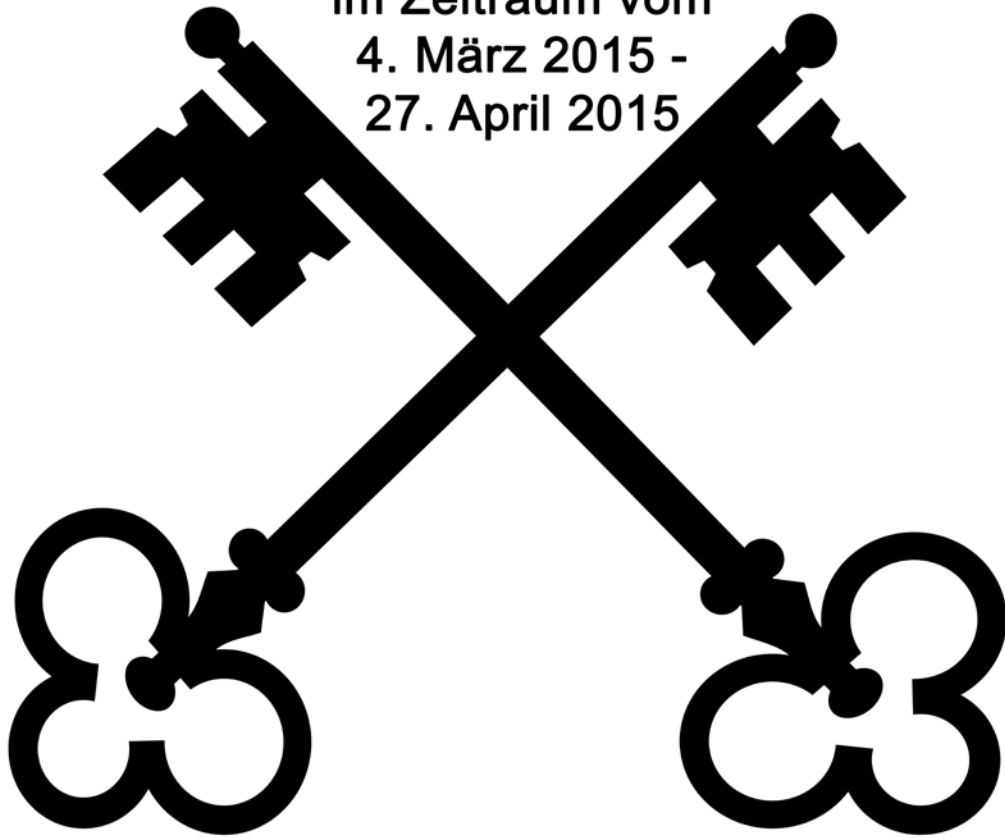


The Daily Cryptograph

Entwicklung einer Programmsammlung
zur Verschlüsselung im Rahmen
des Seminarfaches am Gymnasium Trittau
im Zeitraum vom
4. März 2015 -
27. April 2015



Gruppenmitglieder

Fabian K., Henrik T., Jan R., Melvin H.,
Mirko P., Paul S. und Thore K.

Inhalt

Einleitung	4
Kryptographie	4
Bedeutende Verschlüsselungen der Geschichte	4
Monoalphabetische Substitution ²	4
Polyalphabetische Substitution/ Vigenère-Chiffre	5
Die Enigma.....	6
Die unknackbare Verschlüsselung- Das One-Time-Pad (OTP)	7
UTF-8 (Universal Character Set Transformation Format)	8
Abgrenzung zwischen Kryptographie und Steganographie	8
Symmetrische und asymmetrische Verschlüsselung.....	8
Aktualität.....	9
Der Advanced Encryption Standard - AES.....	11
Geschichte des AES.....	11
Funktionsweise von Rijndael	11
Ablauf von Rijndael	12
Die einzelnen Funktionen erläutert.....	12
Kryptoanalyse von AES	14
Betriebsmodi von Blockchiffren.....	15
CBC – Cipher ⁸ Block Chaining Mode.....	15
Das RSA-Kryptosystem - Asymmetrische Verschlüsselung.....	16
Generierung eines Schlüsselpaars:.....	16
Euklidischer Algorithmus	16
Verschlüsselung und Entschlüsselung:	17
Signieren von Nachrichten:.....	17
Kryptoanalyse.....	17
Elgamal mit ECC.....	19
Allgemein zu Elgamal	19
Historisch zu Elgamal	19
ECC - Elliptic Curve Cryptography.....	19
Rechenregeln auf elliptischen Kurven	20
Die Addition auf elliptischen Kurven.....	20
Verdopplung eines Punktes	20
Multiplikation eines Punktes mit einem Skalar	20
Elgamal mit ECC	20
Die Schlüsselgenerierung	20
Die Verschlüsselung	21
Die Entschlüsselung.....	21
Kryptoanalyse.....	21
Benutzeroberfläche	23
Der Startbildschirm:	23
AES:	24

RSA:	25
Fazit	26
Glossar	28
Anhang	31
1: Allgemein	31
2: AES	38
3: Das Elgamal-Kryptosystem	43
Quellverzeichnis	45
Literatur	45
Allgemeine Quellen	45
AES Quellen	46
RSA Quellen	46
Elgamal Quellen	47
Bildquellen	47

Einleitung

Das grundlegende Ziel dieses Projektes war es, ein Programm zu schreiben, welches moderne Verschlüsselungen umsetzt. Die Hauptaspekte waren daher das Verstehen der Algorithmen und somit der zugrundeliegenden Mathematik, die programmiertechnische Umsetzung dieser und schließlich das Erstellen einer Benutzeroberfläche, um die Scripte leicht nutzbar zu machen. Ein besonderes Augenmerk sollte natürlich auf der Sicherheit, um die es in der Kryptographie schließlich geht, liegen. Als Frage kann man somit formulieren: Ist es einer Schülergruppe im Rahmen des Projektunterrichts möglich, ein funktionierendes, sicheres, aber auch zeitgleich ein leicht nutzbares Verschlüsselungsprogramm zu schreiben?

Kryptographie

Unter dem Begriff Kryptographie versteht man allgemein die Verschlüsselung von Texten mittels eines bestimmten Verfahrens.

Sie hat vorwiegend zum Ziel, den Inhalt des Textes Außenstehenden zu verschleiern und so die Privatsphäre und die Information an sich zu schützen. Der Gegenspieler zur Kryptographie ist die Kryptoanalyse, welche es zum Ziel hat, ebendiese Verschlüsselung zu brechen. Die Kryptographie ist schon fast so alt wie die Zivilisation selbst, doch hat sie sich im Laufe der Zeit stark verändert. Die Begebenheit, ob zu einem Zeitpunkt gerade die Kryptographie, oder die Kryptoanalyse überlegen war, hat oft entscheidenden Einfluss auf das Geschehen, z.B. auf diplomatische Verhandlungen, Kriege oder Ähnliches, genommen.

Bedeutende Verschlüsselungen der Geschichte

Monoalphabetische Substitution²

Die erste Form der Verschlüsselung wird als monoalphabetische Substitution bezeichnet. Hierbei handelt es sich um eine einfache Vertauschung der Buchstaben im Alphabet. Es wird jedem Buchstaben ein Substituent, also ein anderer Buchstabe fest zugewiesen. Dies ist im Anhang in Tabelle 1.1 dargestellt und anhand eines Beispiels erläutert.

Um die Verschlüsselung rückgängig zu machen, muss dem Empfänger zwar auch das „neue“ Alphabet vorliegen, jedoch ist dann eine verschlüsselte Kommunikation möglich. Diese Verschlüsselung stellte eine lange Zeit eine sichere Möglichkeit dar, um Informationen zu verbergen. Erst im siebten Jahrhundert gelang es arabischen Gelehrten eine Methode zu entwickeln, mit deren Hilfe man auf das „neue“ Alphabet schließen und somit alle derartig verschlüsselten Nachrichten lesen konnte. Die beiden Bedingungen waren, dass man wusste in welcher Sprache die verschlüsselte Nachricht verfasst war und dass der Text relativ lang war. Der Ansatz war folgender: Jeder Buchstabe kommt in einer Sprache unterschiedlich häufig vor. Hatte man nun genug verschlüsselten Text vorliegen, konnte man z.B. darauf schließen, dass der häufigste auftauchende Buchstabe das Pendant zum „e“ sein muss, da das „e“ mit 17,4% relativer Häufigkeit mit Abstand am häufigsten in der deutschen Sprache verwendet wird. Mit Hilfe einer Häufigkeitstabelle (siehe Anhang 1.5) konnte man so Stück für Stück jedem Buchstaben seinen jeweiligen Partner zuweisen. Dieses Wissen gelangte jedoch nie nach Europa, sondern wurde dort erst erheblich später selbst entdeckt.

Eine andere bekannte monoalphabetische Verschlüsselung ist die Caesar-Verschlüsselung. Hierbei wird das „neue“ Alphabet nicht willkürlich gewählt, sondern das Zugrundeliegende einfach um eine bestimmte Anzahl an Stellen verschoben. Auch hierfür findet sich in Tabelle 1.2 eine Darstellung.

Da es sich ebenfalls um eine monoalphabetische Verschlüsselung handelt, ist auch sie durch einen Häufigkeitsangriff leicht zu knacken, wie auch die meisten anderen Variationen dieser.

Bei einer dieser Variationen wird das „neue“ Alphabet nicht um eine gewisse Anzahl verschoben, sondern ein „Passwort“ gewählt, welches an den Anfang geschrieben wird. Die restlichen Buchstaben werden danach in der regulären Reihenfolge ergänzt. Bedingung für das Passwort ist jedoch, dass kein Buchstabe doppelt vorkommt. Dies ist in Tabelle 1.3 im Anhang mit einem Beispiel erläutert.

Polyalphabetische Substitution/ Vigenère-Chiffre

Da die monoalphabetische Verschlüsselung geknackt war und auch jeweilige Variationen dieser, musste eine neue Verschlüsselungsmethode gefunden werden. Dies stellte sich als Problem heraus, da es den Kryptoanalytikern immer möglich war, wenn auch teilweise nur mit größerem Aufwand, die Verschlüsselungen zu knacken. Erst im 16. Jahrhundert gelang es Blaise de Vigenère eine vorerst sichere Methode zu entwickeln. Diese Vigenère-Chiffre und später auch „*Le Chiffre indéchiffrable*“ basierte auf der klassischen Caesar-Verschlüsselung. Jedoch wird nicht eine mögliche Verschiebung verwendet, sondern alle 26. Das entstehende Quadrat stellte nun die Grundlage für die Verschlüsselung dar. Diese polyalphabetische Tabelle ist im Anhang 1.4 zu sehen.

Es musste lediglich ein Passwort ausgetauscht werden, damit die Nachricht ver- und entschlüsselt werden konnte. Die Verschlüsselung ist denkbar einfach. Zur Verschlüsselung des ersten Buchstaben wird in der Zeile des Anfangsbuchstaben des Passworts und in der Spalte des zu verschlüsselenden Buchstabens abgelesen. Für den zweiten Buchstaben wird entsprechend der zweite des Passworts verwendet. Wenn der letzte Buchstabe des Passworts verwendet wurde, beginnt man wieder von vorne, bis der gesamte Text Verschlüsselt ist. Ein Beispiel hierfür findet sich unter der Grafik 1.4. Da bei diesem Verfahren die Buchstaben keine festen Paare bilden, ist ein einfacher Häufigkeitsangriff nicht mehr zielführend. So kam es dazu, dass die Kryptoanalytiker keine Handhabe gegen dieses Verfahren mehr hatten und zunächst auch keine fanden. Man bezeichnete die Vigenère-Chiffre daher auch als „*Le Chiffre indéchiffrable*“, da es schier unmöglich schien sie zu knacken. Erst im Jahr 1854 gelang es dem britischen Mathematiker Charles Babbage. Sein Ansatz basiert darauf, dass der Schlüssel relativ kurz ist und somit immer wieder die gleichen Teil-Alphabete verwendet werden. Lautet der Schlüssel z.B. „kurz“, so werden nur die Alphabete der Spalten k, u, r und z verwendet. Das bedeutet, dass man wieder auf die Häufigkeitsanalyse zurückgreifen kann, da alle, in unserem Fall vier Buchstaben, die gleiche Verschiebung benutzen. Außerdem zeigte Babbage, dass man durch den Vergleich des verschlüsselten Textes mit einer verschobenen Version dieses auf die Schlüssellänge schließen kann. Da Babbage jedoch die Lösung nicht veröffentlichte, dauerte es weitere neun Jahre, bis auch der Preuße Friedrich Kasiski auf die gleiche Weise die „*Chiffre indéchiffrable*“ knackte und somit wieder keine sichere Kommunikation mehr möglich war.

Die Enigma

Auch wenn im Laufe der Zeit viele weitere Verschlüsselungen entworfen und dann auch wieder geknackt wurden, stellt die Enigma wohl das beste Beispiel dafür dar, welche Folgen eine sichere und später geknackte Methode haben kann. Die Enigma hebt sich von bisherigen Verschlüsselungsmethoden dadurch ab, dass es sich hierbei um eine Maschine und nicht um manuelle Arbeit handelt. Das Patent für die „erste“ Enigma wurde im Jahr 1918 von Arthur Scherbius angemeldet. Dieser bot noch in diesem Jahr die Enigma der deutschen Marine zum Kauf an. Diese lehnt jedoch ab, da sie keine Notwendigkeit sah, auf maschinelle Kryptographie zurückzugreifen. Daraufhin verkaufte Scherbius die Maschine auf dem zivilen Markt, wo sie schließlich im Jahr 1926 doch noch auf das Interesse des Militärs stieß. Nach zunächst testweiser Verwendung kam es dann zur standardmäßigen Nutzung. Die Massenproduktion begann ab 1929 und sollte besonders durch die folgende Aufrüstung der Wehrmacht noch weiter gesteigert werden. Man schätzt, dass ca. 100.000 Enigmas hergestellt wurden. Im Laufe der Jahre wurde die Enigma immer weiter verbessert und mit weiteren Variationsmöglichkeiten ausgestattet, sodass es den ausländischen Geheimdiensten nur zeitweise möglich war, abgehörte Funksprüche der Deutschen zu verstehen. Kurz vor Kriegsbeginn 1939 verbesserten die Deutschen die Enigma erneut, sodass alle bisher noch möglichen Angriffe wirkungslos wurden. Da die gesamte deutsche Kommunikation über Funk lief, konnten zwar alle Meldungen mitgeschnitten werden, jedoch war es nicht mehr möglich diese auch zu verstehen. Erst zwei Jahre und sehr viel investierte Arbeit seitens des britischen Militärs später gelang es erneut die Verschlüsselung zu brechen. Zwar konnte man nicht sofort aktiv auf jede abgefangene und entschlüsselte Mitteilung reagieren, da die Deutschen dann gemerkt hätten, dass ihre Kommunikation nicht mehr geheim war, jedoch war nun ein taktisch viel effizienteres Vorgehen möglich.

