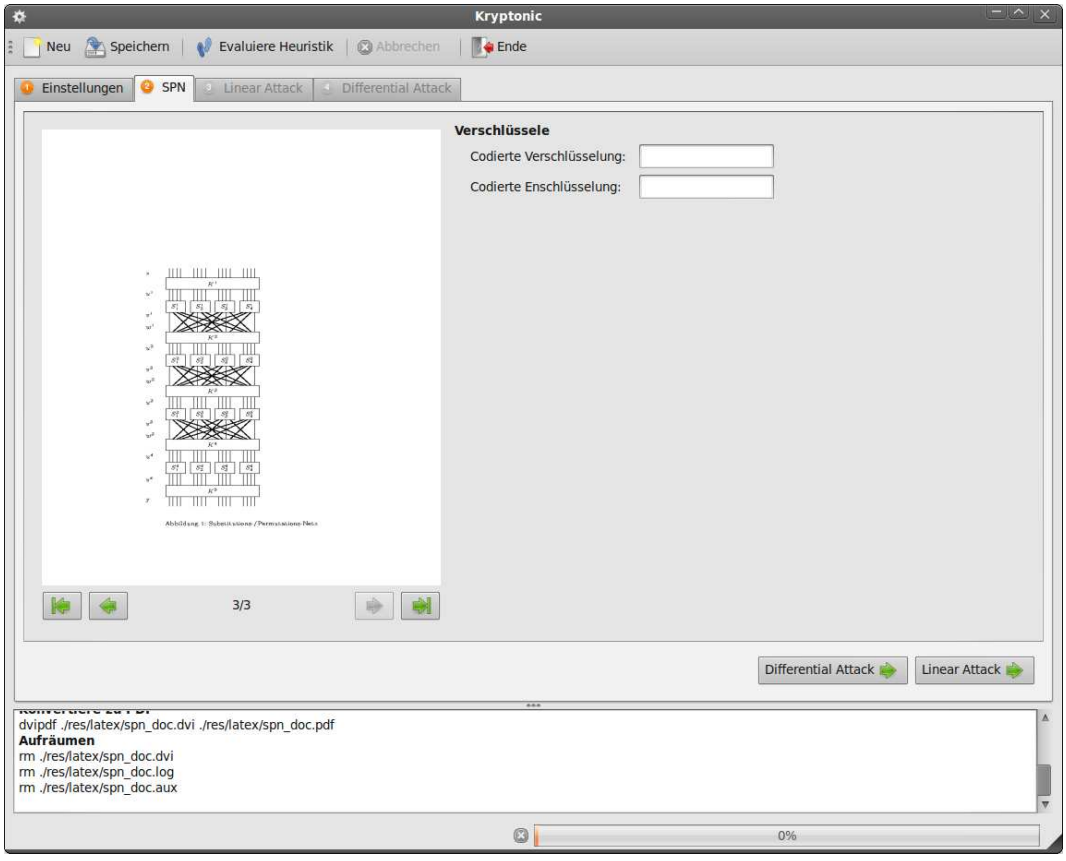

Abbildung 17: Anzeige des SPNes

In diesem Fenster bietet sich außerdem die Möglichkeit, das gegebene Netz mit der linearen oder auch der differentiellen Kryptoanalyse anzugreifen. Bei der linearen Krypto-

[11] http://pages.cs.wisc.edu/~ghost/.

analyse besteht zusätzlich die Möglichkeit, die Approximation des Netzes, das heißt das Finden einer Gleichung der Art (14) mit einem betragsmäßig hohen Bias, mit Hilfe einer Heuristik oder der Brute-Force-Suche durchzuführen. Aufgrund der hohen Rechen- und Speicherintensität empfiehlt sich die Brute-Force-Suche lediglich für sehr kleine Netze, da die Möglichkeiten exponentiell mit der Rundenanzahl des SPNes steigt. Bei der differentiellen Kryptoanalyse wird für die Suche nach einem Weg durch das Netz, der für eine Eingabedifferenz mit einer hohen Wahrscheinlichkeit eine Ausgabedifferenz liefert, automatisch ausschließlich die Heuristik genutzt, um die aktiven S-Boxen zu finden.

Die Heuristik für die lineare Kryptoanalyse erstellt zuerst eine Initial-Lösung, indem für die erste Runde eine S-Box aus einer Menge mit den lin-schwächsten S-Boxen als aktive S-Box gewählt wird und auf Grund deren linearer Approximation die aktiven S-Boxen der nächsten Runde bestimmt werden. Für diese Eingabesummen wird daraufhin die Ausgabesumme gewählt, sodass die daraus folgende Linearkombination den betragsmäßig höchsten Bias aufweist. Dieses Verfahren wird bis zur vorletzten Runde fortgeführt. Von den durch dieses Verfahren erstellten Angriffen wird der Angriff mit dem höchsten Bias, gewichtet durch die Anzahl der aktiven S-Boxen der letzten Runde, gewählt.

Daraufhin wird diese Initial-Lösung mit Hilfe eines heuristischen Optimierungsalgorithmus (Sintflut-Algorithmus [Wikb] oder Threshold-Accepting-Algorithmus [Wikd]) verbessert. Standardmäßig ist momentan der Sintflut-Algorithmus eingeschaltet, wobei auch der Threshold-Accepting-Algorithmus implementiert wurde und in einer zukünftigen Version der Implementierung im GUI wählbar ist. Diese Verfahren benötigen eine Nachbarlösung, mit welcher sie die bestehende beste Lösung vergleichen können und in Abhängigkeit von Parametern auch schlechtere Lösungen als neue beste Lösung zulassen, damit lokale Maxima überwunden werden können. Der Algorithmus zum Finden einer Nachbarlösung ist sehr einfach implementiert, indem zufällig eine aktive S-Box ausgewählt, deren Ausgabesumme zufällig verändert und daraufhin die lineare Approximations-Ausweitung bis zur vorletzten Runde wie bei der Initial-Lösung neu berechnet wird. Die Heuristik für die differentielle Kryptoanalyse ist mit minimalen Anpassungen analog implementiert.

