

# Asymetrische Verschlüsselungsverfahren am Beispiel vom Merkle-Hellman-Kryptosystem

Niklas Gögge  
Viktoriaschule Aachen  
Kurs: Mathematik  
Fachlehrerin: Kleines  
Schuljahr 2014/15

6. März 2015

# Inhaltsverzeichnis

<b>1</b>	<b>Einleitung</b>	<b>2</b>
<b>2</b>	<b>Mathematische Grundlagen</b>	<b>3</b>
2.1	Restrechnung . . . . .	3
2.2	Euklidischer Algorithmus . . . . .	3
2.3	Erweiterter Euklidischer Algorithmus . . . . .	4
2.4	Einwegfunktionen . . . . .	4
2.4.1	Einwegfunktionen mit Falldür . . . . .	4
2.4.2	Beispiel . . . . .	4
<b>3</b>	<b>Verschlüsselung</b>	<b>5</b>
3.1	Ziele der Verschlüsselung . . . . .	5
3.2	Verschlüsselungsprotokolle . . . . .	6
3.3	Geschichte . . . . .	6
3.4	Anwendungsbereiche . . . . .	6
<b>4</b>	<b>Symmetrische Verschlüsselung</b>	<b>6</b>
4.1	Prinzip symmetrischer Verschlüsselung . . . . .	7
4.2	Stromchiffren . . . . .	7
4.3	Blockchiffren . . . . .	8
4.4	Sicherheit . . . . .	8
<b>5</b>	<b>Asymmetrische Verschlüsselung</b>	<b>8</b>
5.1	Prinzip asymmetrischer Verschlüsselung . . . . .	8
5.2	Sicherheit . . . . .	9
<b>6</b>	<b>Merkle-Hellman Kryptosystem</b>	<b>9</b>
6.1	Über die Erfinder . . . . .	9
6.2	Rucksackproblem . . . . .	10
6.3	Schlüsselerzeugung . . . . .	11
6.4	Verschlüsselung . . . . .	11
6.5	Entschlüsselung . . . . .	12
6.6	Sicherheit und Angriffsmöglichkeiten . . . . .	12
<b>7</b>	<b>Fazit</b>	<b>14</b>

# 1 Einleitung

Diese Facharbeit entsteht im Rahmen einer Schulaufgabe im Fach Mathematik und beschäftigt sich mit asymmetrischen Verschlüsselungsverfahren am Beispiel vom Merkle-Hellman Kryptosystem. Ich habe dieses Thema gewählt da ich sehr mathematisch und informationstechnisch interessiert bin und mein späteres Studium wahrscheinlich auch in eine dieser Richtungen verlaufen wird. Des Weiteren sind Sicherheit und Geheimhaltung im heutigen Medienzeitalter ein sehr interessantes und vor allem wichtiges Thema, da ein sehr großer Teil der Kommunikation über Telefone beziehungsweise über das Internet läuft und Sicherheit an manchen Stellen sehr wichtig ist.

Zu Beginn der Arbeit werden die Mathematischen Grundlagen, welche für diese Arbeit gebraucht werden, erläutert. Dazu gehört: Restrechnung, Der euklidische Algorithmus und der erweiterte euklidische Algorithmus und Einwegfunktionen mit und ohne Falltür. Im darauf vollgenden Kapitel wird über Verschlüsselung im Allgemeinen, ihre Ziele und Anwendungsbereiche geredet. Die zwei darauf folgenden Kapitel befassen sich mit symmetrischer und asymmetrischer Verschlüsselung. Beide Prinzipien werden erläutert und die Sicherheit und Anwendungsbereiche beider Arten werden beschrieben. Das sechste Kapitel dreht sich um das Merkle-Hellman Kryptosystem welches als Beispiel für ein asymmetrisches Verschlüsselungsverfahren erläutert wird. Dazu gehört eine kurze Biographie der Erfinder, das Rucksackproblem, das Verfahren an sich, also Schlüsselerzeugung und Ver- und Entschlüsselung und es wird erklärt warum das Verfahren nicht sicher ist. Zum Schluss der Arbeit werde ich noch ein persönliches Fazit aus meiner Bearbeitung des Themas ziehen.

Im Rahmen dieser Arbeit werde Ich zusätzlich das Merkle-Hellman Kryptosystem selber in der Programmiersprache Java Programmieren.

## 2 Mathematische Grundlagen

In diesem Kapitel werden die Mathematischen Grundlagen erklärt die man zum Verständnis dieser Arbeit benötigt.

