Kryptologie

Kryptologie mit Java

Claudia Bergstein

Was ist das? Was macht man da? Kann man das essen?

Diese und noch andere Fragen haben wir (wir – das sind 10 wissbegierige Schüler aus ganz Baden-Württemberg) uns auch gestellt, als wir unser Kursthema das erste Mal gelesen haben.

Um der Sache auf den Grund zu gehen, trafen wir uns erstmals zu einem sogenannten Vorbereitungswochenende in Adelsheim. Dort zusammengetroffen erhielten wir auch sofort eine kleine Einführung in unsere Kursarbeit im Sommer.

Sichtlich begeistert und unter währender Dramatik schilderte uns unser Kursleiter Matthias Taulien die tollsten Geschichten aus der Welt der Verschlüsselungsalgorithmen (Kryptologie), die von den alten Ägyptern bis hin zu unserer Zeit reichten.

Von seiner Begeisterung angesteckt, beschafften wir uns alle das Buch „Geheime Botschaften“ von Simon Singh, um in die Welt der Ver- und Entschlüsselung einzutauchen.

Durch das Buch inspiriert und hochmotiviert konnten wir es kaum erwarten, unser erarbeitetes Wissen im Sommer anzuwenden. Als der Tag des Wiedersehens endlich gekommen war, lernten wir unseren zweiten Kursleiter Jörg Richter kennen, da er am Vorbereitungswochenende leider verhindert war. Noch dazu kamen vier weitere Kursteilnehmer aus dem fernen China, um sich unserer wagemutigen Truppe anzuschließen.

Doch wie sollten wir uns jetzt verständigen? Mit den Händen oder mit den Füßen? Wir brauchten also eine Sprache, die wir alle verstanden...

Aus eben diesem Grund gestaltete sich unser Kurs von da an auf bilingualer Ebene, mal deutsch-englisch, mal englisch-deutsch, was jedoch kein Problem in der Zusammenarbeit darstellte. Die chinesischen Schüler sprachen alle sehr gut Englisch und auch wir deutschen Schüler gaben unser Bestes.

So konnten wir alle recht gut in unsere Diskussionen mit einbeziehen und schließlich standen uns unsere Kursleiter und unsere Schülermentorin Tabea Tscherpel mit Rat und Tat zur Seite, falls dennoch Probleme auftreten sollten.

Unsere vier chinesischen Freunde (Ning, Nancy, Leo und Justin) waren im Laufe der Zeit ein fester Bestandteil unserer Gruppe geworden und so gestaltete sich aus unserem bunt zusammengewürfelten Haufen bald ein starkes Team, das durch keine noch so schwere Verschlüsselung zu erschrecken war. Wir arbeiteten uns durch die verschiedensten Kapitel der Kryptologie und verzweifelten auch nicht, als wir uns an schweren mathematischen Beweisen zu schaffen machten. Unser Kursleiter trug unseren „Leitspruch“ auf der Brust: „Mathe macht glücklich!“

Dass das alles jedoch sehr anspruchsvoll und kompliziert war, zeigte sich auch an einigen Begriffen wie ugly, teriffic und tricky, die von nun an des Öfteren in den Erklärungen unserer Kursleiter zu hören sein sollten.

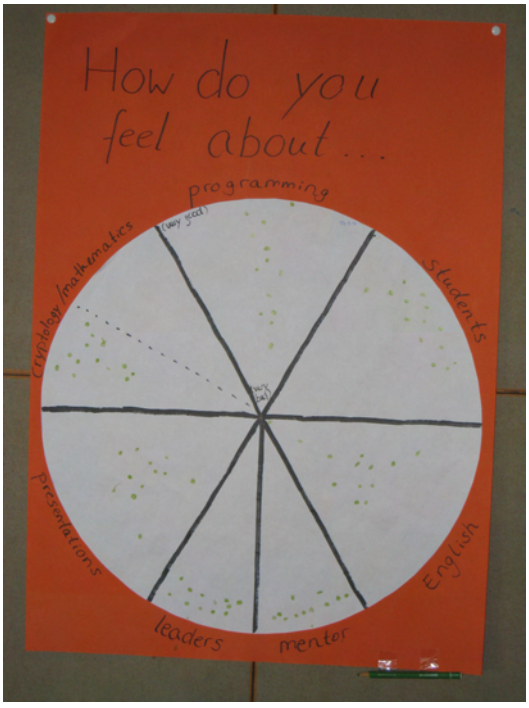
Wenn jedoch Schwierigkeiten oder Probleme auftraten, konnten diese meist schnell und unkompliziert geklärt werden.

Da wir ja nicht nur neues Wissen erlernen, sondern auch als Team zusammenarbeiten sollten, beispielsweise zur Präsentationsvorbereitung, hing in unserem Kursraum ein kleines Pinnboard, an welches jeder seine Empfindungen, Gefühle, Kritik und Wünsche schreiben konnte.

Somit konnten Probleme beseitigt und Missverständnisse rasch aufgeklärt werden.

Um an einen aktuellen Informationsstand über jeweilige Empfindungen gegenüber einzelnen Rubriken wie die Kursarbeit, die Kursleiter, die Schülermentorin und das Englisch im Kurs zu kommen, hing neben dem Pinnboard ein Kreis, in den jeder einmal pro Woche (also insgesamt zweimal) seine Punkte eintragen konnte.

So erhielten wir und unsere Leiter stets ein Feedback zu den einzelnen Themen und einen Gesamteindruck über unser Befinden.



Am Anfang waren die Punkte in einigen Bereichen eher zentriert, was grundsätzlich ein eher negatives Zeichen ist, in unserem Fall jedoch durchaus zu erwarten war, da wir in gewissen Fachbereichen schlicht und einfach noch keine Ahnung hatten und uns daher auch noch nicht sofort dabei wohlgefühlt haben.

Doch bereits Anfang der zweiten Woche lichtete sich das Feld und die Tendenz der Punkte richtete sich deutlich zum äußeren Rand des Kreises aus.

Wir waren also auf einem guten Weg, was sich auch in der Zusammenarbeit zur Vorbereitung der Abschlusspräsentation bemerkbar machte. Hatten wir bei der Rotationsvorbereitung noch mit einigen Schwierigkeiten zu kämpfen, so lief nun nahezu alles wie am Schnürchen. Gruppen wurden eingeteilt, jeder wusste was er zu tun hatte und so kamen wir schnell und ohne unter Zeitdruck zu geraten ans Ziel unserer Bemühungen.

Gegen Ende der Akademie gab es eigentlich nur noch am äußeren Rand der Scheibe Punkte zu sehen. Im Rückblick lässt sich durchaus sagen: in diesen beiden Wochen intensiver Zusammenarbeit haben wir sehr viel gelernt. Sowohl aus fachlicher als auch aus zwischenmenschlicher Sicht. Wir haben gelernt teamfähig und vor allem selbständig zu arbeiten. Und das sind wir:

Über zwei Wochen hindurch arbeiteten, lachten, programmierten, verschlüsselten und diskutierten wir, um am Ende sagen zu können: Es hat echt Spaß gemacht, und es hat sich gelohnt, zwei Wochen der Sommerferien dafür zu opfern, denn es herrschte stets eine ganz andere Arbeitsatmosphäre, als man es in der Schule gewohnt ist. Das Lernen und Arbeiten fällt viel leichter und geht auch schneller voran, wenn man unter Gleichgesinnten ist.

Noch dazu war es insbesondere für die deutschen Schüler unter uns sehr interessant, mit einer uns doch so fremden Kultur wie China Bekanntschaft zu machen. So haben sie uns beispielsweise erzählt, dass eine Schulklasse in China aus bis zu 60 Schülern/Schülerinnen besteht, oder dass sie keinen bzw. kaum Sport machen, da sie a) keine Zeit dazu hätten und b) es nicht die Möglichkeit gäbe, in eine Sportgruppe zu gehen.

An dieser Stelle ein herzliches Dankeschön an alle, die auf irgendeine Art und Weise etwas zu diesen zwei erfüllten Wochen in Adelsheim beigetragen haben.

Familie Krypto

Katharina Barbu, Claudia Bergstein,
Saskia Bluhm

Andreas stellte intelligente Fragen zum Kursinhalt und war auch sonst sehr lebhaft am Kursgeschehen beteiligt. Am Sportfest konnte er auf Grund seiner schmerzhaften Fußverletzung nicht teilnehmen, doch übernahm er tapfer die Rolle des Teamcoachs und feuerte uns tatkräftig an.

Claudia sprudelte nur so vor Energie, die sie an das ganze Team weitergab. Ob das vielleicht an ihrer Wasserflasche lag, die sie immer bei sich trug? Vor allem aber in den Diskussionsrunden und beim Denken lebte sie richtig auf.

Justin war ein eher stilles und für sich arbeitendes Kursmitglied, was ihn jedoch nicht von Diskussionen ausschloss. Auch sorgte er für gute Stimmung in seinem Präsentationsteam.

Jörg war einer unserer Kursleiter. Er arbeitete immer sehr konzentriert und konnte uns neue Algorithmen und mathematische Beweise klar und deutlich näher bringen. Falls uns dennoch etwas unklar geblieben wäre, konnten wir auf seine Hilfe zurückgreifen.

Arne war der motivierteste und begeisterte Computerfachmann unter uns. Wenn es um computertechnische Arbeiten ging, war er sofort in seinem Element und arbeitete dem entsprechend konzentriert und begeistert an seinem Laptop.

Katharina gehörte zu den fleißigsten und verlässlichsten Kursteilnehmern, was sie zu einem unverzichtbaren Teil unserer Gruppe machte. Noch dazu überlegte sie sich raffinierte Varianten, einen bestimmten Algorithmus zu programmieren.

Leo war fasziniert von den verschiedensten europäischen Automodellen, die er auch gerne einmal googlete. Er war zwar eher zurückhaltend, doch bei Präsentationen und Vorträgen zeigte er sich souverän.

Matthias war ebenfalls einer unserer Kursleiter. Auch er konnte uns neue Themenbereiche gut erklären und unser Interesse für die Kryptologie wecken beziehungsweise wach halten. Bei Fragen und Missverständnissen stand er uns selbstverständlich hilfsbereit zur Verfügung.

Hubert war unser Sprachgenie Nummer Eins: Anfangs hatten wir nicht die geringste Ahnung davon, dass er fließend Chinesisch spricht. Dies war bei Kommunikationsproblemen zwischen deutschen und chinesischen Schülern sehr hilfreich.

Lea wirkte auf den ersten Blick zurückhaltend, doch im Laufe der zwei Wochen entwickelte sie sich zu einem aufgeschlossenen und geistig produktiven Gruppenmitglied. Bei Präsentationen und Textabarbeiten brachte sie immer gute Verbesserungsvorschläge.

Nancy war nur gut gelaunt und hatte immer ein Lächeln auf den Lippen, weshalb wir sie bald alle in unser Herz geschlossen hatten.