Auch wenn die Heuristik relativ rudimentär implementiert wurde, bietet sie viele Einstellungsmöglichkeiten, mit denen die Ergebnisse beeinflusst werden können. Diese werden bislang noch nicht zum GUI durchgereicht, können aber einfach im Quell-Code geändert werden. Eine Hilfestellung dazu bietet die Möglichkeit mit dem Button „Evaluiere Heuristik“ die Heuristik auszuwerten, wobei eine Vielzahl von Werten für die Einstellungsmöglichkeiten ausprobiert und die Ergebnisse abschließend in einem PDF-Dokument dargestellt werden, welches direkt gespeichert werden kann.

Mit Hilfe der implementierten Heuristik ließ sich ein linearer Angriff auf das Beispiel-SPN

aus [Hey02] finden, welcher den Bias, wie in Abbildung 18 zu sehen ist, um den Faktor von ungefähr $2,5$ verbessert hat und damit im Schnitt rund 7000 Klartext-Geheimtext-Paare weniger benötigte, um zu den richtigen Schlüsselzeichen zu gelangen und dabei außerdem den kompletten letzten Rundenschlüssel anstatt lediglich acht Zeichen des letzten Rundenschlüssels geliefert hat.

	[Hey02]	Neu
Betrag des Bias	$\frac{1}{32} = 0,03125$	$\frac{81}{1024} = 0,0791016$
Anzahl benötigter Paare	≈ 8000	≈ 864
Anzahl Schlüsselzeichen	8	16

Abbildung 18: Vergleich Beispiel aus [Hey02] und verbesserter Angriff

Der neu gefundene Angriff verwendet die in Abbildung 19 zu sehenden fünf aktiven S-Boxen.

S-Box	Linearkombination	Bias
S_3^0	$L_3^0 = U_8^0 \oplus U_{11}^0 \oplus V_8^0$	$-\frac{1}{4}$
S_1^1	$L_1^1 = U_2^1 \oplus V_0^1 \oplus V_1^1 \oplus V_2^1$	$-\frac{3}{8}$
S_1^2	$L_1^2 = U_0^2 \oplus V_0^2 \oplus V_1^2 \oplus V_2^2 \oplus V_3^2$	$-\frac{3}{8}$
S_2^2	$L_2^2 = U_4^2 \oplus V_4^2 \oplus V_5^2 \oplus V_6^2 \oplus V_7^2$	$-\frac{3}{8}$
S_3^2	$L_3^2 = U_8^2 \oplus V_8^2 \oplus V_9^2 \oplus V_{10}^2 \oplus V_{11}^2$	$-\frac{3}{8}$

Abbildung 19: Aktive S-Boxen des verbesserten Angriffs

Damit ist die lineare Approximation des SPNes:

$$X_8 \oplus X_{11} \oplus U_0^3 \oplus U_1^3 \oplus U_2^3 \oplus U_4^3 \oplus U_5^3 \oplus U_6^3 \oplus U_8^3 \oplus U_9^3 \oplus U_{10}^3 \oplus U_{12}^3 \oplus U_{13}^3 \oplus U_{14}^3 \oplus k$$

mit $k := K_8^0 \oplus K_{11}^0 \oplus K_2^1 \oplus K_{11}^0 \oplus K_0^2 \oplus K_4^2 \oplus K_8^2 \oplus K_0^3 \oplus K_1^3 \oplus K_2^3 \oplus K_4^3 \oplus K_5^3 \oplus K_6^3 \oplus K_8^3 \oplus K_9^3 \oplus K_{10}^3 \oplus K_{12}^3 \oplus K_{13}^3 \oplus K_{14}^3$. Abbildung 20 zeigt eben diesen linearen Angriff in dem Werkzeug, wobei der linke PDF-Betrachter die Approximationstabellen und alle Linearkombinationen der Eingabe- und Ausgabesummen aller S-Boxen und der rechte PDF-Betrachter die Visualisierung des linearen Angriffs zeigt.

Das Werkzeug bietet die Möglichkeit, das SPN mit immer wieder neuen, zufällig erstellten Hauptschlüsseln anzugreifen oder aber mit ein und demselben Hauptschlüssel den Angriff mehrfach durchzuführen. Beiden Arten ist gemein, dass sie bei jedem neuen Angriff mit einer zufälligen Neuanordnung der Klartext-Geheimtext-Paare arbeiten. Die Werte aus

