Kryptographie

Studienbrief 2

universität wien

Klassische Verschlüsselungsverfahren

Gerald und Susanne Teschl

Druckfehler/Feedback bitte an:
gerald.teschl@univie.ac.at

SS 23

Version:
2023-06-30

Studienbrief 2

Klassische Verschlüsselungsverfahren

Inhalt

2.1 Grundbegriffe

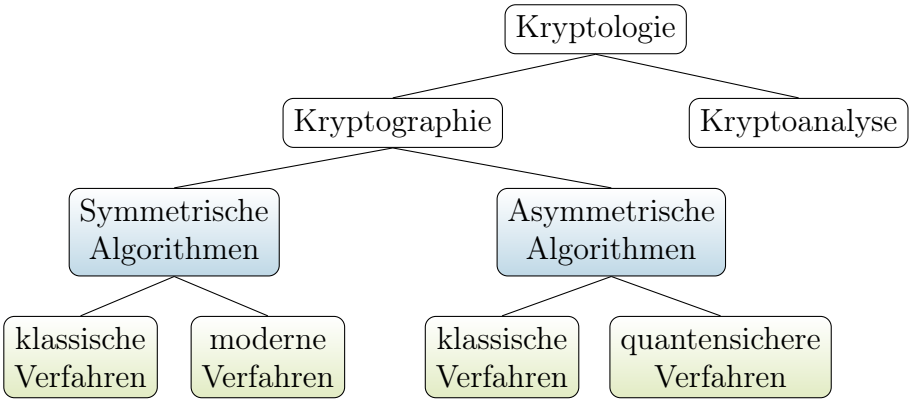
Der Einsatz kryptographischer Verfahren dient unterschiedlichen Zielsetzungen (**Sicherheitsdienste**, ***security services***), siehe z.B. [Bar20]:

- **Vertraulichkeit/Geheimhaltung (*confidentiality*):** Das Lesen einer Nachricht soll für Unbefugte unmöglich gemacht werden.
- **Authentizität (*authentication*):** Der Sender kann gegenüber dem Empfänger seine Identität beweisen bzw. der Empfänger kann sicher sein, dass die erhaltene Nachricht nicht von einem anderen (unbefugten) Absender stammt.
- **Integrität/Unversehrtheit (*data integrity*):** Die Nachricht wird davor geschützt, dass sie von Unbefugten verändert wird.
- **Verbindlichkeit/Nichtzurückweisbarkeit (*non-repudiation*):** Der Sender kann nachträglich nicht abstreiten, eine Nachricht gesendet zu haben.

Die Bezeichnungen *Kryptographie* und *Kryptologie* werden in der Literatur unterschiedlich verwendet, meistens findet man aber folgende Einteilung: **Kryptographie** bedeutet die Wissenschaft der Sicherung von Nachrichten durch Verschlüsselung, **Kryptoanalyse** (auch: **Kryptanalyse**) ist die Wissenschaft, eine verschlüsselte Nachricht ohne Kenntnis des geheimen Schlüssels zu entschlüsseln und die **Kryptologie** vereinigt beide Wissenschaften.

Kryptographie und Kryptologie zu verwechseln ist also weniger dramatisch als die Verwechslung von Astronomie und Astrologie.

Kryptographische Verfahren kann weiter wie folgt einteilen:



- **Symmetrische Verfahren:** Sie werden seit der Antike eingesetzt. Sender und Empfänger besitzen dabei einen gemeinsamen geheimen Schlüssel. Verfahren bis etwa die 1950er Jahre werden als **klassische Verfahren** bezeichnet (z.B. Cäsar-Verschiebung, Vigenère-Verschlüsselung, Enigma). Das derzeit wichtigste **moderne symmetrische Verfahren** ist der Advanced Encryption Standard (AES).

- **Asymmetrische Verfahren:** Sie wurden in den 1970er Jahren entwickelt (z.B. der RSA-Algorithmus). Dabei besitzt jeder Kommunikationsteilnehmer einen eigenen geheimen und einen eigenen öffentlichen Schlüssel. Sie ermöglichen neben der klassischen Verschlüsselung auch andere Funktionen wie z.B. digitale Signaturen. Die klassichen Verfahren können mit Quantencomputern gebrochen werden. Seit ca. 2005 gibt es intensive Bemühungen sichere Alternativen zu entwickeln und zu standardisieren.

Für eine genauere Beschreibung benötigen wir zunächst ein paar Grundbegriffe: Ein **Alphabet** A ist eine endliche Menge, zum Beispiel das Alphabet der 26 Buchstaben $A = \{a, b, c, \ldots, z\}$, die Zahlen $A = \mathbb{Z}_{26} = \{0, 1, 2, \ldots, 25\}$, der ASCII Code $A = \{\texttt{NUL}, \texttt{SOH}, \ldots, \sim, \texttt{DEL}\}$, oder auch $A = \mathbb{Z}_2 = \{0, 1\}$. Eine **Nachricht** m (engl. *message*) über dem Alphabet A ist eine endliche Folge aus Zeichen von A:

$$m = a_1 \cdots a_k, \qquad \text{mit } k \in \mathbb{N} \quad \text{und } a_1, \ldots, a_k \in A.$$

Beispiele: „*dasisteinenachricht*“ ist eine Nachricht über dem Alphabet der 26 Buchstaben $A = \{a, b, c, \ldots, z\}$. „00“ oder „1001111“ sind Nachrichten über dem Alphabet $\{0, 1\}$.

Die Nachricht x, die verschlüsselt werden soll, wird **Klartext** (*plaintext*) genannt. Sie wird aus Zeichen des **Klartextalphabets** gebildet. Die verschlüsselte Nachricht y wird als **Geheimtext** oder **Chiffretext** (*ciphertext*) bezeichnet und aus Buchstaben des **Geheimtextalphabets** gebildet. Klartext- und Geheimtextalphabet können gleich oder verschieden sein.

Der Verschlüsselungsvorgang wird auch **Chiffrieren** (*encryption*) genannt, das Entschlüsseln auch als **Dechiffrieren** (*decryption*) bezeichnet.

Die Kommunikationspartner heißen in der Literatur typischerweise „Alice“ und „Bob“. Ein unbefugter Dritter („Angreifer“) heißt meist „Oskar”, „Mallory“ (von *opponent* bzw. *malicious*; damit meint man in der Regel Angreifer, die aktiv eingreifen) oder „Eve“ (passive Angreiferin, von *eavesdrop* = lauschen).

Was ist die Grundidee einer symmetrischen Verschlüsselung?
Alice und Bob einigen sich auf einen Verschlüsselungsalgorithmus E (z.B. im einfachsten Fall die Cäsar-Verschiebung) und vereinbaren einen gemeinsamen geheimen Schlüssel k (*key*) über einen sicheren Kanal (z.B. vorab in einem persönlichen Treffen). Algorithmus E und Schlüssel k ergeben zusammen die (bijektive) Verschlüsselungsfunktion E_k bzw. ihre Umkehrfunktion (Entschlüsselungsfunktion) $E_k^{-1} = D_k$.

Alice	**unsicherer Kanal**	**Bob**
kennt k		kennt k
verschlüsselt Klartext x zum Geheimtext $y = E_k(x)$		
	$y \longrightarrow$	
		entschlüsselt $x = D_k(y)$

Angreiferin Eve kann zwar y belauschen, ist aber nicht in der Lage, x aus y zu ermitteln, da sie k nicht kennt.

Früher gab es Algorithmen ohne geheimen Schlüssel, bei denen die Sicherheit der Verschlüsselung nur darauf beruhte, dass der *Verschlüsselungsalgorithmus* geheimgehalten wird. Das hat sich in der Praxis aber aus mehreren Gründen als problematisch herausgestellt: (1) Wenn eine Person die Gruppe verläßt (z.B. die Firma), dann muss der Algorithmus geändert werden. (2) Ist der Algorithmus in eine Maschine (oder Software) verpackt, so kann ein Angreifer durch Analyse der Maschine (des Maschinenprogramms) den Algorithmus rekonstruieren. (3) Qualitätskontrolle findet nicht in ausreichendem Maß statt, weil das entwickelte

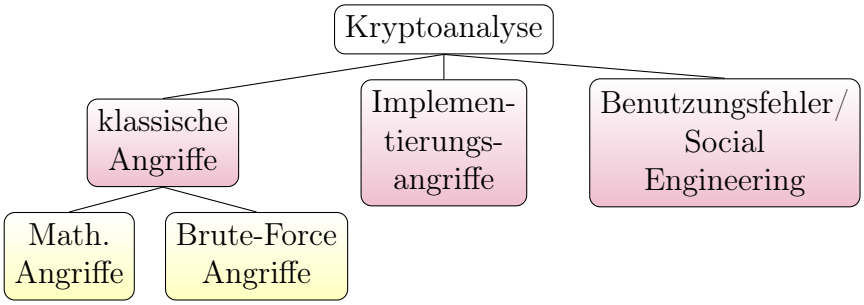
Verfahren nicht der Kritik und den Angriffen der Öffentlichkeit bzw. von Wissenschaftler:innen standhalten muss.

Bereits im 19. Jahrhundert hat deshalb der Niederländer Auguste Kerckhoffs (1835–1903) gefordert (**Kerckhoffs'sches Prinzip**):

> *Die Sicherheit eines Verschlüsselungsverfahrens darf nur von der Geheimhaltung des Schlüssels abhängen, nicht jedoch von der Geheimhaltung des Algorithmus.*

Die Beachtung dieses Prinzips ist ein wichtiger Bestandteil der modernen sogenannten **starken Kryptographie**. Die Sicherheit einer Verschlüsselung darf nicht darunter leiden, dass der Algorithmus öffentlich bekannt wird. Im Gegenteil: Die Praxis hat gezeigt, dass durch eine möglichst frühzeitige Veröffentlichung der Algorithmen die Sicherheit eines Kryptosystems erheblich vergrößert wird. Denn wenn ein Algorithmus der Kritik der Öffentlichkeit ausgesetzt wird und den Attacken von Expert:innen standhält, dann erst bewährt er sich und stärkt auch das Vertrauen der Benutzer:innen in die Sicherheit des Verschlüsselungsverfahrens.

Grundsätzlich kann man also davon ausgehen, dass einem Angreifer der verwendete Algorithmus (nicht aber der geheime Schlüssel) bekannt ist. Die Methoden, die ein Angreifer anwenden kann, kann man grob so einteilen:



Klassische Angriffe:

- Suche nach **mathematischen Schwächen** in der Struktur des Verschlüsselungsverfahrens (z.B. statistische Häufigkeitsanalyse, lineare und differentielle Kryptoanalyse); Suche nach bestimmten Parameterwerten (z.B. schwache Schlüssel), die zu Schwachstellen in der Verschlüsselung führen.
- **Brute-Force-Angriff**, bei dem alle möglichen Schlüssel durchprobiert werden.

Implementierungsangriffe: Der Angriff nutzt

- **kryptographische Implementierungsfehler** (z.B. Verwendung schwacher Parameter, mehrfache Verwendung von Einmalzahlen (Nonce), Verwendung vorhersagbarer *Zufallszahlen*, ...),

- **Programmierfehler** („software bug“, inkl. mutwillig eingebauter Schwachstellen) oder
- Nebeninformationen (**Seitenkanalangriff**) (z.B. Stromverbrauch, elektromagnetische Abstrahlung, ...)

um den geheimen Schlüssel zu ermitteln. Diese Angriffe erfordern, dass man Zugriff auf die Software (evtl. sogar den Quellcode) bzw. die Hardware (z.B. Chipkarte) hat.

Benutzungsfehler/Social Engineering: Der Angriff versucht häufige Anwenungsfehler auszunutzen bzw. durch Betrug bewusst zu provozieren:

- Man nutzt **schwach gewählte Geheimnisse** (z.B. zu einfaches Passwort), die sich erraten bzw. durch Vorberechnen (**Wörterbuchangriff**, Rainbow-Table) durchprobieren lassen. Auch das klassische Post-it mit dem Passwort unter der Tastatur gehört hier dazu ;)
- Man täuscht falsche Tatsachen vor, z.B. **Man in the Middle Angriff**, **Phishing**.

Der einzige Weg, um an den geheimen Schlüssel zu kommen, ist aber heute oft der, eine Person mit Zugang zum geheimen Schlüssel zu bedrohen, zu überlisten oder zu bestechen. Bei den Verfahren der starken Kryptographie ist also meist der Mensch die größte Sicherheitslücke.

2.2 Sicherheit kryptographischer Verfahren

Als Nächstes wollen wir uns überlegen, wann wir ein Kryptosystem als *sicher* betrachten. Wie immer gibt es einen Unterschied zwischen Theorie und Praxis. Man bezeichnet ein Kryptosystem als

- **praktisch sicher** (auch: berechenbarkeitstheoretisch sicher, *computationally secure*), wenn der zum Entschlüsseln nötige Aufwand den Wert der verschlüsselten Daten übersteigt oder die zum Aufbrechen notwendige Zeit größer ist als die Zeit, für die die Daten geheim gehalten werden müssen. Alle in der Praxis gängigen (korrekt implementierten, mit hinreichend langen Schlüsseln) Verfahren sind nach derzeitigem Wissensstand praktisch sicher, z.B. AES, RSA.
- **perfekt sicher** (auch: uneingeschränkt sicher, beweisbar sicher, *unconditionally secure*), wenn der Geheimtext alleine (ohne den zugehörigen Schlüssel) *keinerlei Rückschlüsse* auf den Klartext zulässt. In diesem Fall kann man auch mit beliebig großem Aufwand und aller Zeit der Welt den Geheimtext

nicht entschlüsseln. Das *One-Time-Pad*, auf das wir später zurückkommen, ist perfekt sicher.