Tabea war für uns die hilfsbereiteste, netteste und beste Schülermentorin, die man sich vorstellen kann. Bei aufkommenden Fragen stand sie uns mit Rat und Tat zur Seite. Sie war immer mit Block und Stift am Start und dokumentierte unsere Arbeit.

Cornelius war der kleine Kritiker unter uns. Er

beäugte Kursarbeiten sehr kritisch, was sich jedoch positiv auf das Gesamtwerk auswirkte. Er fand immer etwas zu lachen, was die Kursatmosphäre erheblich lockerte.

Nadine war überaus engagiert bei Gruppenarbeiten dabei und übernahm bei nahezu allen organisatorischen Dingen die führende Leitung. Eine endgültige beziehungsweise aktuelle Version unserer Präsentationen war stets auf ihrem Rechner zu finden.

Ning war die Schnellste im Lösen von Problemen und fixierte sich immer darauf, einen guten Beitrag zu leisten. Sie liebte das Gefühl, etwas selbst herausgefunden zu haben.

Patrick war unser „Programmiermeister“: Sofort entwickelte er den passenden Algorithmus zu jeder Aufgabe. Selbst wenn die Kursleiter mit ihrem Latein am Ende waren (was natürlich nie vorkam – Anm. der Kursleiter), konnte Patrick uns immer wieder aus der Patsche helfen.

Saskia integrierte sich so aufgeschlossen und aktiv in die Kursarbeit, dass jeder gerne auf ihre Hilfe zurückgriff. Mit ihrem Karlsruher Temperament sorgte sie für so manchen Lacher im Kurs.

Alice, *Bob* und *Eve* sind unsere guten beziehungsweise schlechten „Freunde“, die uns als Beispielmodelle dienten.

Experience in cryptology class

Nancy Chen Xueying, Ning Huang Yining, Leo Lin Zihang, Justin Zhu Yizhao

Our class is cryptology, which contains encryption and decryption. Matthias and Jörg are our teachers, and Tabea is our teaching assistant. Our class is made up of us and another 10 German students.

In cryptography, computers are widely used, and also in our class we made some programmes ourselves. At first it was a little bit difficult for us to make computer programmes. Luckily, we received a lot of help from our teachers and Teaching Assistant Tabea. They told us how to make programmes to use Caesar, Vigenere and Euler cipher. It is really a fresh experience. We like discussing with the whole class. For example, the teacher gave us a Vigenere ciphertext and asked us to decrypt it without the help of the computer. We worked together; everyone dealt with one part of it. Finally, it was so amusing that we did it just by our hands and our brains. It impressed us deeply, not only because of the difficulty of solving the task, but also the experience of teamwork.

We learnt so much that at the middle presentation we could not help giving our speech for a long time.

The course here is quite different from ours in China. You can see some students sitting closely to the window while the teacher was talking. It is more free and at ease. We really like the atmosphere when we are sitting at a round table and discussing the given topic, although sometimes we could not understand what others said.

While staying in this course, it is easy to find it is like a big family rather than a class. If you have any questions, you just should raise your hand up. There's no need to be shy. No one will laugh at you even if your question is simple. Teachers here are enthusiastic to answer your questions.

Here are some impressions from us:

Ning: The teachers always make sure that all of us understand the question and what we have learnt. Besides, Matthias is a very interesting teacher, Jörg and Tabea as well. When I'm discussing something with them, I really feel the atmosphere is quite harmonious. It makes me eager to ask questions. I think also because of this, German students are fond of

asking and answering questions. Everyone takes part in the course. Everyone enjoys themselves.

Justin: These days, we have learnt a lot of things from the course. We understand how the RSA cipher (the newest and the most useful cipher) works. In my opinion, it is a good chance for us to learn something here. In the class, everybody can say his or her opinion. The students here like communicating with each other to exchange their ideas. It is much better to work in the team than by oneself. Here, teachers tell us what we can do rather than what we must do. This is why all the students have fun in the course. It is different from the course in China.

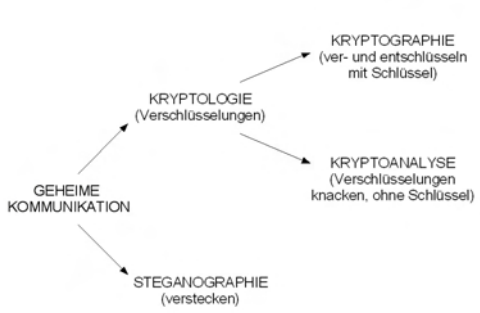
Leo: On Monday morning, Mr. Esslinger, a cryptography professor who works with the Deutsche Bank, came to our class and spoke to us about the use of cryptography in daily communication. We asked him a lot of questions, such as the future of the cryptography, how to protect the customer's private data from being stolen. I think it was a wonderful experience to have the chance to talk to the professor. I thought I was fortune enough to choose this course.

Nancy: I think I am very lucky to have had a chance to join the cryptography course. I like cryptography, because cryptography is a course that includes mathematics, IT and language. Although learning cryptography is a hard work, I like the challenge. In this class, I learnt many things that I had never heard before. It is amazing. With the great help of the teachers and the students, I can get hang of the knowledge well. In this class, I not only have learnt the knowledge, but also have learnt how to study. I am very happy to stay here during these two weeks. Best wishes to all the teachers and the classmates.

Einführung

Patrick Caspari

Möchte man eine geheime Nachricht von A nach B übertragen, ohne dass jemand anders herausfindet, was in der Nachricht steht, kann man das auf zwei verschiedene Arten machen. Entweder richtet man es so ein, dass niemand merkt, dass überhaupt eine Nachricht übertragen wurde – das nennt man dann Steganographie –, oder man schreibt die Nachricht in einer Weise, die nur der Empfänger, für den die Nachricht bestimmt ist, lesen kann, verschlüsselt sie also. Dieses Verfahren nennt man Kryptographie. Wissenschaftler, die sich damit beschäftigen, wie man eine Verschlüsselung knackt, nennt man Kryptoanalytiker; was sie tun, also den Klartext aus einem verschlüsselten Text herauszufinden, heißt einen Code angreifen. Der Oberbegriff von Kryptographie und Kryptoanalyse ist Kryptologie. Simon Singh verdeutlichte das in seinem Buch „Geheime Botschaften“ mit folgender Graphik:



In der Kryptologie spricht man der Anschaulichkeit halber gerne von Alice und Bob, zwei fiktiven Figuren, die sich verschlüsselte Nachrichten schicken wollen, wobei immer Alice eine Nachricht an Bob schicken möchte. Den „man in the middle“, das ist die Person, die versucht die Nachricht abzufangen und zu entschlüsseln, nennt man Eve.

Die Verschlüsselungstechniken kann man in die folgenden zwei Kategorien einteilen:
Es gibt Substitutionsverfahren, das sind Verschlüsselungen, bei denen die Klartextbuchstaben durch andere Buchstaben oder Zeichen ersetzt werden, und Transpositionsverfahren, bei denen die Buchstaben des Klartexts nach einem bestimmten System vertauscht werden. Bei den Substitutionen unterscheidet man monoalphabetische und polyalphabetische Substitution. Bei monoalphabetischen Substitutionen wird jeder Klartextbuchstabe durch einen einzigen Buchstaben beziehungsweise ein Symbol ersetzt. Bei polyalphabetischen Substitutionen dagegen gibt es mehrere Geheimtextal-

phabete, das heißt, jeder Klartextbuchstabe kann durch verschiedene Geheimtextbuchstaben ersetzt werden und jeder Geheimtextbuchstabe kann für verschiedene Klartextbuchstaben stehen. Eine polyalphabetische Substitution, die wir näher behandelt haben, ist beispielsweise die Vigenère-Verschlüsselung.

Steganographie

Arne Hansen-Dörr

Steganographie und Kryptographie beabsichtigen beide das Gleiche: geheime Nachrichten sollen versendet werden, ohne dass irgendeiner mitbekommt, um was es sich dabei handelt. Bei der Steganographie geschieht das dadurch, dass man die Nachricht versteckt. Wenn jemand die Nachricht also finden würde, könnte er sie ohne weiteres lesen. Die Sicherheit der Nachricht beruht also nur auf einem guten Versteck derselben.

Zu Cäsars Zeiten rasierte man Sklaven eine Glatze und schrieb ihnen die Nachricht auf den Hinterkopf. Dann wartete man, bis ihnen die Haare wieder gewachsen waren und somit die Nachricht unerkennbar geworden war. Nun sendete man die Sklaven zu dem Empfänger der Nachricht, der dem Sklaven die Haare wiederum rasierte und so die Nachricht lesen konnte.

Eine andere Möglichkeit, die schneller und vermutlich auch sicherer als das „Glatzeschneiden“ von statten geht, ist die folgende Methode: Man sucht sich ein beliebiges Buch und eine dünne Nadel. Man schlägt das Buch beispielsweise auf der zehnten Seite auf und sticht mit der Nadel bestimmte Buchstaben. Jetzt wird das Buch einem Gefangenen ins Gefängnis geschickt und passiert die Sicherheitskontrolle ohne Schwierigkeiten, da die Nadelstiche auf den ersten Blick nicht zu erkennen sind. Der Gefangene schlägt nun das Buch auf der zehnten Seite auf, hält jene Seite gegen das Licht und erkennt die unterlochten Buchstaben, die die Botschaft: „Pech gehabt“ ergeben. – Da scheint es wohl keine Hoffnung mehr zu geben.

In der heutigen Zeit findet die Steganographie trotz des mit heutigen Mitteln unangreifbaren RSA-Verfahrens immer noch Anwendungen, die natürlich anders ausgeführt werden. Heute versteckt man Nachrichten zum Beispiel in Bildern. Wir haben dies im Kurs mit einem kleinen Computerprogramm ausprobiert:

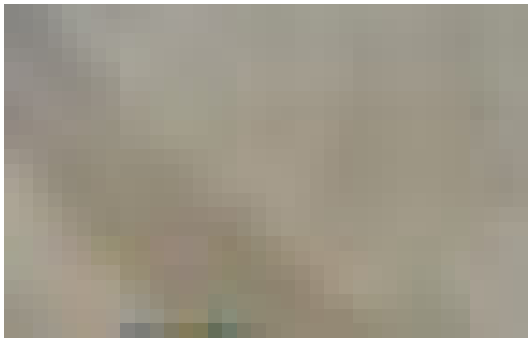
Originalbild

Zuerst hat man die Möglichkeit, eine Texteingabe zu machen, etwa: „Hallo wie geht’s“ als Klartext. Nun kann man noch eingeben bei welcher Stelle die Änderung der Bildpunkte anfangen soll. Zuletzt muss man noch ein zu bearbeitendes Bild auswählen. Wenn man das Programm jetzt startet, sucht es für jeden der Buchstaben beziehungsweise jedes Zeichen die dazugehörige Farbe in einer Tabelle, welche im Programm integriert ist. Dann wird die gefundene Farbe in der Tabelle in Form eines geänderten Bildpunktes im Bild integriert, indem die Farbe eines Bildpunktes auf dem Bild durch eine neue ersetzt wird.