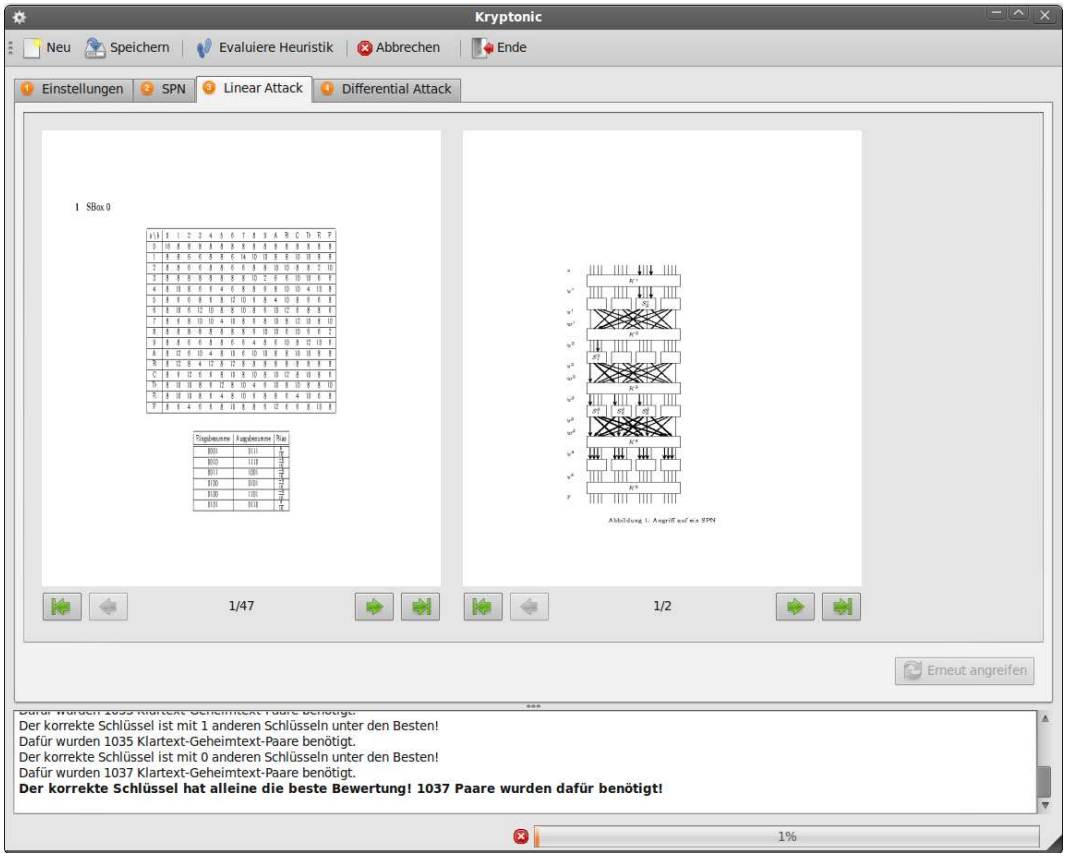

Abbildung 20: Linearer Angriff auf das SPN aus [Hey02]

Abbildung 18 arbeiten mit ein und demselben Hauptschlüssel aus [Hey02]. Aber auch bei zufälligen Schlüsseln verschlechtert sich der Wert lediglich auf rund 1100 Klartext-Geheimtext-Paare, die benötigt werden, bis der richtige letzte Rundenschlüssel die höchste Bewertung erhält und diese nicht wieder abgibt.

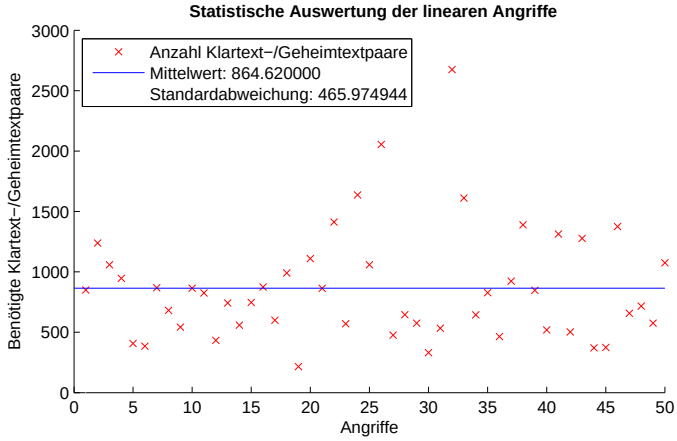

Abbildung 21: Statistik linearer Angriff auf das SPN aus [Hey02]

Der Wert für die durchschnittlich benötigte Anzahl von Klartext-Geheimtext-Paaren aus Abbildung 18 basiert auf einer Stichprobengröße von 50 Angriffen, die in Abbildung 21 eingetragen sind.

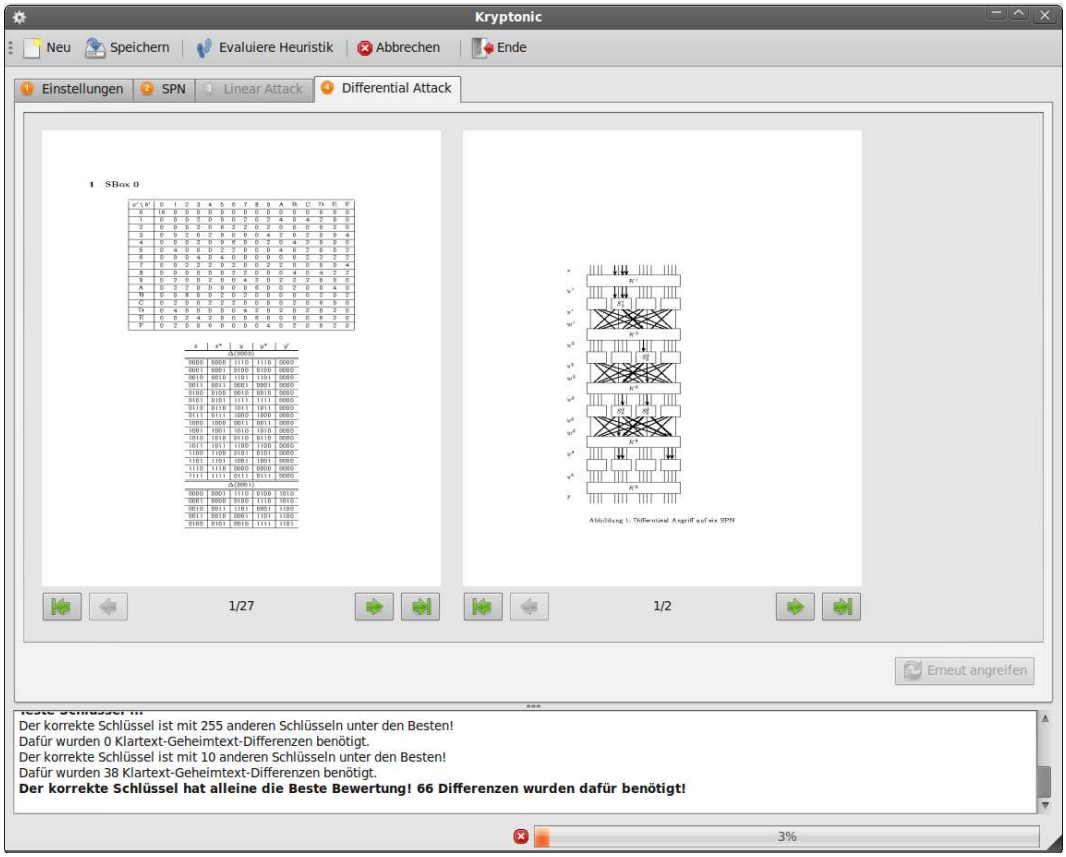

Abbildung 22: Differentieller Angriff auf das SPN aus [Hey02]

Die Abbildung 22 zeigt die differentielle Kryptoanalyse auf das SPN aus [Hey02]. In dem linken PDF-Betrachter sind die Differenzen-Distributions-Tabellen der S-Boxen und die einzelnen Differenzen nachzuschlagen und der rechte PDF-Betrachter zeigt die Visualisierung des Angriffes. Auch diese Dokumente lassen sich einschließlich der Latex-Quellen extern speichern.

