Fachhochschule Wedel
Seminararbeit

Fachrichtung Wirtschaftsingenieurwesen

Der Diffie-Hellman-Schlüsselaustausch und das elGamal-Verfahren

Erstellt von:	Melanie Maßmann (Mat-Nr. 101599) wing101599@fh-wedel.de
Erarbeitet im	6. Semester
Abgegeben am:	10. Juni 2018
Betreuender Dozent:	Prof. Dr. Michael Anders Fachhochschule Wedel Feldstraße 140 22880 Wedel Tel. (04103) 804824 E-Mail: an@fh-wedel.de

Inhaltsverzeichnis

Abkürzungsverzeichnis

DH-Schlüsselaustausch	Diffie-Hellman-Schlüsselaustausch
MitM-Angriff	Man-in-the-middle-Angriff
CDH-Problem	Computational-Diffie-Hellman-Problem
DDH-Problem	Decisional-Diffie-Hellman-Problem

1. Einleitung

Das Internet wurde von Anfang an unsicher entworfen. denn das Modell, auf dem es basiert ist „eine Gruppe sich gegenseitig vertrauender Benutzer, die an ein transparentes Netzwerk angeschlossen sind“.[1] Bei diesem Modell ist keine Sicherheit notwendig. Die Internetarchitektur ist eindeutig in vielerlei Hinsicht von dieser Vorstellung geprägt. Beispielsweise ist es Normalität, dass ein Benutzer ein Paket an jeden anderen senden kann, anstatt das jemand dies erst gestattet, nachdem er eine Anfrage bekommen hat. Des Weiteren wird auch der Benutzeridentität vertraut und nicht grundlegend geprüft.[2]

Das Problem dabei heute ist, dass es im Internet kaum Benutzer gibt, die sich gegenseitig vertrauen. Sie möchten jedoch trotzdem miteinander kommunizieren können, wobei sie nicht nur anonym bleiben wollen, sondern auch, dass die von Ihnen ausgetauschten Daten nicht in falsche Hände geraten. Dies gestaltet sich nicht so einfach, da sie sich nicht nur gegenseitig nicht trauen, sondern auch die Hardware, die Software und das Transportmedium, mit denen sie Daten austauschen, als unsicher erachten. Ein Mittel das Internet jedoch ein wenig sicherer zu machen ist die Verschlüsselung von Daten.

In dieser Seminararbeit werden der Diffie-Hellman-Schlüsselaustausch (DH-Schlüsselaustausch) und das elGamal-Verfahren behandelt. Beides sind Verschlüsselungsverfahren, die dabei helfen Daten über unsichere Kanäle verschicken zu können, welche nur für Personen lesbar sind, denen vertraut wird.

2. Grundlagen der Verschlüsselung

Die Kryptografie beschäftigt die Menschheit schon seit geraumer Zeit. Die erste dokumentierte Chiffrierung findet sich bereits im dritten Jahrtausend v. Chr. im alten

[1] (Blumenthal, 2001)
[2] (Kurose, Ross, 2008)

Ägypten, wo Namen von Göttern, die nicht genannt werden durften, mit Hilfe von Bildern und Wörtern, die für einzelne Buchstaben standen, verschlüsselt wurden.

Eine gelungene Verschlüsselung macht es möglich, Informationen vor unbefugten geheim zu halten, sodass nur eingeweihte Menschen Zugang zu ihr haben.

Der DH-Schlüsselaustausch und auch das elGamal-Verfahren sind Verfahren mit asymmetrischer Verschlüsselung. Dies bedeutet, dass sie auf komplexen mathematischen Berechnungen basieren für die es noch keine Vereinfachung gibt. Der Vorteil dabei ist, dass Daten verschlüsselt sicher über unsichere Kanäle übermittelt werden können. Da es nicht unbegrenzt viele geeignete mathematische Berechnungen gibt ist auch die Zahl an asymmetrischen Verschlüsselungsverfahren begrenzt.[3]

3. Der Diffie-Hellman-Schlüsselaustausch

Der DH-Schlüsselaustausch ist ein Verfahren mit dem Schlüssel über unsichere Kanäle ausgetauscht werden können, ohne dass jemand unbefugtes diesen erfährt. Demnach ist es ein asymmetrisches Verfahren zum Austauschen eines geheimen Schlüssels, welches außerdem die Grundlage für das Public-Key-Verschlüsselungsverfahren von Herrn elGamal darstellt.[4] Es beruht auf mathematischen Berechnungen, die es möglich machen, einen Schlüssel leicht zu verschlüsseln, es aber fast unmöglich machen, diesen ohne bestimmte Informationen wieder zu entschlüsseln. [5]

3.1 Geschichte

Martin Hellman, Whitfield Diffie und Ralph Merkle waren die ersten, die mit dem DH-Schlüsselaustausch, oder auch Diffie-Hellman-Merkle-Schlüsselaustausch, ein asymmetrisches Schlüsselaustauschverfahren veröffentlicht haben. Seit 1976 war es mit diesem möglich, geheime Schlüssel über offene Kanäle und über weite Strecken hinweg auszutauschen, ohne dass ein Dritter diese erfährt.[6] Mit diesem Verfahren

[3] (Schnabel, *kein Datum*)
[4] (Buchmann, 2016)
[5] (Schnabel, *kein Datum*)
[6] (Hellman, 1976)

war es damals zwar noch nicht möglich geheime Textnachrichten asymmetrisch verschlüsselt zu verschicken, jedoch konnte der gemeinsame Schlüssel genutzt werden, um beispielsweise Texte symmetrisch zu verschlüsseln. Die Anerkennung für das erste asymmetrische Verschlüsselungsverfahren konnten Sie allerdings nur bis 1997 für sich beanspruchen, da in dem Jahr öffentlich wurde, dass schon 1973 die erste asymmetrische Verschlüsselung für die englische Regierung entwickelt wurde, welche dem heutigen RSA-Verfahren sehr ähnelt. Diese blieb aber bis dato unter Verschluss. [7]

3.2 Prinzip

Beim DH-Schlüsselaustausch wird ein öffentlicher Schlüssel (p) zwischen zwei Kommunikationspartnern über einen unsicheren Kanal zusammen mit einem Generator (g) gesendet. Beide besitzen einen geheimen Schlüssel (i_a und i_b), mit dem sie eine in eine Richtung leicht zu berechnende Berechnung durchführen und nur das Ergebnis miteinander teilen (A und B), um dann anschließend jeder den gemeinsamen geheimen Schlüssel (K) berechnen zu können. Im Endeffekt wird sich zeigen, dass beide die gleiche Berechnung durchführen, nur in verschiedener Reihenfolge, sodass sie zwangsläufig auf das gleiche Ergebnis, den geheimen Schlüssel, kommen mussten. Das Prinzip und auch die Sicherheit dieses Schlüsselaustausches basieren auf drei Elementen.[8]

1. eine Einwegfunktion
2. eine endliche zyklische Gruppe über einer Primzahl p
3. ein Generator g, der eine Primitivwurzel Modulo p der zyklischen Gruppe ist

3.2.1 Einwegfunktion

Die Formeln des DH-Schlüsselaustausches sind bekannt als Einweg-Funktionen. Die Stärke einer Einwegfunktion basiert auf der Zeit, die es kostet, sie zurückzuverfolgen bzw. zurückzurechnen.

