Security – Software Development

Schuljahr 2023/2024

Tischvorlage

Wischounig Philipp

Abteilung für Wirtschaftsingenieure – Betriebsinformatik

HTBLVA Innsbruck

1 Aspekte der Datensicherheit

Was fällt alles unter „Datensicherheit“?

Von John McCumber wurde ein Würfelmodell (*Cybersecurity Cube*, *McCumber Cube*) vorgeschlagen, das drei Dimensionen der Datensicherheit darstellen soll, siehe Abbildung 1.

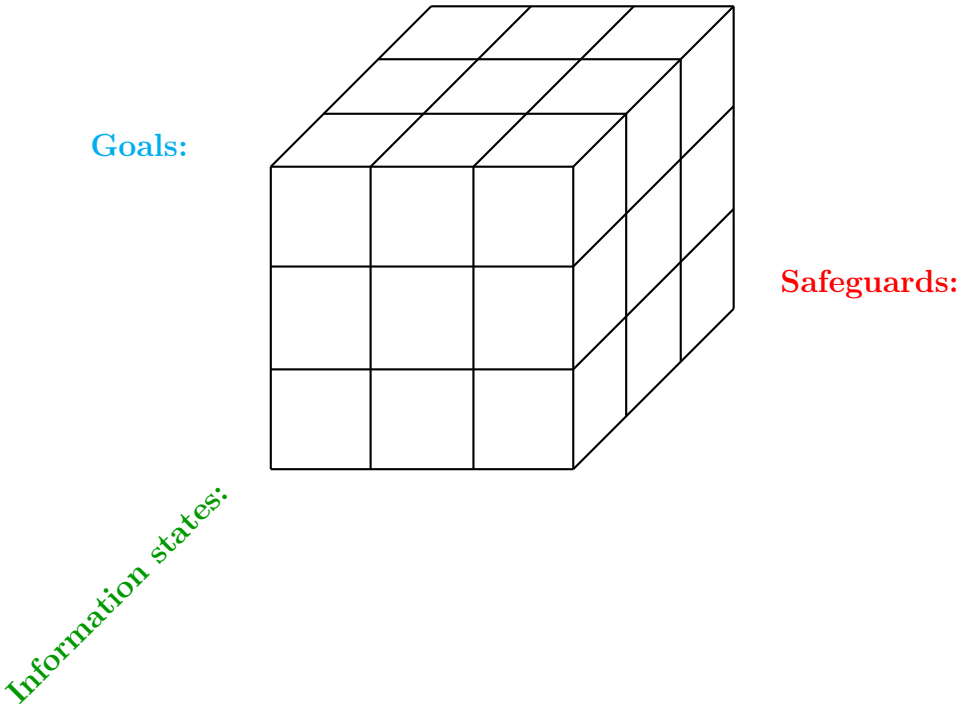

Abbildung 1: Das von McCumber vorgeschlagene Modell für Datensicherheit in Form eines Würfels.

Die dreidimensionale Darstellung als Würfel

. . . . . .

Bei der Entwicklung von Datensicherheit, müssen alle Faktoren und auch ihre Verbindungen untereinander berücksichtigt werden. Der Würfel soll zu einer methodischen Vorgehensweise verhelfen.

- Beispielsweise ist es für Vertraulichkeit nicht ausreichend, wenn Daten auf der Festplatte verschlüsselt vorliegen, aber unverschlüsselt über das Internet versendet werden oder Mitarbeiter den Schlüssel auf USB-Sticks herumliegen lassen.
- Ohne Policy mit Richtlinien für regelmäßige Sicherungen wird die Verfügbarkeit kaum gewährleistet sein, auch wenn Personen gut geschult sind und Technologie (Firewall) vor Angriffen schützt (vor wortwörtlichem Feuer fehlt der Schutz).

Ein paar Kommentare zu den einzelnen Aspekten:

- **Safeguards/Schutzmaßnahmen:**
  - Persons: Es benötigt *fortwährende* Bewusstseinsschulungen zum Umgang mit Daten und Geräten, zum Erkennen von *Social Engineering* etc.
  - Policies: Definieren (in-)akzeptables Verhalten und wie bei einem Vorfall reagiert wird. Regeln Passwortrichtlinie, wer wohin darf, definieren Konsequenzen etc.
  - Technology: Sowohl Software als auch Hardware wie Firewall, Intrusion Detection System, physische Zugangskontrollen etc.
- **Information States/Zustände:**
  - Data at rest: Daten, die gespeichert sind, z. B. auf HDD, USB-Stick, RAID, NAS, Cloud.
  - Data in process: Umfasst Dateneingabe (Datenformat, Eingabefehler usw.), -ausgabe und -veränderung.
  - Data in transmission: Daten werden versendet über LAN, WLAN, Sneaker net[1]) usw.
- **Goals/Ziele:**
  - Confidentiality (Vertraulichkeit): Neben Zugriffskontrollen und Verschlüsselung gelten als weitere Strategien *Datensparsamkeit* (Nur vorhandene Daten können in falsche Hände geraten.), *Anonymisierung* (Speichere ich nur Pseudonyme zu Gesundheitsdaten, ist der Schaden gering, wenn sie öffentlich werden.), *Tokenization* (Kreditkartennummer wird durch andere Nummer ersetzt.) und *Steganographie* (Daten werden versteckt.).
  - Integrity: Durch Abwehr bösartiger Angriffe, Konsistenzchecks, Prüfsummen etc.
  - Availability (Verfügbarkeit): Häufige Probleme sind DoS-Angriffe, Stromausfälle, Naturkatastrophen. Sichergestellt wird sie durch Abwehrmaßnahmen, Redundanz, Back-Ups, Wartung der Bauteile (Reinigen[2]), Wiederherstellungspläne, Testen uvw.

Neben den drei Zielen der *CIA-Triade* werden oft noch weitere genannt:

- **Authenticity (Authentizität):**

. . . . . .

[1] zu Deutsch: Turnschuhnetzwerk. „Man sollte nicht die Bandbreite einer Autobahn, über die ein mit Festplatten beladenes Fahrzeug fährt, unterschätzen.“

[2] siehe auch Ursprung des Begriffes *Bug*

- **Non-repudiation (Nichtabstreitbarkeit):**

  .....

  .....

  .....

- **Accountability (Zurechenbarkeit):**

  .....

- **Anonymity (Anonymität):**

  .....

2 Kryptologie

Die Kryptologie ist die Wissenschaft des Verschlüsselns. Sie wird unterteilt in:

- **Kryptographie**: Wissenschaft des Verschlüsselns (und berechtigten Entschlüsselns)
- **Kryptoanalyse**[3]: Wissenschaft, kryptographische Verfahren zu brechen

Wozu dient Verschlüsselung?

Verschlüsselt wird klarerweise, um das Ziel der ..... zu erreichen. Die asymmetrische Kryptographie kann aber auch zur Erlangung von

.....

Wie funktioniert Verschlüsselung?

Alice möchte eine *Nachricht*, den *Klartext*, engl. *plaintext*, an *Bob* schicken und zwar so, dass unberechtigte Zuhörer wie *Eve*[4] diesen *Chiffretext*, engl. *ciphertext*, nicht verstehen können, siehe Abbildung 2. Die Daten (z. B. Buchstaben der Nachricht) werden dazu so durcheinandergebracht und verändert, dass sie für Eve nicht rekonstruierbar sind, für Bob aber schon. Alice und Bob benötigen für das „Unverständlichmachen“ und das „Verständlichmachen“ der Nachricht *Schlüssel*.

[3] oder: Kryptanalyse

[4] von engl. eavesdropper, Lauscher; Setzt die Angreiferin aktiv böswillige Aktionen und lauscht nicht bloß, wird sie als *Mallory* bezeichnet, von engl. malicious attacker.

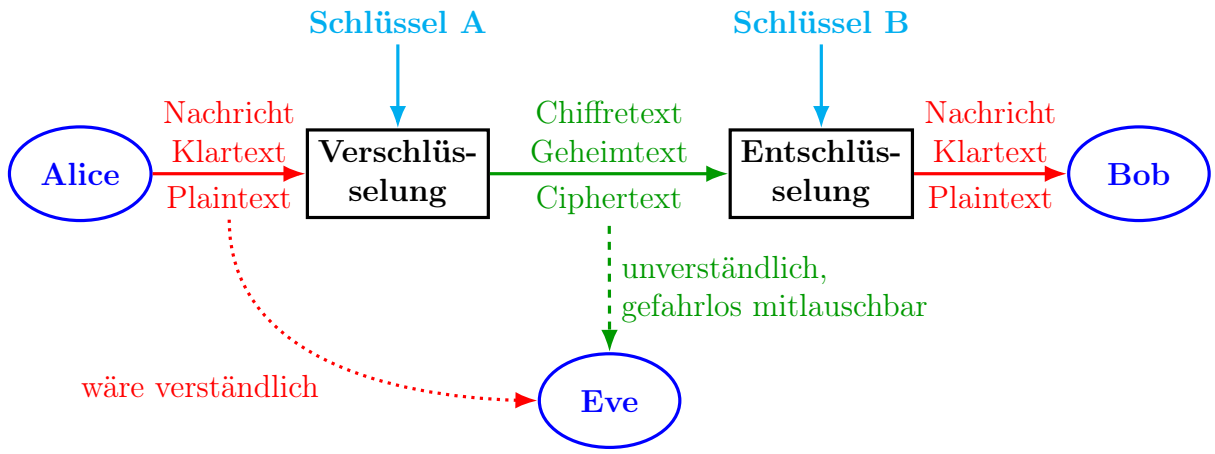

Abbildung 2: Die grundlegende Situation bei verschlüsselter Kommunikation: Alice und Bob wollen Nachrichten austauschen, die unberechtigte Lauscher wie Eve nicht hören (eigentlich: nicht verstehen) dürfen (rot). Deshalb verändert Alice die Nachricht so, dass sie für Eve unverständlich wird (grün) und versendet sie erst dann über einen unsicheren Kanal. Bob muss in der Lage sein, den Chiffretext wieder zu entziffern. Er benötigt dafür einen Schlüssel, den Eve nicht hat.

Geschichtliche Einordnung

Der Wunsch, militärische, politische und persönliche Geheimnisse vor Unbefugten zu schützen, ist mit Sicherheit alt.

Einfache Verfahren dazu sind z. B. von ..... bekannt. Es gab neuzeitliche Verfahren, die sich lange der Analyse widersetzten (z. B.

.....

Während des *Ersten* und *Zweiten Weltkrieges* wurde die Kommunikation zwischen den Truppen verschlüsselt, mitgehört, geknackt, anders verschlüsselt usw. Im Zweiten Weltkrieg wurden erstmals MathematikerInnen miteinbezogen (in England Alan Turing zur Analyse der *Enigma* der deutschen Wehrmacht). Im Gegensatz zu früher, als die Kryptoanalyse eher von SprachwissenschaftlerInnen betrieben wurde, ist sie heute stark durch InformatikerInnen und MathematikerInnen geprägt.

Vielleicht war früher der Bedarf nach Verschlüsselung für Herrschende und Regierungen höher als für Privatpersonen. Im heutigen Informationszeitalter ist sie sicherlich auch im privaten Bereich von Bedeutung.[5]

Die Zeit der asymmetrischen Kryptosysteme, die das *Schlüsselaustauschproblem* lösten, begann 1976. Mit ihnen ist es auch möglich, dass zwei Personen, die noch nie Kontakt

[5] Aber auch heute noch haben Regierungen und ihre Geheimdienste Verschlüsselungen besonders im Blick und hätten oft gerne das Monopol. Das sieht man an den ständigen Versuchen, Bundestrojaner zu legalisieren, Vorstößen, Verschlüsselung zu verbieten oder den Einbau von Hintertüren zu erzwingen, oder Gesetzen, die den Import und Export von kryptographischen Geräten wie den von Waffen regeln.

hatten (Internet!), abhörsicher kommunizieren können.

Wie wird die Unverständlichkeit für Eve erreicht?

Früher wurde vor allem darauf gesetzt, dass Unbefugten ..... war. Das hat viele Nachteile:

- Es funktioniert vielleicht, wenn meine Armee die Methode kennt, aber die gegnerische nicht (klares Freund-Feind-Schema). Wenn ich aber mit zwei Personen kommunizieren möchte, ohne das sie dem Verkehr mit der jeweils anderen lauschen können, brauche ich zwei verschiedene Verfahren!

- Ein neues Verfahren zu entwickeln ist schwierig. Ebenso, es an alle Befugten (aber an sonst niemanden!) zu verbreiten.

- Das Verfahren geheimzuhalten, ist schwierig, ganz besonders, wenn ich es im größeren Maßstab anwenden möchte. Irgendwer könnte sich verplappern, gefangengenommen und gefoltert, bestochen oder bedroht werden. Selbst wenn ich das bemerkte, bräuchte ich ein neues Verfahren.

- Ein von mir entwickeltes Verfahren hat mit hoher Wahrscheinlichkeit Schwachstellen. Moderne Verfahren sind öffentlich bekannt (AES wurde sogar ausgeschrieben) und seit Jahrzehnten erforscht, analysiert und diskutiert.[6]

  Beispielsweise war ein wesentlicher Schwachpunkt der Enigma, dass *fixpunktfreie Permutationen* verwendet wurden. Der Klartextbuchstabe „M“ konnte als Chiffretext nicht wieder „M“ werden. Dadurch verringert sich die Anzahl der Möglichkeiten unnötig und es ergaben sich Angriffspunkte.

Deswegen folgen *heute* anerkannte Verfahren (DES, AES, RSA etc.) dem

.....

„Das Verfahren darf nicht der Geheimhaltung bedürfen und soll ohne Schaden in Feindeshand fallen können.“

Moderne Verfahren sind so sicher, dass sie vor Angriffen sicher sind, auch wenn die Angreifer das Verfahren kennen. Die Sicherheit liegt alleine in der Geheimhaltung des Schlüssels.[7]

Schlüssel lassen sich leichter erzeugen als Verfahren, einfacher verbreiten, einfacher geheimhalten. Außerdem kann ich ohne Schwierigkeiten für die Kommunikation mit mehreren Personen verschiedene Schlüssel verwenden. Nebenbei machen sich die Entwickler des Verfahrens durch dessen Veröffentlichung nicht länger erpressbar (Folter ist zwecklos).

[6] Trotzdem bilden sich immer wieder Institutionen ein, dass sie es besser könnten. (Beispiele sind die unsicheren GSM-Algorithmen A5/1 und A5/2 und *Magenta*, die alle bald geknackt wurden.)

[7] Bei der Enigma war das zum Beispiel nicht der Fall. Deren Sicherheit hing wesentlich von der Verdrahtung der Walzen ab. Sie durfte also nicht in Feindeshand gelangen!

Key Management

Auch wenn es einfacher ist, die Sicherheit über Geheimhaltung des Schlüssels statt des Verfahrens zu erlangen, bleibt das *Key Management* heikel. Es bezeichnet das Erzeugen von Schlüsseln, ihren Austausch, ihre Speicherung, Nutzung und Vernichtung. Moderne Chiffren sind so gut entworfen, dass die meisten Attacken Social Engineering sind oder Seitenkanalangriffe oder am Key Management ansetzen.

Es ist viel schwieriger, als man vermuten möchte, „gute“ Zufallszahlen zu generieren. Außerdem müssen zuverlässig die Verwendung schwacher Schlüssel und generell ihre Wiederverwendung verhindert werden.

Wie hört Eve überhaupt ab?

Aus Glasfaserkabeln .....

Bei WLAN und anderen drahtlosen Übertragungen

.....

Weitere Möglichkeiten sind .....

2.1 Symmetrische und asymmetrische Kryptosysteme

Bei wie in Abbildung 2 beschriebenen Kryptosystemen gibt es zwei grundverschiedene Möglichkeiten für die Schlüssel von Alice und Bob:

- Beide besitzen **den gleichen** Schlüssel. Man spricht dann von einem **symmetrischen Kryptosystem**.
  - Folglich müssen beide den Schlüssel .....

    weil sonst Eve .....

    oder Mallory .....
  - Beide können Nachrichten .....
  - Typische Schlüssellängen sind .....
  - Die Algorithmen sind typischerweise ....., weil sie auf einfachen Operationen (Addition, XOR, Bitshifts, Substitutionen etc.) beruhen.
  - Vergleichen lässt sich das Kryptosystem mit einer Kiste mit Vorhängeschloss, das *vor Versand versperrt* werden muss, und zwei identischen Schlüsseln für Alice und Bob.

– Alle symmetrischen Kryptosysteme haben das Problem des

  .....

  Wie können Alice und Bob an den gemeinsamen Schlüssel gelangen? Ohne persönlichen Kontakt würde man einen sicheren Kommunikationskanal benötigen, siehe Abbildung 3.

- Beide besitzen **verschiedene** Schlüssel, die aber natürlich zusammenpassen müssen. Man spricht dann von einem **asymmetrischen Kryptosystem**.

  – Bob hat einen ..... Schlüssel, mit dem die Nachricht *entschlüsselt* werden kann. Der Schlüssel zum *verschlüsseln* von Nachrichten, die an Bob versandt werden, ist nicht nur Alice bekannt sondern sogar

  ..... So kann Eve ....., aber Mallory

  .....

  – Alice ihre Nachrichten verschlüsseln .....

  – Typische Schlüssellängen sind .....

  – Die Algorithmen sind typischerweise ....., weil sie auf vergleichsweise schwierigeren mathematischen Operationen beruhen (z. B. Potenzieren).

  – Vergleichen lässt sich das Kryptosystem mit offenen Vorhängeschlössern, die Bob verteilt und für die nur Bob den Schlüssel hat.

  – Da Alice nur .....
  bentötigt, der nicht sicher übertragen werden muss, lösen asymmetrische Kryptosysteme das Schlüsselaustauschproblem, siehe Abbildung 4.

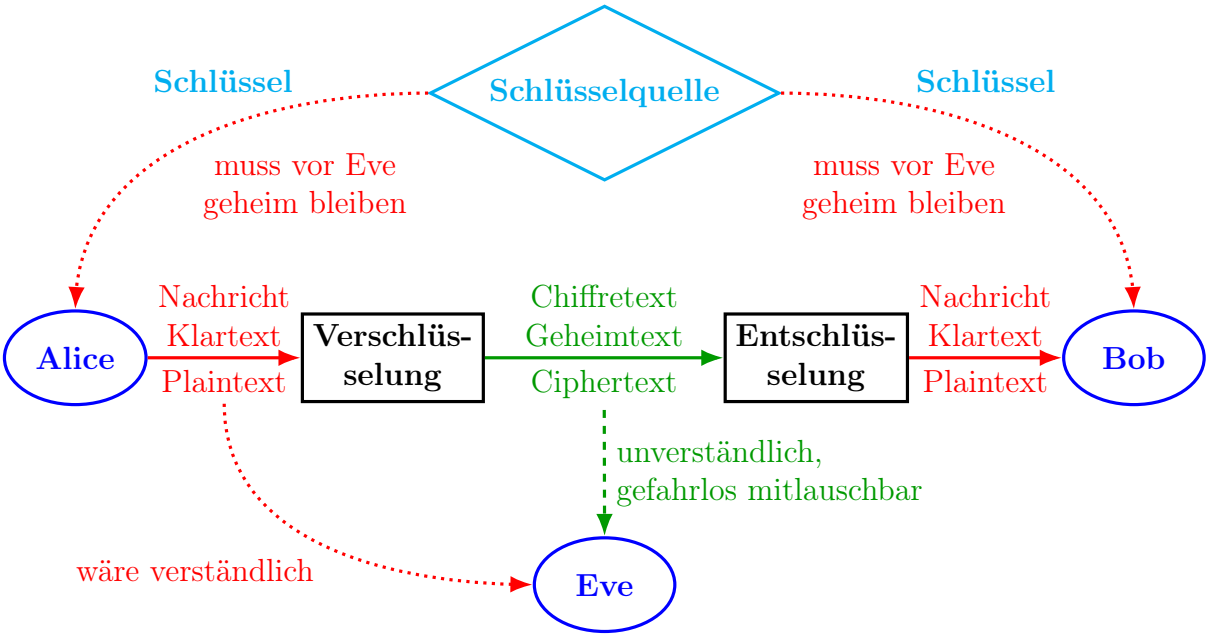

Abbildung 3: Bei symmetrischen Kryptosystemen wird der Schlüssel von einem der Teilnehmer oder einer vertrauenswürdigen Quelle erzeugt. Er muss über einen sicheren Kommunikationskanal verteilt werden.

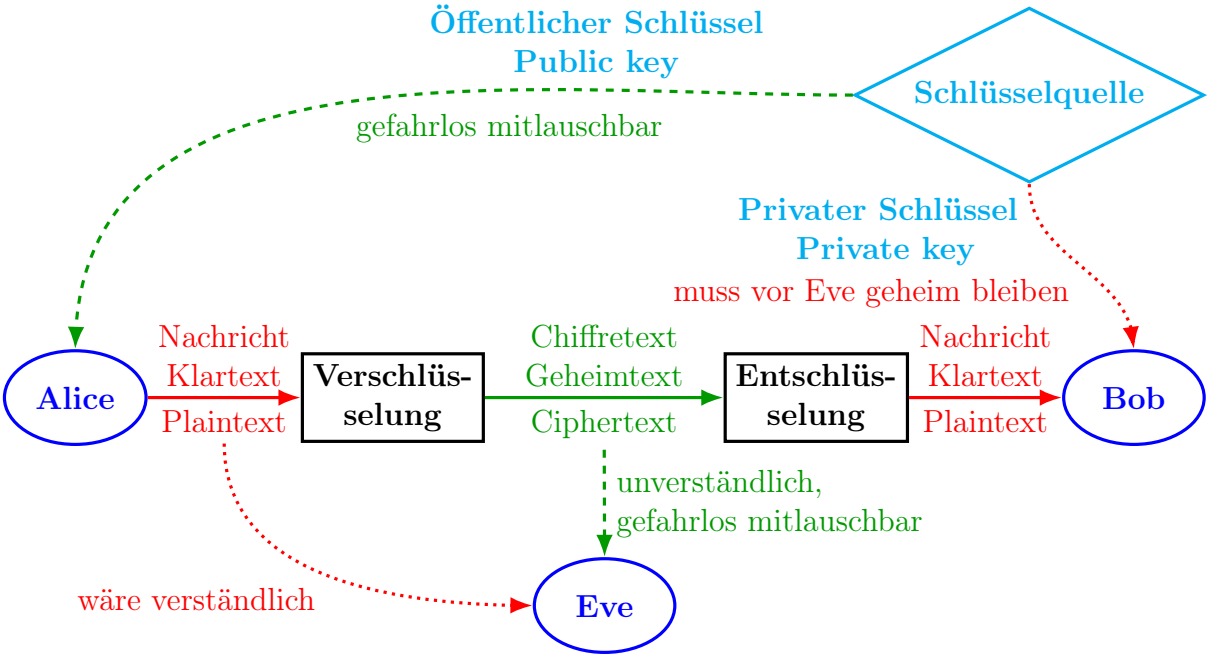

Abbildung 4: Wenn bei asymmetrischen Systemen Bob in der Lage ist, gute Schlüssel selbst zu erzeugen, kann er den Schlüssel für Alice gefahrlos an sie übermitteln.
Wenn Eve ihn mithört, ist sie zwar in der Lage, selbst Nachrichten zu verschlüsseln und an Bob zu senden, aber nicht, Alice's Nachricht zu entschlüsseln.

Neben den bisher erwähnten Unterschieden zwischen symmetrischen und asymmetrischen Kryptosystemen gibt es noch einen weiteren Aspekt: die Anzahl der benötigten Schlüssel, um mit vielen Personen zu kommunizieren.