Mit Hilfe der Heuristik wird bei der differentiellen Kryptoanalyse derselbe Angriff wie in [Hey02] gefunden, sodass auch nach circa 87 4-Tupeln die richtigen Schlüsselzeichen die beste Bewertung bekommen und diese nicht mehr abgeben.

6 Fazit und Ausblick

Im Entwicklungsprozess dieser Arbeit wurde ein Beispiel-SPN gefunden, welches so minimal ist, dass alle Grundberechnungen für die Analyseverfahren per Hand durchgeführt und vollständig abgedruckt werden können, was die Verständlichkeit der Verfahren erhöht. Ferner wurde ein Werkzeug zum Erstellen und Angreifen von SPNen entwickelt, mit dem die Möglichkeit besteht, weitere kleine Netze zu erstellen, sodass nach Belieben eigene Beispiele entworfen werden können und somit der Missstand behoben wurde, dass im wesentlichen bisher lediglich zwei relativ komplexe Beispiele – DES (zum Beispiel in [BS90][Mat94]) und das Beispiel in [Hey02] – für diese Analyseverfahren existieren. Auch wurden Schlussfolgerungen und Zusammenhänge, die in einschlägiger Literatur häufig als offensichtlich dargestellt werden oder gar keine Erwähnung finden, detailliert und umfassend erläutert. Insbesondere wurde die Verschlüsselung mit SPNen, einschließlich deren Anforderungen und Eigenschaften, sehr ausführlich dargestellt und untersucht.

Das Werkzeug bietet außerdem die Möglichkeit, den häufig zitierten linearen Angriff aus [Hey02] mit relativ einfachen Mitteln signifikant zu verbessern. An der Stelle besteht zwar die Möglichkeit die Implementierung weiter zu verbessern, jedoch war dies für die Untersuchungen in dieser Arbeit nicht notwendig. So können die heuristische Suche und die Optimierungsverfahren durch andere Parameter, neue Algorithmen zum Finden von Nachbarschaftslösungen oder auch zum Finden von Initial-Lösungen, weiter optimiert werden. Auch könnte der Angriff selbst parallelisiert werden, sodass die Laufzeit für die Schlüsselsuche weiter verkürzt werden könnte.

Auch wenn diese Arbeit schon einen tiefen Einblick in den Vorgang der linearen und differentiellen Kryptoanalyse liefert, bietet dieses Thema noch eine Vielzahl interessanter und weiterführender Aspekte, wovon einige im Folgenden erwähnt werden.

Ein Punkt ist die Komplexitätsanalyse der vorgestellten Verfahren und damit auch deren Verbesserungsmöglichkeiten, was zum Beispiel für die lineare Kryptoanalyse für den Algorithmus zum Bewerten der potentiellen Schlüsselzeichen der letzten Runde (Methode 2) in [CSQ07] getan wurde und mit Hilfe der Fast-Fourier-Transformation (FFT) eine signifikate Verbesserung erreicht wurde.

Ein weiterer interessanter Aspekt ist eine genauere Untersuchung, wie SPNe erstellt oder auch abgewandelt werden müssen, sodass sie resistent gegen die lineare und differentielle Kryptoanalyse sind. Zum Beispiel wird in [HT96] die Möglichkeit behandelt, größere S-Boxen mit hoher Diffusion zu wählen und die Transposition mit einer anderen geeigneten linearen Transformation zu ersetzen, sodass die Sicherheit der Netze gegenüber diesen Angriffen steigt. Ebenso werden in [CV95] Funktionen untersucht, die resistent ge-

genüber der linearen und der differentiellen Kryptoanalyse sind und der Zusammenhang aufgestellt, dass spezielle Funktionen, die resistent gegen die lineare, auch resistent gegen die differentielle Kryptoanalyse sind. Diese Ergebnisse basieren auf Untersuchungen zu Gemeinsamkeiten und Unterschieden der differentiellen und linearen Kryptoanalyse, was auch ein weiterer interessanter Ansatzpunkt für Untersuchungen darstellt und zum Beispiel zu einem Angriff ([LH94]), der beide Verfahren im gewissen Sinne vereint, führt.

Die Erfolge sowohl der linearen als auch der differentiellen Kryptoanalyse basieren auf Wahrscheinlichkeiten bestimmter Eigenschaften, sodass eine genauere Untersuchung der Erfolgswahrscheinlichkeit dieser Analyseverfahren und von welchen Parametern diese abhängt, wie sie zum Beispiel in [Sel07] betrachtet wurde, einen weiteren wissenswerten Aspekt darstellt.

Ebenso interessant ist die Betrachtung der Analyseverfahren auf größere Chiffren, wie es zum Beispiel in [Mat94] und [BS90] für den DES getan wurde und dabei zusätzlich noch einen genaueren Blick auf das Keyscheduling zu werfen, sodass möglicherweise der Schlüsselraum weiter eingeschränkt werden kann. Auch stellen die Untersuchungen der Entwickler von AES in [DR02], die den AES sicher gegen die bei der Entwicklung schon bekannte lineare und differentielle Kryptoanalyse gemacht haben, einen weiteren aufschlussreichen Aspekt dar.