### 2.1 Restrechnung

Die Restrechnung ist den meisten vermutlich noch aus der Grundschule bekannt, denn bevor Rationale zahlen eingeführt wurden, war zum Beispiel: 123 geteilt durch 10 nicht etwa 12.3 sondern das Ergebnis lautete: 12 Rest 3. Allgemein ausgedrückt heißt es das bei einer Division zweier Zahlen  $a, b$ , mit  $b \neq 0$ ,  $a, b \in \mathbb{Z}$  ein Rest  $r$  entstehen kann. Eine mögliche Schreibweise dafür wäre:

$$r = a \bmod b$$

Auf das obrige Beispiel bezogen würde man also  $12 = 123 \bmod 10$  schreiben. Das mod steht für modulo.

Bei Restrechnung geht es auch oft um Restklassen. Eine Restklasse besteht immer aus Zahlen welche geteilt durch eine Zahl  $b$ , mit  $b \in \mathbb{Z}$  den selben Rest ergeben. So sind zum Beispiel die Zahlen 2; 5; 11; 38; 41; 86; 101 alle ein Teil der Restklasse modulo 3, denn sie alle ergeben bei der Division durch 3 den gleichen Rest. Im Zusammenhang mit Restklassen wird auch oft von Kongruenz geredet. Zwei zahlen  $a, b$  sind kongruent zueinander, wenn sie dividiert durch eine dritte Zahl  $c$  den selben Rest  $r$  überlassen. Die Schreibweise dafür ist:

$$a \equiv b \bmod c$$

Ausgesprochen heißt das:  $a$  ist kongruent  $b$  modulo  $c$ . Das heißt also das alle Zahlen einer Restklasse Kongruent zueinander sind.

### 2.2 Euklidischer Algorithmus

Viele asymmetrische Verschlüsselungsverfahren müssen häufig den größten gemeinsamen Teiler oder kurz den ggT zweier oder mehrerer Zahlen errechnen und der euklidische Algorithmus ist nichts anderes als eine Methode um dies zu tun. Er erhielt seinen Namen von seinem Erfinder Euklid.

Um den ggT zweier ganzer Zahlen  $a, b$  mit  $a > b$  zu bestimmen, wird zu erst  $c = a \bmod b$  berechnet. Dann wird der Wert von  $a$  in den von  $b$  geändert und der von  $b$  in den von  $c$ . Danach wird dieser Vorgang mit "neuem"  $a$  und  $b$  wiederholt solange bis  $c$  gleich 0 ist. Und dann ist der letzte von  $b$  angenommene Wert der ggT von  $a$  und  $b$ .

## 2.3 Erweiterter Euklidischer Algorithmus

Der erweiterte euklidische Algorithmus bestimmt zusätzlich zum ggT von zwei ganzen Zahlen  $a, b$  mit  $b \neq 0$  noch zwei weitere ganze Zahlen  $s, t$ . Die Gleichung die dabei gelöst wird ist folgende:

$$\text{ggT}(a, b) = s \times a + t \times b$$

Die häufigste Anwendung dieses Algorithmus ist die Bestimmung von Multiplikativen Inversen.

## 2.4 Einwegfunktionen

Eine Einwegfunktion ist eine Funktion welche leicht zu berechnen ist, also  $y = f(x)$  ist schnell zu berechnen, aber es nur sehr schwer bis unmöglich ist die Funktion umzukehren, also  $x = f^{-1}(y)$  ist nicht effizient zu berechnen. Es wird angenommen das solche Funktionen existieren und es gibt auch einige Funktionen bei denen man annimmt das es sich um Einwegfunktionen handelt, allerdings ist dies nicht bewiesen. Einwegfunktionen werden vorallem in der asymmetrischen Verschlüsselung häufig angewendet, allerdings wird dort eine spezielle Art von Einwegfunktionen verwendet, sogenannte Einwegfunktionen mit Falltür.

### 2.4.1 Einwegfunktionen mit Falltür

Einwegfunktionen mit Falltür sind Einwegfunktionen mit einer zusätzlichen Eigenschaft. Sie sind Umkehrbar allerdings nur mit den richtigen Falltür- /Geheiminformationen.

### 2.4.2 Beispiel

Ein Beispiel für eine Einwegfunktion ist die Multiplikation zweier Primzahlen. Eine solche Funktion könnte  $f(a, b) = a \times b$  lauten.

Wählt man jetzt zum Beispiel:  $a = 37967$  und  $b = 41389$  so ist das ergebnis von  $f(37967, 41389)$  die 1571416163. Das multiplizieren von  $a$  und  $b$  geht schnell, also ist die erste Bedingung einer Einwegfunktion erfüllt und ohne weiteres ist es auch nicht leicht aus dem Ergebnis die Beiden Primzahlen  $a, b$  herauszuziehen außer eine der Beiden ist bekannt und somit ist die zweite Bedingung der Definition auch erfüllt. Allerdings kann ein Computer in diesem Fall  $a, b$  trotzdem schnell errechnen, da  $a, b$  Verhältnis mäßig sehr klein sind. Wählt man  $a, b$  sehr groß, so ist praktisch unmöglich aus dem Ergebnis die beiden Primzahlen zu zerlegen. Diese Einwegfunktion wird vom RSA Kryptosystem verwendet, welches in dieser Arbeit allerdings nicht erläutert wird.