Es wird geschätzt, dass ohne die Entschlüsselung der Enigma der Zweite Weltkrieg etwa 2 Jahre länger gedauert und entsprechend mehr Menschenleben gekostet hätte.

Wie funktioniert die Enigma?

Die Enigma besteht aus einer Tastatur, zum Eingeben des Textes, einer Verschlüsselungseinheit und einem Lampenfeld für die Ausgabe des Verschlüsselten.

Die Verschlüsselungseinheit besteht grundlegend aus drei einzelnen Elementen: Ein Schema der Elemente sowie ein Bild der Enigma findet sich in den Abbildungen 1.6 und 1.7.

Der wohl wichtigste Teil sind die sogenannten Walzen. Diese sind ineinander verdrahtet und drehbar gelagert. Läuft jetzt ein Signal in die erste Walze hinein, so wird es durch die Verdrahtung an einer anderen Stelle an die zweite Walze weiter gegeben. Nach jedem durchgelaufenen Signal dreht sich die erste Walze ein Stück weiter. Hat sie sich einmal komplett um sich selbst gedreht, so dreht sich dann die zweite Walze ein Stück. Die Funktionsweise des Weiterdrehens kann man sich also so vorstellen wie den Kilometerzähler im Auto. Durch dieses sich gegeneinander

Verdrehen wird dafür gesorgt, dass ein Buchstabe immer mit einem anderen verschlüsselt wird.

Der zweite Teil ist der sogenannte Reflektor oder auch die Umkehrwalze. Dieser befindet sich hinter den Walzen und „reflektiert“ das Signal, sodass es auf dem Rückweg noch einmal alle Walzen durchläuft.

Bei dem dritten Teil handelt es sich um ein Steckbrett, welches sich direkt hinter der Tastatur befindet. Hier kann man die Eingabe mit den Walzen auf unterschiedliche Weisen verbinden. Alle „Versionen“ der Enigma verfügen über diese drei Elemente, auch wenn die Anzahl der Steckverbindungen oder Walzen variiert wurde. Die Sicherheit der Enigma kommt daher, dass es extrem viele mögliche Einstellungen, auch als Schlüsselraum bezeichnet, gibt. Angenommen eine Enigma hat 3 Walzen und 6 Steckverbindungen beim Steckbrett, so ergibt sich die Gesamtzahl an möglichen Einstellungen aus:

- den Walzenstellungen. Jede der drei Walzen hat 26 mögliche Stellungen.
→ $26 \cdot 26 \cdot 26 = 17576$ Möglichkeiten
- den Walzenanlagen. Man kann die Walzen untereinander vertauschen mögliche Reihenfolgen sind also: 123,132,213,231,312,321.
→ 6 Möglichkeiten
- dem Steckbrett. Wenn man sechs von 26 Buchstabenverbindungen vertauschen kann ergibt sich die Anzahl von 100391791500 möglichen Variationen.

Alle drei Komponenten ermöglichen also $17576 \cdot 6 \cdot 100391791500$ Einstellungsmöglichkeiten.

Das sind gerundet 10 Milliarden mögliche Variationen. Da die Deutschen alle 24 Stunden die Einstellungen änderten, ist die Leistung der polnischen und britischen Kryptoanalytiker, dieses Problem zu lösen, umso beachtlicher.

Die unknackbare Verschlüsselung- Das One-Time-Pad (OTP)

Bei diesem Verfahren handelt es sich, wie bei der Vigenère-Chiffre um eine polyalphabetische Substitution. Dieses Verfahren wurde um 1920 sowohl von amerikanischen als auch deutschen Kryptologen vorgestellt. Es funktioniert nach dem gleichen Prinzip wie die Vigenère Verschlüsselung, nur dass nicht ein kurzes Passwort immer wieder verwendet wird, sondern dass das Passwort genauso lang sein muss wie der Text. Des Weiteren muss der Schlüssel aus komplett zufälligen Zeichenfolgen bestehen, d.h. er darf in sich keine Regelmäßigkeiten aufweisen. Eine letzte, aber auch essentielle Bedingung für die Sicherheit ist, dass jeder Schlüssel nur einmal verwendet wird.

Sind alle drei Bedingungen erfüllt, so ist diese Art der Verschlüsselung, wie Claude Shannon in den 1940 Jahren mathematisch bewiesen hat, unknackbar. Es stellt sich nun also die Frage, warum man überhaupt nach anderen Methoden sucht und andere Verfahren nutzt, wenn es doch schon etwas Sicheres gibt. Die Antwort auf diese Frage liegt schlichtweg in dem Aufwand, den das One-Time-Pad erfordert. Die erste Problematik besteht im Generieren eines zufälligen Schlüssels. Zwar ist es mit Hilfe von Programmen möglich Schlüssel hierfür zu erzeugen, jedoch sind diese nur pseudozufällig³, d.h. dass man ein Muster darin erkennen kann, insofern der Schlüssel nur lang genug ist. Es müsste somit für die Schlüsselgenerierung auf

einen so genannten physikalischen Zufallsgenerator⁴ zurückgegriffen werden, was jedoch einen großen zeitlichen Aufwand bedeuten würde. Das nächste Problem ist die Schlüsselübermittlung. Angenommen man hat es geschafft einen geeigneten Schlüssel zu erstellen, so muss meine Kontaktperson ebenfalls über ihn verfügen. Da ein digitales Versenden immer Risiken mit sich bringt, wäre also eine manuelle Weitergabe von Nöten. Dies wäre nicht weiter problematisch, wenn nicht jeder Schlüssel nur einmal verwendet werden dürfte. Ein drittes ebenfalls schwerwiegendes Problem ist die Begebenheit, dass der Schlüssel gleich lang sein muss wie die zu verschlüsselnden Daten. Würde man jetzt z.B. eine volle Festplatte verschlüsseln wollen, so würde man eine weitere Festplatte benötigen, nur um einen Schlüssel überhaupt speichern zu können. Dies sind die Gründe, warum das OTP kaum Anwendung im zivilen Bereich findet. Ein prominentes Beispiel für die Verwendung des OTP ist das so genannte „rote Telefon“, also die Verbindung des Weißen Hauses mit dem Kreml.

UTF-8 (Universal Character Set Transformation Format)

Da im Zeitalter der Computer nicht mehr manuell verschlüsselt wird und Computer einzig mit binären Zahlen arbeiten können, müssten unsere Texteingaben zunächst umgewandelt werden. Für diese Umwandlung wird in der Praxis meistens UTF-8 verwendet. UTF-8 liegt die Überlegung zugrunde, dass man jedem Zeichen und einigen Operationen eine Zahl zuweist (vgl. Tabelle Nr 1.8). Das „UTF-8 Alphabet“ verfügt über mehrere tausend Zeichen und deckt damit unter anderem das lateinische, arabische und kyrillische Alphabet ab und verfügt des Weiteren über eine große Zahl an Sonderzeichen.

Wenn man z.B. das Wort „hallo“ in Zahlen umwandeln möchte erhält man: 104, 97,108,108,111. Da der Computer jedoch mit Bits⁵, also dem Dualsystem arbeitet müsste man diese Zahlen noch umrechnen und erhält:

$[110100]_2, [1100001]_2, [1101100]_2, [1101100]_2, [1101111]_2$

Ein Auszug aus der UTF-8 Wertetabelle befindet sich in Tabelle 1.8

Abgrenzung zwischen Kryptographie und Steganographie

Zwar haben beide Disziplinen die Absicht Informationen sicher zu übermitteln, jedoch gibt es eine klare Abgrenzung zwischen ihnen. Während die Kryptographie darauf basiert, dass man die Informationen durch Vertauschen, Verrechnen und Substituieren unverständlich macht, handelt es sich bei der Steganographie um das versteckte Übermitteln von Information. Häufige Anwendung fand z.B. die Verwendung eines doppelten Bodens oder das Verwenden von „unsichtbarer Tinte“. Auch im technischen Zeitalter findet die Steganographie weiter ihre Anwendung. So ist es z.B. möglich „hinter“ Bildern oder anderen Dateien anderen Informationen einzubinden, welche für Außenstehende zunächst unsichtbar sind. Maximale Sicherheit für seine Informationen erhält man natürlich durch die Kombination von Krypto- und Steganographie.

Symmetrische und asymmetrische Verschlüsselung

Grundlegend wird zwischen zwei „Verschlüsselungstypen“ unterschieden: Der symmetrischen und der asymmetrischen Verschlüsselung. Die beiden Varianten unterscheiden sich in der Art, wie die Nachrichten und jeweiligen Schlüssel

übermittelt werden. Bei der symmetrischen Verschlüsselung wird der Text mit einem Schlüssel verschlüsselt und dann müssen sowohl der verschlüsselte Text, als auch der Schlüssel selber übertragen werden. Der Empfänger erhält beides und kann die Verschlüsselung somit rückgängig machen und den Inhalt lesen. Hierbei birgt sich jedoch in der Schlüsselübermittlung eine Gefahr. Findet diese ebenfalls über einen unverschlüsselten Kanal statt, so besteht die Möglichkeit, dass er mitgeschnitten wird und somit der verschlüsselte Text gelesen werden kann. Die symmetrische Verschlüsselung wird auch nochmal im Schaubild 1.9 dargestellt. Damit der Schlüssel nicht mitgeschnitten werden kann, wurde die asymmetrische Verschlüsselung entwickelt. Unter diesem Verfahren kann man sich folgendes bildlich vorstellen:

Bob¹ möchte Alice eine Nachricht schicken. Alice möchte auf keinen Fall, dass diese Nachricht von jemand anderem gelesen werden kann. Deshalb fertigt sie einige offene Schlösser (public key) an, die alle nur mit einem Schlüssel (private key) geöffnet werden können. Diese Schlösser macht sie öffentlich zugänglich. Bob hat jetzt die Möglichkeit seine Nachricht verschlüsselt an Alice zu verschicken, indem er die Nachricht in eine Box legt und diese mit dem Schloss verschließt (Verschlüsselung). Nun kann niemand mehr diese Nachricht lesen, solange er den passenden Schlüssel zur Box nicht besitzt. Nur Alice kann dieses Schloss öffnen, da nur sie den passenden Schlüssel zu den Schlössern besitzt (Entschlüsselung). Auch für dieses Verfahren gibt es ein Schaubild unter 1.10.

Aktualität

Im Zeitalter des Internets, der digitalen Kommunikation und der immer weiter greifenden Verknüpfungen der Technik, gewinnt die Kryptographie eine ganz neue Bedeutung. Wie man am Beispiel der Enigma gesehen hat, spielte die Verschlüsselung schon in der Geschichte eine bedeutende Rolle. Jedoch wird heutzutage nicht nur ab und an über Funk kommuniziert, sondern alles läuft digital ab. Beginnend bei Telefonaten, über Kommunikation via Internet, bis hin zur Kreditkartenabrechnung findet alles rein digitalisiert statt. Wie also kann man dafür sorgen, dass die Telefonate privat, das Suchverhalten anonym und die Bankverbindungen sicher bleiben? Diese Frage ist mit nur einem Wort zu beantworten: Kryptographie.

Es werden jegliche Verbindungen verschlüsselt und somit vor dem Zugriff von Außenstehenden geschützt – theoretisch zumindest. In der Praxis jedoch gibt es, wie schon seit jeher, das Bestreben Verschlüsselungen zu brechen, sei es aus Neugier, finanziellen oder kriminellen Gründen. In Anbetracht der momentanen Entwicklung, dass sich das gesamte gesellschaftliche Leben immer weiter ins Digitale verschiebt, bekommt eine sichere Verschlüsselung eine zunehmend größere Bedeutung. Sollen auch weiterhin Privatsphäre und freie Meinungsäußerungen möglich sein, ist es zwingend notwendig, dass uns die Bedeutung der Kryptographie bewusst wird. Das beste Beispiel dafür, wie sehr Staaten mithilfe ihrer Geheimdienste ebendiese grundlegenden Rechte verletzen, ist die NSA-Affäre. Doch anstatt aus den Veröffentlichungen zu lernen, geht vom Großteil der Bevölkerung gerade einmal milde Empörung aus. Auch die von der Politik dann hastig in der „Digitalen Agenda“ angekündigten Verbesserungen auf diesem Gebiet, wurden nicht nur nicht umgesetzt, sondern auch kurz darauf zurückgenommen. So besteht zur Zeit die Überlegung in der europäischen, aber auch in der deutschen

Politik, Verschlüsselung ganz oder zumindest teilweise zu verbieten, um die „Sicherheit“ der Bürger gewährleisten zu können. Einen ähnlichen Ansatz verfolgt die NSA, welche kürzlich ihre Vorstellungen für eine „Vordertür in Verschlüsselungen“ vorstellte. In diesem Ansatz geht es darum, dass für jede verschlüsselte Kommunikation ein Zweitschlüssel bei der NSA vorliegt, mit welcher sie diese im Zweifelsfall wieder entschlüsseln kann.

Somit ist es hinsichtlich der gesellschaftlich strukturellen Veränderungen und der politischen Lage umso bedeutender die Thematik sowohl theoretisch, als auch praktisch ins Bild der Öffentlichkeit zu rücken.

Der Advanced Encryption Standard - AES

Geschichte des AES

AES ist der Advanced Encryption Standard. Er wurde im Jahre 1997 vom NIST (National Institute of Standards and Technology) international in einem Wettbewerb ausgeschrieben. Dies war nötig, da der damals aktuelle Algorithmus DES (Data Encryption Standard) nur eine sehr geringe Schlüssellänge verarbeiten konnte und durch ausprobieren effizienter gebrochen werden kann als durch mathematische Angriffsmethoden. In diesem Wettbewerb wurden weltweit Kryptologen aufgerufen ihre Vorschläge einzureichen, welche bestimmten Anforderungen entsprachen:

- AES sollte ein symmetrischer Algorithmus in Form einer Blockchiffre⁶ sein
- Die Länge der Blöcke wurde auf 128 Bit⁵ festgelegt
- Eine leichte Umsetzung in Hard- oder Software sollte gewährleistet sein
- Der Algorithmus sollte überdurchschnittliche Geschwindigkeitswerte liefern
- Es sollten keine damals bekannten Angriffsmethoden darauf anwendbar sein
- Er sollte geringe Anforderungen an die Hardware stellen, für Einsätze in z.B. Smartcards
- Es sollten keine patentlich geschützten Funktionen genutzt werden und später international für jeden unentgeltlich zur Verfügung stehen

Nach zwei Konferenzen standen von 15 Einreichungen fünf Finalisten fest, die erneut international zur Kryptoanalyse ausgeschrieben wurden.