A Anhang

A.1 Linearkombinationen der S-Box S_1 von SPN_{bsp}

Überschrift x bedeutet: $Eingabesumme_{10} * 2^3 + Ausgabesumme_{10} = x$

X_0	X_1	X_2	Y_0	Y_1	Y_2	0	1	2	3	4	5	6	7	8	9	10	11	12	13	14	15	16	17	18	19
0	0	0	0	1	0	0	0	1	1	0	0	1	1	0	0	1	1	0	0	1	1	0	0	1	1
0	0	1	1	0	1	0	1	0	1	1	0	1	0	1	0	1	0	0	1	0	1	0	1	0	1
0	1	0	0	0	0	0	0	0	0	0	0	0	0	0	0	0	0	0	0	0	0	1	1	1	1
0	1	1	1	0	0	0	0	0	0	1	1	1	1	1	1	1	1	0	0	0	0	1	1	1	1
1	0	0	1	1	0	0	0	1	1	1	1	0	0	0	0	1	1	1	1	0	0	0	0	1	1
1	0	1	0	1	1	0	1	1	0	0	1	1	0	1	0	0	1	1	0	0	1	0	1	1	0
1	1	0	0	0	1	0	1	0	1	0	1	0	1	0	1	0	1	0	1	0	1	1	0	1	0
1	1	1	1	1	1	0	1	1	0	1	0	0	1	1	0	0	1	0	1	1	0	1	0	0	1
Bias:						$\frac{4}{8}$	$\frac{0}{8}$	$\frac{0}{8}$	$\frac{0}{8}$	$\frac{0}{8}$	$\frac{0}{8}$	$\frac{0}{8}$	$\frac{0}{8}$	$\frac{0}{8}$	$\frac{2}{8}$	$\frac{0}{8}$	$\frac{-2}{8}$	$\frac{2}{8}$	$\frac{0}{8}$	$\frac{2}{8}$	$\frac{0}{8}$	$\frac{0}{8}$	$\frac{0}{8}$	$\frac{-2}{8}$	$\frac{-2}{8}$

Überschrift x bedeutet: $Eingabesumme_{10} * 2^3 + Ausgabesumme_{10} = x$

X_0	X_1	X_2	Y_0	Y_1	Y_2	20	21	22	23	24	25	26	27	28	29	30	31	32	33	34	35	36	37	38	39
0	0	0	0	1	0	0	0	1	1	0	0	1	1	0	0	1	1	0	0	1	1	0	0	1	1
0	0	1	1	0	1	1	0	1	0	1	0	1	0	0	1	0	1	0	1	0	1	1	0	1	0
0	1	0	0	0	0	1	1	1	1	1	1	1	1	1	1	1	1	0	0	0	0	0	0	0	0
0	1	1	1	0	0	0	0	0	0	0	0	0	0	1	1	1	1	0	0	0	0	1	1	1	1
1	0	0	1	1	0	1	1	0	0	0	0	1	1	1	1	0	0	1	1	0	0	0	0	1	1
1	0	1	0	1	1	0	1	1	0	1	0	0	1	1	0	0	1	1	0	0	1	1	0	0	1
1	1	0	0	0	1	1	0	1	0	1	0	1	0	1	0	1	0	1	0	1	0	1	0	1	0

1	1	1	1	1	1	0	1	1	0	0	1	1	0	1	0	0	1	1	0	0	1	0	1	1	0
Bias:						$\frac{0}{8}$	$\frac{0}{8}$	$\frac{-2}{8}$	$\frac{2}{8}$	$\frac{0}{8}$	$\frac{2}{8}$	$\frac{-2}{8}$	$\frac{0}{8}$	$\frac{-2}{8}$	$\frac{0}{8}$	$\frac{0}{8}$	$\frac{-2}{8}$	$\frac{0}{8}$	$\frac{2}{8}$	$\frac{2}{8}$	$\frac{0}{8}$	$\frac{0}{8}$	$\frac{2}{8}$	$\frac{-2}{8}$	$\frac{0}{8}$

Überschrift x bedeutet: $Eingabesumme_{10} * 2^3 + Ausgabesumme_{10} = x$

X_0	X_1	X_2	Y_0	Y_1	Y_2	40	41	42	43	44	45	46	47	48	49	50	51	52	53	54	55	56	57	58	59
0	0	0	0	1	0	0	0	1	1	0	0	1	1	0	0	1	1	0	0	1	1	0	0	1	1
0	0	1	1	0	1	1	0	1	0	0	1	0	1	0	1	0	1	1	0	1	0	1	0	1	0
0	1	0	0	0	0	0	0	0	0	0	0	0	0	1	1	1	1	1	1	1	1	1	1	1	1
0	1	1	1	0	0	1	1	1	1	0	0	0	0	1	1	1	1	0	0	0	0	0	0	0	0
1	0	0	1	1	0	1	1	0	0	0	0	1	1	1	1	0	0	0	0	1	1	1	1	0	0
1	0	1	0	1	1	0	1	1	0	0	1	1	0	1	0	0	1	1	0	0	1	0	1	1	0
1	1	0	0	0	1	1	0	1	0	1	0	1	0	0	1	0	1	0	1	0	1	0	1	0	1
1	1	1	1	1	1	0	1	1	0	1	0	0	1	0	1	1	0	1	0	0	1	1	0	0	1
Bias:						$\frac{0}{8}$	$\frac{0}{8}$	$\frac{-2}{8}$	$\frac{2}{8}$	$\frac{2}{8}$	$\frac{2}{8}$	$\frac{0}{8}$	$\frac{0}{8}$	$\frac{0}{8}$	$\frac{-2}{8}$	$\frac{0}{8}$	$\frac{-2}{8}$	$\frac{0}{8}$	$\frac{2}{8}$	$\frac{0}{8}$	$\frac{-2}{8}$	$\frac{0}{8}$	$\frac{0}{8}$	$\frac{0}{8}$	$\frac{0}{8}$

Überschrift x bedeutet: $Eingabesumme_{10} * 2^3 + Ausgabesumme_{10} = x$

X_0	X_1	X_2	Y_0	Y_1	Y_2	60	61	62	63
0	0	0	0	1	0	0	0	1	1
0	0	1	1	0	1	0	1	0	1
0	1	0	0	0	0	1	1	1	1
0	1	1	1	0	0	1	1	1	1
1	0	0	1	1	0	0	0	1	1
1	0	1	0	1	1	0	1	1	0

1	1	0	0	0	1	0	1	0	1
1	1	1	1	1	1	0	1	1	0
Bias:						$\frac{2}{8}$	$\frac{-2}{8}$	$\frac{-2}{8}$	$\frac{-2}{8}$