[7] (Unbekannt, DH-Schlüsselaustausch, 2018)
[8] (Schmeh, 2016)

Generell bei allen asymmetrischen Verschlüsselungsverfahren wählt man eine Funktion, die einfach zu berechnen ist, aber sehr schwer wieder umzukehren ist.[9] Die diskrete Exponentialfunktion ist eine sehr häufig angewandte Einwegfunktion, welche auch beim DH-Schlüsselaustausch genutzt wird.

$g^i \bmod p$

Sie liefert den Rest bei Division von g^i durch p. Mit „diskret" ist in diesem Zusammenhang ganzzahlig gemeint.[10] Der diskrete Logarithmus ist die Umkehrfunktion der diskreten Exponentialfunktion.[11]

Die Exponentialfunktion ist selbst bei sehr großen Zahlen einfach und exakt zu berechnen, jedoch gibt es bis heute keinen effizienten Algorithmus den Exponenten i zu berechnen, wenn g und p und das berechnete Ergebnis von $g^i \bmod p$ gegeben ist. Es gibt schlichtweg keine Darstellung mittels einer Formel der diskreten Logarithmusfunktion, um die Berechnung umzukehren.[12] Ein Beispiel des Prinzips des DH-Schlüsselaustausches zur Veranschaulichung der Problematik kann mit Farben gut erklärt werden (siehe Abbildung 1). Alice und Bob sind zwei Kommunikationspartner und möchten beide die gleiche Farbe mischen, ohne dass jemand anderes diese Farbe weiß oder nachmischen kann.[13] Dazu gelten die gleichen Regeln wie für die mathematischen Funktionen, nur dass sie ggf. einfacher nachzuvollziehen sind:

- Es ist einfach zwei Farben miteinander zu mischen
- Es ist praktisch unmöglich diese wieder in die beiden Ausgangsfarben zu entmischen.

[9] (Beutelspacher, Schwenk, Wolfenstetter, 2015)
[10] (Unbekannt, diskrete Exponentialfunktion, 2018)
[11] (Unbekannt, diskrete Logarithmusfunktion, 2018)
[12] (Unbekannt., *kein Datum*)
[13] (Cruise, 2012)

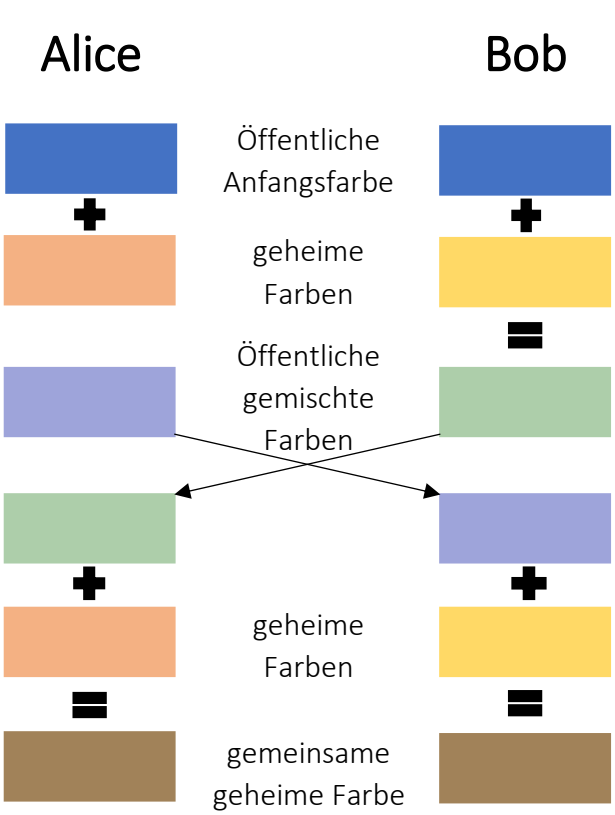

Abbildung 1: Prinzip des DH-Schlüsselaustausches

Alice mischt ihre geheime Farbe mit der öffentlichen Anfangsfarbe und sendet sowohl die öffentliche Anfangsfarbe als auch die gemischte öffentliche Farbe an Bob. Die geheime Farbe behält sie für sich. Bob mischt jetzt die öffentliche Anfangsfarbe mit seiner eigenen geheimen Farbe und sendet Alice seine gemischte öffentliche Farbe zu. Außenstehende können weder mit dem gemischten Violett von Alice noch mit dem gemischten Grün von Bob herausfinden, welche geheimen Farben sie haben, da es praktisch unmöglich ist, eine gemischte Farbe wieder in die Ausgangsfarben zu entmischen. Sie könnten höchstens versuchen durch ausprobieren die richtige Farbe mit dem Blau, der Anfangsfarbe, zu mischen. Alice und Bob haben dagegen alles was sie brauchen, um beide die geheime gemeinsame Farbe zu mischen. Alice mischt dafür ihre geheime rosa Farbe mit Bobs öffentlicher gemischten Farbe, dem Grün, und Bob mischt seine geheime gelbe Farbe mit der öffentlichen gemischten Farbe von Alice, dem Violett. Dadurch erhalten beide das gleiche Braun, ohne die geheime Farbe des anderen kennen zu müssen.

Wieder in Zahlen ausgedrückt bedeutet das demnach für eine Einwegfunktion, die diskrete Exponentialfunktion, dass der Exponent i, im Farbbeispiel die geheime Farbe, nur durch Ausprobieren und Erstellen einer Wertetabelle ermitteln werden kann. Wenn $3^i \bmod 7 = 6$ wäre, könnten wir also nur durch ausprobieren herausfinden, dass $i = 3$ ist. Bei größeren Zahlen wie Beispielsweise $453^i \bmod 21997 = 355$ wird das schon aufwendiger. Die heute verwendeten Zahlen für Verschlüsselungen bringen selbst

leistungsstarke Computer an ihre Grenzen. Um beim DH-Schlüsselaustausch den geheimen gemeinsamen Schlüssel zu berechnen, bzw. die geheime gemeinsame Farbe mischen zu können, braucht man aber mindestens eine der beiden geheimen Farben, also einen geheimen Exponenten i. Hier stößt man folglich auf das diskrete Logarithmusproblem.