## 3 Verschlüsselung

Verschlüsselung oder auch Kryptographie ist wie in der Einleitung schon erwähnt, in der heutigen Zeit ein sehr wichtiges Thema. Aber auch früher wurde schon verschlüsselt. Die ersten Verschlüsselungsverfahren entstanden vor ca. 2000 Jahren, diese wurden natürlich per Hand auf Papier ausgeführt. Heutzutage läuft alles über Computer.

Es gibt zwei Arten der Verschlüsselung: die symmetrische (mehr dazu in Kapitel 4) und die asymmetrische (mehr dazu in Kapitel 5) Verschlüsselung. Das Prinzip hinter Verschlüsselung ist bei allen Verschlüsselungsverfahren das gleiche. Zu erst gibt es irgendeine Nachricht, einen Text, eine Datei oder irgendwelche Daten die verschlüsselt werden sollen, meist werden diese Daten als Klartext bezeichnet. Nach dem dieser Klartext mit irgendeinem Verfahren, symmetrisch oder asymmetrisch, verschlüsselt wurde erhält man eine verschlüsselte Form dieses Klartextes, welche meist als Geheimtext bezeichnet werden. Dieser Geheimtext wird dann verschickt und der Empfänger kann dann mit Hilfe des passenden Entschlüsselungsverfahrens aus dem Geheimtext den ursprünglichen Klartext erhalten. Beim Ver- /Entschlüsseln werden meist Schlüssel benutzt. Allerdings gibt es, was das angeht, Unterschiede in symmetrischer und asymmetrischer Verschlüsselung.

### 3.1 Ziele der Verschlüsselung

In der Kryptographie gibt es vier Hauptziele:

- *Vertraulichkeit*. Es sollte nur berechtigten Personen möglich sein die Daten verstehen zu können.
- *Datenintegrität*. Es sollte für den Empfänger nachvollziehbar sein, ob die erhaltenen Daten manipuliert wurden.
- *Authentizität*. Der Empfänger muss nachvollziehen können von wo die Daten kamen. Es darf einem Angreifer nicht möglich sein den Ursprung zu fälschen.
- *Nichtabstreitbarkeit/Abstreitbarkeit*. Je nach Kontext, sollte es für den Sender nicht die Möglichkeit geben, abzustreiten oder nicht abzustreiten, dass er die Nachricht gesendet hat.

Die Vertraulichkeit wird erreicht in dem ein sicherer Verschlüsselungsalgorithmus benutzt wird. Ein Verschlüsselungsalgorithmus wird als sicher bezeichnet, wenn es nur sehr schwer ist den Geheimtext, ohne passenden Schlüssel, in den Klartext umzuwandeln. Datenintegrität wird erreicht durch so genannte message authentication codes oder kurz MAC. Solche Codes werden durch einen Algorithmus (meist Hashalgorithmen) erzeugt, welcher dafür den Schlüssel und den Klartext benutzt. Das Erreichen von Authentizität und Nichtabstreitbarkeit wird bei asymmetrischen Verfahren durch

digitale Signaturen sichergestellt. Diese sind zu vergleichen mit einer Unterschrift auf einem Blatt Papier.

### **3.2 Verschlüsselungsprotokolle**

Wenn alle Bestandteile eines Verschlüsselungsverfahrens (Ver- /Entschlüsselung und Methoden zum Erreichen von Datenintegrität, Authentizität, Nichtabstreitbarkeit) zusammen kommen, nennt man die Abfolge dieser einzelnen Bestandteile ein Verschlüsselungsprotokoll.

### **3.3 Geschichte**

Die ersten Verschlüsselungsverfahren waren symmetrische Verfahren(4) und entstanden wie schon erwähnt vor sehr langer Zeit. Asymmetrische Verschlüsselung(5) ist im Vergleich zu symmetrischer noch recht jung. Im Jahr 1975 veröffentlichten Martin Hellman und Whitfield Diffie ihr Diffie-Hellman-Schlüsselaustausch Protokoll und damit die ersten Ideen zur asymmetrischen Verschlüsselung. Das erste wirkliche asymmetrische Verschlüsselungsverfahren wurde allerdings erst 1977 von R. L. Rivest, A. Shamir und L. M. Adleman erfunden. Es erhielt den Namen RSA und ist bis heute immer noch eins der bekanntesten und am häufigsten genutzten Verfahren. Andere bekannte Verfahren sind McEliece(1978), Rabin(1979), Chor-Rivest(1984) und ElGamal(1985).