Wie erwähnt, gibt es zwar ein perfekt sicheres System, dieses ist aber für praktische Anwendungen meist zu aufwendig und man verwendet daher in der Regel nur *praktisch* sichere Verfahren. Um die Sicherheit solcher Verfahren zu beurteilen, werden folgende **Angriffsszenarien** anhand der Möglichkeiten, die dem Angreifer zur Verfügung stehen, unterschieden ([DH76], p. 646 f.). Ziel ist es, aus den gegebenen Informationen den Klartext oder den geheimen Schlüssel zu finden.

- **Angriff mit bekanntem Geheimtext** (*ciphertext-only attack*): Der Angreifer besitzt nur ein (meist relativ großes) Stück Geheimtext. Abgesehen von gewissen elementaren Eigenschaften des Klartexts (z.B. die Sprache) hat er jedoch keine weiteren Informationen.

- **Angriff mit bekanntem Klartext** (*known-plaintext attack*): Der Angreifer besitzt zu einem Teil des Geheimtextes den zugehörigen Klartext. Er kennt z.B. das Dateiformat (und damit die ersten Bytes) oder er weiß, dass es sich um einen Geschäftsbrief handelt, der immer mit dem gleichen Briefkopf beginnt.

- **Angriff mit frei gewähltem Klartext** (*chosen-plaintext attack*): Der Angreifer kann einen beliebigen Klartext wählen und hat die Möglichkeit, an den zugehörigen Geheimtext zu gelangen. Durch geschickte Wahl des Klartextes und des erhaltenen Geheimtextes kann er nun Informationen über den verwendeten geheimen Schlüssel ableiten. Eine solche Art von Angriff ist z.B. möglich, wenn der Angreifer Zugang zur Maschine (mit aktuellem Schlüssel) hat, mit der verschlüsselt wird; oder bei Public-Key-Systemen, auf die wir später zurückkommen.

Das letzte Szenario mag auf den ersten Blick unwahrscheinlich erscheinen, aber Angreifer können sehr einfallsreich sein: Im zweiten Weltkrieg hat zum Beispiel die Royal Air Force bewusst in der Nordsee Minen abgeworfen. Darauf folgte eine mit einer ENIGMA verschlüsselte Minenwarnung mit genauen Positionsangaben der deutsche Wehrmacht, die dann anschließend von den britischen Kryptoanalytikern von Bletchley Park bei der Ermittlung des geheimen Schlüssels genutzt werden konnte.

Ein praktisch sicheres Verfahren muss gegen diese Angriffszenarien resistent sein. Das bedeutet, dass es in diesen Situationen keine Möglichkeit geben darf, den Klartext (oder den Schlüssel) mit praktikablem Aufwand zu ermitteln.

Natürlich ist es immer möglich, alle Schlüssel durchzuprobieren, was als **Brute-Force-Angriff** (auch: vollständige Schlüsselsuche, *brute force attack*, *exhaustive key search*), bekannt ist. **Im Mittel** muss dabei etwa die **Hälfte aller Schlüssel getestet** werden.

Denken Sie an das Beispiel eines Einbrechers mit einem Schlüsselbund mit 10 Schlüsseln, die er systematisch durchprobiert (Aufwärmübung 1). Man kann berechnen, dass er im Mittel 5.5

Sicherheits-niveau	Symmetrische Algorithmen	Asymmetrische Algorithmen	
		RSA, DSA DH, El Gamal über $\mathbb{Z}_p^*$	ECDH ECDSA
≤ 80	2DES	1024	160
112	3DES	2048	224
128	AES-128	3072	256
192	AES-192	7680	384
256	AES-256	15360	512

Tabelle 2.1: Empfohlene Schlüssellängen nach [Bar20]. Bei den symmetrischen Verfahren ist die Bitlänge des geheimen Schlüssels gemeint, bei den asymmetrischen Verfahren ist die Bitlänge von bestimmten, für die Sicherheit maßgeblichen, Parametern gemeint.

Schlüssel probieren muss, bis er den richtigen gefunden hat. Allgemein gilt: Bei n Schlüsseln ist der Erwartungswert der Anzahl der Versuche, bis der richtige gefunden ist, gleich $\frac{n+1}{2}$.

Ein Schlüssel ist in der Regel eine natürliche Zahl, dargestellt im Binärsystem. Die Anzahl der Bits, die für den Schlüssel zur Verfügung stehen, wird als **Schlüssellänge** bezeichnet. Ist die Schlüssellänge ℓ Bit, so gibt es 2^ℓ mögliche Schlüssel. Der notwendige Aufwand für einen Brute-Force-Angriff lässt sich durch Vergrößerung der Schlüssellänge beliebig hoch treiben.

Beispiel 2.1 Aufwand beim Brute-Force-Angriff bei verschiedenen Schlüssellängen
Wie viele mögliche Schlüssel der Länge a) 56 Bit b) 128 Bit gibt es?
Wie lange würde ein Brute-Force-Angriff in beiden Fällen im Mittel brauchen, wenn 10^9 Schlüssel pro Sekunde getestet werden können?

Lösung zu 2.1

a) $2^{56} \approx 7.2 \cdot 10^{16}$ Schlüssel. Die mittlere Testzeit ist die Zeit, die für die Hälfte der Schlüssel gebraucht wird: $3.6 \cdot 10^{16} \cdot 10^{-9} = 3.6 \cdot 10^7$ Sekunden, also über ein Jahr.

b) $2^{128} \approx 3.4 \cdot 10^{38}$ Schlüssel. Die mittlere Testzeit ist $1.7 \cdot 10^{29}$ Sekunden, also etwa $5.4 \cdot 10^{21}$ Jahre (das ist mehr als 10^{11} mal das Alter des Universums). ■

Man sagt, ein Algorithmus (symmetrisch / asymmetrisch) hat ein **Sicherheitsniveau** (***security strength***) von ℓ Bit, wenn der beste (klassische) Angriff von der Ordnung $O(2^\ell)$ ist. Von modernen symmetrischen Algorithmen wird erwartet, dass Brute-Force der beste (klassische) Angriff ist. In diesem Fall ist das Sicherheitsniveau gleich der Schlüssellänge. Bei asymmetrischen Algorithmen erreicht man

dasselbe Sicherheitsniveau, indem man Bitlängen („Schlüssellängen“) für bestimmte, für die Sicherheit verantwortliche, Parameter vorgibt, wie Tabelle 2.2 zeigt.

Wenn man zum Beispiel beim RSA-Algorithmus von der Schlüssellänge spricht, die für ein bestimmtes Sicherheitsniveau notwendig ist, so meint man die Bitlänge des RSA-Moduls. Das ist eine nicht geheimzuhaltende natürliche Zahl, die groß genug gewählt werden muss, damit der Aufwand für ihre Primfaktorzerlegung zu hoch für einen Angriff ist.

Um zu beurteilen, welche Schlüssellänge für praktische Sicherheit notwendig ist, muss man die Leistungsfähigkeit aktueller und zukünftiger Hardware einbeziehen. Hierbei hilft das **Moore'sche Gesetz**. Dabei handelt es sich um eine Faustregel, die besagt, dass die Rechenleistung von Computern sich bei gleichbleibenden Kosten etwa alle 18 Monate verdoppelt.

Gordon Moore (1929–2023), einer der Mitbegründer der Firma Intel, hat im Jahr 1965 beobachtet, dass sich in den vorangegangenen Jahren die Anzahl der elektronischen Bauteile einer integrierten Schaltung jährlich verdoppelt hatte. Er stellte damals die Frage, was wäre, wenn die Entwicklung in den folgenden zehn Jahre so weiterginge. Der von Moore beobachtete Zusammenhang gilt im Wesentlichen bis heute und es gibt dementsprechende Prognosen für die nächsten 10 Jahre. Dennoch ist allein aus physikalischen Gründen ein Ende dieses Wachstums absehbar, denn die Bauteile müssen laut Annahme des Mooreschen Gesetzes immer kleiner werden, ihre Kleinheit/Dichte ist aber physikalisch begrenzt.

Beispiel 2.2 Moore'sches Gesetz
Angenommen, es ist heute ein Budget von 1 Million Euro notwendig, um irgendeine Chiffre XY innerhalb eines Tages mittels Brute-Force zu brechen. Wenn wir die Gültigkeit des Moore'schen Gesetzes für die nächsten 10 Jahre voraussetzen: Mit welchem Budget kann man diese Chiffre in zehn Jahren brechen?

Lösung zu 2.2 Die Kosten halbieren sich alle 18 Monate, also alle 1.5 Jahre. Wenn wir heute $K(0) = 1\,000\,000$ Euro ausgeben müssen, so sind es in 3 Jahren nur mehr $K(3) = 1\,000\,000 \cdot \frac{1}{2^2} = 250\,000$ und allgemein in t Jahren nur mehr $K(t) = K(0) \cdot (\frac{1}{4})^{t/3}$ Euro. Insbesondere braucht man nach 10 Jahren nur noch $K(10) = K(0) \cdot (\frac{1}{4})^{10/3} = 9\,843$, also knapp 10 000 Euro. ■

Inzwischen sollte man auch die Energiekosten einer Schlüsselsuche nicht vernachlässigen: Auch hier gibt es physikalische Grenzen. Das Prinzip von Landauer besagt, dass alleine das Durchlaufen des gesamten Schlüsselraums bei 128 Bit mindestens so viel Energie benötigt, wie uns die Sonne in ca. einer Sekunde liefert (was um ein Vielfaches mehr ist, als die gesamte Menschheit derzeit erzeugt). In [Sch16], S. 101 ff, findet sich dazu folgende Überlegung: Angenommen, Mallory kann eines Tages einen Supercomputer bauen, der mittels Brute-Force-Angriff einen 128-Bit Schlüssel sehr viel schneller als „Vielfaches des Alters des Universums“ knackt. Dann müssten sämtliche Kraftwerke der Welt über 100 Jahre lang ausschließlich für Mallory arbeiten, um einen Schlüssel zu knacken. Die dazu notwendigen Kosten wären deutlich höher als das derzeitige weltweite Bruttosozialprodukt. Zudem würden die Siliziumatome im Universum knapp werden, wenn ein solcher Spezialrechner gebaut werden soll.

Weitreichende Prognosen über die Sicherheit kryptographischer Verfahren sind jedenfalls schwer, denn abgesehen von der Abschätzung der Zunahme der Leistungs-

fähigkeit von Rechnern ist auch der wissenschaftliche Fortschritt kryptoanalytischer Verfahren einzubeziehen. Für die gängigen Verfahren gibt es nämlich keinen mathematischen Beweis, dass es keine besseren Angriffs-Algorithmen gibt als die bisher bekannten. Das Sicherheitsniveau eines Verfahrens kann sich also jederzeit ändern, wenn neue Algorithmen bekannt werden. Im schlimmsten Fall kann ein Algorithmus also über Nacht unsicher werden. Unter https://www.keylength.com findet sich ein Überblick über Empfehlungen bekannter akademischer und nichtakademischer Organisationen für minimale Schlüssellängen.

Zusammenfassend ist es also wichtig, bei der Wahl der Schlüssellänge eine entsprechend hohe Reserve einzuplanen, denn:

- Es könnten Schwachstellen gefunden werden, die das Sicherheitsniveau reduzieren.
- Bei gewissen Angriffen (z.B. Meet in the Middle Angriff, Geburtstagsangriff) ist die Schlüssellänge effektiv halbiert.
- Man rechnet damit, dass Quantencomputer (sobald entsprechend leistungsfähige verfügbar sind) die Schlüssellänge *symmetrischer* Verfahren effektiv halbieren werden. Die bekannten *asymmetrischen Verfahren*, z.B. RSA oder DLP Verfahren, werden gegen ausreichend starke Quantencomputer nicht mehr sicher sein und man muss auf einen quantensicheren Algorithmus wechseln.

2.3 Monoalphabetische Verschlüsselung

Wir werden uns zum Aufwärmen die sogenannten **klassischen Verschlüsselungsalgorithmen** genauer ansehen. Das sind symmetrische Verfahren, die bis ca. 1950 entwickelt wurden (also bevor Computer verfügbar waren). Sie sind für das Verständnis der modernen Verfahren hilfreich. Wir konzentrieren uns dabei auf sogenannte **Substitutionsverfahren**: Das sind Algorithmen, bei denen jedes Klartextzeichen durch ein Geheimtextzeichen ersetzt wird. Die Reihenfolge der Buchstaben bleibt dabei unverändert.

Was ist zum Beispiel *kein* Subsititutionsverfahren? – Zum Beispiel, wenn der Geheimtext aus dem Klartext entsteht, indem die Buchstaben jedes Wortes in umgekehrter Reihenfolge geschrieben werden. Ein Verfahren, bei dem wie hier nur die *Reihenfolge* der Klartextbuchstaben verändert wird, heißt **Transpositionsverfahren**.