3.2.2 endliche zyklische Gruppen

In der Kryptografie sind Primzahlen für p besonders interessant, da sie für alle Zahlen zwischen 1 und p – 1 ein inverses Element (mod p) haben, da alle Zahlen zwischen 1 und p – 1 teilerfremd zu p sind. Mit Hilfe von Primitivwurzeln lässt sich diese Verknüpfung herstellen, sodass eine endliche zyklische Gruppe mit allen Zahlen von 1 bis p – 1 gebildet werden kann. Endliche zyklische Gruppen müssen nicht alle Zahlen von 1 bis p – 1 enthalten, das ist nur der Fall wenn man eine Primitivwurzel zur Erstellung verwendet und wenn p eine Primzahl ist. In dem Fall wird die endliche zyklische Gruppe auch als prime Restklassengruppe und die dazugehörige Primitivwurzel als Erzeuger dieser bezeichnet. Endliche zyklische Gruppen enthalten jedoch ausschließlich Zahlen, die zwischen 1 und p – 1 liegen.[14]

3.2.3 Generator g

Ein Generator g sollte möglichst eine Primitivwurzel Modulo p sein. Dies ist der Fall, wenn folgende Bedingung, was gleichzeitig auch die Aufgabe einer Primitivwurzel ist, erfüllt ist.

$g^i \bmod p = \{1, 2, \ldots, p\text{-}1\}$ mit $i = \{1, 2, \ldots, p\text{-}1\}$

Denn das bedeutet, dass mit einer Primitivwurzel alle Zahlen von 1 bis p -1 als Rest der Funktion $g^i \bmod p$ dargestellt werden können, wenn für i alle Zahlen von 1 bis p – 1 eingesetzt werden. Diese stellen dann zusammen die größtmögliche endliche zyklische Gruppe dar. Dadurch ist gleichzeitig die sogenannte Ordnung des Generators so hoch wie es nur möglich ist, und zwar ist die Ordnung = p – 1. Die Ordnung ist im Allgemeinen gleich der Anzahl an Elementen in einer zyklischen Gruppe. Sobald eine 1 als Rest errechnet wird hat man die Ordnung der zyklischen Gruppe bestimmt,

[14] (Schmeh, 2016)

i	g = 3	g = 4
1	3	4
2	9	16
3	8	7
4	5	9
5	15	17
6	7	11
7	2	6
8	6	5
9	18	1
10	16	4
11	10	16
12	11	7
13	14	9
14	4	17
15	12	11
16	17	6
17	13	5
18	1	1

Abbildung 2: zyklische Gruppen mit und ohne Primitivwurzel

indem man die berechneten Elemente zählt.[15] In dem Beispiel aus Abbildung 2, wenn die Primzahl 19 ist und g einmal gleich 3 und einmal gleich 4 ist, dann werden, wenn für i die Werte 1 bis p-1 einsetzt werden, nebenstehende Werte berechnet. Die zyklische Gruppe mit dem Generator 3 ist also {1, 2, 3, 4, 5, 6, 7, 8, 9, 10, 11, 12, 13, 14, 15, 16, 17, 18} mit der Ordnung 18 wohingegen die zyklische Gruppe mit dem Generator 4 nur die Elemente {1, 4, 5, 6, 7, 9, 11, 16, 17} besitzt und somit nur die Ordnung 9 hat. Welche Bedeutung für die DH-Verschlüsselung hat nun die Ordnung einer zyklischen Gruppe? Für die Berechnung des gemeinsamen Schlüssels ist es irrelevant welcher geheime Schlüssel verwendet wird, solange mit dem geheimen Schlüssel das gleiche Element aus der zyklischen Gruppe berechnet wird, wie das eines der beiden Kommunikationspartner. D.h. wenn der Rest A = 4, der mit der Formel errechnet wurde bekannt ist, welcher öffentlich verschickt wird, dann ist die Chance durch ausprobieren ebenfalls eine 4 mit einsetzen von i zu erhalten bei g = 4 doppelt so hoch und damit doppelt so schnell wie bei g = 3, da die zyklische Gruppe nur halb so groß ist. Ob, wenn g = 4 ist, als geheimer Schlüssel demnach eine 1 oder eine 10 verwendet wird ist egal, beide Male wird der gleiche geheime gemeinsame Schlüssel errechnet. Bei der Ermittlung des Schlüssels wird dieses Problem noch einmal erläutert. Es steht nichtsdestotrotz fest, Primitivwurzeln sind die Wahl, um sicherzustellen, dass die maximale Anzahl an i ausprobiert werden muss, um auf das richtige Ergebnis für A oder B zu kommen.

[15] (Scharlau, 2012)

3.2.4 Ermittlung des Schlüssels

Wenn Alice und Bob sich jetzt für den öffentlichen Schlüssel auf unser Beispiel die 19 einigen und den Generator 3, da es eine Primitivwurzel Modulo 19 ist, dann kommt folgende Rechnung zustande:

Alice entscheidet sich bspw. für den geheimen Schlüssel $i_a = 4$ aus dem für i zuvor definierten Bereich und berechnet mit p = 19 und g = 3 ihr A.

$A = g^{i_a} \bmod p$ also $A = 3^4 \bmod 19 \rightarrow$ $A = 5$

Alice gibt ihr errechnetes A zusammen mit p = 19 und g = 3 an Bob weiter. Dessen zufällig gewählter geheimer Schlüssel ist $i_b = 5$, womit er mit Hilfe von p und g gleichermaßen B bestimmt.

$B = g^{i_b} \bmod p$ also $B = 3^5 \bmod 19 \rightarrow$ $B = 15$

Das errechnete B sendet er jetzt ebenfalls an Alice. Beide kennen nun die Werte für A, B, p und g und haben jeweils ihren geheimen Schlüssel. Die Voraussetzungen, den geheimen Schlüssel berechnen zu können ist demnach erfüllt.

Alice berechnet den geheimen Schlüssel folgendermaßen.

$K = B^{i_a} \bmod p$ also $K = 15^4 \bmod 19 \rightarrow$ $K = 9$

Bob berechnet den geheimen Schlüssel folgendermaßen.

$K = A^{i_b} \bmod p$ also $K = 5^5 \bmod 19 \rightarrow$ $K = 9$

Zur besseren Verdeutlichung wird in Abbildung 3 dargestellt, welche Parameter von wem berechnet, ermittelt und übertragen werden.

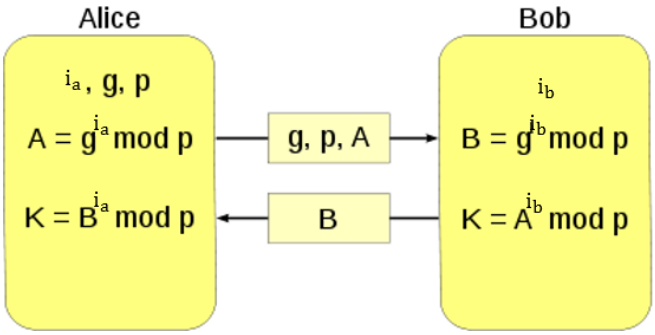

Abbildung 3: Kommunikation beim DH-Schlüsselaustausch

Nun können Alice und Bob den Schlüssel bspw. als Sitzungsschlüssel für ein symmetrisches Verfahren verwenden.

