# Kryptologie

Christoph Plüss Christian Hennecke Khalid Harrar

15. Januar 2025

# Inhaltsverzeichnis

# 1 Konzeption der Unterrichtseinheit

## 1.1 Vorwissen

Zuvor befassten sich die Lernenden mit dem Thema Data Science und Sicherheit, welches im Kapitel 2 "Geheimschriften und Datensicherheit" des Buches "Informatik: Data Science und Sicherheit" [1] beschrieben ist.

Wir werden in dieser Unterrichtseinheit aufbauend auf dieses Vorwissen die Kryptologie weiterführen.

Folgenden Themen wurden bereits besprochen:

- Geheimschriften der Antike: Transpositionen und Substitutionen.
- Kryptosysteme: Kryptosystem Caesar, Kryptosystem mit vielen Schlüsseln
- Stochastische Kryptoanalyse
- Häufigkeitanalyse
- Stochastik und polyalphabetische Kryptosysteme
- Jules Verne und Kryptoanalyse
- Kryptosystem Vigenère
- Angriff auf Vigenère
- Bestimmung der Schlüssellänge mit dem Kasiski-Test
- Bestimmung der Schlüssellänge mit Friedman'scher Charakteristik

## 1.2 Zusammenfassung des Vorwissens

### 1.2.1 Geheimschriften der Antike

Die Geheimschriften werden verwendet um Daten zu schützen. So wurden schon in der Antike vertrauliche Daten gegen die unerwünschte Weitergabe mit einer Geheimschrift geschützt. Die ersten Funde von Geheimschriften sind ungefähr 4000 Jahre alt und basierten auf Transpositionen, also Austausch der Positionen von Buchstaben. Vor etwa 2500 Jahren begann sich eine neue Methode zur Entwicklung von Geheimschriften durchzusetzen, die sogenannte Substitution, wo man die Buchstaben durch andere Symbole kodiert hat.

**Definition 1.1**

Eine *Geheimschrift* besteht aus vier Elementen

(1) Klartextalphabet

(2) Geheimtextalphabet

(3) Algorithmus[a] zur Umwandlung von Klartext in Geheimtext

(4) Algorithmus[b] zur Umwandlung von Geheimtext in den Klartext

**Bemerkung 1.** Die Zeichen des Geheimtextes bzw. der Geheimtext wird als Chiffre bezeichnet.

[a] Chiffrieralgorithmus
[b] Dechiffrieralgorithmus

Der Chiffrierungsalgorithmus und der Dechiffrierungsalgorithmus kann als injektive Funktionen betrachtet werden, die den Klartext zum Geheimtext und den Geheimtext zum Klartext abbildet.

Beim CAESAR Kryptosystem erfolgt die Chiffrierung und die Dechiffrierung pro Buchstabe. Oft werden auch mehrere Buchstaben zu einem Block zusammen genommen und blockweise chiffriert und dechiffriert.

### 1.2.2 Sammlungen von Geheimschriften, Kryptosysteme

Wenn zur Geheimhaltung von Daten immer die gleiche Vorschrift verwendet wird, wächst das Risiko, dass das Geheimnis, die Vorschrift für die Verschlüsselung, irgendwann gelüftet wird. Deswegen hat man sich schon in der Antike überlegt, wie man abwechselnd unterschiedliche Schlüssel verwenden kann.

**Definition 1.2**

Ein *Kryptosystem* ist eine Sammlung von Geheimschriften, wobei jede Geheimschrift einen eindeutigen Namen hat. Die einzelnen Namen der Geheimschriften nennt man *Schlüssel*.

### 1.2.3 Beschreibung verschiedener Kryptosysteme

#### Kryptosystem Caesar

Das Kryptosystem CAESAR besteht aus 25 Geheimschriften, die man mit Zahlen $1, 2, \ldots, 25$ bezeichnete. Alle benutzten das lateinische Alphabet für Klartexte sowie für Geheimtexte. Wenn man die Geheimschrift " $i$ " verwendet hat, hat man jeden Buchstaben des Klartextes bei der Chiffrierung durch den Buchstaben ersetzt, der $i$-Positionen rechts davon im Alphabet steht. Wenn man über Z hinausgegangen ist, setzte man am Anfang des Alphabets fort.

#### Kryptosystem Jules Verne

In einem Buch von "Jules Verne" geht es um die Dechiffrierung eines Geheimtextes. Der Schlüssel ist eine Dezimalzahl. Man schreibt die Zahl Ziffer für Ziffer unter einem Klartext. Die Chiffrierung der Buchstaben erfolgt mit CAESAR. Jede Buchstabe wird um so viel Positionen verschoben, wie die Ziffer unter den Buchstaben angibt.

#### Kryptosystem Vigenère

Das Kryptosystem *Vigenère* verwendet die Buchstaben eines Wortes statt dezimalen Zahlen als Schlüssel für die Verschiebung der Buchstaben des Klartextes Vor der Chiffrierung schneidet man den Klartext in Stücke der Länge des Schlüsselwortes und jedes Stück wird mit der gleichen Chiffrierungsmethode verschlüsselt.

#### Kryptosystem mit vielen Schlüsseln

Hat das Kryptosystem wenige Schlüssel wie z.B. der CAESAR ist das Kryptosystem nur bedingt sicher. Der Schlüssel kann mit wenig Aufwand z.B. mit einem Programm gesucht werden. Variationen des Kryptosystems CAESAR wie z.B. die Kryptosysteme JULES VERNE und VIGENÈRE sind unwesentlich sicherer. Die Schwachstelle ist das von CAESAR verwendete Chiffrierverfahren, das keine effizientere Nutzung des Schlüssels erlaubt.

**Bemerkung 2.** Beim CAESAR Kryptosystem wird die Ordinalzahl des Buchstabens im Klartext Alphabet als Schlüssel verwendet. Das ist eine Zahl zwischen 1 und 25 die mit 5 Bit im binären Zahlensystem dargestellt werden kann. Würde ein zwei Buchstabenschlüssel verwendet und würde abwechselnd mit dem ersten und dem zweiten Buchstaben verschlüsselt gäbe es schon 25 * 24 = 600 unterschiedliche Schlüssel.

**Bemerkung 3** (Injektive Funktion)**.** In vielen kommerziellen Anwendungen wird ein Kryptosystem verwendet um die Echtheit oder Authentizität der Zahlungsdaten sicher zu stellen. Es wird z.B. das folgende Verfahren eingesetzt:

Die Zahlungsdaten werden verschlüsselt und Teile des Geheimtextes werden an die Zahlungsdaten angehängt. Die verschlüsselten Zahlungsdaten können, unter bestimmten Umständen, als Signatur des Senders des Zahlungsauftrags betrachtet werden.

Zur Überprüfung verschlüsselt der Empfänger die Zahlungsdaten und vergleicht seinen Geheimtext mit der Signatur des Senders. Stimmt der Geheimtext des Empfängers mit der Signatur des Senders überein, ist der Zahlungsauftrag echt und die Zahlung kann ausgeführt werden.

Mit diesem Vorgehen ist keine Entschlüsselung und keine Plausibilisierung der Entschlüsselung notwendig.

Wird für die Verschlüsselung keine injektive Funktion verwendet kann der Empfänger des Zahlungsauftrags nicht aufgrund der Signatur entscheiden ob die Zahlung authentisch oder manipuliert ist. Die Anwendung von Kryptographie macht so keinen Sinn!

Der Empfänger des Zahlungsauftrags kann nur die Echtheit des Zahlungsauftrags überprüfen, wenn er, mit guter Sicherheit, davon ausgehen kann, dass nur der Sender und der Empfänger die Geheimschrift verwenden. Bei einer nicht injektiven Chiffrierung kann er das nicht.

Jedes Nutzerpaar eines Kryptosystems benötigt einen eigenen Schlüssel. Nehmen wir an dass eine Million Schweizer, ihre Zahlungen elektronisch abwickeln. Wir haben entsprechend $10^6$ Nutzerpaare. Für gute Sicherheit benötigen wir schnell einmal $10^9$ Schlüssel. Dies um zu verhindern, dass 2 Nutzerpaare nicht zufällig den gleichen Schlüssel haben.

Heute werden Kryptosysteme mit $10^{20}$ Schlüsseln als nicht sicher betrachtet. Momentan werden Kryptosysteme mit $10^{40}$ Schlüsseln und einem guten Chiffrierverfahren noch als sicher betrachtet.

**Bemerkung 4** (Injektive Funktion)**.** Die Anforderung injektive Funktion für die Chiffrierung hilft gute Kryptosysteme zu finden. Das ist aus dem folgenden Experiment ersichtlich:

Nehmen wir an wir verwenden die bitweise AND oder OR Funktion für die Verschlüsselung der Bytes eines Bildes. Als Schlüssel könnte z.B. 3A[1] verwendet werden.

**Frage:** Ist das Bild nach der Verschlüsselung noch erkennbar?

**Optionale Aktivität:** Die Schüler und Schülerinnen sollen Ideen für bessere Verschlüsselungsfunktionen entwickeln und beschreiben.

**Kommentar 1.** Optimal ist die Durchführung des Experimentes als Computerexperiment, mit einem Programm bei dem die Schüler und Schülerinnen mit verschiedenen Verschlüsselungsverfahren und Schlüsseln experimentieren können. Verfügen die Schülerinnen und Schüler schon über Programmierkenntnisse, können sie ihr eigenes Verschlüsselungsverfahren, das sie vorher konzipiert haben integrieren. Diese Aktivität eignet sich auch gut für eine Think-Pair-Share Organisation.

[1] Hexadezimalschreibweise

**Definition 1.3**

Eine Geheimschrift heisst *monoalphabetisch*, wenn bei der Chiffrierung ein Buchstabe aus dem Klartextalphabet durch ein und denselben Buchstaben aus dem Geheimtextalphabet verschlüsselt wird.
Die Chiffrierung eines Klartext Buchstabens ist unabhängig von der Position des Buchstabens im Klartext immer gleich.

Ist das Klartextalphabet und das Geheimtextalphabet identisch und wird keine Ersetzung vorgenommen liegt einen Null-Verschlüsselung vor. Ein interessanter Spezialfall, der z.B. als Verschlüsselung in einer Testumgebung verwendet werden kann.

**Definition 1.4**

Eine Geheimschrift heisst *polyalphabetisch*, wenn ein Buchstabe aus dem Klartextalphabet durch mehrere Buchstaben aus dem Geheimtextalphabet verschlüsselt werden kann.

Die Wahl, welcher Buchstabe zu Verschlüsselung genommen wird, kann z.B. von der Position des Klartextbuchstabens im Klartext, der verschlüsselt wird, abhängen.

**Anforderung.** Wir suchen Kryptosysteme mit möglichst vielen Schlüsseln und einer guten Chiffrierung.

Eine Anforderung an eine gute Chiffrierung ist ihre effiziente Ausführung. Dies bezieht sich sowohl auf den Speicherplatz, wie auf die Ausführungszeit für die Verschlüsselung eines Buchstabens.

**Bemerkung 5.** Bei einer guten Chiffrierung kann der Schlüssel nicht einfach aus dem Geheimtext ermittelt werden. Gegenüber dieser Anforderung ist CAESAR kein gutes Kryptosystem, da es nur 26 Geheimschriften erlaubt und der Schlüssel z.B. mit einer Häufigkeitsanalyse aus dem Geheimtext ermittelt werden kann.

### 1.2.4 Stochastische Kryptoanalyse

Die interdisziplinäre Wissenschaft, in welcher Kryptosysteme auf Schwachstellen untersucht werdem, bezeichnet man als *Kryptoanalyse*. Die Entwicklung von sicheren Kryptosystemen oder die sichere Anwendung eines Kryptosystems ohne begleitende Kryptoanalyse resultiert in Unsicherheit. Erst aus dem Zusammenspiel zwischen dem Entwurf und der Kryptoanalyse entsteht die gewünschte Sicherheit.

**Bemerkung 6.** In den 90-iger Jahren wurden bei der Suche nach guten Kryptosystemen Wettbewerbe, zum Teil auch mit Preisgeldern, veranstaltet. Auch heute noch ist die Kryptoanalyse eine sehr aktives und bedeutendes Forschungsgebiet. Dies um auch in Zukunft geeignete Kryptosysteme z.B. für die sichere Nutzung des Internets zur Verfügung zu stellen. Das neue Standard Verschlüsselungsverfahren AES wurde im Rahmen eines Auswahlprozesses ausgewählt. Theoretische Schwachstellen waren ein Auswahlkriterium. Der Wikipedia Eintrag zu AES ist lesenswert.

Die stochastische Kryptoanalyse benutzt für die Analyse Mittel aus der Statistik wie z.B die Häufigkeitsanalyse der Buchstaben des Klartextes. Der Anhaltspunkt für die Kryptoanalyse ist die Tatsache, dass sich in monoalphabetischen Geheimschriften die Häufigkeit der Buchstaben im Klartext auf die Häufigkeit der Buchstaben in Geheimtext überträgt.