Da alle bei der Sicherheit zu sehr ähnlichen Ergebnissen führten, hat man sich für den belgischen Algorithmus Rijndael (gesprochen:"raindahl") entschieden. Dieser wurde nach seinen Entwicklern Daemen und Rijmen benannt. Der Hauptgrund für diesen Algorithmus war die geringe Komplexität und herausragenden Geschwindigkeiten auf allen getesteten Plattformen.

Nach einer erneuten Phase der Kryptoanalyse, welche zu keinen Beanstandungen führte, wurde Rijndael schließlich im Jahre 2000 durch das NIST zum Advanced Encryption Standard erklärt und ist seit dem für Verschlüsselungen bis zur höchsten Geheimhaltungsstufe in den USA zugelassen.

Diese Art des offenen Wettbewerbs wurde international sehr positiv bewertet und wurde in der Kryptographie mehrfach wiederholt.

Funktionsweise von Rijndael

Der von Rijmen und Daemen entwickelte Algorithmus basiert auf mehreren Unterprozessen, die eine Tabelle aus vier Zeilen und Spalten, Block genannt, mit einem Schlüssel verschlüsseln. Auch beim Schlüssel handelt es sich um einen Block. Die Verschlüsselung findet in mehreren Durchläufen, sogenannten Runden, statt, deren Anzahl abhängig von der Länge des Schlüssels ist. Die Rundenzahl variiert zwischen 10 und 14. Wichtig ist auch, dass Rijndael nicht für jede Runde den selben Schlüssel verwendet, sondern diesen zuvor in einem Verfahren erweitert und daher für jede Runde einen vom Hauptschlüssel abgeleiteten Rundenschlüssel verwendet.

Bedeutung des Schlüssels:

Der Schlüssel ist nicht, wie man erwarten könnte, direkt das Passwort des Nutzers. Schon ein 128 Bit Schlüssel, was nicht mehr den aktuellen US-Sicherheitsanforderungen für die höchste Geheimhaltungsstufe entsprechen würde, erfordert es sich 128 Binärzeichen zu merken. Das sind dann 32 Zeichen aus dem Hexadezimalsystem. Da diese Länge für Passwörter kaum praktikabel ist, wird der Schlüssel mithilfe eines spezifizierten Verfahrens (in diesem Falle SHA3⁹) aus dem Passwort abgeleitet.

Ablauf von Rijndael

- Erweiterung des Schlüssels (Key Schedule)
- Vorrunde
 - Füge den ersten Rundenschlüssel hinzu (AddRoundKey)
- Vollständige Verschlüsselungsrunden (9-13 Wiederholungen)
 - Übersetze in die S-Box (SubBytes)
 - Verschiebe Werte innerhalb der Zeilen (ShiftRows)
 - Verrechne die Zeilen mit einer Matrix (MixColumns)
 - Füge den aktuellen Rundenschlüssel hinzu (AddRoundKey)
- Letzte Runde
 - SubBytes
 - ShiftRows
 - AddRoundKey

Für den Ablauf befindet sich im Anhang unter 2.1 auch ein Flussdiagramm.

Die einzelnen Funktionen erläutert

Schlüsselerweiterung - KeySchedule

Bei der Schlüsselerweiterung wird für jede auszuführende Runde ein eigener Schlüssel erstellt. Dieser wird aus dem jeweils vorherigen abgeleitet. Der eigentliche Schlüssel kommt somit nur ein einziges mal im gesamten Algorithmus zum Einsatz.

Der Ablauf ist im Flussdiagramm 2.2 auch noch einmal grafisch dargestellt.

Zunächst wird der Schlüssel als Block betrachtet. Um nun die nächste Spalte zu ermitteln wird zunächst die vorherige Spalte gewählt und der oberste Wert nach ganz unten geschrieben. Die anderen Werte rücken anschließend nach. Darauf erfolgt der Einsatz der S-Box (siehe hierfür auch SubBytes), welche die Zeichen durch andere ersetzt. Abschließend wird diese Spalte mit der Spalte vier weiter links und einer Rundenkonstante (siehe 2.3 bis 2.5) mit einem "Exklusiven Oder" (XOR) verknüpft.

Eine XOR Verknüpfung findet auf binärer Ebene statt und ähnelt der schriftlichen Addition. Allerdings wird hier der Übertrag nicht beachtet. Eine tabellarische Darstellung findet sich in 2.6.

Dieser umfangreiche Ablauf findet nur alle vier Durchgänge statt. Es handelt sich hier jeweils um die erste Spalte eines Rundenschlüssels. Die drei folgenden Schlüsselerweiterungen sind weniger umfangreich. Hier wird nur die vorherige Spalte und die Spalte vier vorher in einer XOR Verknüpfung zusammengeführt.

Wie oft dieser Prozess durchläuft ist abhängig von der Länge des Schlüssels und damit der Anzahl der Runden. Für jede Runde wird ein Schlüssel benötigt.

Die Addition des Rundenschlüssels - AddRoundKey

In diesem Schritt wird der zu verschlüsselnde Block mit dem aktuellen Rundenschlüssel, welcher ebenfalls als ein Block betrachtet wird, XOR verknüpft. Dies bedeutet, dass der erste Wert aus der ersten Spalte des Blocks mit dem ersten Wert aus der ersten Spalte des Keys verknüpft wird. Der zweite dann mit dem zweiten usw. Nur in diesem Schritt kommt der Schlüssel des Nutzers zum Einsatz.

Dies ist in Abbildung 2.7 nochmal verdeutlicht.

Die Übersetzung in die S-Box - SubBytes

Ein jeder Wert im Block wird durch einen zugehörigen Wert aus der S-Box ersetzt. Bei diesem Arbeitsschritt handelt es sich um eine Form der monoalphabetischen Verschlüsselung. Die S-Box ist öffentlich und für Rijndael spezifiziert.

Auch für diesen Arbeitsschritt findet sich eine Abbildung unter 2.8

Die S-Box:

Bei der S-Box handelt es sich um eine Tabelle, wo jedem Paar von Hexadezimalwerten ein anderes Paar zugewiesen wird. Sie ist eine Grundkomponente symmetrischer Verschlüsselungsverfahren. In jedem Algorithmus kommen eigene, zum Teil mehrere, S-Boxen zum Einsatz. Der Vorgänger von AES, der Data Encryption Standard, benötigte acht S-Boxen. Die von Rijndael benötigte S-Box wurde mithilfe von Matrixmultiplikationen über einem bestimmten endlichen Körper ermittelt und anschließend anhand bestimmter Sicherheitsanforderungen getestet.

Die Verschiebung von Werten innerhalb der Zeilen - ShiftRows

Bei ShiftRows werden die einzelnen Werte innerhalb einer Zeile nach hinten verschoben. Die anderen Werte rücken dementsprechend nach. Die Anzahl der Verschiebungen hängt von der Zeile und von der Schlüssellänge ab.

Die Anzahl der Verschiebungen befindet sich in Tabelle 2.9 und ein Beispiel findet sich in Abbildung 2.10

Die Vermischung der Spalten - MixColumns

Bei der MixColumns Operation werden die Werte der Spalten mit einer gegebenen Matrix multipliziert. Diese enthält die Ziffern 1, 2 und 3. Da wir uns bei dieser Rechenoperation allerdings in einem bestimmten Körper befinden, einen Galois Körper, in dem jede Zahl als ein Polynom bis zu 27 dargestellt wird, gelten hier andere Rechenregeln.

Eine Binäre Zahl wie zum Beispiel 10101010 wäre im von Rijndael verwendeten $GF(2^8)$:

$$1 \cdot x^7 + 0 \cdot x^6 + 1 \cdot x^5 + 0 \cdot x^4 + 1 \cdot x^3 + 1 \cdot x^2 + 1 \cdot x^1 + 0 \cdot x^0$$

Zusätzlich wird das Ergebnis dieses Polynoms modulo⁷ mit $x^8 + x^4 + x^3 + x + 1$ gerechnet.

Dies ist in Abbildung 2.11 verdeutlicht.

Kryptoanalyse von AES

Wie eingangs erwähnt, wurde bereits beim Auswahlverfahren Rijndael international zur Kryptoanalyse ausgeschrieben und war gegen damals bekannte Verfahren abgesichert. Durch die Möglichkeit sehr lange Schlüssel verwenden zu können ist es auch durch ausprobieren von Schlüsseln nicht effizient möglich Rijndael zu brechen (Brute-Force-Attack).

Erst im Jahr 2001 ist es den Forschern Niels Ferguson, Richard Schroeppel und Doug Whiting gelungen einen theoretischen Angriff auf Rijndael zu entwerfen. Dies gelang ihnen dadurch, dass sie den gesamten Algorithmus in eine Formel schrieben, welche aus mehreren verketteten Brüchen besteht, wie in 2.12 als Auszug dargestellt. Eine praktische Relevanz hat diese Erkenntnis jedoch bislang nicht.

Ein weiterer Angriff ist 2005 von Daniel J. Bernstein vorgestellt worden. Diesem gelang es mithilfe von bekannten Texten, die er verschlüsselte und dem Messen der Zeiten für die Verschlüsselung, den Schlüssel zu ermitteln. Dies liegt daran, dass die Umwandlungen, wie die S-Box und die MixColumns Tabellen, zur Optimierung in Tabellen gespeichert sind und daher über die vergangene Zeit Rückschlüsse auf den verwendeten Schlüssel möglich sind. Ob dieses Szenario in der Praxis funktioniert ist jedoch äußerst umstritten.

Im Jahr 2009 stellten Alex Biryukov und Dmitry Khovratovich einen weiteren theoretischen Angriff auf Rijndael vor, der die benötigte Menge an auszuprobierenden Schlüsseln reduzierte. Bei der Anwendung eines Schlüssels von 256 Bit wäre der Schlüsselpool 2^{256} . Durch diese Angriffsmethode lässt er sich auf 2^{119} und in einer späteren Veröffentlichung auf $2^{99,5}$ reduzieren. Dies bedeutet, dass Rijndael nun in der Theorie gebrochen ist. Allerdings liegt ein durchprobieren von $2^{99,5}$ möglichen Kombinationen noch immer weit jenseits eines positiven Kosten-Nutzen-Verhältnisses für einen Angreifer, was das Ziel jeder Sicherheitsmaßnahme ist. Dies sagte auch Bruce Schneier, als er diesen Angriff veröffentlichte:

"While this attack is better than brute force -- and some cryptographers will describe the algorithm as "broken" because of it -- it is still far, far beyond our capabilities of computation. The attack is, and probably forever will be, theoretical. But remember: attacks always get better, they never get worse. Others will continue to improve on these numbers. While there's no reason to panic, no reason to stop using AES, no reason to insist that NIST choose another encryption standard[...]"¹

¹https://www.schneier.com/blog/archives/2009/07/new_attack_on_a.html
(Zugriffsdatum: 24.04.2015)

Betriebsmodi von Blockchiffren

Betriebsmodi dienen dazu Texte, die über die Länge eines Blockes hinausgehen, verschlüsseln zu können. Hierfür werden verschiedene Verfahren angewandt, welche allesamt den Text auf geeignete Längen erweitern (padding) und zum Teil noch einen Initialisierungsvektor verwenden, um eine weitere Zufallskomponente im Verfahren zu haben.

CBC - Cipher⁸ Block Chaining Mode

Beispielhaft wird nun noch der Cipher Block Chaining Mode erklärt:

Beim auch als CBC abgekürzten Cipher Block Chaining Mode wird jeder Teil der Nachricht in Achtersätze unterteilt und mit dem vorherigen Satz XOR verknüpft. Dafür wird eingangs ein Initialisierungsvektor verwendet.

Dieses Verfahren bietet den Vorteil, dass jeder Satz vom vorherigen abhängig ist und daher nicht einfach ausgetauscht werden kann, da alle anschließenden Sätze damit nicht mehr sinnvoll zu entschlüsseln wären.

Das RSA-Kryptosystem - Asymmetrische Verschlüsselung

In den 70er Jahren entwickelten Ralf Merkle und Martin Hellmann das erste asymmetrische Verschlüsselungsverfahren. Dieses System wies für Adi Shamir, der schon mit zwei weiteren Kollegen, Ronald Rivest und Leonard Adleman, an einem noch sichereren System arbeiteten, Angriffsflächen auf. 1977 publizierten Rivest, Shamir und Adleman ihr neues asymmetrisches Verschlüsselungssystem, bei dem sie keinerlei Angriffspunkte fanden. Aus den Anfangsbuchstaben ihrer Familiennamen abgeleitet nannten sie dieses RSA.

1983 schaffte es Adi Shamir das Verfahren von Ralf Merkle und Martin Hellmann zu knacken.

Das RSA System ist ein asymmetrisches kryptographisches Verfahren, das sowohl zur Ver- und Entschlüsselung als auch zur digitalen Signatur verwendet werden kann.

Generierung eines Schlüsselpaars:

1. Man wählt zwei große, zufällige Primzahlen p & q mit der Bedingung: $0,1 < |\log_2(p) - \log_2(q)| < 30$
2. Das RSA-Modul n berechnet sich folgendermaßen:
 $n = p \cdot q$
3. Berechnung der eulerschen ϕ -Funktion von n :
 $\phi(N) = (p-1) \cdot (q-1)$
4. Wähle für e eine Zahl mit der Bedingung: $1 < e < N$ und der größte gemeinsame Teiler mit $\phi(N)$ gleich eins.
 $\text{ggT}(e, \phi(N)) = 1$
5. $e \cdot d + \phi(N) \cdot k = 1 = \text{ggT}(e, N)$
Mit Hilfe des erweiterten euklidischen Algorithmus' lassen sich nun die Faktoren d und k ermitteln.

Euklidischer Algorithmus

Der euklidische Algorithmus dient zur Ermittlung des größten gemeinsamen Teilers(ggT) zweier Zahlen.

Er besagt:

Wenn $b = 0$ dann ist der $\text{ggT}(a,b) = |a|$

und wenn $b \neq 0$ dann ist der $\text{ggT}(a,b) = \text{ggT}(|b|, a \bmod |b|)$

Der erweiterte euklidische Algorithmus ermittelt zwei Koeffizienten für die gilt:

$$\text{ggT}(a, b) = a \cdot x + b \cdot y$$

und gibt diese zurück.