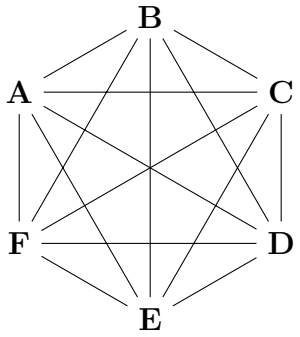

Abbildung 5: Die sechs Personen A bis F wollen paarweise miteinander privat kommunizieren können. Aufgrund der vielen Kombinationsmöglichkeiten sind dafür viele Schlüssel notwendig, wenn ein symmetrisches Kryptosystem eingesetzt wird.

Aufgabe 1:
Eine Gruppe aus sechs Personen möchte verschlüsselt kommunizieren. Dabei will jede Person in der Lage sein, mit jeder anderen Person privat zu kommunizieren, siehe Abbildung 5

1. Wie viele verschiedene Schlüssel müssen erzeugt werden, wenn ein symmetrisches Kryptosystem eingesetzt wird?

   .....

2. Wie viele Schlüssel muss eine Person sicher speichern? .....

3. Wie viele verschiedene Schlüssel müssen erzeugt werden, wenn ein asymmetrisches Kryptosystem eingesetzt wird?

   .....

4. Wie viele Schlüssel muss eine Person sicher speichern? .....

5. Beantworte dieselben Fragen, wenn es 100 Teilnehmer:innen gibt.

   .....

   .....

6. Beantworte dieselben Fragen, wenn es n Teilnehmer:innen gibt.

. . . . . .

. . . . . .

Denkt man an das Internet und seine Milliarden verbundenen Geräte und den Wunsch, dass jedes davon theoretisch mit jedem anderen kommunizieren können sollte, wird klar, dass man mit rein symmetrischen Verfahren nicht weit kommt. Möglich wäre nur, dass alle einer zentralen Stelle vertrauen, mit der sie sicher kommunizieren können und die die Schlüsselerzeugung und -verteilung übernimmt. Das widerspricht dem dezentralen Charakter des Internets und würde ein enormes Risiko in Hinblick auf Ausfallsicherheit und Überwachung darstellen!

2.2 Angriffsarten

In diesem Abschnitt geht es um die möglichen Angriffe auf Kryptosysteme, die teilweise danach eingeteilt werden, was der Angreifer weiß und kann. Es wird *immer* davon ausgegangen, dass der Angreifer das *verwendete Verfahren* kennt, weiß, wie Schlüssel aussehen etc.

Es bezeichnet k den verwendeten Schlüssel, m_i die i-te Klartext-Nachricht und c_i den zugehörigen Chiffretext.

Ciphertext-Only-Angriff

Eve hat nur die Ciphertexte c_i, $i = 1, \ldots, n$, zur Verfügung. Mögliche Ziele wären:

. . . . . .

. . . . . .

Von dieser (schwachen) Angriffsmöglichkeit ist immer auszugehen, weil Eve nur die Nachrichten abfangen und analysieren muss.

Welche Möglichkeiten einer Analyse gibt es?

. . . . . .

Known-Plaintext-Angriff

Eve hat neben den Ciphertexten c_i, $i = 1, \ldots, n$, auch die Klartexte m_i zur Verfügung, oder zumindest einige davon oder zumindest eine Ahnung davon. Mögliche Ziele wären:

. . . . . .

. . . . . .

Wieso sollte Eve die Klartexte kennen?

- Sie fängt mehrere Nachrichten ab, beobachtet das Verhalten des Empfängers und schließt so auf den Inhalt der Nachricht. („Finden Sie sich um 17:00 im Park ein.“, „Finden Sie sich um 12:00 im Park ein.“ und eventuell Nachrichtenteile, die sich verändern und solche, die es nicht tun!)
- Eve vermutet, dass das immer gleiche Ende eines Briefes „Beste Grüße“ heißen könnte.
- Manche Wehrmacht-Posten mussten immer zur gleichen Zeit einen Wetterbericht senden. Wegen des militärischen Schreibstils kam immer an derselben Stelle „Wetter“ vor. Zusätzlich kann Eve das tatsächliche Wetter beobachten...
- Ein deutscher Funker meldete regelmäßig, dass er nichts zu melden hat.
- Jede Nachricht der Wehrmacht endete auf „Heil Hitler“.

Chosen-Plaintext-Angriff

Der Angriff entspricht dem Known-Plaintext-Angriff, bloß dass Eve die Möglichkeit hat, die Klartexte m_i vorzugeben oder zumindest zu beeinflussen. Wie kann Eve Alice dazu bringen, bestimmte Klartexte zu verschlüsseln und an Bob zu senden?

- Ich inszeniere für einen feindlichen Agenten etwas, das er verschlüsselt berichtet.
- England „säte“ Gebiete in der Nordsee mit Minen (genannt „gardening“). Die daraufhin versendeten Nachrichten der Deutschen beinhalteten höchstwahrscheinlich Koordinaten der Gebiete.
- Die Amerikaner wussten, dass Japan „AF“ angreifen will und vermuteten, weil andere hawaiianische Inseln mit „A“ begannen, dass es sich um Midway handeln könnte. Sie fingierten eine Nachricht von dort wegen angeblichen Bedarfs an Vorräten. Die Japaner sendeten daraufhin eine Nachricht, dass „AF“ das Wasser ausginge.
- Ein Chosen-Plaintext-Angriff ist bei asymmetrischen Kryptosystemen immer möglich, weil der Schlüssel, der zum Verschlüsseln verwendet wird, ja öffentlich bekannt ist. So könnte Eve auch mehrere „wahrscheinliche Nachrichten“ (wie „ja“ oder „nein“) selbst verschlüsseln und einfach mit c_i vergleichen!

Brute-Force-Angriff

Dieser Angriff bezeichnet das bloße *Durchprobieren* aller möglichen Schlüssel. Dieser Angriff ist natürlich immer möglich. Ihn abzuwehren gelingt durch

.....

.....

Wörterbuchangriff

Wenn die Schlüssel nicht alle gleich wahrscheinlich sondern von Menschen ausgewählt sind (Passwörter), ist es vernünftig, wahrscheinliche Schlüssel zuerst zu probieren (Wörter aus Wörterbüchern, Kombinationen dieser, Falschschreibungen dieser oder überhaupt Wörter aus gestohlenen Passwortlisten). Früher wurden leichtsinnigerweise häufig berühmte Sprüche und Zitate als Schlüssel verwendet (wie: „In principio erat verbum“).

Replay-Angriff

Mallory hört die verschlüsselte Nachricht an Bob ab und sendet sie später, ohne sie zu verstehen, wieder an Bob. (Wenn Alice einen Aktienkauf autorisiert hat, muss sie nun das Vielfache bezahlen.)
Abwehren lassen sie Replay-Angriffe über

. . . . . .

Man-in-the-Middle-Angriff

Mallory schaltet sich zwischen Alice und Bob, so dass Alice denkt, sie würde mit Bob kommunizieren, und umgekehrt. In Wahrheit interagieren beide mit Mallory, die die Nachrichten von Alice an Bob, nachdem sie sie selbst gelesen hat, weiterleitet und umgekehrt. So merken Alice und Bob nicht, dass ihre Kommunikation abgehört wird, zumindest nicht, bis Mallory mit dem Angriff aufhört. (Somit kann Mallory die Kommunikation auch nicht unterbrechen, ohne aufzufliegen.)

Verkehrsflussanalysen

Eve bekommt lediglich mit, dass Alice und Bob kommunizieren. Und wie oft, wann, in welchem Rhythmus, in Kombination mit welchen Handlungen etc. Meist lässt sich daraus schon sehr viel ableiten, ohne den verschlüsselten Inhalt kennen zu müssen.[8]

Wenn es sich bei Alice und Bob um zwei CEOs handelt, kann die Kommunikation alleine schon bedeuten, dass eine Fusion bevor steht oder Preisabsprachen getroffen werden.

Wenn Eve mitbekommt, dass ihre Komplizin Alice mit den Bobbies redet, kann sie Schlüsse ziehen. . .

Eve bekommt mit, dass die CEO Alice mit dem Börsianer Bob spricht, welcher daraufhin viele Aktien ihres Unternehmens kauft. Der Verdacht des Insiderhandels drängt sich auf.

Abgewehrt werden können Verkehrsflussanalysen prinzipiell nicht. Allerhöchstens hilft das Versenden von Dummy-Nachrichten bei der Verschleierung.

[8] die Polizei erfährt auch sehr viel aus Handydaten-Auswertungen

Harvest-now-decrypt-later-Angriff

Eve hört jetzt die Nachrichten ab und knackt die Verschlüsselung später, wenn sie bessere Werkzeuge dafür hat. Deswegen muss man sich auch heute bereits mit Verschlüsselungen beschäftigen, die „Quantum-ready“ sind.

Seitenkanalangriffe

Darunter werden Angriffe verstanden, die nicht das Verfahren an sich sondern seine Benutzung, Umsetzung oder Implementierung mit scheinbar nutzloser Zusatzinformation angreifen.

Zeitangriff: Bereits 1996 wurde RSA geknackt, indem die Zeit für die Verschlüsselung gemessen wurde. Die Nachricht wurde variiert und aus der Dauer auf die Anzahl der Einsen im Exponenten (privater Schlüssel) geschlossen. Einige Tausend Messungen reichen, um RSA mit 1024 Bit zu brechen.
Abgewehrt werden kann der Angriff, indem zufällige Zahlen hinein- und später wieder herausgerechnet oder künstliche Zeitverzögerungen eingebaut werden.

Stromangriff: Ist die Verschlüsselung in Hardware implementiert, sind die Multiplikationen von den Quadrierungen im Square-and-Multiply-Algorithmus am Oszilloskop gut unterscheidbar. Ebenso Substitution und Permutation.
Durch parallel dazu laufende Dummy-Operationen lässt sich das vermeiden.

Elektromagnetische Strahlung: Elektronische Geräte emittieren elektromagnetische Strahlung, aus der sich auf die Operationen schließen lassen könnte. TEMPEST[9] ist der Deckname für ein Abhörverfahren bzw. für Spezifikationen, die ebendies verhindern sollen.
In Botschaftsgebäuden gibt es speziell abgeschirmte Räume (Faraday'sche Käfige). Auch bei Mils Electronic gab es so etwas, zusätzlich wurden die Signale in den Kabeln gefiltert, damit die Tastendrücke nicht dem Stromnetz aufgeprägt werden.

Eingebaute Hintertür: Es wird absichtlich eine Schwachstelle eingebaut. Verhindern lässt sich das teilweise durch offengelegte Verfahren und Quellcodes.
In der Operation „Trojan Shield“ wurden 800 Personen aus dem Organisierten Verbrechen festgenommen. Das FBI brachte gegen Straffreiheit den Entwickler einer App, die es Kriminellen ermöglichen sollte, verschlüsselt zu kommunizieren, und hinter der Taschenrechner-App versteckt war, dazu, eine Hintertür einzubauen. Eineinhalb Jahre hörten FBI und Europol den Verbrechern zu (12 000 Krypto-Handys, 27 Mio. Nachrichten).

Implementierungsfehler: Nicht alle Fehler fallen sofort auf, so funktioniert RSA auch mit schlechten Zahlen. Besonders anfällig sind Verfahren „ohne Umkehrung“ wie Hashfunktionen und Zufallszahlengeneratoren.

Das Problem des sicheren Löschens: Daten zu verschlüsseln ist völlig nutzlos, wenn ich die Originale nicht sicher löschen kann! Bedenke auch temporäre Dateien im Betriebssystem.

[9] Transient Electromagnetic Pulse Emanation Standard

Angriffe auf die BenutzerInnen:

Folter, Erpressung, Bestechung, ... Die meisten Geldautomatenbetrüger sind Insider (Bankangestellte, Techniker, ...). Ebenso kommt der Großteil der IT-Angriffe von innen! Abgemildert kann dieses Problem werden durch

.....

.....

Schwachstelle Mensch: Menschen nutzen aus *Faulheit* schwache Passwörter oder haben sie auf Post-its am Schreibtisch, chiffrieren nicht, übertreiben Sorgfalt (immer die gleiche Anrede mit allen Titeln, militärische Sprache[10]), bedienen Hard- und Software falsch etc.

Aufgabe 2:
Zur Selbstreflexion: Verwende ich Passwörter mehrfach? Sind meine Passwörter in Wörterbüchern zu finden oder anderweitig trivial (Stichwort: „12345“)? Habe ich sie notiert? Verschlüssle ich meine Mails/Dateien/...? Habe ich schon Unbekannten die Türe aufgehalten?

Verkleinert werden kann das Risiko Mensch durch

.....

.....

Ein häufiges Verständnisproblem ist, zu denken, dass Verfahren „doch irgendwie unsicher“ sind, die sich unter Einsatz aller Computer in Jahrzehnten brechen lassen.

- Angriffe kosten Geld!
- Sie werden nicht durchgeführt, wenn sie teurer sind als das, was sie preisgeben sollen!
- Es ist sinnlos sich vor Angriffen zu wappnen, wenn einfachere und billigere möglich sind.
  - Ich brauche keine Wohnungstüre mit Stahlkern, wenn meine Terrassentür aus einfachem Glas besteht.
  - Das Nachrüsten auf Iris- und Fingerabdruckscanner bei Türen mit Katzenklappe ist zwecklos.
  - 10-stellige PINs sind kontraproduktiv, weil die BenutzerInnen sie sich nicht merken können und auf die Karte schreiben.

[10] Die Engländer „knackten“ mit Hilfe der *Turing-Bombe* die Enigma-Verschlüsselung der Wehrmacht, nicht aber die der Deutschen Reichsbahn. Die Angriffe beruhten auf wahrscheinlichen Wörtern im Klartext (vgl. Knwon-Plaintext), die vom Militär oft benutzt wurden, während die Eisenbahnersprache „zu schwierig“ war.

- Es gilt das Rollenspiele-Zitat: „If you find yourself in the company of a halfling and an ill-tempered dragon, remember that you do not have to outrun the dragon; you simply have to outrun the halfling.“
  Ein Computersystem muss nicht absolut sicher sein, sondern so sicher, dass sein Wert die Kosten eines Angriffes nicht überschreitet. Eigentlich muss sich der Angriff nur weniger lohnen als der auf jemand anderen.

- Die einfachsten, billigsten und effektivsten Angriffe sind **Social engineering**. Dort gilt es zuerst anzusetzen!

3 Mathematische Grundlagen der Kryptographie

3.1 Modulo

Es sei $a \in \mathbb{Z}$ eine ganze Zahl und $m \in \{2, 3, 4, \ldots\}$ der sogenannte *Modul*. Wenn man nun a ganzzahlig durch m dividiert, erhält man den ganzzahligen Quotienten q und den Rest $r \in \{0, 1, 2, \ldots, m-1\}$, so dass gilt:

$$a = q \cdot m + r$$

Für den Rest dieser Ganzzahldivision schreibt man

$$r = a \bmod m \quad (\text{„}a \text{ modulo } m\text{“}).$$

Beachte:

- Es gibt nur m verschiedene Reste bei ganzzahliger Division durch m.
- Der Rest r ist nicht-negativ! Das gilt auch bei $a < 0$, siehe Aufgabe 3.

Aufgabe 3:
Berechne den Rest und schreibe die Gleichung der Form $a = q \cdot m + r$ an:

(a) $13 \bmod 3 =$ .....

(b) $42 \bmod 5 =$ .....

(c) $42 \bmod 9 =$ .....

(d) $-13 \bmod 3 =$ .....

(e) $-42 \bmod 9 =$ .....

(f) $-36 \bmod 9 =$ .....

Dieses mod ist eine **Rechenoperation**. Sie erhält zwei Argumente a und m und liefert eine Zahl r als Ergebnis. So ein Ergebnis lässt sich mit dem Taschenrechner ausrechnen.

3.2 Kongruenz modulo m

Wenn zwei ganze Zahlen $a, b \in \mathbb{Z}$ nach Division durch m den gleichen Rest haben, nennt man sie **kongruent modulo** m und schreibt:

$$a \equiv b \pmod{m}$$

Beispielsweise ist

$$72 \equiv 12 \pmod{10}.$$

Aufgabe 4:
Finde andere Moduln m, sodass

$$72 \equiv 12 \pmod{m}.$$

.....

Wir werden sehen, dass man dann a und b in gewissem Sinne als gleich ansehen kann. Deshalb verwendet man ein Symbol, das an Gleichheit erinnert.

Bei der Kongruenz modulo m handelt es sich nicht um eine Rechenoperation sondern um eine **Aussage**, dass sich nämlich a und b nur um ein Vielfaches des Moduls m unterscheiden:

$$a \equiv b \pmod{m} \qquad \Leftrightarrow \qquad a - b = k \cdot m,\ k \in \mathbb{Z}$$

Aufgabe 5:
Welche der folgenden Ausdrücke sind wahr, welche falsch und bei welchen macht diese Fragestellung keinen Sinn?

(a) $25 \equiv 0 \pmod{5}$ .....

(b) $25 \equiv 20 \pmod{5}$ .....

(c) $25 \equiv 20 \pmod{10}$ .....

(d) $25 \bmod 5$ .....

Aufgabe 6:
Berechne, falls möglich:

(a) $25 \equiv 10$ .....

(b) $25 \bmod 10$ .....

Aufgabe 7:
Entscheide, ob im Fall $a \equiv b \pmod{m}$ die folgende Aussage stimmt:

$$a \bmod m = b \bmod m \qquad .....$$

Muss/darf das „=“ durch „≡“ ersetzt werden? .....
Finde Beispiele dafür, dass auf keiner Seite der Gleichung „mod m“ weggelassen werden darf.

.....

3.3 Uhrenarithmetik

Um die Rechenregeln im Zusammenhang mit modulo kennenzulernen, rechnen wir mit Uhrzeiten, also modulo 24.

(a) Ein Fertigungsprozess startet um 17 Uhr und dauert 21 h. Um welche Uhrzeit endet er?

$$17 + 21 = 38 \equiv 14 \pmod{24}$$

Er endet um 14 Uhr.

(b) Ein Fertigungsprozess startet um Mitternacht und dauert 38 h für den ersten Arbeitschritt und dann noch einmal 40 h für den zweiten. Man kann rechnen:

$$38 + 40 = 78 \equiv 6 \pmod{24}$$

Alternativ wäre auch möglich gewesen:

$$\begin{aligned} 38 + 40 &\equiv 14 + 40 \pmod{24} \\ &\equiv 14 + 16 \pmod{24} \\ &\equiv 30 \pmod{24} \\ &\equiv 6 \pmod{24} \end{aligned}$$

Wir erkennen: **Bei der Addition modulo m ist es egal, ob vor dem Addieren oder in Zwischenschritten um m reduziert wird.**

(c) Ein Arbeitsschritt dauert 22 h und muss mit Start um Mitternacht 7-mal durchgeführt werden. Um wie viel Uhr ist er zu Ende? Neben

$$22 \cdot 7 = 154 \equiv 10 \pmod{24}$$

wäre auch möglich:

$$22 \cdot 7 \equiv (-2) \cdot 7 \pmod{24} \equiv -14 \pmod{24} \equiv 10 \pmod{24}$$

Wir erkennen: **Bei der Multiplikation modulo m ist es egal, ob vor dem Multiplizieren oder in Zwischenschritten um m reduziert wird.**

(d) Daraus folgt, dass **auch beim Potenzieren die Basis (nicht der Exponent!) im Voraus um m reduziert werden darf.** Berechne 3^5 mod 2 auf beide Arten:

. . . . . .

Aufgabe 8:
Berechne:

(a) $(12 + 20) \bmod 3 =$ . . . . . .

(b) $(81 - 40) \bmod 7 =$ .....

(c) $(23 \cdot 15) \bmod 4 =$ .....

(d) $(15 \cdot 16 \cdot 17) \bmod 5 =$ .....

(e) $(18 + 34 \cdot 23) \bmod 8 =$ .....

(f) $17^8 \bmod 7 =$ .....
(Übrigens ist $17^8 = 6\,975\,757\,441$.)

Aufgabe 9:
Berechne in $\mathbb{C}$:

(a) $j^2 =$ .....

(b) $j^3 =$ .....

(c) $j^4 =$ .....

(d) $j^{10} =$ .....

(e) $j^{2024} =$ .....

(f) $j^{9876543210} =$ .....

Aufgabe 10:
Berechne: $365 \bmod 7 =$ .....
Der 24.12.2024 ist ein Dienstag. Bestimme den Wochentag der folgenden Daten:

(a) 24.12.2025: .....

(b) 24.12.2026: .....

(c) 24.12.2027: .....

(d) 24.12.2028: .....

Aufgabe 11:
Berechne:

(a) $(100 + 823) \bmod 5 =$ .....

(b) $(12 + 5 \cdot 18) \bmod 7 =$ .....

(c) $(23 \cdot 18) \bmod 5 =$ .....

(d) $(19 \cdot 37 \cdot 22) \bmod 7 =$ .....

(e) $(41 \cdot 23 \cdot 25) \bmod 9 =$ .....

(f) $(49 \cdot 78 \cdot 25) \bmod 11 =$ .....

(g) $(181 \cdot 208 \cdot 54) \bmod 25 =$ .....

(h) $(34 \cdot 68 \cdot 108) \bmod 12 =$ .....

(i) $(31 \cdot 42 + 53 \cdot 19) \bmod 9 =$ .....

(j) $14^3 \bmod 13 =$ .....

(k) $2^{20} \bmod 3 =$ .....

(l) $(212 \cdot 31^2) \bmod 21 =$ .....

(m) $(19^4 + 14 \cdot 25^5) \bmod 8 =$ .....

Aufgabe 12:
Für die Zahlen a, b, m gilt: $a \equiv b \pmod{m}$. Begründe, ob die folgenden Aussagen wahr sind:

(a) Man darf auf beiden Seiten der Kongruenz 5 addieren, und die Kongruenz stimmt weiter, also

$$(a + 5) \equiv (b + 5) \pmod{m}. \quad \ldots\ldots$$

(b) Man darf auf beiden Seiten der Kongruenz 2 subtrahieren, und die Kongruenz stimmt weiter, also

$$(a - 2) \equiv (b - 2) \pmod{m}. \quad \ldots\ldots$$

(c) Man darf auf beiden Seiten der Kongruenz mit 3 multiplizieren, und die Kongruenz stimmt weiter, also

$$(3 \cdot a) \equiv (3 \cdot b) \pmod{m}. \quad \ldots\ldots$$

(d) Man darf auf beiden Seiten der Kongruenz durch 4 dividieren, und die Kongruenz stimmt weiter, also aus

$$8 \equiv 28 \pmod{20}$$

folgt

$$2 \equiv 7 \pmod{20}. \quad \ldots\ldots$$

Die Aufgabe 12 (d) zeigt schon, dass man beim Dividieren vorsichtig sein muss.

3.4 Der Restklassenring $\mathbb{Z}_m$

In der „Uhrenarithmetik“ im vorigen Abschnitt haben wir mit **ganzen Zahlen** gerechnet und **am Ende** durch die Operation „mod 24“ den Rest berechnet. Dieser Rest war selbst eine **ganze Zahl**. (Der Zweck des Kapitels war, zu sehen, dass man die Operation auch zwischendurch durchführen kann. Man rechnet dann öfter modulo, hat aber kleinere Zahlen zu verarbeiten.)

Man kann die Aufgaben aber auch anders wahrnehmen: Es gibt eigentlich nur 24 verschiedene Uhrzeiten, weil für uns 2 Uhr und 26 Uhr dasselbe sind. Wir fassen alle Zahlen, die nach Division durch 24 denselben Rest ergeben, zu einer **Restklasse modulo 24** zusammen:

$$\{\ldots, -22, 2, 26, 50, \ldots\}$$

Als Kurzschreibweise für diese (unendlich vielen) Zahlen nehmen wir eine beliebige Zahl als **Repräsentanten**:

$$\overline{2} = \{\ldots, -22, 2, 26, 50, \ldots\} = \overline{26} = \ldots$$

Durch den Querstrich wird markiert, dass es sich bei $\overline{2}$ nicht um die ganze Zahl 2 sondern die Restklasse **aller** ganzen Zahlen handelt, die nach Division durch 24 den Rest 2 hinterlassen.