Ein Angreifer oder Zuschauer, der den geheimen Schlüssel von Alice und Bob herausfinden möchte steht vor dem Problem, dass er zwar die berechneten Werte von g^{i_a} und g^{i_b} kennt, jedoch muss er, um den Schlüssel berechnen zu können wissen, welchen Wert i_a oder i_b hat. Um dieses Problem zu lösen muss das diskrete Logarithmusproblem gelöst werden. Die Sicherheit gegen einen Angreifer hängt dabei jedoch auch ausschlaggebend von der Größe der gewählten Primzahl ab. Außerdem ist es ratsam eine sogenannte Sophie-Germain-Zahl zu verwenden. Dies ist eine Primzahl bei der (p-1)/2 ebenfalls eine Primzahl ergibt. Sie gelten als besonders sicher. Bei der Wahl des Generators ist es nur wichtig, dass die Zahl eine Primitivwurzel modulo p darstellt.

Warum Alice und Bob im Endeffekt die gleiche Zahl als geheimen gemeinsamen Schlüssel berechnen liegt daran, dass sie beide die gleiche Rechnung, nur in einer anderen Reihenfolge gemacht haben. Durch Einsetzen der Gleichungen, die für die Berechnung der öffentlichen Schlüssel verwendet werden, in die Gleichungen zur Schlüsselberechnung erkennt man, dass die Berechnungen tatsächlich gleich sind:

$$K = B^{i_a} \bmod p = (g^{i_b} \bmod p)^{i_a} \bmod p = g^{i_b i_a} \bmod p$$

$$K = A^{i_b} \bmod p = (g^{i_a} \bmod p)^{i_b} \bmod p = g^{i_a i_b} \bmod p$$

$$g^{i_a i_b} \bmod p \quad = \quad g^{i_b i_a} \bmod p$$

Für Alice und Bobs Berechnungen ist nur relevant, dass beide sowohl die versendete Zahl als auch den Schlüssel jeweils einmal mit der geheimen Zahl berechnet haben,

sodass sie die gleiche Rechnung nur in einer anderen Reihenfolge gemacht haben, ohne ihre geheime Zahl preisgeben zu müssen.

Um auf das Problem zurück zu kommen, wenn keine Primitivwurzel modulo p als Generator verwendet wird, kann das Beispiel mit p = 19 und g = 4 verwendet werden, da g kein Erzeuger von p ist.

Alice entscheidet sich bspw. für den geheimen Schlüssel i_a = 3 und berechnet gleichermaßen wie vorhin mit p = 19 und g = 4 ihr A.

$A = g^{i_a} \bmod p$ also $A = 4^3 \bmod 19 \rightarrow$ $A = 7$

Sie ist sich dabei nicht im Klaren, dass auch mit i = 11 die Zahl 7 für A hätte berechnet werden können.

Alice gibt daher unwissend das errechnete A zusammen mit p und g an Bob weiter. Dessen zufällig gewählter geheimer Schlüssel ist i_b = 8, womit er mit Hilfe von p und g gleichermaßen B bestimmt.

$B = g^{i_b} \bmod p$ also $B = 4^8 \bmod 19 \rightarrow$ $B = 5$

Auch Bob ist sich nicht bewusst, dass die 5 für B genauso gut mit i = 17 hätte berechnet werden können.

Daher schickt er sein B nichtsahnend an Alice. Beide berechnen nun Ihren Schlüssel.

Alice berechnet den geheimen Schlüssel folgendermaßen.

$K = B^{i_a} \bmod p$ also $K = 5^3 \bmod 19 \rightarrow$ $K = 11$

Hier führt sich das Problem fort, sodass auch mit i = 11 K berechnet werden würde.

Bob berechnet den geheimen Schlüssel folgendermaßen.

$K = A^{i_b} \bmod p$ also $K = 7^8 \bmod 19 \rightarrow$ $K = 11$

Wobei auch in Bobs Gleichung die 17 für i eingesetzt zum Ziel K = 11 geführt hätte.

Ein außenstehender müsste also statistisch gesehen weniger Werte für i ausprobieren, um auf den geheimen gemeinsamen Schlüssel zu kommen, nur weil der Generator

keine Primitivwurzel ist. Da es beim Generator g = 3 für jedes i verschiedene Werte gibt, müsste ein Angreifer im Beispiel alle 18 Optionen ausprobieren.

3.3 Schlüsselaustausch mit mehr als zwei Partnern

Der DH-Schlüsselaustausch kann grundsätzlich mit beliebig vielen Kommunikationspartnern erfolgen. Dafür muss jeder Teilnehmer einen öffentlichen Schlüssel erhalten, der je ein Mal von jedem Kommunikationspartner mit seinem geheimen Schlüssel verschlüsselt wurde, um mit diesem öffentlichen Schlüssel und seinem eigenen geheimen Schlüssel dann den gemeinsamen geheimen Schlüssel berechnen zu können. Als Beispiel wollen Alice, Bob und Carol einen gemeinsamen Schlüssel generieren.[16]

Dafür wird wieder eine zyklische Gruppe über einer Primzahl p mit Hilfe eines Generators g erstellt. Jeder der drei Kommunikationspartner überlegt sich jetzt wie gewohnt eine geheime Zahl zwischen 1 und p-1 also i_a, i_b, und i_c, und Alice berechnet dann ihren öffentlichen Schlüssel A mit Hilfe von i_a und sendet diesen an Bob. Bob berechnet sein B, um es Carol schicken zu können und Carol sendet ihr berechnetes C an Alice.

Alice sendet Bob $A = g^{i_a} \bmod p$

Bob sendet Carol $B = g^{i_b} \bmod p$

Carol sendet Alice $C = g^{i_c} \bmod p$

Wie oben erwähnt müssen diese öffentlichen Schlüssel jetzt noch einmal mit einem geheimen Schlüssel verschlüsselt werden, damit alle zusammen einen gemeinsamen Schlüssel berechnen können. Dafür verschlüsselt Alice das B, Bob verschlüsselt das A und Carol verschlüsselt das C.

Alice sendet Bob $C' = C^{i_a} \bmod p$

Bob sendet Carol $A' = A^{i_b} \bmod p$

[16] (Eilers, 2016)

Carol sendet Alice B' = B^{i_c} mod p

Jetzt besitzt jeder einen öffentlichen Schlüssel, der von allen anderen Kommunikationspartnern einmal verschlüsselt wurde und jeder kann damit den gemeinsamen geheimen Schlüssel berechnen.

Alice berechnet K = B' ^{i_a}mod p

Bob berechnet K = C' ^{i_b}mod p

Carol berechnet K = A' ^{i_c}mod p

Wenn man die Gleichungen wieder auflöst wie zuvor erkennen wir, dass alle erneut die gleiche Rechnung gemacht haben, nur in anderer Reihenfolge und die Gleichung des Schlüssels lässt sich wie folgt darstellen:

K = $g^{i_a i_b i_c}$ mod p

Bei größeren Gruppen wird dieses Verfahren möglicherweise aufwendig, da jeder einen öffentlichen Schlüssel braucht, der von all seinen Kommunikationspartnern einmal verschlüsselt wurde. Bei 20 Kommunikationspartnern bedeutet das dann, dass jeder 19 Berechnungen machen muss, bevor ein gemeinsamer geheimer Schlüssel berechnet werden kann. Dies erfolgt reihum und kostet daher mehr Zeit.