Auf RSA übertragen bedeutet dies, dass d dem x entspricht.

Die Werte p , q , k und $\phi(N)$ sind für RSA nicht mehr relevant und werden nicht mehr benötigt. Man erhält also N , e und d . Daraus bilden dann e und n den öffentlichen

Schlüssel sowie d und n den privaten Schlüssel. Das wirklich geheime an diesem Algorithmus sind also ausschließlich d , p und q .

Verschlüsselung und Entschlüsselung:

Für die Verschlüsselung einer Nachricht wird folgende, im Vergleich zu anderen Verschlüsselungsverfahren, simple Formel:

$$c = m^e \bmod n$$

Hierbei ist m der zu verschlüsselnde Text und c der geheime Text.

Die Entschlüsselung des geheimen Textes c findet mit einer ähnlich kurzen Formel statt:

$$m = c^d \bmod n$$

Wie man sieht unterscheiden die Formeln sich lediglich im Exponenten. Beide sind aber nur dem Besitzer der privaten Schlüssels bekannt.

Signieren von Nachrichten:

Das Signaturverfahren von RSA ist der Verschlüsselung sehr ähnlich. Allerdings ist das Ziel dieses Verfahrens, dass ein Empfänger mit dem öffentlichen Schlüssel des Absenders, in der Lage ist herauszufinden, ob diese Nachricht tatsächlich vom Absender stammt.

Dafür generiert der Absender für die Nachricht m einen Hashwert¹⁰ $H(m)$ und führt mit diesem folgendes durch:

$$s = H(m)^d \bmod n$$

Ein Empfänger kann nun den Hashwert der Nachricht ermitteln, indem er rechnet:

$$H(m) = s^e \bmod n$$

Wenn der Empfänger nun selber auch den Hashwert der erhaltenen Nachricht ermittelt und dieser mit dem errechneten Wert übereinstimmt, so kann er mit Sicherheit sagen, dass der Absender im Besitz des Gegenschlüssels zu dem öffentlichen Schlüssel ist.

Kryptoanalyse

Die Vorteile des RSA-Kryptosystems sind:

- Eine hohe Sicherheit, aufgrund der einfachen Formeln
- Das Schlüsselverteilungsproblem entfällt, da keine geheime Schlüsselübergabe nötig ist
- Die privaten Schlüssel müssen nicht zentral gespeichert werden, so dass die Gefahr des Missbrauches eingegrenzt wird
- Wenige Schlüssel werden gebraucht, da nicht für jede Kommunikation ein neues Passwort gewählt werden muss und eine, in irgendeiner Form, gesicherte Schlüsselübergabe entfällt.
- Überprüfung durch elektronische Unterschrift (digitale Signatur)

- Ein einfaches “zurückrechnen” ist aufgrund des Potenzierens nicht möglich, da zum einen als schwere Rechenoperationen gelten, zum Anderen, da durch die Modulooperation wird dieser Rechenaufwand nochmals deutlich erhöht → Diskreter-Logarithmus-Problem

Die Nachteile des RSA Kryptosystems sind:

- RSA arbeitet sehr langsam, weil die Schlüssel für die Ver- und Entschlüsselung wesentlich länger sind. Im Vergleich zu AES arbeitet RSA um den Faktor 100 langsamer.
- Darüber hinaus ist eine Anfälligkeit gegenüber Implementierungsfehlern gegeben.
- Um eine hohe Sicherheit zu gewährleisten wird eine große Schlüssellänge benötigt.
- Es muss gesichert werden, dass der öffentliche Schlüssel wirklich von der Kontaktperson stammt. Hier sind bereits erste Missbräuche aufgetreten

Auch dieses asymmetrische Verfahren ist nur so lange sicher, bis es möglich ist in realistischer Zeit zu faktorisieren, also p und q aus n zu errechnen, oder das Diskreter-Logarithmus Problem zu lösen.

Elgamal mit ECC

Allgemein zu Elgamal

Elgamal ist ein asymmetrisches Verschlüsselungsverfahren, welches sowohl mit normalen Gruppen als auch mit elliptischen Kurven betrieben werden kann.

In normalen Gruppen gelten die Regeln der klassischen Mathematik wohingegen bei der Rechnung mit elliptischen Kurven eigene Rechenregeln gelten, da dort die Operatoren umdefiniert wurden. Im Allgemeinen wird Elgamal mit ECC, also Elliptic Curve Cryptography, betrieben.

Historisch zu Elgamal

Elgamal wurde 1985 vom Kryptologen Taher Elgamal basierend auf dem Diffie-Hellman-Schlüsselaustausch entwickelt. Der Schlüsselaustausch selbst wurde 1976 von Martin Hellman, Whitfield Diffie und Ralph Merkle entworfen.

ECC - Elliptic Curve Cryptography

Elliptic Curve Cryptography ist ein Bereich der Kryptographie, in dem die mathematischen Besonderheiten elliptischer Kurven genutzt werden, um Verschlüsselungssysteme aufzubauen, die dadurch auffallen, dass sie auch mit vergleichsweise kurzen Schlüsseln sicher sind. Die erwähnten mathematischen Besonderheiten elliptischer Kurven sind die Definitionen der Operatoren. Diese sind geometrisch immer gleich definiert, jedoch mathematisch unterschiedlich je nach Kurventyp.

Besonders bekannt sind folgende Kurventypen:

Die Weierstraß-Kurven:

Gleichung: $y^2 = x^3 + ax^2 + b$

Bedingung: $4a^3 + 27b^2 \neq 0$; a, b sind konstant

Ein Bild einer Weierstraß Kurve findet sich im Anhang unter 3.1

Die Edwards-Kurven:

Gleichung: $x^2 + y^2 = c^2(1 + dy^2x^2)$

Bedingung: $cd(1 - c^4d) \neq 0$; c, d sind konstant

Die Montgomery-Kurven:

Gleichung: $by^2 = x^3 + ax^2 + x$

Bedingung: $b(a^2 - 4) \neq 0$; a, b sind konstant

Es wird vermutet, dass es der NSA gelungen ist in annehmbarer Zeit Implementierungen, die auf Weierstraß-Kurven zurückgreifen, unabhängig davon welche Parameter a, b benutzt werden, zu knacken. Daher werden zunehmend Edwards- und Montgomery-Kurven genutzt.

Rechenregeln auf elliptischen Kurven

Die Addition auf elliptischen Kurven

Auf elliptischen Kurven ist die Addition wie folgt. P und Q seien zwei Punkte auf der Kurve M . Es gilt: $P \neq Q$, $P \neq (-Q)$. Unter diesen Bedingungen ist die Summe $P + Q$ der negative Schnittpunkt der Schnittgeraden durch P und Q und der Kurve M . Ein Bild zu diesem Verfahren findet sich im Anhang unter 3.1.

Wenn die Y -Koordinate von P der negativen Y -Koordinate von Q entspricht und die X -Koordinaten identisch sind, gilt $P = (-Q)$, folglich gilt in dem Fall auch $P + Q = P + (-P) = 0$, oder als Punkt $(0|0)$.

Die mathematische Umsetzung der Punktaddition ist abhängig davon, auf was für einem Kurventypen man addieren will. Systematisch betrachtet ist dabei eine logische Vorgehensweise erst die Steigung der Schnittgeraden zu ermitteln und daraufhin durch einsetzen eines Punktes den Y -Achsenabschnitt der Schnittgeraden zu errechnen. Wenn man die Gleichung der Schnittgeraden kennt, ermittelt man den Schnittpunkt mit der Kurve und rechnet die Y -Koordinate mal (-1) .

Verdopplung eines Punktes

Die Verdopplung eines Punktes läuft ähnlich wie die Addition ab. Der einzige Unterschied ist, dass die Rolle der Schnittgeraden, die bei einer Verdopplung nicht existiert, von der Tangente an dem zu verdoppelten Punkt eingenommen wird.

Multiplikation eines Punktes mit einem Skalar

Die Multiplikation an sich ist nicht unabhängig auf elliptischen Kurven definiert. Folglich verwendet man teilweise das „doubling and add“-Verfahren. Dieses basiert darauf, dass jede Multiplikation mit einem ganzzahligen Skalar sich in eine Kette aus Verdopplungen und Additionen der Ergebnisse der Verdopplung mit dem ursprünglichem Punkt umsetzen lässt.

Spezialfälle sind die Multiplikationen mit 1 und 0. Bei der Multiplikation mit 0 ist das Ergebnis immer $(0|0)$ und bei der Multiplikation mit 1 ist das Ergebnis P , also der zu multiplizierende Punkt.

Elgamal mit ECC

Die Schlüsselgenerierung

Elgamal ist ein asymmetrisches Verschlüsselungssystem, folglich werden private - als auch public-key benötigt.

Der private-key ist eine einfache natürliche Zahl, welche ein Gruppenelement der gewählten Gruppe ist. Der public-key ist ein Erzeugerpunkt multipliziert mit dem private-key.

Beispiel:

Die Kurve: $2y^2 = x^3 - x^2 + x$

$x = 5$ (private-key)

$P = (2|3^{0,5})$ (Erzeugerpunkt)

$Y = xP = (3.757484035|-4.620039821)$ (public-key)

Auch wenn das womöglich relativ leicht und unsicher erscheint, ist dies nicht der Fall. Da keine Division auf elliptischen Kurven vorhanden ist, ist es nicht möglich den private-key zu errechnen. Eigentlich ist nur die Brute-Force-Methode möglich, die jedoch bei größeren Zahlen Jahrhunderte dauert.

Neben dem öffentlichen Schlüssel muss auch bekannt gemacht werden, welcher Punkt der Erzeugerpunkt ist, welche Kurve genutzt wird und welche große Primzahl die Gruppe definiert.

Die Verschlüsselung

p ist eine sehr große Primzahl

G ist die Gruppe, also die Menge aller rationalen Punkte der Kurve, die mit Hilfe der Primzahl p definiert ist.

k ist eine zufällige Zahl ; $0 < k < p$

m ist der zu verschlüsselnde Text, als Zahl dargestellt, z.B. durch UTF-8.

P ist der Erzeugerpunkt der Gruppe G .

Y ist der gegebene öffentliche Schlüssel.

C ist das Gruppenelement von k , also $C = k * P$ (multipliziert auf der Kurve)

Q_x ist die X-Koordinate von $Q = k * Y$ (multipliziert auf der Kurve)

$d = Q_x * m \bmod p$

Der Punkt C und die Zahl d ergeben zusammen den Cipher.

Die Entschlüsselung

$c1_x$ ist die X-Koordinate von $c1 = x * C$ (multipliziert auf der Kurve)

$m = d / c1_x \bmod p = (k * P * x)_x * m / (k * P * x)_x$

Der Index x bedeutet, dass vom Ergebnis die X-Koordinate gemeint ist.

Kryptoanalyse

An sich ist Elgamal in Durchführung mit ECC bei einer Schlüssellänge von 512 Bit utopisch sicher, weswegen häufig Schlüssel mit einer Länge von 256 Bit oder sogar nur 128 Bit verwendet werden. Teilweise wird geschätzt, dass Elgamal mit ECC und einem 512 Bit Schlüssel einer 15.000 Bit Verschlüsselung mit RSA, betreffend der Sicherheit, entspricht. Eine Grafik im Anhang unter 3.2 stellt die Zusammenhänge der Schlüssellängen nochmal dar.

Unabhängig von der Schlüssellänge können aber fatale Fehler bei der Implementierung geschehen:

1. k ist zu klein, beziehungsweise wird in einem zu kleinen Bereich gewählt: Wenn k in einer Implementierung zum Beispiel maximal den Wert 100 annehmen kann, ist die Verschlüsselung absolut unsicher. Über die Brute-Force-Methode hat man k schnell ermittelt und damit die Verschlüsselung geknackt.
Auch sollte k immer nur einmal verwendet werden. Denn bei einer Known-

Plaintext-Attack ist k ermittelbar und damit zukünftige Verschlüsselungen, wenn sie auf den selben Wert für k zugreifen unsicher.

2. P ist zu klein:

Wenn der Erzeugerpunkt zu klein ist, ist es möglich, dass er sehr kleine Untergruppen auf der Kurve bildet. Das kann soweit gehen das Chiphtrat und Nachricht übereinstimmen. Selbst wenn dies nicht passiert, kann bei einem zu kleinem P durch Ausprobieren die Nachricht entschlüsselt werden.

3. Die Zeit ist nicht geregelt:

Wenn die Zeit für die Verschlüsselung unreguliert ist, ist ein Bereich für k schätzbar, wenn nicht sogar berechenbar. Dann ist durch Einsetzen der Werte des Bereiches nach akzeptabler Zeit die Verschlüsselung geknackt.

Elgamal wird als schwer lösbar betrachtet, wenn die genannten Implementierungsfehler nicht gemacht wurden und das Diskreter-Logarithmus-Problem sich nicht effizient lösen lässt. Ist dies gegeben, ist Elgamal sogar IND-CPA-sicher, was bedeutet, dass ein Angreifer der zwei Klartexte und zwei Ciphertexte hat, nicht bestimmen kann, welcher Klartext und welcher Cipher zusammen gehören oder ob er überhaupt ein Klartext-Cipher-Pärchen vorliegen hat. Der Vorteil bei Elgamal mit ECC ist, dass eine geringe Schlüsselvergrößerung, wie zum Beispiel um 20 Stellen, schon einen enormen Unterschied in der Sicherheit macht. Wenn also irgendwann Elgamal mit ECC und einem 256 Bit-Schlüssel als unsicher betrachtet wird, ist dennoch enormer Aufwand im Vergleich nötig, um Elgamal mit ECC und einem 276 Bit-Schlüssel zu knacken. Daher sehen viele Kryptographen in ECC die Zukunft der Kryptographie, da bei einer guten Implementierung eine hohe Sicherheit bei geringen Rechenaufwand geschaffen wird, die seines gleichen sucht.