In stochastischen, polyalphabetischen Krypotsystemen mit gut gewählten Schlüsseln kommen alle Buchstaben ungefähr gleich verteilt vor. In diesem Fall liefert die stochastische Kryptanalyse keine Hinweise für die Eingrenzung der Schlüsselsuche.

### 1.2.5 Stochastische Kryptoanalyse des Vigenère Kryptosystems

Wenn die Schlüssellänge mit dem Kasiski-Test oder mit der Friedman'scher Charakteristik ermittelt wurde, kann mit der Häufigkeitsverteilung von Buchstabenfolgen der Länge z.B. 1 2, 3 oder 4 die Schlüsselsuche eingeschränkt und der Schlüssel letztendlich ermittelt werden.

#### Bestimmung der Schlüssellänge mit dem Kasiski-test

Der Kasiski-Test versucht zuerst die Länge des Schlüssels zu bestimmen. Man sucht gleiche Trigramme im Geheimtext. Die Abstände dieser Trigramme sind ein Mehrfaches der Schlüssellänge wenn sie Chiffrierungen der gleichen Trigramme des Klartextes sind, was in der Regel der Fall ist. Wenn man die Schlüssellänge kennt, teilt man die Geheimschrift in entsprechend viele Gruppen und wendet in einzelnen Gruppen eine Häufigkeitsanalyse der Buchstaben an.

#### Bestimmung der Schlüssellänge mit Friedman'scher Charakteristik

Mit der Friedman'scher Charakteristik kann man die Buchstabenverteilung im Text einfach durch eine Zahl charakterisieren. Diese Charakterisierung ist so ausgelegt, dass man daraus ablesen kann, ob der Text monoalphabetisch oder polyalphabetisch chiffriert wurde. Bei polyalphabetischen Kryptosystemen kann mittels Probieren verschiedener Schlüssellängen und der erneuten Berechnung der Friedman'schen Charakteristik die definitive Schlüssellänge bestimmt werden.

## 1.3 Analyse des Stoffes und Vorgehen

In dieser Unterrichtseinheit wird auf der Basis des Vorwissens Ansätze und Möglichkeiten für sicherere Kryptosysteme vorgestellt. Hierbei ist ein Kryptosystem sicherer wenn es über viele Schlüssel verfügt und es wenig Möglichkeiten für die Eingrenzung der Schlüsselsuche bietet. Der Zusammenhang der relevanten Themen sind in der Concept-Map in Abschnitt 7.1 dargestellt. Beim Kryptosystem Vigenère kann die Schlüsselsuche eingegrenzt werden indem zuerst die Länge des Schlüssels mit dem Kasiski-Test oder mit der Friedman'scher Charakteristik bestimmt wird. Mit den vorgestellten neuen Kryptosystemen werden die Buchstabenhäufigkeiten verschleiert z.B. mit einer homophonen Verschlüsselung.

**Definition 1.5**

Die *homophone Verschlüsselung* ist eine polyalphabetische Verschlüsselungsmethode, bei der ein Klartextzeichen durch mehrere Geheimtextzeichen substituiert werden kann.
Hierfür wird ein Geheimtext-Alphabet gewählt das wesentlich grösser als das Klartext-Alphabet ist. Es wird z.B. ein Geheimtext-Alphabet mit 100 Zeichen verwendet. Die Chiffrierung eines Klartextbuchstabens erfolgt etwa proportional zu seiner Häufigkeit in unterschiedliche Zeichen des Geheimtextes. Die Wahrscheinlichkeit sehr seltener Buchstaben wird hierbei aufgerundet. Die Wahrscheinlichkeit von sehr häufigen Buchstaben wird abgerundet. Wichtig ist, dass nicht mehr aufgerundet als abgerundet wird.
Bei der Dechiffrierung werden die Geheimtextzeichen die das gleiche Klartextzeichen verschlüsseln zu dem entsprechenden Klartextzeichen entschlüsselt (homophone == gleich-

klingende Entschlüsselung).

Im ersten Teil der Lerneinheit wird die homophone Verschlüsselung vorgestellt.

Als weiteren Ansatz zur Verschleierung der Buchstabenhäufigkeit in einem Geheimtext werden wir das Konzept der Kontextsensitivität einführen. Die Kontextsensitivität wird im zweiten Teil der Lerneinheit vorgestellt. Zunächst wird an einem sehr einfachen Beispiel das Konzept vermittelt, so dass die Lernenden in einem Folgeschritt damit ein eigenes Kryptosystem entwickeln können. Konkret verwenden wir das bekannte Kryptosystem von Caesar und erweitern es beispielhaft für die Verschlüsselung um ein kontextsensitives Element auf das System Caesar+. Wir passen hierbei den Schlüssel in jeder Verschlüsselungsoperation basierend auf dem Schlüssel in der der vorausgegangenen Operation an. Das Verständnis wird durch Übungsbeispiele vertieft. Als finalen Schritt und Vorbereitung für die im Folgeabschnitte angestrebte eigenständige Weiterentwicklung eines Kryptosysxtems leiten die Lernenden den Entschlüsselungsalgorithmus her, was bei Caesar+ weiterhin intuitiv möglich ist.

Im darauf folgenden Abschnitt sollen nun die gelernten Konzepte[2] und Vorgehensweisen zur Verbesserung von bestehenden Methoden selbständig angewendet werden, um aus Vigènere analog zu Caesar+ einen Vigenère+ zu entwickeln. Hierbei soll durch klare Rahmenbedingungen und Hinweise zur Lösungsfindung sichergestellt werden, dass möglichst viele Lernende zumindest Teilerfolge beim Lösen der Aufgaben haben werden.

Im dritten und letzten Teil der Lerneinheit werden die grundlegenden Ideen moderner Kryptosysteme vorgestellt. Moderne Kryptosysteme basierend auf dem Kerckhoffs Prinzip: Das Geheimnis liegt hierbei im Schlüssel, das Verfahren für die Ver- und Entschlüsselung ist öffentlich. An einem sehr einfachen Beispiel wird diese Idee illustriert und auf wesentliche Aspekte des Ansatzes hingewiesen.

## 1.4 Lernziele

### 1.4.1 Leitidee

Bei vielen Anwendungen senden wir sensible Daten über das Netzwerk. Damit ein Dritter sie nicht missbrauchen kann, werden sie geschützt, indem wir vor dem Versand der Daten den Klartext chiffrieren. Aus früheren Lerneinheiten wissen wir, dass die Schlüssel der Geheimtexte aus der Antike leicht ermittelt werden können. Im Fall des Vigenère Kryptosystems kann die Schlüsselsuche beispielsweise mit dem Kasiski-Test oder der Friedman'scher Charakteristik eingegrenzt werden.

Dennoch gelangen unsere Daten relativ sicher zum Empfänger. Dies deutet darauf hin, dass andere Verfahren eingesetzt werden, als die bisher bekannten. Um moderne Verfahren einfacher zugänglich zu machen, werden wir zugrunde liegende Begriffe und methodische Ansätze kennenlernen. Wir werden uns so schrittweise an die modernen Techniken herantasten und werden ein Verständnis dafür aufbauen, weshalb diese Verfahren mehr Sicherheit bieten.

**Leitidee**

Für ein absolut sicheres Verschlüsselungsverfahren darf die Länge des Klartextes nur ein kleines Vielfaches der Länge des Schlüssels sein. Aus mathematischer Sicht kann unter diesen Umständen der Geheimtext nicht von einer Zufallsfolge unterschieden werden.

[2] Kontextexsensitivität

Wenn wir beliebig lange Klartexte z.B. mit 128 oder 256 Bit Schlüsseln chiffrieren, ist absolute Sicherheit nicht möglich.

Wir bezeichnen ein Verschlüsselungsverfahren als relativ sicher wenn es gute Gründe gibt, zu glauben, dass es keinen EFFIZIENTEN Dechiffrieralgorithmus ohne Kenntnis des Schlüssels gibt.

Die Lernenden sollen daher ein Verständnis für relative Sicherheit sowie grundlegende Ansätze, wie sie realisiert werden kann, entwickeln.

Da wir in diesem Zusammenhang täglich neuen Herausforderungen begegnen, sollten Lernende einen Einblick gewinnen, welche Algorithmen zum Schützen unserer Daten eingesetzt werden. Sie sollen einerseits Grenzen erkennen und andererseits verstehen, wie relative Sicherheit ermöglicht werden kann.

Viel Sicherheit entsteht aus dem Zusammenspiel zwischen der Kryptoanalyse und dem Entwurf und der Anwendung von Kryptosystemen.

Ein Verständnis der zugrunde liegenden Mathematik und der erforderlichen Denkweisen wie z.B. Gedankenexperimente sind wichtige Werkzeuge. Der resultierende Werkzeugkasten kann dann benutzt werden um die Stärken und die Schwächen eines Kryptosystems oder einer Anwendung des Kryptosystems analysieren zu können.

### 1.4.2 Dispositionsziel

In der Kryptoanalyse kann beispielsweise darauf aufgebaut werden, dass gewisse Zeichen oder Kombinationen davon oft verwendet werden und andere wiederum selten. Diese Information kann dann für die Schlüsselsuche verwendet werden. Dies führen zu den folgenden Dispositionszielen:

Dispositionsziele

(1) Die Lernenden haben ein Verständnis für relative Sicherheit erlangt und sind sich bewusst, dass mit geeigneten Ansätzen und ausreichend Zeit sowie Rechenleistung Schlüssel (fast) aller Kryptosysteme ermittelt werden können.

(2) Die Lernenden finden einen leichteren Einstieg in moderne Kryptosysteme, indem die Grundbegriffe Kontextfreiheit, Kontextsensitivität und Blockchiffrierung klar sind und selbständig auf neue Verfahren übertragen werden können.

(3) Die Lernenden sind in der Lage, aufbauend auf bekannten Konzepten und Vorgehensweisen eigene Kryptosysteme zu entwickeln.

### 1.4.3 Operationalisierte Lernziele

Operationalisierte Lernziele

(1) Die Lernenden können den Unterschied zwischen der kontextfreier und der kontextsensitiver Verschlüsselung mit einfachen Worten und Beispielen erklären.

(2) Die Lernenden kennen das Homophone Kryptosystem und können die Verbesserung gegenüber Vigenère aus dem Stand aus erklären.

(3) Die Lernenden können mit Hilfe von Verschlüsselungstabellen vorgegebene oder frei

gewählte Texte verschlüsseln und entschlüsseln.

(4) Die Lernenden können neue Verschlüsselungstabellen mit Skalierung oder aus einer Referenz-Verschlüsselungstabelle gemäss ihren Vorstellungen der gewünschten Sicherheit ableiten.

# 2 Beschreibung der Unterrichtssequenz

Die traditionellen Kryptosysteme wie Caesar und Vigenère weisen eine geringe Anzahl von Schlüsseln auf:

- Die Schlüssel können mit Hilfe von Programmen und Computern ermittelt werden.
- Für die Verschlüsselung z.B. für die sichere Kommunikation in einem WLAN benötigt jeder Nutzer des WLAN's einen eigenen Schlüssel. Bei 10 Nutzern werden 1000 oder mehr Schlüssel benötigt um mit guter Sicherheit auszuschliessen, dass 2 Nutzer des WLAN's den selben Schlüssel verwenden.

Für eine gute Informationssicherheit werden leistungsfähigere Kryptosystemem mit viel mehr Schlüsseln benötigt. Vor 30 Jahren wurde das Standard Kryptosystem DES mit etwa $10^{20}$ Schlüsseln als sicher betrachtet. Seit etwa 20 Jahren ist das nicht mehr der Fall, es ist nur noch bedingt sicher und sollte in neuen Anwendungen nicht mehr verwendet werden. Momentan werden Kryptosysteme mit mindestens $10^{40}$ Schlüsseln und einem guten Chiffrierverfahren als sicher betrachtet.

In den folgenden Lerneinheiten werden verschiedenen Ideen vorgestellt um die Schwächen traditioneller Kryptosysteme zu beheben:

- In der ersten Lerneinheit wird das Geheimtextalphabet gegenüber dem Klartextalphabet stark vergrössert. Es wird gezeigt, dass mit einer geeignet gewählten Verschlüsselung die Buchstabenhäufigkeiten verschleiert werden können.

  Es wird ersichtlich, dass bei dieser Art der Verschlüsselung im Idealfall die Buchstabenhäufigkeit im Geheimtext gleich verteilt ist. Eine Einschränkung der Schlüsselsuche z.B. mit Friedman'scher Charakteristik ist damit nicht mehr möglich.