Wie aber rechnet man mit diesen Restklassen?
Man rechnet stattdessen einfach mit den Repräsentanten.

Aufgabe 13:
Es sind in $\mathbb{Z}_{24}$ die Restklassen

$$\{\ldots, -19, 5, 29, \ldots\} \quad \text{und} \quad \{\ldots, -1, 23, 47, \ldots\}$$

zu addieren. Wähle je verschiedene Repräsentanten aus und addiere sie. Welche Menge ergibt sich?

. . . . . .

Wie hätte man also kürzer rechnen können?

$$\overline{5} + \overline{23} = \overline{5 + 23} = \overline{28} = \overline{4} \qquad \text{oder:}$$
$$\overline{5} + \overline{23} = \overline{5} + \overline{-1} = \overline{5 + (-1)} = \overline{4}$$

Aufgabe 14:
Probiere dasselbe für die Multiplikation am Beispiel $\overline{13} \cdot \overline{25}$ aus:

. . . . . .

oder:

. . . . . .

Aufgrund der Rechenregeln, die wir im Abschnitt 3.3 kennengelernt haben („Es ist egal, ob man zwischendurch modulo rechnet.“), können wir also beim Rechnen mit Restklassen stattdessen einfach mit beliebigen (!) Repräsentanten rechnen.

Der Rechenaufwand kann jedoch je nach Wahl des Repräsentanten extrem unterschiedlich sein (Aufgabe 14 zeigt das in harmloser Weise), deswegen ist es wichtig, die Repräsentanten geschickt zu wählen und immer so früh wie möglich „klein zu machen“.

3.4.1 Die Restklassenringe $\mathbb{Z}_5$ und $\mathbb{Z}_6$

Wir betrachten als konkretes Beispiel den Restklassenring $\mathbb{Z}_5$. Er enthält die Elemente

.....

Es gibt also nur ..... „Zahlen“. Damit gibt es auch nur je ..... verschiedene Additionen und Multiplikationen, die man tabellarisch auflisten kann.

Aufgabe 15:
Erstelle die Additions- und die Multiplikationstabelle für $\mathbb{Z}_5$.

$+$	$\overline{0}$	$\overline{1}$	$\overline{2}$	$\overline{3}$	$\overline{4}$
$\overline{0}$					
$\overline{1}$					
$\overline{2}$					
$\overline{3}$					
$\overline{4}$					

$\cdot$	$\overline{0}$	$\overline{1}$	$\overline{2}$	$\overline{3}$	$\overline{4}$
$\overline{0}$					
$\overline{1}$					
$\overline{2}$					
$\overline{3}$					
$\overline{4}$					

Da es so wenige Rechnungen gibt, kann man, statt zu rechnen, auch einfach dieses „kleine (!) Einmaleins“ als Nachschlagwerk benutzen.

Computerprozessoren rechnen genaugenommen nicht mit ganzen Zahlen sondern Restklassen. Beispielsweise bezeichnet der 16 bit-Datentyp `short integer` die Zahlen bzw. Restklassen

.....
Lässt man in einer Endlosschleife eine solche Variable schrittweise um 1 erhöhen, kommt nach

..... die Zahl .....

Aufgabe 16:
Erstelle die Additions- und die Multiplikationstabelle für $\mathbb{Z}_6$.

$+$	$\overline{0}$	$\overline{1}$	$\overline{2}$	$\overline{3}$	$\overline{4}$	$\overline{5}$
$\overline{0}$						
$\overline{1}$						
$\overline{2}$						
$\overline{3}$						
$\overline{4}$						
$\overline{5}$						

$\cdot$	$\overline{0}$	$\overline{1}$	$\overline{2}$	$\overline{3}$	$\overline{4}$	$\overline{5}$
$\overline{0}$						
$\overline{1}$						
$\overline{2}$						
$\overline{3}$						
$\overline{4}$						
$\overline{5}$						

Aufgabe 17:
Beim Ausfüllen der Additions- und Multiplikationstabellen sind ja sicher schon einige Regelmäßigkeiten und Muster aufgefallen. Suche nach weiteren Regelmäßigkeiten, die sowohl in $\mathbb{Z}_5$ als auch $\mathbb{Z}_6$ auftreten. Suche auch nach Unterschieden!

3.4.2 Allgemeine Eigenschaften der Restklassenringe, algebraische Strukturen

Die Restklassenringe $\mathbb{Z}_m$ haben folgende Eigenschaften:

($G1$) Die Addition ist **assoziativ**, d. h. $(a + b) + c = a + (b + c)$.

($G2$) Es gibt ein **neutrales Element der Addition**, nämlich $\overline{0}$. Wenn man es zu irgendeinem Element addiert, ändert sich dieses nicht: $a + \overline{0} = \overline{0} + a = a$
(Die erste Zeile der Additionstabelle stimmt mit den Spaltenköpfen überein.)

($G3$) Jedes Element a hat ein **additiv Inverses**, nämlich $-a$. Wenn man es zu a addiert, erhält man das neutrale Element: $a + (-a) = \overline{0}$
(In jeder Zeile der Additionstabelle kommt einmal $\overline{0}$ vor.
Z. B. in $\mathbb{Z}_5$ gilt $\overline{2}+\overline{3} = \overline{0}$, also kann man auch zu $\overline{2}$ etwas addieren, sodass das neutrale Element $\overline{0}$ entsteht. Da $\overline{3} = \overline{-2}$ überrascht das nicht.)

(GK) Die Addition ist **kommutativ**, d. h. $a + b = b + a$.
(Die Additionstabelle ist spiegelsymmetrisch bezüglich der Diagonalen.)

($R1$) Die Multiplikation ist **assoziativ**, d. h. $(a \cdot b) \cdot c = a \cdot (b \cdot c)$.

($R2$) Es gibt ein **neutrales Element der Multiplikation**, nämlich $\overline{1}$. Wenn man es zu irgendeinem Element multipliziert, ändert sich dieses nicht: $a \cdot \overline{1} = \overline{1} \cdot a = a$
(Die zweite Zeile der Multiplikationstabelle stimmt mit den Spaltenköpfen überein.)

($R3$) Es gilt das **Distributivgesetz**, d. h. $(a+b)\cdot c = a\cdot c+b\cdot c$ und $c\cdot(a+b) = c\cdot a+c\cdot b$.

(RK) Die Multiplikation ist **kommutativ**, d. h. $a \cdot b = b \cdot a$.
(Die Multiplikationstabelle ist spiegelsymmetrisch bezüglich der Diagonalen.)

Nur in $\mathbb{Z}_5$, nicht aber in $\mathbb{Z}_6$, gilt außerdem:

(K) Jedes Element a außer $\overline{0}$ hat ein **multiplikativ Inverses**, bezeichnet als a^{-1}. Wenn man es zu a multipliziert, erhält man das neutrale Element: $a \cdot a^{-1} = a^{-1} \cdot a = \overline{1}$
(In jeder Zeile der Multiplikationstabelle (außer der $\overline{0}$er) kommt einmal $\overline{1}$ vor.
Z. B. in $\mathbb{Z}_5$ gilt $\overline{4} \cdot \overline{4} = \overline{1}$, also kann man auch $\overline{4}$ mit etwas multiplizieren, sodass das neutrale Element $\overline{1}$ entsteht. Da $\overline{4} = \overline{-1}$ überrascht $\overline{-1} \cdot \overline{-1} = \overline{1}$ nicht.)

Ein paar Bemerkungen:

1. Ganz allgemein nennt man eine Menge von Objekten mit einer Rechenoperation, die ($G1$), ($G2$) und ($G3$) erfüllt, eine **Gruppe**. Ist zusätzlich (GK) erfüllt, eine **kommutative Gruppe**.

2. Gelten für eine Menge von Objekten und zwei Rechenoperationen zusätzlich noch ($R1$), ($R2$) und ($R3$), nennt man sie einen **Ring**. Ist auch noch (RK) erfüllt, einen **kommutativen Ring**.

3. Gilt für einen kommutativen Ring außerdem noch (K), nennt man ihn einen **Körper**.

Aufgabe 18:
Bestimme die algebraische Struktur der folgenden Mengen:

(a) $\mathbb{N}$ .....

(b) $\mathbb{Z}$ .....

(c) $\mathbb{Q}$ .....

(d) $\mathbb{R}$ .....

(e) $\mathbb{C}$ .....

(f) Drehungen in der Ebene .....

(g) Drehungen im Raum .....

(h) Vektoren im $\mathbb{R}^3$ .....

(i) Matrizen .....

(j) $\mathbb{Z}_m$ .....

Wir haben die Addition und Multiplikation in $\mathbb{Z}_m$ über das addieren bzw. multiplizieren von Repräsentanten definiert. Was aber lässt sich als Subtraktion und Division verstehen?

Unter „a subtrahieren“ verstehen wir, dass das additiv Inverse von a addiert wird. Das kennt man auch aus der Mittelschule:

$$7 - 3 = 7 + (-3)$$

Wir berechnen in $\mathbb{Z}_5$ auf zwei Arten

$\overline{3} - \overline{4} = \overline{3} + \left(-\overline{4}\right) = \overline{3} +$ .....

$\overline{3} - \overline{4} = \overline{3} + \overline{-4} =$ .....

Unter „dividieren durch a“ verstehen wir, dass mit dem Inversen von a multipliziert wird. Das ist ebenfalls bekannt aus $\mathbb{Q}$:

$$15 : 3 = \frac{15}{3} = 15 \cdot \frac{1}{3} = 15 \cdot 3^{-1},$$

wobei 3^{-1} das Inverse von 3 ist. (Es ist jene Zahl, die mit 3 multipliziert 1 ergibt.) Wir berechnen jetzt in $\mathbb{Z}_5$

„$\overline{4} : \overline{2}$“$= \overline{4} \cdot \overline{2}^{-1} = \overline{4} \cdot$ .....

Probe: .....

Aufgabe 19:
Berechne in $\mathbb{Z}_5$:

(a) „$\overline{3} : \overline{4}$“= .....

Probe: .....

(b) „$\overline{1} : \overline{3}$“= .....

Probe: .....

(c) „$\overline{2} : \overline{0}$“= .....

Probe: .....

Aufgabe 20:
Berechne in $\mathbb{Z}_6$:

(a) $\overline{2} - \overline{4} =$ .....

Probe: .....

(b) „$\overline{3} : \overline{5}$“= .....

Probe: .....

(c) „$\overline{1} : \overline{5}$“= .....

Probe: .....

(d) „$\overline{3} : \overline{4}$“= .....

Probe: .....

(e) „$\overline{2} : \overline{4}$“= .....

Probe: .....

.....

In $\mathbb{Z}_6$ gibt es **Nullteiler**, nämlich $\overline{2}, \overline{3}, \overline{4}$. Das sind Elemente, die selbst nicht $\overline{0}$ sind, aber trotzdem mit einem anderen Element multipliziert $\overline{0}$ ergeben.

$$\overline{2} \cdot \overline{3} = \overline{0} \qquad \text{und} \qquad \overline{4} \cdot \overline{3} = \overline{0}$$

Der Grund dafür ist, dass $2 \cdot 3 = 6$ bzw. $4 \cdot 3 = 2 \cdot 6$ Vielfache des Moduls 6 sind. Wenn also der Modul m faktorisiert werden kann (wie z. B. $6 = 2 \cdot 3$), ...

- ... dann gibt es Nullteiler,
- ... dann gibt es Elemente außer $\overline{0}$, die nicht invertierbar sind,
- ... dann gilt in $\mathbb{Z}_m$ der Produkt-Null-Satz nicht,

$$a \cdot b = \overline{0} \quad \not\Rightarrow \quad a = \overline{0} \text{ oder } b = \overline{0}$$

- ... dann ist $\mathbb{Z}_m$ .....

Wenn also der Modul m faktorisiert werden kann, dann ist $\mathbb{Z}_m$ nur ein kommutativer Ring, aber kein Körper. Das ist dann der Fall, wenn m

.....

Wenn umgekehrt m ..... ist, dann ...

- ... lässt sich m **nicht faktorisieren**,
- ... gibt es **keine Nullteiler** in $\mathbb{Z}_m$,
- ... gilt in $\mathbb{Z}_m$ der Produkt-Null-Satz,
- ... ist $\mathbb{Z}_m$ ein Körper,
- ... steht in jeder Zeile der Multiplikationstabelle (außer der $\overline{0}$er-Zeile) an irgendeiner Stelle $\overline{1}$,
- ...„kommt man von jedem Element (außer $\overline{0}$) überall hin", das heißt, es gibt für jedes $a \neq \overline{0}$ und jedes c irgendein b, sodass $a \cdot b = c$,
- ... stehen in jeder Zeile der Multiplikationstabelle (außer der $\overline{0}$er) alle Elemente einmal.

Aufgabe 21:
Begründe, dass in einer Zeile, in der $\overline{1}$ steht, jede Zahl einmal vorkommen muss.

.....

.....

.....

Aufgabe 22:
Markiere in den Multiplikationstabellen von Aufgabe 15 und Aufgabe 16 farblich die Einträge $\overline{0}$ (neutrales Element der Addition) und $\overline{1}$ (neutrales Element der Multiplikation). Man sieht:

(a) Kommt in einer Zeile einmal $\overline{1}$, dann kein zweites Mal, weil

.....

(b) Kommt in einer Zeile außer in der ersten Spalte noch einmal $\overline{0}$ vor, dann müssen die Einträge wiederholt auftreten, weil

.....

Aufgabe 23:
Erstelle die Multiplikationstabelle für $\mathbb{Z}_{12}$. (Additionstabellen sind langweilig, sie sehen alle gleich aus. Dort passiert nichts Überraschendes.)

$\cdot$	$\overline{0}$	$\overline{1}$	$\overline{2}$	$\overline{3}$	$\overline{4}$	$\overline{5}$	$\overline{6}$	$\overline{7}$	$\overline{8}$	$\overline{9}$	$\overline{10}$	$\overline{11}$
$\overline{0}$												
$\overline{1}$												
$\overline{2}$												
$\overline{3}$												
$\overline{4}$												
$\overline{5}$												
$\overline{6}$												
$\overline{7}$												
$\overline{8}$												
$\overline{9}$												
$\overline{10}$												
$\overline{11}$												

Liste alle invertierbaren Elemente auf und gib ihr Inverses an:

. . . . . .

. . . . . .

Aufgabe 24:
Färbe die Multiplikationstabelle für $\mathbb{Z}_{13}$, diskutiere die Eigenschaften und vergleiche mit $\mathbb{Z}_5$, $\mathbb{Z}_6$ und $\mathbb{Z}_{12}$.

$\cdot$	$\overline{0}$	$\overline{1}$	$\overline{2}$	$\overline{3}$	$\overline{4}$	$\overline{5}$	$\overline{6}$	$\overline{7}$	$\overline{8}$	$\overline{9}$	$\overline{10}$	$\overline{11}$	$\overline{12}$
$\overline{0}$													
$\overline{1}$													
$\overline{2}$													
$\overline{3}$													
$\overline{4}$													
$\overline{5}$													
$\overline{6}$													
$\overline{7}$													
$\overline{8}$													
$\overline{9}$													
$\overline{10}$													
$\overline{11}$													
$\overline{12}$													

Gib zwei invertierbare Elemente und ihre Inversen an:

.....

Aufgabe 25:
In welchem der beiden Restklassenringe $\mathbb{Z}_{13}$ und $\mathbb{Z}_{12}$ ist die Rechnung $\overline{8} \cdot \overline{4}^{-1}$ durchführbar?

in $\mathbb{Z}_{13}$: .....

in $\mathbb{Z}_{12}$: .....
Wieso ist das Ergebnis nicht einfach $\overline{2}$, wo doch $8 \cdot 4^{-1} = 8 \cdot \frac{1}{4} = \frac{8}{4} = 2$ ist?

.....

.....

Aufgabe 26:
Berechne in $\mathbb{Z}_{15}$:

(a) $\overline{11} + \overline{7} =$ .....

(b) $\overline{11} - \overline{7} =$ .....

(c) $\overline{7} - \overline{11} =$ .....

(d) $\overline{14} + \overline{14} =$ .....

(e) $\overline{11} - \overline{7} =$ .....

(f) $\overline{11} \cdot \overline{7} =$ .....

(g) $\overline{11} \cdot \overline{11} =$ .....

(h) $\overline{14} \cdot \overline{14} =$ .....

(i) $\overline{10} \cdot \overline{6} =$ .....

(j) $\overline{14}^{51} =$ .....

(k) $\overline{5} \cdot \overline{7}^{-1} =$ .....

(l) $\overline{7} \cdot \overline{5}^{-1} =$ .....

Aufgabe 27:
Hat jedes Element von $\mathbb{Z}_7$ ein multiplikativ Inverses? Erstelle die Multiplikationstabelle für $\mathbb{Z}_7$, färbe sie und kontrolliere deine Vermutung.

$\cdot$	$\overline{0}$	$\overline{1}$	$\overline{2}$	$\overline{3}$	$\overline{4}$	$\overline{5}$	$\overline{6}$
$\overline{0}$							
$\overline{1}$							
$\overline{2}$							
$\overline{3}$							
$\overline{4}$							
$\overline{5}$							
$\overline{6}$							

Aufgabe 28: (Verschlüsselung durch modulare Addition)
Wir codieren die Buchstaben durch Zahlen ($A \to 0$, $B \to 1, \ldots, Z \to 25$). Damit lassen sich die Buchstaben der Klartexts als $m \in \mathbb{Z}_{26}$ auffassen. Wir wählen einen Schlüssel $k \in \mathbb{Z}_{26}$. Verschlüsselt wird durch modulare Addition:

$$c = m + k$$

Führe das für den Klartext „SALVE“ und einen selbst gewählten Schlüssel durch.

Chiffretext: .....

Wie heißt dieses Verfahren? .....

Wie kann dieser Chiffretext entschlüsselt werden?

durch .....

bzw. .....
Wie viele verschiedene Schlüssel gibt es in diesem Verfahren? Wie sicher ist es?

.....

.....

.....

Aufgabe 29: (Verschlüsselung durch modulare Multiplikation)
Wir codieren die Buchstaben durch Zahlen ($A \to 0$, $B \to 1, \ldots, Z \to 25$). Damit lassen sich die Buchstaben der Klartexts als $m \in \mathbb{Z}_{26}$ auffassen. Wir wählen einen Schlüssel $k \in \mathbb{Z}_{26}$.

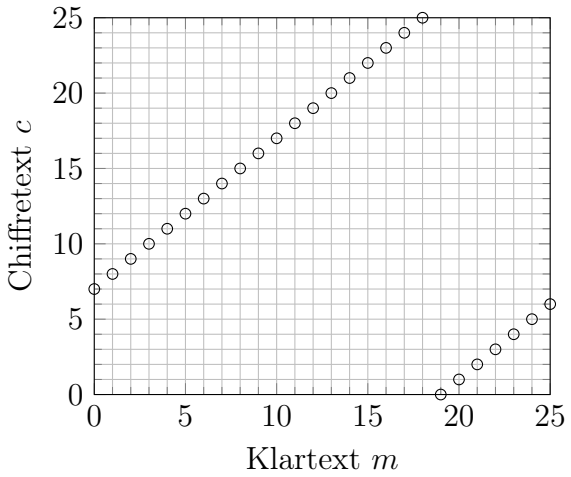

Abbildung 6: Visualisierung der Verschlüsselung durch modulare Addition mit Schlüssel $k = \overline{5}$. Die Regelmäßigkeit ist auf den ersten Blick erkennbar. Der Buchstabe F folgt auf E, und der Chiffretext von F folgt auf den von E. Die Verschlüsselung hat also nicht viel „Chaos“ gestiftet.
Gezeichnet ist der Graph der Funktion $c = \big(m + k\big) \bmod 26$, was nur die „abgeschnittene“ Version von $c = m + k$ ist, also einer **linearen** Funktion mit Steigung 1 und reiner Verschiebung um k.

Verschlüsselt wird durch modulare Multiplikation:

$$c = m \cdot k$$

Führe das für den Klartext „ACHTUNG“ und den Schlüssel $k = \overline{13}$ durch. Was fällt dabei auf? Wie kann entschlüsselt werden?

. . . . . .

. . . . . .

. . . . . .

Wie sieht die Situation mit dem Schlüssel $\overline{5}$ aus?

. . . . . .

. . . . . .
Prüfe das für die ersten drei Buchstaben nach:

. . . . . .

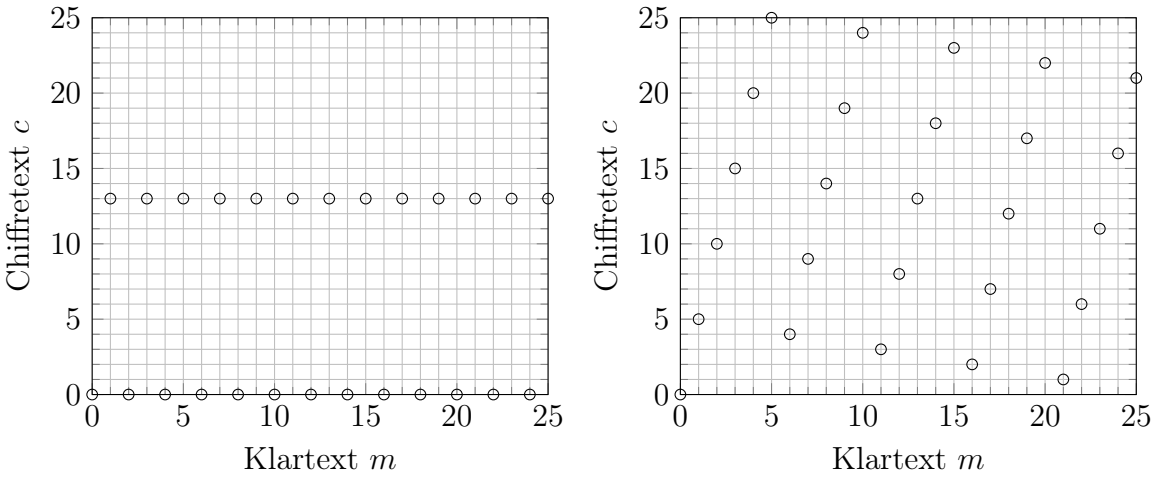

Abbildung 7: Visualisierung der Verschlüsselung durch modulare Multiplikation mit den Schlüsseln $k = \overline{13}$ (links) und $k = \overline{5}$ (rechts).
Links sticht sofort ins Auge, dass manche Zahlen ($\overline{0}$ und $\overline{13}$) als Chiffretext mehrfach verwendet werden. Damit ist die Verschlüsselung nicht umkehrbar.
In beiden Bildern ist leicht eine Regelmäßigkeit erkennbar, es wurde also nicht stark „durcheinandergewürfelt“. Gezeichnet ist schließlich der Graph der Funktion $c = \big(m \cdot k\big) \bmod 26$, was nur die „abgeschnittene“ Version von $c = m \cdot k$ ist, also einer **linearen** Funktion mit Steigung k.
Man sieht hier, welche Aufgabe die modulo-Bildung hat: Durch das Abschneiden kommt (scheinbare) Unregelmäßigkeit ins Spiel. Es ist zwei Chiffretext-Buchstaben nicht mehr auf Anhieb anzusehen, wessen zugehöriger Klartext-buchstabe früher im Alphabet kommt.
Im nächsten Abschnitt werden wir hingegen lernen, dass die Umkehrung der Multiplikation leicht zu berechnen ist, die Unregelmäßigkeit also nur auf den allerersten Blick besteht.

. . . . . .

. . . . . .