Benutzeroberfläche

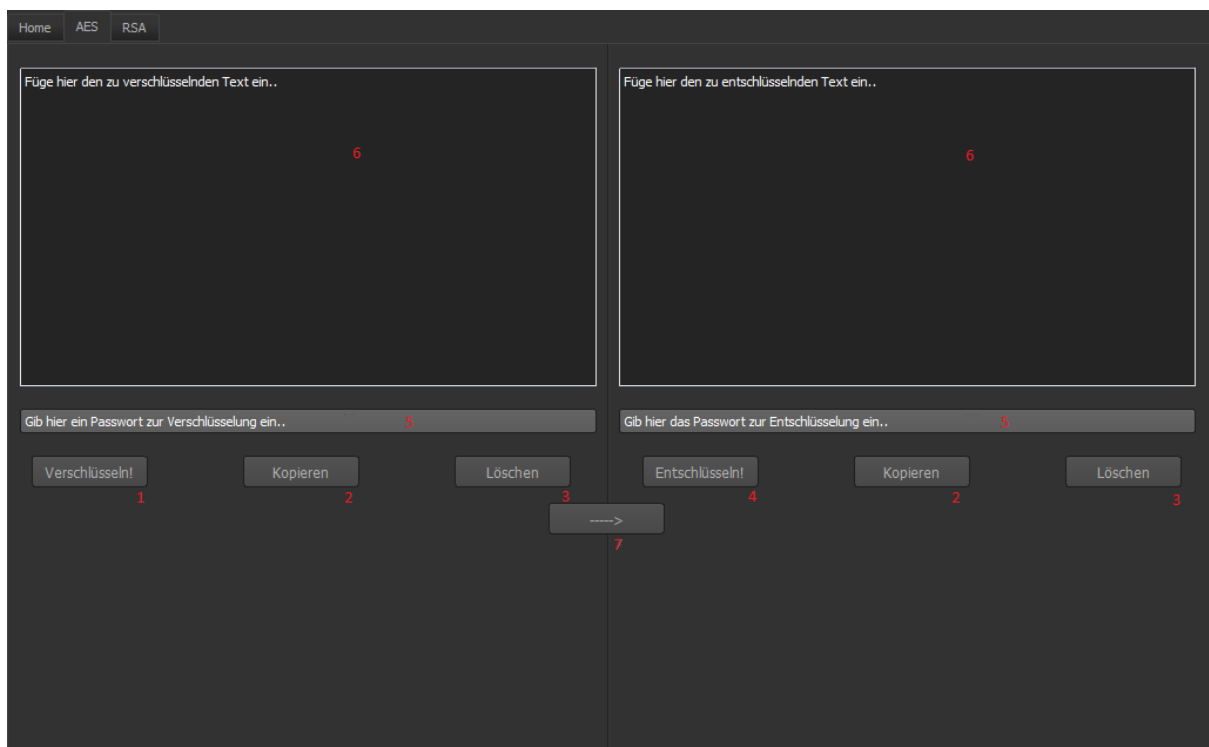
Im folgendem Kapitel wird erläutert, welche Möglichkeiten der Benutzer mit unserem Programm hat. Es dient somit der Funktion eines Handbuchs.

Der Startbildschirm:



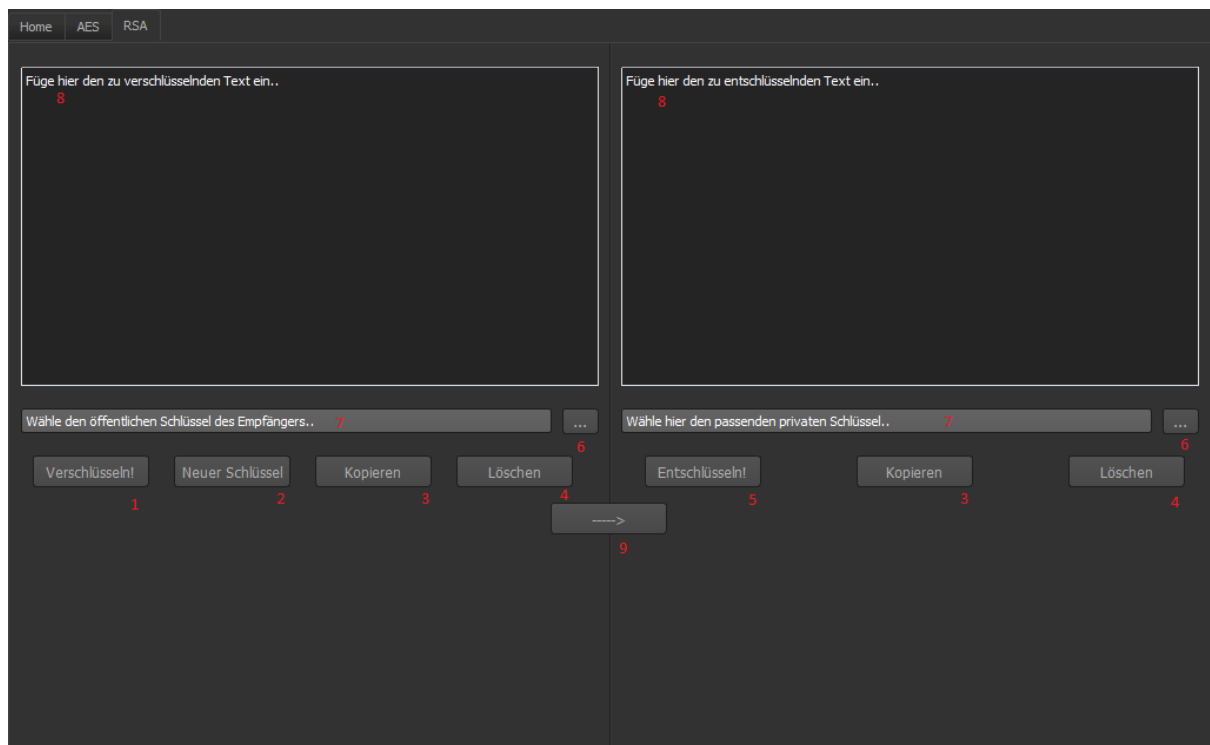
Auf dieser Seite werden dem Nutzer die verschiedenen Möglichkeiten im Umgang mit dem Programm erläutert. Neben einigen generellen Nutzungshinweisen sind hier auch die Autoren noch einmal vermerkt. Oben finden sich drei Reiter, von welchen der Erste zur Verschlüsselung mit dem Advanced Encryption Standard und der Zweite zu RSA führt, welche nun im Folgenden erläutert werden.

AES:



- 1: Startet die Verschlüsselung eines eingegebenen Textes
- 2: Kopiert den eingegebenen oder verschlüsselten Text
- 3: Löscht den geschriebenen Text aus dem Fenster
- 4: Entschlüsselt einen verschlüsselten Text
- 5: Eingabefeld für das Passwort
- 6: Textfeld für die Eingabe des jeweiligen Textes (links unverschlüsselt, rechts verschlüsselt)
- 7: Dieser Button kopiert den verschlüsselten Text und das Passwort zum Entschlüsseln in die rechte Hälfte

RSA:



- 1: Startet die Verschlüsselung eines eingegebenen Textes.
- 2: Generiert einen neuen Schlüssel.
- 3: Kopiert den eingegebenen Text in die Zwischenablage.
- 4: Löscht den Inhalt aus dem Fenster.
- 5: Entschlüsselt einen eingegebenen verschlüsselten Text.
- 6: Öffnet den Dateibrowser zum Suchen und Öffnen von bereits generierten Schlüsseln.
- 7: Eingabefeld für den Dateipfad zum öffentlichen/privaten Schlüssel
- 8: Textfeld für die Eingabe des jeweiligen Textes (links unverschlüsselt, rechts verschlüsselt)
- 9: Dieser Button kopiert den verschlüsselten Text und das Passwort zum Entschlüsseln in die rechte Hälfte

Fazit

Nachdem ein Großteil der Projektzeit vorbei ist, kann man an dieser Stelle ein Fazit ziehen. Dieses soll anhand der zu Beginn genannten drei Hauptkriterien erfolgen. Zunächst zum Erschließen der Algorithmen und der zugrunde liegenden Mathematik: Die drei von uns umgesetzten Verschlüsselungen haben alle unterschiedliche mathematischen Anforderungen. Während RSA auf wenigen Rechnungen basiert, welche alle mithilfe von Oberstufenmathematik mehr oder weniger verständlich sind, gestalten sich AES und Elgamal komplexer.

Die besondere Schwierigkeit von AES, betreffend des Verständnisses, liegt in der Begründung, warum es auf diese Art und Weise am sichersten ist. Die hierbei zugrundeliegende Mathematik reicht weit ins Mathematikstudium hinein und ist uns nicht vollständig verständlich geworden. Zu unserem Glück reicht es zum Implementieren jedoch zu Verstehen wie etwas berechnet wird, ohne dabei vollständig zu verstehen warum dies so erfolgt.

Zuletzt wurde Elgamal geschrieben. Hier ist ein Verständnis der zugrundeliegenden Mathematik notwendig, da die Verschlüsselung stark mit der Mathematik elliptischer Kurven zusammenhängt. Diese weicht jedoch stark von der Schulmathematik ab, konnte jedoch soweit verstanden und auch partiell umgesetzt werden. Der zweite Punkt, das einfache Benutzen, ist auch gelungen. Die Nutzeroberfläche ist durch ihre übersichtliche Gestaltung auch für Laien leicht zu bedienen. Es wird weder zur Ver- noch zur Entschlüsselung Wissen im Bereich der Kryptographie benötigt und durch das relativ einfach gehaltene Design ist eine gute Übersicht gewährleistet.

Ein besonderes Interesse sollte der Sicherheit gegeben werden. Zunächst ist zu sagen, dass die Algorithmen, besonders AES und RSA, fast komplett der Praxis entsprechend umgesetzt wurden und somit theoretisch sicher sind. Jedoch haben wir aus technischen, aber auch zeitlichen Gründen präparierende Schritte nur in gekürzter Version eingebunden. Dies ist zwar nicht Teil der eigentlichen Verschlüsselung, würde jedoch in der praktischen Anwendung die Sicherheit mindern. Die größte "Sicherheitslücke" in unserem Programm befindet sich in der Verknüpfung von den Scripten zum GUI. Die hierfür benötigten Auslagerungsdateien sind zwar gut zu bedienen, jedoch nicht sicher. In der Praxis würden diese keine Anwendung finden, jedoch stellen sie für unsere Anliegen die beste Option dar.

Darüber hinaus ist es leider nicht gelungen, eine Kompatibilität zu real verwendeten Implementierungen herzustellen, da diese nur schlecht dokumentiert waren, gerade hinsichtlich des Umwandelns von Texten und Betriebsmodi bei AES sowie dem Speicherformat und auch wieder der Umwandlung bei RSA.

Ebenfalls ein Sicherheitsrisiko für AES stellt die Tatsache dar, dass nur eine sehr geringe Schlüssellänge unterstützt wird, da der zeitliche Rahmen keine Neuimplementierung davon abhängiger Funktionen ermöglichte, besonders unter dem Aspekt, dass es für diese Varianten keine ausführliche Dokumentation gibt. Des Weiteren wird die Sicherheit des Programms durch eine Anforderung, die wir an es stellen, gemindert:

Es soll bei der asymmetrischen Verschlüsselung schnell auf alle benötigten Schlüssel zugreifen können. Diese werden hierfür zusammen gespeichert, was in der praktischen Anwendung die Sicherheit ad absurdum führen würde.

Angesichts der gegebenen zeitlichen Grenzen, jedoch besonders hinsichtlich der Komplexität dieses Themas, ist die Gruppe mit ihrem Arbeitsergebnis zufrieden, auch wenn dieses noch nicht marktreif, beziehungsweise nicht für ein großes Sicherheitsbedürfnis geeignet, ist. Ein derartig zufriedenstellendes Ergebnis ist jedoch nur aufgrund eines großen Engagements der Gruppenmitglieder und der Bereitschaft sehr viel Zeit zu investieren möglich gewesen.

Glossar

1. Notationen in der Kryptographie:

In einem Paper beschrieben Ronald Rivest, Adi Shamir und Leonard Adleman 1978 erstmals Kommunikationsprozesse mit der, seit dem weit verbreiteten, Notation von Alice und Bob, welche die beiden Hauptgesprächsteilnehmer darstellen und als Ersatz für A und B angedacht wurden. Erweitert wird die Anzahl an Personen meist durch Carol und Dave. In der Notation finden sich noch weitere Instanzen, welche in der Kryptographie vorkommen, wie zum Beispiel die Angreifer Eve und Mallory, welchen unterschiedliche Methoden die Kommunikation zu belauschen zur Verfügung stehen.

2. Substitution:

Unter einer Substitution versteht man das Austauschen eines Wertes mit einem anderen Wert.

3. Pseudozufall:

Mit dem Begriff Pseudozufall bezeichnet man Ereignisse, welche auf den ersten Blick zufällig scheinen, jedoch auf reproduzierbaren Parametern basieren und somit nachberechenbar sind. So handelt es sich zum Beispiel bei einem Programm zur Generierung von Zufallszahlen um einen Pseudozufallsgenerator. Dies hat zur Folge, dass Regelmäßigkeiten in den "Zufallszahlen" auftreten. Je besser ein Generator ist, desto weniger Regelmäßigkeiten treten auf. Verwendet man ein pseudozufälliges Passwort, so ist es einem Angreifer möglich, diese Regelmäßigkeiten auszunutzen und die Verschlüsselung zu brechen. Um diese Problematik zu umgehen greift man bei hohem erforderten Sicherheitsniveau z.B. auf einen physikalischen Zufallsgeneratoren zurück.

4. Physikalische Zufallsgeneratoren:

Der physikalische Zufallsgenerator nutzt Prozesse in der Natur, um Zufallszahlen zu erstellen. Diese Prozesse sind zwar auch deterministisch und somit streng betrachtet auch nur pseudozufällig, jedoch wird auf Ereignisse zurückgegriffen, welche polykausal und somit mit heutigen Mittel nicht erklär- bzw. reproduzierbar sind. Beispiele für solche Verfahren sind das Aufzeichnen von Zerfallsprozessen von radioaktiven Stoffen oder von atmosphärischem Rauschen. Zwar bieten diese Verfahren "sichere Zufallszahlen", jedoch sind all diese Verfahren sehr zeit- und kostenaufwändig. Daher werden sie in der Praxis nur äußerst selten genutzt.

5. **Bit:**

Unter einem Bit versteht man eine Ziffer im Dualsystem, d.h. sie kann den Wert 0 oder 1 annehmen. Bits werden verwendet, um Informationen darzustellen oder weiter zu geben. So kann z.B. in der Elektrotechnik 1 Strom und 0 keinen Strom bedeuten. Werden größere Informationsmengen bearbeitet, greift man auf die Bezeichnung Bytes zurück. Ein Byte entspricht 8 Bit, also acht Stelle, die jeweils 0 oder 1 sein können. Für größere Mengen werden entsprechend KiloByte(KB), MegaByte(MB), GigaByte(GB) und TeraByte(TB) verwendet. Innerhalb dieser Kette, die noch länger fortläuft, handelt es sich bei jeder neuen Einheit um eine Vergrößerung mit den Faktor Tausend.