- In der zweiten Lerneinheit werden für die Verschlüsselung eines Buchstaben des Klartextes zusätzliche Buchstaben verwendet. Das kann z.B. der vorhergehende Buchstabe des Klartextes sein.

  Es wird erkennbar, dass die Möglichkeiten zur Verschlüsselung eines Buchstabens damit stark vergrössert werden können. Beim Kryptosystem Caesar wird die Anzahl von 25 auf $25 * 25 = 625$ Varianten der Verschlüsselung erweitert.

- In der dritten Lerneinheit werden zentrale Ideen der modernen Kryptographie vorgestellt. Als Basis wird das Kerckhoff Prinzip verwendet. Gemäss diesem Prinzip ist ein Kryptosystem sicher, wenn das Kryptosystem öffentlich bekannt ist und es trotzdem ohne Kenntnis des verwendeten Schlüssels nicht möglich ist, aus einem Geheimtext den ursprünglichen Klartext abzuleiten.

  Mit den vorgestellten Methoden können Kryptosysteme analysiert und Alternativen zu den vorgestellten Kryptosystemen entwickelt werden.

  Die Vorschrift für die Verschlüsselung und die Entschlüsselung der vorgestellte Geheimschriften und Kryptosysteme werden in Tabellenform dargestellt und können in dieser Form dazu verwendet werden, um Texte zu ver- und entschlüsseln.

  Zudem kann mit Hilfe der Verschlüsselungstabellen auf Eigenschaften der Geheimschrift geschlossen werden und die Ttabellen können gemäss eigener Ideen weiterentwickelt werden.

# 3 Homophone Kryptosysteme

Beim Kryptosystem von VIGENÈRE kann über die Häufigkeitsverteilung der Buchstaben der jeweiligen Sprache, das System geknackt werden. Dies wird hier als Wiederholung des Vorwissens nochmals gezeigt.

## 3.1 Prüfung des Vorwissens

**Aufgabe 3.1**

Nutzen Sie den KASISKI-Test, um die Schlüssellänge der VIGENÈRE-Verschlüsselung zu bestimmen.

NEQSWRNEQDSPSRBOQBOVQMLJEIQCIJSWR

**Lösung Aufgabe 3.1**

Wir suchen im Geheimtext nach Trigrammen, die sich wiederholen.

N E Q S W R N E Q DS P S R B O Q B O V Q M L J E I Q C I J S W R

Das Trigramm NEQ ist auf den Positionen 1 und 7 und somit ist die Differenz 6.

Das Trigramm SWR ist auf den Positionen 4 und 31, was die Differenz 27 ergibt.

ggt(6,27) = 3 und somit kann die Schlüssellänge nur 1 (monoalphabetische Chiffrierung) oder 3 sein. Somit ist die Aufgabenstellung erfolgreich bearbeitet worden. Der Text ist zu kurz, um ihn mit kleinem Aufwand mit statistischen Methoden zu dechiffrieren. Für diejenigen, die es trotzdem versucht haben, ist der Schlüssel K E Y und der Klartext ist:

DAS IST DAS TOR IN DEM DER SCHUESSEL IS

**Frage:** Wie könnte man die Häufigkeiten der Buchstaben verschleiern?

## 3.2 Einleitung

Eine mögliche verbesserte Lösung stellt die sogenannte *Homophone Verschlüsselung* dar. Der Name *homophon* kommt aus dem Griechischen. *Homos* steht für gleich und *phone* steht für Klang. Die Idee dahinter könnte wie folgt beschrieben werden: Ein Klartextzeichen kann durch mehrere Geheimtextzeichnen ersetzt werden. Die Anzahl hängt von der Häufigkeit des Buchstabens ab. Diese Geheimtextzeichnen werden auch *Homophone* genannt.

Es ist jedoch darauf zu achten, dass zwei verschiedene Klartextzeichnen auch durch verschiedene Geheimtextzeichnen ersetzt werden, damit wieder eindeutig entschlüsselt werden kann.

Zum Beispiel eine Möglichkeit würde dahin bestehen, einen Buchstaben durch genauso viele Zahlen von 0 bis 99 zu ersetzen, die der Häufigkeit des Buchstabens in der jeweiligen Sprache entspricht.

Dabei zeigt sich, dass beispielsweise der Buchstabe E in der deutschen Sprache mit der höchsten Häufigkeit von rund 17% auftaucht; der Buchstabe A hingegen nur eine Häufigkeit von rund 6% aufweist. Anhand dieser Häufigkeitsverteilungen könnte man nun den Buchstaben $E$ durch 17 Zahlen und den Buchstaben $A$ durch 6 ersetzen.

Tabelle 1: Häufigkeitsverteilung der Buchstaben in deutschen Texte

| Buchstabe | Häufigkeit | Buchstabe | Häufigkeit |
|---|---|---|---|
| A | 6.51 | N | 9.78 |
| B | 1.89 | O | 2.51 |
| C | 3.06 | P | 0.79 |
| D | 5.08 | Q | 0.02 |
| E | 17.40 | R | 7.00 |
| F | 1.66 | S | 7.27 |
| G | 3.01 | T | 6.15 |
| H | 4.76 | U | 4.35 |
| I | 7.55 | V | 0.67 |
| J | 0.27 | W | 1.89 |
| K | 1.21 | X | 0.03 |
| L | 3.44 | Y | 0.04 |
| M | 2.53 | Z | 1.13 |

## 3.3 Verschlüsseln

Es gibt zwei Möglichkeiten, welcher Geheimtextbuchstabe für einen Klartextbuchstaben genutzt wird. Die erste ist, immer zufällig einen passenden Geheimtextbuchstaben aus der Tabelle zu wählen. Die zweite Methode ist, die Geheimtextbuchstaben abwechselnd zu nutzen.

Bei dem ersten Vorkommen eines Buchstabens im Klartext wird der erste zu diesem Buchstaben passende Geheimtextbuchstabe gewählt. Kommt der Buchstabe danach erneut vor, wird der zweite Geheimtextbuchstabe gewählt. Der Vorteil dieser Methode ist, dass ausgeschlossen wird, dass durch Zufall manche Geheimtextbuchstaben gar nicht genutzt werden. Der Nachteil dieser Methode ist die Regelmässigkeit, die bei der Kryptoanalyse hilfreich sein könnte.

Ein möglicher Schlüssel ist z.B. in der folgenden Tabelle dargestellt.

| A | B | C | D | E | F | G | H | I | J | K | L | M | N | O | P | Q | R | S | T | U | V | W | X | Y | Z |
|---|---|---|---|---|---|---|---|---|---|---|---|---|---|---|---|---|---|---|---|---|---|---|---|---|---|
| 88 | 42 | 60 | 87 | 32 | 76 | 94 | 21 | 57 | 20 | 29 | 38 | 65 | 01 | 82 | 9 | 18 | 27 | 90 | 53 | 07 | 43 | 52 | 61 | 70 | 79 |
| 97 | 51 | 69 | 96 | 41 | 85 | 03 | 30 | 66 | | | 47 | 74 | 10 | 91 | | | 36 | 99 | 62 | 16 | | | | | |
| 06 | | 78 | 05 | 50 | | 12 | 39 | 75 | | | 56 | 83 | 19 | 00 | | | 45 | 08 | 71 | 25 | | | | | |
| 15 | | | 14 | 59 | | | 48 | 84 | | | | 92 | 28 | | | | 54 | 17 | 80 | 34 | | | | | |
| 24 | | | 23 | 68 | | | | 93 | | | | | 37 | | | | 63 | 26 | 89 | | | | | | |
| 33 | | | | 77 | | | | 02 | | | | | 46 | | | | 72 | 35 | 98 | | | | | | |
| | | | | 86 | | | | 11 | | | | | 55 | | | | 81 | 44 | | | | | | | |
| | | | | 95 | | | | | | | | | 64 | | | | | | | | | | | | |
| | | | | 04 | | | | | | | | | 73 | | | | | | | | | | | | |
| | | | | 13 | | | | | | | | | | | | | | | | | | | | | |
| | | | | 22 | | | | | | | | | | | | | | | | | | | | | |
| | | | | 31 | | | | | | | | | | | | | | | | | | | | | |
| | | | | 40 | | | | | | | | | | | | | | | | | | | | | |
| | | | | 49 | | | | | | | | | | | | | | | | | | | | | |
| | | | | 58 | | | | | | | | | | | | | | | | | | | | | |
| | | | | 67 | | | | | | | | | | | | | | | | | | | | | |

Abbildung 1: Beispiel eines Schlüssels

## 3.4 Entschlüsseln

Man erstellt sich eine Tabelle, in der zu jedem Homophon der dazugehörige Klartextbuchstabe aufgelistet ist. Für jeden Geheimtextbuchstaben muss nun einfach der entsprechende Klartextbuchstabe eingesetzt werden. Die Abbildung eines Klartextbuchstabens auf einen Geheimtextbuchstaben ist zwar nicht eindeutig, die Gegenrichtung jedoch schon. Zu jedem Geheimtextbuchstaben gibt es jeweils genau einen dazugehörigen Klartextbuchstaben.

| **Homophon** | 01 | 02 | 03 | 04 | 05 | 06 | 07 | 08 | 09 | 10 | 11 | 12 | 13 | 14 | 15 | ··· |
|---|---|---|---|---|---|---|---|---|---|---|---|---|---|---|---|---|
| **Klartext** | N | I | G | E | D | A | U | S | P | N | I | G | E | D | A | ··· |

**aufgabe 3.2**

Entschlüsseln Sie mit dem Schlüssel aus der Tabelle folgenden Geheimtext:
613241645545242051

**Lösung Aufgabe 3.2**

613241645545242051 in Blöcke von zwei Ziffern aufteilen: 61 32 41 64 55 45 24 20 51, dann mit der Tabelle dechiffrieren: G E H E I M N I S

**Aufgabe 3.3**

Verschlüsseln Sie den Klartext "Diese Nachricht ist sehr geheim", sodass die Häufigkeit der häufig vorkommenden Buchstaben, verschleiert wird, nutzen Sie dazu die Tabelle aus Abbildung 1. Am besten Sie gehen einfach die Menge der möglichen zwei Ziffer für die Substitution der Buchstaben der Reihe nach durch. Sobald Sie alle Elementen benutzt haben, können Sie wieder von vorne beginnen.

**Lösung Aufgabe 3.3**

Diese Nachricht ist sehr geheim. Zuerst werden alle Leerschläge eliminiert, dass alles gross geschrieben: DIESENACHRICHTISTGEHEIM, dann werden die Buchstaben einzeln chiffriert:
D I E S E N A C H R I C H T I S T G E H E I M
08 09 00 27 01 24 02 03 29 15 18 87 35 05 20 44 16 25 07 41 11 42 63

## 3.5 Kryptoanalyse

Für den Angriff reichen Häufigkeiten allein dann nicht mehr aus. Er gelingt aber wieder über Buchstabengruppen: Die Tatsache beispielsweise, dass im Deutschen auf Q fast immer U folgt, zeigt sich auch im Kryptotext darin, dass nach der 18 fast immer 07, 16, 25 und 34 stehen, und zwar jeweils gleich häufig. Solche Muster erlauben es, den Geheimtext zu brechen.

Hierzu benötigt man jedoch deutlich längere Texte. Hinreichend kurze, homophon verschlüsselte Texte sind gegen unbefugte Entschlüsselung recht gut geschützt. Analog gilt dies auch für das Kryptosystem Vigenere. Kurze Texte sind ebenfalls relativ sicher. Wenn der Schlüssel genauso

lang ist, wie der Klartext, ist VIGENERE. sogar absolut sicher, d.h. mittels Kryptoanlyse nicht angreifbar.

Um Angriffe auf ein homophones Kryptosystem weiter zu erschweren, könnte man dessen Geheimtext-Alphabet erweitern, konkret beispielsweise durch eine Verzehnfachung von den 100 Homophonen des oben eingesetzten Geheimtext-Alphabets auf 1000 Homophone.

# 4 Kontextsensitive Kryptosysteme

## 4.1 Kontextsensitivität

In den bisher betrachteten Geheimschriften und Kryptosystemen erfolgt die Chiffrierung jeweils basierend auf den 4 Elementen Klartext, Schlüssel sowie Verschlüsselungs- bzw. Entschlüsselungs-Algorithmus. Diese Algorithmen verschlüsseln hierbei blockweise und jeweils unabhängig von anderen Blöcken.

**Definition 4.1**

Wir chiffrieren nzw. dechiffrieren *blockweise*, indem wir in einem ersten Schritt den Klartext in gleich lange Stücke schneiden, welche wir als *Blöcke* bezeichnen. In einem 2. Schritt erfolgt die Chiffrierung jedes einzelnen dieser Blöcke mit derselben Chiffrierungmethode. Die Länge eines Blocks, also die Anzahl Zeichen, welche er umfasst, entspricht der Schlüssellänge.