Daher empfielt es sich kleinere Runden zu bilden, die einen Schlüssel generieren, damit diese Schlüssel dann als geheime Schlüssel verwendet werden können, um einen gemeinsamen geheimen Schlüssel zu erzeugen.[17] Wenn also bspw. eine Gruppe von 20 Personen in Fünfergruppen eingeteilt werden, muss jeder nur sieben statt 19 Verschlüsselungen machen, bevor sie den geheimen Schlüssel berechnen können. Denn in der Fünfergruppe muss jeder vier mal einen öffentlichen Schlüssel berechnen und um dann mit dem Schlüssel einen globalen Schlüssel der 20er-Gruppe zu generieren muss jede Fünfergruppe drei öffentliche Schlüssel berechnen, da es ja vier Gruppen sind.

[17] (Hellman, 1976)

4. Das elGamal-Verschlüsselungsverfahren

Das elGamal-Verfahren wurde 1985 von Taher elGamal entwickelt und ist ein Public-Key-Verfahren, demnach ein Verschlüsselungsverfahren bei dem jeder Teilnehmer sowohl einen öffentlichen als auch einen privaten Schlüssel besitzt mit deren Hilfe Nachrichten asymmetrisch verschlüsselt ausgetauscht werden können. Es basiert auf dem Prinzip des DH-Schlüsselaustausches, wobei es eine Veränderung im Protokoll, dem Ablauf der mathematischen Berechnungen, aufweist, was es möglich macht nicht nur Schlüssel gemeinsam zu berechnen, sondern auch Nachrichten sicher verschicken zu können. Das Verfahren erreichte nie eine ähnlich starke Verbreitung wie das DH-Verfahren, da es aber keinem Patent unterliegt ist es besonders in Open-Source-Projekten beliebt.

4.2 Prinzip

Beim elGamal-Verfahren ist das Protokoll an einer Stelle zum DH-Verfahren verändert, was einem Sender ermöglicht, eine Nachricht mit Hilfe des öffentlichen Schlüssel eines Empfängers zu verschlüsseln, sodass der Empfänger die verschlüsselte Nachricht mit Hilfe seines geheimen Schlüssels und einer Funktion wieder entschlüsseln kann. Dabei muss kein gemeinsamer geheimer Schlüssel erstellt werden.

Als Beispiel möchte Bob Alice eine verschlüsselte Nachricht zukommen lassen. Bis zum Austausch der öffentlichen Schlüssel wird das Protokoll des DH-Schlüsselaustausches auch nicht verändert. Das bedeutet, dass Alice wie zuvor eine große Primzahl wählt, als vereinfachtes Beispiel sei p = 23, und generiert einen Erzeuger einer primen Restklassengruppe g = 7. Mit diesen Parametern und einem geheimen Schlüssel, der die Bedingung i ϵ {1, …, p-2} erfüllen muss, berechnet sie ihr A.

$i_a = 6 \quad \epsilon \{1, \ldots, p-2\}$

$A = g^{i_a} \bmod p$ also $A = 7^6 \bmod 23 \rightarrow A = 4$

Bob erhält A, p und g von Alice, sodass er wie im DH-Verfahren sein B berechnen kann. Dafür überlegt er sich ebenfalls einen geheimen Schlüssel.

$i_b = 3 \quad \epsilon \{1, \ldots, p-2\}$

$B = g^{i_b} \bmod p$ also $B = 7^3 \bmod 23 \rightarrow B = 21$

Jetzt verschlüsselt er seinen Klartext m, den er Alice schicken möchte, indem er die ggf. vorher in Zahlen kodierte Nachricht mit der diskreten Exponentialfunktion multipliziert und dafür den öffentlichen Schlüssel von Alice, seinen geheimen Schlüssel und die Primzahl p verwendet.

m = 7

$c = A^{i_b} * m \bmod$ also $c = 4^3 * 7 \bmod 23 \rightarrow c = 11$

Nach Erhalt von B und c kann Alice mit ihrem geheimen Schlüssel und folgender Formel die Nachricht von Bob entschlüsseln.

$m = B^{p-1-i_a} * c \bmod p$ also $m = 21^{23-1-6} * 11 \bmod 23 \rightarrow m = 7$

Einem Außenstehenden wäre i_a unbekannt, er scheitert am diskreten-Logarithmus-Problem, wenn er versucht, von A auf i_a zu schließen. Ohne i_a kann er die Formel für m jedoch nicht anwenden.[18]

Um die Formel für m zu beweisen wird der kleine Satz von Fermat verwendet. Dieser besagt, dass eine beliebige Zahl d hoch die Primzahl p minus 1 geteilt durch p immer den Rest 1 ergibt. Wenn d jetzt außerdem kein Vielfaches von p ist gilt die Formel:

$d^{p-1} \bmod p = 1$

In Folgenden wird die Gleichung für m bewiesen:[19]

$X = p - 1 - i_a$

I: $B^X = B^{p-1-ia}$ | $B = g^{ib} \bmod p$

$= (g^{ib} \bmod p)^{p-1-ia}$

$= g^{ib\,(p-1-ia)} \bmod p$

[18] (Lang, 2016)
[19] (Menezes, van Oorschot, Vanston, 1996)

$= g^{ib\,(p-1)}\, g^{-\,iaib} \bmod p$ | $g^{p-1} \bmod p = 1$ (Fermat)

$= g^{-\,iaib} \bmod p$

II: $m = B^X c \bmod p$ | $B^X = g^{-\,iaib} \bmod p$

| $c = A^{ib}\, m \bmod p$

$= ((g^{-\,iaib} \bmod p)(A^{ib}\, m \bmod p)) \bmod p$

$= (g^{-\,iaib} \bmod p)(A^{ib}\, m \bmod p)$

$= g^{-\,iaib}\, A^{ib}\, m \bmod p$ | $A = g^{ia} \bmod p$

$= g^{-\,iaib}\, (g^{ia} \bmod p)^{ib}\, m \bmod p$

$= g^{-\,iaib}\, g^{\,iaib} \bmod p\; m \bmod p$

$= m \bmod p \bmod p$

$= m \bmod p$

$= m$

Wie hier im letzten Schritt erkennbar ist, muss die Bedingung $m < p - 1$ erfüllt sein, damit die Gleichung stimmt. Daher müssen Nachrichten ggf. unterteilt werden, sodass gewährleistet ist, dass die in Zahlen kodierten Buchstaben kleiner als die verwendete Primzahl sind.[20]

5. Sicherheit und Unsicherheit beider Verfahren

Die Sicherheit einer Verschlüsselung wird an der Zeit gemessen, die man braucht, sie zu entschlüsseln.

5.1 DH-Schlüsselaustausch

Die Sicherheit des DH-Schlüsselaustausches wird grundsätzlich von den mathematischen Elementen, die in 3.2 erläutert wurden gewährleistet. Dies bedeutet, dass das diskrete Logarithmusproblem zum einen für Sicherheit sorgt und gleichzeitig

[20] (Spitz, Pramateftakis, Swoboda, 2011), (ElGamal, 1985)

prime Restklassengruppen mit Primitivwurzeln als Erzeuger für genügend Sicherheit benötigt werden.