6. **Blockchiffre:**

Bei Blockchiffren handelt es sich um Verschlüsselungsverfahren, bei denen nicht einzelne Werte, sondern Blöcke verschlüsselt werden. Blöcke lassen sich als Tabellen mit begrenzter Zeilen- und Spaltenzahl betrachten. Die Besonderheit von Blockchiffren ist, dass sie nur einen Block verschlüsseln kann und auch nur ein Chiffrat von der Länge eines Blockes wieder ausgeben kann. Wenn man längere Texte verschlüsseln möchte kommen Betriebsmodi zum Einsatz (Siehe AES → CBC)

7. **Modulo Rechnung:**

Unter dem mathematischen Operator Modulo versteht man die Division mit Rest. D.h. es wird überprüft, wie oft der Divisor vollständig in den Dividend passt. Das Ergebnis der Modulorechnung ist die Differenz zwischen dem Dividend und dem Produkt des x mal hineinpassenden Divisors (x Element der ganzen Zahlen)

Beispielrechnung:

$$17 \bmod 5$$

$$3 \cdot 5 = 15$$

$$\rightarrow 17 \bmod 5 = 17 - 15 = 2$$

Die 5 passt 3 mal vollständig in die 17

8. **Chiper:**

Bei Cipher oder Chiffrat handelt es sich um in der Kryptographie verbreitete Begriffe für verschlüsselte Texte.

9. **SHA3:**

Bei SHA3 handelt es sich um den Standard Hash Algorithm in der dritten Generation. Da die beiden Vorgänger (SHA1 und SHA2) mathematisch gebrochen sind und SHA1 bereits in der Praxis ausgehebelt werden konnte, schrieb das NIST einen Wettbewerb für einen neuen Hashstandard aus, der analog zum Auswahlverfahren von AES ablief.

Der Gewinner dieses Wettbewerbs war der Algorithmus Keccak, welcher unter anderem auch vom "Erfinder" von AES Joan Daemen entwickelt wurde.

10. Hash:

Bei einem Hashwert handelt es sich um einen kryptographischen Fingerabdruck. Dieser Fingerabdruck hat die besondere Eigenschaft, dass er nur zu einer Zeichenfolge passen kann, diese aber nicht aus ihm ermittelt werden kann. Eine weitere Besonderheit ist, dass die Länge des Hashwertes immer gleich ist. Es macht keinen Unterschied, ob man einen Satz oder den Text eines Romans hasht, die Anzahl der Zeichen des Hashwertes ist immer identisch. Sobald man allerdings einen Buchstaben im jeweiligen Text verändert, verändert sich auch der Hashwert.

11. GUI:

Der Begriff GUI steht für Graphical User Interface und bedeutet somit grafische Benutzeroberfläche. Es dient hauptsächlich der vereinfachten Benutzung eines Programms. Anstatt Befehle in die Konsole zu tippen, kann ein Nutzer es so z.B. durch das Verwenden von Knöpfen und Textfeldern verwenden. Desweiteren kann durch eine ansprechend und logisch gestaltetes GUI eine bestimmte Atmosphäre geschaffen werden, in welcher ein Nutzer besonders zügig zu gewünschten Ergebnissen kommt, beispielsweise durch das Positionieren von Aufrufen, für häufig genutzte Funktionen, an zentralen Stellen.

Anhang

1: Allgemein

Tabelle 1.1 Substitution

normal	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
neues	q	w	e	r	t	z	u	i	o	p	a	s	d	f	g	h	j	k	l	m	y	x	c	v	b	n

Beispiel: Der Satz „Verschlüsselung ist cool“ würde somit „xtkleisytltsyfuolmeggs“ heißen.

Tabelle 1.2 Caesar-Verschlüsselung

normal	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
neues	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

Tabelle 1.3 Erweiterte Caesar-Verschlüsselung

normal	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
neues	g	a	r	t	e	n	b	c	d	f	h	i	j	k	l	m	o	p	q	s	u	v	w	x	y	z

Beispiel: Das Beispielpasswort ist „garten“. Die restliche Buchstaben ergeben sich aus einem einfachen Aufrücken.

Tabelle 1.4 Vigenère-Chiffre

	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

Beispiel: Lautete das Passwort z.B. „geheim“ und der zu verschlüsselnde Text „Beispiel“, so wurde für den ersten Buchstaben in der Spalte „B“ und in der Reihe „g“ geguckt. Man erhält also als ersten Buchstaben das „h“. Der zweite zu verschlüsselnde Buchstabe ist das „e“ und der zweite Teil des Schlüssels ist ebenfalls ein „e“. Also sieht man in Reihe „e“ und Spalte „e“ nach und erhält ein „i“. Dies setzt man nach diesem Prinzip fort, bis der gesamte Text verschlüsselt ist.

Tabelle 1.5 Relative Häufigkeit der Buchstaben im Deutschen

Platz	Buchstabe	Relative Häufigkeit
1.	E	17,40 %
2.	N	9,78 %
3.	I	7,55 %
4.	S	7,27 %
5.	R	7,00 %
6.	A	6,51 %
7.	T	6,15 %
8.	D	5,08 %
9.	H	4,76 %
10.	U	4,35 %
11.	L	3,44 %
12.	C	3,06 %
13.	G	3,01 %
14.	M	2,53 %
15.	O	2,51 %
16.	B	1,89 %
17.	W	1,89 %
18.	F	1,66 %
19.	K	1,21 %
20.	Z	1,13 %
21.	P	0,79 %
22.	V	0,67 %
23.	ß	0,31 %
24.	J	0,27 %
25.	Y	0,04 %
26.	X	0,03 %
27.	Q	0,02 %

Bild 1.6 Bild der Enigma



Abbildung 1.7 Schaltschema der Enigma

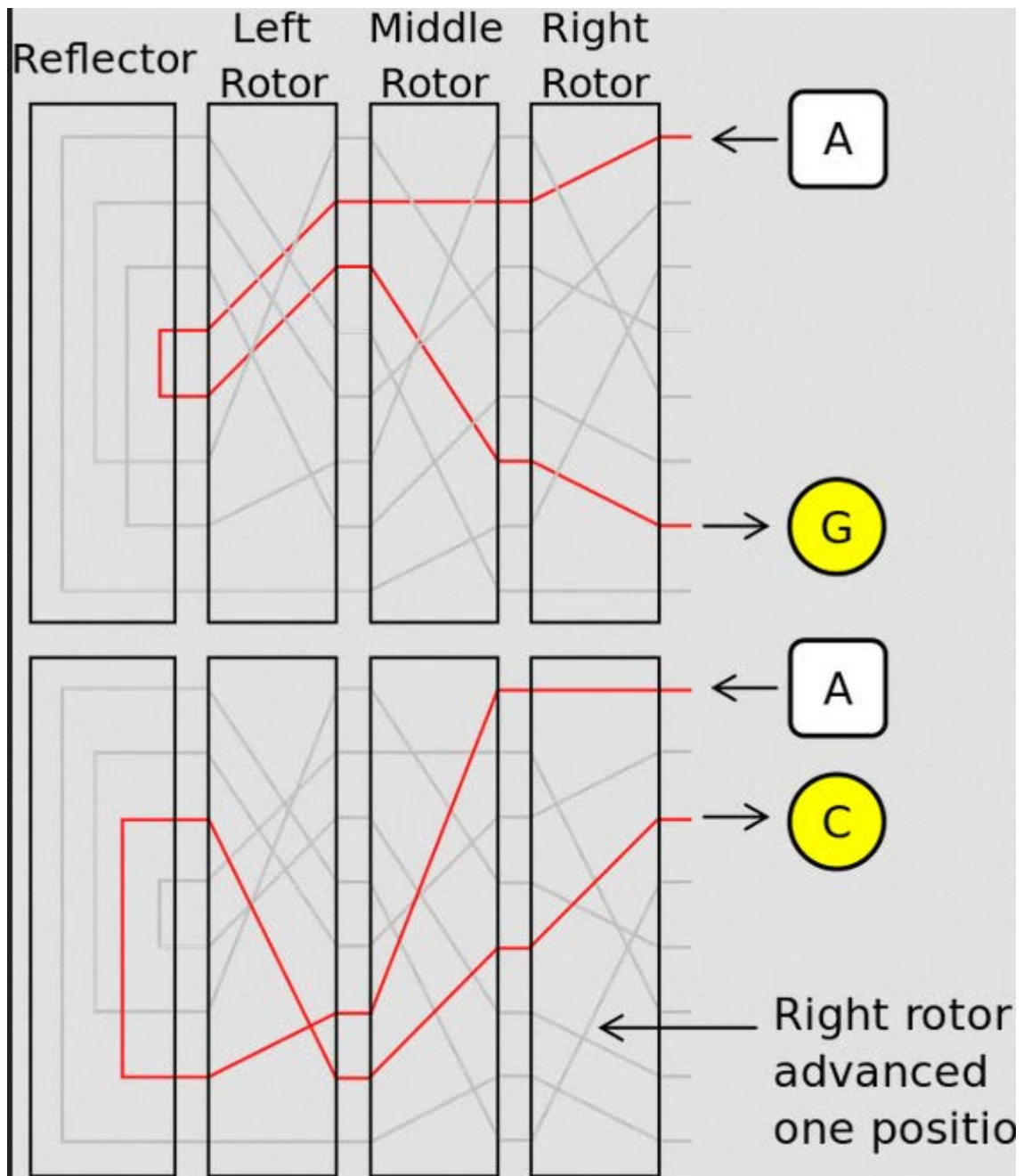


Tabelle 1.8 Auszug aus der UTF-8 Tabelle

Dec	Hx	Oct	Char	Dec	Hx	Oct	Html	Chr	Dec	Hx	Oct	Html	Chr	Dec	Hx	Oct	Html	Chr
0	0	000	NUL (null)	32	20	040	 	Space	64	40	100	@	@	96	60	140	`	`
1	1	001	SOH (start of heading)	33	21	041	!	!	65	41	101	A	A	97	61	141	a	a
2	2	002	STX (start of text)	34	22	042	"	"	66	42	102	B	B	98	62	142	b	b
3	3	003	ETX (end of text)	35	23	043	#	#	67	43	103	C	C	99	63	143	c	c
4	4	004	EOT (end of transmission)	36	24	044	$	\$	68	44	104	D	D	100	64	144	d	d
5	5	005	ENQ (enquiry)	37	25	045	%	%	69	45	105	E	E	101	65	145	e	e
6	6	006	ACK (acknowledge)	38	26	046	&	&	70	46	106	F	F	102	66	146	f	f
7	7	007	BEL (bell)	39	27	047	'	'	71	47	107	G	G	103	67	147	g	g
8	8	010	BS (backspace)	40	28	050	((72	48	110	H	H	104	68	150	h	h
9	9	011	TAB (horizontal tab)	41	29	051))	73	49	111	I	I	105	69	151	i	i
10	A	012	LF (NL line feed, new line)	42	2A	052	*	*	74	4A	112	J	J	106	6A	152	j	j
11	B	013	VT (vertical tab)	43	2B	053	+	+	75	4B	113	K	K	107	6B	153	k	k
12	C	014	FF (NP form feed, new page)	44	2C	054	,	,	76	4C	114	L	L	108	6C	154	l	l
13	D	015	CR (carriage return)	45	2D	055	-	-	77	4D	115	M	M	109	6D	155	m	m
14	E	016	SO (shift out)	46	2E	056	.	.	78	4E	116	N	N	110	6E	156	n	n
15	F	017	SI (shift in)	47	2F	057	/	/	79	4F	117	O	O	111	6F	157	o	o
16	10	020	DLE (data link escape)	48	30	060	0	0	80	50	120	P	P	112	70	160	p	p
17	11	021	DC1 (device control 1)	49	31	061	1	1	81	51	121	Q	Q	113	71	161	q	q
18	12	022	DC2 (device control 2)	50	32	062	2	2	82	52	122	R	R	114	72	162	r	r
19	13	023	DC3 (device control 3)	51	33	063	3	3	83	53	123	S	S	115	73	163	s	s
20	14	024	DC4 (device control 4)	52	34	064	4	4	84	54	124	T	T	116	74	164	t	t
21	15	025	NAK (negative acknowledge)	53	35	065	5	5	85	55	125	U	U	117	75	165	u	u
22	16	026	SYN (synchronous idle)	54	36	066	6	6	86	56	126	V	V	118	76	166	v	v
23	17	027	ETB (end of trans. block)	55	37	067	7	7	87	57	127	W	W	119	77	167	w	w
24	18	030	CAN (cancel)	56	38	070	8	8	88	58	130	X	X	120	78	170	x	x
25	19	031	EM (end of medium)	57	39	071	9	9	89	59	131	Y	Y	121	79	171	y	y
26	1A	032	SUB (substitute)	58	3A	072	:	:	90	5A	132	Z	Z	122	7A	172	z	z
27	1B	033	ESC (escape)	59	3B	073	;	;	91	5B	133	[[123	7B	173	{	{
28	1C	034	FS (file separator)	60	3C	074	<	<	92	5C	134	\	\	124	7C	174	|	
29	1D	035	GS (group separator)	61	3D	075	=	=	93	5D	135]]	125	7D	175	}	}
30	1E	036	RS (record separator)	62	3E	076	>	>	94	5E	136	^	^	126	7E	176	~	~
31	1F	037	US (unit separator)	63	3F	077	?	?	95	5F	137	_	_	127	7F	177		DEL