**Beispiel 4.1**

(1) Bei CAESAR beträgt die Blocklänge 1, da bei der Chiffrierung jeweils ein Buchstabe nach dem anderen durch eine Verschiebung um die jeweils gleiche Anzahl Positionen im Alphabet substituiert wird.

(2) Im diesem Beispiel betrachten wir eine blockweise Chiffrierung auf der Basis von VIGENERE.

(a) Wir werden den Klartext VIGENERE mit dem Schlüssel AZ chiffrieren. Die Blocklänge entspricht der Schlüssellänge und beträgt somit 2.

(b) Block 0 umfasst die 2 Buchstaben des Klartextes “V“ an Position 0 und “I“ an Position 1 des Blocks. Die Chiffrierung erfolgt nach dem bekannten Algorithmus von VIGENERE. Nach der Chiffrierung des Blocks 0 fährt man mit Block 1 in analoger Weise gemäss der nachfolgenden Abbildung fort.

(c) Die Abkürzung *Ord(b)* steht für die Ordnung des Buchstabens b im Alphabet und *Pos(p)* für den Buchstaben an der Position p. $Pos(k_p)$ bezeichnet somit den Buchstaben an der Position p im Klartext und und $Pos(g_p)$ den Buchstaben an der Position p im Geheimtext.

| Block n | n = 0 | | n = 1 | | n = 2 | | n = 3 | |
|---|---|---|---|---|---|---|---|---|
| Position p | 0 | 1 | 2 | 3 | 4 | 5 | 6 | 7 |
| Klartext k | V | I | G | E | N | E | R | E |
| Schlüssel s | A | Z | A | Z | A | Z | A | Z |
| $Ord(Pos(k_p))$ | 21 | 8 | 6 | 4 | 13 | 4 | 17 | 4 |
| $Ord(Pos(s_p))$ | 0 | 25 | 0 | 25 | 0 | 25 | 0 | 25 |
| $Ord(Pos(k_p)) + Ord(Pos(s_p))$ | ↓ | ↓ | ↓ | ↓ | ↓ | ↓ | ↓ | ↓ |
| $Ord(Pos(g_p))$ | 21 | 7 | 6 | 3 | 13 | 3 | 17 | 3 |
| Geheimtext g | V | H | G | D | N | D | R | D |

## Beispiel 4.2

Wir betrachten im Folgenden eine einfache, auf Substitution basierende Geheimschrift:

(1) Algorithmus: Ein Buchstabe an der Position p des Klartextes k ersetzen wir durch einen anderen Buchstaben, indem wir zur Ordnung dieses Buchstabens die Ordnung des direkt darauffolgenden Buchstaben an der Position p + 1 addieren und aus der Summe den Rest der ganzzahligen Division durch die Anzahl Zeichen in Alphabet berechnen. Die resultierende Ordnung des Buchstabens $g_p$ an der Position p im Geheimtext wird somit folgendermassen berechnet:

$$Ord\Big(Pos(g_{\mathrm{p}})\Big) = Ord\Big(Pos(k_{\mathrm{p}})\Big) + Ord\Big(Pos(k_{\mathrm{p+1}})\Big) \bmod 26$$

(2) Besonderheit: Bei der Chiffrierung des letzten Buchstabens des Klartextes verwenden wir als Position p + 1 das erste Zeichen des Klartextes.

(3) Mit diesem Algorithmus chiffrieren wir nun das Wort "KONTEXT" folgendermassen:

| Position p | 0 | 1 | 2 | 3 | 4 | 5 | 6 |
|---|---|---|---|---|---|---|---|
| Klartext k | K | O | N | T | E | X | T |
| Ord(Pos($k_p$)) | 10 | 14 | 13 | 19 | 4 | 23 | 19 |
| Ord(Pos($k_{p+1}$)) | 14 | 13 | 19 | 4 | 23 | 19 | 10 |
| (Ord(Pos($k_p$)) + Ord(Pos($k_{p+1}$))) mod 26 | ↓ | ↓ | ↓ | ↓ | ↓ | ↓ | ↓ |
| Ord(Pos($g_p$)) | 24 | 1 | 6 | 23 | 1 | 16 | 3 |
| Geheimtext g | Y | B | G | X | B | Q | D |

Abweichend zu den bisher bekannten Geheimschriften wird in Beispiel 4.2(2) ein Buchstabe nicht nur auf der Basis eines Schlüssels chiffriert, sondern wir verwenden einen benachbarten Buchstaben des Klartextes. Das bedeutet, dass beispielsweise der Buchstabe "A" im Wort "Affe" (mit "F") nicht gleich chiffriert wird wie im Wort "Anker" (mit "N").

## Definition 4.2

Wir betrachten auf Substitution basierende Geheimschriften, in welchen jeder Buchstabe durch ein Symbol bzw. eine Symbolfolge ersetzt wird. Die Geheimschrift ist *kontextfrei*, wenn für jeden Buchstaben K an einer Position k die Änderung eines Buchstaben im Klartext auf einer anderen Position als k keinen Einfluss auf die Chiffrierung von K auf der Position k hat. Geheimschriften, die nicht kontextfrei sind, bezeichnen wir als *kontextsensitiv*

## Beispiel 4.3

Bei der Verschlüsselung in Kryptosystem Caesar wird jeder Buchstabe mit einem anderen ersetzt, der die Anzahl Positionen der Geheimschrift entsprechend rechts davon im Alphabet steht. Diese Verschlüsselung wird auf jeden Buchstaben des Klartextes unabhängig von allen anderen Buchstaben identisch angewendet. Das Caesar-

Kryptosystem ist somit kontextfrei.

In der kontextsensitiven Geheimschrift im Beispiel 4.2(2) ist die Chiffrierung jedes Buchstabens abhängig vom direkt nachfolgenden Buchstaben im Klartext. Für eine praktikable Geheimschrift ist neben der Chiffrierung die Dechiffrierung ebenso wichtig. Genau hier haben wir bei dieser Chiffrierung jedoch eine Herausforderung. Da der Geheimtext zur Chiffrierung den direkt nachfolgenden Buchstaben aus dem Klartext verwendet und zudem beim letzten Buchstaben auf den ersten Klartextbuchstaben zurückgreift, sind diese zur Dechiffrierung notwendigen Informationen im Augenblick der Dechiffrierung nicht bekannt. Somit ist eine solche nicht ohne Weiteres möglich. In diesem Beispiel wurde eine uns bereits bekannte, wichtige Eigenschaft eines Chiffrierungsfunktion nicht berücksichtigt: eine solche Funktion muss injektiv, also umkehrbar sein.

**Definition 4.3**

Der Algorithmus einer praktikablen Geheimschrift sollte die folgenden Eigenschaften haben

(a) muss sich kurz beschreiben lassen

(b) sowohl die Chiffrierung als auch die Dechiffrierung muss effizient ausführbar sein

(c) die Chiffrierfunktion muss injektiv sein

In nachfolgenden Kapitel 4.2 werden wir ein Beispiel kennenlernen, das auf praktikablen Geheimschriften basiert.

**Aufgabe 4.1**

Geben Sie für jedes der nachfolgenden Kryptosysteme an, ob es kontextsensitiv oder kontextfrei ist:

(1) Vigenere

(2) Ein Buchstabe an der Position p des Klartextes k ersetzen wir durch einen anderen Buchstaben, indem wir zur Ordnung dieses Buchstabens die Ordnung des ersten Buchstabens des Klartextes, d.h. jenem an der Position 0, addieren und aus der Summe den Rest der ganzzahligen Division durch die Anzahl Zeichen in Alphabet berechnen. Die resultierende Ordnung des Buchstabens an der Position p im Geheimtext g wird somit folgendermassen berechnet:

$$Ord\Big(Pos(g_{\mathrm{p}})\Big) = Ord\Big(Pos(k_0)\Big) + Ord\Big(Pos(k_{\mathrm{p}})\Big) \bmod 26$$

(3) Homophones Kryptosystem

**Lösung der Aufgabe 4.1**

Die Kryptosysteme sind:

(1) kontextfrei, der Chiffrierungsalgorithmus berechnet jeden Buchstaben des Geheimtextes ausschliesslich mittels des an identischer Position befindlichen Buchstabes des

Klartextes, d.h. unabhängig von allen anderen Buchstaben

(2) kontextsensitiv, der Chiffrierungsalgorithmus berechnet jeden Buchstaben des Geheimtextes mittels des an identischer Position befindlichen Buchstabes und zusätzlich dem ersten Buchstaben des Klartextes, d.h. jeweils abhängig von diesem ersten Buchstaben

(3) kontextfrei, der Chiffrierungsalgorithmus berechnet jeden Buchstaben des Geheimtextes ausschliesslich mittels des an identischer Position befindlichen Buchstabes des Klartextes, d.h. unabhängig von allen anderen Buchstaben

Bisher haben wir Geheimschriften betrachtet. Die Verallgemeinerung von Kontextsensitivität in Geheimschriften auf Kryptosysteme erfolgt folgendermassen:

**Definition 4.4**

Wir bezeichnen ein Kryptosystem als *kontextsensitiv*, wenn mindestens eine seiner Geheimschriften kontextsensitiv ist.

## 4.2 Ein Kontextsensitives Kryptosystem basierend auf Caesar

Wie wir bereits wissen, lässt sich das Kryptosystem von Vigenere einfach basierend auf der prozentualen Häufigkeit der Buchstaben brechen, wenn die Schlüssellänge bekannt ist. Die Schlüssellänge kann mittels Kasiski-Tests oder der Friedmann'schen Charakteristik bestimmt werden. Wir werden daher als nächstes versuchen, die Buchstabenhäufigkeit zu verschleiern und so diese bekannten stochastische Kryptoanalyse-Ansätze zu erschweren. Ein möglicher Ansatz wäre die Einführung von kontextsensitiven Elementen. In einem ersten Schritt werden wir dies am Kryptosystem Caesar tun.

**Aufgabe 4.2**

Wie könnte ein einfacher, kontextsensitiver Algorithmus für die Substitution im Kryptosystem Caesar aussehen?

**Lösung Aufgabe 4.2**

Es gibt viele mögliche Lösungen, nachfolgend 2 nicht abschliessende Beispiele:

- Zusätzlich zu dem zu substituierenden Buchstaben an der Position n wird der nächste sich im Klartext rechts davon befindliche Buchstabe an der Position n + 1 mit einbezogen. Die Substitution erfolgt durch zweimalige Anwendung des Caesar-Algorithmus, einmal mit der Verschiebung gemäss Schlüssel und einmal mit der Verschiebung um die Ordnung des Buchstabens an der Position n + 1. Beim letzten Buchstaben des Klartextes wird kontextfrei substituiert.

  Anmerkung: Im Gegensatz zu Beispiel 4.2(2) ist hier eine Dechiffrierung möglich, indem rückwärts, d.h beginnend mit dem letzten Buchstaben des Geheimtextes dechiffriert wird.

- Zusätzlich zum zu chiffrierenden Buchstaben an der Position n wird der nächste sich im Klartext links davon befindliche Buchstabe an der Position n - 1 mit einbezogen. Die Chiffrierung erfolgt durch zweimalige Anwendung Caesar-Chiffrierung, einmal mit der Verschiebung gemäss Schlüssel und einmal mit der Verschiebung um die Ordnung

des Buchstabens an der Position n - 1. Beim ersten Buchstaben des Klartextes mit Position n = 0 wird CAESAR kontextfrei angewendet.

Ein kontextsensitiver Algorithmus wie in der Aufgabe 4.2 verschleiert die Buchstabenhäufigkeit, indem neben dem zu chiffrierende Buchstaben beispielsweise ein weiterer Buchstabe aus dem Klartext in die Chiffrierung miteinbezogen wird. Neben dieser gewünschten Erweiterung, müssen die übrigen Anforderungen an einen sicheres Kryptosystem weiterhin bestand haben. So muss Funktion zur Berechnung des Geheimtextes injektiv und das Kerckhoffs'sche Prinzip erfüllt sein.

**Definition 4.5**

Wir definieren eine Variante CAESAR+ des Kryptosystems CAESAR basierend auf den folgenden Elementen:

- Klartext mit Buchstaben $k \in \{a, b, ..., z\}$
- Geheimtext mit Buchstaben $g \in \{a, b, ..., z\}$
- Schlüssel $s \in \{a, b, ..., z\}$
- initialer Schlüssel $i \in \{a, b, ..., z\}$
- temporärer Schlüssel $t \in \{a, b, ..., z\}$

Die Verschlüsselung erfolgt mittels des nachfolgenden Algorithmus:

(1) Für jeden zu verschlüsselnden Buchstaben berechnen wir einen kontextsensitiven Schlüssel basierend auf dem Schlüssel s sowie dem in der vorausgegangenen Verschlüsselung verwendeten temporären Schlüssel t:

$$Ord(t) = Ord(s) + Ord(t) \bmod(26)$$

(2) Für die Berechnung des ersten temporären Schlüssels t gilt:

$$Ord(t) = Ord(s) + Ord(i) \bmod(26)$$

(3) Die Verschlüsselung erfolgt nach dem bekannten Verfahren CAESAR:

$$Ord(g) = Ord(k) + Ord(t) \bmod(26)$$

**Beispiel 4.4**

Wir verschlüsseln nun folgendes Beispiel mit dem Kryptosystem CAESAR+: Klartext "AVE CAESAR", Schlüssel s = "F" und initialer Schlüssel i = "X".