Als weitere Sicherheit sollten nur Sophie-Germain-Primzahlen verwendet werden, da sie als besonders sicher gelten.

5.1.1 DH-Problem

Das DH-Problem ist eng mit dem diskreten Logarithmusproblem verbunden. Es gibt eine Unterteilung in zwei Problemstellungen. Beim Computational-Diffie-Hellman-Problem (CDH-Problem) wird die Frage gestellt, wenn ein Element g einer Gruppe und die Werte A = g^{i_a} und B = g^{i_b} gegeben sind, welchen Wert hat dann K = $g^{i_a i_b}$ mit i_a, i_b unbekannt? Wer diskrete Logarithmen mod p berechnen kann, ist in der Lage das CDH-Problem zu lösen. Solange das CDH-Problem jedoch nicht in vertretbarer Zeit lösbar ist, gilt der DH-Schlüsselaustausch als sicher. Der zweite Teil, das Decisional-Diffie-Hellman-Problem (DDH-Problem) besagt, dass wenn (g, g^{i_a}, g^{i_b}, $g^{i_a i_b}$) öffentlich ausgetauscht werden, ein Angreifer oder Zuschauer nicht erkennen kann, ob $g^{i_a i_b}$ aus den anderen Zahlen berechnet werden kann oder ob es nicht eine neue Zahl g^{i_c} ist, da die Zahlen alle stark zufällig aussehen. Das Problem besteht also darin, bei gegebenem $g^{i_a} \bmod p$, $g^{i_b} \bmod p$ und $g^{i_c} \bmod p$ zu entscheiden, ob $g^{i_c} = g^{i_a i_b}$ ist. Wenn $g^{i_c} = g^{i_a i_b}$ ist, dann liegt ein sogenanntes Diffie-Hellman-Tripel vor.

Wegen folgendem Theorem kann bei einer Auswahl von g als Primitivwurzel das DDH-Problem angegriffen werden:

„Sei p eine Primzahl, sei g eine Primitivwurzel modulo p und seien i_a , $i_b \in \{ 0 , \ldots , p - 2 \}$. Dann ist $g^{i_a i_b}$ genau dann ein quadratischer Rest modulo p, wenn g^{i_a} oder g^{i_b} ein quadratischer Rest ist modulo p.“[21]

Z.B. $i_a = 4$, $i_b = 5$, $g = 3$:

$3^4 = 81$ -> $\sqrt{81} = 9$

$3^{4*5} = 3.486.784.401$ -> $\sqrt{3.486.784.401} = 59049$

[21] (Unbekannt, DH-Schlüsselaustausch, 2018)

Denn eine Potenz von g ist genau dann ein quadratischer Rest modulo p, wenn der Exponent gerade ist.[22] Ein Angreifer kann also prüfen, ob das Kriterium aus diesem Theorem erfüllt ist, zwar weiß er nicht genau, ob ein Diffie-Hellman-Tripel vorliegt, er antwortet aber zu 75 % richtig, womit er einen Vorteil von 50% gegenüber reinem Raten hat.

Wenn das CDH-Problem lösbar ist, kann auch das DDH-Problem gelöst werden, da $g^{i_a i_b}$ berechnet werden und mit g^{i_c} verglichen werden kann.

Um sich vor diesem Angriff und auch vor dem lösen des CDH-Problems und damit letztendlich vor dem Lösen der diskreten Logarithmusfunktion zu schützen, sollten sehr große Primzahlen gewählt werden, wodurch die Anzahl an möglichen Schlüsseln erhöht wird, denn heutzutage gelten DH-Schlüsselaustauschverfahren nur als sicher, wenn Schlüssellängen von 2048 Bit verwendet werden. Sobald jedoch Quantenrechner für die Allgemeinheit zugänglich werden, wird dies nicht mehr der Fall sein.

5.1.2 Der Man-In-The-Middle-Angriff

Bei asymmetrischen Verfahren stellt der Man-In-The-Middle-Angriff (MitM-Angriff) einen bedeutenden Faktor da, warum asymmetrische Verfahren nie zu 100% als sicher erachtet werden können.

Hierbei befindet sich ein Angreifer oder Zuhörer auf dem Nachrichtenweg zwischen zwei kommunizierenden Einheiten z.B. zwei echten Menschen, zwei Routern oder zwei Email-Servern, hier sind es Alice und Bob. Der Angreifer, hier Mallory, kann sich als Kommunikationspartner ausgeben und Datenpakete einfügen, verändern und löschen, wodurch er die Integrität der zwischen Alice und Bob ausgetauschten Dateien kompromittiert.[23]

[22] (Buchmann, 2016)
[23] (Kurose, Ross, 2008)

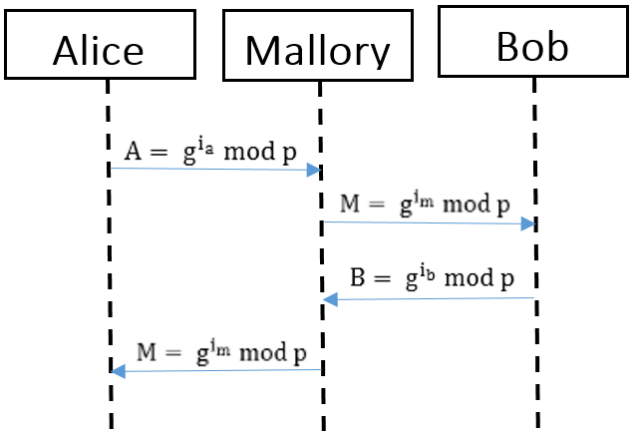

Abbildung 4: Prinzip des MitM-Angriffs

Dafür fängt Mallory die Nachrichten von Alice und Bob ab und sendet einen eigenen öffentlichen Schlüssel, der mit einem beliebigen geheimen Schlüssel erzeugt wurde, zu beiden (siehe Abbildung 4). So führt Mallory je einen Schlüsselaustausch sowohl mit Bob als auch mit Alice durch, ohne dass diese beiden davon wissen. Dadurch ergeben sich folgende Schlüssel:

öffentliche Schlüssel:		geheime Schlüssel:
Alice:	A	i_a
Bob:	B	i_b
Mallory:	M	i_m

Alice: $K_A = M^{i_a} \bmod p = g^{i_a i_m} \bmod p$

Bob: $K_B = M^{i_b} \bmod p = g^{i_b i_m} \bmod p$

Mallory: $K_A = A^{i_m} \bmod p = g^{i_a i_m} \bmod p$

$K_B = B^{i_m} \bmod p = g^{i_b i_m} \bmod p$

Mallory kann dann ggf. symmetrisch verschlüsselte Nachrichten von beiden abfangen und mit dem jeweiligen Schlüssel entschlüsseln. Bevor diese dann an den eigentlichen

Empfänger weitergeleitet werden, werden sie mit dem zugehörigen Schlüssel verschlüsselt, wobei der Angreifer diese Nachricht dabei beliebig verändern kann. [24]

Wegen dem Problem der Authentizität öffentlicher Schlüssel werden Alice und Bob ihre öffentlichen Schlüssel nicht oft austauschen, damit Mallory sich nicht zwischenschalten kann, jedoch wird dadurch der Sitzungsschlüssel immer der gleiche sein. Wenn sie damit viele symmetrisch verschlüsselte Nachrichten senden verfügt der Angreifer über viel Chiffrentext, was es leichter macht, diesen zu entschlüsseln. Beim elGamal-Verfahren tritt dieses Problem nicht auf, da die Nachricht asymmetrisch verschlüsselt wird und jedes Mal ein neuer Sitzungsschlüssel verwendet wird.