### **3.4 Anwendungsbereiche**

Verschlüsselt wird immer dann, wenn bestimmte Daten nur für ausgewählte Personen zugänglich sein sollen. Zum Beispiel sollte, bei einer Unterhaltung zweier Staatsechefs, über das Internet, niemand in der Lage sein die Unterhaltung einfach mitlesen zu können. Da bei asymmetrischen Verfahren meist größere Datenmengen verschickt werden müssen, werden für große Datenmengen meist symmetrische Verfahren benutzt werden(z.B.: Videostreaming), allerdings findet die Schlüsselverteilung(mehr in Kapitel 4.4) dabei über asymmetrische Verfahren statt. Ein sehr bekanntes Beispiel dafür ist das kleine Schloss Symbol, welches bei manchen Websites oben neben der URL erscheint(z.B. bei: [www.google.com](http://www.google.com)).

## **4 Symmetrische Verschlüsselung**

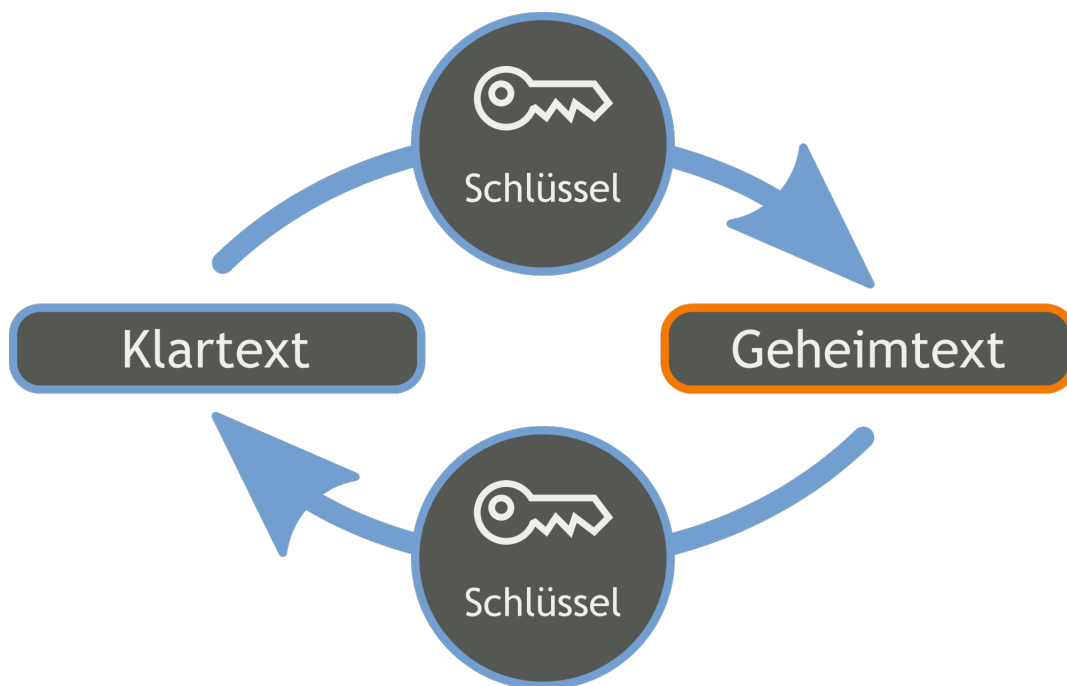
Symmetrische Verschlüsselungsverfahren gibt es schon deutlich länger als asymmetrische Verfahren und sie arbeiten im Gegensatz zu ihnen auch mit jeweils immer nur einem einzigen Schlüssel für Ver- und Entschlüsselung. Dieser muss unter den Kommunikationspartnern vorher ausgetauscht werden. Das nennt man auch Schlüsselverteilung(mehr dazu in Kapitel 4.4).

Bei symmetrischen Verfahren wird zwischen Strom- und Blockchiffren<sup>1</sup> unterschieden. Beispiele für bekannte symmetrische Verfahren sind: Caesar, A5, Blowfish, Twofish, AES, DES.

## 4.1 Prinzip symmetrischer Verschlüsselung

Das Prinzip hinter symmetrischer Verschlüsselung ist recht simpel. Es gibt nur einen Schlüssel für Ver- und Entschlüsselung das bedeutet, dass alle Parteien die an einer Kommunikation teilnehmen den selben Schlüssel haben. Wenn jetzt Partei A einer zweiten Partei B eine Nachricht schicken will, wird diese Nachricht mit dem Schlüssel durch den Algorithmus des jeweiligen Verfahrens verschlüsselt dadurch erhält Partei A einen Geheimtext. Dieser wird nun an Partei B geschickt. Partei B kennt den zuvor Ausgetauschten Schlüssel und kann nun mit ihm durch den passenden Entschlüsselungsalgorithmus den Geheimtext entschlüsseln und erhält die von Partei A verschlüsselte Nachricht.