(1) Wir berechnen im ersten Schritt den temporären Schlüssel t basierend auf dem Schlüssel s und dem initialen Schlüssel t: $Ord(t) = Ord(\text{“}F\text{“}) + Ord(\text{“}X\text{“}) \bmod(26)$ bzw. $Ord(t) = 5 + 23 \bmod(26) = 2$

(2) Darauf basierend verschlüsseln wir den ersten Buchstaben "A" des Klartextes durch $Ord(\text{“}A\text{“}) + 2 \bmod(26) = 0 + 2 \bmod(26) = 2$, was dem Buchstaben "C" entspricht.

(3) Nun berechnen den neuen temporären Schlüssel t: $Ord(t) = 5 + 2 \bmod(26) = 7$

(4) Damit Verschlüsseln wir den 2. Buchstaben $Ord(\text{“}V\text{“}) + 7 \bmod(26) = 21 + 7 \bmod(26) = 2$, was erneut dem Buchstaben “C“ entspricht.

(5) Dies machen wir für sämtliche Buchstaben der Klartextes.

**Aufgabe 4.3**

Wie lautet der vollständige Geheimtext für das obenstehende Beispiel “AVE CAESAR“ mit Schlüssel s = “F“ und initialem Schlüssel i = “X“?

**Lösung Aufgabe 4.3**

Wir führen die gesamte Berechnung direkt auf Basis der Ordnung der Buchstaben und Schlüssel durch.

(1) Die Ordnung der Buchstaben des Klartextes lautet: 0|21|4|2|0|4|18|0|17

(2) Die Berechnung der Ordnung der temporären Schlüssel erfolgt folgendermassen:

- $Ord(t_0) = Ord(\text{“}F\text{“}) + Ord(\text{“}X\text{“}) \bmod(26)$ bzw. $Ord(t_0) = Ord(5) + Ord(23) \bmod(26) = 2$
- $Ord(t_1) = Ord(5) + Ord(2) \bmod(26) = 7$
- $Ord(t_2) = Ord(5) + Ord(7) \bmod(26) = 12$
- $Ord(t_3) = Ord(5) + Ord(12) \bmod(26) = 17$
- $Ord(t_4) = 22 | Ord(t_5) = 1 | Ord(t_6) = 6 | Ord(t_7) = 11 | Ord(t_8) = 16$

(3) Durch Addition der Ordnung der Buchstaben des Klartextes mit jener des temporären Schlüssels mod 26 bekommen wir als Resultat die Ordnung der Buchstaben des Geheimtextes: 2|2|16|19|22|5|24|11|7. Dies entspricht dem Geheimtext “CCQTWFYLH“.

**Aufgabe 4.4**

Wir haben gelernt, wie man mit CAESAR+ verschlüsseln kann. Nun fehlt uns allerdings noch der Entschlüsselungs-Algorithmus.

(1) Wie könnte dieser Algorithmus aussehen. Bitte beschreiben Sie ihn analog zum Verschlüsselungs-Algorithmus in Definition 4.4. Hinweis: Stellen Sie sicher, dass die berechnete Ordnung des Klartextes Ordnung(k) in jedem Fall einen positiven Wert hat.

(2) Verifizieren Sie Ihren Algorithmus durch Entschlüsselung des Geheimtextes “QR“ und dem Schlüssel s = “F“ und dem Initialschlüssel i = “X“.

**Lösung Aufgabe 4.4**

Entschlüsselung bei CAESAR+

(1) Die Entschlüsselung erfolgt mittels des nachfolgenden Algorithmus:

(a) Die Berechnung des temporären Schlüssels t erfolgt identisch wie bei der Verschlüsselung.

(b) Die Entschlüsselung erfolgt analog zum bekannten Verfahren von Caesar. Somit kann basierend auf dem vorliegenden Geheimtext g und dem berechneten temporären Schlüssel t der Klartext folgendermassen berechnet werden: $Ord(k) = Ord(g) - Ord(t) + 26 \bmod(26)$. Um negative Zahlen und damit ein nicht definierter Wert für die $Ord(k)$ im Falle von $Ord(t) \geq Ord(g)$ zu verhindern, zählen wir 26, die Anzahl Zeichen des Alphabets, hinzu. Da wir im Anschluss den Rest einer ganzzahligen Division durch die Anzahl Zeichen des Alphabets (26) berechnen, verändert sich hierbei die resultierende $Ord(k)$ für Werte grösser oder gleich 0 nicht.

(2) Wir führen die gesamte Berechnung direkt auf Basis der Ordnung der Buchstaben und Schlüssel durch.

(a) Die Ordnung der Buchstaben des Geheimtextes lautet: 16|17

(b) Die berechnete Ordnung der temporären Schlüssel lautet: 2|7

(c) Nach der Subtraktion der Ordnung der Buchstaben des temporären Schlüssels von jener des Geheimtextes sowie der Addition von 26 rechnen wir das Resultat mod 26 und bekommen so die Ordnung der Buchstaben des Klartextes: 14|10. Dies entspricht dem Klartext "OK".

Mit caesar+ haben wir ein Kryptosystem kennengelernt, welches in jedem Schritt einen neuen Schlüssel verwendet, der jeweils auf Basis des zuvor verwendeten Schlüssels verändert wird und so eine Kontextsensitivität auf Ebene des Schlüssels schafft. Da wir bei diesem Ansatz neben dem eigentlichen Schlüssel s zusätzlich einen initialen Schlüssel i verwenden, existieren insgesamt 26 x 26 = 676 Schlüsselkombinationen. Um Caesar+ anzugreifen, könnte man diese beispielsweise durchprobieren. Das ist zwar deutlich aufwändiger als bei Caesar aber noch immer möglich.

## 4.3 Kontextsensitives Kryptosystem basierend auf Vigenere.

Am anschaulichen Beispiel von Caesar+ haben wir das Konzept der Kontextsensitivität kennen und anwenden gelernt. Dieses Kryptosystem ist allerdings aufgrund der kurzen Block- und Schlüssellänge von 1 sowie einer zwar kontextsensitiven, jedoch überschaubaren Anzahl an Schlüsselkombinationen relativ leicht brechbar. In diesem Abschnitt werden wir daher schrittweise gemeinsam ein neues Kryptosystem entwerfen und testen, welches eine relativ höhere Sicherheit im Vergleich zu Caesar+ erreichen soll. Hierzu werden wir das Ihnen bereits bekannte Kryptosystem von Vigenere mit dem Konzept der Kontextsensitivität zu einem neuen Kryptosystem Vigenere+ kombinieren, indem wir den in vorausgegangenen Blöcken erstellten Geheimtext für die Chiffrierung und Dechiffrierung nutzen. Wir stellen hierbei sicher, dass alle Anforderungen an eine praktikable Geheimschrift gemäss Definition 4.3 berücksichtig werden, insbesondere auch die Injektivität der Chiffrierfunktion.

**Definition 4.6**

Wir definieren eine Variante Vigenere+ des Kryptosystems Vigenere basierend auf den folgenden Elementen:

- Klartext mit Buchstaben $k \in \{a, b, ..., z\}$

- Geheimtext mit Buchstaben $g \in \{a, b, ..., z\}$
- Schlüssel mit Buchstaben $s \in \{a, b, ..., z\}$
- Blocklänge $n \in \mathbb{N}$, entspricht der Länge des Schlüssels

Die Verschlüsselung erfolgt mittels des nachfolgenden Algorithmus:

(1) Wir unterteilen den Klartext k in Blöcke der Länge n.

(2) Die Verschlüsselung erfolgt in 2 Schritten:

   (a) Im ersten Schritt wird ein Block B des Klartextes der Länge n gemäss dem Originalverfahren von VIGENERE durch Substitution basierend auf der Ordnung der Buchstaben des Schlüssels s verschlüsselt.

   (b) Im zweiten Schritt stellen wir die Kontextsensitivität sicher, indem wir die Ordnung der Buchstaben des Geheimtextes g des vorausgegangenen Blocks B-1 zur Ordnung der Buchstaben des aktuellen Blocks sowie des Schlüssels jaddieren und daraus den Rest der ganzzahligen Division durch die Anzahl Buchstaben des Alphabets (26) kalkulieren.

   (c) Für die Berechnung des ersten Blocks (B = 0) liegt kein Geheimtext aus dem vorausgegangenen Block vor, der für die Verschlüsselung genutzt werden könnte. Daher verwenden wir in diesem Fall stattdessen eine kontextfreie Verschlüsselung.

Somit wird für jeden Block in der Verschlüsselung die Chiffrierung in zwei Schritten vorgenommen. Zuerst durch den Schlüssel s und danach durch den Geheimtext als Chiffrierung des vorherigen Blocks. Formal betrachtet berechnen wir einen Buchstaben des Geheimtextes an der Position $a \in \{1, 2, ..., n-1\}$ innerhalb eines Blocks B folgendermassen:
Für $B = 0$ gilt

$$Ord\Big(Pos\big(g_{\mathrm{B}}(a)\big)\Big) = Ord\Big(Pos\big(k_{\mathrm{B}}(a)\big)\Big) + Ord\Big(Pos\big(s(a)\big)\Big) \bmod(26)$$

Für $B > 0$ gilt

$$Ord\Big(Pos\big(g_{\mathrm{B}}(a)\big)\Big) = Ord\Big(Pos\big(k_{\mathrm{B}}(a)\big)\Big) + Ord\Big(Pos\big(s(a)\big)\Big) + Ord\Big(Pos\big(g_{\mathrm{B\text{-}1}}(a)\big)\Big) \bmod(26)$$

**Beispiel 4.5**

Wir verwenden die Blocklänge 2 und verschlüsseln den Klartext "CAESAR" mit dem Schlüssel "OK" nach VIGENERE+. Aus der Länge des Schlüssels von 2 ergibt sich die Blocklänge 2:

| Block B | B = 0 | | B = 1 | | B = 2 | |
|---|---|---|---|---|---|---|
| Position a | 0 | 1 | 0 | 1 | 0 | 1 |
| Klartext k | C | A | E | S | A | R |
| Schlüssel s | O | K | O | K | O | K |
| Ord(Pos($k_B$(a))) | 2 | 0 | 4 | 18 | 0 | 17 |
| Ord(Pos(s(a))) | 14 | 10 | 14 | 10 | 14 | 10 |
| Ord(Pos($g_{B-1}$(a))) | 0 | 0 | 16 | 10 | 8 | 12 |
| (Ord(Pos($k_B$(a))) + Ord(Pos(s(a))) + Ord(Pos($g_{B-1}$(a)))) mod 26 | ↓ | ↓ | ↓ | ↓ | ↓ | ↓ |
| Ord(Pos($g_B$(a)) | 16 | 10 | 8 | 12 | 22 | 13 |
| Geheimtext g | Q | K | I | M | W | N |

## Aufgabe 4.5

In dieser Aufgabe verändern wir nun am Beispiel 4.5 einen einzelnen Buchstaben im Klartext und beobachten die Auswirkungen davon auf den Geheimtext.

(1) Wir verwenden weiterhin VIGENERE+ mit einer Blocklänge 2. Wenden Sie den Verschlüsselung-Algorithmus auf den Klartext "ZAESAR" und den Schlüssel "OK" an. Wie lautet der Geheimtext?

(2) Was fällt Ihnen auf? Ist für Sie diese Beobachtung eher überraschend oder können Sie sie erklären?