Um sich vor dem MitM-Angriff trotzdem möglichst gut zu schützen gibt es einige Möglichkeiten.

Beispielsweise kann ein öffentlicher Schlüssel authentifiziert werden, indem ein vertrauenswürdiger Dritter mit einbezogen wird. Hierbei gibt es zwei Varianten. Die erste Variante ist eine unabhängige Autorität, der von allen Kommunikationspartnern vertrauen wird, welche öffentliche Schlüssel verifizieren kann. Diese wird auch als Certificate Authority (CA) bezeichnet. Um sicherzustellen, dass jemand derjenige ist, für den er sich ausgibt, wird dabei der öffentliche Schlüssel mit Hilfe einer digitalen Signatur verifiziert und wenn ein Kommunikationspartner abklären möchte, ob der empfangene Schlüssel der richtige ist, kann dieser von der CA abgeglichen werden. Die andere Variante ist ein Netzwerk aus sich gegenseitig vertrauenden Personen, die in der Lage sind, den öffentlichen Schlüssel untereinander zu verifizieren. Dieses Netzwerk wird Web of Trust genannt (WoT). [25]

Auch durch eine IP-Standortermittlung ist es einem Kommunikationspartner möglich, einen Angreifer zu enttarnen, dafür muss er wissen, wo sich der echte Kommunikationspartner befindet, sodass er den Standort von der IP-Standortermittlung mit seinem Wissen abgleichen kann. Denn wenn sich bspw. Alice in Hamburg befindet, sich bei der IP-Standortermittlung aber herausstellt, dass sich

[24] (Ertel, 2012)
[25] (Czeschik, Lindhorst, Jehle, 2015)

der Kommunikationspartner in Shanghai aufhält, sicher ist, dass nicht mit der richtigen Person kommuniziert wird. Hierbei könnte ein VPN-Server dem Angreifer einen Vorteil bieten, wenn er selbst den Standort von dem Kommunikationspartner, für den er sich ausgeben will, kennt und einen VPN-Zugriff für diesen Standort besitzt.

Es gibt auch die Möglichkeit einer out of Brand Authentification. Dies ist eine Authentifizierung mit zwei unabhängigen Kanälen. Über einen zweiten Informationskanal wird also geprüft, ob die öffentlichen Schlüssel übereinstimmen. Im Bankenwesen wird z.B. die TAN als Sicherheitscode übers Handy geschickt, welches dafür vorher authentifiziert wurde.

All diese Verfahren haben jedoch ein Problem. Jedes Mal müssen die Kommunikationspartner Teile ihrer Anonymität aufgeben, um sicherzustellen, dass sie vor einem MitM-Angriff geschützt sind. Um nicht seine Identität preisgeben zu müssen, aber trotzdem einen MitM-Angriff deutlich zu erschweren, kann z.B. das sogenannte Interlock-Modell verwendet werden, bei dem die Nachricht aufgeteilt wird. Dies ist möglicherweise viel Aufwand, aber dafür ist diese Methode nicht nur sicher, sondern auch anonym.[26]

Am sichersten ist es generell, wenn mehrere Verfahren zur Absicherung gegen einen MitM-Angriff verwendet werden.

5.2 ElGamal-Verfahren

Das elGamal-Verfahren hat dadurch, dass es stark auf dem DH-Schlüsselaustausch aufbaut sehr ähnliche mathematische Problemstellungen und kann auch in ganz ähnlicher Weise angegriffen werden. Hier ist es jedoch besonders wichtig, dass bei jeder verschlüsselten Nachricht, die verschickt wird, einen neuen Schlüssel K verwendet wird. Denn durch eine Known-Plaintext-Attack, wenn der Klartext m zu einem geheimen Text c bekannt ist, lässt sich K berechnen, indem die Formel $k = c \cdot m^{-1} \mod p$ verwendet wird. Daher muss der Sender für jede neue Verschlüsselung

[26] (Ertel, 2012)

eine neue zufällige Zahl wählen, sodass jedes Mal K und B einen anderen Wert haben.[27]

6. Heute genutzte Verschlüsselungsverfahren

Das DH-Schlüsselaustauschverfahren und das elGamal-Verfahren werden beide häufig dafür verwendet, Sitzungsschlüssel oder Hashes auszutauschen, z.B. beim https Protokoll werden sie unter anderem dafür genutzt. Am häufigsten werden asymmetrische Verschlüsselungsverfahren jedoch in hybrider Verschlüsselung vorgefunden, wobei ein Schlüssel mit Hilfe der asymmetrischen Verschlüsselung erzeugt wird, der dann zur symmetrischen Verschlüsselung von Nachrichten verwendet werden kann.[28]

Der DH-Schlüsselaustausch und deshalb auch das elGamal-Verfahren sind seit dem 29.04.1997 nicht mehr patentgeschützt, da zu dem Zeitpunkt das Patent für den DH-Schlüsselaustausch auslief. Das elGamal-Verfahren ist daher wahrscheinlich der erste patentfreie asymmetrische Algorithmus und demzufolge in Open Source Projekten beliebt.[29]

7. Fazit

So gut wie alle asymmetrischen Verfahren beruhen entweder auf der Faktorisierung von Primzahlen oder aber auf dem Prinzip des diskreten Logarithmus. Daher sind diese Verfahren nur so lange sicher, bis eine Vereinfachung der Umkehrung der Berechnungen gefunden wird. Beim DH-Schlüsselaustausch müsste man dafür das Diskrete-Algorithmus-Problem lösen. Außerdem werden die heute verwendeten Schlüssellängen nicht mehr ausreichend sein, sobald leistungsstärkere Rechner wie z.B. die Quantencomputer der Allgemeinheit zur Verfügung stehen, da diese für die Entschlüsselung nur noch einen Bruchteil an Zeit benötigen werden als die heute bekannte Technik.