A.2 Linearkombinationen der S-Box S_2 von SPN_{bsp}

Überschrift x bedeutet: $Eingabesumme_{10} * 2^3 + Ausgabesumme_{10} = x$

X_0	X_1	X_2	Y_0	Y_1	Y_2	0	1	2	3	4	5	6	7	8	9	10	11	12	13	14	15	16	17	18	19
0	0	0	0	1	0	0	0	1	1	0	0	1	1	0	0	1	1	0	0	1	1	0	0	1	1
0	0	1	1	0	1	0	1	0	1	1	0	1	0	1	0	1	0	0	1	0	1	0	1	0	1
0	1	0	1	0	0	0	0	0	0	1	1	1	1	0	0	0	0	1	1	1	1	1	1	1	1
0	1	1	1	1	0	0	0	1	1	1	1	0	0	1	1	0	0	0	0	1	1	1	1	0	0
1	0	0	0	0	1	0	1	0	1	0	1	0	1	0	1	0	1	0	1	0	1	0	1	0	1
1	0	1	0	0	0	0	0	0	0	0	0	0	0	1	1	1	1	1	1	1	1	0	0	0	0
1	1	0	1	1	1	0	1	1	0	1	0	0	1	0	1	1	0	1	0	0	1	1	0	0	1
1	1	1	0	1	1	0	1	1	0	0	1	1	0	1	0	0	1	1	0	0	1	1	0	0	1
Bias:						$\frac{4}{8}$	$\frac{0}{8}$	$\frac{0}{8}$	$\frac{0}{8}$	$\frac{0}{8}$	$\frac{0}{8}$	$\frac{0}{8}$	$\frac{0}{8}$	$\frac{0}{8}$	$\frac{0}{8}$	$\frac{0}{8}$	$\frac{0}{8}$	$\frac{0}{8}$	$\frac{0}{8}$	$\frac{0}{8}$	$\frac{-4}{8}$	$\frac{0}{8}$	$\frac{0}{8}$	$\frac{2}{8}$	$\frac{-2}{8}$

Überschrift x bedeutet: $Eingabesumme_{10} * 2^3 + Ausgabesumme_{10} = x$

X_0	X_1	X_2	Y_0	Y_1	Y_2	20	21	22	23	24	25	26	27	28	29	30	31	32	33	34	35	36	37	38	39
0	0	0	0	1	0	0	0	1	1	0	0	1	1	0	0	1	1	0	0	1	1	0	0	1	1
0	0	1	1	0	1	1	0	1	0	1	0	1	0	0	1	0	1	0	1	0	1	1	0	1	0
0	1	0	1	0	0	0	0	0	0	1	1	1	1	0	0	0	0	0	0	0	0	1	1	1	1
0	1	1	1	1	0	0	0	1	1	0	0	1	1	1	1	0	0	0	0	1	1	1	1	0	0

1	0	0	0	0	1	0	1	0	1	0	1	0	1	0	1	0	1	1	0	1	0	1	0	1	0
1	0	1	0	0	0	0	0	0	0	1	1	1	1	1	1	1	1	1	1	1	1	1	1	1	1
1	1	0	1	1	1	0	1	1	0	1	0	0	1	0	1	1	0	1	0	0	1	0	1	1	0
1	1	1	0	1	1	1	0	0	1	0	1	1	0	0	1	1	0	1	0	0	1	1	0	0	1
Bias:						$\frac{2}{8}$	$\frac{2}{8}$	$\frac{0}{8}$	$\frac{0}{8}$	$\frac{0}{8}$	$\frac{0}{8}$	$\frac{-2}{8}$	$\frac{-2}{8}$	$\frac{2}{8}$	$\frac{-2}{8}$	$\frac{0}{8}$	$\frac{0}{8}$	$\frac{0}{8}$	$\frac{2}{8}$	$\frac{0}{8}$	$\frac{-2}{8}$	$\frac{-2}{8}$	$\frac{0}{8}$	$\frac{-2}{8}$	$\frac{0}{8}$

Überschrift x bedeutet: $Eingabesumme_{10} * 2^3 + Ausgabesumme_{10} = x$

X_0	X_1	X_2	Y_0	Y_1	Y_2	40	41	42	43	44	45	46	47	48	49	50	51	52	53	54	55	56	57	58	59
0	0	0	0	1	0	0	0	1	1	0	0	1	1	0	0	1	1	0	0	1	1	0	0	1	1
0	0	1	1	0	1	1	0	1	0	0	1	0	1	0	1	0	1	1	0	1	0	1	0	1	0
0	1	0	1	0	0	0	0	0	0	1	1	1	1	1	1	1	1	0	0	0	0	1	1	1	1
0	1	1	1	1	0	1	1	0	0	0	0	1	1	1	1	0	0	0	0	1	1	0	0	1	1
1	0	0	0	0	1	1	0	1	0	1	0	1	0	1	0	1	0	1	0	1	0	1	0	1	0
1	0	1	0	0	0	0	0	0	0	0	0	0	0	1	1	1	1	1	1	1	1	0	0	0	0
1	1	0	1	1	1	1	0	0	1	0	1	1	0	0	1	1	0	1	0	0	1	0	1	1	0
1	1	1	0	1	1	0	1	1	0	0	1	1	0	0	1	1	0	0	1	1	0	1	0	0	1
Bias:						$\frac{0}{8}$	$\frac{2}{8}$	$\frac{0}{8}$	$\frac{2}{8}$	$\frac{2}{8}$	$\frac{0}{8}$	$\frac{-2}{8}$	$\frac{0}{8}$	$\frac{0}{8}$	$\frac{-2}{8}$	$\frac{-2}{8}$	$\frac{0}{8}$	$\frac{0}{8}$	$\frac{2}{8}$	$\frac{-2}{8}$	$\frac{0}{8}$	$\frac{0}{8}$	$\frac{2}{8}$	$\frac{-2}{8}$	$\frac{0}{8}$

Überschrift x bedeutet: $Eingabesumme_{10} * 2^3 + Ausgabesumme_{10} = x$

X_0	X_1	X_2	Y_0	Y_1	Y_2	60	61	62	63
0	0	0	0	1	0	0	0	1	1
0	0	1	1	0	1	0	1	0	1
0	1	0	1	0	0	0	0	0	0

0	1	1	1	1	0	1	1	0	0
1	0	0	0	0	1	1	0	1	0
1	0	1	0	0	0	0	0	0	0
1	1	0	1	1	1	1	0	0	1
1	1	1	0	1	1	1	0	0	1
Bias:						$\frac{0}{8}$	$\frac{2}{8}$	$\frac{2}{8}$	$\frac{0}{8}$