## Lösung Aufgabe 4.5

Die Auswirkungen der Veränderung eines Buchstabens sind beträchtlich:

(1) Der Geheimtext kann folgendermassen berechnet werden:

| Block B | B = 0 | | B = 1 | | B = 2 | |
|---|---|---|---|---|---|---|
| Position a | 0 | 1 | 0 | 1 | 0 | 1 |
| Klartext k | Z | A | E | S | A | R |
| Schlüssel s | O | K | O | K | O | K |
| Ord(Pos($k_B$(a))) | 25 | 0 | 4 | 18 | 0 | 17 |
| Ord(Pos(s(a))) | 14 | 10 | 14 | 10 | 14 | 10 |
| Ord(Pos($g_{B-1}$(a))) | 0 | 0 | 13 | 10 | 5 | 12 |
| (Ord(Pos($k_B$(a))) + Ord(Pos(s(a))) + Ord(Pos($g_{B-1}$(a)))) mod 26 | ↓ | ↓ | ↓ | ↓ | ↓ | ↓ |
| Ord(Pos($g_B$(a)) | 13 | 10 | 5 | 12 | 19 | 13 |
| Geheimtext g | N | K | F | M | T | N |

(2) Der Geheimtext NKFMTN für die Verschlüsselung von ZAESAR unterscheidet sich sehr stark vom Geheimtext QKIMWN der Verschlüsselung von CAESAR obwohl

im Klartext nur 1 Buchstabe verändert wurde. Genauer betrachtet hat jeweils der 1. Buchstabe eines jeden Blockes einen anderen Wert und der jeweils 2. ist unverändert. Die Ursache hierfür liegt in der Kontextsensitivität des Algorithmus begründet. Da dieser das Resultat der Verschlüsselung aus dem vorausgegangenen Block an derselben Stelle für die Verschlüsselungsoperation des Folgeblock nutzt, hat die Veränderung eines Buchstabens eine Auswirkung auf alle nachfolgenden Buchstaben an derselben Position innerhalb eines Blocks.

**Aufgabe 4.6**

Die Definition 4.6 hat an der einen oder anderen Stelle noch Verbesserungspotenzial. Gelingt es Ihnen dieses zu identifizieren?

(1) Für ein funktionierendes Kryptosystem fehlt ein wichtiges Element. Was könnte das sein? Hinweis: Überlegen Sie sich hierzu, wie Sie beim Empfang eines mit VIGENERE+ verschlüsselten Geheimtextes vorgehen würden.

(2) Die Berechnung des ersten Blocks (B = 0) erfolgt ohne Verwendung des Geheimtextes aus dem vorausgegangenen Block und somit kontextfrei. Finden Sie alternative Möglichkeiten, für die Verschlüsselung dieses ersten Blocks? Hinweis: Betrachten Sie nochmals das Kryptosystem CAESAR+, dort wird einer von diversen möglichen Ansätzen verwendet.

**Lösung Aufgabe 4.6**

(1) Der Entschlüsselungs-Algorithmus ist nicht definiert.

(2) Es gibt vielen denkbare Ansätze hierfür. Nachfolgend sind einige nicht abschliessende Beispiele aufgeführt:

   (a) Es wird im Algorithmus ein fixer Wert definiert und für jede Position innerhalb des ersten Blocks anstelle des Geheimtextes des vorausgegangenen Blocks zur Ordnung des Buchstabens des aktuellen Blocks addiert, z.B. 10.

   (b) Es wird der Schlüssel s als Ersatz für den Geheimtextes des vorausgegangenen Blocks verwendet.

   (c) Es wird analog zu CAESAR+ ein zusätzlicher, initialer Schlüssel i als Ersatz für den Geheimtextes des vorausgegangenen Blocks verwendet.

In Aufgabe 4.6 haben wir festgestellt, dass bei der aktuellen Definition von VIGENERE+ der Entschlüsselungs-Algorithmus noch nicht definiert ist. Bei der Entschlüsselung von CAESAR+ haben wir den Algorithmus der Verschlüsselung umgedreht, d.h. wir haben von der Ordnung eines Buchstabens im Geheimtext die Ordnung des temporären Schlüssels subtrahiert. Durch die Subtraktion entstand jedoch das Risiko, dass die Ordnung des resultierenden Klartextes negativ werden konnte. Da eine negative Ordnung eines Buchstabens nicht definiert ist, haben wir vorgängig sicherheitshalber die Anzahl Buchstaben des Alphabets (26) addiert. Der Entschlüsselungs-Algorithmus für VIGENERE+ kann nun ganz analog zu jenem im CAESAR+ aufgebaut werden.

### Aufgabe 4.7

Wie könnte ein Entschlüsselungs-Algorithmus aussehen? Testen Sie ihren Algorithmus anhand des in Beispiel 4.5 berechneten Geheimtextes QKIMWN. Bekommen Sie den Klartext "CAESAR"?

### Lösung Aufgabe 4.7

Die Entschlüsselung kann analog zu CAESAR+ und wie bei Verschlüsselung von VIGENERE+ in 2 Schritten erfolgen, indem wir einerseits die Ordnung des Schlüssels subtrahieren und andererseits basierend auf der Ordnung der Buchstaben aus dem vorausgegangenen Block kontextsensitiv in umgekehrter Richtung substituieren:

$$Ord\Big(Pos\big(k_{\mathrm{B}}(a)\big)\Big) = Ord\Big(Pos\big(g_{\mathrm{B}}(a)\big)\Big) - Ord\Big(Pos\big(s(a)\big)\Big) - Ord\Big(Pos\big(g_{\mathrm{B\text{-}1}}(a)\big)\Big) + 52 \bmod (26)$$

Um negative Zahlen in Folge der beiden Subtraktionen zu verhindern addieren wir für jede Subtraktion die Anzahl Zeichen des Alphabets, also $2 \cdot 26 = 52$.
Der Klartext kann folgendermassen berechnet werden:

| Block B | B = 0 | | B = 1 | | B = 2 | |
|---|---|---|---|---|---|---|
| Position a | 0 | 1 | 0 | 1 | 0 | 1 |
| Geheimtext g | Q | K | I | M | W | N |
| Schlüssel s | O | K | O | K | O | K |
| Ord(Pos($g_B$(a))) | 16 | 10 | 8 | 12 | 22 | 13 |
| Ord(Pos(s(a))) | 14 | 10 | 14 | 10 | 14 | 10 |
| Ord(Pos($g_{B-1}$(a))) | 0 | 0 | 16 | 10 | 8 | 12 |
| (Ord(Pos($g_B$(a))) - Ord(Pos(s(a))) - Ord(Pos($g_{B-1}$))) + 52) mod 26 | ⬇ | ⬇ | ⬇ | ⬇ | ⬇ | ⬇ |
| Ord(Pos($k_B$(a))) | 2 | 0 | 4 | 18 | 0 | 17 |
| Klartext k | C | A | E | S | A | R |

# 5 Zusammenfassung und Lernkontrolle

## 5.1 Zusammenfassung

Das Verschlüsselungsverfahren von Vigenère konnte mit dem Kasiski-Test oder mit der Friedman'scher Charakteristik gebrochen werden. Die Hauptursache war, dass sich Buchstaben im Geheimtext wiederholen und darauf aufbauend die Schlüssellänge bestimmt werden sowie mittels der Häufigkeitsverteilung im Anschluss der Geheimtext in den Klartext dechiffriert werden kann. Der Ansatz, mittels eines homophonen Kryptosystems die Häufigkeitsverteilung zu verschleiern, erschwert die oben erwähnte Methode zwar, dennoch können seltene Buchstaben und vor allem Doppelbuchstaben schnell gefunden und darauf basierend der Geheimtext entschlüsselt werden. Das ist zwar sehr viel aufwendiger als bei VIGENERE, aber mit viel Geduld und Erfahrung ist es möglich. Dies zeigte, dass auch die homophonen Kryptosystemen nicht absolut sicher sind.

Einen Schritt weiter in die Richtung von modernen Kryptosysteme bringt uns die Chriffrierung von Klartext Blöcken mit Geheimtext Blöcken oder das Konzept der Kontextsensitivität. Es können auch beide Ideen kombiniert werden. Bei der Kontextsensitivität verschleiern wir die Häufigkeitsverteilung der Buchstaben, indem der eingesetzte Algorithmus bei der Ver- oder Entschlüsselung eines Blocks eine Abhängigkeit zu einem anderen Block schafft oder der Schlüssel abhängig von anderen Schlüsseln bei jedem Durchlauf verändert wird. Am einfachen, jedoch aufgrund der Blockgrösse von nur 1 wenig sicheren Beispiel von CAESAR+ haben wir die kontextsensitive Veränderung des Schlüssels kennengelernt. Bei der Erweiterung von VIGENERE auf VIGENERE+ erfolgte die Ver- und Entschlüsselung kontextsensitiv durch Verwendung des im vorausgegangenen Block erzeugten Geheimtextes. Wir haben hierbei die Blockgrösse 2 verwendet. Durch eine Erweiterung der Blockgrösse lässt sich der Aufwand für ein Brechen, beispielsweise mit einer Klartext-Attacke, deutlich erhöhen.

## 5.2 Selbsttest

(1) Wie kann man testen, ob ein Klartext mit einem monoalphabetischen Kryptosystem verschlüsselt worden ist?

(2) Bitte erkläre die Grundidee der Verschlüsselung beim homophonen Kryptosystem. Welche Schwäche wird durch die homophone Verschlüsselung behoben?

(3) Wie kann das homophone Kryptosystem trotzdem geknackt werden?

(4) Woran erkennt man ein kontextsensitives Kryptosystem?

(5) Ist das homophone Kryptosystem kontextsensitiv oder kontextfrei? Weshalb?

## Lösungen

(1) Falls die Häufigkeit aller Buchstaben stark unterschiedlich ist.

(2) Beim homophonen Kryptosystem werden die häufigsten Buchstaben durch mehrere Buchstaben ersetzt, dadurch fallen sie bei der Analyse der Häufigkeitsverteilung mit der Fried man'sche Charakteristik nicht auf.

(3) Doppelte selten vorkommende Buchstaben, wie z.B. PP deuten auf den Code für P hin, da das P nur auf eine Zahl abgebildet wird. Oder allgemein mit der Suche nach speziellen Bigramme, wie PP, MM, CH, CK, QU usw.

(4) Die Ver- und Entschlüsselung erfolgt nicht nur auf Basis eines statischen Schlüssels, sondern der Geheimtext wird in jedem Schritt abhängig von einem vorausgegangenen generiert.

(5) Das homophone Kryptosystem ist kontextfrei, weil die Änderung eines beliebigen anderen Buchstaben des Klartextes keinen Einfluss auf die Verschlüsselung des chiffrierten Buchstabens hat.

## 5.3 Zusätzliche Aufgaben

(1) Entwickle eine kontextsensitive Variante des homophonen Kryptosystems.

# 6 Ein möglichst einfaches modernes Kryptosystem

## 6.1 Präambel

Dieser Abschnitt adressiert Lehrer, Lehrerinen, Schülerinnen und Schüler die sich in der Kryptographie vertiefen wollen. Das Kapitel ist eher ein Ausblick zur Fortsetzung und Vertiefung des Themas im Unterricht als eine ausgearbeitete Unterrichtssequenz wie in den vorhergehenden Kaptiteln.
Es wird einiges an Vorwissen z.B. über die Häufigkeitsverteilung von Buchstaben in den verschiedenen Sprachen oder auch an Informatik Wissen vorausgesetzt. So sollten die Bitoperationen XOR, AND, OR, Bitnegation und die binäre Addition bekannt sein. Es sollte bekannt sein wie die 4 Buchstaben 0, 1, 2, 3 mit 2 Bits codiert werden können. Geläufigkeit in einer Programmiersprache wie Java oder C-Sharp, die die genannten Bitoperationen unterstützt ist von Vorteil.

Es werden zentrale Konzepte und Ideen der modernen Kryptographie vorgestellt. Mit ihrer Hilfe sollte der Zugang und das Verständnis von heute eingesetzten Kryptosystemen wie z.B. IDEA ermöglicht bzw. erleichtert werden.

Die Darstellung ist im Gegensatz zu den vorhergehenden Kapiteln knapp.
Die Lösung der Aufgaben ist vielfach nur skizzenhaft. Oft gibt es nicht eine, sondern viele Lösungen einer Aufgabe. Weiterhin sollten die Leser sich ihre eigenen Gedanken zu einer Frage machen und für sich den Lösungsraum einer Frage oder Aufgabe entdecken.

## 6.2 Einführung

In Abschnitt 3 haben wir homophone Verschlüsselungen kennen gelernt. In Abschnitt 4 haben wir kontextsensitive Verschlüsselungen kennen gelernt.

Sind wir zufrieden? Nicht wirklich!
- Homophone Verschlüsselungen bieten keinen Schutz gegen Kryptoanalyse mit Bi-, Tri- oder Tetragrammen.
- Der Schlüssel ist lang und kompliziert.
- Für Texte mit unterschiedlichen Häufigkeitsverteilungen müssen bei homophonen Verfahren unterschiedliche Verschlüsselungen definiert und für die Verschlüsselung eines Textes gewählt werden – nicht wirklich praktikabel!
- Es ist nicht klar wie und unter welchen Bedingungen mit kontextsensitiven Verschlüsselungen die geforderte grosse Anzahl von Schlüssel realisiert werden kann.

Moderne Kryptosysteme basieren auf Kerckhoffs Prinzip.