Schaubild 1.9 Symmetrische Verschlüsselung

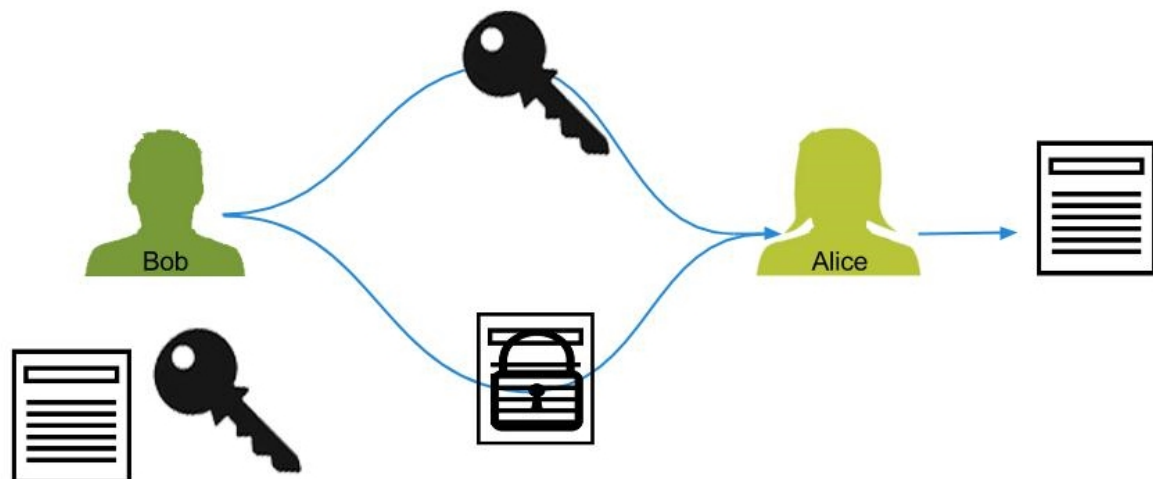
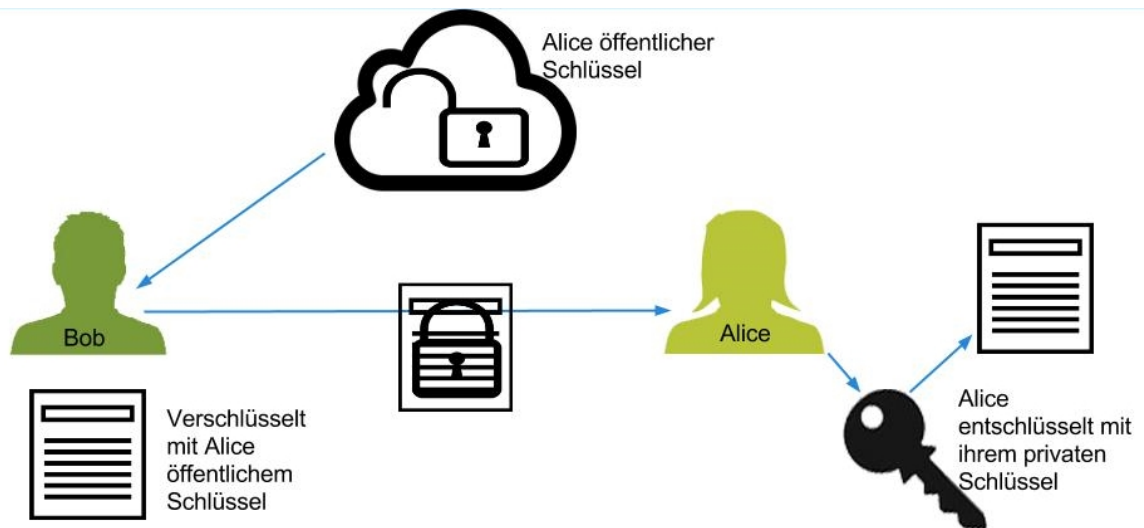


Schaubild 1.10 Asymmetrische Verschlüsselung



2: AES

Bild 2.1: Flussdiagramm zum Ablauf des AES

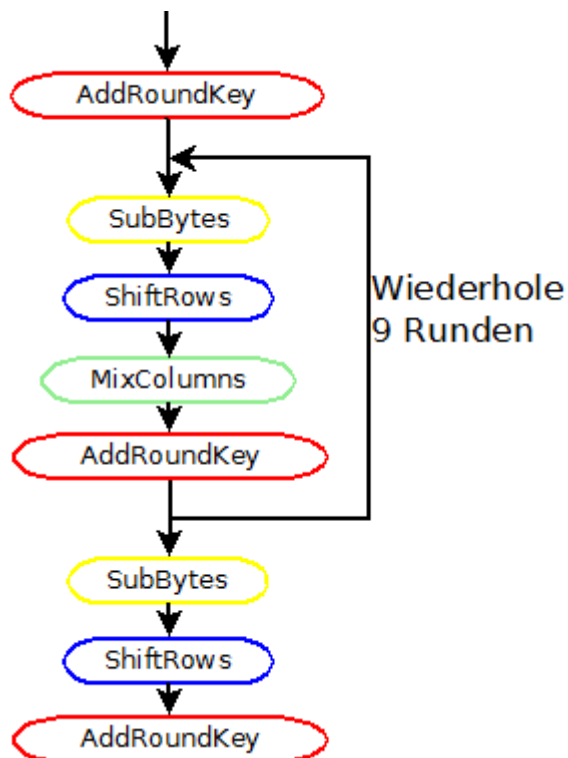


Bild 2.2: Flussdiagramm zu KeySchedule

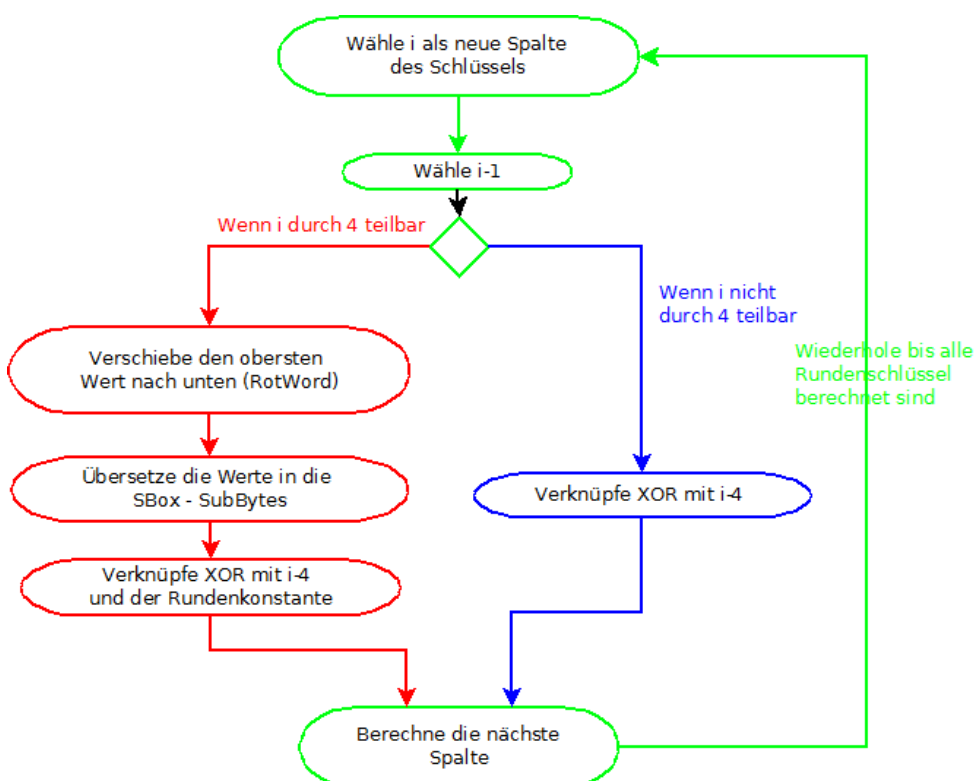


Bild 2.3: Die umfangreiche KeySchedule Operation

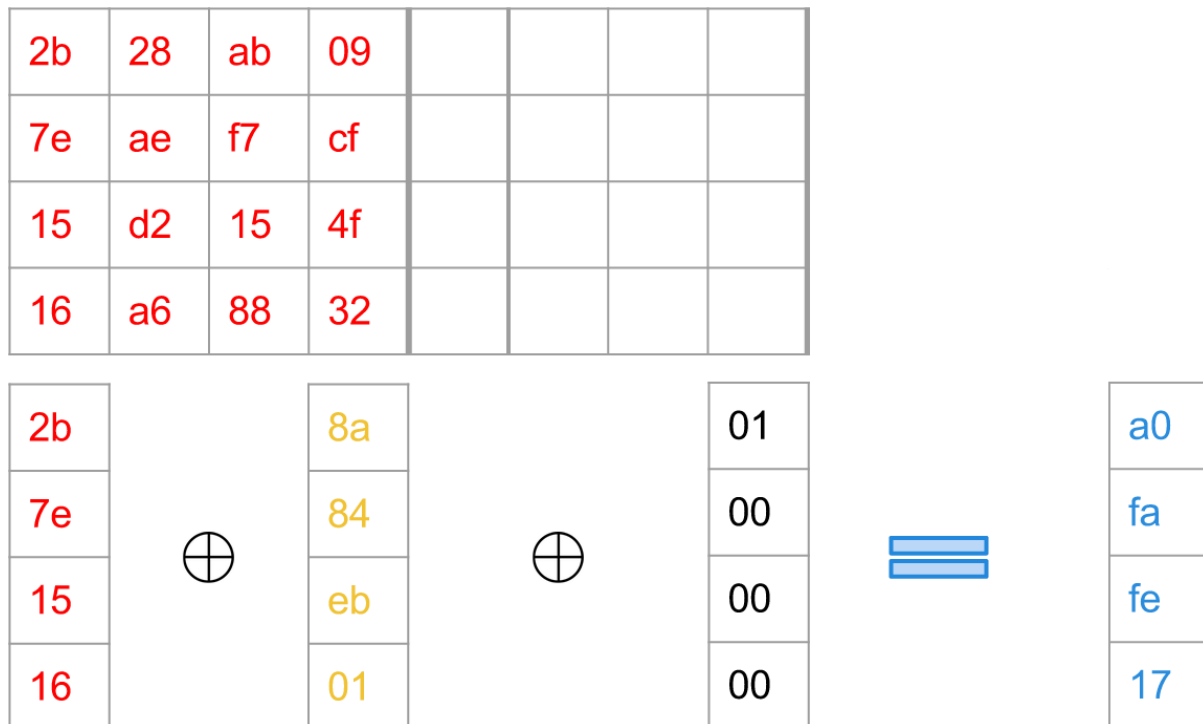


Bild 2.4: Die kurze KeySchedule Erweiterung

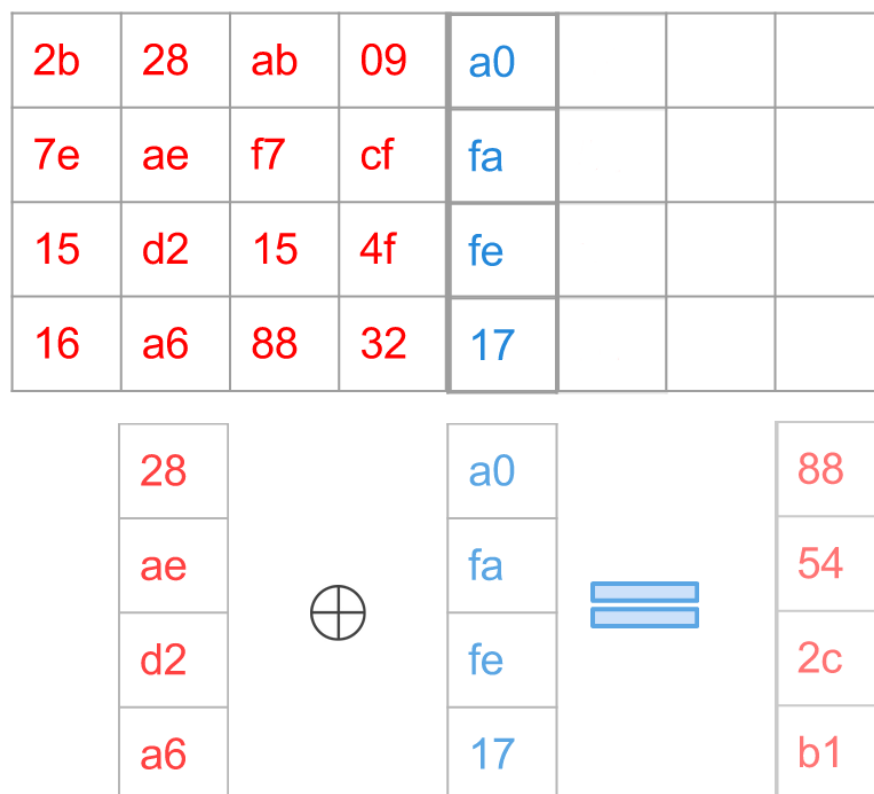


Bild 2.5: Das abgeschlossene KeySchedule für den ersten Rundenschlüssel

2b	28	ab	09	a0	88	23	2a
7e	ae	f7	cf	fa	54	a3	6c
15	d2	15	4f	fe	2c	39	76
16	a6	88	32	17	b1	39	05

Tabelle 2.6 Kombinationen eines XOR

Wert A	Wert B	XOR
0	0	0
0	1	1
1	0	1
1	1	0

Bild 2.7: Die AddRoundKey Operation

32	88	31	e0
43	5a	31	37
f6	30	98	07
a8	8d	a2	34

 \oplus

2b	28	ab	09
7e	ae	f7	cf
15	d2	15	4f
16	a6	88	3c

 $=$

19	a0	9a	e9
3d	f4	c6	f8
e3	e2	8d	48
be	2b	2a	08

Bild 2.8: SubBytes

Übersetzung von $[19]_{16}$ in die S-Box:

	0	1	2	3	4	5	6	7	8	9	a	b	c	d	e	f
0	63	7C	77	7B	F2	6B	6F	C5	30	01	67	2B	FE	D7	AB	76
1	CA	82	C9	7D	FA	59	47	F0	AD	D4	A2	AF	9C	A4	72	C0
2	B7	FD	93	26	36	3F	F7	CC	34	A5	E5	F1	71	D8	31	15
3	04	C7	23	C3	18	96	05	9A	07	12	80	E2	EB	27	B2	75
4	09	83	2C	1A	1B	6E	5A	A0	52	3B	D6	B3	29	E3	2F	84
5	53	D1	00	ED	20	FC	B1	5B	6A	CB	BE	39	4A	4C	58	CF
6	D0	EF	AA	FB	43	4D	33	85	45	F9	02	7F	50	3C	9F	A8
7	51	A3	40	8F	92	9D	38	F5	BC	B6	DA	21	10	FF	F3	D2
8	CD	0C	13	EC	5F	97	44	17	C4	A7	7E	3D	64	5D	19	73
9	60	81	4F	DC	22	2A	90	88	46	EE	B8	14	DE	5E	0B	DB
a	E0	32	3A	0A	49	06	24	5C	C2	D3	AC	62	91	95	E4	79
b	E7	C8	37	6D	8D	D5	4E	A9	6C	56	F4	EA	65	7A	AE	08
c	BA	78	25	2E	1C	A6	B4	C6	E8	DD	74	1F	4B	BD	8B	8A
d	70	3E	B5	66	48	03	F6	0E	61	35	57	B9	86	C1	1D	9E
e	E1	F8	98	11	69	D9	8E	94	9B	1E	87	E9	CE	55	28	DF
f	8C	A1	89	0D	BF	E6	42	68	41	99	2D	0F	B0	54	BB	16
	0	1	2	3	4	5	6	7	8	9	a	b	c	d	e	f