Ein Substitutionsverfahren wird **monoalphabetisch** genannt, wenn ein Klartextzeichen bei jedem Auftreten durch dasselbe Zeichen aus dem Geheimtextalphabet ersetzt wird; wenn also zum Beispiel A immer als M verschlüsselt wird. Ist das nicht der Fall, wird also z.B. A beim ersten Mal als M und beim nächsten Auftreten

als R verschlüsselt, so heißt das Substitutionsverfahren **polyalphabetisch**. Wir werden polyalphabetische Verfahren etwas später besprechen.

Eine monoalphabetische Verschlüsselung ist mathmematisch gesehen eine bijektive Abbildung vom Klartextalphabet in das Geheimtextalphabet. Gehen wir nun davon aus, dass Klartext- und Geheimtextalphabet gleich sind:

$$\begin{aligned} E \;&:\; A \to A \\ &\quad x \mapsto y = E(x) \end{aligned}$$

Eine Wertetabelle der Verschlüsselungsvorschrift könnte dann zum Beispiel so aussehen:

Klartextbuchstabe x:	A	B	C	D	E	F	G	...	W	X	Y	Z
Geheimtextbuchstabe y:	Q	W	E	R	T	Y	U	...	V	B	N	M

Also jedes A im Klartext wird zu einem Q im Geheimtext, jedes B zu einem W, etc.

Jede monoalphabetische Verschlüsselung entspricht einer Permutation (Änderung der Reihenfolge) der Zeichen des Alphabets (wir haben ja vorausgesetzt, dass Klartextalphabet und Geheimtextalphabet übereinstimmen). Bei n Buchstaben gibt es $n!$ Permutationen der Zeichen, also $n!$ Möglichkeiten für eine monoalphabetische Verschlüsselung (die identische Abbildung des Alphabets auf sich selbst wird zwar niemand zur Verschlüsselung verwenden, sie ist hier aber mitgezählt). Für $n = 26$ Zeichen sind das zum Beispiel $26! \approx 4{\cdot}10^{26}$ monoalphabetische Chiffren. Trotz dieser großen Zahl sind monoalphabetische Chiffren für natürliche Sprachen (also Deutsch, Englisch, ...) relativ leicht zu knacken. Natürliche Sprachen weisen nämlich immer eine bestimmte *Häufigkeitsverteilung* der Buchstaben im Klartext auf, die sich auf den Geheimtext überträgt.

Bereits Julius Cäsar hat für vertrauliche Nachrichten eine monoalphabetische Verschlüsselung verwendet: Er hat das Alphabet zyklisch (d.h., nach Z wird wieder mit A begonnen) um drei Stellen verschoben:

Klartextbuchstabe x:	A	B	C	D	E	F	G	...	W	X	Y	Z
Geheimtextbuchstabe y:	D	E	F	G	H	I	J	...	Z	A	B	C

Zur Verschlüsselung wird jeder Buchstabe durch den darunterliegenden, zur Entschlüsselung durch den darüberliegenden ersetzt. Allgemein nennt man einen Algorithmus, bei dem ein Buchstabe durch den um e Stellen weiter liegenden Buchstaben ersetzt wird, eine **Verschiebechiffre** oder **Cäsarverschiebung**. Dabei kann e als der Schlüssel aufgefasst werden. Für die mathematische Beschreibung wird zunächst jeder Buchstabe des Alphabets durch die entsprechende Ziffer in $\mathbb{Z}_{26}$ codiert (A $= 0$, B $= 1$, ..., Z $= 25$). Dann lautet die Verschlüsselungsvorschrift:

$$y = E_e(x) = (x + e) \bmod 26$$

Entschlüsselt wird, indem wir in die entgegengesetzte Richtung verschieben

$$x = D_e(y) = (y - e) \bmod 26.$$

Da das additive Inverse von e in $\mathbb{Z}_{26}$ gleich $d = 26 - e$ ist, können wir alternativ auch um d in dieselbe Richtungverschieben:

$$x = E_d(y) = (y + d) \bmod 26.$$

Beispiel 2.3 (→CAS) Cäsarverschiebung
Verschlüsseln Sie `DIESPINNENDIEGALLIER` mittels einer Verschiebechiffre um $e = 3$ Stellen unter der Verwendung modularer Arithmetik.

Lösung zu 2.3 Dazu codieren wir zunächst den Klartext in Zahlen aus $\mathbb{Z}_{26}$: `D` $= 3$, `I` $= 8$, usw., letztendlich:

$$\{3, 8, 4, 18, 15, 8, 13, 13, 4, 13, 3, 8, 4, 6, 0, 11, 11, 8, 4, 17\}$$

Die Verschlüsselung ist nun einfach. Für jede Klartextzahl x ist $y = (x+3) \bmod 26$ die zugehörige Geheimtextzahl:

$$\{6, 11, 7, 21, 18, 11, 16, 16, 7, 16, 6, 11, 7, 9, 3, 14, 14, 11, 7, 20\}$$

Das wollen wir wieder in Buchstaben sehen:

`GLHVSLQQHQGLHJDOOLHU`

Beachten Sie, dass – wie bei jeder monoalphabetischen Verschlüsselung – das mehrfache Auftreten eines Buchstaben im Klartext sich auf den Geheimtext überträgt: `NN` im Klartext wird z.B. zu `QQ` im Geheimtext. ■

Sie kennen vielleicht aus dem Internet die Verschiebechiffre **ROT13**, die einen Buchstaben aus dem Alphabet $\{a, \ldots, z\}$ durch den um genau $e = 13$ Stellen verschobenen Buchstaben ersetzt. Diese einfache Verschlüsselung ist nicht zum Schutz von Daten gedacht, sondern soll vor einem versehentlichen Mitlesen schützen – so wie eine Zeitung die Auflösung eines Rätsels oft kopfstehend druckt. Warum wird gerade um 13 Stellen verschoben? – Weil dann mit demselben Schlüssel wieder entschlüsselt werden kann: $d = 26 - e = 13$.

Wechseln wir nun die Seiten und machen uns, als Angreifer, an die Kryptanalyse. Gehen wir zunächst davon aus, dass wir einen Geheimtext vor uns haben, von dem wir wissen, dass er durch Verschiebe-Verschlüsselung eines deutschen Textes entstand, zum Beispiel `INJXJWYJCYNXYSNHMYRJMWLJMJNR`.

Es gibt 26 mögliche Verschiebe-Verschlüsselungen. Eine erste Möglichkeit wäre, alle Schlüssel e durchzuprobieren (Brute-Force), und zu prüfen, für welche Verschiebung sich aus dem Geheimtext ein sinnvoller Klartext ergibt. Das ist für einen Computer nicht aufwendig, aber es geht noch einfacher. Die Buchstaben eines deutschen Textes treten ja mit bestimmten Häufigkeiten auf:

Mittlere Buchstabenhäufigkeiten (Prozent) der deutschen Sprache

a	6.51	b	1.89	c	3.06	d	5.08	e	17.40	f	1.66
g	3.01	h	4.76	i	7.55	j	0.27	k	1.21	l	3.33
m	2.53	n	9.78	o	2.51	p	0.79	q	0.02	r	7.00
s	7.27	t	6.15	u	4.35	v	0.67	w	1.89	x	0.03
y	0.04	z	1.13								

So ist also zum Beispiel `E` im Mittel der häufigste Buchstabe in einem deutschen Text. Daraus können wir schließen, dass in einem Geheimtext der Buchstabe, der dem Klartextbuchstaben `E` entspricht, am häufigsten vorkommen sollte. Diese statistische Methode ist natürlich umso zuverlässiger, je länger der Geheimtext ist, den man untersucht.

Beispiel 2.4 (→CAS) Kryptanalyse einer Verschiebechiffre
Sie erhalten den Geheimtext `INJXJWYJCYNXYSNHMYRJMWLJMJNR`, von dem Sie wissen, dass es sich um einen deutschen Text handelt, der durch eine Verschiebechiffre $y = (x + e) \bmod 26$ verschlüsselt wurde. Entschlüsseln Sie den Geheimtext.

Lösung zu 2.4 Wenn wir bei einer Verschiebechiffre die Verschlüsselung eines *einzigen* Buchstaben kennen, so kennen wir bereits die Verschiebung e. Der häufigste Buchstabe des Geheimtexts ist `J`. Das legt nahe, dass er dem Klartextbuchstaben `E` entspricht. (Dieser Geheimtext ist nicht sonderlich lang, wir könnten also mit dieser Annahme leicht danebenliegen.) In diesem Fall wäre die Verschiebung $e = 5$, die Verschlüsselungsvorschrift also $y = (x + 5) \bmod 26$, und die Entschlüsselungsvorschrift $x = (y + 21) \bmod 26$. Damit ergibt sich: `DIESERTEXTISTNICHTMEHRGEHEIM`. Da wir einen sinnvollen Text erhalten, ist der Code geknackt. Hätten wir keinen sinnvollen Text erhalten, so hätten wir als nächstes die Zuordnung Geheimtext-`J` ist Klartext-`N` (das ist der Buchstabe, der in einem deutschen Text im Mittel am zweithäufigsten auftritt) probieren können. ■

Das war nicht schwer, denn wir wussten, dass es eine Verschiebechiffre ist. Wenn man nicht weiß, *welcher* monoalphabetische Verschlüsselungsalgorithmus verwendet wurde, dann muss man schon etwas mehr Aufwand in Kauf nehmen (es gibt 26! Möglichkeiten). Man muss dann für jeden Klartextbuchstaben den zugehörigen Geheimtextbuchstaben finden. Wieder hilft Statistik: Die häufigsten Buchstaben des Geheimtextes werden gezählt, und mithilfe der bekannten mittleren Häufigkeiten im deutschen Text kann eine Zuordnung versucht werden. Ein Problem dabei ist, dass in der Regel die gezählten Häufigkeiten von einzelnen Zeichen im Geheimtext nahe beieinander liegen werden, sodass eine eindeutige Zuordnung von Klartext- zu Geheimtextbuchstaben zunächst nicht möglich ist. Man kann aber zusätzlich die Häufigkeiten von **Buchstabenpaaren** (sogenannten **Bigrammen**) zu Hilfe nehmen:

Mittlere Bigrammhäufigkeiten (Prozent) der deutschen Sprache

en	3.88	er	3.75	ch	2.75	te	2.26	de	2.00	nd	1.99
ei	1.88	ie	1.79	in	1.67	es	1.52				

In deutschen Texten kommen also zum Beispiel die Bigramme `EN` oder `CH` besonders häufig vor. Beispiel: Wenn die Buchstaben `G` und `R` im Geheimtext am häufigsten vorkommen, so kann man vermuten, dass der eine `E` und der andere

N bedeutet. Wenn diese Häufigkeiten sehr nahe beieinander liegen, kann man die richtige Zuordnung finden, indem man zusätzlich die Häufigkeiten von GR und RG im Geheimtext ermittelt. Das häufigere Bigramm wird dann laut obiger Bigramm-Tabelle EN entsprechen.

Ein Brute-Force-Angriff, der das Entschlüsselungsverfahren aufgrund einer hohen Trefferquote weitestgehend automatisiert, ist das folgende Verfahren: Man zählt die Buchstabenhäufigkeiten des Geheimtexts aus und sucht jene Permutation F_e, für die die Summe der quadratischen Abweichungen zu den Buchstabenhäufigkeiten der deutschen Sprache minimal wird:

$$\sum_{x\in A} \left(h_D(x) - h_G(F_e(x))\right)^2 \text{ minimal}$$

Hier ein Beispiel, das die Idee verdeutlichen soll: Angenommen, das Klartext- und Geheimtextalphabet besteht einfachheitshalber nur aus den 4 Buchstaben A, B, C, D und in einem typischen Klartext sind die relativen Häufigkeiten von A, B, C, D gleich $50, 30, 15$ bzw. 5%. Nehmen wir weiters an, dass die Auszählung der Buchstaben im Geheimtext die Häufigkeiten $14, 52, 4, 30$ ergeben hat. Wenn der Computer nun alle $4! = 24$ möglichen Zuordnungen F_e durchprobiert, so wird sich herausstellen, dass für die Zuordnung

$$\begin{aligned}
\mathtt{A} &\mapsto F_e(\mathtt{A}) = \mathtt{B}\\
\mathtt{B} &\mapsto F_e(\mathtt{B}) = \mathtt{D}\\
\mathtt{C} &\mapsto F_e(\mathtt{C}) = \mathtt{A}\\
\mathtt{D} &\mapsto F_e(\mathtt{D}) = \mathtt{C}
\end{aligned}$$

die Summe der (quadratischen) Abweichungen der entsprechenden Häufigkeiten,

$$(50-52)^2 + (30-30)^2 + (15-14)^2 + (5-4)^2,$$

minimal ist. (Dass das höchstwahrscheinlich die richtige Zuordnung ist, sehen wir hier auch durch „Hinsehen“.)

Diese Strategie funktioniert auch bei Sprachen, bei denen es keinen eindeutig häufigsten Buchstaben gibt, und auch bei relativ kurzen Texten.

Beispiel 2.5 (→CAS) Minimale quadratische Abweichung der Buchstabenhäufigkeiten
Entschlüsseln Sie den Geheimtext

```
PUQMZFIADFMGRPUQRDMSQZMOTPQYXQNQZPQYGZU
HQDEGYGZPPQYSMZLQZDQEFUEFLIQUGZPHUQDLUS
```

von dem Sie wissen, dass es sich um einen deutschen Text handelt, der mit einer Verschiebechiffre verschlüsselt wurde.