Wir erinnern uns:
Ein Kryptosystem ist sicher, wenn das Kryptosystem öffentlich bekannt ist und es trotzdem ohne Kenntnis des verwendeten Schlüssels nicht möglich ist, aus einem Geheimtext den ursprünglichen Klartext abzuleiten.

Eine Interpretation von Kerckoffs Prinzip:
Der Schlüssel kann nicht einfach ermittelt werden wenn der Klartext, der Geheimtext und das Verschlüsselungsverfahren bekannt ist.

Aus dieser Perspektive gibt es Zweifel an der Sicherheit kontextsensitiver Verschlüsselungen.

**Kommentar** Kerckhoffs Prinzip:
Das Kerckhoffs Prinzip erlaubt eine Trennung des Ver- und Entschlüsselungsverfahrens

vom Schlüssel. Aufgrund dieser Trennung können die beiden folgenden zentralen Fragen gestellt und getrennt voneinander bearbeitet werden:

- Was sind geeignete Ver- und Entschlüsselungsverfahren?
  Von einem geeigneten Verschlüsselungsverfahren könnte z.B. gefordert werden, dass kleine Änderungen im Schlüssel zu grossen Änderungen im Geheimtext führen.
- Wie muss der Schlüssel für eine gewünschte Sicherheitsstufe beschaffen sein?

Ein Kryptographie Experte, oder wer sich dafür hält, könnte sofort mehr oder weniger gut Definitionen für die Begriffe klein und gross liefern. Er könnte auch mehr oder weniger schlüssige Anforderungslisten für die Ver- und Entschlüsselungsverfahren und Schlüssel präsentieren.

Die Lehrperson ist kein Kryptographie Experte und die Schüler und Schülerinnen auch nicht. Hierfür müssen sie eine Entwicklung durchlaufen. In der Schule wird für diese Entwicklung der Begriff Lernen verwendet. Ein Ingenieur verwendet den Begriff Entwickeln und ein Forscher untersucht eine Fragestellung aus dem Gebiet der Kryptographie.
Der Name ist unterschiedlich, die Tätigkeit ist die gleiche. Am Anfang steht eine Frage oder Aufgabe und die ist im Moment die Suche nach einer Verschlüsselung mit vielen Schlüsseln.
Hierzu gehören Ideen, Vorschläge für mögliche Verbesserungen und ihre Prüfung. Ergibt die Prüfung ein negatives Resultat wird der Vorschlag verworfen, ansonsten wird weiterentwickelt.

Das heisst:
Fehler und Irrtümer gehören dazu und sollten konstruktiv genutzt werden.

Wir starten also mit einem intuitiven Verständnis von klein und gross und suchen Ideen für geeignete Verschlüsselungen.

Eine Bemerkung zum Kerckhoffs Prinzip:
Das Kerckhoffs Prinzip wurde 1883 formuliert. Lange, bis in die 90-iger Jahre wurde in weiten Kreisen davon ausgegangen, dass das Geheimnis sowohl im Verschlüsselungsverfahren wie auch im Schlüssel liegt. Entsprechend wurde das Verschlüsselungsverfahren als geheim betrachtet und nur einer sehr kleinen Community zugänglich gemacht. Dies weil Verschlüsselungsverfahren bis Anfang 90-iger Jahre fast nur vom Militär verwendet wurden. Erst mit der Vernetzung von Computern in den 90-iger Jahren wurde die vertrauenswürdige und vertrauliche Kommunikation zwischen Computern und damit die Kryptographie ein Thema für die Informatik.

Die ungenügende Würdigung des Kerckhoffs Prinzips hat im harmloseren Fall dazu geführt, dass unsichere Verfahren als sicher betrachtet wurden. Hier liegt ein Beispiel von Security-by-Obscurity vor.
Im weniger harmlosen Fall wurden in das Verschlüsselungsverfahren Schwachstellen eingebaut. Mit der Kenntnis der Schwachstellen kann der Geheimtext von den interessierten Stellen wie z.B. Geheimdiensten einfacher entschlüsselt werden. Berühmt hierfür ist der Fall der Crypto AG.
Nett ist das nicht! In einer guten Gesellschaft muss ich mich darauf verlassen können, dass Zahlungsdaten nicht gefälscht sind, dass ich mit einem echten und keinem gefakten Server kommuniziere, und meine vertraulichen Daten auch vertraulich bleiben.
Noch in den 90-iger Jahren haben das Geheimdienste und auch die amerikanische Regierung anders gesehen. So hat die amerikanische Regierung noch in den 90-iger Jahren wichtige Verschlüsselungsverfahren wie RSA mit einem Exportverbot belegt und damit ein vertrauensvolles Zusammenleben behindert.

Im Gegensatz hierzu, wenn das Kerckhoffs Prinzip angewendet wird:
Das Verschlüsselungsverfahren ist öffentlich und kann von einer grossen Community z.B. auf seine Sicherheit geprüft werden. Seriöse Partner schlagen geprüfte Verfahren für ein vertrau-

ensvolles Zusammenleben vor. Natürlich sind auch bei Kryptosystemen die auf der Basis von Kerckhoffs Prinzip entwickelt wurden Fehleinschätzungen möglich aber eher unwahrscheinlich.

Ab den 90-iger Jahren wurde das Kerckhoffs Prinzip bei der Entwicklung von sicheren Kryptosystemen angewendet. Die grossen Verbesserungen im Bereich der Verschlüsselungsverfahren in den letzten 20 bis 30 Jahren ist durch die Wechselwirkung zwischen der Entwicklung von Verschlüsselungsverfahren und der Kryptoanalyse zurückzuführen.

## 6.3 Beschreibung eines möglichst einfachen, modernen Kryptosystems

Ein Kryptosystem heisst modern wenn es das Kerckhoffs Prinzip beachtet.

Beschreibung:
- Wir betrachten die Buchstaben 0, 1, 2, 3. Der Klartext, der Geheimtext und der Schlüssel bestehen aus diesen Buchstaben.
- Der Schlüssel S kann aus mehreren Buchstaben bestehen. Zunächst besteht er nur aus einem Buchstaben $S_1$.
- Der Klartext Buchstabe $K_x$ wird mit dem Schlüsselbuchstaben $S_y$ gemäss der Verschlüsselungstabelle in den Geheimtext Buchstaben $C_z$ verschlüsselt.

| Text<br>Schlüssel | 0 | 1 | 2 | 3 |
|---|---|---|---|---|
| 0 | 0 | $1_8$ | $2_9$ | $3_{11}$ |
| 1 | $1_3$ | 0 | $3_7$ | $2_{10}$ |
| 2 | $2_2$ | $3_4$ | 0 | $1_6$ |
| 3 | $3_0$ | $2_1$ | $1_5$ | 0 |

Wie funktioniert das?
Nehmen wir an, dass 2 der Schlüsselbuchstabe ist.
Damit werden die Klartext Buchstaben 0, 1, 2, 3 in die Geheimtext Buchstaben 2, 3, 0, 1 transformiert.

Die Indices der Tabelleneinträge werden für die weitere Darstellung benötigt und sind im Moment ohne Bedeutung.

Nehmen wir an, der Geheimtext Buchstabe ist 2. Dieser Buchstabe wird mit dem Schlüsselbuchstaben 2 gemäss der Verschlüsselungstabelle in den Buchstaben 0 transformiert. Aber das ist ja genau der ursprüngliche Klartext Buchstabe!

**Behauptung:** Das funktioniert für alle anderen Buchstaben auch!

> **Aufgabe 6.1**
>
> Die Schülerinnen und Schüler sollen das nachprüfen.

Beim vorgestellten Verschlüsselungsverfahren kann also mit dem gleichen Schlüssel und der gleichen Verschlüsselungsoperation ver- und entschlüsselt werden.

**Aufgabe 6.2**

Die Schülerinnen und Schüler sollen die Auffälligkeiten der Verschlüsselungstabelle identifizieren und Interpretationen suchen.

Kommentar Auffälligkeiten der Verschlüsselungstabelle

- Aufffällig ist der 0 Schlüssel – es findet keine Verschlüsselung statt. Ist das ein Problem? Nicht wirklich! In einer Testumgebung kann dieser Schlüssel mit dieser Eigenschaften viel helfen. In einer Produktivumgebung darf dieser Schlüssel einfach nicht zufällig gewählt werden.
- Die Tabelle ist symmetrisch. Der Schlüssel kann als Text und der Text als Schlüssel betrachtet werden.
- Die Diagonale ist der 0 Buchstabe. Gleiche Buchstaben werden in den 0 Buchstaben verschlüsselt. Das zusammen mit der Symmetrie der Verschlüsselungstabelle ist das ein Ausdruck dafür, dass für die Ver- und Entschlüsselung der gleiche Buchstaben verwendet werden kann.
  Also:
  $K_x \otimes S_y = G_z$ Behauptung: $G_z \otimes S_y = K_x$
  $(K_x \otimes S_y) \otimes S_y = K_x \otimes (S_y \otimes S_y) = K_x \otimes 0 = K_y$

  Bemerkung:
  Das Zeichen $\otimes$ bedeutet verschlüsselt. Müssen mehrere Verschlüsselung unterschieden werden wird neben dem Symbol $\otimes$ auch noch das Symbol $\oplus$ für alternative Verschlüsselung 1 oder $\odot$ für alternative Verschlüsselung 2 verwendet.
- Es treten in der Tabelle ausserhalb der Hauptdiagonalen die 3 Geheimbuchstaben 1, 2, 3 auf. Diese Zahlen können jetzt gegeneinander ausgetauscht werden. Es ergibt sich eine neue Verschlüsselungstabelle und damit eine neues Krytographiesystem!

  Die beiden folgenden Verschlüsselungstabellen sind Beispiele hierfür.

| Text<br>Schlüssel | 0 | 1 | 2 | 3 |
|---|---|---|---|---|
| 0 | 0 | $1_8$ | $3_7$ | $2_{10}$ |
| 1 | $1_3$ | 0 | $2_9$ | $3_{11}$ |
| 2 | $3_0$ | $2_1$ | 0 | $1_6$ |
| 3 | $2_2$ | $3_4$ | $1_5$ | 0 |

| Text<br>Schlüssel | 0 | 1 | 2 | 3 |
|---|---|---|---|---|
| 0 | 0 | $2_9$ | $2_{10}$ | $3_0$ |
| 1 | $3_4$ | 0 | $1_8$ | $3_{11}$ |
| 2 | $1_3$ | $1_5$ | 0 | $3_7$ |
| 3 | $2_1$ | $2_2$ | $1_6$ | 0 |

An Hand der Indices der Tabelleneinträge können die Veränderungen gegenüber der ursprünglichen Verschlüsselungstabelle nachvollzogen werden. Bei der zweiten Verschlüsselungstabelle wurden die Tabelleneinträge mit Index um eine Position nach links verschoben.

Bemerkung:
Die Frage, welche Eigenschaften die resultierende Verschlüsselung hat ist interessant. Wenn das resultierende Kryptosystem gute Eigenschaften wie z.B. die Entschlüsselbarkeit hat, kann das Kryptosystem als Baustein für ein komplexes Kryptosystem betrachtet werden.

Hinweis:
Das vorgestellte Kryptographiesystem ist das kleinste, das diese Möglichkeit zur Bildung neuer Kryptographiesysteme bietet.

Die Buchstaben 0, 1, 2, 3 können durch die Bitfolgen 00, 01, 10, 11 ersetzt werden. Es ergeben sich die beiden folgenden Verschlüsselungstabellen (links Buchstabendarstellung, rechts Bitdarstellung):

| Text<br>Schlüssel | 0 | 1 | 2 | 3 |
|---|---|---|---|---|
| 0 | 0 | 1 | 2 | 3 |
| 1 | 1 | 0 | 3 | 2 |
| 2 | 2 | 3 | 0 | 1 |
| 3 | 3 | 2 | 1 | 0 |

| Text<br>Schlüssel | 00 | 01 | 10 | 11 |
|---|---|---|---|---|
| 00 | 00 | 01 | 10 | 11 |
| 01 | 01 | 00 | 11 | 10 |
| 10 | 10 | 11 | 00 | 01 |
| 11 | 11 | 10 | 01 | 00 |

**Aufgabe 6.3**

Die Schüler und Schülerinnen sollen überlegen mit welcher Bitoperation aus dem Klartext und dem Schlüssel der Geheimtext gemäss der rechten Verschlüsselungstabelle berechnet wurde.

Hinweis:
Die beiden Bit von Klartext und Schlüssel ergeben unabhängig voneinander die beiden Bits des Geheimtextes.

Das Zeichen $\otimes$ bedeutet verschlüsselt. Gemäss der Verschlüsselungstabelle ist:

$0 \otimes 0 = 0$
$0 \otimes 1 = 1$
$1 \otimes 0 = 1$
$1 \otimes 1 = 0$

Welche Bitoperation ist $\otimes$?