Aufgabe 30: (Verschlüsselung durch modulare Multiplikation)
Wir codieren die Buchstaben durch Zahlen ($A \to 0$, $B \to 1$, …, $Z \to 25$). Damit lassen sich die Buchstaben der Klartexts als $m \in \mathbb{Z}_{26}$ auffassen.

(a) Du und Bob haben im Vorfeld $k = \overline{7} \in \mathbb{Z}_{26}$ als Schlüssel vereinbart. Verschlüsselt wird durch modulare Multiplikation:

$$c = m \cdot k$$

Du bekommst von Bob die Nachricht „GUNV“ geschickt. Bestimme den Klartext.

. . . . . .

. . . . . .

(b) Du und Bob haben im Vorfeld vereinbart, dass wie folgt verschlüsselt wird:

$$c = \overline{11} \cdot m + \overline{2}$$

Du bekommst von Bob die Nachricht „SDUHPU“ geschickt. Bestimme den Klartext.

. . . . . .

. . . . . .

Als Nächstes widmen wir uns den folgenden Fragen:

- Gibt es eine Möglichkeit herauszufinden, ob ein Element in $\mathbb{Z}_m$ invertierbar ist, ohne die gesamte Tabellenzeile zu berechnen?
- Gibt es eine Möglichkeit, von invertierbaren Elementen das Inverse zu finden, ohne die gesamte Tabellenzeile (bis zum Auftreten von $\overline{1}$) zu berechnen?
- Gibt es eine Möglichkeit, die Anzahl der invertierbaren Elemente in $\mathbb{Z}_m$ herauszufinden, ohne die gesamte Tabelle zu berechnen?

3.5 Die Berechnung des ggT

Wir möchten den **größten gemeinsamen Teiler** zweier natürlicher Zahlen a und b berechnen. Es gibt dazu die folgenden Möglichkeiten, die anhand des Beispiels $a = 2079$ und $b = 735$ vorgeführt werden:

3.5.1 Primfaktorzerlegung beider Zahlen

Diese Methode ist aus der Mittelschule bekannt. Sie hilft beim Verständnis – auch in Hinblick auf Terme –, ist aber bei großen Zahlen **extrem aufwändig**. Man zerlegt beide Zahlen in ihre Primfaktoren. Der ggT wird dann aus allen **gemeinsamen** Primfaktoren gebildet. (Taucht ein Primfaktor mehrfach auf, wird er nur so oft verwendet, wie er in beiden Zahlen vorkommt.):

$$2079 = \ldots\ldots\ldots\ldots\ldots\ldots\ldots\ldots\ldots\ldots\ldots\ldots\ldots\ldots\ldots\ldots$$

$$735 = \ldots\ldots\ldots\ldots\ldots\ldots\ldots\ldots\ldots\ldots\ldots\ldots\ldots\ldots\ldots\ldots$$

$$\mathrm{ggT}(2079, 735) = \ldots\ldots\ldots\ldots\ldots\ldots\ldots\ldots\ldots\ldots\ldots\ldots\ldots\ldots\ldots\ldots$$

3.5.2 Euklidischer Algorithmus

Man subtrahiert immer wieder die kleinere von der größeren Zahl und fährt mit kleinerer Zahl und Differenz fort. Die letzte Zahl, die nicht durch subtrahieren auf null führt, ist der größte gemeinsame Teiler.

$$
\begin{aligned}
2079 - 735 &= 1344 \\
1344 - 735 &= 609 \\
735 - 609 &= 126 \\
609 - 126 &= 483 \\
483 - 126 &= 357 \\
357 - 126 &= 231 \\
231 - 126 &= 105 \\
126 - 105 &= 21 \\
105 - 21 &= 84 \\
84 - 21 &= 63 \\
63 - 21 &= 42 \\
42 - 21 &= 21 \\
(21 - 21 &= 0)
\end{aligned}
\qquad\qquad
\begin{aligned}
2079 \bmod 735 &= 609 \\
735 \bmod 609 &= 126 \\
609 \bmod 126 &= 105 \\
126 \bmod 105 &= 21 \\
(105 \bmod 21 &= 0)
\end{aligned}
$$

Mehrmaliges Subtrahieren derselben Zahl (wie es links gemacht wurde) lässt sich durch **eine** Modulo-Operation ersetzen (rechts).

Aufgabe 31:
Was lässt sich aus $\mathrm{ggT}(2079, 735) = 21$ alles schließen?

. . . . . .

. . . . . .

3.5.3 Erweiterter euklidischer Algorithmus

Bei zwei ganzen Zahlen $a, b \in \mathbb{Z}$ ist es immer möglich, den größten gemeinsamen Teiler als Vielfachensumme von a und b zu schreiben, also in der Form

$$\mathrm{ggT}(a, b) = s \cdot a + t \cdot b,$$

wobei $s, t \in \mathbb{Z}$. Wir wollen nun diese Multiplikatoren s und t bestimmen. Dazu schreiben wir zwei Zeilen an, die offensichtlich stimmen:

$$
\begin{aligned}
a &= 1 \cdot a + 0 \cdot b \\
b &= 0 \cdot a + 1 \cdot b
\end{aligned}
$$

Mit der linken Seite führen wir den einfachen euklidischen Algorithmus durch. Mit der rechten Seite führen wir dieselben Rechenschritte durch, damit die Gleichheiten weiterhin stimmen. Irgendwann steht dann links der $\mathrm{ggT}(a, b)$ und rechts eine Vielfachensumme von a und b, die gleich dem ggT ist.

$$
\begin{aligned}
2079 &= 1 \cdot 2079 + 0 \cdot 735 && \\
735 &= 0 \cdot 2079 + 1 \cdot 735 && \Big| \cdot(-2) \quad \text{und mit obiger Zeile addieren} \\
609 &= 1 \cdot 2079 + (-2) \cdot 735 && \Big| \cdot(-1) \quad \text{und mit obiger Zeile addieren} \\
126 &= (-1) \cdot 2079 + 3 \cdot 735 && \Big| \cdot(-4) \quad \text{und mit obiger Zeile addieren} \\
105 &= 5 \cdot 2079 + (-14) \cdot 735 && \Big| \cdot(-1) \quad \text{und mit obiger Zeile addieren} \\
21 &= (-6) \cdot 2079 + 17 \cdot 735 && \text{fertig, weil 21 die 105 teilt bzw.} \\
& && \text{als nächstes 0 folgen würde}
\end{aligned}
$$

Was nützt nun dieser Algorithmus?
In Aufgabe 29 brauchten wir, um die Verschlüsselung „mit $\overline{5}$ multiplizieren“ umzukehren, das Inverse von $\overline{5}$ in $\mathbb{Z}_{26}$. Wir wenden dazu den erweiterten euklidischen Algorithmus auf die Zahlen 5 und 26 an:

$$
\begin{aligned}
26 &= 1 \cdot 26 + 0 \cdot 5 && \\
5 &= 0 \cdot 26 + 1 \cdot 5 && \Big| \cdot(-5) \quad \text{und mit obiger Zeile addieren} \\
1 &= 1 \cdot 26 + (-5) \cdot 5 && \text{fertig, weil 1 die 5 teilt}
\end{aligned}
$$

Wir erfahren aus der letzten Gleichung zweierlei:

- Der $\mathrm{ggT}(26, 5) = 1$, deshalb ist $\overline{5} \in \mathbb{Z}_{26}$ invertierbar.
- Das Inverse zu $\overline{5}$ ist $\overline{-5} = \overline{21}$. Es gilt nämlich

$$\overline{1} = \overline{1 \cdot 26 + (-5) \cdot 5} = \overline{1} \cdot \underbrace{\overline{26}}_{=\overline{0}} + \overline{-5} \cdot \overline{5} = \overline{-5} \cdot \overline{5} = \overline{21} \cdot \overline{5}.$$

Ob ein Element eines Restklassenringes invertierbar ist oder nicht, lässt sich mit Hilfe des erweiterten euklidischen Algorithmus feststellen. Ist der ggT aus der Zahl (einem Repräsentanten der Restklasse) und dem Modul gleich 1, ist es invertierbar. Das Inverse kann in der letzten Zeile abgelesen werden. Damit sind die ersten beiden Fragen von Seite 35 beantwortet.

Aufgabe 32:
Entscheide, ob die Elemente $\overline{15}$ und $\overline{44}$ von $\mathbb{Z}_{81}$ invertierbar sind. Falls ja, bestimme ihr Inverses.

.....

.....

.....

.....

.....

.....

.....

3.6 Die Eulersche Phi-Funktion

Wir führen die nach LEONHARD EULER benannte Phi-Funktion ein, die einer positiven natürlichen Zahl n die Anzahl der zu n teilerfremden Zahlen von 1 bis n zuordnet. Damit gibt $\varphi(n)$ die **Anzahl** der invertierbaren Elemente von $\mathbb{Z}_n$ an.

Aufgabe 33:
Berechne (p bezeichnet eine Primzahl):

(a) $\varphi(1) = 1$ Begründung: 1 ist zu sich selbst teilerfremd

(b) $\varphi(2) =$ .....

(c) $\varphi(3) =$ .....

(d) $\varphi(5) =$ .....

(e) $\varphi(7) =$ .....

(f) $\varphi(11) =$ .....

(g) $\varphi(p) =$ .....

(h) $\varphi(4) =$ .....

(i) $\varphi(6) =$ .....

(j) $\varphi(8) =$ .....

(k) $\varphi(15) =$ .....

Es gibt für $\varphi(n)$ eine Berechnungsmöglichkeit, wenn man die Primfaktorzerlegung von n kennt:

- Ist n prim, gilt $\varphi(n) = n - 1$. (z. B. $\varphi(11) = 10$)
- Ist $n = p^k$, gilt $\varphi(n) = p^{k-1} \cdot (p-1)$. (z. B. $\varphi(8) = \varphi(2 \cdot 2 \cdot 2) = 2 \cdot 2 \cdot 1 = 4$)
- Über verschiedene Primfaktoren, spaltet sich φ auf: $\varphi(p^k \cdot q^l) = \varphi(p^k) \cdot \varphi(q^l)$ (z. B. $\varphi(15) = \varphi(3 \cdot 5) = \varphi(3) \cdot \varphi(5) = 2 \cdot 4 = 8$)

Aufgabe 34:
Berechne mit diesen Regeln $\varphi(810) = \varphi(2 \cdot 3^4 \cdot 5)$.

.....

Das heißt, es gibt ..... Zahlen in $\{1, 2, \ldots, 810\}$, die zu 810 teilerfremd sind. Folglich hat $\mathbb{Z}_{810}$ genauso viele invertierbare Elemente.

Zwei Kommentare zu dieser Methode:

- Es ist beruhigend, dass man für die Berechnung von $\varphi(n)$ nicht alle Zahlen bis n durchprobieren muss, um festzustellen, **wie viele** davon in $\mathbb{Z}_n$ invertierbar sind.
- Andererseits ist die Primfaktorzerlegung gerade bei großen Zahlen **nicht leicht** zu bekommen.

Wir rufen die drei Fragen von Seite 35 in Erinnerung und fassen die Antworten zusammen:

Das Berechnen von Inversen ist nicht nur lösbar sondern auch komplexitätstheoretisch **leicht**. Er benötigt nämlich nur ein Anzahl von Schritten, die logarithmisch von der Größe von a und b abhängt bzw. proportional ist zur Bitanzahl von a und b.[11] Das Bestimmen, wie viele invertierbare Elemente es gibt, ist nur bei bekannter Primfaktorzerlegung leicht, ansonsten schwer.

3.7 Einweg- und Falltürfunktionen

Bekanntlich weist eine **Funktion** *jedem* Element der *Definitionsmenge genau ein* Element der *Wertemenge* zu (siehe beispielsweise Abbildung 8).

[11] Das gilt für die Variante, in der b von a gleich $(a//b)$-mal abgezogen wird. Der ungünstigste Fall sind dann zwei aufeinanderfolgende Fibonacci-Zahlen $a = f_{n+1}$ und $b = f_n$, mit denen der Algorithmus $n - 1$ Schritte benötigt.

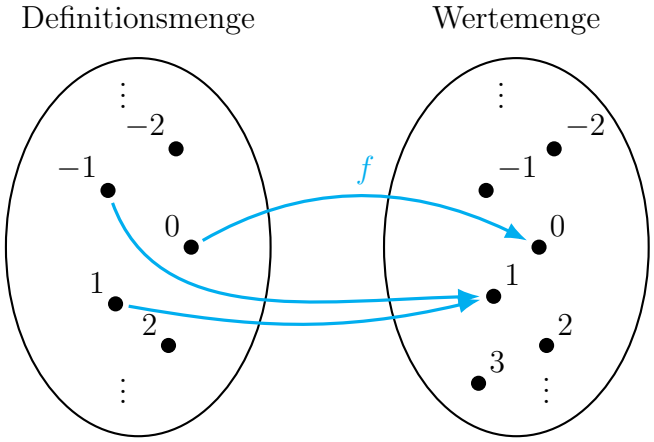

Abbildung 8: Als Beispiel für eine Funktion $f\colon \mathbb{Z} \to \mathbb{Z}; x \mapsto x^2$. Jedes Element der Definitionsmenge bekommt *genau ein* Element der Wertemenge zugeordnet. Nicht zwangsläufig wird jedem Element der Wertemenge etwas zugeordnet (z. B. -1). Nicht zwangsläufig wird jedem Element der Wertemenge nur ein Element zugeordnet (z. B. 1).

Zur Verschlüsselung kann eine Funktion wie in Abbildung 8 nicht verwendet werden,

. . . . . .

Allerdings haben *Hashfunktionen* diese Eigenschaft, dass mehrere Nachrichten denselben Hashwert ergeben.

Eine **umkehrbare Funktion (bijektive Funktion)** verwendet jedes Element der Wertemenge *genau einmal*. Bei solchen Funktionen ist es aufgrund der Eindeutigkeit möglich, „vom y wieder zurück zum x zu kommen“ (siehe beispielsweise Abbildung 9).

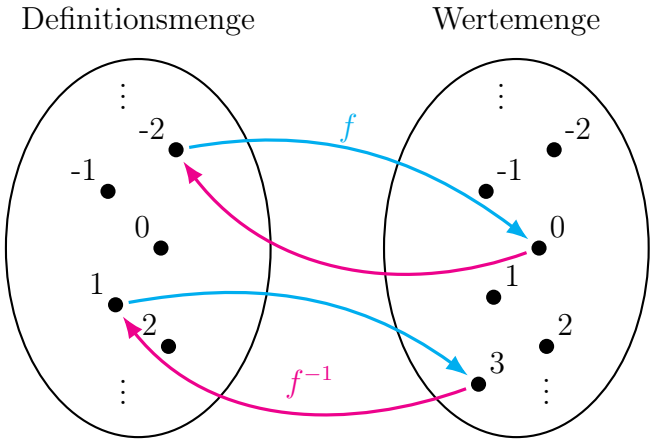

Abbildung 9: Als Beispiel für eine umkehrbare Funktion $f\colon \mathbb{Z} \to \mathbb{Z}; x \mapsto x + 2$. Jedes Element der Definitionsmenge bekommt *genau ein* Element der Wertemenge zugeordnet und jedem Element der Wertemenge wird *genau ein* Element der Definitionsmenge zugeordnet.

Von einer **Einwegfunktion** spricht man, wenn die Funktion zwar umkehrbar ist, die Umkehrung aber nur *sehr aufwändig* berechenbar ist (siehe Abbildung 10).

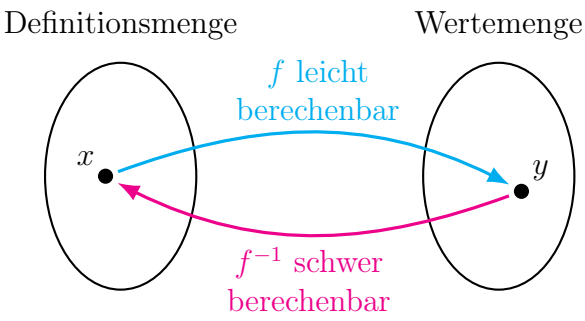

Abbildung 10: Eine Einwegfunktion ist theoretisch umkehrbar. Praktisch aber nicht, weil die Umkehrung nur unter großem Aufwand berechnet werden kann.

Zur Verschlüsselung kann eine Einwegfunktion nicht verwendet werden,

. . . . . .

. . . . . .

Wie kann man sich „leicht“ und „schwer berechenbar“ vorstellen?

- . . . . . .
- . . . . . .
- . . . . . .

. . . . . .

Wer noch nicht überzeugt ist von der Primfaktorzerlegung: Die Faktorisierung der Zahl `RSA-768` (sie hat 768 Bit bzw. 232 Dezimalstellen) dauerte 2009 unter Verwendung von hunderten Computern zwei Jahre. Das Ergebnis ist:

$$
\begin{aligned}
&123018668453011775513049495838496272077285356959533479219732245215172640050726365751874520219978646938995647494277406384592519255732630345373154826850791702612214291346167042\\
&92143116022212404792747377940806653514195974598569021434133478071698956898786044169848212690817704794983713768568912431388982883793878002287614711652531743087737814467999489\cdot\\
&\cdot 36746043666799590428244633799627952632279158164343087642676032283815739666511279233373417143396810270092798736308917
\end{aligned}
$$

Das Berechnen des Produktes ist eine Sache von Sekundenbruchteilen.

Von einer **Falltürfunktion** spricht man, wenn die Umkehrung *im Allgemeinen* nur *sehr aufwändig* berechenbar ist, mit einer *zusätzlichen Information* aber *leicht berechenbar* wird (siehe Abbildung 11).

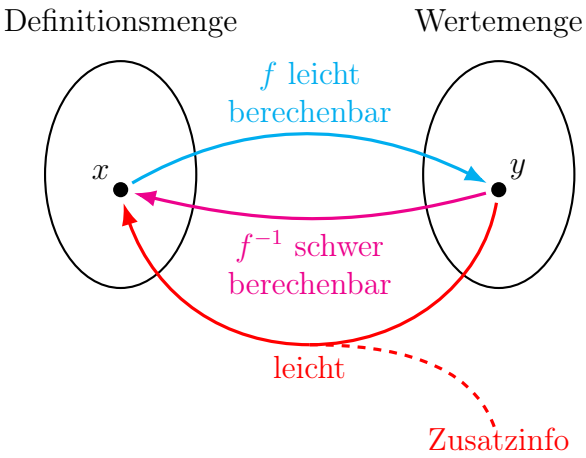

Abbildung 11: Eine Falltürfunktion ist theoretisch umkehrbar. Praktisch aber nur, wenn eine gewisse Zusatzinformation vorliegt, die das Umkehren „leicht“ macht.

Ein Analogon für eine Falltürfunktion wäre ein Brief, der in einen Briefkasten gesteckt wird (einfach), aber kaum herauszubekommen ist (schwere Umkehrung). Hat man jedoch eine Zusatzinformation (Schlüssel, Zahlencode), um den Briefkasten zu öffnen, wird die Umkehrung auf einmal leicht.

Das Multiplizieren in Restklassenringen ist eine

. . . . . .

. . . . . .

. . . . . .

Anders ist das mit der nächsten Rechenoperation.

3.8 Modulares Potenzieren und diskreter Logarithmus

3.8.1 Modulares Potenzieren

Für ein Element $a \in \mathbb{Z}_m$ definieren wir Potenzen mit natürlichen Hochzahlen n auf die naheliegende Weise

$$a^n = \underbrace{a \cdot a \cdot \ldots \cdot a}_{n\text{-mal}}.$$

Da $\mathbb{Z}_m$ nur m Elemente hat, wird es nach spätestens m Schritten zu Wiederholungen kommen müssen.

Aufgabe 35:
Berechne die Potenzen in $\mathbb{Z}_{10}$ und in $\mathbb{Z}_8$. Markiere die nicht invertierbaren Elemente und die $\overline{1}$ in verschiedenen Farben.

n	1	2	3	4	5	6	7	8	9	10
$\overline{0}^n$										
$\overline{1}^n$										
$\overline{2}^n$										
$\overline{3}^n$										
$\overline{4}^n$										
$\overline{5}^n$										
$\overline{6}^n$										
$\overline{7}^n$										
$\overline{8}^n$										
$\overline{9}^n$										

n	1	2	3	4	5	6	7	8
$\overline{0}^n$								
$\overline{1}^n$								
$\overline{2}^n$								
$\overline{3}^n$								
$\overline{4}^n$								
$\overline{5}^n$								
$\overline{6}^n$								
$\overline{7}^n$								

Wir können an Aufgabe 35 schon ein paar Gesetzmäßigkeiten beobachten:

- Die nicht invertierbaren Elemente „bleiben unter sich“. Das stimmt generell und liegt daran, dass Nullteiler miteinander multipliziert wieder Nullteiler ergeben. (Gibt $a \cdot b = \overline{0}$, dann gibt $(a \cdot a) \cdot (b \cdot b) = \overline{0}$.)
- Die invertierbaren Elemente bleiben ebenfalls „unter sich“. Ihre Menge bezeichnet man mit $\mathbb{Z}_m^\times$. Auch das stimmt generell und liegt daran, dass das Produkt invertierbarer Elemente invertierbar ist. (Hat a das Inverse a^{-1}, dann hat $a \cdot a$ das Inverse $a^{-1} \cdot a^{-1}$, weil $\big(a \cdot a\big) \cdot \big(a^{-1} \cdot a^{-1}\big) = a \cdot \overline{1} \cdot a^{-1} = \overline{1}$.)

- Es gibt in $\mathbb{Z}_{10}$ und $\mathbb{Z}_8$ jeweils vier invertierbare Elemente:

$$\varphi(10) = \varphi(2 \cdot 5) = 1 \cdot 4 = 4$$
$$\varphi(8) = \varphi(2 \cdot 2 \cdot 2) = 1 \cdot 2 \cdot 2 = 4$$

  Die Potenzen der invertierbaren Elemente wiederholen sich alle vier Schritte. Das stimmt generell.[12] Teilweise wiederholen sie sich auch schneller.
  In jedem Fall muss aber die „Periodenlänge“ ein Teiler von $\varphi(m)$ sein.

- Kann sie auch gleich $\varphi(m)$ sein? Dann müssten alle invertierbaren Elemente einmal vorkommen.
  Im Fall von $\mathbb{Z}_{10}$ gibt es Zeilen (die von $\overline{3}$ und $\overline{7}$), in denen **alle** invertierbaren Elemente einmal vorkommen. Man nennt in diesem Fall $\overline{3}$ und $\overline{7}$ **Generatoren** von $\mathbb{Z}_{10}^\times$ und $\mathbb{Z}_{10}^\times$ **zyklisch**.
  Hingegen ist $\mathbb{Z}_8^\times$ **nicht** zyklisch, es hat keine Generatoren. Es gibt kein Element in $\mathbb{Z}_8^\times$, so dass durch Potenzieren jedes andere erreicht wird.[13]

- Bei den nicht invertierbaren Elementen kommt es natürlich auch irgendwann zu Wiederholungen, aber es kann „am Anfang“ ein „nicht-periodischer Teil“ auftreten.

Aufgabe 36:
Wende alle diese Erkenntnisse und Eigenschaften auf $\mathbb{Z}_{13}$ an. Versuche, so viel wie möglich über die Potenzen von $\mathbb{Z}_{13}$ vorherzusagen:

. . . . . .

. . . . . .

. . . . . .

. . . . . .

. . . . . .