Bild mit geänderten Bildpunkten, die die Nachricht enthalten

Bei der Ersetzung fängt das Programm bei den Koordinaten an, die man beim Programmstart angegeben hatte. Die geänderten Bildpunkte enthalten die Nachricht. Das neue Bild wird in dem gleichen Ordner wie das Ausgangsbild

gespeichert. Nun lädt man beispielsweise das Originalbild auf eine Webseite und ersetzt es ein paar Tage später durch das modifizierte Bild. Der Empfänger lädt die Bilder herunter und vergleicht sie dann mit dem gleichen Programm, das natürlich die geänderten Bildpunkt erkennt und die Nachricht in Form des Klartextes ausgibt. Diese Art wird zum Beispiel von Leuten genutzt, die nicht offen zeigen wollen, dass sie miteinander kommunizieren, wie zum Beispiel Terroristen. Keiner würde auf die Idee kommen, dass sich auf einer Webseite etwas geändert hat, wenn man nach zwei Tagen ein Bild mit einem scheinbar gleichen vertauscht und wegen dieses Unterschiedes eine Nachricht übermittelt werden kann.



Ausschnitt des unteren Bildrandes mit sichtbaren, geänderten Bildpunkten.

Wenn man das bearbeitete Bild jetzt vergrößert, kann man die geänderten Bildpunkte am unteren Bildrand erkennen, die die Nachricht „Hallo wie geht's“ enthalten.

In der Praxis macht man das natürlich geschickter, so dass man den Unterschied der Bilder mit bloßem Auge nicht bemerkt.

Cäsar-Verschlüsselung

Hubert Cao, Andreas Mayer

Diese Verschlüsselungsmethode geht auf Gaius Julius Cäsar zurück. Er benutzte sie bereits zu seiner Zeit, um wichtige militärische Botschaften geheim zu übermitteln. Sie ist wahrscheinlich eine der ältesten Verschlüsselungen der Welt. Sie ist eine monoalphabetische Verschlüsselung, die sehr einfach aber auch unsicher ist.

Man schreibt dabei das Geheimtextalphabet und das Klartextalphabet untereinander und verschiebt dann das Geheimtextalphabet um eine bestimmte Anzahl von Buchstaben nach links. Die Zahl der Stellen, um die das Geheimtextalphabet verschoben wird, ist der Schlüssel. Der Klartext wird dann immer mit dem darunter liegenden Buchstaben des Geheimtextalphabetes verschlüsselt.

Man kann die Cäsar-Verschlüsselung auch mit einer Alberti-Scheibe durchführen. Diese wurde im 15. Jahrhundert nach ihrem Erfinder Leon Battista Alberti (1404-1472) benannt. Sie besteht aus zwei unterschiedlich großen, unabhängig zueinander drehbaren Scheiben. Auf der inneren Scheibe steht das Geheimtextalphabet, auf der äußeren befindet sich das Klartextalphabet. Man kann sie beliebig in eine gewünschte Position verschieben.

Alberti-Scheibe

Ein Beispiel mit einer Verschiebung um drei Stellen, der Schlüssel wäre also die Zahl 3.

Klartextalphabet
`abcdefghijklmnopqrstuvwxyz`
`DEFGHIJKLMNOPQRSTUVWXYZABC`
Geheimtextalphabet

Zum Verschlüsseln wird ein Klartextbuchstabe durch den unter ihm stehenden Geheimtextbuchstaben ersetzt.

Klartext	`der krypto kurs ist cool`
Geheimtext	`GHU NUBSWR NXUV LVW FRRO`

Beim Entschlüsseln wird der Geheimtext immer mit dem darüber liegenden Buchstaben des Klartextalphabets entschlüsselt.

Programmieren

Patrick Caspari

Sind einem nun aber die darüber- beziehungsweise darunterliegenden Buchstaben unsympathisch, gibt es noch einen anderen Weg. Unser Kursname hatte auch noch einen 2. Teil, doch was ist nun dieses Java? Hat das etwa etwas mit den kleinen, braungewandeten Wesen aus Star Wars, die auf Tatooine Droiden aufsammeln und verkaufen, zu tun?

Nein, hat es nicht. Oder wenn, dann nur sehr entfernt. Java ist eine Computersprache, die es uns ermöglicht, die Algorithmen, die für die verschiedenen Verschlüsselungen notwendig sind, schlicht vom Computer durchführen zu lassen, statt selbst herumknobeln zu müssen.

Der große Vorteil hierbei ist, dass man sich nur einmal die Arbeit machen muss, das entsprechende Programm zu schreiben, und danach immer, wenn man die Verschlüsselung benutzt, auf ebendieses Programm zurückgreifen kann.

Ein Beispiel: Wir wollen einen Text mit Cäsar verschlüsseln; der Schlüssel ist 5. Das heißt: Jeder Buchstabe wird mit dem Buchstaben verschlüsselt, der 5 Stellen weiter im Alphabet steht. Wir beginnen also mit dem ersten Buchstabe, sei es ein „e“, und gehen von diesem „e“ aus 5 Stellen weiter im Alphabet, also zum „j“.

Jetzt haben wir einen Buchstaben mit Cäsar verschlüsselt, und je nachdem wie gut wir uns im Alphabet zurechtfinden, haben wir dafür 5 Sekunden bis eine Viertelstunde gebraucht, bei einer komplizierteren Verschlüsselung ein Vielfaches davon. Verschlüsselt man jetzt beispielsweise einen Brief mit 200 Wörtern, zu durchschnittlich 5-6 Buchstaben, kommt man damit auf ca. eineinhalb Stunden bis 2 Tage. Und will nun J. K. Rowling das Manuskript ihres geheimen, 2000 Seiten langen 8. Harry Potter-Buchs verschlüsseln und ihrem Verlag zuschicken, dann Gnade Gott dem zuständigen Kryptographen.

Mit Java jedoch setzt man sich einmal konzentriert an den Computer, programmiert das entsprechende Programm und setzt sich danach gemütlich zurück und trinkt eine Tasse Kaffee, während der Computer allein mit Buchstaben und Zahlen jonglieren darf. Zumindest bei dem Manuskript, bei den anderen Texten reicht die Zeit nicht für eine ganze Tasse.

Programmieren wir also das Programm für die Cäsar-Verschlüsselung:

```
public static String encrypt(
        String klartext, int key){
 String geheimtext = "";
 char cipherc;

 for(int i=0; i<klartext.length(); i++){
  cipherc=(char)(klartext.charAt(i)+key);
  if(cipherc > 'Z'){
   cipherc -= 26;
  }
  geheimtext += cipherc;
 }
 return geheimtext;
}

public static String decrypt(
        String geheimtext, int key){
 return encrypt(geheimtext, 26 - key);
}
```

Die beiden kursiv hervorgehobenen Zeilen stellen die eigentliche Verschlüsselungsarbeit dar. Die erste markierte Zeile beschreibt, wie der Computer den Klartext verschlüsseln soll: Er wandelt (implizit) jeden Buchstaben einzeln in eine Zahl um, zählt den Schlüssel dazu und wandelt die Zahl wieder in einen Buchstaben um; die folgenden 2 Zeilen stellen sicher, dass unsere Zahl nicht größer als die Alphabetlänge

wird.

Die zweite kursive Zeile ist für die Entschlüsselung eines Geheimtextes zuständig; praktischerweise kann man dazu eifach die Methode *encrypt* verwenden, wenn man als Schlüssel 26 minus den Schlüssel, mit dem die Nachricht verschlüsselt wurde, benutzen.

Nichts leichter als das.

Substitution

Hubert Cao, Andreas Mayer

Eine weiterentwickelte Form der monoalphabetischen Verschlüsselung ist die Substitution. Das Geheimtextalphabet ist nicht wie bei der Cäsar-Verschlüsselung zum Klartextalphabet verschoben, sondern durch beliebige Symbole oder andere Buchstaben ersetzt. Gleiche Buchstaben im Klartext werden bei diesem Verfahren mit gleichen Symbolen beziehungsweise Geheimtextbuchstaben verschlüsselt. Man sollte dabei darauf achten, dass die Reihenfolge der Buchstaben oder Symbole zufällig ausgewählt werden.

Hier ein Beispiel:

Klartextalphabet
`a b c d e f g h i j k l m n o p q r s t u v w x y z`
❁❂❃❄❅❆❇❈❉❊❋●❍■❏❐❑❒▲▼◆❖◗ ✲✡✣
Geheimtextalphabet

Noch ein Beispiel mit Buchstaben:

Klartextalphabet
`abcdefghijklmnopqrstuvwxyz`
`LTAYSMPVHQGWCNEBUJRKIZXDFO`
Geheimtextalphabet

Zum Verschlüsseln wird – genau wie schon bei der Cäsar-Verschlüsselung – ein Klartextbuchstabe durch den unter ihm stehenden Geheimtextbuchstaben ersetzt.

Klartext	`der krypto kurs ist cool`
Geheimtext	`YSJ GJFBKE GIJR HRK AEEW`

Beim Entschlüsseln wird der Geheimtext wieder mit dem darüber liegenden Buchstaben des Klartextalphabets entschlüsselt.

Vigenère-Verschlüsselung

Cornelius Kuhn

Geschichte

Die Cäsar-Verschlüsselung und auch die monoalphabetische Substitution sind vergleichsweise leicht durch eine Häufigkeitsanalyse zu knacken. Es liegt jetzt also bei den Kryptographen, ein neues, komplizierteres Verschlüsselungsverfahren zu entwickeln.

Die entscheidende Idee kam Leon Alberti, der übrigens auch die Alberti-Scheibe (siehe Cäsar) erfand, um das Jahr 1460: Man könnte doch statt eines Geheimtextalphabetes mehrere Geheimtextalphabete verwenden, zwischen denen man hin und her wechselt.

KT	`abcdefghijklmnopqrstuvwxyz`
G1	`FZBVKIXAYMEPLSDHJORGNQCUTW`
G2	`GOXBFWTHQILAPZJDESVYCRKUHN`

Hier ein Beispiel[1] mit zwei Geheimtextalphabeten G1 und G2, KT ist das Klartextalphabet. Die Häufigkeitsanalyse würde hier nicht funktionieren, weil z. B. „e“ manchmal als „K“ und manchmal als „F“ verschlüsselt wird.

Doch Alberti entwickelte diese Idee nicht weiter. Diese Aufgabe fiel anderen Gelehrten zu, zuletzt dem Diplomaten Blaise de Vigenère.

Funktion

Vigenère kam Mitte des 16. Jahrhunderts die Idee, alle Cäsar-Alphabete (mit der Verschiebung von 0 bis 25) untereinander zu schreiben (in das so genannte Vigenère-Quadrat) und

[1] Beispiel nach Simon Singh: „Geheime Botschaften“, Seite 66

zwischen ihnen abhängig von einem Schlüsselwort hin und her zu springen.