Folgende Grafik verdeutlicht dieses Schema:



## 4.2 Stromchiffren

Stromchiffren verarbeiten immer jeden Bit einer Nachricht einzeln, also ein Bit rein ein Bit raus. Dabei kommt es zu dem Problem, dass der Empfänger nicht weiß, an welcher Stelle der Sender im Strom gerade ist und falls mal ein Teil der Nachricht, bei der Übertragung verloren gehen sollte, führt das auch zu Problemen.

---

<sup>1</sup>Chiffren sind methoden zum Ver- und Entschlüsseln.



## 4.3 Blockchiffren

Blockchiffren verarbeiten immer eine bestimmte Menge an Bits einer Nachricht, in einem ganzen Block. Das Problem, dabei ist, dass manchmal ein Block zu klein ist und somit die Verschlüsselung überhaupt nicht stattfinden kann.

## 4.4 Sicherheit

Die Sicherheit von symmetrischen Verfahren hängt stark vom Schlüssel ab. Ein zu kurzer Schlüssel zum Beispiel könnte schnell errechnet werden und somit wäre die Kommunikation nicht mehr sicher, deshalb ist ein langer Schlüssel<sup>2</sup> wichtig, allerdings bringt auch ein langer Schlüssel nichts, wenn er eines Tages an die Öffentlichkeit geraten sollte, denn dann könnte jeder der die Entsprechenden Geheimdaten gesammelt hat den Schlüssel nutzen um diese zu Entschlüsseln. Es ist also sehr wichtig dass der Schlüssel geheim bleibt, deshalb ist die Schlüsselverteilung auch ein wichtiger Punkt, wenn man über Sicherheit von symmetrischen Verfahren redet. Die Schlüsselverteilung findet Heutzutage häufig über asymmetrische Verfahren statt oder durch direkten Kontakt der Kommunikationspartner.