[12] Der *Satz von Euler* besagt, dass für teilerfremde $a, n \in \mathbb{N}$ gilt: $a^{\varphi(n)} \equiv 1 \pmod{n}$

[13] Zyklisch ist $\mathbb{Z}_m^\times$ genau dann, wenn $m = 1, 2, 4$ oder die Potenz (p^k) oder das Doppelte einer Potenz $(2p^k)$ einer Primzahl $p > 2$ ist. Bei zyklischen Gruppen mit n Elementen gibt es genau $\varphi(n)$ Generatoren. Außerdem gibt es für jeden Teiler k von n genau $\varphi(k)$ Zeilen mit k verschiedenen Elementen.
Ein Algorithmus, der effizient Generatoren berechnet, ist allerdings unbekannt.
Betrachten wir $\mathbb{Z}_{10}$: Da $10 = 2 \cdot 5$, ist $\mathbb{Z}_{10}^\times$ zyklisch. Wegen $\varphi(10) = 4$, enthält $\mathbb{Z}_{10}^\times$ vier Elemente, darunter sind $\varphi(4) = 2$ Generatoren (nämlich $\overline{3}$ und $\overline{7}$). Weitere Teiler von 4 sind 1 und 2. Wegen $\varphi(1) = 1$ gibt es ein Element, mit dem durch Potenzieren nur ein Elemente erreicht wird (nämlich $\overline{1}$). Wegen $\varphi(2) = 1$ gibt es ein Element, für das die Potenzen eine „Periodenlänge“ von zwei haben (nämlich $\overline{9} = \overline{-1}$).
Betrachten wird $\mathbb{Z}_8$: Da 8 weder $1, 2, 4, p^k$ noch $2p^k$ ist für eine Primzahl $p > 2$, ist $\mathbb{Z}_8^\times$ nicht zyklisch, hat keine Generatoren. Die „Periodenlängen“ sind aber trotzdem Teiler von $\varphi(8) = 4$.

. . . . . .

. . . . . .

. . . . . .

. . . . . .

. . . . . .

Berechne die Potenzen in $\mathbb{Z}_{13}$. Markiere die nicht invertierbaren Elemente und die $\overline{1}$ in verschiedenen Farben.

n	1	2	3	4	5	6	7	8	9	10	11	12
$\overline{0}^n$												
$\overline{1}^n$												
$\overline{2}^n$												
$\overline{3}^n$												
$\overline{4}^n$												
$\overline{5}^n$												
$\overline{6}^n$												
$\overline{7}^n$												
$\overline{8}^n$												
$\overline{9}^n$												
$\overline{10}^n$												
$\overline{11}^n$												
$\overline{12}^n$												

3.8.2 Diskreter Logarithmus

Zuerst erinnern wir uns an die Definition des Logarithmus auf den reellen Zahlen. Wenn

$$b^x = a,$$

dann heißt

$$x = \log_b(a)$$

der *Logarithmus von a zur Basis b*. Die Logarithmusfunktion zur Basis b ist also die Umkehrfunktion zur Exponentialfunktion zur Basis b.

Genauso definieren wir den *diskreten Logarithmus zur Basis* $\overline{b}$: Wenn

$$b^n = a,$$

(und n kleinstmöglich) dann heißt

$$n = \log_b(a)$$

der *diskrete Logarithmus von a zur Basis b*.

Aufgabe 37:
Berechne in $\mathbb{Z}_{13}$ (bzw. nutze dafür die Tabelle aus Aufgabe 36!):

(a) $\log_{\overline{5}}(\overline{8}) =$ .....

(b) Ist der Logarithmus 3 oder $\overline{3}$? .....

(c) $\log_{\overline{5}}(\overline{1}) =$ .....

(d) $\log_{\overline{5}}(\overline{7}) =$ .....

(e) $\log_{\overline{2}}(\overline{7}) =$ .....

(f) Ist $\log_{\overline{b}}(\overline{4})$ definiert? .....

(g) Ist $\log_{\overline{12}}(a)$ definiert? .....

(h) Ist $\log_{\overline{7}}(a)$ definiert? .....

(i) $\log_{\overline{11}}(\overline{1}) =$ .....

Wir sehen bereits am Beispiel $\mathbb{Z}_{13}$, dass das Berechnen von diskreten Logarithmen gewissermaßen kompliziert ist. Aufgrund der Wiederholungen in Restklassenringen, liefern mehrere Hochzahlen dasselbe Ergebnis. Als diskreten Logarithmus verwenden wir nur die kleinste. Da nicht jedes Element in $\mathbb{Z}_m^\times$ durch Potenzieren der Basis erreicht wird (außer

sie ist Generator), ist der diskrete Logarithmus für viele Argumente **undefiniert**. Da die Potenzen zu einer Basis scheinbar chaotisch „herumhüpfen“, wirken auch die Logarithmen „unvorhersehbar“.

Aufgabe 38:
Wir erinnern uns an Aufgabe 29, wo wir die Idee hatten in $\mathbb{Z}_{26}$ (alle Buchstaben) die Nachricht m zu verschlüsseln, indem wir sie mit dem Schlüssel $k = 13$ bzw. $k = 5$ multiplizieren. Wir haben graphisch gesehen, dass die Chiffretexte c nur auf den ersten Blick hin- und herspringen, in Wahrheit aber die Multiplikation linear ist (abgeschnittene Geraden).

Wir probieren dasselbe jetzt mit Potenzieren aus. Da $26 = 2 \cdot 13$, ist $\mathbb{Z}_{26}$ zyklisch. Folglich gibt es Generatoren, deren Potenzen ganz $\mathbb{Z}_{26}^{\times}$ erreichen. Solche Generatoren sind z. B. 7 und 11. Deren Potenzen 7^m bzw. 11^m erreichen alle $\varphi(26) = \varphi(2 \cdot 13) = 1 \cdot 12$ invertierbaren Elemente, wobei alle 12 Schritte sich alles wiederholt. (Eigentlich haben wir also nur 12 Zeichen beim Klartext und beim Chiffretext. Die Diagramme zeigen aber trotzdem, worum es geht.)

Offensichtlich hüpfen die 7^m viel unregelmäßiger und unvorhersehbarer herum als bei der Multiplikation, siehe Abbildung 12.

3.8.3 Aufwand des modularen Potenzierens

Aufgabe 39:
Wie viel Aufwand ist es,

$$174^{231} \bmod 23$$

zu berechnen?

Wir kennen bereits ein paar Strategien, um sich die Arbeit zu verringern:

- .....
- .....

.....

Allerdings ist $13^{11} = 1\,792\,160\,394\,037$ immer noch recht groß, um es zu berechnen und danach mod 23 zu nehmen. Der Aufwand lässt sich aber noch weiter reduzieren.

Wir vermischen im Folgenden Zahlen in Dezimal- und Binärdarstellung. Um die Schreibweise übersichtlich zu halten, schreibe ich Binärziffern im `Schreibmaschinenstil` ($6 =$ `110`).

Wir rufen uns Potenzrechenregeln und Binärzahlen in Erinnerung:

- Wird die Potenz b^n mit der Basis multipliziert,

  .....

  Das bedeutet für den Exponenten, wenn seine letzte Ziffer in Binärdarstellung `0` ist,

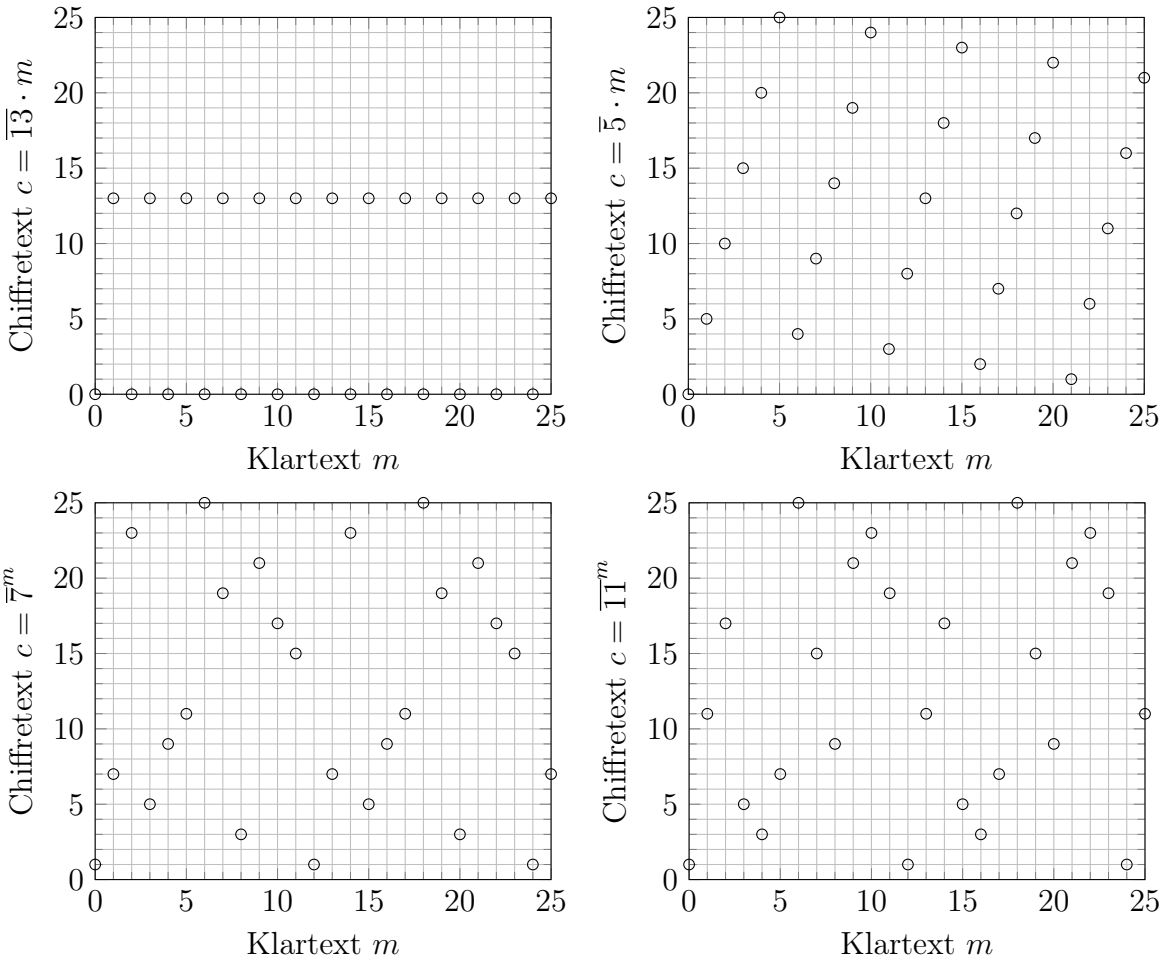

Abbildung 12: Visualisierung der Verschlüsselung durch modulare Multiplikation mit den Schlüsseln $k = \overline{13}$ (links oben) und $k = \overline{5}$ (rechts oben) sowie durch modulare Exponentiation mit Basen $k = 7$ (links unten) und $k = 11$ (rechts unten).
Bei genauer Betrachtung bemerkt man, dass sich die Potenzen nach $12 = \varphi(26)$ Schritten wiederholen (Satz von Euler). Dementsprechend werden auch nur 12 verschiedene Werte erreicht (z. B. kommt 14 nie vor).
Bei den Multiplikationen ist der lineare Zusammenhang leicht erkennbar. Die Potenzen sind deutlich „chaotischer“.

. . . . . .

- Wird die Potenz b^n quadriert,

  . . . . . .

  Das bedeutet für den Exponenten in Binärdarstellung,

  . . . . . .

  . . . . . .

- Jede Binärzahl lässt sich schrittweise „aufbauen“, indem man rechts eine `0` anhängt oder die letzte Ziffer von `0` nach `1` ändert. Zum Beispiel entsteht $11 = \texttt{1011}$ schrittweise aus:

$$\texttt{1} \to \texttt{10} \to \texttt{100} \to \texttt{101} \to \texttt{1010} \to \texttt{1011}$$

Square-and-Multiply-Algorithmus
Damit beim Potenzieren modulo m auch die Zwischenergebnisse klein gehalten werden, wird nach jedem Rechenschritt mod m reduziert. Die Potenz modulo m lässt sich schrittweise durch Quadrieren und Multiplizieren berechnen:

- Als Vorbereitung rechnet man den Exponenten e in Binärdarstellung um. (Diese Arbeit entfällt im Computer, weil e sowieso schon so abgespeichert ist!)
- Man beginnt mit der Basis b und geht die Binärziffern des Exponenten von der zweithöchsten abwärts durch.
- Man quadriert und reduziert modulo m. Sollte die nächste Binärziffer eine `1` sein, multipliziert man mit b und reduziert modulo m.

Aufgabe 40:
Zu berechnen ist

$$13^{11} \bmod 23 = 13^{\texttt{1011}} \bmod 23.$$

$$\begin{aligned}
13^1 = 13^{\texttt{1}} &= 13 && \xrightarrow{\text{mod } 23} && 13 && \xrightarrow{\text{Quadrieren}} \\
13^2 = 13^{\texttt{10}} &= 13 \cdot 13 = 169 && \xrightarrow{\text{mod } 23} && 8 && \xrightarrow{\text{Quadrieren}} \\
13^4 = 13^{\texttt{100}} &= 8 \cdot 8 = 64 && \xrightarrow{\text{mod } 23} && 18 && \xrightarrow{\text{Multipl. m. } 13} \\
13^5 = 13^{\texttt{101}} &= 18 \cdot 13 = 234 && \xrightarrow{\text{mod } 23} && 4 && \xrightarrow{\text{Quadrieren}} \\
13^{10} = 13^{\texttt{1010}} &= 4 \cdot 4 = 16 && \xrightarrow{\text{mod } 23} && 16 && \xrightarrow{\text{Multipl. m. } 13} \\
13^{11} = 13^{\texttt{1011}} &= 16 \cdot 13 = 208 && \xrightarrow{\text{mod } 23} && 1
\end{aligned}$$

Bemerkenswert an der Berechnung von

$$174^{231} \bmod 23$$

ist, dass kein Zwischenergebnis je größer als $22 \cdot 22 = 484$ werden kann und dass nur je fünf Modulo-Schritte und fünf Multiplikationen (inklusive Quadrierungen) nötig waren. Würde man naiv

$$174^{231} = \underbrace{174 \cdot 174 \cdot \ldots \cdot 174}_{\text{231-mal}}$$

berechnen, wären 230 Multiplikationen nötig.

Allgemein gilt: Um b^e zu berechnen, müssen, wenn der Exponent e genau n Binärstellen hat,

. . . . . .

durchgeführt werden. Da

$n =$ . . . . . .

. . . . . .

Jede Verdoppelung des Exponenten bewirkt, dass der Aufwand

. . . . . .

Aufgabe 41:
Um diesen Sachverhalt numerisch zu demonstrieren, betrachten wir die Situation von RSA, wo heutzutage zur Entschlüsselung Potenzen berechnet werden müssen, bei denen Basis und Exponent an die 2048 Bit (oder 4096 Bit) haben. Man blättere noch einmal auf Seite 41 zurück, wo eine 768 Bit-Zahl zu sehen ist. Der Exponent hier ist dreimal so lang! So oft wäre naiv die Basis mit sich selbst zu multiplizieren. Das ist praktisch undurchführbar.

Mit Hilfe des Square-and-Multiply-Algorithmus sind im schlimmsten Fall nur

. . . . . .

. . . . . .

Aufgabe 42:
Berechne:

(a) $14^{2174658914561} \bmod 7 =$ . . . . . .

(b) $19^{2174658914561} \bmod 9 =$ . . . . . .

(c) $17^{2174658914561} \bmod 9 =$ . . . . . .

(d) $7^{1298} \bmod 11 =$ ....

$$\begin{array}{lllllllll}
 & = & & = & & \xrightarrow{\bmod\ 11} & & \longrightarrow & \\
 & = & & = & & \xrightarrow{\bmod\ 11} & & \longrightarrow & \\
 & = & & = & & \xrightarrow{\bmod\ 11} & & \longrightarrow & \\
 & = & & = & & \xrightarrow{\bmod\ 11} & & \longrightarrow & \\
 & = & & = & & \xrightarrow{\bmod\ 11} & & \longrightarrow & \\
 & = & & = & & \xrightarrow{\bmod\ 11} & & \longrightarrow & \\
 & = & & = & & \xrightarrow{\bmod\ 11} & & \longrightarrow & \\
 & = & & = & & \xrightarrow{\bmod\ 11} & & \longrightarrow & \\
 & = & & = & & \xrightarrow{\bmod\ 11} & & \longrightarrow & \\
 & = & & = & & \xrightarrow{\bmod\ 11} & & \longrightarrow & \\
 & = & & = & & \xrightarrow{\bmod\ 11} & & \longrightarrow & \\
 & = & & = & & \xrightarrow{\bmod\ 11} & & \longrightarrow & \\
 & = & & = & & \xrightarrow{\bmod\ 11} & & \longrightarrow & \\
 & = & & = & & \xrightarrow{\bmod\ 11} & & &
\end{array}$$

Weitere Zahlenbeispiele kann man sich leicht selbst stellen, indem man Zahlen erfindet und z. B. mit dem gewöhnlichen WINDOWS-RECHNER oder dem TI N-SPIRE überprüft.

3.8.4 Aufwand der Berechnung des diskreten Logarithmus

Wir haben bisher diskrete Logarithmen zur Basis b so bestimmt, dass wir der Reihe nach Potenzen b^n für $n = 1, 2, 3, \ldots$ berechnet haben, bis wir auf das Argument gestoßen sind (oder auf 1, was bedeutet, dass das Argument nie als Potenz b^n vorkommt und dieser diskrete Logarithmus nicht definiert ist). Letztlich läuft diese Strategie auf **bloßes Ausprobieren** hinaus.

Es stellt sich natürlich die Frage, ob sich der diskrete Logarithmus „besser“ berechnen lässt:

„Es sind bisher keine schnellen Algorithmen zur Berechnung des diskreten Logarithmus bekannt. Deren Laufzeit verhielte sich polynomial zur Länge der Eingabe. Es gibt aber Algorithmen, die die Lösung gezielter finden als bloßes Ausprobieren. Aufgrund des angesprochenen Laufzeitverhaltens und der in der Kryptografie üblichen Größenordnungen (mehrere hundert Dezimalstellen in Numerus und Basis) spielen sie praktisch aber keine Rolle.“[14]

[14] https://de.wikipedia.org/wiki/Diskreter_Logarithmus

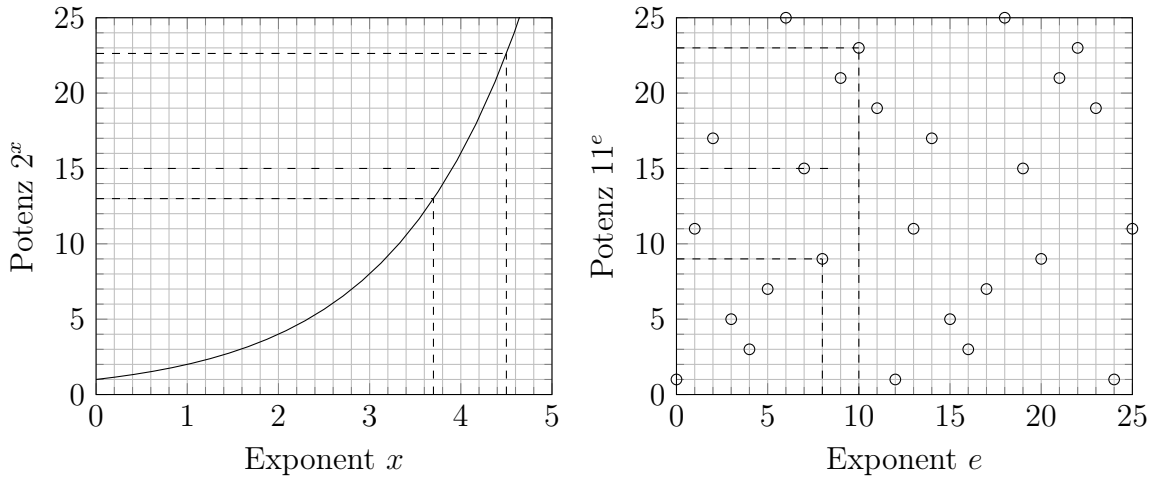

Abbildung 13: Beim Logarithmus auf den reellen Zahlen ist es möglich, sich schrittweise anzunähern: Weil $2^{4,5} > 15$ ist, muss 4,5 zu groß sein für $\log_2(15)$. Weil $2^{3,7} < 15$ ist, muss 3,7 zu klein sein und die Lösung zwischen 3,7 und 4,5 liegen. So arbeitet man sich immer weiter vor und der mögliche Bereich kann mit jedem Schritt halbiert werden. Das heißt, seine Länge verschwindet mit exponentieller Geschwindigkeit!
Beim diskreten Logarithmus gibt es diese Möglichkeit nicht. Obwohl $11^8 \bmod 26 = 9 < 15$ und $11^{10} \bmod 26 = 23 > 15$, liegt $\log_{\overline{11}}(\overline{15}) = 7$ nicht zwischen 8 und 10.

Warum kann hingegen ein Taschenrechner mühelos Logarithmen in $\mathbb{R}$ berechnen?
Das liegt letztlich an der Tatsache, dass $f(x) = 2^x$ eine **stetige** Funktion ist und man hier ausnützen kann, dass die Lösung zwischen Stellen mit zu hohem und zu niedrigem Wert liegen muss. So kann man sich schrittweise an die Lösung heranarbeiten. Die modularen Potenzen springen im Gegensatz dazu schwer vorhersehbar herum, weshalb ein Annähern nicht möglich ist, siehe Abbildung 13.

Damit ist das Berechnen des diskreten Logarithmus viel aufwändiger als das des gewöhnlichen Logarithmus. Kryptographisch bedeutsam ist aber das Folgende: **Das Berechnen des diskreten Logarithmus ist viel aufwändiger als das modulare Potenzieren. Wir haben hier eine kryptographische Einwegfunktion vorliegen!**[15]

Aufgabe 43:
Berechne den Aufwand, eine Zahl mit einer 128 Bit-Zahl zu potenzieren:

schlimmstenfalls: .....

bestenfalls: .....

Berechne den Aufwand, eine solche Zahl zu logarithmieren:

[15] Allerdings keine Falltürfunktion: Es ist kein Algorithmus bekannt, der mit einer Zusatzinformation den diskreten Logarithmus effizient berechnen kann.

schlimmstenfalls: .....

Erwartungswert: .....

Lass die folgenden Zahlen auf dich wirken:

$$254 = \ldots\ldots\ldots$$

$$170\,141\,183\,460\,469\,231\,731\,687\,303\,715\,884\,105\,728 = \ldots\ldots\ldots$$

Überlege, was mit diesen beiden Zahlen passiert, wenn man 2048 Bit-Zahlen verwendet:

.....

.....

4 Diffie-Hellman-Schlüsselaustausch

Ist es möglich, dass Alice und Bob sich öffentlich Informationen zurufen, so dass sie beide danach etwas wissen, aber sonst niemand?
Überraschenderweise ist die Antwort darauf: Ja!

Aufgabe 44:
Erfahre die grundlegende Idee dahinter auf der Website[16] anhand von Farben.

Wesentliches Element dabei ist, dass einerseits Farben unkompliziert gemischt werden können und Mischungen ungesichert übertragen werden können, weil Eve praktisch nicht in der Lage ist, die Mischung rückgängig zu machen. Mischen ist also eine Einwegfunktion.

Es handelt sich um keine Falltürfunktion. Auch Bob kann die Farben nicht entmischen. Er mischt nur weiter.

Dieses Grundschema funktioniert auch mit anderen Einwegfunktionen. Der **Diffie-Hellman-Schlüsselaustausch**[17] verwendet das **modulare Potenzieren**.