	A	B	C	D	E	F	G	H	I	J	K	L	M	N	O	P	Q	R	S	T	U	V	W	X	Y	Z
A	A	B	C	D	E	F	G	H	I	J	K	L	M	N	O	P	Q	R	S	T	U	V	W	X	Y	Z
B	B	C	D	E	F	G	H	I	J	K	L	M	N	O	P	Q	R	S	T	U	V	W	X	Y	Z	A
C	C	D	E	F	G	H	I	J	K	L	M	N	O	P	Q	R	S	T	U	V	W	X	Y	Z	A	B
D	D	E	F	G	H	I	J	K	L	M	N	O	P	Q	R	S	T	U	V	W	X	Y	Z	A	B	C
E	E	F	G	H	I	J	K	L	M	N	O	P	Q	R	S	T	U	V	W	X	Y	Z	A	B	C	D
F	F	G	H	I	J	K	L	M	N	O	P	Q	R	S	T	U	V	W	X	Y	Z	A	B	C	D	E
G	G	H	I	J	K	L	M	N	O	P	Q	R	S	T	U	V	W	X	Y	Z	A	B	C	D	E	F
H	H	I	J	K	L	M	N	O	P	Q	R	S	T	U	V	W	X	Y	Z	A	B	C	D	E	F	G
I	I	J	K	L	M	N	O	P	Q	R	S	T	U	V	W	X	Y	Z	A	B	C	D	E	F	G	H
J	J	K	L	M	N	O	P	Q	R	S	T	U	V	W	X	Y	Z	A	B	C	D	E	F	G	H	I
K	K	L	M	N	O	P	Q	R	S	T	U	V	W	X	Y	Z	A	B	C	D	E	F	G	H	I	J
L	L	M	N	O	P	Q	R	S	T	U	V	W	X	Y	Z	A	B	C	D	E	F	G	H	I	J	K
M	M	N	O	P	Q	R	S	T	U	V	W	X	Y	Z	A	B	C	D	E	F	G	H	I	J	K	L
N	N	O	P	Q	R	S	T	U	V	W	X	Y	Z	A	B	C	D	E	F	G	H	I	J	K	L	M
O	O	P	Q	R	S	T	U	V	W	X	Y	Z	A	B	C	D	E	F	G	H	I	J	K	L	M	N
P	P	Q	R	S	T	U	V	W	X	Y	Z	A	B	C	D	E	F	G	H	I	J	K	L	M	N	O
Q	Q	R	S	T	U	V	W	X	Y	Z	A	B	C	D	E	F	G	H	I	J	K	L	M	N	O	P
R	R	S	T	U	V	W	X	Y	Z	A	B	C	D	E	F	G	H	I	J	K	L	M	N	O	P	Q
S	S	T	U	V	W	X	Y	Z	A	B	C	D	E	F	G	H	I	J	K	L	M	N	O	P	Q	R
T	T	U	V	W	X	Y	Z	A	B	C	D	E	F	G	H	I	J	K	L	M	N	O	P	Q	R	S
U	U	V	W	X	Y	Z	A	B	C	D	E	F	G	H	I	J	K	L	M	N	O	P	Q	R	S	T
V	V	W	X	Y	Z	A	B	C	D	E	F	G	H	I	J	K	L	M	N	O	P	Q	R	S	T	U
W	W	X	Y	Z	A	B	C	D	E	F	G	H	I	J	K	L	M	N	O	P	Q	R	S	T	U	V
X	X	Y	Z	A	B	C	D	E	F	G	H	I	J	K	L	M	N	O	P	Q	R	S	T	U	V	W
Y	Y	Z	A	B	C	D	E	F	G	H	I	J	K	L	M	N	O	P	Q	R	S	T	U	V	W	X
Z	Z	A	B	C	D	E	F	G	H	I	J	K	L	M	N	O	P	Q	R	S	T	U	V	W	X	Y

Vigenère-Quadrat

Die oberste Reihe im Vigenère-Quadrat stellt den Klartext da. Die linke Spalte ist der Schlüssel.

Verschlüsseln

Beispiel:

Klartext	`Vigenèreistcool`
Schlüssel	`TEXTTEXTTEXTTEX`

(der Schlüssel `TEXT` wird so lange wiederholt, bis er so lang ist wie der Text.)

Wir suchen zuerst die Zeile mit dem Schlüsselbuchstaben „T“, dann die Spalte mit dem Klartextbuchstaben „v“. An der Stelle, an der sich die beiden schneiden, finden wir den Geheimtextbuchstaben „o“. Wir fahren fort mit dem Schlüsselbuchstaben „E“ und dem Klartextbuchstaben „i“ und erhalten den Geheimtextbuchstaben „m“ – und so weiter.

Wir haben schließlich den

Geheimtext: `OMDXGIOXBWQVHSI`

Entschlüsseln

Das Entschlüsseln funktioniert umgekehrt.

Beispiel:

Klartext	`NYFKWROWEOWAREDPX`
Schlüssel	`KEYKEYKEYKEYKEYKE`

Nun suchen wir zuerst die Schlüsselzeile, in diesem Fall die Zeile mit dem Schlüsselbuchstaben „K“. Dann gehen wir waagerecht weiter, bis wir zum Geheimtextbuchstaben (hier „N“) kommen. Der Buchstabe, der jetzt oben in dieser Spalte steht, ist der Klartextbuchstabe („d“). Weiter geht's mit dem Schlüsselbuchstaben „E“ bis zum Geheimtextbuchstaben „Y“, von da senkrecht nach oben zum Klartextbuchstaben „u“.

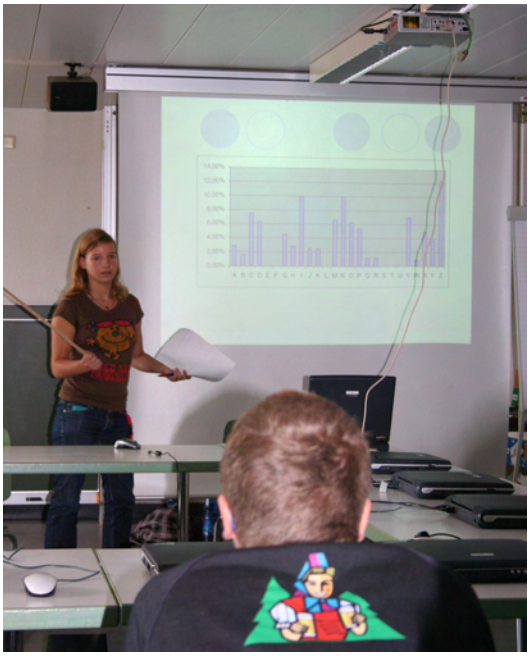
Letztendlich kommen wir zum

Klartext: `du hast es geschafft`.

Implementierung der Vigenère-Verschlüsselung

Patrick Caspari

Und weil die Vigenère-Verschlüsselung handschriftlich ebenfalls zu langwierig und langweilig ist, haben wir sie auch wieder programmiert. Als Grundlage dafür haben wir unser Programm für die Cäsar-Verschlüsselung genommen, nur dass bei Vigenère jeder Buchstabe mit einem anderen Schlüssel verschlüsselt werden muss.



In der ersten hervorgehobenen Zeile rufen wir wieder die Cäsar-Verschlüsselungsmethode auf. Zuvor wurde für den jeweiligen Vigenère-Buchstaben der entsprechende Cäsar-Schlüssel berechnet.

Das Entschlüsseln geschieht ganz analog zum Verschlüsseln, beachten Sie hierzu die zweite markierte Zeile.

```
public static String encrypt(String geheimtext, String schluessel) {
   String geheimext = "";
   int key;
   for (int i = 0; i < klartext.length(); i++) {
      key = schluessel.charAt(i % schluessel.length()) - 'A';
      geheimtext += Caesar.encrypt(""+ klartext.charAt(i), key);
   }
   return geheimtext;
}

public static String decrypt(String geheimtext, String schluessel) {
   String klartext = "";
   int key;
   for (int i = 0; i < geheimtext.length(); i++) {
      key = schluessel.charAt(i % schluessel.length()) - 'A';
      klartext += Caesar.decrypt(""+ geheimtext.charAt(i), key);
   }
   return klartext;
}
```

One-Time-Pad

Arne Hansen-Dörr

Das One-Time-Pad ist im Prinzip eine Weiterentwicklung des Vigenère-Codes. Der Unterschied besteht darin, dass beim One-Time-Pad der Schlüssel so lang wie der Klartext ist, und dass der Schlüssel aus zufällig hintereinandergereihten Buchstaben besteht. Die Ver- und Entschlüsselung findet so wie beim Vigenère-Code statt. Die Zufälligkeit und die Einmaligkeit des Schlüssels machen das One-Time-Pad unangreifbar. Würde man allerdings einen Schlüssel mehrere Male benutzten, ließe sich der Schlüssel herausfinden, indem man mehrere Texte, von denen man weiß, dass sie mit dem gleichen Schlüssel verschlüsselt sind, vergliche und dann die Häufigkeitsanalyse anwendete.

Im Gegensatz zur Substitution kann beim One-Time-Pad der gleiche Buchstabe im Klartext auf unterschiedlichen Schlüsselbuchstaben treffen und somit auch durch unterschiedlichen Geheimtextbuchstaben repräsentiert werden:

Substitution

Ersetzungstabelle:

a b c d e f g h i j k l m n o p q r s t u v w x y z
❁✪✻❄✼❆✳✺✸✷✹●❍■❑❒❐❑❒▲▼◆❖◗ ✲✡✤

Klartext	i c h b i n h i e r
Geheimtext	✻✼✻✪✻■✻✸✼❐

Man sieht: Der Buchstabe „i“ im Klartext wird dreimal durch das gleiche Geheimtextzeichen repräsentiert.

One-Time-Pad Verschlüsselung

Klartext	ichbinhier
Schlüssel	FSYBLOFBZM
Geheimtext	PUFCDBMJDD

Der Buchstabe „i“ im Klartext wird dreimal durch unterschiedliche Geheimtextbuchstaben repräsentiert. Ein Angriff per Häufigkeitsanalyse scheitert.

Die Zufälligkeit des Schlüssels schützt selbst vor einem Brute-Forcce-Angriff: Hat man mit einem One-Time-Pad verschlüsselten Text vor sich liegen und versucht mit Brute-Force, einen sinnvollen Klartext herauszubekommen, dann stößt man irgendwann zufällig auf Schlüssel, die einen Klartext ergeben, der sinnvoll ist:

Geheimtext	LGNFUHMXGHGAFDVI
Schlüssel	DFGEDHSVZDKANLSR
Klartext	ichbrauchewasser

Klartext: Ich brauche Wasser (dringender Hilferuf)

Geheimtext	LGNFUHMXGHGAFDVI
Schlüssel	EGCUGLETADZHNANR
Klartext	hallowiegehtsdir

Klartext: Hallo wie geht's dir (eine ganz harmlose Frage nach dem Befinden)

Man kann mit verschiedenen Schlüsseln also durchaus sinnvolle, aber verschiedene Klartexte mit ganz unterschiedlicher Bedeutung erhalten. Probiert man wirklich alle theoretisch möglichen Schlüssel aus, so wird man auch jeden Text als Klartext erhalten, der so viele Buchstaben hat wie der Geheimtext, und somit viele sinnvolle Texte, bei denen man nicht weiß, welcher der richtige ist.