Lösung zu 2.5 Mithilfe eines Computerprogramms finden wir die minimale Abweichung bei $e = 12$. Versuchen wir, mit dem zugehörigen $d = 26-12 = 14 \pmod{26}$ zu entschlüsseln, so erhalten wir

```
DIEANTWORTAUFDIEFRAGENACHDEMLEBENDEMUNIVERSUMUNDDEMGANZEN
RESTISTZWEIUNDVIERZIG
```

Da wir einen *sinnvollen* Text erhalten, wurde der Text also tatsächlich um 12 Stellen verschoben. Wäre der Text sinnlos gewesen, hätten wir als nächstes die Verschiebung mit der zweitkleinsten quadratischen Abweichung probieren können. ■

Die Buchstabenhäufigkeiten kann man auch dazu verwenden, um zu erkennen, ob ein Text monoalphabetisch oder polyalphabetisch verschlüsselt wurde: Bei einer monoalphabetischen Verschlüsselung werden die Buchstaben ja nur umbenannt und somit bleiben die Buchstabenhäufigkeiten im Geheimtext erhalten. Somit kann man die (am besten geordneten) Buchstabenhäufigkeiten im Geheimtext mit den Buchstabenhäufigkeiten der vermuteten Sprache vergleichen. Das ist in Abbildung 2.1 zu sehen. Der linke Text wurde monoalphabetisch verschlüsselt und der rechte Text mit einer polyalphabetischen Verschlüsselung, wie wir sie im nächsten Abschnitt besprechen werden. Bei einer polyalphabetischen Verschlüsselung werden die Buchstabenhäufigkeiten des Klartextes besser verschleiert und sind dadurch im Geheimtext weniger ausgeprägt. Das ist in Abbildung 2.1 zum Beispiel am deutlich kleineren Wert für den häufigsten Buchstaben zu sehen.

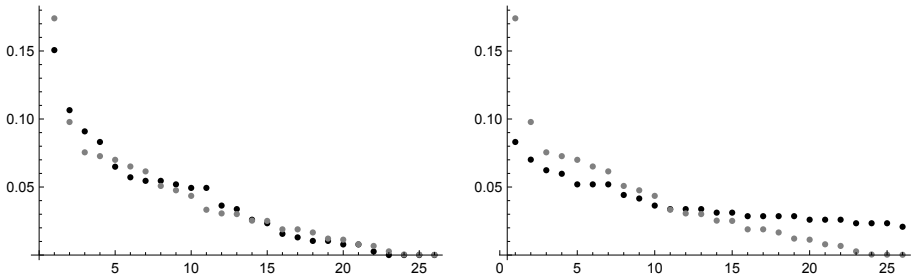


Abbildung 2.1: Vergleich der Buchstabenhäufigkeiten bei einer monoalphabetischen Verschlüsselung (links) und einer polyalphabetischen Verschlüsselung (rechts). Zusätzlich sind jeweils grau die deutschen Buchstabenhäufigkeiten eingezeichnet.

Quantitativ kann man die Abweichung vom Fall, dass alle Buchstaben gleich wahrscheinlich sind, mit Hilfe des **Koinzidenzindex** K (auch **Friedman-Charakteristik** nach dem amerikanischen Kryptologen William F. Friedman (1891–1969)) beschreiben. Der Koinzidenzindex ist gegeben durch

$$K = \frac{n}{n-1} \sum_{i=0}^{n-1} \left(h_i - \frac{1}{n} \right)^2 = \frac{n}{n-1} \left(\sum_{i=0}^{n-1} h_i^2 - \frac{1}{n} \right),$$

wobei h_i die relative Häufigkeit des i-ten Buchstaben im untersuchten Text ist (also absolute Häufigkeit des i-ten Buchstaben im Text geteilt durch die Länge

des Texts) und n die Anzahl der Buchstaben im Alphabet. Der Koinzidenzindex gibt also die quadratische Abweichung von einer Gleichverteilung (bei der jeder der n Buchstaben mit Häufigkeit $\frac{1}{n}$ auftritt) an und der Normierungsfaktor ist so gewählt, dass $0 \leq K \leq 1$ gilt. Er ist umso größer, je ausgeprägter die Häufigkeiten sind und wird maximal, wenn der Text nur einen Buchstaben enthält.

Warum? – Die zweite Darstellung von K erhält man, indem man das Quadrat in der ersten Darstellung ausmultipliziert und $\sum_{i=0}^{n-1} h_i = 1$ und $\sum_{i=0}^{n-1} 1 = n$ verwendet. Aus der ersten Darstellung sieht man $K \geq 0$; aus der zweiten Darstellung folgt $K \leq 1$, da $h_i^2 \leq h_i$ (beachte $0 \leq h_i \leq 1$) und daher $\sum_{i=0}^{n-1} h_i^2 \leq \sum_{i=0}^{n-1} h_i = 1$ gilt, mit Gleichheit genau dann, wenn $h_i^2 = h_i$ für alle i gilt. Letzteres impliziert $h_i \in \{0, 1\}$ und das kann nur eintreten, wenn der Text aus genau einem Buchstaben besteht.

Mit den Buchstabenhäufigkeiten der deutschen Sprache erhalten wir den Koinzidenzindex für einen typischen deutschen Text:

$$K_d = 0.036.$$

Für den linken Text aus Abbildung 2.1 erhält man $K = 0.035$ und für den rechten $K = 0.006$.

Diese statistischen Methoden sind auch für jede andere natürliche Sprache anwendbar, bzw. allgemein für jeden Klartext, bei dem zu erwarten ist, dass die Klartextzeichen im Mittel mit bestimmten Häufigkeiten auftreten. Bei Anwendungen im Computerbereich wird man sich zum Beispiel nicht auf die Buchstaben `A` bis `Z` beschränken. Da alle Daten üblicherweise als eine Folge von Bytes vorliegen, bietet es sich an, die Zahlen von 0 bis 255, also $\mathbb{Z}_{256}$, als Alphabet zu nehmen. Solange die Verteilung dieser Klartextzeichen (also der Bytes) nicht zufällig ist, lassen sich die oben beschriebenen statistischen Methoden zur Kryptoanalyse anwenden. Die zu erwartenden mittleren Häufigkeiten der einzelnen Klartextzeichen werden ermittelt, indem ein repräsentativer Musterklartext ausgezählt wird. Bei einem C-Programm werden zum Beispiel die geschwungenen Klammern „{“ und „}“ besonders häufig auftreten.

Da vielen Softwareentwicklern bitweise logische Operationen vertrauter sind als modulare Arithmetik, wird oft das logische xor zur monoalphabetischen Verschlüsselung verwendet. Das Klartextzeichen $x \in \mathbb{Z}_{256}$ wird dabei mithilfe des Schlüssels $e \in \mathbb{Z}_{256}$ nach der Vorschrift

$$y = x \oplus e$$

verschlüsselt. Entschlüsselt wird mit $x = y \oplus e$.

Beispiel 2.6 (→CAS) Monoalphabetische Verschlüsselung mit xor
Verschlüsseln Sie den Klartext

Streng geheim!

mittels xor mit dem Schlüssel $e = 127$.

Lösung zu 2.6 Der Klartext lautet im ASCII-Code: $83, 116, 114,\ 101, 110, 103,\ 32, 103, 101,\ 104, 101, 105, 109, 33$. Zur Verschlüsselung müssen wir alle Zahlen ins Dualsystem umwandeln und dann bitweise das logische xor bilden: $x = 83 = (1010011)_2$ und $e = 127 = (1111111)_2$ ergibt $y = 83 \oplus 127 = (0101100)_2 = 44$ (wegen $0 \oplus 0 = 1 \oplus 1 = 0$ und $1 \oplus 0 = 0 \oplus 1 = 1$). Analog für die weiteren Buchstaben, sodass der Geheimtext im ASCII-Code lautet: $44, 11, 13,\ 26, 17, 24,\ 95, 24, 26,\ 23, 26,\ 22, 18, 94$. Eine Umwandlung des Geheimtexts in Buchstaben ist hier in der Regel nicht mehr möglich, weil ja nicht jedes mögliche Byte im ASCII-Code einem darstellbaren Zeichen entspricht. ■

Wie bei den Verschiebechiffren genügt bei diesem monoalphabetischen Verfahren die Zuordnung eines einzigen Geheimtextzeichens zum zugehörigen Klartextzeichen, um den Schlüssel zu bestimmen und damit die Verschlüsselung zu knacken.

Zusammenfassend kann man sagen, dass bei einem kleinen Alphabet monoalphabetische Verschlüsselungen eines Klartextes, bei dem die einzelnen Klartextzeichen mit bestimmten Häufigkeiten auftreten, mit statistischen Methoden gebrochen werden können, und daher keinerlei Sicherheit bieten.

2.4 Polyalphabetische Verschlüsselung

Bei einer polyalphabetischen Verschlüsselung wird ein Klartextzeichen nicht stets zu demselben Geheimtextzeichen verschlüsselt. Dadurch übertragen sich die Häufigkeiten der Klartextbuchstaben nicht auf den Geheimtext, und dadurch wird einem Angreifer die Arbeit erschwert.

Ein bekanntes polyalphabetisches Verfahren ist die **Vigenère-Verschlüsselung**. Dabei werden mehrere Verschiebechiffren periodisch verwendet. Die Verschiebung eines Klartextzeichens hängt dabei von dessen Position im Klartext ab.

Die Ursprünge der Vigenère-Verschlüsselung gehen zurück ins 15. Jahrhundert. Da eine einfache monoalphabetische Verschlüsselung zu unsicher war, war ging man dazu über, verschiedene Verschiebungen für verschiedene Buchstaben zu verwenden. Dazu gab es die **Tabula recta** (lat. für Quadratische Tafel), die vom Benediktinerabt Johannes Trithemius (1462–1516) in seinem sechsbändigen Werk *Polygraphiae Libri Sex* veröffentlicht wurde. Mithilfe dieser Tafel waren die Verschiebungen leicht ablesbar. Er schlug auch vor, bei jedem Buchstaben zur nächsten Verschiebung zu wechseln. Da das Verfahren fest (also ohne geheimen Schlüssel) war, konnte jeder, der das Verfahren kannte, entschlüsseln. Im Jahr 1553 wurde vom italienischen Kryptologen Giovan Battista Bellaso (ca. 1505–1568/81) vorgeschlagen, die Reihenfolge der Verschiebungen durch ein frei zu wählendes Kennwort festzulegen. Unabhängig, und vor Trithemius, schlug der italienische Gelehrte Leon Battista Alberti (1404–1472) vor, *verwürfelte Alphabete* (also statt einer Verschiebung eine allgemeine Permutation der Buchstaben) zu verwenden. Er empfahl auch, die Alphabete nach drei bis vier Worten zu wechseln. Der neapolitanische Gelehrte Giovanni Battista della Porta (1535–1615) kombinierte schlussendlich beide Ideen. Das wurde 1585 vom französischen Diplomaten Blaise de Vigenère (1523–1596) aufgegriffen, wobei er vorschlug, zur Verschlüsselung die Tabula recta zu verwenden, diese aber nicht mit verschobenen, sondern mit verwürfelten Alphabeten zu befüllen. Sein Vorschlag geriet in Vergessenheit und hat sich später, allerdings nun wie ursprünglich mit verschobenen Alphabeten, verbreitet.

Betrachten wir gleich ein Beispiel zur Vigenère-Verschlüsselung: Wir verwenden als Schlüssel das Wort `VIGENERE`. Es wird periodisch fortgesetzt über den Klartext geschrieben. Jeder Buchstabe im Klartext wird nun mit einer anderen Verschiebechiffre verschlüsselt, und zwar mit jener, die zum darüberstehenden Buchstaben des Schlüsselwortes gehört:

Schlüsselwort	`VIGENEREVIGENEREVIG`
Klartext	`DIESERTEXTISTGEHEIM`
Geheimtext	`YQKWRVKISBOWGKVLZQS`

Der erste Buchstabe, `D`, wird also um `V`=21 Stellen auf `Y` verschoben. Der zweite Buchstabe `I` wird um `I`=8 Stellen auf `Q` verschoben, etc.

Das Verfahren kann mathematisch folgendermaßen beschrieben bzw. implementiert werden: Nachdem wir die Buchstaben in Zahlen umgewandelt haben, gilt $y_i = (x_i + e_i) \bmod 26$. Der Unterschied zur Verschiebechiffre ist also, dass jetzt für jeden Buchstaben x_i eine eigene Verschiebung e_i verwendet wird. Ist k die Schlüssellänge, und beginnen wir bei 0 zu zählen, so ist für den Klartextbuchstaben x_k wieder e_0 zu verwenden. Wir erhalten somit die **Vigenère-Verschlüsselungsvorschrift**

$$y_i = (x_i + e_{i \bmod k}) \bmod 26.$$

Entschlüsselt wird wieder mit diesem Vigenère-Algorithmus, nur mit einem anderen Schlüsselwort:

$$x_i = (y_i + d_{i \bmod k}) \bmod 26,$$

wobei $d_i = -e_i \pmod{26}$ gilt.

Beispiel 2.7 (→CAS) Vigenère-Verschlüsselung
Verschlüsseln Sie `DIESERTEXTISTGEHEIM` mit dem Schlüsselwort `VIGENERE` und entschlüsseln Sie danach wieder.