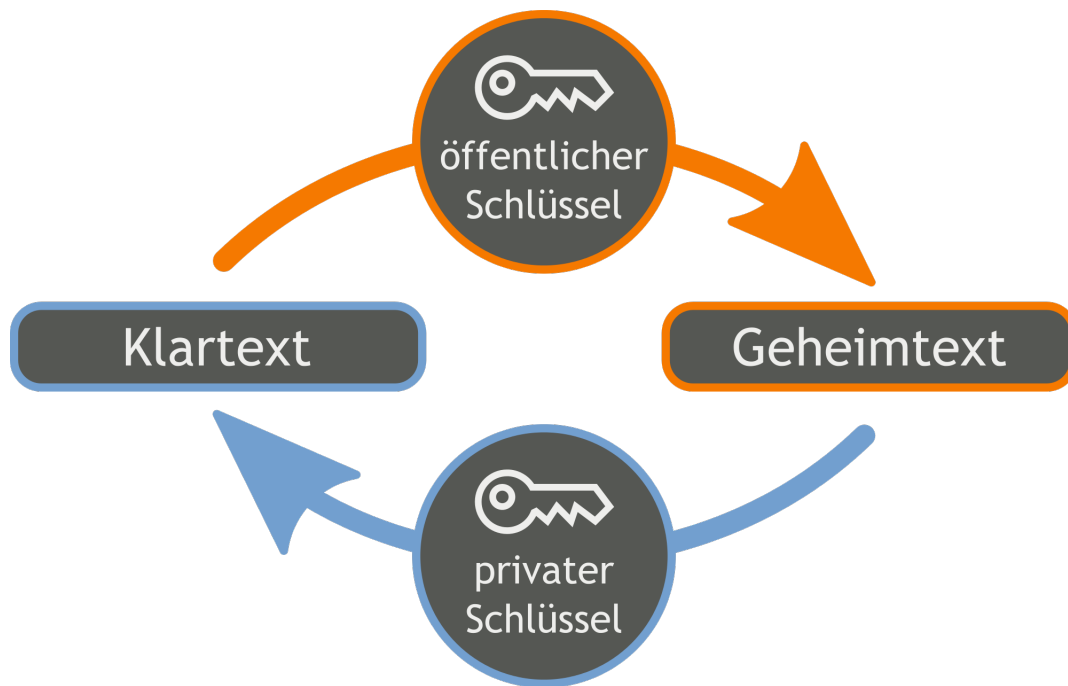
# 5 Asymmetrische Verschlüsselung

## 5.1 Prinzip asymmetrischer Verschlüsselung

Asymmetrische Verschlüsselung ist das Gegenstück zu symmetrischer Verschlüsselung, denn hier gibt es nicht nur einen Schlüssel, der für Ver- und Entschlüsselung gebraucht wird, sondern jede Partei einer Kommunikation besitzt einen öffentlichen Schlüssel, welcher für jeden zugänglich ist, und einen privaten Schlüssel, welcher nur der jeweiligen Partei bekannt ist. Wenn jetzt zum Beispiel Partei A einer zweiten Partei B eine Nachricht schicken will, fragt Partei A bei Partei B nach ihrem öffentlichen Schlüssel. Die Übertragung dieses Schlüssels muss nicht sicher sein. Nach dem sie diesen erhalten hat verschlüsselt Partei A ihre Nachricht mit diesem öffentlichen Schlüssel über einen Verschlüsselungsalgorithmus und erhält einen Geheimtext, den sie selber nicht mehr entschlüsseln kann. Dieser Geheimtext wird dann zu Partei B geschickt und diese kann dann mit Hilfe ihres privaten Schlüssels und dem passenden Entschlüsselungsalgorithmus den Geheimtext entschlüsseln. Die Ver- und Entschlüsselungsalgorithmen, welche Partei A und B benutzen, benutzen eine Einwegfunktion mit Falltür(Definition in Kapitel 2.4.1), welche als Falltürinformation den jeweiligen privaten Schlüssel benutzt. Folgende Grafik verdeutlicht dieses Schema:

---

<sup>2</sup>Heutzutage 1024 Bit.



## 5.2 Sicherheit

Asymmetrische Verschlüsselungsverfahren haben das Problem der Schlüsselverteilung nicht, weshalb sie gegen über symmetrischen Verfahren in dem Punkt sicherer sind. Des weiteren gibt es bis jetzt keine effizienten Algorithmen um die beschriebenen Einwegfunktionen umzukehren ohne die entsprechenden Geheiminformationen zu kennen.

# 6 Merkle-Hellman Kryptosystem

Das asymmetrische Merkle-Hellman Kryptosystem wurde 1978 von Ralph Merkle und Martin Hellman erfunden und ist eins der ersten asymmetrischen Kryptosysteme. Es verwendet anders als andere Verfahren(z.B.: RSA) keine digitalen Signaturen. Das Verfahren basiert auf dem Rucksackproblem, welches in Kapitel 6.2 erläutert wird.

## 6.1 Über die Erfinder

Ralph Merkle wurde 1952 in Californien geboren. Er schloss 1974 sein Informatikstudium in UC Berkley ab. Im Jahr 1974 stellte er seinen Professoren seine Entdeckung zum kryptographischen Schlüsselaustausch, heute bekannt als Merkle's Puzzles, vor. Allerdings hatten sie kein weiteres Interesse an seiner Idee. Doch als er auf Whitfield Diffie und Martin Hellman traf kam er seine Idee wieder auf und zusammen entwickelten sie Das Merkle-Hellman Kryptosystem. Heutzutage ist Merkle in der Nanotechnologieberanche tätig.

Martin Hellman wurde 1945 in New York geboren. Er schloss sein Studium in Elektrotechnik 1969 an der Stanford University ab. Im Jahre 1976 publizierte er zusammen

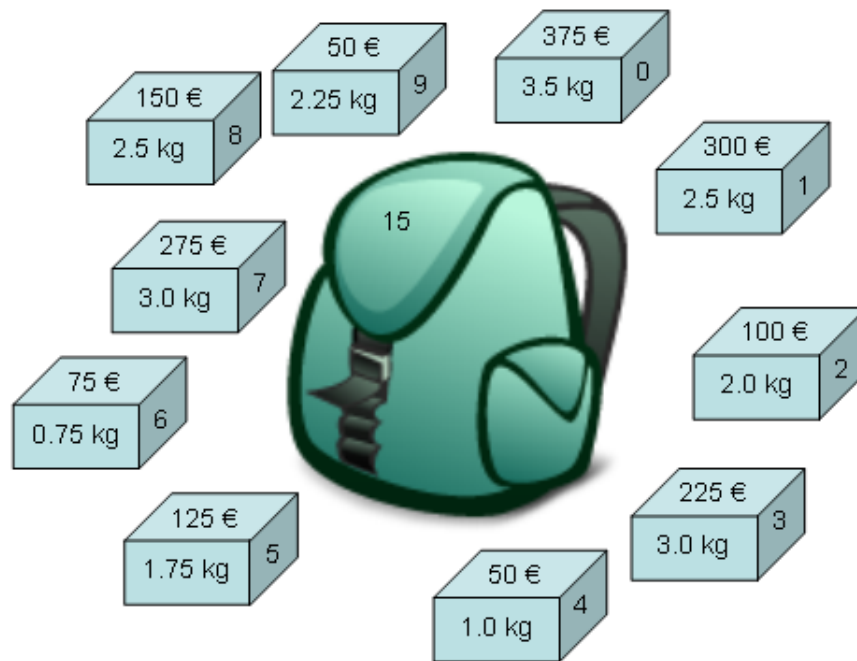
mit Whitfield Diffie einen Artikel über den Diffie-Hellman Schlüsselaustausch und erste Ideen zur asymmetrischen Verschlüsselung. Heutzutage ist Hellman Professor in Stanford im Fach Elektrotechnik.