Die Zufallsschlüssel bringen aber auch ein Problem mit sich: In Krisenzeiten müssen tausende Seiten Zufallsschlüssel erzeugt werden, die dann jeder auf Vorrat haben muss. Die Logistik, die nötig wäre, um Zufallsschlüssel zur richtigen Zeit am richtigen Ort zu haben, wäre so teuer und aufwendig, dass es sich nicht lohnt, zumal das Ganze ja geheim geschehen muss.

In der heutigen Zeit findet das One-Time-Pad aber noch an einer Stelle Anwendung, bei der es sehr wichtig ist, dass keiner die Nachricht lesen kann: beim Roten Telefon zwischen dem Kreml und dem Weißen Haus. Dort wird tatsächlich noch das One-Time-Pad angewendet, was natürlich wieder das sichere und geheime Verteilen von zufälligen Schlüsseln mit sich bringt. Es wird gelöst, indem man die Zufallsschlüssel in einen versiegelten Koffer packt, der persönlich überbracht wird. Außerdem werden nicht die Schlüssel für die nächsten zehn Jahre transportiert, da der eventuelle Verlust viel Geld kostet. Da wäre dann nur noch das Problem der Zufälligkeit der Schlüssel. Sie wird durch neuere Computer gewährleistet, und da das Rote Telefon nur zwei Enden hat, ist die Logistik nicht so aufwendig wie als hätte es tausend oder mehr.

Angriff auf verschlüsselte Texte

Hubert Cao, Cornelius Kuhn, Andreas Mayer

Cäsar-Verschlüsselung

Brute-Force

Fängt man eine mit dem Cäsar-Algorithmus chiffrierte Nachricht ab und kennt den Schlüssel nicht, kann man diese Art der Verschlüsslung sehr einfach „knacken“. Da es insgesamt nur 25 verschiedene Schlüssel gibt, kann man einfach einen nach dem anderen durchprobieren, bis man spätestens nach dem 25. Versuch den Schlüssel hat. Diese Art eines Angriffs – also alle möglichen Schlüssel auszuprobieren – nennt man Brute-Force.

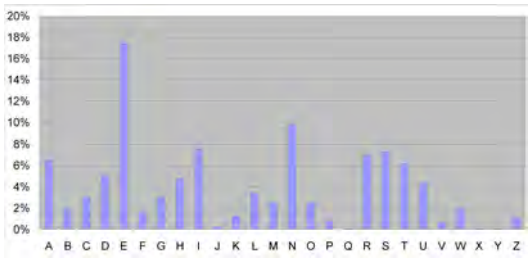
Häufigkeitsanalyse

Die Häufigkeitsanalyse beruht darauf, dass die Buchstaben des Alphabets bei einem typischen Text unterschiedlich oft vorkommen. Sie ist bei einfachen Substitutionsverfahren sehr hilfreich, wenn man die Verschlüsselung knacken will. Im Folgenden wird das es anhand eines mit Cäsar verschlüsselten Textes erläutert. Wir wollen mit der Häufigkeitsanalyse den Schlüssel des Textes herausfinden.

Zuerst ermittelt man die relativen Häufigkeiten der einzelnen Buchstaben im Geheimtext.

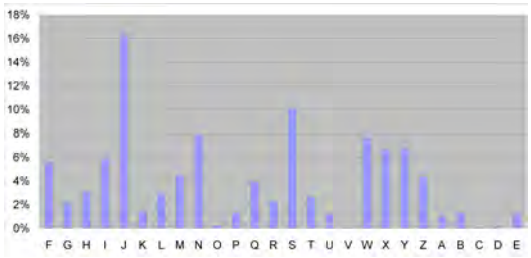
Weiß man, in welcher Sprache der Text verfasst ist (in unserem Fall ist es ein deutscher Text), so kann man das Ergebnis der Häufigkeitsanalyse mit der Häufigkeitsverteilung in der jeweiligen Sprache vergleichen. (Umlaute und Sonderzeichen haben wir der Einfachheit halber weggelassen.)

Buchstabenhäufigkeit eines mit Cäsar verschlüsselten deutschen Beispieltextes



Buchstabenhäufigkeit in der deutschen Sprache

Da es ein verschlüsselter Text ist, können die Diagramme ja nicht übereinstimmen. Somit muss man das Alphabet so lange verschieben, bis sie ungefähr übereinstimmen. In unserem Fall müssen wir unser Ergebnis so verschieben, bis das „J“ (der häufigste Buchstabe in unserem Text) auf Höhe des „E“ (dem häufigsten Buchstabe in der deutschen Sprache) ist (siehe unten), da wir davon ausgehen, dass das „E“ mit dem „J“ verschlüsselt wurde.



Unser Beispieltext mit Verschiebung

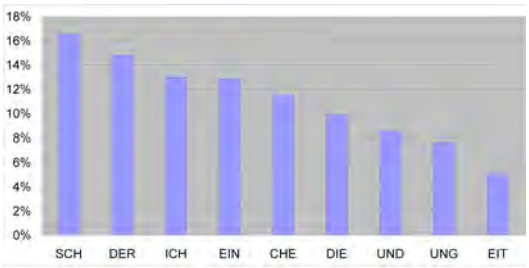
Wir sehen, dass die beiden Diagramme (Diagramm 2 und 3) ziemlich genau übereinstimmen und können nun mit hoher Sicherheit davon ausgehen, dass der Schlüssel 5 ist, da sich das Alphabet um 5 Stellen verschiebt (A → F; E → J).

Substitution

Eine Substitution ist im Vergleich zur Cäsar-Verschlüsselung deutlich schwerer zu knacken, da es weit mehr als 25 verschiedene Schlüssel gibt. Aber auch sie kann man aber mit der Häufigkeitsanalyse angreifen:

Bei Geheimtextbuchstaben, die im Text häufiger vorkommen, kann man mit einer recht hohen Wahrscheinlichkeit die zugehörigen Klartextbuchstaben herausfinden – es sind die häufigsten Buchstaben in der jeweiligen Sprache. Ebenso sucht man nach den häufigsten Buchstabenpaaren und -tripeln im Text. Danach ordnet man die Paare beziehungsweise Tripel den am häufigsten vorkommenden Buchstabenpaaren und -tripeln in der jeweiligen Sprache zu. (Die häufigsten vorkommenden Buchstabenpaare im Deutschen sind ER und EN, und die häufigsten vorkommenden Buchstabentripel sind SCH und DER).

Häufigkeiten der Buchstabenpaare im Deutschen



Häufigkeiten der Buchstabentripel im Deutschen

Bei seltenen Buchstaben liegen die Häufigkeiten sehr nah beieinander. Dadurch kann man nicht genau erkennen, zu welchem Klartextbuchstaben welcher Geheimtextbuchstabe gehört. Man muss dann selbst die Buchstaben zuordnen, indem man den Kontext überprüft und schaut, dass ein sinnvoller Text entsteht.

Vigenère

Nachdem die Vigenère-Verschlüsselung jahrzehntelang ungeknackt blieb, wurde sie Mitte des 19. Jahrhundert von Charles Babbage (1791-1871) erfolgreich angegriffen. Doch Babbage veröffentlichte seine Arbeit nicht, so dass sein Verfahren später von Friedrich Wilhelm Kasiski (1801-1881) wiederentdeckt werden musste; es wurde dann nach ihm Kasiski-Test genannt.

Bei der Vigenère-Verschlüsselung funktioniert die Häufigkeitsanalyse zunächst nicht, da ein bestimmter Klartextbuchstabe mit verschiedenen Schlüsselbuchstaben verschlüsselt werden kann, wodurch sich verschiedene Geheimtextbuchstaben ergeben. Wüssten wir jetzt allerdings die Länge des Schlüssels, dann könnten wir den Text so einteilen, dass er aus verschiedenen Teiltexten besteht, die jeweils mit demselben Schlüsselbuchstaben verschlüsselt wurden. Bei diesen Teiltexten kann dann die Häufigkeitsanalyse angewendet werden, um den jeweiligen Schlüsselbuchstaben herauszufinden.

Beispiel

*EIWEM Q*CMVR JLKPG KKWWT UTMCV GILTW
KKQTE HNGIA *EIWEM Q*CMVR HFPIA QLQPG
FJMUR GRPHA QNCKX TLLIB NKFTF KUJAX
QW*RWX* VNCCM KVRWV GERJK AZRLT UPRDI
KTMUB OGMGM CEAIM QRQBT NCGCM GICHM
GUEGH WGBXI NFKPM URLSM JVYGF GUDDK
EVQXG CEWBB NZRPK ATMCY NZAIX ZTFPG
IVMUV QEDXW GERXT NZLUH TDYIB QEYCW
GRTTL FIMEI KEEXL EISRB CCDDK ULPKB
XRJSN GKMIA GJCCL KKGKX ULZYX EKKPM
VVPTA GMYHM ORHDK KKWDY *EIWEM Q*ICHX
CITRN WJBDG GZLHX EICRR KE*RWX* RRQIM
JVQTV WIGIR QWYRK AGRDZ TRNWB EJWHM
GDKJL VEMIW GGCCW QE*RWX* UVAGX EPMUM
JVYAZ QIGIA OKFTL GTSGB VPGHH PGHEK
GUGRT VVBDG VYCHX EICRR QW*RWX* MVW

Wir wollen also die Länge des Schlüssels herausfinden. Dazu suchen wir im Text nach Wiederholungen. Da der Schlüssel sich ja immer wiederholt, könnte es sein, dass dasselbe Wort oder dieselbe Buchstabenfolge auf denselben Schlüssel fällt. Die erste Wiederholung ist `EIWEMQ`, der Abstand dazwischen beträgt 40 Zeichen.

Wenn wir Glück haben, dann handelt es sich beide Male um denselben Klartext, und es wurde auch derselbe Schlüsselteil verwendet. Jetzt müssen wir herausfinden, wie oft der Schlüssel wiederholt wurde. Bei 40 Buchstaben wäre z. B. eine Schlüssellänge von 5 möglich: Dann wäre der Schlüssel 8 mal wiederholt worden, bis er wieder beim E anfängt. Dagegen wäre eine Schlüssellänge von 7 nicht möglich, da 40 kein Vielfaches von 7 ist. Alle mögliche Schlüssellängen sind: 2, 4, 5, 8, 10, 20 oder 40; 2, 20 und 40 schließen wir allerdings aus, da sie zu kurz bzw. zu lang sind, dass wir nicht glauben, dass sie jemand verwendet. Bei einer Schlüssellänge von 1 läge eine Cäsarverschlüsselung vor.