→ Der Wert $[19]_{16}$ wird ersetzt durch $[D4]_{16}$

Tabelle 2.9: Die ShiftRows Verschiebungen

Zeile	Verschiebung
Zeile 1:	0
Zeile 2:	1
Zeile 3:	2
Zeile 4:	3

Bild 2.10: ShiftRows Beispielhaft dargestellt

11	12	13	14
21	22	23	24
31	32	33	34
41	42	43	44



11	12	13	14
22	23	24	21
33	34	31	32
44	41	42	43

Bild 2.11: MixColumns

d4	e0	b8	1e
bf	b4	41	27
5d	52	11	98
30	ae	f1	e5

04	e0	48	28
66	cb	f8	06
81	19	d3	26
e5	9a	7a	4c

$$\begin{vmatrix} 2 & 3 & 1 & 1 \\ 1 & 2 & 3 & 1 \\ 1 & 1 & 2 & 3 \\ 3 & 1 & 1 & 2 \end{vmatrix} \cdot \begin{vmatrix} d4 \\ bf \\ 5d \\ 30 \end{vmatrix} = \begin{vmatrix} 04 \\ 66 \\ 81 \\ e5 \end{vmatrix}$$

Bild 2.12 Auszug aus der Schreibweise von AES als Kettenbruch:

$$a_{i,j}^{(6)} = K + \sum_{\substack{e_5 \in \mathcal{E} \\ d_5 \in \mathcal{D}}} \frac{C}{K^* + \sum_{\substack{e_4 \in \mathcal{E} \\ d_4 \in \mathcal{D}}} \frac{C}{K^* + \sum_{\substack{e_3 \in \mathcal{E} \\ d_3 \in \mathcal{D}}} \frac{C}{K^* + \sum_{\substack{e_2 \in \mathcal{E} \\ d_2 \in \mathcal{D}}} \frac{C}{K^* + \sum_{\substack{e_1 \in \mathcal{E} \\ d_1 \in \mathcal{D}}} \frac{C}{K^* + p_*^*}}}}$$

3: Das Elgamal-Kryptosystem

Bild 3.1: Der Verlauf einer Montgomery Kurve

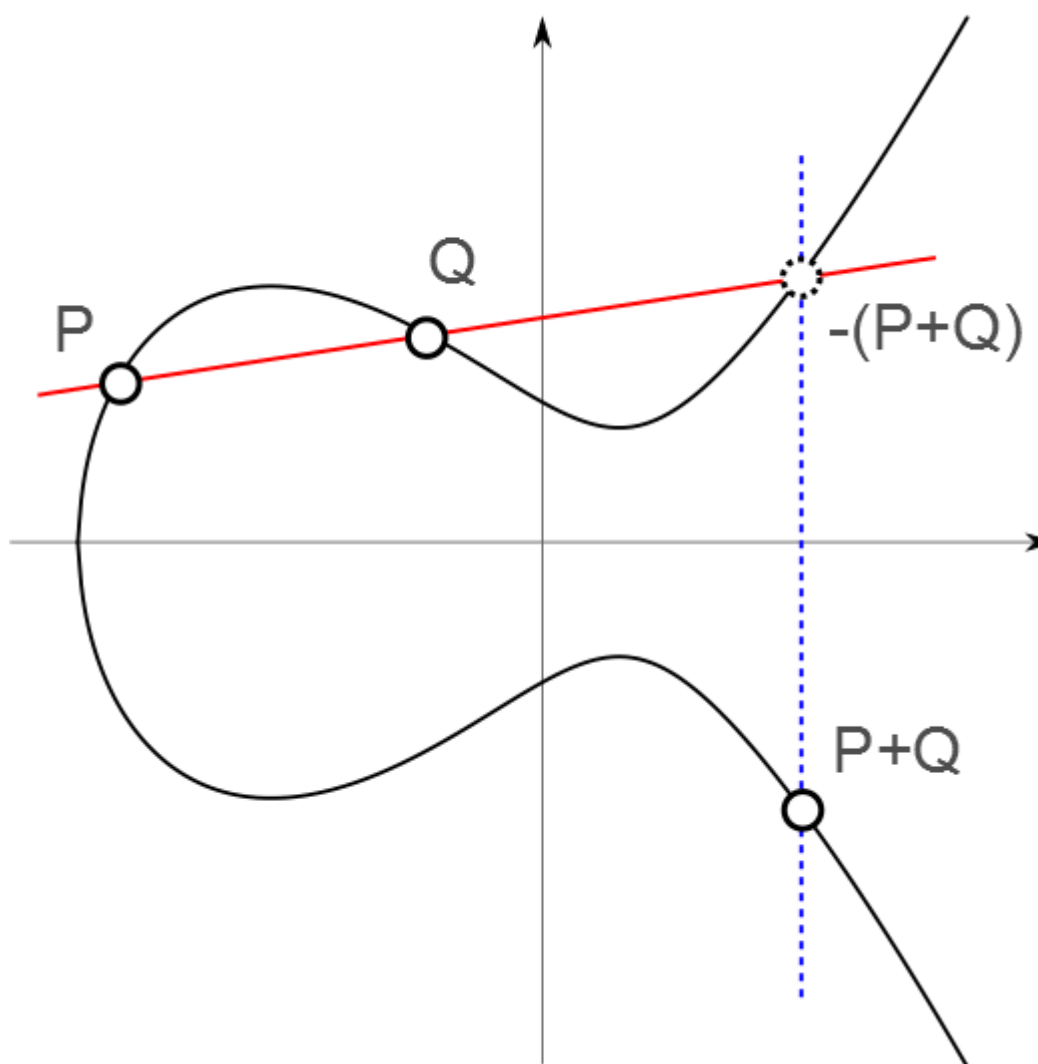
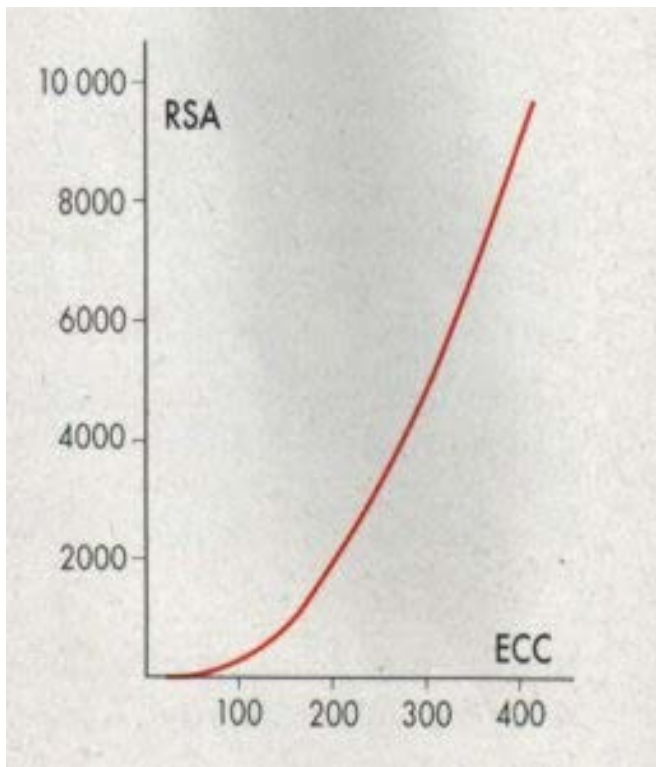


Diagramm 3.2: Diese Grafik zeigt die RSA-Schlüssellänge den ECC-Schlüssellängen zugeordnet, bei denen das gleiche Sicherheitsniveau gegeben ist.



Quellverzeichnis

Literatur

Titel	Autor	Erscheinungsjahr
Einführung in die Kryptographie	Johannes Buchmann	2., erweiterte Auflage 2001
Geheime Botschaften: Die Kunst der Verschlüsselung von der Antike bis in die Zeiten des Internet	Simon Singh	2000

Allgemeine Quellen	Zugriffsdatum
https://de.wikipedia.org/wiki/Twofish	8.03.2015
https://de.wikipedia.org/wiki/Polyalphabetische_Substitution#Vigen.C3.A8re-Verschl.C3.BCsselung https://de.wikipedia.org/wiki/Monoalphabetische_Substitution	24.04.2015
https://de.wikipedia.org/wiki/UTF-8 https://de.wikipedia.org/wiki/Enigma https://de.wikipedia.org/wiki/Steganographie https://de.wikipedia.org/wiki/One-Time-Pad	27.04.2015
https://de.wikipedia.org/wiki/Public-Key-Verschl%C3%BCsselungsverfahren	29.04.2015
http://www.spiegel.de/netzwelt/netzpolitik/nsa-neue-plaene-um-an-verschluesselte-daten-zu-kommen-a-1028284.html	7.05.2015
https://de.wikipedia.org/wiki/SHA-3	12.05.2015
https://de.wikipedia.org/wiki/Alice_und_Bob	18.05.2015

AES Quellen	Zugriffsdatum
http://www.codeplanet.eu/tutorials/cpp/51-advanced-encryption-standard.html#aes_operations_mixcolumns http://www.codeplanet.eu/files/flash/Rijndael_Animation_v4_eng.swf https://de.wikipedia.org/wiki/Advanced_Encryption_Standard https://en.wikipedia.org/wiki/Rijndael_key_schedule#Rcon http://www.adamberent.com/documents/AESbyExample.htm https://www.youtube.com/watch?v=ZE3cOe7SiMw https://en.wikipedia.org/wiki/Rijndael_S-box https://de.wikipedia.org/wiki/Rijndael_MixColumns	8-10.03.2015
https://de.wikipedia.org/wiki/Cipher_Block_Chaining_Mode	12.03.2015
https://en.wikipedia.org/wiki/Block_cipher_mode_of_operation	14.03.2015
http://csrc.nist.gov/publications/fips/fips197/fips-197.pdf http://csrc.nist.gov/archive/aes/rijndael/Rijndael-ammended.pdf http://www.staff.uni-mainz.de/pommeren/Kryptologie/Bitblock/Bitblock.pdf http://www.staff.uni-mainz.de/pommeren/Kryptologie/Bitblock/Bitblock.pdf http://www.codeplanet.eu/tutorials/cpp/52-blockchiffre-operationsmodi.html	16.03.2015
https://github.com/htx1219/Python/blob/master/387/CBC%20implementat ion.py	2.04.2015
http://keccak.noekeon.org/files.html	20.04.2015
http://kubieziel.de/studium/aes-ausarbeitung.pdf http://www.heise.de/security/artikel/Timing-Attacken-auf-AES-270720.html http://cr.yp.to/antiforgery/cachetiming-20050414.pdf http://taz.newffr.com/TAZ/Cryptologie/hash-lib-algo/aes/rdalgeq.pdf https://www.schneier.com/blog/archives/2009/07/new_attack_on_a.html	24.04.2015
https://de.wikipedia.org/wiki/Betriebsmodus_%28Kryptographie%29	3.05.2015
https://de.wikipedia.org/wiki/Blockverschl%C3%BCsselung	12.05.2015

RSA Quellen	Zugriffsdatum
http://arndt-bruenner.de/mathe/scripts/erweitertereuklid.htm https://de.wikipedia.org/wiki/RSA-Kryptosystem	9-15.03.2015
http://rosettacode.org/wiki/Miller-Rabin_primality_test	12.04.2015
http://zach.in.tu-clausthal.de/teaching/programming_0506/literatur/linuxfibel/krypto.htm http://de.wikipedia.org/wiki/RSA-Kryptosystem	24.04.2015
http://www.vorratsdatenspeicherung.de/CD/CD_1.0/cryptocd/doku/macos/verschluesselung_funktion/verschluesselung_funktion.html	27.04.2015
https://www.sec.in.tum.de/assets/lehre/ss09/kryptographie/Kapitel.10.pdf	10.05.2015
http://www.heise.de/ct/ausgabe/2015-6-Gefaelachte-PGP-Keys-im-Umlauf-2549724.html	10.05.2015

Elgamal Quellen	Zugriffsdatum
http://de.wikipedia.org/wiki/Elliptic_Curve_Cryptography http://en.wikipedia.org/wiki/Montgomery_curve http://de.wikipedia.org/wiki/Elgamal-Verschl%C3%BCsselungsverfahren http://de.wikipedia.org/wiki/Ciphertext_Indistinguishability http://de.wikipedia.org/wiki/Diffie-Hellman-Schl%C3%BCsselaustausch http://de.wikipedia.org/wiki/Gruppe_%28Mathematik%29 https://homepages.thm.de/~hg10013/Lehre/MMS/SS01_WS0102/Elyps http://crypto.stackexchange.com/questions/9987/elgamal-with-elliptic-curves http://www.ams.org/journals/mcom/1987-48-177/S0025-5718-1987-0866109-5/S0025-5718-1987-0866109-5.pdf http://en.wikipedia.org/wiki/Curve25519	8.03.2015 26.04.2015
http://crypto.stackexchange.com/questions/9987/elgamal-with-elliptic-curves	26.04.2015
http://www.demonstrations.wolfram.com/AddingPointsOnAnEllipticCurve/	19.05.2015

Bildquellen	Zugriffsdatum
http://www.formaestudio.com/rijndaelinspector/archivos/Rijndael_Animation_v4_eng.swf http://csrc.nist.gov/publications/fips/fips197/fips-197.pdf	28.04.2015
http://www.heise.de/ct/zcontent/14/08-hocmsmeta/1396445432022925/contentimages/ju_Saeulen_Gespraech_aak_IG.jpg http://www.1001freedownloads.com/free-icon/text-icon http://www.1001freedownloads.com/free-vector/old-and-modern-keys-silhouettes https://commons.wikimedia.org/wiki/Category:Padlock_icons#/media/File:El-lock.svg http://uxrepo.com/static/icon-sets/linecons/png32/256/000000/cloud-256-000000.png	30.04.2015
https://3.bp.blogspot.com/-FF_g85ZlcZO/UIWEBdXHbyI/AAAAAAAAAFK4/q8FPWOHKMx8/s1600/Invisible+Pink+Unicorn.png	3.05.2015
http://de.wikipedia.org/wiki/Enigma_%28Maschine%29#/media/File:Enigma_Verkehrshaus_Luzern_cropped.jpg http://de.wikipedia.org/wiki/Enigma_%28Maschine%29#/media/File:Enigma-action.svg http://www.asciitable.com/ http://de.wikipedia.org/wiki/Polyalphabetische_Substitution http://de.wikipedia.org/wiki/Buchstabenh%C3%A4ufigkeit	07.05.2015
http://www.ffonts.net/English-Towne-Medium.font.download http://pixabay.com/de/schl%C3%BCssel-antik-verschn%C3%B6kelte-306941/	25.05.2015