## 6.2 Rucksackproblem

Da das Merkle-Hellman Kryptosystem auf dem Rucksackproblem basiert wird es in diesem Unterkapitel erläutert.

Das Rucksackproblem ist leicht mit einem Beispiel zu erklären: Man stelle sich vor man hat einen Rucksack, der nur mit einer bestimmten Menge an Gewicht voll gepackt werden kann. Des weiteren liegen neben dem Rucksack ein paar Gewichte, welche nicht gleich schwer sind und jeweils einen bestimmten Wert haben. Welche Gewichte kommen in den Rucksack so das die Gewichtsgrenze nicht überschritten wird und der maximale Wert (Die Summe aller Werte, der Gewichte im Rucksack) erzielt wird?

Hier ein Beispiel:



Der abgebildete Rucksack kann maximal 15kg tragen. Um ihn herum liegen 10 Gewichte aber welche werden eingepackt? Um dies heraus zu finden werden alle Möglichkeiten, inklusive keine und alle Gewichte im Rucksack, durch probiert und dann geschaut welche Möglichkeiten unter der Gewichtsgrenze liegen und den größten Wert erzielen. Für  $n$  Gewichte gibt  $2^n$  verschiedene Möglichkeiten. Für das Beispiel gibt es also  $2^{10} = 1024$  Möglichkeiten. Das sind noch nicht allzu viele und lässt sich deshalb schnell berechnen. Hat man jetzt allerdings weit aus mehr Gewichte lässt sich das Problem nicht mehr effizient lösen. Es gibt eine Ausnahme, welche für das Merkle-Hellman Kryptosystem sehr wichtig ist, diese wird in den nächsten unter Kapiteln erklärt.

### 6.3 Schlüsselerzeugung

Bei der Schlüsselerzeugung wird zuerst der private Schlüssel generiert, dieser muss ein stark wachsender Vektor sein, dass spielt bei der Entschlüsselung eine entscheidene Rolle, und eine Länge von acht Elementen haben, da ein Buchstabe in Binärcode acht Bits hat. Ein solcher privater Schlüssel  $S_p$  könnte folgender Maßen aussehen:

$$S_p = \{44, 45, 90, 180, 360, 720, 1440, 2880\}$$

Nachdem der private Schlüssel generiert wurde, wird der öffentliche Schlüssel erstellt. Dazu benötigt man zwei weitere Zahlen  $m, n$ , mit  $m, n \in \mathbb{Z}$ .  $m$  wird so gewählt, dass sie größer ist als die Summe aller Elemente des privaten Schlüssels. Im Fall von  $S_p$ , mit  $k$  als die Anzahl aller Elemente:

$$m > \sum_{i=0}^{k-1} S_p[i]$$

$n$  wird zufällig gewählt allerdings so, dass der  $ggT(m, n) = 1$ . Für unser Beispiel wählen wir  $m = 16511$  und  $n = 11111$ . Um jetzt den öffentlichen Schlüssel  $S_o$  zu erstellen, rechnet man für jedes Element  $i$ :

$$S_o[i] = (S_p[i] \times n) \bmod m$$

Für unseres Beispiel ergibt sich dann der öffentliche Schlüssel:

$$S_o = \{10065, 4665, 9330, 2149, 4298, 8596, 681, 1362\}$$

### 6.4 Verschlüsselung

Bei der Verschlüsselung wird der Klartext  $T_k$  erst in Binärcode umgewandelt. Für das Beispiel wählen wir Hallo als Klartext.

$$T_k = \{01101000, 01100001, 01101100, 01101100, 01101111\}$$

Nun wird jeder Bit von jedem Element von  $T_k$  mit jedem Element von  $S_o$  multipliziert und diese jeweils immer acht Produkte werden addiert, so erhält man die einzelnen "Geheimbuchstaben", welche zusammen den Geheimtext  $T_g$  ergeben. Für unser Beispiel:

$$T_g = \{18293, 15357, 26889, 26889, 28932\}$$

## 6.5 Entschlüsselung

Zum Entschlüsseln benötigt man  $S_p, n, m$  und das modulare Inverse  $m^{-1}$  von  $m$  und  $n$ , welches mit dem erweiterten euklidischen Algorithmus berechnet wird. Für unser Beispiel  $m^{-1} = 584$ . Nun wird, um den Klartext zu erhalten, zu erst jedes Element  $i$  von  $T_g$  mit  $m^{-1}$  multipliziert und mit Rest durch  $m$  Dividiert.