Diffie-Hellman-Schlüsselaustausch – Das Verfahren

Das 1976 von Whitfield Diffie und Martin Hellman veröffentlichte Verfahren funktioniert wie folgt, siehe auch Abbildung 14:

1. Alice und Bob einigen sich auf eine Primzahl p und eine Zahl $g < p$ (die idealerweise eine Generator von $\mathbb{Z}_p^\times$ ist). Diese Vereinbarung geschieht öffentlich und deshalb sind p und g auch Eve bekannt.[18]

2. Alice und Bob wählen jeweils $x < p$ bzw. $y < p$ und potenzieren g damit (modulo p).

3. Das Ergebnis schicken sie dem anderen, der es wieder mit seiner privaten Zahl x bzw. y potenziert.

4. Da $(g^x)^y = g^{x \cdot y} = (g^y)^x$, wissen Alice und Bob nun dasselbe Geheimnis.

5. Eve hingegen hat die Informationen p, g, $g^x \bmod p$ und $g^y \bmod p$. Könnte sie effizient den diskreten Logarithmus berechnen, käme sie aus $g^y \bmod p$ auf y und damit leicht auf das Geheimnis $(g^x)^y \bmod p$.
   Da das Bilden des diskreten Logarithmus aber enorm aufwändig ist (wie das Entmischen von Farben), gelingt ihr das nicht.

[16] https://www.inf-schule.de/kryptologie/modernechiffriersysteme/exkurs_diffie

[17] Teilweise auch **Diffie-Hellman-Schlüsselvereinbarung** genannt, schließlich werden weder Schlüssel ausgetauscht noch versendet.

[18] In der Realität ist die Wahl einer Primzahl diffiziler. Es sind nämlich nicht alle Primzahlen gleich sicher. Und da die Entscheidung, ob eine Primzahl gut geeignet ist, ziemlich rechenaufwändig ist, werden oft dieselben verwendet. Die Spezifikationen für IPsec geben beispielsweise eine Liste an Primzahlen vor. Wie so etwas aussieht, kann man unter https://datatracker.ietf.org/doc/html/rfc3526 sehen.

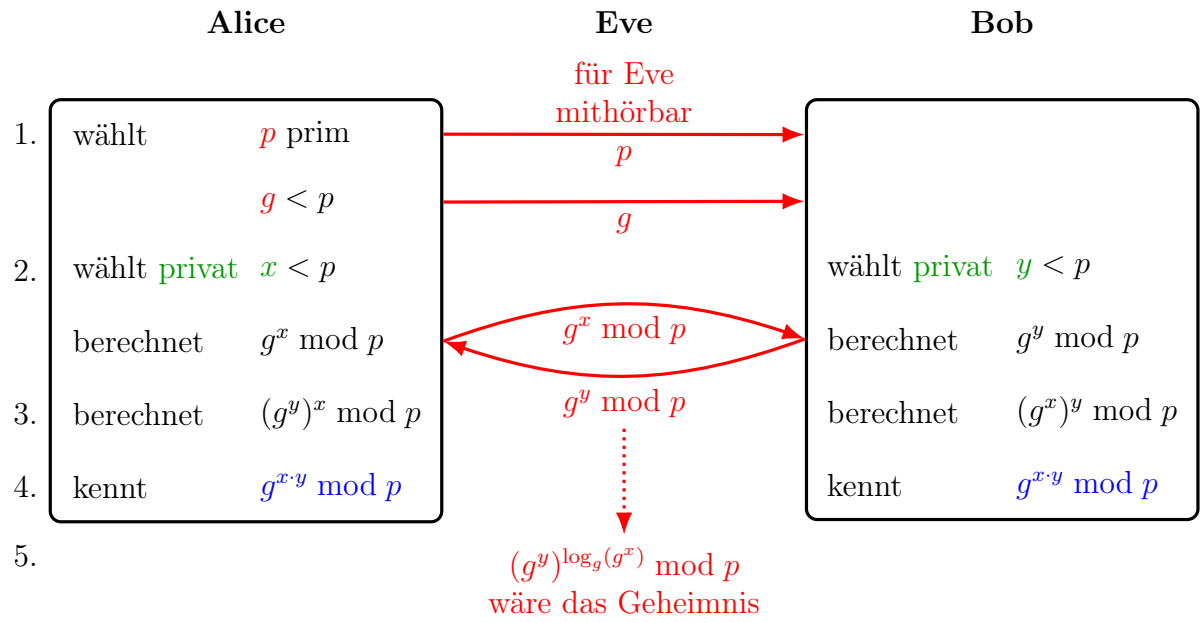

Abbildung 14: Der Diffie-Hellman-Schlüsselaustausch: Alice wählt die Zahlen p und g und verschickt sie über einen unsicheren Kanal an Bob. Beide berechnen die modularen Potenzen von g und einer privaten Hochzahl (grün) und tauschen sie aus. Die Potenz des anderen potenzieren sie mit ihrer eigenen privaten Hochzahl und kommen auf dasselbe Ergebnis (blau).
Lauscherin Eve ist nicht in der Lage, aus den öffentlich zugänglichen Informationen (rot) auf das Geheimnis zu schließen, weil sie in der Praxis nicht logarithmieren kann.

Aufgabe 45:
Führe gemeinsam mit deinem Sitznachbarn/deiner Sitznachbarin den Diffie-Hellman-Schlüsselaustausch durch. Ihr einigt euch auf $p = 23$ und $g = 5$. (Es ist ein Generator von $\mathbb{Z}_{23}^{\times}$.) Du und dein Sitznachbar/deine Sitznachbarin wählen jeweils $x < p$ und $y < p$ (ohne es dem/der anderen mitzuteilen), berechnen $g^x \bmod p$ bzw. $g^y \bmod p$ und teilen dem/der anderen dieses Ergebnis mit. Dann berechnet ihr beide das Geheimnis. Tragt die Zahlen in Abbildung 15 ein.

. . . . . .

. . . . . .

. . . . . .

Versuche danach, das Verfahren zu brechen, indem du in die Rolle eines Angreifers/einer Angreiferin schlüpfst. Berechne mit Hilfe der von dir abgehörten Zahlen $p = 23$, $g = 5$ und $5^y \bmod 23$ das Geheimnis. Wieso kann man diese Aufgabe stellen, während in der Realität ein Brechen nicht möglich ist?

. . . . . .

. . . . . .

. . . . . .

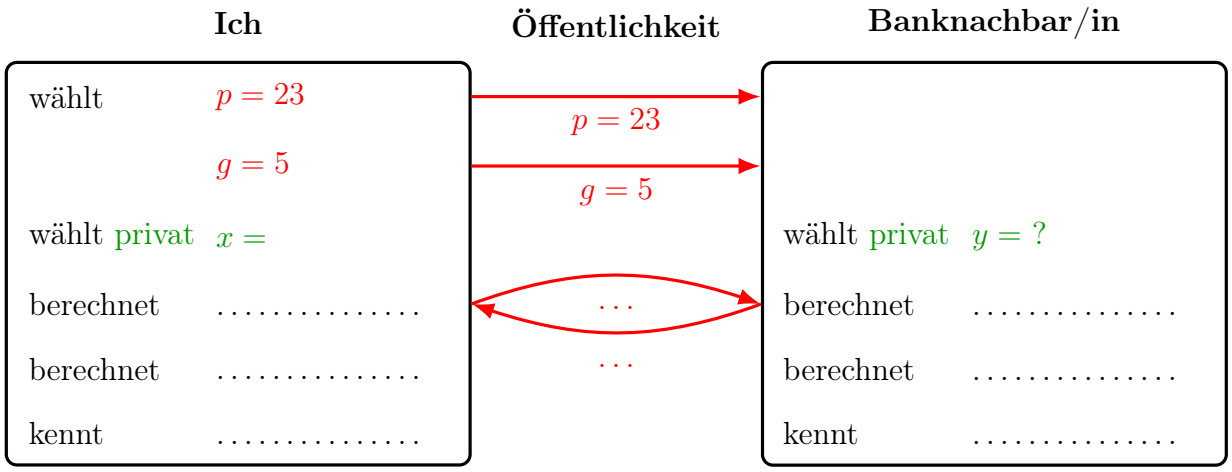

Abbildung 15: Der Diffie-Hellman-Schlüsselaustausch von Aufgabe 45 mit selbst gewählten Zahlen zur Demonstration

Diskussion

Der Diffie-Hellman-Schlüsselaustausch ermöglicht es nicht, direkt Informationen verschlüsselt zu übertragen (auch keinen Schlüssel). Aber er **generiert** für beide Parteien dasselbe Geheimnis. Dieses kann dann als Schlüssel für ein symmetrisches Verschlüsselungsverfahren dienen.

Im Gegensatz zu den meisten symmetrischen Verfahren ist beim Diffie-Hellman-Schlüsselaustausch die vollständige Schlüsselsuche (Brute-Force-Angriff) **nicht** der beste Angriff. Jedoch sind die bekannten Algorithmen trotzdem zu schwach, um Eve in der Praxis zu ermöglichen, das Geheimnis herauszufinden.

Der Diffie-Hellman-Schlüsselaustausch gilt als **sicher**, wenn g, x und y 1024 Bit-Zahlen sind.[19]

Generell sind asymmetrische Verfahren auf eine andere Art „sicher“ als symmetrische. Ihre Sicherheit ist nicht mathematisch bewiesen sondern beruhren auf der **Unbekanntheit** ausreichend schneller Algorithmen bzw. der Nichtexistenz des Quantencomputers.[20] Da

[19] Das gilt für herkömmliche Angreifer unterhalb des NSA-Kalibers. Es ist für häufig verwendete Primzahlen möglich, Dinge vorauszuberechnen. Dass das praktisch funktioniert, wurde 2015 für 512 Bit gezeigt. Die Autoren schätzten damals, dass dieselbe Vorgehensweise bei 1024 Bit 100 Mio. Dollar kosten würde, was für staatliche Angreifer in Reichweite wäre. Bei Verwendung von 2048 Bit wäre das bereits 10^9-mal aufwändiger. Alternativ könnte man auf Kryptographie auf *Elliptischen Kurven* ausweichen.

[20] Alle asymmetrischen Verfahren werden durch den Quantencomputer unsicher, weil dieser mit Hilfe des *Shor-Algorithmus* effizient Primfaktorzerlegungen und diskrete Logarithmen berechnen kann.

asymmetrische Verfahren schwierigere mathematische Methoden verwenden als symmetrische, sind sie naturgemäß anfälliger für Neuentdeckungen und Implementierungsschwächen. Nachweisbar sichere mathematische Probleme, die sich zur asymmetrischen Verschlüsselung einsetzen lassen, sind nicht bekannt.

Typischerweise haben asymmetrische Verfahren keine festen Schlüssellängen und sind **deutlich langsamer** als symmetrische, die nur einfache Funktionen (XOR, Bitshifts, Substitutionen, ...) benützen. (RSA benötigt mindestens 100-mal länger als AES!)

Ein mit dem Diffie-Hellman-Verfahren verwandtes stammt von ElGamal, welches in PGP (Pretty Good Privacy) verwendet wird. Es enthält zusätzlich einen probabilistischen Anteil, weshalb der Chiffretext doppelt so groß ist wie der Klartext.

Man-in-the-Middle-Angriff

Überhaupt keinen Schutz bietet das Verfahren gegen Man-in-the-Middle-Angriffe. Sollte es Mallory gelingen, zwischen Alice und Bob zu sitzen, kann sie ihnen vortäuschen, sie würden miteinander kommunizieren und Schlüssel austauschen. In Wahrheit führt sie aber zwei Schlüsselaustäusche mit jedem der beiden durch, siehe Abbildung 16.

Solange Mallory Nachrichten von Alice zuverlässig abfangen kann (ansonsten fliegt sie auf, weil Bob merkt, dass sie mit anderem Schlüssel verschlüsselt sind), kann sie sie entschlüsseln (weil sie mit Alice den Schlüssel $g^{x \cdot m} \bmod p$ teilt), lesen (oder verändern!), wieder verschlüsseln (weil sie mit Bob den Schlüssel $g^{y \cdot m} \bmod p$ teilt) und an Bob senden.

Verhindert werden kann das nur durch *Authentifizierung*.

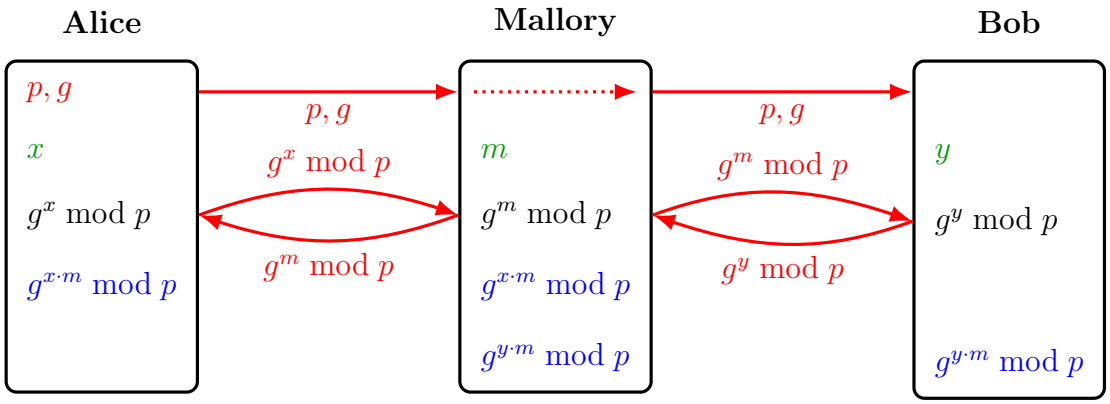

Abbildung 16: Wenn Mallory in der Lage ist, Datenpakete zwischen Alice und Bob abzufangen und zu verändern, dann kann sie mit beiden einen Diffie-Hellman-Schlüsselaustausch durchführen, während Alice und Bob glauben, mit dem jeweils anderen zu kommunizieren. Mallory kennt dann zwei Schlüssel: Einen, um mit Alice zu kommunizieren (ihre Nachrichten lesen und ihr Bobs Nachrichten, möglicherweise verändert oder überhaupt vorgetäuscht, schicken), und einen, um dasselbe mit Bob zu machen.

5 RSA

Das asymmetrische Kryptosystem *RSA* wurde 1977 von Ronald Linn Rivest, Adi Shamir und Leonard Max Adleman veröffentlicht. Während der Diffie-Hellman-Schlüsselaustausch auf der modularen Potenzierung als Einwegfunktion basiert, verwendet RSA als Falltürfunktion die Primzahlmultiplikation, nützt also die Schwierigkeit der Primfaktorzerlegung aus.

Das RSA-Verfahren zur Verschlüsselung

Wenn Alice von Bob verschlüsselte Nachrichten geschickt bekommen möchte, trifft sie folgende Vorbereitungen, siehe auch Abbildung 17:

(V1) Alice wählt zufällig und unabhängig[21] zwei Primzahlen p und q.[22]

(V2) Alice berechnet das Produkt $n = p \cdot q$ und veröffentlicht es. Diese Zahl n dient als Modul.

(V3) Alice berechnet $\varphi(n)$. Das ist leicht, weil sie die Primfaktorzerlegung kennt:

$$\varphi(n) = (p-1) \cdot (q-1)$$

(V4) Alice wählt eine zu $\varphi(n)$ teilerfremde Zahl e (mit $1 < e < \varphi(n)-1$) und veröffentlicht sie. Sie dient als Verschlüsselungsexponent (vgl. to **e**ncrypt).

(V5) Da e in $\mathbb{Z}_{\varphi(n)}$ invertierbar ist, lässt sich mit dem erweiterten euklidischen Algorithmus ein multiplikativ Inverses d (in $\mathbb{Z}_{\varphi(n)}$) berechnen. Dieses dient als Entschlüsselungsexponent (vgl. to **d**ecrypt) und bleibt privat.[23]

[21] Da es für das Verfahren essenziell ist, dass Eve die Primzahlen p und q nicht erfährt, ist es wichtig, dass Alice sie so unvorhersehbar wie möglich wählt.

[22] Herauszufinden, ob so große Zahlen Primzahlen sind, ist eine Herausforderung. Es gilt aber noch auf einige andere Dinge zu achten: Sind die Primzahlen zu unterschiedlich groß, kann Eve sie leichter herausfinden. Sind sie sich zu ähnlich, kann Eve das an ihrem Produkt n ablesen und weiß, wo sie zu probieren beginnen muss.

[23] Bis heute werden diese vorbereitenden Schritte so beschrieben. Allerdings wird heutzutage umgekehrt vorgegangen: Man legt zuerst e fest und sucht dann Primzahlen p und q, so dass $p-1$ und $q-1$ dazu teilerfremd sind. Dann ist e auch zu $\varphi(n) = (p-1) \cdot (q-1)$ teilerfremd.
Sehr häufig verwendet man für e die Primzahlen $3 = \texttt{11}$, $17 = \texttt{10001}$ oder $65\,537 = 2^{16} + 1 = \texttt{10000000000000001}$. Dann ist das Verschlüsseln wenig Aufwand, weil im Square-and-Multiply-Algorithmus wenige Multiplikationen zu erledigen sind. Andererseits muss Alice nur testen, dass $p, q \bmod e \neq 1$, was leichter geht als der erweiterte euklidische Algorithmus.
Teilweise bestehen Bedenken gegen $e < 65537$. Einerseits, weil für kleine Nachrichten m dann der Chiffretext m^e kleiner als n ist und deshalb der Modul n gar nicht benützt wird und so Muster im Chiffretext leichter erkennbar sind. Andererseits gibt es Angriffe, die funktionieren, wenn dieselbe Nachricht m an mehrere (mindestens e) Empfänger geschickt wird, die denselben Exponenten e benutzen.

Das Senden und Empfangen der Nachricht $m < n$ geschieht dann wie folgt:

(S1) Bob verschlüsselt m mittels $c = m^e \bmod n$ und sendet den Chiffretext an Alice.

(S2) Alice entschlüsselt c über $m = c^d \bmod n$.

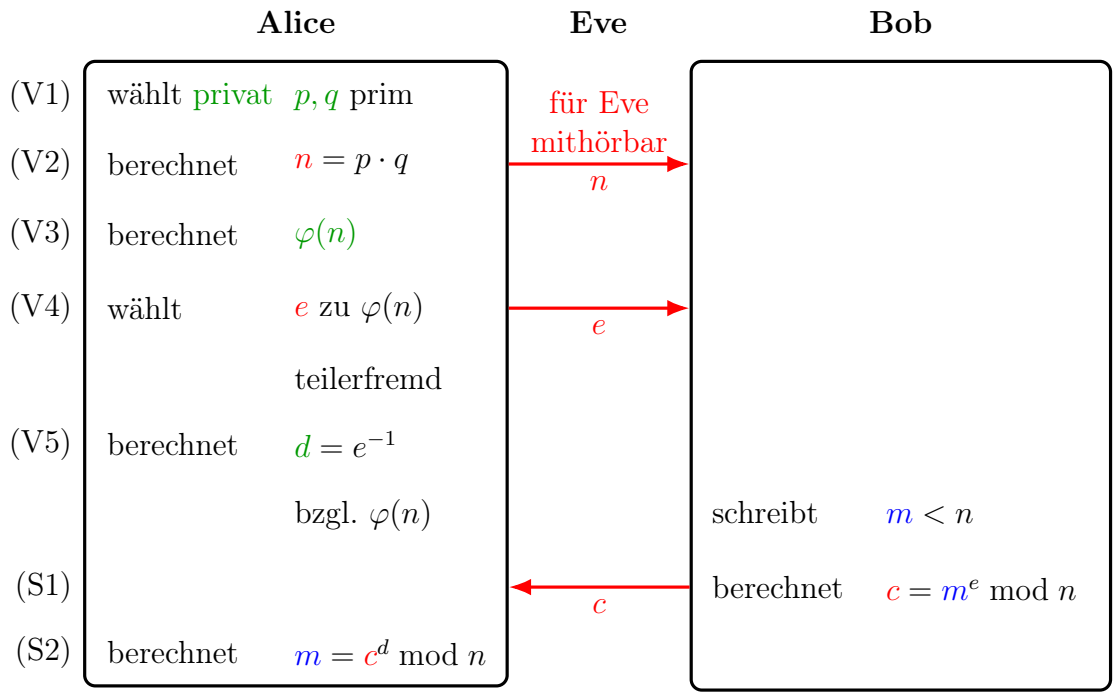

Abbildung 17: Die vorbereitenden Arbeiten der Empfängerin Alice, die Verschlüsselung einer Nachricht durch Bob und das Entschlüsseln der Nachricht durch Alice im RSA-Kryptosystem.
Der Schlüssel zur Entschlüsselung d sowie die Zahlen p, q und $\varphi(n)$ müssen von Alice geheimgehalten werden (grün). Letztere können nach der Vorbereitung vernichtet werden. Die versendeten Informationen n und e sind öffentlich und können genauso wie der Chiffretext c (rot) mitgehört werden. Aus diesen Informationen lässt sich die Nachricht m nicht gewinnen. Alice hingegen erlangt sie mit Hilfe ihres Schlüssels d. Alice und Bob kennen jetzt beide die Nachricht (blau).

Aufgabe 46: (Ein Zahlenbeispiel zur Demonstration)
Alice wählt die Primzahlen $p = 5$ und $q = 13$.

Sie berechnet $n =$ ..... und $\varphi(n) =$ .....

Sie wählt $e = 11$. Das ist möglich, weil .....

Alice veröffentlicht auf ihrer Homepage:

„Wer mir etwas schreiben will, soll es RSA-verschlüsseln mit .....“
Sie selbst berechnet für die Entschlüsselung

.....

.....

.....

.....
Bob will Alice eine Nachricht m schicken. Wähle dafür selbst eine Zahl, verschlüssle sie und lass Alice sie wieder entschlüsseln.

.....

.....

.....

.....

.....

Diskussion

Gleich wie beim Diffie-Hellman-Schlüsselaustausch gibt es auch beim RSA-Verfahren bessere Angriffe als Brute-force. Deshalb wird n deutlich länger gewählt als bei typischen symmetrischen Verfahren wie z. B. AES (1024 bis 2048 Bit gegenüber 256 Bit).

Um sich einen RSA-Modul vorstellen zu können, ist hier die Zahl `RSA-2048` abgedruckt. Für ihre Faktorisierung war bis 2007 ein Preisgeld von 200 000 $ ausgesetzt.

25195908475657893494027183240048398571429282126204032027777137836043662020
70759555626401852588078440691829064124951508218929855914917618450280848912
00728449926873928072877767359714183472702618963750149718246911650776133798
59095700097330459748808428401797429100642458691817195118746121515172654632
28221686998754918242243363725908514186546204357679842338718477444792073993
42365848238242811981638150106748104516603773060562016196762561338441436038
33904414952634432190114657544454178424020924616515723350778707749817125772
46796292638635637328991215483143816789988504044536402352738195137863656439
12120103971228221207203 57

Man muss hier beachten, dass es im Gegensatz zu symmetrischen Kryptosystemen

bei RSA trivialerweise die Möglichkeit zu Chosen-Plaintext-Angriffen gibt. Da e öffentlich ist, kann sich jeder beliebige Nachrichten verschlüsseln. Eine Möglichkeit wäre, alle m nacheinander zu verschlüsseln und mit dem abgefangenen Chiffretext c zu vergleichen.

Wenn Eve weiß, dass entweder „Ja“ oder „Nein“ geschickt wurde, ist „alle m durchprobieren“ sehr wenig Arbeit! Verhindern lässt sich ein Chosen-Plaintext-Angriff also nur, indem man ihn Eve zu teuer macht: Bei einem n mit 256 Bit muss Eve im Schnitt

..... Nachrichten durchprobieren (also auf Verdacht verschlüsseln). Das sind mehr, als es Atome im Universum gibt.

Weiters ist der *Faktorisierungsangriff* möglich, bei dem Eve versucht, die Primfaktorzerlegung des öffentlichen Moduls n zu finden. Dann könnte sie selbst $\varphi(n)$ berechnen, mit dem erweiterten euklidischen Algorithmus e invertieren und hätte den Schlüssel, um jede Nachricht zu entschlüsseln. Heutzutage muss man wohl 512 Bit-Zahlen als nicht mehr sicher gegen diesen Angriff ansehen.