---

[27] [Lang, 2016]
[28] (Schnabel, *kein Datum*)
[29] (Wobst, 2001)

Aber weil das Internet, wie am Anfang der Seminararbeit erwähnt wurde, unsicher entworfen wurde, ist es wichtig, sich auch zukünftig gegen Angreifer schützen zu können und die Möglichkeit zu haben, anonym Nachrichten verschlüsselt auszutauschen. Gerade in der heutigen Zeit der Digitalisierung, bei der die Gefahr des gläsernen Menschen realer erscheint als je zuvor. Dazu muss nicht nur die Weiterentwicklung der Public-Key-Systeme weiter vorangetrieben werden. Es sollten auch mehr Verschlüsselungen von „ganz normalen“ Personen genutzt werden, die vielleicht nicht in erster Linie daran denken, dass ihre Daten abgegriffen werden könnten. Dazu wird auch häufig nicht bedacht, dass sensible Daten auch dadurch gesichert werden, indem selbst ganz unwichtige Daten verschlüsselt versendet werden, um Angreifern die wichtigen Daten nicht durch eine Verschlüsselung kenntlich zu machen.

8. Literaturverzeichnis

[Beutelspacher, Schwenk, Wolfenstetter, 2015] Beutelspacher, A., Schwenk, J.,Wolfenstetter, K.D., (2015). *Moderne Verfahren der Kryptographie: Von RSA zu Zero-Knowledge*. 8. Aufl. Wiesbaden, Deutschland: Springer-Verlag. S. 14.

[Blumenthal, 2001] M. Blumenthal, D. Clark. (2001). „Rethink the Design oft he Interne: the End-to-end Arguments vs. the Brave New Worlds“. *ACM Transactions on Internet Technology*. Cambridge: MIT Press. S. 70 – 109.

[Buchmann, 2016] Buchmann, J.. (2016). *Einführung in die Kryptographie.* 5. Aufl. Darmstadt, Deutschland: Springer Verlag. [Online]

Retrieved from:

https://books.google.de/books?id=xZEFDAAAQBAJ&printsec=frontcover&hl=de#v=onepage&q&f=false

[Zugriff am 10.05.2018]

[Cruise, 2012] Cruise, B., (2012). *Public Key Cryptography: Diffie-Hellman Key Exchange (short version)*. [Online]

Retrieved from:

https://www.youtube.com/watch?v=3QnD2c4Xovk&feature=youtu.be

[Zugriff am 06.05.2018]

[Czeschik, Lindhorst, Jehle, 2015] Czeschik, J. C., Lindhorst, M., Jehle, R. (2015). *Gut gerüstet gegen Überwachung im Web: Wie Sie verschlüsselt mailen, chatten und surfen*. [Online]

Retrieved from:

https://books.google.de/books?id=MvnrCgAAQBAJ&pg=PP55&dq=pers%C3%B6nliche+pr%C3%BCfung+%C3%B6ffentlicher+schl%C3%BCssel&hl=de&sa=X&ved=0ahUKEwir2d2EmJ_bAhUsiKYKHbiUDKUQ6AEIMzAC#v=onepage&q=pe

rs%C3%B6nliche%20pr%C3%BCfung%20%C3%B6ffentlicher%20schl%C3%BCssel&f=false

[Zugriff am 05.06.2018]

[Eilers, 2016] Eilers, C., (2016). *Verfahren der Kryptographie, Teil 12: Der Diffie-Hellman-Schlüsselaustausch.* Ceilers. [Online]

Retrieved from:

https://www.ceilers-news.de/serendipity/798-Verfahren-der-Kryptographie,-Teil-12-Der-Diffie-Hellman-Schluesselaustausch.html

[Zugriff am 20.05.2018]

[ElGamal, 1985] ElGamal, T., (1985). *A public key cryptosystem and a signature scheme based on discrete logarithms*. [Online]

Retrieved from:

https://people.csail.mit.edu/alinush/6.857-spring-2015/papers/elgamal.pdf

[Zugriff am 20.05.2018]

[Ertel, 2012] Ertel, W., (2012). *Angewandte Kryptographie*. 4. Aufl. München: Carl Hanser Verlag GmbH Co KG. S. 90

[Hellman, 1976] Hellman, D. u., 1976. *New Directions in Cryptography,* Stanford University: IEEE Transactions on Information Theory.22. [Online]

Retrieved from:

https://ee.stanford.edu/~hellman/publications/24.pdf

[Zugriff am 10.05.2018]

[Kurose, Ross, 2008] Kurose, James F., Ross, Keith W.. (2008). *Computernetzwerke: Der Top-Down-Ansatz.* München: Pearson Deutschland GmbH. S. 83.

[Lang, 2016] Lang, H.W., (2016). *ElGamal-Verschlüsselung*. [Online]

Retrieved from:

http://www.inf.fh-flensburg.de/lang/krypto/protokolle/elgamal.htm

[Zugriff am 20.05.2018]

[Menezes, van Oorschot, Vanston, 1996] Menezes , A. J., van Oorschot, P. C., Vanston, S. A., (1996). *Handbook of Applied Cryptography.* Massachusetts: Institute of Technology. S. 286. [Online]

Retrieved from:

http://math.fau.edu/bkhadka/Syllabi/A%20handbook%20of%20applied%20cryptography.pdf

[Zugriff am 04.06.2018]

[Scharlau, 2012] Scharlau, R., (2012). *Zyklische Gruppen und die Ordnung von Elementen.* [Online]

Retrieved from:

http://www.mathematik.tu-dortmund.de/~algebra/Algebra_2012/Skript/algebra_kap2_1.pdf

[Schmeh, 2016] Schmeh, K. (2016). *Kryptografie: Verfahren, Protokolle, Infrastrukturen.* 6. Aufl. Heidelberg, Deutschland: dpunkt.verlag GmbH.

[Schnabel, *kein Datum*] Schnabel, P., *kein Datum*. Asymmetrische Kryptografie (Verschlüsselung). *Elektronik Kompendium.*

Retrieved from:

https://www.elektronik-kompendium.de/sites/net/1910111.htm

[Zugriff am 10.05.2018]

[Spitz, Pramateftakis, Swoboda, 2011] Spitz, S., Pramateftakis, M., Swoboda, J. (2011). *Kryptographie und IT-Sicherheit: Grundlagen und Anwendungen.* Wiesbaden: Springer-Verlag. S.134.

[Unbekannt, Diskreter Logarithmus, 2018] Unbekannt, (2018). *Diskreter Logarithmus*. [Online]

Retrieved from:

https://de.wikipedia.org/wiki/Diskreter_Logarithmus

[Zugriff am 06.05.2018]

[Unbekannt, Diskrete Exponentialfunktion, 2018] Unbekannt, (2018). *Diskrete Exponentialfunktion*. [Online]

Retrieved from:

https://de.wikipedia.org/wiki/Diskrete_Exponentialfunktion

[Zugriff am 06.05.2018]

[Unbekannt, DH-Schlüsselaustausch, 2018] Unbekannt, (2018). *Diffie-Hellman-Schlüsselaustausch.* [Online]

Retrieved from:

https://de.wikipedia.org/wiki/Diffie-Hellman-Schl%C3%BCsselaustausch

[Zugriff am 01.05.2018]

[Wobst, 2001] Wobst, R., (2001). *Abenteuer Kryptologie: Methoden, Risiken und Nutzen der Datenverschlüsselung.* Deutschland: Pearson. S. 176.