Lösung zu 2.7 Der erste Klartextbuchstabe $x_0 = 3$ wird mit $e_0 = 21$ zu $y_0 = 24$ verschlüsselt, was dem Buchstaben `Y` entspricht. Analog wird $x_1 = 8$ mit $e_1 = 8$ zu $y_1 = 16$ verschlüsselt, also zu `Q`. Insgesamt ergibt sich:

`YQKWRVKISBOWGKVLZQS`

Für die Entschlüsselung berechnen wir zunächst das Schlüsselwort zum Entschlüsseln: $d_0 = 26 - e_0 = 26 - 21 = 5$, also Buchstabe `F`. Analog $d_1 = 26 - e_1 = 26 - 8 = 18$, also Buchstabe `S` u.s.w. Insgesamt ergibt sich `FSUWNWJW` und damit erhält man auch wieder den ursprünglichen Klartext. ■

Nun zur **Kryptanalyse der Vigenère-Verschlüsselung**: Wenn für den Schlüssel bis zu n Stellen zur Verfügung stehen, so ergibt das bei einem 26-elementigen Alphabet $26 + 26^2 + \ldots, +26^n = 26\frac{26^n - 1}{25} \approx 26^n$ mögliche Schlüssel. Für genügend

große Schlüssellänge n wird es daher auch mithilfe von Computern nicht mehr möglich sein, alle Schlüsselwörter *durchzuprobieren*.

Wie sieht es mit einer statistischen Analyse aus? Ein bestimmter Klartextbuchstabe wird nun nicht stets auf den gleichen Geheimtextbuchstaben abgebildet. Zum Beispiel wird das erste `E` im obigen Beispiel auf `K`, das zweite `E` aber auf `R` abgebildet. Es sieht also auf den ersten Blick so aus, als ob tatsächlich ein brauchbarer Algorithmus gefunden wäre, der nicht so leicht geknackt werden kann. Und so dauerte es auch in der Tat einige Zeit, bis im Jahr 1863 der preußische Infanteriemajor und Kryptograph Friedrich Wilhelm Kasiski (1805–1881) eine Methode zur Entschlüsselung veröffentlichte (die Methode war zuvor auch schon dem englischen Mathematiker Charles Babbage (1791–1871) bekannt, der sie aber geheim gehalten hatte). Unsere Kryptoanalyse hier basiert auf späteren Arbeiten von W. Friedman. Wiederum hilft die Häufigkeitsverteilung des Klartexts.

Überlegen wir zunächst, dass alles, was ein Angreifer braucht, die Schlüsselwortlänge k ist. Kennt man diese, dann ist die Kryptanalyse auf die einer gewöhnlichen Verschiebeverschlüsselung reduziert, wobei k Geheimtexte separat analysiert werden müssen. Warum? Betrachten wir nochmals das Beispiel mit dem Schlüsselwort `VIGENERE`. Das Schlüsselwort hat die Länge $k = 8$. Das heißt, dass das 1., das 9., das 17. usw. Klartextzeichen mit derselben Verschiebung verschlüsselt wird:

DIESERTE**X**TISTGEH**E**IM

Man braucht also nur den Geheimtext in diese, den $k = 8$ Verschiebungen entsprechenden, Teiltexte zu zerlegen (Teiltext 1 wäre also in diesem Beispiel der 1., 9., und 17. Buchstabe des Geheimtexts, also `YSZ`, Teiltext 2 wäre der 2., 10., und 18. Buchstabe des Geheimtexts, also `QBQ`, etc.). Jeder dieser Teiltexte ist monoalphabetisch mit einer einfachen Verschiebung verschlüsselt, und kann daher mit den im letzten Abschnitt besprochenen Verfahren entschlüsselt werden.

Wie finden wir aber nun die Länge des Schlüsselwortes? Das können wir wieder mit dem Koinzidenzindex bewerkstelligen. Dazu zerlegen wir den Text wie eben beschrieben, wobei wir alle möglichen Werte für k durchprobieren. Liegt der Koinzidenzindex aller Teile nahe am Wert eines deutschen Texts, so ist das ein Hinweis, dass die Teile monoalphabetisch verschlüsselt wurden. So können wir erkennen, ob wir das richtige k gefunden haben.

Im Prinzip kann dafür jedes Teilstück verwendet werden. Bei kurzen Klartexten werden die Teilstücke mit wachsendem k aber schnell kürzer. In jedem Fall bietet es sich an, den Mittelwert über alle Teilstücke zu verwenden, um das Ergebnis zu verbessern.

Beispiel 2.8 (→CAS) Kryptoanalyse einer Vigenère-Verschlüsselung
Entschlüsseln Sie den Text

```
LWAGZRUHUWVLMVRRWTOLFVZIAOQYVZTLQNHMEQMKAEIIPTAVMZFSLVU
YYMFSYENDRKVIZGDIICOEXUXMAJVRJHLQKWXEZGYGUJLJLQPGWTOALP
```

```
HPVNBLLUDCHORBRRMTOIEYVPTRQVLLSKXUKERHXLLBVUKMMRQAYUTVE
YYMFSYENHZHYFMVUIISOIHYETFNFRSDQXLVAGCIEOECPBRLQZREKEEK
HPBJNRRJHMHYJWYFIEKSKXYWMVMDZLLVUWECFDRCVSSLZYVGDRWPECX
UHPVDVMLRCRYIZQTVTOSVLUWBEDFVUACVSIBMSIJUUEDBGPANGYKAJO
LFMAZRULMVUNEVMAIJVNUHYWPVMKULRSOLGPFBLRKEEZHVVHQKVYIEJ
```

der durch Vigenère-Verschlüsselung eines deutschen Klartextes entstand.

Lösung zu 2.8 Als erstes bestimmen wir die Schlüssellänge. Dazu zerlegen wir den Text wie oben beschrieben in $k = 1, 2, 3, \ldots$ Teiltexte: Für $k = 1$ ist es der gesamte Geheimtext; für $k = 2$ erhalten wir die beiden Teiltexte WGRHW… (bestehend aus dem 1., 3., 5., usw. Buchstaben) und AZUU… (bestehend aus dem 2., 4., 6., usw. Buchstaben); und so weiter. Wir berechnen zu jedem k den mittleren Koinzidenzindex K der Teiltexte. Lassen wir uns K für $k = 1, 2, \ldots, 35$ vom Computer berechnen und stellen das Ergebnis graphisch dar (siehe Abbildung 2.2), so sehen wir, dass die Werte für $k = 11, 22, 33$ deutlich näher bei dem für einen deutschen Text zu erwartenden Wert von 0.036 liegen als alle anderen Werte. Das ist ein starkes Indiz für eine Schlüssellänge von 11. Der Anstieg ergibt sich, da die einzelnen Texte immer kürzer werden (ab 16 Teiltexten gibt es z.B. schon weniger als 26 Buchstaben pro Teiltext und immer mehr Häufigkeiten müssen daher Null sein, was den Wert nach oben treibt).

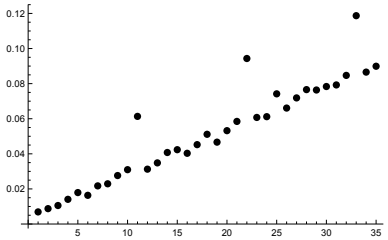


Abbildung 2.2: Für $k = 11, 22, 33$ wurden die Teilstücke vermutlich monoalphabetisch verschlüsselt.

Schauen wir nun, ob wir mit der Schlüssellänge 11 richtig liegen. Dazu müssen wir den Geheimtext wie oben beschrieben in 11 Teiltexte zerlegen: Text 1 besteht also aus dem 1., dem 12., dem 23., usw. Buchstaben des Geheimtext; Text 2 aus dem 2., dem 13., usw. Buchstaben des Geheimtextes, usw. Jeder dieser elf Texte ist (falls 11 die Schlüssellänge ist) mit derselben Verschiebechiffre verschlüsselt. Wir entschlüsseln also jeden Teiltext mit der Kryptoanalyse für die Cäsarverschiebung und erhalten folgende Werte: $20, 4, 8, 13, 25, 4, 17, 7, 0, 17, 3$ Damit haben wir einen Kandidaten für das Schlüsselwort: `UEINZERHARD`. Das zugehörige Schlüsselwort zum Entschlüsseln wäre: `GWSNBWJTAJX`. Versuchen wir, damit zu entschlüsseln:

```
RSSTANDAUFSRINESSCHLOSFESBRUESTUNTDERRITTERFVPSINVOLLER
EUESTUNGDAHBERTEERVONUATENKRACHUNQSPRACHZUSIPHICHSCHAUM
```

```
NLNACHUNDLEUNTESICHINVBLLERRUESTUAGWEITUEBERQIEERWHNTEB
EUESTUNGHIEEBEIVERLOREEALSOBALDZURRSTDENHELMHNDDANNDENH
NLTWONACHVEEFOLGENDSTUESEINZIELERCAUSENLOSBIFUNTENFIELU
ADHIERVERLOEERDURCHSEIASTREBENALSYETZTESNUNAHCHNOCHDASL
RBENANDEMERTANZBESONDEESHINGDERBLRCHSCHADENWNRNURGERING
```

Das ist aber leider noch kein ganz sinnvoller Textt. Es scheint so, als ob der erste Buchstabe des Schlüsselwortes falsch ist. Damit der Klartext Sinn ergibt, sollte der erste Buchstabe (das kann man schon an den ersten Wörtern des Klartestes ablesen) im Klartext wohl ein `E` sein. Das würde dem Schlüsselwort `HEINZERHARD` entsprechen. Damit erhalten wir tatsächlich auch insgesamt einen sinnvollen Klartext[1]. Diese kleine *Panne* ist nicht verwunderlich, denn der Geheimtext besteht nur aus 385 Buchstaben, und daher besteht jeder der 11 Teile nur aus 35 Buchstaben, was für eine zuverlässige statistische Auswertung eben nicht ausreicht. ■

Es gibt noch weitere (bessere) Verfahren (z.B. den **Zeichenkoinzidenzindex**), um die Schlüssellänge zu bestimmen. Sie beruhen auf ähnlichen Überlegungen wie oben. Da hier nur die wesentliche Idee veranschaulicht werden sollte, gehen wir nicht weiter darauf ein.

Die ersten Verfahren zum Brechen der Vigenère-Chiffre wurden, wie schon erwähnt, in der Mitte des 19. Jahrhunderts publiziert. Doch trotz der schon so lange bekannten Schwächen wurden Vigenère-Chiffren (in verschiedenen Varianten) bis in die Gegenwart verwendet. In Microsoft Word 2.0 wurde zum Beispiel ein 16 Byte Schlüsselwort verwendet, das, analog wie bei der Vigenère-Chiffre, mit jedem Byte im Klartext mittels xor verknüpft wird. Die Kryptoanalyse mittels Koinzidenzindex ist also problemlos anwendbar. Sogar diesen Aufwand konnte sich aber ein Angreifer ersparen, denn es wurde auch der Dateiheader mitverschlüsselt, in dem sich an einer bestimmten Stelle immer 16 Nullbytes befinden. Daher konnte man dort das Schlüsselwort im Klartext ablesen (a xor $0 = a$)! In der Version Word 6.0 wurde deshalb ein Klartextbyte nur mit xor verknüpft, wenn es selbst oder das Ergebnis von Null verschieden ist. Da aber die Dokumentzusammenfassung sowohl verschlüsselt als auch unverschlüsselt in der Datei enthalten war, war es weiterhin möglich (auch ohne statistische Analyse), das Schlüsselwort zu finden (Known-Plaintext-Angriff). Dementsprechend gibt es im Internet auch eine Reihe von Programmen, die mit Word 6.0 verschlüsselte Dateien knacken. Mehr dazu finden Sie in [Wob01].

Wird das Alphabet $\mathbb{Z}_{256}$ verwendet, so ist die **xor-Vigenère-Verschlüsselung** möglich:

$$y_i = x_i \oplus e_{i \bmod k},$$

wobei $e_i \in \mathbb{Z}_{256}$ die Zeichen des Schlüssels der Länge k sind. Beispiel: Schlüssel `KEY` der Länge 3: Der erste Buchstabe des Klartextes wird dann mit `K`, dargestellt als Byte laut ASCII-Code, xor-verknüpft, der zweite mit `E`, der dritte mit `Y`, der vierte wieder mit `K`, usw. Auch die xor-Vigenère-Verschlüsselung kann mit den oben beschriebenen statistischen Methoden geknackt werden. Allgemein funktioniert der Angriff unabhängig davon, welche monoalphabetische Verschlüsselung

[1] Bei dem Text handelt es sich um das Gedicht „Ritter Fips“ von Heinz Erhardt. Mit freundlicher Genehmigung das Lappan Verlags, http://www.lappan.de

der auf diese Weise (also mit periodisch wiederholtem Schlüsselwort) erzeugten polyalphabetischen Verschlüsselung zugrunde liegt.