A.3 Differenzen der S-Box S_1 und S_2 von SPN_{bsp}

$\Delta(000)$				
x	x^*	y	y^*	y'
000	000	010	010	000
001	001	101	101	000
010	010	000	000	000
011	011	100	100	000
100	100	110	110	000
101	101	011	011	000
110	110	001	001	000
111	111	111	111	000
$\Delta(001)$				
x	x^*	y	y^*	y'
000	001	010	101	111
001	000	101	010	111
010	011	000	100	100
011	010	100	000	100
100	101	110	011	101
101	100	011	110	101
110	111	001	111	110
111	110	111	001	110
$\Delta(010)$				
x	x^*	y	y^*	y'
000	010	010	000	010
001	011	101	100	001
010	000	000	010	010
011	001	100	101	001
100	110	110	001	111
101	111	011	111	100
110	100	001	110	111
111	101	111	011	100
$\Delta(011)$				
x	x^*	y	y^*	y'
000	011	010	100	110
001	010	101	000	101
010	001	000	101	101
011	000	100	010	110
100	111	110	111	001
101	110	011	001	010
110	101	001	011	010
111	100	111	110	001

(a) Die Differenzen von S_1 (Teil 1)

$\Delta(000)$				
x	x^*	y	y^*	y'
000	000	010	010	000
001	001	101	101	000
010	010	100	100	000
011	011	110	110	000
100	100	001	001	000
101	101	000	000	000
110	110	111	111	000
111	111	011	011	000
$\Delta(001)$				
x	x^*	y	y^*	y'
000	001	010	101	111
001	000	101	010	111
010	011	100	110	010
011	010	110	100	010
100	101	001	000	001
101	100	000	001	001
110	111	111	011	100
111	110	011	111	100
$\Delta(010)$				
x	x^*	y	y^*	y'
000	010	010	100	110
001	011	101	110	011
010	000	100	010	110
011	001	110	101	011
100	110	001	111	110
101	111	000	011	011
110	100	111	001	110
111	101	011	000	011
$\Delta(011)$				
x	x^*	y	y^*	y'
000	011	010	110	100
001	010	101	100	001
010	001	100	101	001
011	000	110	010	100
100	111	001	011	010
101	110	000	111	111
110	101	111	000	111
111	100	011	001	010

(b) Die Differenzen von S_2 (Teil 1)

Abbildung 23: Die Differenzen der S-Boxen von SPN_{bsp} (Teil 1)

$\Delta(100)$				
x	x^*	y	y^*	y'
000	100	010	110	100
001	101	101	011	110
010	110	000	001	001
011	111	100	111	011
100	000	110	010	100
101	001	011	101	110
110	010	001	000	001
111	011	111	100	011

$\Delta(101)$				
x	x^*	y	y^*	y'
000	101	010	011	001
001	100	101	110	011
010	111	000	111	111
011	110	100	001	101
100	001	110	101	011
101	000	011	010	001
110	011	001	100	101
111	010	111	000	111

$\Delta(110)$				
x	x^*	y	y^*	y'
000	110	010	001	011
001	111	101	111	010
010	100	000	110	110
011	101	100	011	111
100	010	110	000	110
101	011	011	100	111
110	000	001	010	011
111	001	111	101	010

$\Delta(111)$				
x	x^*	y	y^*	y'
000	111	010	111	101
001	110	101	001	100
010	101	000	011	011
011	100	100	110	010
100	011	110	100	010
101	010	011	000	011
110	001	001	101	100
111	000	111	010	101

(a) Die Differenzen von S_1 (Teil 2)

$\Delta(100)$				
x	x^*	y	y^*	y'
000	100	010	001	011
001	101	101	000	101
010	110	100	111	011
011	111	110	011	101
100	000	001	010	011
101	001	000	101	101
110	010	111	100	011
111	011	011	110	101

$\Delta(101)$				
x	x^*	y	y^*	y'
000	101	010	000	010
001	100	101	001	100
010	111	100	011	111
011	110	110	111	001
100	001	001	101	100
101	000	000	010	010
110	011	111	110	001
111	010	011	100	111

$\Delta(110)$				
x	x^*	y	y^*	y'
000	110	010	111	101
001	111	101	011	110
010	100	100	001	101
011	101	110	000	110
100	010	001	100	101
101	011	000	110	110
110	000	111	010	101
111	001	011	101	110

$\Delta(111)$				
x	x^*	y	y^*	y'
000	111	010	011	001
001	110	101	111	010
010	101	100	000	100
011	100	110	001	111
100	011	001	110	111
101	010	000	100	100
110	001	111	101	010
111	000	011	010	001

(b) Die Differenzen von S_2 (Teil 2)

Abbildung 24: Die Differenzen der S-Boxen von SPN_{bsp} (Teil 2)

Abbildungsverzeichnis

Literaturverzeichnis

[BH08] Büchter, Andreas und Hans Wolfgang Henn: *Elementare Stochastik*. Springer, Berlin Heidelberg New York, 2. Auflage, 2008, ISBN 3540453814.

[Bos05] Bosch, Siegfried: *Algebra*. Springer, Berlin Heidelberg New York, 6. Auflage, 2005, ISBN 3540298800.

[BS90] Biham, E. and A. Shamir: *Differential cryptanalysis of DES-like cryptosystems*. Technical Report CS90-16, Weizmann Institute of Science, July 1990. http://www.cs.technion.ac.il/~biham/Reports/Weizmann/cs90-16.ps.gz.

[CSQ07] Collard, Baudoin, François Xavier Standaert, and Jean Jacques Quisquater: *Improving the Time Complexity of Matsui's Linear Cryptanalysis*. In Nam, K. H. and G. Rhee (editors): *The International Conference on Information Security and Cryptology - ICISC 2007*, volume 4817 of *Lecture Notes in Computer Science*, pages 77–88. Springer, November 2007. http://www.dice.ucl.ac.be/~fstandae/PUBLIS/48.pdf.