Würde man RSA so einsetzen, dass eine Nachricht beispielsweise Buchstabe für Buchstabe verschlüsselt wird, wäre das natürlich völlig unsicher!

.....

.....

.....

.....

.....

.....

.....

.....

.....

.....

In der Praxis erfolgen Angriffe auf RSA oft über Seitenkanäle.

Wieso funktioniert das Entschlüsseln von RSA?

Da d das Inverse zu e in $\mathbb{Z}_{\varphi(n)}$ ist, gilt

$$d \cdot e = 1 + k \cdot \varphi(n) = 1 + k \cdot (p-1) \cdot (q-1), \quad k \in \mathbb{Z}.$$

Wenn Alice den Chiffretext c von Bob mit ihrem privaten Schlüssel d potenziert, erhält sie

$$c^d \bmod n = (m^e)^d \bmod n = m^{e \cdot d} \bmod n =$$
$$= m^{1+k \cdot (p-1) \cdot (q-1)} \bmod n = m \cdot m^{(p-1) \cdot (q-1) \cdot k} \bmod n.$$

Sollte die Nachricht m teilerfremd zu $n = p \cdot q$ sein, folgt die Aussage aus dem Satz von Euler, weil dann $m^{\varphi(n)} \equiv 1 \bmod n$ ist:

$$m \cdot m^{(p-1) \cdot (q-1) \cdot k} \bmod n = m \cdot 1^k \bmod n = m$$

Wenn m nicht teilerfremd zu n ist, ist sie ein Vielfaches genau eines Primfaktors ($m < n$!) – wir nehmen an von p, aber nicht von q. Dann folgt aus dem Satz von Euler

$$m^{\varphi(q)} \equiv 1 \bmod q \Longrightarrow m^{(p-1) \cdot (q-1) \cdot k} \equiv 1 \bmod q \Longrightarrow m \cdot m^{(p-1) \cdot (q-1) \cdot k} \equiv m \bmod q,$$

somit ist $\left(m \cdot m^{(p-1) \cdot (q-1) \cdot k} - m\right)$ ein Vielfaches von q. Da m ein Vielfaches von p ist, ist auch $\left(m \cdot m^{(p-1) \cdot (q-1) \cdot k} - m\right)$ ein Vielfaches von p. Da p und q prim sind, muss dann $\left(m \cdot m^{(p-1) \cdot (q-1) \cdot k} - m\right)$ auch ein Vielfaches von $p \cdot q$ sein. Schlussendlich heißt das

$$\left(m \cdot m^{(p-1) \cdot (q-1) \cdot k} - m\right) \equiv 0 \bmod n$$
$$m \cdot m^{(p-1) \cdot (q-1) \cdot k} \equiv m \bmod n$$

Aufgabe 47:
Begründe, warum der öffentliche RSA-Schlüssel ($n = 143, e = 15$) nicht möglich ist.

. . . . . .

Aufgabe 48:
Alice hat den öffentlichen Schlüssel ($n = 221, e = 7$). Du, Eve, bekommst die verschlüsselte Nachricht $c = 2$ in die Hände. Knacke die Verschlüsselung und finde die Klartextnachricht heraus.

. . . . . .

Aufgabe 49:
Alice hat den öffentlichen Schlüssel ($n = 221, e = 5$). Du, Bob, willst ihr die Nachricht FOTO schicken. Die Buchstaben werden alphabetisch codiert (Leerzeichen$\rightarrow 00$, $A \rightarrow 01$, $B \rightarrow 02$,..., $Z \rightarrow 26$). Berechne den zu sendenden Chiffretext.

. . . . . .

. . . . . .

. . . . . .

Erkläre, welche Schwachstelle dieses Verschlüsselungsschema hat.

. . . . . .

. . . . . .

Jetzt werden die Buchstaben so codiert, dass jeder Buchstabe zwei Ziffern liefert. Die Ziffernkette wird in Blöcke der Länge drei unterteilt und dann verschlüsselt. (Beispiel: HUND $\rightarrow$ 08 21 14 04 $\rightarrow$ 082 114 040 $\rightarrow$ verschlüsseln.) Berechne den zu sendenden Chiffretext.

. . . . . .

. . . . . .

. . . . . .

Aufgabe 50:
Alice hat den öffentlichen Schlüssel ($n = 1271, e = 13$). Du, Bob, willst ihr die Nachricht NICHT ANNEHMEN schicken. Die Buchstaben werden ASCII-codiert (Leerzeichen$\rightarrow$ 32, $A \rightarrow 65$, $B \rightarrow 66$,..., $Z \rightarrow 90$). Die Ziffernkette wird in Blöcke der Länge drei unterteilt (evtl. Nullen am Ende anhängen, um einen vollen Block zu bekommen) und dann verschlüsselt. Berechne den zu sendenden Chiffretext.

. . . . . .

. . . . . .

. . . . . .

. . . . . .

Aufgabe 51:
Alice hat den öffentlichen Schlüssel ($n = 1333, e = 13$). Du, Bob, willst ihr die Nachricht RSA schicken. Die Buchstaben werden alphabetisch codiert (Leerzeichen$\rightarrow$ 00, $A \rightarrow 01$, $B \rightarrow 02$,..., $Z \rightarrow 26$), so dass jeder Buchstabe zwei Ziffern liefert. Die Ziffernkette wird in Blöcke der Länge drei unterteilt und dann verschlüsselt. (Beispiel: HUND $\rightarrow$ 08 21 14 04 $\rightarrow$ 082 114 040 $\rightarrow$ verschlüsseln.)

Berechne den zu sendenden Chiffretext.

. . . . . .

. . . . . .

. . . . . .

Aufgabe 52:
Alice hat den öffentlichen Schlüssel ($n = 1763, e = 19$). Du, Eve, bekommst die verschlüsselte Nachricht $(118, 462, 65, 54)$ in die Hände. Die Buchstaben wurden alphabetisch codiert (Leerzeichen$\rightarrow$ 00, $A \rightarrow 01$, $B \rightarrow 02$,..., $Z \rightarrow 26$), so dass jeder Buchstabe zwei Ziffern liefert. Die Ziffernkette wurde in Blöcke der Länge drei unterteilt und dann verschlüsselt. (Beispiel: HUND $\rightarrow$ 08 21 14 04 $\rightarrow$ 082 114 040 $\rightarrow$ verschlüsseln.)

Knacke die Verschlüsselung und finde die Klartextnachricht heraus.

. . . . . .

. . . . . .

. . . . . .

. . . . . .

Aufgabe 53:
Alice hat den öffentlichen Schlüssel ($n = 1073, e = 25$). Du, Eve, bekommst die verschlüsselte Nachricht $(6, 337, 744)$ in die Hände. Die Buchstaben wurden alphabetisch codiert (Leerzeichen$\rightarrow$ 00, $A \rightarrow 01$, $B \rightarrow 02$,..., $Z \rightarrow 26$), so dass jeder Buchstabe zwei Ziffern liefert. Die Ziffernkette wurde in Blöcke der Länge drei unterteilt und dann verschlüsselt. (Beispiel: HUND $\rightarrow$ 08 21 14 04 $\rightarrow$ 082 114 040 $\rightarrow$ verschlüsseln.)

Knacke die Verschlüsselung und finde die Klartextnachricht heraus.

. . . . . .

. . . . . .

. . . . . .

. . . . . .

Das RSA-Verfahren lässt sich aber auch (und das ist häufiger) zu einem anderen Zweck einsetzen, nämlich zur *Authentifizierung*: Alice „unterschreibt“ eine Datei, indem sie sie mit ihrem privaten Schlüssel verschlüsselt. Bob kann sie (wie jedeR andere) mit dem öffentlichen Schlüssel entschlüsseln und damit lesen. Außerdem weiß er so, dass die Datei tatsächlich von Alice stammt.

In Kombination mit einer *Public Key Infrastructure (PKI)* lassen sich so *Man-in-the-Middle-Angriffe* vermeiden.

6 Das One-Time-Pad

Nach den asymmetrischen Verfahren, die es auch ohne im Vorfeld vereinbarte Schlüssel ermöglichen, verschlüsselt zu kommunizieren, kommen wir nun zu symmetrischen Kryptosystemen, die deutlich schneller in der Anwendung sind.

Caesar-Verfahren

Das Caesar-Verfahren ist eines der ältesten und bekanntesten symmetrischen Kryptosysteme. Leider verletzt es das *Prinzip von Kerckhoff*, ist also sehr unsicher gegen Angreifer:innen, die das Verfahren kennen.

Das liegt einerseits am viel zu *kleinen Schlüsselraum*, der

.....

ermöglicht, und andererseits daran, dass man durch Erraten von nur einem Zeichen Klartext

.....

Wie die meisten alten Chiffren ist das Caesar-Verfahren also sehr anfällig gegen

.....

Monoalphabetische Substitution

Als Verallgemeinerung des Caesar-Verfahrens kann man die monoalphabetische Substitution ansehen. Sie scheint auf den ersten Blick die Schwachpunkte des Caesar-Verfahrens abzumildern:

- Statt der 26 Schlüssel gibt es .....
- Kennt man einen Teil des Klartextes, erfährt man, durch welche Buchstaben *diese* Klartextbuchstaben ersetzt wurden. Die Substitutionsvorschrift für die restlichen noch nicht.

Allerdings ist die Anzahl der Möglichkeiten, die restlichen Buchstaben zu substituieren, deutlich kleiner. Über Häufigkeitsanalysen ist das Verfahren leicht zu brechen.

Vigenère-Verfahren

Alle *monoalphabetischen* Verfahren haben das Problem, dass ein Klartextzeichen immer durch dasselbe Chiffretextzeichen ersetzt wird. Dadurch werden die Häufigkeiten der Klartextzeichen nicht verwischt.

Dieses Problem wird beim Vigenère-Verfahren behoben, das *polyalphabetisch* ist. Das heißt, für Klartextzeichen eins wird eine andere Substitutionstabelle („ein anderes Alphabet“) verwendet als für Klartextzeichen zwei usw. Damit der Schlüssel darüber hinaus merkbar bleibt, verwendet man ein Schlüssel*wort*, das immer wiederverwendet wird.

Mit dem *Kasiski-Angriff* lässt sich aber relativ einfach die Schlüssellänge herausfinden, mit Häufigkeitsanalysen die einzelnen Buchstaben des Schlüsselwortes.

Vernam-Verfahren

Das Problem der Wiederverwendung des Schlüsselwortes wird durch das Vernam-Verfahren gelöst. Es verwendet ein Schlüssel„wort“, das gleich lang ist wie der zu verschlüsselnde Klartext. Da das nicht merkbar ist, verwendet man typischerweise den Text eines leicht verfügbaren Buches (Telefonbuch, Bibel etc.).

Die Schlüsselraumgröße, um einen Klartext der Länge n zu verschlüsseln, ist

. . . . . .

also viel zu groß für einen Brute-Force-Angriff. Eher würde man selbstverständlich

. . . . . .

. . . . . .

Die einzigen Angriffsmöglichkeiten ergeben sich daraus, dass sowohl im Klartext als auch im Schlüssel manche Zeichen, Zeichenkombinationen und Muster häufiger auftreten.

Das One-Time-Pad

Um auch noch diese verbleibende Schwäche des Vernam-Verfahrens auszumerzen, ist die letzte Idee, eine **völlig zufällige** Kette aus **unabhängigen** und **gleichverteilten** Zeichen als Schlüssel zu verwenden. Die einzelnen Zeichen sind dann der Schlüssel für eine Caesar-Verschiebung für **ein** Zeichen des Klartextes. **Kein Schlüssel-Zeichen wird jemals wieder verwendet, daher der Name!**

Zur Demonstration werden nur die Großbuchstaben ($A \to 00$, $B \to 01$,…, $Z \to 25$) und die Addition modulo 26 verwendet.

Klartext:	`ABEND`	=	00	01	04	13	03	
Schlüssel:	`EOVKL`	=	04	14	21	10	11	$\oplus_{26}$
Chiffretext:	`EPZXO`	=	04	15	25	23	14	

Der Klartextbuchstabe `N` entspricht $m_4 = 13$. Zu diesem Klartextzeichen wird das vierte Zeichen des Schlüssels (`K` bzw. $k_4 = 10$) addiert, so dass das entsprechende Chiffretextzeichen $c_4 = (13 + 10) \bmod 26 = 23$ (bzw. `X`) ist.

Natürlich kommen im Klartext `E`, `R`, `N` usw. häufiger vor als `Q` und `Y`. Aber da jede Zahl von 00 bis 25 als k_4 **gleich wahrscheinlich** ist,

. . . . . .

. . . . . .

Würde der Klartext stattdessen `FRUEH` lauten, käme beim Schlüssel von oben (`EOVKL`) natürlich ein anderer Chiffretext heraus. **Aber:** Es gäbe ebenso eine Schlüssel (nämlich `ZYFTH`), der den Klartext `FRUEH` zum Chiffretext von oben `EPZXO` verschlüsseln würde.

Klartext:	`FRUEH`	=	05	17	20	04	07	
Schlüssel:	`ZYFTH`	=	25	24	05	19	07	$\oplus_{26}$
Chiffretext:	`EPZXO`	=	04	15	25	23	14	

Klartext:	`WARTE`	=	22	00	17	19	04	
Schlüssel:	`IPIEK`	=	08	15	08	04	10	$\oplus_{26}$
Chiffretext:	`EPZXO`	=	04	15	25	23	14	

Das hat zur Folge, dass Eve, die den Chiffretext `EPZXO` abgefangen hat, nicht wissen kann, welcher der (gleich wahrscheinlichen!) Schlüssel (`EOVKL`, `ZYFTH`, `IPIEK`, . . .) zu diesem Chiffretext geführt hat und damit auch nicht, welcher Klartext dahinter steckt (`ABEND`, `FRUEH`, `WARTE`, `PASSE`, `WAFFE`, `NICHT`, `DURST`, `FRODO`, . . .).

Diskussion des One-Time-Pads

Unter der Voraussetzung, dass der Schlüssel mit guten Zufallszahlengeneratoren erzeugt wird, ist das One-Time-Pad *perfekt sicher*. Es sind überhaupt keine Angriffe darauf möglich. Selbst ein Chosen-Plaintext-Angriff offenbart mir nur den Teil des Schlüssels, mit dem dieser vorgegebene Klartext verschlüsselt wurde. Dieser wird aber nie mehr verwendet!

Ein Brute-Force-Angriff ist praktisch unmöglich wegen der Zahl der Möglichkeiten, aber auch theoretisch nutzlos, weil es für *jeden* Text dieser Länge einen Schlüssel gibt, der daraus diesen Chiffretext gemacht hätte!

Würde Eve also mittels Brute-Force Schlüssel durchprobieren und auf einen plausiblen Klartext stoßen

. . . . . .

. . . . . .

Laien vermuten ja oft, dass Kryptographie mit komplexer Mathematik zu tun hat. Das One-Time-Pad zeigt, dass sogar *perfekte Sicherheit* mit der einfachsten mathematischen Operation (XOR) erlangt werden kann.[24]

Das Unternehmen *Mils Electronic* war auf die Herstellung von One-Time-Pad-Hardwarelösungen spezialisiert.[25]

Wenn es ein *perfekt sicheres* Verfahren gibt, warum wird es dann nicht verwendet?

- Weiterhin ungelöst ist, wie bei allen symmetrischen Verfahren, der

  .....

  .....

  .....

  .....

  .....

- Die Erzeugung guter Zufallszahlen ist schwerer als man vermuten würde.
- Gehen Nachrichten verloren, hat man das Problem der Synchronisation. Der Empfänger muss genau wissen, zu welchem Schlüsselzeichen das Chiffretextzeichen gehört.

Aufgabe 54:
Begründe, wieso bei jedem Zeichen wieder nur Caesar-Verschiebungen verwendet werden und keine allgemeinen monoalphabetischen Substitutionen.

.....

.....

Aufgabe 55:
Begründe, wieso die Zeichen des Schlüssel nicht wiederverwendet werden dürfen.

.....

.....

.....

Begründe, wieso die Zeichen des Schlüssel perfekt zufällig und unabhängig sein müssen.

[24] Der Beweis der perfekten Sicherheit erfordert natürlich Mathematik, der Entwurf und das Testen von Zufallszahlengeneratoren ebenfalls uvm.

[25] https://www.cryptomuseum.com/manuf/mils/index.htm

7 Design moderner Blockchiffren

Von einer Blockchiffre spricht man, wenn sowohl Klar- als auch Chiffretext Blöcke einer gewissen, festen Länge sind (typisch: 64 oder 128 Bit).

Wir starten mit ein paar grundsätzlichen Überlegungen zum Entwurf von Kryptosystemen.

- Der nicht empfehlenswerte Ansatz *Security by intricacy* (Sicherheit durch Komplexität) beruht darauf, die Chiffre so kompliziert zu machen, dass Mallory hoffentlich keine Schwachstellen findet.
- Alternativ baut man die Chiffre so systematisch auf, dass man genau weiß, welcher Teil was bewirkt.

Alle anerkannten Verfahren folgen der zweiten Strategie und sind außerdem öffentlich (folgen also dem *Kerckhoff'schen Prinzip* statt *Security by obscurity* (Sicherheit durch Unklarheit)).

Von symmetrischen Verfahren erwartet man sich heutzutage, dass es keinen besseren als einen Brute-force-Angriff gibt. Ciphertext-only-Attacken sind gegen moderne Verfahren aussichtslos, nur Chosen- und Known-Plaintext-Angriffe sind teilweise bekannt.

Idealerweise sind sie *Zufallsorakel*. Das heißt, dass kein erkennbarer Zusammenhang zwischen Eingabe (Klartext und Schlüssel) und Ausgabe (Chiffretext) existiert. (Das wäre bereits der Fall, wenn bei Verwendung des Schlüssels

`0000...0`

das letzte Bit des Chiffretextes zu 60 % eine `1` ist.) Anders ausgedrückt: Die Änderung eines Klartext-Bits soll durchschnittlich die Hälfte der Bits der Ausgabe ändern.

Selbst der DES erfüllt das allerdings nicht perfekt (u. a.: verschlüsseltes Komplement ist Komplement des Verschlüsselten).

Aufgabe 56:
Gibt es eine Chiffre, die ein perfektes Zufallsorakel ist?

Man legt eine *zufällige* angelegte „Zuordnungstabelle“ für 64-Bit-Blöcke des Klartexts auf 64-Bit-Blöcke des Chiffretexts fest. Das entspricht einer

$$\begin{array}{ccc} \text{Klartext} & & \text{Chiffretext} \\ \begin{bmatrix} 0 & 0 & 0 \\ 0 & 0 & 0 \\ 0 & 0 & 0 \end{bmatrix} & \rightarrow & \begin{bmatrix} 1 & 0 & 1 \\ 0 & 1 & 0 \\ 0 & 1 & 1 \end{bmatrix} \\ \begin{bmatrix} 0 & 0 & 0 \\ 0 & 0 & 0 \\ 0 & 0 & 1 \end{bmatrix} & \rightarrow & \begin{bmatrix} 0 & 1 & 1 \\ 1 & 0 & 0 \\ 0 & 0 & 0 \end{bmatrix} \\ \vdots & & \vdots \end{array} \qquad \begin{array}{ccc} \text{Klartext} & & \text{Chiffretext} \\ \begin{bmatrix} 0 & 0 & 0 \\ 0 & 0 & 0 \\ 0 & 1 & 0 \end{bmatrix} & \rightarrow & \begin{bmatrix} 1 & 1 & 0 \\ 0 & 0 & 1 \\ 1 & 1 & 1 \end{bmatrix} \\ \begin{bmatrix} 0 & 0 & 0 \\ 0 & 0 & 0 \\ 0 & 1 & 1 \end{bmatrix} & \rightarrow & \begin{bmatrix} 0 & 0 & 1 \\ 1 & 0 & 0 \\ 1 & 0 & 0 \end{bmatrix} \\ \vdots & & \vdots \end{array}$$

Allerdings ist der „Schlüssel“

. . . . . .

. . . . . .

Von Claude Shannon stammen die realistischeren Forderungen nach:

- **Konfusion**: Die Verteilung der Chiffretexte hängt so kompliziert von Verteilung der Klartexte ab, dass ein Angreifer sie nicht nutzen kann. (Die *monoalphabetische Substitution* erfüllt das nicht, da die Häufigkeitsverteilung der Klartextzeichen sich auf die der Chiffretextzeichen überträgt.)
- **Diffusion**: Jedes Bit des Klartextes und jedes des Schlüssels beeinflusst möglichst viele Bit des Chiffretextes. (Nicht: Das erste Schlüssel-Bit spielt nur im ersten Chiffretext-Byte eine Rolle.)

Nebenerfordernisse sind die effiziente (arbeitsspeicher-, zeit- und stromsparende) Umsetzbarkeit in Software und Hardware.

Deshalb werden einfache Operationen verwendet:

. . . . . .

. . . . . .

Ebenso Bitpermutationen (wie sie in DES viel vorkommen): Sie sind in Hardware einfach umsetzbar (Bit-Leitungen entsprechend legen!), in Software

. . . . . .

Die *Art und Reihenfolge der Operationen* soll **nicht** vom Schlüssel oder Klartext abhängen, weil sonst Seitenkanalangriffe (über den Stromverbrauch oder die Rechenzeit) möglich wären.

Erreicht werden diese Ziele typischerweise als **iterierte Ausführung gleichartiger Runden** (das spart Speicherplatz). In die Runden gehen unterschiedliche **Rundenschlüs-**

sel ein, die aus dem Hauptschlüssel abgeleitet werden. Außerdem beinhalten sie **Permutationen**, die für die Diffusion sorgen, und **Substitutionen** („**S-Boxen**“), die die Chiffre nichtlinear machen. (Lineare Chiffren sind anfällig gegenüber Known-Plaintext-Angriffen, welche auf das Lösen linearer Gleichungssysteme hinauslaufen.)

Theoretisch könnte man ausschließlich S-Boxen verwenden, allerdings bräuchte man dann sehr große, siehe Aufgabe 56. Deshalb verwendet man kleine Boxen öfters mit Durchmischung dazwischen.

Was ist mit Linearität gemeint? Das versteht man am besten, wenn man sich in Abbildung 18 ansieht, wie sich „einfache“ Operationen wie Linksrotationen oder Bit-Permutationen auswirken. Dem ist der Ausschnitt einer S-Box des DES gegenübergestellt: Der Zusammenhang zwischen Eingang und Ausgang ist hier viel weniger „systematisch“.