**Lösung und Kommentar:**
Für die Berechnung der Verschlüsselungstabelle wurde die binäre XOR Funktion verwendet.

Die XOR Funktion ist in der Krypotographie von grosser Bedeutung und ein guter Freund. Dies weil sie die Bit von zwei Operanden sehr gut vermischt.

Wenn der Schlüssel aus zwei unterschiedlichen Schlüsselbuchstaben besteht kann auch doppelt verschlüsselt werden. Zuerst wird mit dem ersten Teilschlüssel verschlüsselt und dann mit dem zweiten Teilschlüssel oder umgekehrt.

**Aufgabe 6.4**

Die Schüler und Schülerinnen sollen mit der doppelten Verschlüsselung experimentieren. Hierfür sollen sie als Schlüssel das Buchstabenpaar 1 3 verwenden. Als erstes sollen sie überlegen wie entschlüsselt werden kann.

**Kommentar:**
Die Schüler und Schülerinnen sollen entdecken, dass eine doppelte Verschlüsselung genau so funktioniert wie mit einen neuen Schlüssel $S3 = S1 \otimes S2$

**Was ist gewonnen?**
Sehr viel, da die Alphabete vom echten Kryptographiesystem ohne weiteres nicht aus $2^2 = 4$

Buchstaben wie im Unterricht, sondern aus $2^{32}$ Buchstaben bestehen könnten. Die Ver- und Entschlüsselung kann mit der XOR Funktion einfach berechnet werden. Und wir haben etwa $10^{10}$ und damit, gemäss unseren Anforderungen, knapp genügend viele Schlüssel.

Allerdings kann der Schlüssel einfach berechnet werden wenn der Klartext und der Geheimtext bekannt ist.

**Idee:**
Wenn zwei oder mehr Verschlüsselungen kombiniert werden wäre das vermutlich nicht mehr so einfach möglich.

Wir suchen also noch eine zweite Verschlüsselung die in etwa so wie die XOR Verschlüsselung funktioniert. Ein guter Kandidat ist eine spezielle Form der Addition $\oplus$, bei der die 4 zum Buchstaben 0 wird, die 5 wird zum Buchstaben 1 und die 6 wird zum Buchstaben 2.
Das ist in etwa so wie bei dem Ziffernblatt einer Uhr bei dem nach der 12 das Zählen wieder bei der 1 beginnt.

Genauso wie die XOR Funktion vermischt die Addition $\oplus$ die Bits der Operanden sehr gut.

Die angesprochene Form der Addition funktioniert wie folgt:
- Die Verschlüsselungsoperation $\oplus$ ist kommutativ.
- Die Addition der Buchstaben $0 \oplus 0$ ergibt den Buchstaben 0.
  Die Addition der Buchstaben $0 \oplus 1$ ergibt den Buchstaben 1.
  Die Addition der Buchstabenm $0 \oplus 2$ und $1 \oplus 1$ ergibt den Buchstaben 2.
  Die Addition der Buchstabten $1 \oplus 2$ und $0 \oplus 3$ ergibt den Buchstaben 3.
- Die Addition der Buchstaben $2 \oplus 2$ und $1 \oplus 3$ ergibt den Buchstaben 0.
- Die Addition der Buchstaben $3 \oplus 2$ ergibt den Buchstaben 1.
- Die Addition der Buchstaben $3 \oplus 3$ ergibt den Buchstaben 2.

Betrachten wir jetzt die Entschlüsselung:
Nehmen wir an der Schlüssel wäre 1. 1 kann aber nicht für die Entschlüsselung verwendet werden, da $1 \oplus 1$ den Buchstaben 2 ergibt. Wir benötigen also für die Entschlüsselung das Inverse von 1, $\overline{1}$, so dass $1 \oplus \overline{1}$ den Buchstabe 0 ergibt. Der Buchstabe 3 ist unser Kandidat!
Es ergeben sich die folgenden inversen Buchstaben: $\overline{0} = 0, \overline{1} = 3, \overline{2} = 2, \overline{3} = 1$

Dies weil unter den aufgeführten Additionsregeln z.B. $1 \oplus \overline{1} = 1 \oplus 3 = 0$ gilt. Es ergeben sich die folgenden Tabellen:

Verschlüsselungstabelle:

| Text<br>Schlüssel | 0 | 1 | 2 | 3 |
|---|---|---|---|---|
| 0 | 0 | 1 | 2 | 3 |
| 1 | 1 | 2 | 3 | 0 |
| 2 | 2 | 3 | 0 | 1 |
| 3 | 3 | 0 | 1 | 2 |

Entschlüsselungstabelle:

| Text<br>Schlüssel | 0 | 1 | 2 | 3 |
|---|---|---|---|---|
| $\overline{0} = 0$ | 0 | 1 | 2 | 3 |
| $\overline{1} = 3$ | 3 | 0 | 1 | 2 |
| $\overline{2} = 2$ | 2 | 3 | 0 | 1 |
| $\overline{3} = 1$ | 1 | 2 | 3 | 0 |

**Bemerkung:**
Die vorgestellte Form der Addition ist eine Modulo 4 Addition. Sie kann einfach mit einer Addition und z.B. einer Modulo Operation realisiert werden.
Auch der inverse Schlüssel kann einfach berechnet werden. Es werden alle Bit invertiert und dann noch 1 addiert.

Im folgenden sollen die beiden Verschlüsselungen $\otimes$ und $\oplus$ mit unterschiedlichen Teilschlüsseln kombiniert werden. Dies weil wir auf der Grundlage von Kerckhoffs Prinzip sichere Verschlüsselungsverfahren entwickeln wollen – und die sind nun einmal nur vom Schlüssel abhängig.

**Aufgabe 6.5**

Die Schülerinnen und Schüler sollen die beiden Verschlüsselungen $\otimes$ und $\oplus$ mit unterschiedlichen Teilschlüsseln kombinieren.

Hinweis:
Die Schüler und Schülerinnen sollen in einem ersten Schritt davon ausgehen, dass der Schlüssel S aus den beiden Teilschlüsseln S1 und S2 besteht.

**Kommentar** Lösungungen
Es gibt sehr viele Lösungsmöglichkeiten. Die aufgeführten Lösungsmöglichkeiten sind besonders einfach.

- $K_i \otimes S_1 \otimes S_2$.
  Das bringt nicht viel, da ja $K_i \otimes S_3$ mit $S_3 = S_1 \otimes S_2$ betrachtet werden kann. Wenn aber für die Berechnung von $S_3$ statt $\otimes$ $\oplus$ verwendet wird schaut alles gleich ganz anders aus.
- Hieraus ergibt sich sofort:
  $Gi = ((Ki \otimes S1) \oplus S2$ und $Gi + 1 = ((K_{i+1} \otimes S_2) \oplus S_3$ mit $S_3 = S_1 \oplus S_2$
- Interessant könnte auch zu sein:
  $G_i = ((K_i \otimes S_1) \oplus (K_{i+1} \otimes S_3))$
  und
  $G_{i+1} = ((K_i \otimes S_2) \oplus (K_{i+1} \otimes S3))$

**Hinweis** zur Methodik:
Im ersten Schritt ist das generieren von Ideen, was gemacht werden könnte, wichtig. Vielleicht ist die erste Idee nicht so gut, aber sie liefert den Auslöser für die zweite Idee, die dann wirklich gut ist. Die Wege zur Erkenntnis sind oft verschlungen und manchmal braucht es auch etwas Hartnäckigkeit.
Dann im zweiten Schritt müssen die Ideen geprüft, und aufgrund dem Resultat der Prüfung verworfen oder weiterentwickelt werden.

Eine nahe liegende Prüfung ist die Entschlüsselbarkeit.

Eine weitere Prüfung ist die Abschätzung des Aufwandes für die Ermittlung des Schlüssels:
Eine Verschlüsselung kann als gut betrachtet werden, wenn bei bekanntem Klartext und Geheimtext die Berechnung des Schlüssels schwierig ist. Für die Analyse können auch sehr regelmässige "pathologische"Klartexte hilfreich sein.
0000000000000000 oder 3333333333333333
1111111111111111 oder 2222222222222222

**Resumée Moderne Kryptographie:**
Selbst für die im Unterricht verwendeten sehr einfachen Verschlüsselungen mit wenigen Buchstaben und wenig möglichen Schlüsseln können wir Kryptographie Systeme zusammenstellen bei denen der Schlüssel nur mit erheblichen Aufwand aus einem Geheimtext ermittelt werden kann. Dies auch wenn der Klartext und der Geheimtext bekannt sind.
Weiterhin kann die Anzahl der Schlüssel der im Unterricht vorgestellten Kryptosysteme ohne jeglichen Aufwand z.B. auf $2^{32}$ oder $2^{64}$ erhöht werden. Die Erhöhung der Schlüsselzahl erhöht die benötigte Zeit für die Verschlüsselung oder Entschlüsselung nur vernachlässigbar.

# 7 Anhang

## 7.1 Concept map

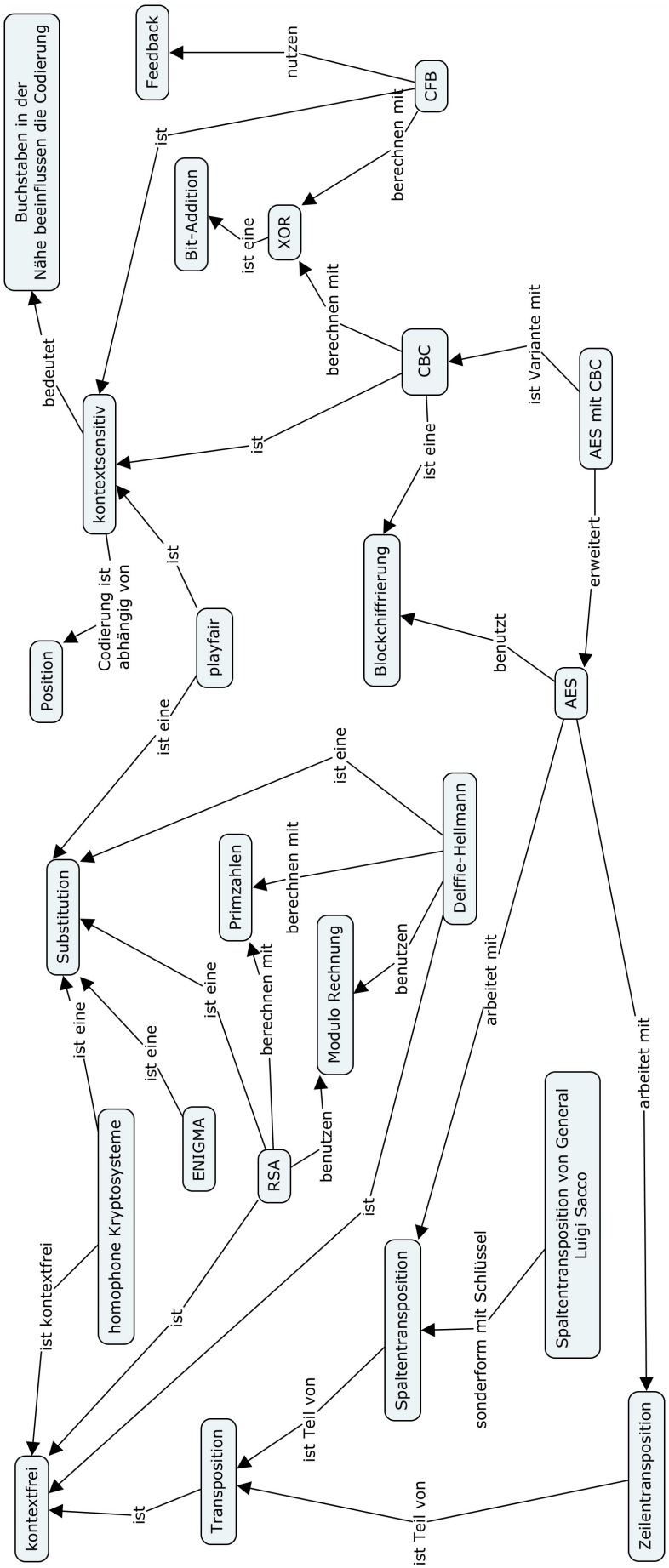


Abbildung 2: Concept Map

# Literatur

[1] Michael Barot, Britta Dorn, Gishlain Fourny, Jens Gallenbacher, Juraj Hromkovič, and Regula Lacher. *Informatik Data Science und Sicherheit*. Klett und Balmer Verlag.

[2] Bernhard Esslinger. *Das CrypTool-Buch: Kryptographie lernen und anwenden mit CrypTool und SageMath*. 2 edition, 2018.

[3] Karin Freiermuth, Juraj Hromkovič, Lucia Keller, and Björn Steffen. *Einführung in die Kryptologie*. 2 edition, 2010, 2014.