[CV95] Chabaud, Florent and Serge Vaudenay: *Links between differential and linear cryptanalysis*. In *Workshop on the Theory and Application of Cryptographic Techniques, EUROCRYPT '94*, volume 950 of *Lecture Notes in Computer Science*, pages 356–365, 1995. http://citeseerx.ist.psu.edu/viewdoc/download?doi=10.1.1.30.4694&rep=rep1&type=pdf.

[DR02] Daemen, Joan and Vincent Rijmen: *The Design of Rijndael: AES – The Advanced Encryption Standard*. Springer Verlag, Berlin, Heidelberg, New York, 2002, ISBN 3-540-42580-2.

[Hey02] Heys, Howard M.: *A tutorial on linear and differential cryptanalysis*. Cryptologia, 26(3):189–221, 2002, ISSN 0161-1194. http://www.engr.mun.ca/~howard/PAPERS/ldc_tutorial.pdf.

[HT96] Heys, Howard M. and Stafford E. Tavares: *Substitution-permutation networks resistant to differential and linear cryptanalysis*. Journal of Cryptology, 9:1–19, 1996. http://m4dch4t.effraie.org/crypto/codebreakers/JournalHeysTavares.pdf.

[Ker83] Kerckhoffs, Auguste: *La Cryptographie Militaire*. Journal des Sciences Militaires, 9:5–38, Januar 1883. http://www.petitcolas.net/fabien/kerckhoffs/crypto_militaire_1.pdf.

[Knu94] Knudsen, Lars Ramkilde: *Block Ciphers - Analysis, Design and Applications*. PhD thesis, July 1994. www.daimi.au.dk/PB/485/PB-485.pdf.

[Koh04] Kohel, David R.: *Elementary cryptography and protocols*, 2004. Script The University of Sydney http://echidna.maths.usyd.edu.au/kohel/tch/USyd/MATH3024/Lectures/lectures_01.pdf.

[LH94] Langford, Susan K. and Martin E. Hellman: *Differential-linear cryptanalysis.* In *Proceedings of the 14th Annual International Cryptology Conference on Advances in Cryptology*, CRYPTO '94, pages 17–25, London, UK, 1994. Springer-Verlag, ISBN 3-540-58333-5. http://www.springerlink.com/content/gdub9r2n9ry92g0y/fulltext.pdf.

[MA93] Matsui, Mitsuru and Yamagishi Atsuhiro: *A new method for known plaintext attack of feal cipher.* In *EUROCRYPT '92: Advances in Cryptology*, pages 81–91. Springer-Verlag New York, Inc., 1993. http://wiki.redbrick.dcu.ie/~rob/MSSF/crypto/matsui.pdf.

[Mat94] Matsui, Mitsuru: *Linear cryptanalysis method for DES cipher.* In *EUROCRYPT '93: Workshop on the theory and application of cryptographic techniques on Advances in cryptology*, pages 386–397, Secaucus, NJ, USA, 1994. Springer-Verlag New York, Inc., ISBN 3-540-57600-2. http://homes.esat.kuleuven.be/~abiryuko/Cryptan/matsui_des.PDF.

[Sch00] Schneier, Bruce: *A self-study course in block-cipher cryptanalysis.* Cryptologia, 24(1):18–33, 2000, ISSN 0161-1194. http://www.schneier.com/paper-self-study.pdf.

[Sel07] Selçuk, Ali Aydin: *On probability of success in linear and differential cryptanalysis.* J. Cryptology, 21(1):131–147, 2007. http://www.springerlink.com/content/d43002t445545665/fulltext.pdf.

[Sha49] Shannon, Claude Elwood: *Communication theory of secrecy systems.* Bell System Technical Journal, Vol 28, pp. 656–715, October 1949. http://netlab.cs.ucla.edu/wiki/files/shannon1949.pdf.

[Sin07] Singh, Simon: *Codes.* DTV, München, 3. Auflage, April 2007, ISBN 9783423621670.

[Sin08] Singh, Simon: *Geheime Botschaften.* DTV, München, 8. Auflage, November 2008, ISBN 9783423330718.

[Sti06] Stinson, Douglas: *Cryptography: Theory and Practice.* CRC/Chapman & Hall, 3rd edition, 2006, ISBN 1584885084.

[Wika] Wikipedia: *Brute-Force-Methode.* http://de.wikipedia.org/w/index.php?title=Brute-Force-Methode&oldid=79350975, besucht: 7. November 2010.

[Wikb] Wikipedia: *Great Deluge algorithm*. http://en.wikipedia.org/w/index.php?title=Great_Deluge_algorithm&oldid=388635674, visited on 7th November 2010.

[Wikc] Wikipedia: *RSA*. http://en.wikipedia.org/w/index.php?title=RSA&oldid=385318583, visited on 7th November 2010.

[Wikd] Wikipedia: *Schwellenakzeptanz*. http://de.wikipedia.org/w/index.php?title=Schwellenakzeptanz&oldid=73991365, besucht: 7. November 2010.

[Wil08] Wilkeit, Elke: *Kryptologie*, 2008. Vorlesungsskript Carl von Ossietzky Universität Oldenburg, WS 2008/2009.

Abkürzungsverzeichnis

AES	Advanced Encryption Standard
DES	Data Encryption Standard
FFT	Fast-Fourier-Transformation
GUI	Graphical User Interface
SPN	Substitutions-/Permutations-Netz

Symbolverzeichnis

$\mathbb{N}$	natürliche Zahlen, einschließlich 0
$\mathbb{Z}_2$	die Menge $\{0,1\}$
$\Delta(x,y)$	die Differenz zwischen zwei Wörtern aus $\mathbb{Z}_2$
$\Delta(x')$	die Paare von Wörtern mit Differenz x'
$\biguplus$	die Vereinigung paarweise disjunkter Mengen
$\varnothing$	die leere Menge
$\mathscr{P}$	die Potenzmenge
$P[a]$	das Wahrscheinlichkeitsmaß
(Ω,P)	der Wahrscheinlichkeitsraum

Index

Erklärung

Hiermit versichere ich, dass ich diese Arbeit selbstständig verfasst und keine anderen als die angegebenen Quellen und Hilfsmittel benutzt habe.

Oldenburg, den

(Manuel Gieseking)