$$T_g[i] = (T_g[i] \times m^{-1}) \bmod m$$

Für das Beispiel ergibt sich die transformierte Version von  $T_g$ :

$$T_g = \{495, 3015, 1215, 1215, 5535\}$$

Danach kommt der Teil in dem die erwähnte Ausnahme des Rucksackproblems wichtig wird. Analog stehen hier die Elemente des privaten Schlüssel  $S_p$  für die Gewichte, welche alle 1kg wiegen und den Wert  $S_p[i]$  haben. Die Gewichtsgrenze des Rucksacks liegt bei 8kg, also würden theoretisch alle Gewichte hinein passen, allerdings wird hier nicht probiert den maximalen Wert zu erreichen, sondern es wird probiert die Werte in  $T_g$  zu erreichen und deshalb ist es wichtig das der private Schlüssel ein stark wachsender Vektor ist, denn nur dann ist das Rucksackproblem schnell durch einen Greedyalgorithmus zu lösen. Man erhält als Lösung den Klartext in Binärcodes, in denen eine 1 für ein zutreffendes Element im Schlüssel zur Lösung steht und eine 0 für ein nicht zutreffendes Element.

Hier das ganze für den ersten Buchstaben am Beispiel:

Für  $T_g[0]$  soll eine Lösung aus den Werten von  $S_p$  gefunden werden, um den ersten Buchstaben in binär zu erhalten.

$$T_g[0] = 495$$

Die Lösung lautet 01101000, da:

$$(0 \times 44) + (1 \times 45) + (1 \times 90) + (0 \times 180) + (1 \times 360) + (0 \times 720) + (0 \times 1440) + (0 \times 2880) = 495$$

Das selbe muss jetzt für jedes Element in  $T_g$  getan werden und man erhält den Klartext:

$$T_k = \{01101000, 01100001, 01101100, 01101100, 01101111\}$$

## 6.6 Sicherheit und Angriffsmöglichkeiten

Zur generellen Sicherheit des Verfahrens sollte man sagen, dass solange  $m, n, m^{-1}, S_p$  nicht bekannt sind, es schwer sein sollte das System zu knacken, allerdings hat Adi Sha-

mir, einer der Erfinder des RSA-Verfahrens, es 1982 trotzdem geschafft. Shamirs Methode, zum Knacken des Verfahrens ist sehr komplex und würde den Rahmen dieser Facharbeit definitiv sprengen. Des Weiteren beinhaltet das Merkle-Hellman-Kryptosystem keine Methoden zum Signieren was mit Hinblick auf die Sicherheit ein großes Problem ist, denn sind zwei Ziele von Verschlüsselung nicht erreicht (Authentizität und Nichtabstreitbarkeit).

## 7 Fazit

[4] [1] [2] [7] [8] [6] [5][3]

# Literatur

- [1] Lukas Dölle. Einwegfunktionen Variationen und Beispiele.  
[http://www2.informatik.hu-berlin.de/Forschung\\_Lehre/algorithmenII/Lehre/WS2002-2003/Perlen/Variationen.pdf](http://www2.informatik.hu-berlin.de/Forschung_Lehre/algorithmenII/Lehre/WS2002-2003/Perlen/Variationen.pdf). (27.2.2015).
- [2] Philip Hauer. Asymmetrische Verschlüsselung. Das Verfahren sowie die Vor- und Nachteile. <http://www.philippbauer.de/info/info/asymmetrische-verschluesselung/>. (23.2.2015).
- [3] Hendrik Kahl. Asymmetrische Verschlüsselungsverfahren und ihre Anwendung an ausgewählten Beispielen. <http://b-cw-waos.logoip.de:8079/fachbereiche/mathematik/msa-und-abitur/assets/facharbeit-hendrik-kahl.pdf>. (25.2.2015).
- [4] Hans Delfs, Helmut Knebl. *Introduction to Cryptography*. Springer-Verlag, 2002.
- [5] Neeraj Kumar. Merkle-Hellman Knapsack Cryptosystem.  
<http://www.neerajkumar.net/blog/2012/11/21/merkle-hellman-knapsack-cryptosystem/>. (1.3.2015).
- [6] Marc Michele. Schlüsselverteilung. <http://www.pgp-tutorial.de/node8.html>. (24.2.2015).
- [7] Computer History Museum. Martin Hellman. <http://www.computerhistory.org/fellowawards/hall/bios/Martin,Hellman/>. (24.2.2015).
- [8] Computer History Museum. Ralph Merkle.  
<http://www.computerhistory.org/fellowawards/hall/bios/Ralph,Merkle/>. (24.2.2015).