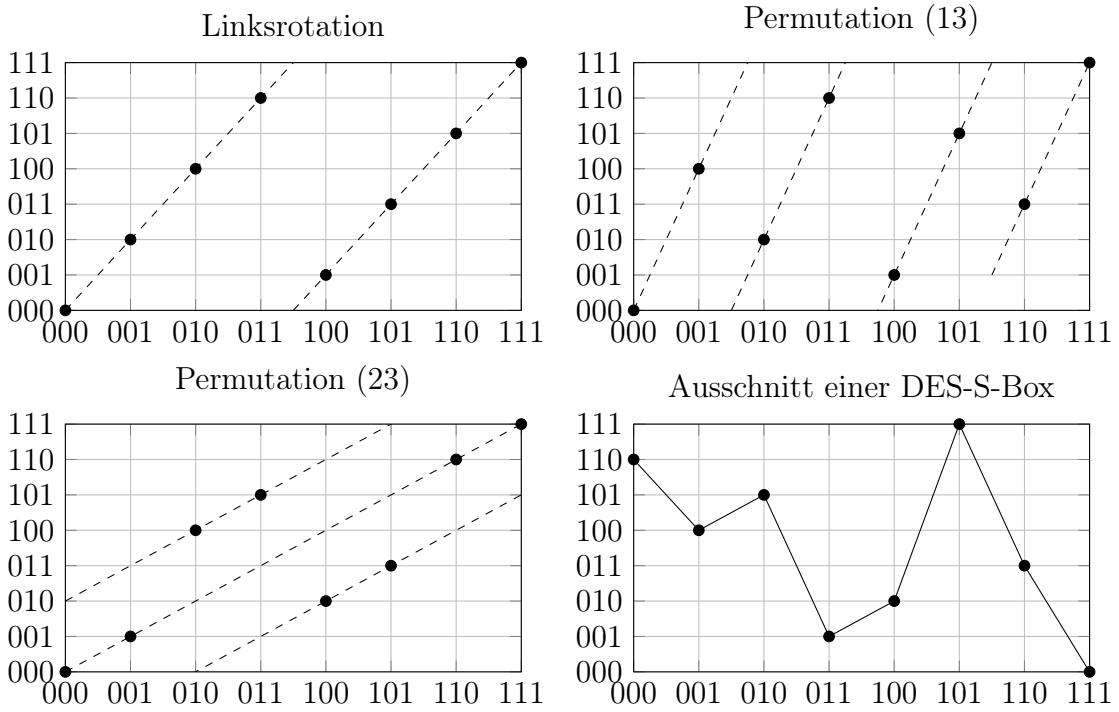

Abbildung 18: Die Auswirkungen von Bit-Linksrotation und Bit-Permutation und einer S-Box des DES auf 3 Bit-Stücke, aus K. Schmeh; Kryptografie: Verfahren, Protokolle, Infrastrukturen (6. Auflage). 2016

7.1 Data Encryption Standard (DES)

Der DES stammt aus dem Jahr 1977 und wurde unter Beteiligung der NSA entwickelt (was zu manchen Verschwörungsmythen geführt hat). Er wird weiterhin am häufigsten von Banken in Chipkartenanwendungen eingesetzt[26].

[26] https://de.wikipedia.org/wiki/Data_Encryption_Standard

Er basiert auf einfachen Operationen und ist vor allem für die Implementierung in Hardware prädestiniert, in Software ist er etwas langsam.

Klartext- und Chiffretext sind 64-Bit-Blöcke. Der Schlüssel theoretisch auch, enthält aber acht Prüfbits, weshalb der Schlüssel effektiv 56 Bit hat, was

.....

Der DES ist eine *Feistel-Chiffre* mit 16 Runden. Damit sind Ver- und Entschlüsseln dasselbe (Nur die abgeleiteten Rundenschlüssel müssen in gestürzter Reihenfolge verwendet werden.) In den Runden kommen **unveränderliche** (auf Sicherheit untersuchte!) S-Boxen und Permutationsschritte vor. (Die S-Boxen haben u. a. die Eigenschaft, dass bei einem Bit Unterschied im Eingang mindestens zwei Bit im Ausgang verschieden sind. So wächst bei zwei Nachrichten, die sich in nur einem Bit unterscheiden und mit dem gleichen Schlüssel verschlüsselt werden, die Anzahl unterschiedlicher Bits exponentiell mit der Rundenzahl.)

Der Schlüssel geht ausschließlich mit XOR ein .....

Wie sicher ist der DES? Dazu betrachten wir zwei mögliche Angriffe und wie man ihnen begegnet, auch um ein Verständnis für die Bit-Zahl bei Blockchiffren zu bekommen.

Der Schlüsselraum

Schon zum Zeitpunkt der Veröffentlichung wurde die Schlüssellänge von 56 Bit als zu schwach kritisiert. Damals wären ein ..... bereits in Reichweite der NSA gewesen. Dieser Angriff läuft darauf hinaus, alle Schlüssel der Reihe nach durchzuprobieren bis ein sinnvoller Klartext herauskommt. Dazu muss man natürlich wissen,

.....

.....

Dazu muss Eve .....-mal entschlüsseln und das Ergebnis auf Sinnhaftigkeit untersuchen. Bereits 1999 wurde mit der Maschine Deep Crack und einem Netzwerk aus 100 000 Rechnern eine Challenge in 22 h gewonnen. Seit damals kann man DES nicht mehr als wirklich sicher ansehen.

Aufgabe 57:
Lässt sich diese Schwäche durch längere Schlüssel beheben?

.....

Rechnen wir grob hoch, wie lange man für die folgenden Schlüssellänge benötigen würde.

Schlüssellänge	Größe des Schlüsselraumes	hochgerechnete Dauer
56 Bit	$7{,}2 \cdot 10^{16}$	22 h
64 Bit		
128 Bit		
256 Bit		

Berechne, wie oft das geschätzte Alter des Universums (13 Milliarden Jahre) in dieser Zeitspanne Platz hat (wenn sich die Anzahl der zur Verfügung stehenden Computer vertausendfacht und ihre Geschwindigkeit verhundertfacht hat).

.....

Gegenüber einem Quantencomputer (*Grover-Algorithmus*) halbiert sich allerdings das Sicherheitsniveau, d. h. 256 Bit sind nur so sicher wie 128 Bit auf klassischen Computern.

In [27] wird auch vorgerechnet, dass zum Knacken des 128 Bit-Schlüssels mit aktuellen Computern $9{,}4 \cdot 10^{19}$ kWh nötig wären, was hundertmal mehr ist, als alle Kraftwerke in einem Jahr erzeugen. Bei 0,20 €/(kW h) müsste einem diese Attacke

.....

Die Blockgröße

Verwendet man DES so, dass die Nachricht in 64-Bit-Blöcke unterteilt wird, die der Reihe nach verschlüsselt werden, handelt es sich im Prinzip um eine

.....

weshalb man .....

fürchten müsste. Geht das auch praktisch?

- **Einserseits:** Die 64 Bit entsprechen ..... verschiedenen Klartexten. Kommen alle gleich oft vor, wiederholt sich ein Block nach

  .....

  Man würde also diese Datenmenge benötigen, damit sich die ersten Blöcke wiederholen und man langsam beginnen kann, Häufigkeiten zu analysieren.

[27] K. Schmeh; Kryptografie: Verfahren, Protokolle, Infrastrukturen (6. Auflage). 2016

- **Andererseits:** Ein 64-Bit-Block Klartext entspricht auch nur ..... ASCII-Zeichen! Kommen ebenso lange Buchstaben-Kombinationen im Klartext öfters vor, dann auch im Chiffretext. Allerdings nur ein Achtel so oft, weil die Blockgrenze zu $\frac{7}{8}$ woanders liegt.

  ```
  |Gandalf |wartete |auf
  |agte Gan|dalf, ob| es
  |Frodo, G|andalf n|ahm
  |dalf fra|gte Saru|man
  | Bilbo, |Gandalf |und
  ```

  Aber selbst wenn mir eine 64-Bit-Folge öfters im Chiffretext auffällt, ist es schwer, ihr das zugehörige Klartext-Wort zuzuordnen. (Eine Statistik über 26 Buchstaben aufstellen ist einfach. Aber über 8-Buchstaben-Kombinationen?)

Nichtsdestoweniger verwenden neuere Verfahren 128-Bit-Blöcke.
Resümee: Nicht nur die Schlüssellänge ist von Bedeutung sondern auch die Blockgröße – und zwar gegen Häufigkeitsanalysen!

Die Sicherheit des DES ist gut erforscht, bislang wurde kein besserer Angriff als das Durchsuchen des Schlüsselraumes gefunden. Andererseits ist dieser mit 56 Bit recht klein. Heutzutage ist DES in Stunden knackbar.

7.2 Triple-DES, 3DES, DESede

Der DES ist bis auf seinen kleinen Schlüsselraum eigentlich gut designt. Auch nach einem halben Jahrhundert kennt man noch keine Schwachstellen. Wäre es nicht möglich, einen Text mit zwei verschiedenen Schlüsseln hintereinander zu verschlüsseln? Dann müsste der Angreifer

.....
Schlüssel durchprobieren, siehe Abbildung 19. Das stimmt nur, wenn zweimal verschlüsseln „mehr macht“ als einmal verschlüsseln.

Aufgabe 58:
Nenne Demonstrationsbeispiele für Verschlüsselungen, bei denen die zweifache Anwendung „etwas Neues“ hervorbringt, und solche, für die das nicht der Fall ist.

.....

.....

.....

.....

. . . . . .

. . . . . .

. . . . . .

Erst 1992 wurde gezeigt, dass DES unter Hintereinanderausführung nicht abgeschlossen ist. Das heißt, dass es für zwei Schlüssel $K1$ und $K2$ im Allgemeinen keinen Schlüssel $K3$ gibt, so dass

$$\mathrm{DES}_{K1}(\mathrm{DES}_{K2}(m)) = \mathrm{DES}_{K3}(m).^{28}$$

Wieso also nicht DES einfach zweimal anwenden?
Wegen der Möglichkeit eines **Meet-in-the-Middle-Angriffes**:

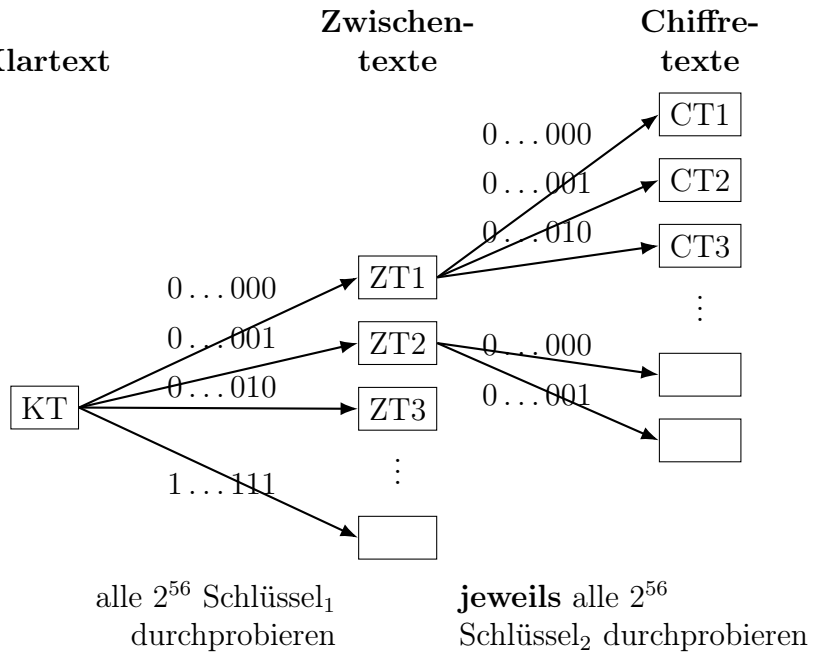

Abbildung 19: Zweifache Anwendung des DES auf einen Klartext: Es gibt 2^{56} Schlüssel für die erste Verschlüsselung und damit 2^{56} verschiedene mögliche Zwischentexte. Diese können jeweils wieder mit 2^{56} verschiedenen Schlüsseln verschlüsselt werden. Dadurch entstehen 2^{128} Chiffretexte.
Eve könnte umgekehrt zu einem abgehörten Chiffretext alle $2^{56} \cdot 2^{56} = 2^{112}$ Schlüssel durchprobieren und jeweils überprüfen, ob das Ergebnis des Entschlüsselns sinnvoll aussieht.

[28] Es ist bewiesen, dass die Gruppenordnung der Verschlüsselungsfunktionen (Anzahl aller mit mehreren DES-Schritten erreichbaren Verschlüsselungen) mindestens 10^{2499} ist, während ja $2^{56} = 7 \cdot 10^{17}$ und sogar $2^{168} = 3{,}7 \cdot 10^{50}$ deutlich weniger ist.

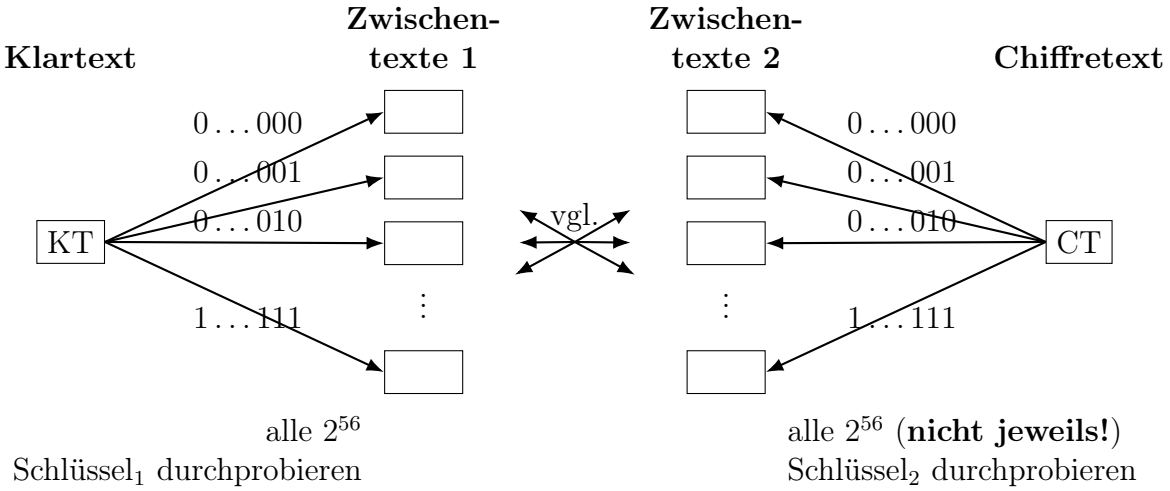

Abbildung 20: Grundgedanke eines Meet-in-the-Middle-Angriffes: Hat Eve ein Klartext-Chiffretext-Paar, kann sie mit allen 2^{56} Schlüsseln den Klartext verschlüsseln und mit allen 2^{56} Schlüsseln den Chiffretext entschlüsseln. Dann vergleicht sie alle Ergebnisse. Findet sie eine Übereinstimmung, kennt sie die beiden Schlüssel, mit denen Alice verschlüsselt hat.

Ist der Angreifer im Besitz eines Paares aus Klar- und Chiffretext (Spezialfall eines

. . . . . .

so kann er die Verschlüsselung von beiden Seiten angreifen, siehe Abbildung 20. Der Klartext wird mit sämtlichen möglichen Schlüsseln für Stufe 1 verschlüsselt (2^{56} Möglichkeiten), der Chiffretext mit sämtlichen möglichen Schlüsseln für Stufe 2 (ebenfalls 2^{56}) entschlüsselt. Die Ergebnisse werden miteinander verglichen. So müssen insgesamt nur

. . . . . .

Ver- und Entschlüsselungen – und zusätzlich die Vergleiche – durchgeführt werden. Man hat also effektiv nicht 112 Bit sondern 57 Bit Schlüssellänge.

Was aber schon funktioniert, ist, DES dreifach anzuwenden. Üblicherweise geschieht das in der Form:

$$\mathrm{DES}_{K1}\Big(\mathrm{DES}^{-1}_{K2}\big(\mathrm{DES}_{K1}(m)\big)\Big)$$

Ein Meet-in-the-Middle-Angriff ist dann zwar wieder möglich, allerdings muss Eve dann

$$2^{112} + 2^{56} \approx 2^{112}$$

Ver- und Entschlüsselungen durchführen.

Weil es noch weitere Angriffsmöglichkeiten gibt, wird 3DES mit zwei Schlüsseln vom NIST mit einem Sicherheitsniveau von 80 Bit bewertet.[29]

[29] https://de.wikipedia.org/wiki/Data_Encryption_Standard

Da jedoch DES in Software etwas langsam ist und wegen des Speicherbedarfs, Strombedarfs etc., erhöht man nicht einfach die Rundenzahl oder Durchläufe, sondern nimmt Alternativen wie Blowfish, AES etc.

7.3 Advanced Encryption Standard (AES)

Im Jahre 1997 kam es zu einer Ausschreibung für den DES-Nachfolger, welche 2000 vom Algorithmus *Rijndael*[30] gewonnen wurde. In Bezug auf theoretische Schwachstellen war er den anderen Finalisten (u. a. *Serpent* und *Twofish*) kaum überlegen, nebenbei der „am wenigsten konservative“ Algorithmus. In Hinblick auf den Ressourcenverbrauch und die Leistung in Hard- und Software wurde er sehr gut bewertet.

In den USA ist seine Verwendung (im Gegensatz zu DES beispielsweise) bis zur höchsten Geheimhaltungsstufe (*top secret*) zugelassen.

Der AES ist keine Feistel-Chiffre sondern ein SP-Netzwerk (Substitution-Permutation), deshalb ist Entschlüsselung nicht dasselbe wie Verschlüsselung und es musste von den Erfindern darauf geachtet werden, dass auch die Entschlüsselung effizient durchführbar ist.

Die Schlüssellänge beim AES kann 128, 192 oder 256 Bit sein. Abhängig davon werden in 10, 12 oder 14 Runden 128-Bit-Blöcke verarbeitet, die jeweils als 4×4-Matrizen von Bytes geschrieben und in vier Schritten verarbeitet werden:

- **SubBytes** – Jedes Byte wird mittels S-Box substituiert. Die S-Box wurde gezielt designt, so dass sie gegenüber linearer und differentieller Kryptoanalyse[31] resistent ist.
- **ShiftRows** – Die vier Einträge je Zeile werden $0, 1, 2$ bzw. 3 Einträge nach links rotiert. So liegen Zahlen aus der selben Spalte danach in verschiedenen Spalten.
- **MixColumn** – Die vier Zahlen in einer Spalte werden vermischt (durch lineare Operationen).
- **AddRoundKey** – Der Rundenschlüssel wird mit dem Datenblock XOR-verknüpft.

Diese vier Operationen bilden eine Runde.
AES wird verwendet in:

- WPA2 (also WLAN) basiert darauf und ist sehr sicher, außer für Passwort-Angriffe, weil diese oft schwach gewählt sind.
- In KeePass (als Standard-Option, mit *Key Derivation Function* und 10^6 Durchgängen).
- LTE verwendet drei verschiedene Verschlüsselungen, eine davon basiert auf AES.

[30] gesprochen wie „Reindahl“
[31] Diese Analysemethoden wurden Anfang der 1990er veröffentlicht.

Auf AES sind inzwischen einige Angriffe bekannt, die aber in der Praxis ohne Bedeutung sind.

Blockchiffren sind prinzipiell denkbar als Ersatz für Hashfunktionen, weil sie die geforderten Eigenschaften bieten. Allerdings sind sie in vielen Fällen zu langsam.

Frühere Prüfungsaufgaben

1. Berechne folgenden Ausdruck effizient, d. h. mit geringstmöglichem Rechenaufwand. Es müssen so viele Zwischenschritte angegeben werden, dass ich erkennen kann, dass die Rechenregeln intelligent genutzt wurden.

$$\left(34^{40} - 142\right) \bmod 5$$

2. Berechne folgenden Ausdruck effizient, d. h. mit geringstmöglichem Rechenaufwand. Es müssen so viele Zwischenschritte angegeben werden, dass ich erkennen kann, dass die Rechenregeln intelligent genutzt wurden.

$$\left(172 \cdot 30^{30}\right) \bmod 9$$

3. Erkläre, was $\mathbb{Z}_{11}$ bezeichnet. Was enthält es für Objekte (aufzählen und erklären!)? Welche Eigenschaften hat es? Welche Eigenschaften hat sein Element $\overline{8}$? Finde das Inverse.

4. Erkläre, was $\mathbb{Z}_{10}$ bezeichnet. Was enthält es für Objekte (aufzählen und erklären!)? Welche Eigenschaften hat es? Welche Eigenschaften hat sein Element $\overline{8}$? Zeige, dass es Nullteiler ist.

5. Welche Möglichkeiten gibt es, das Inverse von $\overline{8} \in \mathbb{Z}_{11}$ zu finden?
   Welche Möglichkeiten gibt es, das Inverse von $\overline{8} \in \mathbb{Z}_{172736479374769832678 21}$ zu finden?

6. Angenommen, jemand entwirft eine eigene Verschlüsselung, indem er die Buchstaben des Klartextes (codiert als Zahlen 0 bis 25) als Elemente von $\mathbb{Z}_{26}$ auffasst und als Schlüssel $k \in \mathbb{Z}_{26}$ verwendet. Die Verschlüsselung geschieht dann so, dass die Klartextnachricht m mit dem Schlüssel k multipliziert wird.
   Nimm dazu Stellung! Funktioniert dieses Kryptosystem? Wie würde man entschlüsseln? Zeige deine Behauptungen durch numerische Beispiele!

7. Angenommen, jemand entwirft eine eigene Verschlüsselung, indem er die Buchstaben des Klartextes (codiert als Zahlen 0 bis 25) als Elemente von $\mathbb{Z}_{27}$ auffasst und als Schlüssel $k \in \mathbb{Z}_{27}$ verwendet. Die Verschlüsselung geschieht dann so, dass die Klartextnachricht m mit dem Schlüssel k multipliziert wird.
   Nimm dazu Stellung! Funktioniert dieses Kryptosystem? Wie würde man entschlüsseln? Zeige deine Behauptungen durch numerische Beispiele!
   Wie sicher wäre dieses Kryptosystem generell? Wie sicher gegenüber einem Known-plaintext-Angriff, bei dem der Angreifer lediglich den ersten Buchstaben des Klartextes kennt?

8. Erkläre, wie man feststellen kann, ob

$$\overline{14} \in \mathbb{Z}_{77}$$

invertierbar ist.
Erkläre, wie man feststellen kann, ob

$$\overline{16871687616761687616874} \in \mathbb{Z}_{773187156761441564506876168 47}$$

invertierbar ist.

9. Erkläre den erweiterten euklidischen Algorithmus am Beispiel der Zahlen 14 und 78. Was folgt aus dem Ergebnis alles?

10. Berechne unter Verwendung des erweiterten euklidischen Algorithmus das Inverse von $\overline{83} \in \mathbb{Z}_{120}$.

11. Berechne die Potenzen von $\overline{3} \in \mathbb{Z}_{13}$.
Wie viele Elemente werden durch Potenzieren von $\overline{3}$ überhaupt „erreicht“?
Welche Antworten wären auf die vorige Frage überhaupt möglich gewesen?
Gib einen diskreten Logarithmus an.
Erkläre, ob logarithmieren immer eine Lösung hat.

12. Berechne $\log_{\overline{10}}(\overline{12})$ in $\mathbb{Z}_{13}$. Erkläre den Aufwand. Wie sähe alles in

$$\mathbb{Z}_{17273647937476983267821}$$

aus? Wieso ist der Logarithmus von $17\,273\,647\,937\,476\,983\,267\,821$ in $\mathbb{R}$ für einen Taschenrechner kein Problem?

13. Berechne $\log_{\overline{4}}(\overline{1})$ in $\mathbb{Z}_{17}$.
Berechne $\log_{\overline{4}}(\overline{8})$ in $\mathbb{Z}_{17}$.

14. Führe den Square-and-Multiply-Algorithmus zur Berechnung von $\overline{6}^{520}$ in $\mathbb{Z}_{11}$ vor. Wie viele Berechnungen müssen durchgeführt werden?

15. Erkläre **unter Nennung von Beispielen**, ob es sinnvoll ist, eine Chiffre mehrfach anzuwenden. Erkläre, ob es sinnvoll ist, DES zweifach anzuwenden.

16. Gib die Hauptziele und Nebenziele von Blockchiffren an. Erläutere diese Begriffe.

17. Wie werden die jeweiligen Ziele erreicht (mit jeweils kurzer Begründung)?

18. Wie geht der Schlüssel in Blockchiffren ein (mit Begründung!)?

19. Welche Angriffe sind gegen Blockchiffren möglich? Wie aussichtsreich sind sie?

20. Erkläre die Bedeutung der Schlüssellänge.

21. Erkläre die Bedeutung der Blockgröße.

22. Erkläre den *Meet-in-the-Middle-Angriff*. Wo kann er eingesetzt werden und wie funktioniert er?