Wir finden jetzt die möglichen Schlüssellängen von allen Wiederholungen heraus:

			Schlüssellänge						
von	bis	Abstand	4	5	6	7	8	9	10	11
EIWEMQ 1	EIWEMQ 2	40	X	X			X		X	
EIWEMQ 2	EIWEMQ 3	280	X	X		X	X		X	
RWX 1	RWX 2	260	X	X					X	
RWX 2	RWX 3	55		X						X
RWX 3	RWX 4	65		X						

5 ist die einzige Zahl, die bei allen Abständen passen würde, daher ist die wahrscheinlichste Schlüssellänge 5.

Wir denken nun, dass jeder fünfte Buchstabe mit demselben Schlüsselbuchstaben, d.h. demselben Cäsar-Alphabet, verschlüsselt ist. Da wir nun eine Cäsarverschlüsselung haben, können wir die Häufigkeitsanalyse. Wir teilen unseren Text also in die fünf Teile auf, die jeweils mit demselben Schlüssel verschlüsselt wurden:

1. Buchstabe des Schlüssels:
EQJKUGKHEQHQFGQTNKQVKGAUKOCQNGGWNU

JGECNANZIQGNTQGFKECUXGGKUEVGOKEQCC
GEKRJWQATEGVGQUEJQOGVPGVVEQM

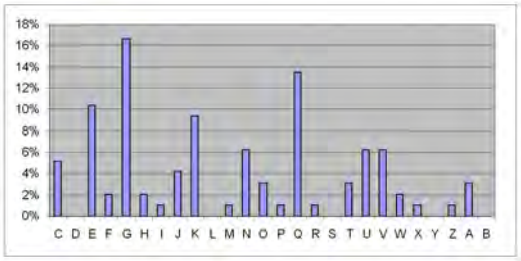
2. Buchstabe des Schlüssels:
ICLKTIKNICFLJRNLKUWNVEZRTGERCIUGFR
VUVEZTZTVEEZDERIEICLRKJKLKVMRKIIIJ
ZIERVIWGRJDEGEVPVIKTPCUVYIWV

3. Buchstabe des Schlüssels:
WMKWMLQGWMPQMPCLFJRCRRRRMMAQGCEBKL
YDQWRMAFMDRLYYTMESDPJMCGZKPYHWWCAB
LCRQQGYRNWKMCRAMYGFSGWGBCCRW

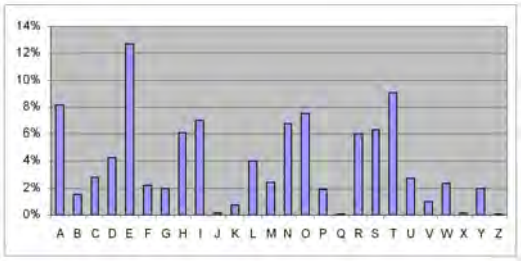
4. Buchstabe des Schlüssels:
EVPWCTTIEVIPUHKITAWCWJLDUGTBCHGXPS
GDXBPCIPUXXUICTEXRDKSICKYPIHDDEHWD
HRWITIRDWHJICWGUAITGHERDHRW

5. Buchstabe des Schlüssels:
MRGTVWEAMRAGRAXBFXXMVKTIBMMTMMHIMM
FKGBKYXGVWTHBWLILBKBNALXXMAMKYMXPG
XRXMVRKZBMLWWXXMZALBHKTGXRX

Auf jeden einzelnen Teil wenden wir nun die Häufigkeitsanalyse an, in unserem Beispiel auf den ersten:



Verteilung im Text



Normalverteilung im Englischen

Wir sehen: Wir haben eine Cäsarverschiebung um 2, also den Schlüsselbuchstaben „C“.

Wir führen das für alle Text durch:
Text 2: Verschiebung 17, Schlüsselbuchstabe R.
Text 3: Verschiebung 24, Schlüsselbuchstabe Y.
Text 4: Verschiebung 15, Schlüsselbuchstabe P.
Text 5: Verschiebung 19, Schlüsselbuchstabe T.
Der Schlüssel ist demnach „CRYPT“.

Wenn wir den kompletten Text mit unserem Schlüssel entschlüsseln, erhalten wir den Klartext:

Cryptology humanity has concerned itself with cryptology for thousands of years however until the middle of the twentieth century it was a topic of importance to a small interested group diplomats and thearmed forces in any military conflict exchange of confidential information and eaves dropping is crucial for survival due to the sensitive subject matter the vast majority of cryptore search was done in secrecy in the past the security of a cryptographic system must not depend on the secrecy of the algorithm the security is only predicated on the secrecy of the key,

Der letzte Satz ist übrigens eine sehr wichtige Regel in der Kryptologie:

„Die Sicherheit eines kryptographischen Systems darf nicht von der Geheimhaltung des Algorithmus abhängen, sondern muss auf der Geheimhaltung des Schlüssels basieren.“

Mathematik im Kurs

Arne Hansen-Dörr, Lea Schön

Modulo-Arithmetik

Wenn wir uns nicht gerade mit Programmieren oder bestimmten Verschlüsselungsverfahren beschäftigten, verbesserten wir unsere mathematischen Kenntnisse. Um die RSA-Verschlüsselung zu verstehen, mussten wir uns zuvor noch einige Grundlagen aneignen. Als erstes behandelten wir die Modulo-Arithmetik. Sie brauchen jetzt keine Angst zu haben, eine neue Rechenart zu lernen, denn Sie benutzen sie praktisch jeden Tag. Wenn Sie zum Beispiel jemand nach der Uhrzeit fragt und wissen will, wie viel Uhr es in 7 Stunden ist, dann schauen Sie auf die Uhr, sehen, dass es 23 Uhr ist und antworten ganz bestimmt *nicht*: „In sieben Stunden ist es 30 Uhr.“, denn diese Uhrzeit existiert nicht. Stattdessen sagen Sie: „In sieben Stunden ist es 6 Uhr.“ Nachdem man

bei 23 Uhr angekommen ist, fängt man wieder bei null Uhr an. Genau das bedeutet „modulo 24“. Man interessiert sich in der Modulo-Arithmetik nur für den Rest bei der Division. Man rechnet also: 23 Uhr + 7 Uhr = 30 Uhr; 30 Uhr : 24 Uhr = 1 Rest 6 Uhr.

Wenn man beispielsweise modulo 5 rechnet, dann benutzt man nur die Zahlen von 0 bis 4. In der Modulo-Arithmetik ergibt also

11 mod 5 = 1 (lies: 11 modulo 5 = 1),
denn 11 : 5 = 2 5er-Rest 1
18 mod 7 = 4, denn 18 : 7 = 2 7er-Rest 4
17 mod 3 = 2, denn 17 : 3 = 5 3er-Rest 2
(statt 5er-, 7er- und 3er-Rest sagt man also mod 5, mod 7 bzw. mod 3)

Auch für die Cäsar-Verschlüsselung benutzt man die Modulo-Arithmetik: Hier rechnet man modulo 26. Jeder Buchstabe von A bis Z wird durch eine Zahl repräsentiert:

A	B	C	D	...	X	Y	Z
0	1	2	3	...	23	24	25

Diese Buchstaben kann man mit Hilfe der folgenden Gleichung verschlüsseln:

Geheimtextzahl
= (Klartextzahl + Verschiebung) mod 26

Beispiel:

Man will den Buchstaben „X“ mit der Verschiebung 6 verschlüsseln. Dazu muss man erst einmal das „X“ in eine Zahl umwandeln. „X“ wird gemäß der oben gezeigten Tabelle der Zahl 23 zugeordnet; also benutzt man statt „X“ die Zahl 23:

Geheimtextahl = (23 + 6) mod 26 = 3

Der Zahl 3 wird laut Tabelle der Buchstaben „D“ zugeordnet. Folglich wird „X“ bei der Verschiebung 6 als „D“ verschlüsselt.

Man kann bei der Cäsar-Verschlüsselung sogar Schlüssel verwenden, die größer als 26 sind. Als Beispiel verschlüsseln wir einmal den Buchstaben „E“, aber diesmal mit einer Verschiebung von 1433:

Geheimtextzahl = (4 + 1433) mod 26 = 7

Die Zahl 7 wird laut Tabelle dem Buchstaben „H“ zugeordnet. Folglich wird „E“ bei der Verschiebung 1433 als „H“ verschlüsselt.

Beim Modulo-Rechnen sind wir außerdem auf Gruppen- und Körper-Eigenschaften gestoßen, wie zum Beispiel auf neutrale und inverse Elemente.

Die Anzahl der Primzahlen ist unendlich

Diesen Beweis kann man nur mit einem Widerspruchsbeweis durchführen, weil man nicht unendlich viele Zahlen nach Primzahlen durchsuchen kann:
Wenn man annimmt, dass es nur endlich viele Primzahlen $p_1, p_2, p_3, \ldots, p_n$ gäbe, diese alle miteinander multipliziert und dann 1 addiert, dann käme als Ergebnis eine Zahl heraus, die durch keine der Primzahlen $p_1, p_2, p_3, \ldots, p_n$ teilbar ist. Jetzt gibt es zwei Möglichkeiten: Entweder ist diese neue Zahl selbst eine neue Primzahl, oder sie ist durch eine andere Primzahl teilbar, die wir noch nicht in der Liste der endlich vielen Primzahlen aufgenommen hatten. Im jedem Falle hätten wir eine neue Primzahl entdeckt. Das ist aber ein Widerspruch zu der Annahme, nur die Zahlen p_1 bis p_n seien Primzahlen. Unsere Annahme war also falsch.

Euklidscher Algorithmus

Mit Hilfe des Euklidschen Algorithmus kann man den größten gemeinsamen Teiler (ggT) zweier Zahlen herausfinden: Man hat zwei vorgegebene Zahlen, zum Beispiel $a = 1239$ und $b = 168$. Jetzt sucht man bei der folgenden Rechnung einen Faktor n für die kleinere der beiden Zahlen, in unserem Falle b, bei dem das Produkt $b \cdot n$ so groß wie möglich, aber dennoch kleiner als a ist. Den Rest notiert man sich an den Rand:

$$1239 = 168 \cdot 7 \quad \text{Rest} \quad 63$$

Nun wird die vorherige Zahl b zur Zahl a und der Rest zur neuen Zahl b, und man sucht wiederum den Faktor n:

$$168 = 63 \cdot 2 \quad \text{Rest} \quad 42$$

Dieses Verfahren führt man solange fort, bis der Rest gleich null ist:

$$63 = 42 \cdot 1 \quad \text{Rest} \quad 21$$
$$42 = 21 \cdot 2 \quad \text{Rest} \quad 0$$

Hier haben wir das Ende erreicht, da der Rest gleich Null ist und sehen, dass der größte gemeinsame Teiler von 1239 und 168 gleich 21 ist.

Eulersche φ-Funktion

Die Eulersche φ-Funktion gibt zu jeder natürlichen Zahl n die Anzahl der positiven ganzen Zahlen kleiner n an, die teilerfremd zu ihr sind. Zu der Zahl n heißt eine Zahl teilerfremd, wenn der ggT von n und der Zahl Eins ist.