Eine andere klassische, polyalphabetische Verschlüsselung ist die **Hill-Chiffre**. Sie wurde erstmals im Jahr 1929 vom amerikanischen Mathematiker Lester S. Hill (1891–1961) veröffentlicht. Sie ist eine **Blockchiffre**, das bedeutet, dass mehrere Klartextzeichen („Blöcke“) auf einmal zu Geheimtextblöcken verschlüsselt werden. Um die Hill-Chiffre übersichtlich angeben zu können, verwendet man Matrixmultiplikation. Im einfachsten Fall ist die Blocklänge gleich zwei:

$$\begin{pmatrix} y_i \\ y_{i+1} \end{pmatrix} = \begin{pmatrix} a & b \\ c & d \end{pmatrix} \begin{pmatrix} x_i \\ x_{i+1} \end{pmatrix}, \text{ für } i = 1, 3, 5, \ldots$$

Die ersten beiden Klartextzeichen x_1, x_2 werden also zu y_1, y_2 verschlüsselt, der nächste Klartextblock, x_3, x_4 wird zu y_3, y_4 verschlüsselt usw. Klartext- und Geheimtextzeichen sind aus dem Alphabet $\mathbb{Z}_n$ ($n \in \mathbb{N}$). Der geheime Schlüssel besteht aus vier Zahlen, $a, b, c, d \in \mathbb{Z}_n$. Diese Zahlen müssen die Bedingung $\mathrm{ggT}(ad - bc, n) = 1$ erfüllen, da nur unter dieser Bedingung der Kehrwert von $ad - bc$ in $\mathbb{Z}_n$ existiert, der für die Entschlüsselungsvorschrift benötigt wird:

$$\begin{pmatrix} x_i \\ x_{i+1} \end{pmatrix} = (ad - bc)^{-1} \begin{pmatrix} d & -b \\ -c & a \end{pmatrix} \begin{pmatrix} y_i \\ y_{i+1} \end{pmatrix}, \text{ für } i = 1, 3, 5, \ldots$$

Sehen wir uns ein einfaches Beispiel an:

Beispiel 2.9 Hill Chiffre
Das Alphabet sei $\mathbb{Z}_{11}$. Alice und Bob einigen sich auf den Schlüssel $a = 4, b = 2, c = 6, d = 7$.
a) Welchen Geheimtext erhält Alice, wenn sie den Klartext $5, 1, 3, 3$ mit der Hill-Chiffre verschlüsselt?
b) Zeigen Sie, dass die Bedingung $\mathrm{ggT}(ad - bc, n) = 1$ erfüllt ist und geben Sie die Entschlüsselungsvorschrift von Bob an.

Lösung zu 2.9 Der erste Klartextblock, $5, 1$, wird verschlüsselt zu

$$\begin{pmatrix} y_1 \\ y_2 \end{pmatrix} = \begin{pmatrix} 4 & 2 \\ 6 & 7 \end{pmatrix} \begin{pmatrix} 5 \\ 1 \end{pmatrix} = \begin{pmatrix} 4 \cdot 5 + 2 \cdot 1 \\ 6 \cdot 5 + 7 \cdot 1 \end{pmatrix} = \begin{pmatrix} 22 \\ 37 \end{pmatrix} = \begin{pmatrix} 0 \\ 4 \end{pmatrix}$$

und der zweite Klartextblock, $3, 3$, wird verschlüsselt zu

$$\begin{pmatrix} y_3 \\ y_4 \end{pmatrix} = \begin{pmatrix} 4 & 2 \\ 6 & 7 \end{pmatrix} \begin{pmatrix} 3 \\ 3 \end{pmatrix} = \begin{pmatrix} 4 \cdot 3 + 2 \cdot 3 \\ 6 \cdot 3 + 7 \cdot 3 \end{pmatrix} = \begin{pmatrix} 18 \\ 39 \end{pmatrix} = \begin{pmatrix} 7 \\ 6 \end{pmatrix}.$$

Der Geheimtext lautet also: $0, 4, 7, 6$. Wir sehen insbesondere, dass der Doppelbuchstabe $3, 3$ des Klartextes im Geheimtext nicht mehr als Doppelbuchstabe aufscheint.

b) $ad-bc = 4\cdot 7-2\cdot 6 = 16$; 16 und 11 sind teilerfremd, daher gibt es den Kehrwert von 16, also 16^{-1}, in $\mathbb{Z}_{11}$. Nach einer kurzen Rechnung finden wir $\frac{1}{16} = 9 \in \mathbb{Z}_{11}$. Daher ist die Entschlüsselungsvorschrift (wir rechnen modulo 11):

$$\begin{pmatrix} x_i \\ x_{i+1} \end{pmatrix} = 9 \cdot \begin{pmatrix} 7 & -2 \\ -6 & 4 \end{pmatrix} \begin{pmatrix} y_i \\ y_{i+1} \end{pmatrix} = \begin{pmatrix} 63 & -18 \\ -54 & 36 \end{pmatrix} \begin{pmatrix} y_i \\ y_{i+1} \end{pmatrix} = \begin{pmatrix} 8 & 4 \\ 1 & 3 \end{pmatrix} \begin{pmatrix} y_i \\ y_{i+1} \end{pmatrix}.$$

In der Tat erhält Bob z.B. aus dem ersten Geheimtextblock $0, 4$ den ersten Klartextblock:

$$\begin{pmatrix} 8 & 4 \\ 1 & 3 \end{pmatrix} \begin{pmatrix} 0 \\ 4 \end{pmatrix} = \begin{pmatrix} 16 \\ 12 \end{pmatrix} = \begin{pmatrix} 5 \\ 1 \end{pmatrix}.$$

■

2.5 One-Time Pad

Die Kryptoanalyse aus dem letzten Abschnitt hat gezeigt, dass auch eine Vigenère-Chiffre durch eine statistische Analyse gebrochen werden kann. In diesem Abschnitt besprechen wir das **One-Time-Pad**. Es handelt sich dabei um eine Vigenere-Verschlüsselung mit folgenden Eigenschaften:

- Der Schlüssel ist gleich lang wie der Klartext.
- Der Schlüssel darf nur einmal verwendet werden (d.h., für jeden Klartext ist ein neuer Schlüssel zu erzeugen).
- Der Schlüssel muss echt zufällig erzeugt werden.

Wird mit einem One-Time-Pad verschlüsselt, so enthält der Geheimtext (außer der Länge) keinerlei Informationen über den Klartext mehr. Ein solches Verschlüsselungsverfahren wird, wie schon im Abschnitt 2.2 besprochen, als **perfekt** oder **uneingeschränkt sicher** bezeichnet.

Ein entsprechendes Verfahren wurde zum ersten Mal 1882 vom amerikanischen Bankier Frank Miller (1842–1925) publiziert, blieb aber weitgehend unbeachtet. 35 Jahre später wurde es vom amerikanischen AT&T-Ingenieur Gilbert S. Vernam (1890–1960) neu entdeckt, der das Verfahren zusammen mit einem entsprechenden Gerät im Jahr 1917 patentierte. Gemeinsam mit seinem Kollegen, dem Offizier Joseph O. Mauborgne (1881–1971), erkannte er auch die theoretische Bedeutung des Verfahrens.

Warum das so ist, kann man sich anhand eines einfachen Beispiels veranschaulichen:

Stellen Sie sich vor, dass Sie als Angreifer den Geheimtext `QKEZFJRVHC` abgefangen haben. Sie wissen, dass mit einem One-Time-Pad verschlüsselt wurde. Da der Geheimtext 10 Zeichen lang ist, probieren Sie alle möglichen Schlüssel mit 10 Zeichen Länge durch und prüfen, ob sich ein sinnvoller Klartext ergibt. Dabei stoßen Sie unter anderem auf die Schlüssel `GWSNBWJTAJ` und `GWSNBCNBOY`. Welche Klartexte erhalten Sie für diese beiden Fälle?

Schlüsselwort	`GWSNBWJTAJ`
Geheimtext	`QKEZFJRVHC`
Klartext	`KOMMENICHT`

und

Schlüsselwort	`GWSNBCNBOY`
Geheimtext	`QKEZFJRVHC`
Klartext	`KOMMEHEUTE`

Welcher ist nun der richtige Klartext? Sie können hier sogar zu *jeder* vorgegebenen Klartextnachricht (mit Länge 10) einen Schlüssel finden, sodass die Geheimtextnachricht `QKEZFJRVHC` zu genau dieser Klartextnachricht entschlüsselt wird.

Warum ist das möglich? – Der Grund liegt darin, dass hier das **Schlüsselwort gleich lang ist wie der zu verschlüsselnde Klartext**. Aus der vorgegebenen Geheimtextnachricht erhalten Sie durch Anwendung der 26^{10} möglichen Schlüssel alle 26^{10} möglichen Klartextnachrichten (= alle möglichen Aneinanderreihungen von 10 Buchstaben).

Der Schlüssel darf nur einmal verwendet werden und man spricht daher auch von einem **Wegwerfschlüssel**. Das Problem wird in Abbildung 2.3 drastisch vor Augen geführt: Dabei wurden zwei Bilder mit dem gleichen Schlüssel verschlüsselt. Beide Chiffrate für sich wirken völlig zufällig und scheinen in der Tat keine Informationen über das jeweilige Originalbild preiszugeben. Bildet man aber die Differenz der Bilder, so kürzt sich der Schlüssel heraus, und die Differenz der Originalnachrichten wird sichtbar!

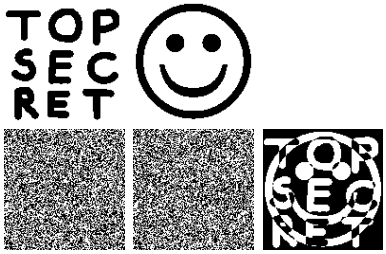


Abbildung 2.3: Mehrfachverwendung eines Schlüssels beim One-Time-Pad (nach [CS08]): Oben zwei Bilder und darunter jeweils das zugehörige Chiffrat. In der Differenz der Chiffrate (unten ganz rechts) kürzt sich der Schlüssel weg und die Differenz der ursprünglichen Bilder wird sichtbar.

Der Name „One Time Pad“ kommt von der Vorstellung, dass man die Schlüsselbuchstaben auf einen Abreißblock schreibt und das Blatt wegwirft, sobald ein Buchstabe verwendet worden ist. Die Mehrfachverwendung von Schlüsseln wurde übrigens dem sowjetischen Geheimdienst zum Verhängnis. Im VENONA-Projekt gelang es den Geheimdiensten von USA und England aufgrund dieses Fehlers Teile der sowjetischen Kommunikation zu entschlüsseln.

In der Praxis sind heute sowohl der Klartext als auch der Schlüssel durch eine Folge von 0 und 1 gegeben, die bitweise mittels XOR verknüpft werden. Es handelt

sich also um die Vigenère-Verschlüsselung eines Klartextes $x_0x_1x_2\ldots$ mit einem Schlüssel $r_0r_1r_2\ldots$ über dem Alphabet $\{0,1\}$ (der für den Schlüssel verwendete Buchstabe r steht für „random“):

$$y_i = (x_i + r_i) \bmod 2 = x_i \oplus r_i.$$

Der Nachteil des One-Time-Pads ist, dass der Schlüssel immer genauso lang wie die Nachricht sein muss. (Wenn Sie also eine Festplatte verschlüsseln wollen, so brauchen Sie eine zweite, gleich große Festplatte, um den Schlüssel zu speichern.) Das ist der Preis, den man für die absolute Sicherheit zahlen muss. Der Aufwand lohnt sich daher nur für sehr sicherheitskritische Anwendungen.

Damit die Sicherheit auch wirklich gegeben ist, muss man bei der Schlüsselerzeugung eines beachten: Denkt man sich einfach nur ein leicht merkbares System zur Schlüsselerzeugung aus, so ist die Wahrscheinlichkeit recht groß, dass dieses System irgendwie bekannt wird. Außerdem können dann wiederum Regelmäßigkeiten des Schlüssels bei der Kryptoanalyse verwendet werden. Der einzige Ausweg ist, für den Schlüssel eine Folge von **Zufallszahlen** zu nehmen. Aber auch hier ist wiederum Vorsicht geboten, denn Zufallszahlen, die auf Computern generiert werden, sind nicht zufällig! In der Regel wird ein Anfangswert („seed“) gewählt, aus dem dann alle weiteren mittels einer Funktion berechnet werden. Man spricht daher von **Pseudozufallszahlen** (siehe Aufwärmübung 7). Gibt es für den Anfangswert nur wenige (z.B. 2^{16}) Möglichkeiten, so können leicht alle „Zufallsfolgen“ durchprobiert werden. Oft wird etwa als Anfangswert die Uhrzeit genommen. Weiß ein Angreifer daher ungefähr, wann der Schlüssel erzeugt wurde, so lässt sich die Anzahl der Möglichkeiten weiter einschränken. Solche Pseudozufallszahlen sind also für kryptographische Anwendungen unbrauchbar. Zur Erzeugung echter Zufallszahlen müssen physikalische Prozesse verwendet werden, deren Verhalten nicht vorhersagbar ist (z.B. thermisches Rauschen eines Widerstandes).