Beispiel:
$n = 15$
Von den Zahlen kleiner als 15 sind die Zahlen $\{1, 2, 4, 7, 8, 11, 13, 14\}$ teilerfremd zu 15, und somit ist $\varphi(15) = 8$.

Berechnet man φ von einer Primzahl p, kann man sich die Eigenschaften von Primzahlen zu Hilfe nehmen: Eine Primzahl ist durch keine andere Zahl außer sich selbst und eins teilbar. Folglich ist $\varphi(p) = p - 1$. Das gilt bei jeder Primzahl.

Außerdem gilt $\varphi(p \cdot q) = (p - 1) \cdot (q - 1)$, falls p und q verschiedene Primzahlen sind.

RSA

Lea Schön

Das One-Time-Pad bietet zwar mathematisch gesehen eine sichere Verschlüsselung, aber die Verteilung der Schlüssel ist enorm aufwendig, risikoreich und kostspielig. Da eine sichere und recht einfach zu benutzende Verschlüsselung aber dringend gebraucht wurde, suchten verschiedene Forschergruppen nach einer Möglichkeit, die Schlüsselverteilung zu umgehen. Dazu entstand ein kleines Gedankenexperiment:

Nehmen wir an, eine Person – Alice – will einer zweiten Person – Bob – eine höchst vertrauliche Nachricht schicken. Also legt Alice ihren Brief in eine kleine Kiste und verschließt diese mit einem Vorhängeschloss, dessen Schlüssel nur sie selbst hat. Dann schickt sie die Kiste an Bob. Dieser kann die Kiste zwar nicht öffnen, aber er kann ebenfalls sein eigenes Vorhängeschloss daran hängen. Danach kann er die Kiste zurück an Alice schicken. Sie entfernt ihr Vorhängeschloss und schickt die Kiste, die jetzt nur noch mit Bobs Schloss gesichert ist, an Bob. Der kann die Kiste öffnen und die Nachricht lesen.

Dieses Verfahren funktioniert ohne persönlichen Schlüsselaustausch, aber es ist noch immer aufwendig, weil man die Nachricht oft hin- und herschicken muss. Außerdem ist das Prinzip mit den Kisten in der Realität nicht zu verwirklichen.

So entstand ein zweites Gedankenexperiment:

Bob hinterlegt an vielen Orten über die ganze Welt offene Vorhängeschlösser, zu denen nur er selbst einen Schlüssel hat. Wenn jetzt Alice eine Nachricht an Bob schicken will, besorgt sie sich eines von Bobs hinterlegten Vorhängeschlössern, legt ihre Nachricht in eine kleine Kiste und verschließt diese mit einem von Bobs Vorhängeschlössern. Dann schickt sie die Kiste an Bob. Dieser kann die Kiste dann mit seinem Schlüssel öffnen und die Nachricht lesen. Falls Eve die Kiste abfangen würde, käme sie nicht an die Nachricht heran, da ja nur Bob alleine den Schlüssel für die Kiste hat.

Auf diesem Grundgedanken aufbauend entwickelten Ronald Rivest, Adi Shamir und Leonard Adleman 1977 das RSA-Verfahren. Es wurde später nach den Anfangsbuchstaben ihrer Nachnamen benannt.

Mathematisch gesehen basiert RSA auf Einwegfunktionen, d. h. Funktionen, die nicht umkehrbar sind, und auf der Modulo-Arithmetik.

Die Mathematik von RSA ist ziemlich kompliziert, aber die Schlüsselherstellung funktioniert relativ einfach. Hier die Schritte im Einzelnen, jeweils mit einem Beispiel:

1. Finde zwei Primzahlen p und q
$(p; q)$ $\quad p = 17; q = 11$

2.Berechne das Produkt n von p und q
$n = p \cdot q$ $\quad n = 17 \cdot 11 = 187$

3. Berechne $\varphi(n)$
$\varphi(n) = (p-1) \cdot (q-1)$
$\varphi(n) = (17-1) \cdot (11-1) = 160$

4. Finde eine Zahl $1 < e < \varphi(n)$ so, dass e und $\varphi(n)$ teilerfremd sind.
$e = 7$ (7 und 160 sind teilerfremd)

5. Finde eine Zahl d so, dass gilt:
$e \cdot d \mod \varphi(n) = 1$
$d = 23$ $\quad 7 \cdot 23 \mod 160 = 1$

Der öffentliche Schlüssel setzt sich aus n und e zusammen.
$(n; e)$ $\quad (187; 7)$

Der private Schlüssel setzt sich aus n und d zusammen.
$(n; d)$ $\quad (187; 23)$

In Wirklichkeit sind die Primzahlen so groß wie nur möglich, um ein hohes Maß an Sicherheit zu gewährleisten. Denn wenn man RSA knacken möchte, müsste man n faktorisieren, das heißt, man müsste die beiden Faktoren p und q herausfinden. Das geht aber nur durch systematisches Ausprobieren, denn es gibt keine Formel zum Faktorisieren von Zahlen. Je größer also n ist, desto länger bräuchte man, um die beiden Faktoren p und q herauszufinden, und desto höher ist die Sicherheit dieses Schlüssels. Die Sicherheit von RSA basiert also nicht etwa auf einem komplizierten Algorithmus, sondern auf der Größe der Primzahlen, die für die Erstellung des Schlüssels verwendet werden.

Seinen privaten Schlüssel hält jeder geheim, seinen öffentlichen Schlüssel stellt man ins Internet.

Zur Verschlüsselung eines Textes benötigt man den öffentlichen Schlüssel der Person, der man die Nachricht schicken will.

Geheimtext und Klartext sind in Zahlen umgewandelt, da Computer nur mit Zahlen arbeiten und RSA nur mit Zahlen funktioniert.

Sei m eine Klartextzahl. Dann kann man die zugehörige Geheimtextzahl c wie folgt berechnen:
$c = m^e \mod n$
$m = 88$ $\quad c = 887 \mod 187 = 11$

Zur Entschlüsselung eines Textes, den man von einer anderen Person bekommen hat, benötigt man seinen eigenen privaten Schlüssel $(n; d)$:
$m = c^d \mod n$
$c = 11$ $\quad m = 1123 \mod 187 = 88$

Man nennt RSA ein asymmetrisches Verschlüsselungsverfahren, weil man zum Ver- und Entschlüsseln zwei unterschiedliche Schlüssel benutzt. Ein asymmetrisches Verschlüsselungsverfahren hat viele Vorteile, so kann man zum Beispiel mit RSA nicht nur E-Mails verschlüsseln, sondern auch digital signieren. Das ist wichtig, weil ansonsten ein Dritter die Nachricht abfangen und etwas dazuschreiben könnte.

Um eine E-Mail digital zu signieren, verschlüsselt Alice ihre Nachricht mit ihrem eigenen privaten Schlüssel, denn dann kann Bob die Nachricht mit Alice öffentlichem Schlüssel entschlüsseln und somit sichergehen, dass die E-Mail auch wirklich von Alice stammt.

RSA ist auch heute noch das am häufigsten benutzte und sicherste Verfahren zur Verschlüsselung von Texten.

Besuch von Herrn Esslinger

Nadine Feirer

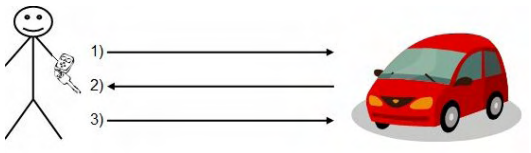
Endlich war es soweit, heute würde Herr Esslinger, der bei der Deutschen Bank für IT-Sicherheit und Kryptographie zuständig ist, zu uns kommen, uns einiges über diese erzählen und hoffentlich auch unsere endlosen Fragen beantworten.

Herr Esslinger erzählte uns, dass sein Interesse für Kryptographie mit einem Volkswirtschaft- und Informatikstudium angefangen hat. Dann ging's zu SAP, bei der er als Verantwortlicher für die firmeninterne IT-Sicherheit arbeitete. Nun ist er bei der Deutschen Bank und dort Leiter für Technology Research und Verantwortlicher für das Kryptographie-Kompetenz-Center und PKI (Public-Key-Infrastruktur). PKI ist ein System, welches es ermöglicht digitale Zertifikate auszustellen, zu verteilen und zu prüfen.[2] Außerdem ist er der Entwickler von CrypTool[3], ein Programm mit dem kryptographische Verfahren angwendet und analysiert werden können und das jeder aus dem Internet herunterladen kann. Zudem unterstützt CrypTool ein modernes Lernen an Schulen und Hochschulen. Des weiteren ist er als Dozent am Institut für Wirtschaftsinformatik der Universität Siegen tätig.

Er erzählte uns vom alltäglichen Gebrauch von Kryptographie, und manchmal waren wir ganz schön überrascht, dass alles gar nicht so sicher ist, aber gleichzeitig auch fasziniert. Als Beispiel erzählte er uns, wie ein elektronischer Autoschlüssel funktioniert.

Eine erste Methode war, dass der Schlüssel ein geheimes Codewort an das Auto sendet. Dieses Verfahren ist jedoch sehr unsicher: Ein Dieb könnte die Verbindung abhören und das ausgespähte Codewort erneut an das Auto senden („Replay-Attacke“).

Mithilfe der RSA-Verschlüsselung kann man dies deutlich sicherer gestalten:



Öffnen einer Autotür mit RSA

Zuerst schickt der Schlüssel einen Code (etwa eine Zahl) an das Auto, der mit dem öffentlichen Schlüssel des Autos verschlüsselt ist (1), danach berechnet das Auto aus diesem Code einen neuen (z. B. das Quadrat dieser Zahl), verschlüsselt ihn mit dem öffentlichen Schlüssel des Autoschlüssels und schickt ihn zurück (2). Der Schlüssel berechnet wiederum aus dem zurückgesendeten Code einen neuen (indem er etwa 1 addiert)und schickt diesen neuen Code an das Auto (3). Die Autotür öffnet sich.

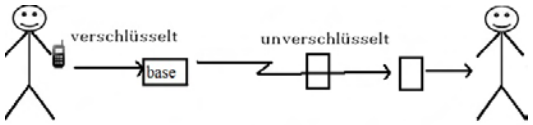
Aber Kryptographie wird nicht nur zum Öffnen einer Autotür verwendet, auch beim Online Banking wird sie benutzt. Hier die drei aktuellen Möglichkeiten, bei den ersten beiden Verfahren sind die TAN-Nummern bekannt, bei Verfahren 3 jedoch nicht:

1. Man gibt eine PIN (Persönliche Identifikationsnummer) und eine TAN (Transaktionsnummer) ein, für die TAN-Nummer erhält man eine Liste, aus der man eine beliebige TAN auswählen kann.
2. Als Weiterentwicklung benutzt man eine indizierte TAN-Liste. Jetzt kann man sich die Nummer nicht mehr aussuchen, sondern bekommt die Indexnummer und muss dann die zugehörige TAN eingeben.
3. Ein ganz neues Verfahren nutzt eine Smartcard, die mit privaten und öffentlichen Schlüsseln umgehen kann und nicht mehr mit TAN-Nummern arbeitet. Man muss jedoch ein Kartenlesegerät anschaffen.

Aber diese zwei Beispiele sind wahrscheinlich für die meisten von uns noch nicht so wichtig, interessanter sind für Jugendliche das Telefonieren mit dem Handy und das Verschicken von E-Mails, die beide nicht wirklich sicher sind und die eigentlich jeder lesen oder mithören kann. Zum Beispiel ist beim Handy nur der erste Teil des Weges verschlüsselt und somit sicher, der Rest ist unverschlüsselt und leicht mitzuhören. Dies kann man indem man einfach selbst eine Basisstation nachbaut, die eine höherer Sendeleistung hat. Dann schickt das Handy die Informationen nämlich nicht an die richtige Basisstation, sondern an die Nachgebaute, somit kann man nicht nur sehr private Informationen des anderen mithören und sie später verwenden, sondern kann zusätzlich auch sehr leicht Personen orten. Dies ist jedoch gesetzlich verboten und nicht zur Nachahmung empfohlen.

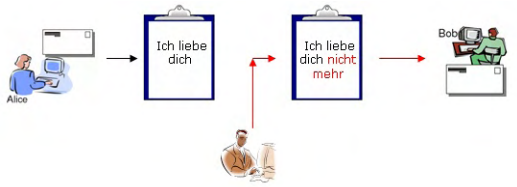
[2] `http://de.wikipedia.org/wiki/Public-Key-Infrastruktur` (letzter Besuch 18.10.2008, 15:48Uhr)

[3] CrypTool: `www.cryptool.de` (letzter Besuch 18.10.2008, 15:48 Uhr)



Weg des Telefonats

Bei den E-Mails ist es noch leichter, da sie gar nicht verschlüsselt sind. So kann z. B. jemand etwas hinzufügen und der Empfänger (Bob) weiß nicht, dass dies gar nicht der Versender (Alice) geschrieben hat. Dies wäre z. B. mit einer Signatur nicht möglich.



Abfangen und Verfälschen einer E-Mail

Deshalb gibt es Programme, mit denen man seine E-Mail verschlüsseln und auch signieren kann. Vorraussetzung ist, dass der Empfänger dieses Programm auch hat, sonst kann er die Mail nicht lesen. Ein solches Programm ist z. B. PGP (Pretty Good Privacy), eine freie Version davon ist GnuPG (Gnu Privacy Guard)[4], die jeder aus dem Internet herunterladen kann.

Dieses Programm erstellt einen privaten und einen öffentlichen Schlüssel. Mit dem öffentlichen Schlüssel des Empfängers kann der Versender einer E-Mail diese verschlüsseln und nur der Empfänger kann sie mit seinem privaten Schlüssel wieder entschlüsseln. Außerdem kann man mit seinem privaten Schlüssel eine Mail signieren, damit der Empfänger sicher sein kann, dass die E-Mail wirklich vom Versender stammt.

Nachdem Herr Esslinger uns dies alles erklärt hatte, waren endlich unsere Fragen an der Reihe. Eine unserer Fragen war z. B., was wäre, wenn RSA geknackt würde. Daraufhin antwortete er, dass dies furchtbar wäre, aber es Alternativen geben würde. Es müssten jedoch schnell neue Hard- und Software installiert werden und der Großteil der Kommunikation würde am Anfang nicht funktionieren. Wir fragten auch, ob es einmal einen unbrechbaren Algorithmus geben wird. Daraufhin sagte er, es gibt nur einen, das One-Time-Pad. Außerdem wollten wir wissen, ob alle Länder RSA benutzen. Seine Antwort war, dass dies alle Länder außer Russland verwenden, da in Russland RSA verboten ist. Dafür hat Russland einen RSA-Ersatz, der es der Regierung ermöglicht, die verschlüsselten Nachrichten zu lesen.

Nach unserer Fragerunde schaute er uns noch ein bisschen bei der Kursarbeit zu und unterstützte uns dabei. An diesen Tag werden wir uns noch lange erinnern.

[4] GnuPG: `http://www.gnupg.org`

Rätselspaß

Patrick Caspari

Jetzt können alle Interessierten ausprobieren, ob sie das Zeug zum Kryptoanalytiker haben.[5]

Zum Einsteigen beginnen wir mit etwas Einfachem, einem Cäsar-verschlüsselten Text:

```
KYYZG TJGTY KOTKY HAXMK YHXAK YZATM
JKXXO ZZKXL OVYOT BURRK XXAKY ZATMJ
GNUKX ZKKXB UTATZ KTQXG INATJ JGINZ
HKOYO INOIN YINGA SGRTG INATJ RKNTZ
KYOIN OTBUR RKXXA KYZAT MCKOZ AKHKX
JOKHK YGMZK HXAKY ZATMN OKXBK XRUXK
XTATG RYHGR JFAKX YZJKT NKRSA TJJGT
TJKTN GRZCU XGALB KXLUR MKTJY ZAXYK
OTFOK RKXVG AYKTR UYHOY ATZKT LOKRA
TJNOK XBKXR UXKXJ AXINY KOTYZ XKHKT
GRYJX OZZKY TATGA INTUI NYKOT RKHKT
```

[5]Die Benutzung von technischen Hilfsmitteln lindert Spaß. Leute, die die Texte in weniger als einer halben Stunde entschlüsseln können, sind vom Wettbewerb ausgeschlossen. Bei Risiken und Nebenwirkungen lesen Sie die Packungsbeilage und fragen Sie ihren Arzt oder Kursleiter. Wer als erstes alle drei entschlüsselten Texte einem Kryptographie-Kursler vorlegt, bekommt einen Bundesverdienstkeks am Bande, gegen Aufpreis auch mit Schleife.

```
GTJKS KXMGT FHKYU TJKXY NOTMJ KXHRK
INYIN GJKTC GXTAX MKXOT MATJJ OKSUX
GRBUT JKXMK YINOI NZLGR RYLGR RKTJJ
ABUSJ GINBK XYINC GTJKY ZYUHX KSYHK
BUXJA ATZKT RGTJK YZ

JGYKT JKJKY XOZZK XLOVY BUTNK OTFKX
NGXJZ
```

War doch ganz einfach, oder? Darum gehen wir jetzt eine Stufe höher und machen mit einem etwas schwierigeren weiter: Einem 649 Zeichen langen Vigenère-Text.

```
VMFWK GWXWG UNVLD TVPZR VWDKZ GZPOB
JISYB VYVXI ZRWKT RRPKT KNCGZ TRIRR
LWIBH WCLKR YFTRL TLDPT ZALXI MOYMF
HBHJE XPVLX EIVIC BTLLK OSFKW CTDHX
IBVRL AEXZX MFVGZ GZIWP RVOYA FIJTR
RYKHI BJIAC IXJGZ GMDMV RPYPU IELYE
QWLTK NEHSI XNIDA IXHAT QEVMU WQKXJ
WAVQX XIFHS XFGDX OJGZL EZIKI FOBEI
IYBFL LJWMA CYYIS GJEPY XKDDX GRSPA
RODXT SJPMA VRPYP LRVWZ KRGZM XUMWH
EGBID WISMH MFSLK KEEOM YTVKX LWUWP
YLPGT WDXWA RLXZR WOUVR XPPVL FJPYX
ZVASC HOMVX XKEKE MHNXD KRLKM KIFYX
WEFIA YKKIL TCHIV QNXIN TVUXY VLEBD
SWGHO KZWUW LVMFT QBXEF IMCVY IFSRK
VRQIG XXXWC CBTLX QPZRV CHHSH WXJDF
GHYPW YDLRV WYGEI FQVSE CISIX XVGHW
```

```
DXZPH ISGVW VICBT LLHFX IHWRO DXEJR
FXIOW CRLTI OPVWR WKICJ MUWRB VVTTM
NDVXH NEVZP RNXCX WOBEI IEDLR GZTWZ
IXLWU IFFOK LJPPC XIZTM WTXIF IXBTL
LKOKR HWIBE VMUWX OKKI
```

Na, geschafft? Nicht schlecht. Wenn Sie den nächsten auch herauskriegen, können Sie sich bei der NSA als Kryptoanalyse-Fachmann bewerben. Bühne frei für den 996-Zeichen-Substitutions-Text.

```
GHTFH TOHSV IHUIH SBSWB LLHTH QIHWL
HFIHP XWWHO WHSWS OWGHT IZOHW ZOWGP
QHTQY FQUWH OWHSW SOWGH TIUQH FHWOW
GPQHT QYQWG HTUIB GIZTO QIBMX LXTBG
XHTSQ HUUVQ AHOWG YHSXH TIHGH WXLUH
WUQVB LIHTP XWZOH WZVXW BIHWD OTGHV
QAHPX WFBOH TLLXC GXLUH WVQIH QWHTB
CIYHA XHEZI BWLBU UDBTH QWFHU OMSGH
TUMSD QHYHT VOIIH TDHLM SHTXL UHWFT
BISBH SWMSH WOUHT PQHTH WYHGB MSIHD
HQLBF HTGBU UIBVV SQTWQ WIBAI YHFLQ
HFHWD BTLHF IHGHT SBSWD HQIHT BLUUH
QWQMS IUYHU MSHSH WHTAX WWIHL BOZHW
OWGBO ZHQWH TUIBW YHUQI HWHTP HTUOM
SIHUX YBTOA TBHSH WGQHU BLLHU YBHWL
QMSXS WHAXE ZXLUH WZOHI IHTIH GBUIQ
HTQWG HWZXL YHWGH WIBYH WVQIS QLZHH
QWHTE QEHII HHTIT XEZIH QSVHQ WHVQU
MSOWY BOUVQ LMSOW GDBUU HTQWG QHUEH
QUHTX HSTHB OUUHT GHVVX MSIHV QAHAL
HQWHA XHTWH TXLUH WUIHL LIHVQ AHBOU
OWGPH TLBWY IHZOH WZOWG DBWQY MHWIH
QWITQ IIGHW AXEZS BHIIH HTYHT WUHEB
TBIPX TYHZO HTIGQ HUHTD BTBLL HTGQW
YUPXW HQWHT ABIHY HZTHU UHWDX TGHWG
HTSBS WPHTG QHWIH FQUOP QHTIB OUHWG
ZOHWZ SOWGH TIGXL LBTQV VXWBI HTDOT
GHUHS TZHII BHSWL QMSDQ HGHTU EBHIH
HLPQU BSLTH QMSHL BWGDQ TIHQW GHWOU
BAXHE ZIHWG BTBOZ SQWQS THSBH SWHQW
GHTSX ZZWOW YBOZH QWHLO ATBIQ PHHQW
WBSVH ROHLL HBLLH QWBLL HBWGH THWIQ
HTHUI BTFHW THMSI UMSWH LL
```

```
BOUGH VYHXU HEIHV FHTBO UYBFH 2008
```