2.6 Kontrollfragen

Fragen zu Abschnitt 2.1: Grundbegriffe

Erklären Sie folgende Begriffe und überprüfen Sie Ihre Antwort mit dem Skriptum: Alphabet, Klartext, Geheimtext, Verschlüsselungsalgorithmus, Schlüssel, Kerckhoffs'sches Prinzip

1. Welche Sicherheitsdienste können kryptographische Verfahren gewährleisten?

   (Lösung zu Kontrollfrage 1)

2. Was ist der Unterschied zwischen Kryptographie und Kryptologie?

   (Lösung zu Kontrollfrage 2)

3. Was ist ein Alphabet? Ist $\mathbb{Z}$ ein Alphabet? Ist $\mathbb{Z}_m$ ein Alphabet ($m \in \mathbb{N}$)?

   (Lösung zu Kontrollfrage 3)

4. Was besagt das Kerckhoffs'sche Prinzip?

   (Lösung zu Kontrollfrage 4)

Fragen zu Abschnitt 2.2: Sicherheit kryptographischer Verfahren

Erklären Sie folgende Begriffe und überprüfen Sie Ihre Antwort mit dem Skriptum: Ciphertext-only attack, known-plaintext attack, chosen-plaintext attack, Brute-Force-Angriff, praktisch sicher, uneingeschränkt sicher, Sicherheitsniveau, Moor'sches Gesetz

1. Was versteht man unter einem (1) Ciphertext-Only-Angriff (2) Known-Plaintext-Angriff (3) Chosen-Plaintext-Angriff?

   (Lösung zu Kontrollfrage 1)

2. Was ist ein Brute-Force-Angriff?

   (Lösung zu Kontrollfrage 2)

3. Was ist ein praktisch sicheres, was ein perfekt sicheres Kryptosystem?

   (Lösung zu Kontrollfrage 3)

4. Was versteht man unter dem Sicherheitsniveau?

   (Lösung zu Kontrollfrage 4)

Fragen zu Abschnitt 2.3: Monoalphabetische Verschlüsselung

Erklären Sie folgende Begriffe und überprüfen Sie Ihre Antwort mit dem Skriptum: Monoalphabetisch, polyalphabetisch, Cäsarverschiebung, Koinzidenzindex.

1. Was ist ein Transpositionsverfahren, was ein Substitutionsverfahren?

   (Lösung zu Kontrollfrage 1)

2. Was ist ein monoalphabetisches, was ein polyalphabetisches Substitutionsverfahren?

   (Lösung zu Kontrollfrage 2)

3. Richtig oder falsch?
   Die affine Chiffre $y = (3 \cdot x + 6) \bmod 26$ ist eine polyalphabetische Chiffre.

   (Lösung zu Kontrollfrage 3)

4. Wie viele monoalphabetische Verschlüsselungen gibt es über einem Alphabet von n Buchstaben (also Klartextaphabet = Geheimtextalphabet)

   (Lösung zu Kontrollfrage 4)

5. Wie muss die Verschiebung $e \in \mathbb{Z}_{10}$ der Cäsarverschiebung $y = x + e \bmod 10$ gewählt werden, wenn die Vorschrift zum Ver- und Entschlüsseln gleich sein soll?

   (Lösung zu Kontrollfrage 5)

6. Wie kann eine monoalphabetische Verschlüsselung gebrochen werden?

   (Lösung zu Kontrollfrage 6)

7. Was ist der Koinzidenzindex eines Textes?

   (Lösung zu Kontrollfrage 7)

Fragen zu Abschnitt 2.4: Polyalphabetische Verschlüsselung

Erklären Sie folgende Begriffe und überprüfen Sie Ihre Antwort mit dem Skriptum: Vigenère-Verschlüsselung, xor-Vigenère-Verschlüsselung.

1. Was ist eine Vigenère-Verschlüsselung? Wie wird (vom Empfänger) entschlüsselt?

   (Lösung zu Kontrollfrage 1)

2. Wie wird eine Vigenère-Verschlüsselung gebrochen?

   (Lösung zu Kontrollfrage 2)

Fragen zu Abschnitt 2.5: One-Time Pad

Erklären Sie folgende Begriffe und überprüfen Sie Ihre Antwort mit dem Skriptum: One-Time Pad, Pseudozufallszahlen.

1. Wie sicher ist das One-Time-Pad?

   (Lösung zu Kontrollfrage 1)

2. Worauf muss bei der Schlüsselerzeugung beim One-Time-Pad geachtet werden?

   (Lösung zu Kontrollfrage 2)

3. Was sind Pseudozufallszahlen?

   (Lösung zu Kontrollfrage 3)

4. Wie kann ein zufälliger Schlüssel mit dem Computer erzeugt werden?

   (Lösung zu Kontrollfrage 4)

Lösungen zu den Kontrollfragen

Lösungen zu Abschnitt 2.1

1. u.a. Geheimhaltung, Authentifizierung, Unversehrtheit oder Verbindlichkeit.

2. Die Verwendung dieser Begriffe ist nicht einheitlich. Meist ist Kryptographie die Wissenschaft der Sicherung von Nachrichten durch Verschlüsselung, Kryptologie umfaßt Kryptographie und Kryptoanalyse (= Wissenschaft des Brechens von Kryptosystemen)

3. Ein Alphabet ist eine endliche Menge; $\mathbb{Z}$ ist daher kein Alphabet, $\mathbb{Z}_m$ ist ein Alphabet.

4. Die Sicherheit des Verschlüsselungsverfahrens darf nicht darauf beruhen, dass das Verschlüsselungsverfahren geheim ist. Es darf nur auf der Geheimhaltung des Schlüssels beruhen.

Lösungen zu Abschnitt 2.2

1. (1) nur (meist relativ viel) Geheimtext ist bekannt; (2) Geheimtext und zugehöriges Stück Klartext (davon meist relativ wenig, oft nur einige Schlagworte) sind bekannt; (3) zu beliebig gewähltem Klartext kann der zugehörige Geheimtext erzeugt werden

2. Ein Angriff, bei dem alle möglichen Schlüssel ausprobiert werden.

3. praktisch sicher: der zum Entschlüsseln nötige Aufwand ist größer als der Wert der verschlüsselten Daten oder die zum Aufbrechen notwendige Zeit übersteigt die Zeit, für die die Daten geheim gehalten werden müssen; perfekt sicher: der Angreifer weiß mit Geheimtext gleich viel wie ohne Geheimtext.

4. Das Sicherheitsniveau ist ℓ Bit, wenn die Anzahl der Operationen des besten bekannten Angriffs $O(2^\ell)$ ist.

Lösungen zu Abschnitt 2.3

1. Transpositionsverfahren: Klartextzeichen werden in andere Reihenfolge gebracht; Substitutionsverfahren: Klartextzeichen werden durch andere Zeichen ersetzt, bleiben aber an Ort und Stelle.

2. monoalphabetisch: ein Klartextzeichen wird bei der Verschlüsselung stets auf dasselbe Geheimtextzeichen abgebildet; polyalphabetisch = nicht monoalphabetisch

3. falsch

4. $n!$ (alle möglichen Permutationen der n Buchstaben)
5. $e = 5$
6. Oftmals durch statistische Analyse (dazu muss der Geheimtext lang genug sein)
7. Er gibt die quadratische Abweichung der Buchstabenhäufigkeiten von einer Gleichverteilung an.

Lösungen zu Abschnitt 2.4

1. Die einzelnen Zeichen des Klartextes werden abwechselnd mit verschiedenen Verschiebechiffren verschlüsselt; die jeweilige Verschiebung eines Klartextzeichens wird durch das zugehörige Zeichen eines Schlüsselwortes bestimmt, das periodisch fortgesetzt über den Klartext geschrieben wird. Entschlüsselt wird mit demselben Verfahren mit jenem Schlüsselwort, dessen Zeichen die additiven Inversen des Verschlüsselungs-Schlüsselwortes sind.
2. Zunächst wird die Schlüssellänge k bestimmt (zum Beispiel mithilfe des Koinzidenzindex). Danach wird der Geheimtext in die entsprechenden k monoalphabetisch verschlüsselten Teiltexte zerlegt und deren Verschlüsselungen separat gebrochen.

Lösungen zu Abschnitt 2.5

1. Das One-Time-Pad ist ein perfektes Chiffriersystem, also uneingeschränkt sicher. Das bezahlt man aber mit einem hohen Aufwand für die Schlüsselerzeugung bzw. für den Schlüsselaustausch.
2. Er muss zufällig erzeugt werden; jegliche Regelmäßigkeit (Funktionsvorschrift zur Schlüsselerzeugung) gibt einen Angriffspunkt und zerstört damit die perfekte Sicherheit. Der Schlüssel darf nur einmal verwendet werden.
3. Das sind Zufallszahlen, die nach einer bestimmten Vorschrift erzeugt werden. Sie sind damit vorhersagbar, wenn man die Parameter für ihre Erzeugung kennt (insbesondere die Initialisierung „seed“). Sie sind für die Kryptographie daher nicht geeignet.
4. Für die Erzeugung von echten Zufallszahlen (im Gegensatz zu Pseudozufallszahlen) werden zufällige physikalische Prozesse verwendet, wie z.B. das thermische Rauschen einer Widerstandes im Prozessor.

2.7 Übungen

Aufwärmübungen

1. **Durchschnittlicher Aufwand für einen Brute-Force-Angriff**: Ein Einbrecher steht mit einem Schlüsselbund von 10 Schlüsseln vor einem Schloss, zu dem genau einer der Schlüssel passt. Er kennt den richtigen Schlüssel nicht und probiert zufällig einen Schlüssel nach dem anderen (ohne Zurücklegen eines bereits probierten Schlüssels). X= *Anzahl der Versuche, bis der richtige Schlüssel gefunden ist.*
   a) Geben Sie die Verteilung von X an (d.h. die möglichen Werte von X und die zugehörigen Wahrscheinlichkeiten).
   b) Wie viele Versuche muss der Einbrecher im Mittel durchführen? (Tipp: Erwartungswert $E(X)$ berechnen)

2. Gegeben ist folgende polyalphabetische Verschlüsselungsvorschrift (ohne Schlüssel), die vom Alphabet $\{a, b, c\}$ auf das Alphabet $\{r, s, t, u, v, w\}$ abbildet:

$$\begin{aligned} e(a) &= r \vee s \\ e(b) &= t \vee u \\ e(c) &= v \vee w \end{aligned}$$

   wobei $\vee$ bedeutet, dass hier eine zufällige Wahl getroffen wird.
   a) Verschlüsseln Sie *abba*. Wie viele Möglichkeiten gibt es?
   b) Wie viele Klartexte/Geheimtexte einer bestimmten Länge n gibt es?
   c) Ist diese Verschlüsselungsvorschrift umkehrbar? Wie lautet in diesem Fall die Entschlüsselungsvorschrift?

3. Geben Sie die Entschlüsselungsvorschrift zur Cäsar-Verschiebung

$$y = (x + e) \bmod 26$$

   mit $e = 5$ an.

4. **Monoalphabetische Verschlüsselung:** Verschlüsseln Sie den Klartext x_1, x_2 = `J,A` über dem Alphabet $\mathbb{Z}_{256}$ (Bytes gemäß ASCII-Code) mittels

$$y_i = x_i \oplus e$$

   unter Verwendung von $e = 01011000 \in \mathbb{Z}_{256}$. Wie entschlüsselt der Empfänger?

5. **Vigenère Verschlüsselung**: Verschlüsseln Sie einen beliebig gewählten Text über $\mathbb{Z}_{26}$ mit dem Vigenère-Algorithmus $y_i = (x_i + e_{i \bmod k}) \bmod \mathbb{Z}_{26}$ mit dem Schlüsselwort `GEHEIM` $= 6, 4, 7, 4, 8, 12$. Wie lauten der Schlüssel und die Vorschrift zum Entschlüsseln?

6. **Known Plaintext Attack (Klartextangriff)**: Sie kennen von einem Geheimtext $5, 1, 3, 3, 7, 9, 4, \ldots$ die Entschlüsselung der ersten vier Zeichen, wissen also, dass der Klartext so beginnt: $2, 2, 3, 1, \ldots$. Sie kennen weiters die Verschlüsselungsvorschrift (**Hill Chiffre**), die mittels Matrixmultiplikation in der Form

$$\begin{pmatrix} y_i \\ y_{i+1} \end{pmatrix} = \begin{pmatrix} a & b \\ c & d \end{pmatrix} \begin{pmatrix} x_i \\ x_{i+1} \end{pmatrix}, \text{ für } i = 1, 3, 5, \ldots$$

definiert werden kann. Das Alphabet ist $\mathbb{Z}_{11}$. Finden Sie mithilfe eines Klartextangriffs die geheimen Bestandteile a, b, c, d (= Schlüssel) der Verschlüsselungsvorschrift.

7. Eine einfache Methode **Pseudozufallszahlen** zu erzeugen ist die sogenannte lineare Kongruenzenmethode. Dazu wählt man vier ganze Zahlen: a, den Modul m, das Inkrement c und den Anfangswert (*seed*) x_0, die die folgenden Bedingungen erfüllen: $2 \leq a < m$, $0 \leq c < m$, und $0 \leq x_0 < m$. Der Startwert x_0 wird zu Beginn auf einen festen Wert gesetzt oder z. B. abhängig von Datum und/oder Uhrzeit initialisiert. Dann berechnet man die Pseudozufallszahlen nach

$$x_0 \text{ beliebig}, \qquad x_n = (a\, x_{n-1} + c) \bmod m.$$

x_n ist also der Rest von $a\, x_{n-1} + c$ bei Division durch m. Der oben beschriebene Zufallszahlengenerator liefert Zufallszahlen zwischen 0 und $m - 1$ (das sind die Reste, die bei Division durch m auftreten können). Nach spätestens m Schritten müssen sich daher die Zufallszahlen wiederholen.

Berechnen Sie die Pseudozufallszahlen, wenn
a) $m = 9$, $a = 7$, $c = 4$ und $x_0 = 3$ b) $m = 16$, $a = 5$, $c = 7$ und $x_0 = 2$
c) $m = 16$, $a = 8$, $c = 7$ und $x_0 = 2$

Man kann zeigen, dass für $c = 0$, m eine Primzahl und a beliebig, alle Werte bis auf die 0 durchlaufen werden, bevor sich ein Wert wiederholt. In der Praxis wird oft $m = 2^{31} - 1$, $a = 7^5$ und $c = 0$ verwendet. Eine schnelle Methode mit besseren statistischen Eigenschaften ist der **Mersenne-Twister**. Für kryptographische Zwecke gibt es eigene (dann aber langsamere) Algorithmen wie z.B. der **Blum-Blum-Shub-Generator**.

Weiterführende Aufgaben

1. **Aufwand für einen Brute-Force-Angriff auf AES mit einem 128-Bit-Schlüssel**: Gegeben sei ein spezielles IC, das $3 \cdot 10^7$ Schlüssel pro Sekunde testen kann.

   a) Wie lange dauert ein Brute-Force-Angriff im Mittel, wenn 100.000 solcher IC parallel verwendet werden? (Geben Sie das Ergebnis relativ zum Alter

des Universums an, das mit 10^{10} Jahren angenommen wird.)

b) Wie viele Jahre muss man warten, bis ein Brute-Force-Angriff im Schnitt nur mehr 1 Tag dauert? Schätzen Sie dies mithilfe des Moore'schen Gesetzes ab (d.h., der Annahme, dass sich die Rechenleistung alle 18 Monate verdoppelt, also alle 3 Jahre vervierfacht).

2. **Affine Chiffre**: Betrachten Sie für das Alphabet $\mathbb{Z}_n$ die folgende monoalphabetische Verschlüsselungsvorschrift:

$$y = (t \cdot x + e) \bmod n$$

a) Geben Sie die Entschlüsselungsvorschrift an.
b) Wie viele mögliche geheime Schlüssel (t, e) gibt es bei einem Alphabet mit $n = 26$ Buchstaben?
c) Wählen Sie eine beliebige kurze Nachricht und verschlüsseln Sie diese mit dem Schlüssel $(t, e) = (3, 6)$. Entschlüsseln Sie danach wieder mit der zugehörigen Entschlüsselungsvorschrift.

3. **Affine Doppelverschlüsselung**: Wird die Sicherheit durch Hintereinanderausführung (Verkettung) $(E_2 \circ E_1)(x)$ zweier affiner Chiffren $E_1(x) = (a \cdot x + b) \bmod n$ und $E_2(x) = (c \cdot x + d) \bmod n$ erhöht?

a) Zeigen Sie, dass dies *nicht* so ist, indem Sie eine affine Einfachverschlüsselung $E_3(x) = (e \cdot x + f) \bmod n$ angeben, die *dieselbe* Ver- und Entschlüsselung durchführt wie die Doppelverschlüsselung $(E_2 \circ E_1)(x)$.

b) Geben Sie e und f an für $a = 3$, $b = 5$, $c = 11$, $d = 7$ und $n = 26$.

c) Verschlüsseln Sie den Buchstaben F zunächst mit $(E_2 \circ E_1)(x)$ und danach mit $E_3(x)$ und überzeugen Sie sich von der Gleichwertigkeit.

d) Wie groß ist der Schlüsselraum, den ein Angreifer bei einem Brute-Force Angriff durchsuchen muss, wenn er weiß, dass eine affine Doppelverschlüsselung angewendet wurde?

4. **Chosen Plaintext Attack**: Betrachten wir nochmals eine affine Chiffre:

$$y = (t \cdot x + e) \bmod n$$

a) Angenommen, Angreifer Mallory kann Alice dazu bringen, zwei von ihm gewählte Klartextbuchstaben x_1, x_2 zu verschlüsseln und ihm die entstandenen Geheimtextbuchstaben y_1 bzw. y_2 zu nennen. Wie kann der Angreifer nun aus Kenntnis von x_1, x_2, y_1, y_2 den geheimen Schlüssel (t, e) ermitteln?
b) Worauf muss der Angreifer achten, wenn er x_1 und x_2 auswählt?

5. **Hill-Chiffre**: Ein Klartext $x_1, x_2, x_3, \ldots$ über dem Alphabet $\mathbb{Z}_n$ wird in Blöcke von je zwei Buchstaben zerlegt, also (x_1, x_2), (x_3, x_4), $\ldots$, und mithilfe

Matrixmultiplikation blockweise verschlüsselt:

$$\begin{pmatrix} y_i \\ y_{i+1} \end{pmatrix} = \begin{pmatrix} a & b \\ c & d \end{pmatrix} \begin{pmatrix} x_i \\ x_{i+1} \end{pmatrix}, \quad i = 1, 3, 5, \ldots$$

Dabei sind $a, b, c, d \in \mathbb{Z}_n$ die Bestandteile des geheimen Schlüssels.
a) Verwenden Sie das Alphabet $\mathbb{Z}_n = \mathbb{Z}_{27}$ und verschlüsseln Sie die Nachricht $x_1, x_2, x_3, x_4 = 11, 5, 24, 2$ mit dem Schlüssel $(a, b, c, d) = (4, 2, 3, 5)$.
b) Berechnen Sie über $\mathbb{Z}_{27}$ die inverse Matrix zu

$$A = \begin{pmatrix} 4 & 2 \\ 3 & 5 \end{pmatrix}$$

und geben Sie dann die Entschlüsselungsvorschrift zu a) an.
c) Können Alice und Bob grundsätzlich bei der Vereinbarung eines geheimen Schlüssels beliebige Zahlen $a, b, c, d \in \mathbb{Z}_n$ wählen?

6. **Known Plaintext Angriff**: Gegeben ist eine Hill Chiffre über dem Alphabet $\mathbb{Z}_{27} = \{0, \ldots, 26\}$ bzw. {`A, B, ..., Z, :`} (d.h., es wird zu den 26 Buchstaben als 27. Zeichen der Doppelpunkt „:“ hinzugefügt):

$$\begin{pmatrix} y_i \\ y_{i+1} \end{pmatrix} = \begin{pmatrix} a & b \\ c & d \end{pmatrix} \begin{pmatrix} x_i \\ x_{i+1} \end{pmatrix}, \quad i = 1, 3, 5, \ldots,$$

mit $a, b, c, d \in \mathbb{Z}_{27}$. Sie als Angreifer wissen, dass der Geheimtext

`WM:LJTZPJIGURIQDLDXWFJMYLXDCIX`

mit diesem Algorithmus verschlüsselt wurde und außerdem, dass die ersten vier Zeichen `FROM` bedeuten.
a) Wie können Sie mit diesem Wissen a, b, c, d berechnen?
b) Von wem stammt der Funkspruch?

7. **Ciphertext Only Attack**: Gegeben ist folgender Geheimtext.
`IMDGYWAQZZQZUZRADYMFUWQDNQEEQDDQOTZQZMXEYMFTQYMFUWQD`
`IQUXEUQYUFUTDQZLQTZRUZSQDZNUEFMGEQZPHUQDGZPLIMZLUSLMQTXQZWAQZZQZ`
Sie wissen, dass er aus einem deutschen Klartext durch eine Cäsarverschiebung $y = (x + e) \bmod 26$ entstand.
a) Finden Sie den geheimen Schlüssel e.
b) Was erfahren wir hier über Informatiker:innen?

8. Verschlüsseln Sie einen beliebig gewählten Text über dem Alphabet $\mathbb{Z}_{256}$ mit

$$y_i = x_i \oplus e$$

mit beliebig gewähltem $e \in \mathbb{Z}_{256}$ (Codierung der Buchstaben als binäre Folge im ASCII-Code).

9. **Known plaintext Angriff**: Der Geheimtext

$$\{111, 66, 75, 75, 72, 7, 80, 72, 85, 75, 67, 6\}$$

entstand durch xor-Verschlüsselung $y = x \oplus e$ eines englischen Textes über dem Alphabet $\mathbb{Z}_{256}$. Sie wissen, dass 111 zu H (= 72 im ASCII Code) entschlüsselt wird. Entschlüsseln Sie den gesamten Geheimtext.

Lösungen zu den Aufwärmübungen

1. a) X kann die Werte $1, 2, \ldots, 10$ annehmen. Die zugehörigen Wahrscheinlichkeiten sind: $X = 1 : p_1 = \frac{1}{10}$; $X = 2 : p_2 = \frac{9}{10} \cdot \frac{1}{9} = \frac{1}{10} \ldots$, allgemein: $p_i = \frac{1}{10}$ (**Gleichverteilung**).
b) Der Erwartungswert ist: $\mu = \frac{1}{10} \sum_{i=1}^{10} i = \frac{1}{10} \frac{10 \cdot 11}{2} = 5.5$. Der Einbrecher muss also im Mittel 5.5 Schlüssel probieren.

2. a) Es gibt 2^4 mögliche Verschlüsselungen von *abba*, z.B. *rtus*.
b) 3^n Klartexte (n Stellen, für jede Stelle 3 Möglichkeiten), 6^n Geheimtexte
c) Die Verschlüsselungsvorschrift ist umkehrbar; entschlüsselt wird mit $d(r) = d(s) = a$, $d(t) = d(u) = b$, $d(v) = d(w) = c$.

3. $x = y - e = y - 5 = y + 21 \bmod 26$

4. Schlüssel $e =$ `X` $= 88$ (ASCII-Code) $= 64 + 16 + 8 = 2^6 + 2^4 + 2^3 = 01011000$. Klartext $x_1, x_2 = J, A = 74, 65 = 01001010, 01000001$. Zugehöriger Geheimtext: $y_1, y_2 = 00010010, 00011001 = 18, 25$. Entschlüsselt wird mit $x_i = y_i \oplus e$.

5. Klartext `DASISTNEU` in Zahlen: $3, 0, 18, 8, 18, 19, 13, 4, 20$:

Schlüsselwort	`06, 04, 07, 04, 08, 12, 06, 04, 07`
Klartext	`03, 00, 18, 08, 18, 19, 13, 04, 20`
Geheimtext	`09, 04, 25, 12, 00, 05, 19, 08, 01`

Schlüssel zum Entschlüsseln: $d_i = 26 - e_i = 20, 22, 19, 22, 18, 14$. Entschlüsselt wird mit $y_i = (x_i + d_{i \bmod 6}) \bmod \mathbb{Z}_{26}$

6. Stelle lineares Gleichungssystem auf: $2a + 2b = 5$, $2c + 2d = 1$, $3a + b = 3$, $3c + d = 3$ und löse es: $a = 3, b = 5, c = 4, d = 2$

7. a) $7, 8, 6, 1, 2, 0, 4, 5, 3$
b) $1, 12, 3, 6, 5, 0, 7, 10, 9, 4, 11, 14, 13, 8, 15$
c) $7, 15, 15, \ldots, 15$

Lösungen zu ausgewählten Aufgaben

1. a) $1.8 \cdot 10^8 \cdot$ Alter des Universums b) ca. 100 Jahre (Moore'schen Gesetz gilt nicht so lange!)

2. a) $x = t^{-1}(y - e) \bmod n$; es muss $\mathrm{ggT}(t, n) = 1$ gelten.
   b) $12 \cdot 26 = 312$ Tauschchiffren
   c) –

3. a) Hinweis: Setzen Sie für E_1 bzw. E_2 in $y = (E_2 \circ E_1)(x)$ ein und vereinfachen anschließend.
   d) gleich groß wie bei einer einfachen affinen Verschlüsselung

4. a) –
   b) Hinweis: Welche Voraussetzung muss erfüllt sein, damit die Formel für t Sinn macht?

5. a) $y_1, y_2, y_3, y_4 = 0, 4, 19, 1$
   b)
   $$\begin{pmatrix} x_i \\ x_{i+1} \end{pmatrix} = (ad - bc)^{-1} \begin{pmatrix} d & -b \\ -c & a \end{pmatrix} \begin{pmatrix} y_i \\ y_{i+1} \end{pmatrix} = \begin{pmatrix} 10 & 23 \\ 21 & 8 \end{pmatrix} \begin{pmatrix} y_i \\ y_{i+1} \end{pmatrix}.$$
   c) –

6. a) $a = 1, b = 1, c = 1, d = 2$.
   b) von Captain Kirk

7. Hinweis: Zum Brechen einer Cäsarverschiebung genügt ein einziges zusammengehörendes Paar Geheimtext-Klartextbuchstabe.

8. Schlüssel $e =$ `X` $= 88$ (ASCII-Code) $= 01011000$. Geheimtext: $y_1, y_2 = 00010010, 00011001 = 18, 25$.

9. $e = 39$; Klartext ist `Hello_world!`

Literatur

[Bar20] E. Barker. „Recommendation for Key Management: Part 1 – General". In: *NIST Special Publication* 800-57 Part 1, Revision 5 (2020). URL: https://doi.org/10.6028/NIST.SP.800-57pt1r5.

[CS08] R. Smith. *Stream Cipher Reuse: A Graphic Example*. URL: https://cryptosmith.com/2008/05/31/stream-reuse/ (besucht am 20. 02. 2018).

[DH76] W. Diffie und M. E. Hellman. „New directions in cryptography“. In: *IEEE Trans. Information Theory* IT-22.6 (1976), S. 644–654. URL: https://www-ee.stanford.edu/~hellman/publications/24.pdf.

[Sch16] K. Schmeh. *Kryptographie*. 6. Aufl. Heidelberg: dpunkt.verlag, 2016.

[Wob01] R. Wobst. *Abenteuer Kryptologie. Methoden, Risiken und Nutzen der Datenverschlüsselung*. 3. Aufl. Bonn: Addison Wesley, 2001.