WESTFÄLISCHE WILHELMS-UNIVERSITÄT MÜNSTER

Schriftliche Hausarbeit im Rahmen der Ersten Staatsprüfung für das Lehramt für die Sekundarstufe II

Das RSA-Verfahren im Schulfach Informatik

– Methoden der unterrichtlichen Vermittlung –

Verfasst von:
Thimo Engelsmeyer

Themensteller:
Prof. Dr. Marco Thomas
Institut für Didaktik der Mathematik und der Informatik
Fachbereich Mathematik und Informatik der Universität Münster
Arbeitsbereich Didaktik der Informatik
Fliednerstrasse 21
48149 Münster

Rheine, den 25. Mai 2013

Inhalt

Verzeichnis des Anhangs

1 Einleitung

Kryptographie, die Wissenschaft der Verschlüsselung von Informationen, wurde schon im Altertum eingesetzt wenn geheime Informationen sicher übermittelt werden sollten. Über Jahrhunderte gab es nur die Verschlüsselung von Hand, die vornehmlich für militärische Zwecke eingesetzt wurde. Erst in der Mitte des 20. Jahrhunderts wurden erste Maschinen zur Verschlüsselung entwickelt, die die Verschlüsslung von Hand ersetzen. Seit ihrem Siegeszug in den 1970er Jahren, werden auch Computer zur Verschlüsselung von Informationen genutzt. Seitdem hat die Kryptographie stark an Bedeutung gewonnen. Es folgte die Entwicklung von der vornehmlich staatlich und militärisch genutzten Geheimwissenschaft, zu einem öffentlichen Forschungsgebiet.

In der heutigen stark vernetzten Welt nimmt die Kryptographie einen immer größeren Stellenwert ein. Elektronisches Geld, online Banking, Cloud Computing, digitale Kommunikation und Smartphones sind Teil der Gesellschaft geworden. Heute sind es nicht mehr nur geheime Botschaften, die sicher übermittelt werden sollen. Immer mehr persönliche Daten und Informationen werden im Netz gespeichert. Kryptographie ist nicht mehr nur der Verschlüsselung von Botschaften, sie hat sich zur Wissenschaft der Informationssicherheit gewandelt. Sie dient unter anderem der Vertraulichkeit und der Integrität von gespeicherten Daten und übermittelten Informationen. Die Sicherheit von Daten und die geschützte Übertragung von Informationen ist zu einem gesamtgesellschaftlichen Thema geworden, das auch die Lebenswelt der Schülerinnen und Schüler betrifft. Aus diesem Grund ist das Thema Kryptographie sowohl Teil der „Richtlinien und Lehrpläne für die Sekundarstufe II – Gymnasium/Gesamtschule in Nordrhein-Westfalen[1]“ für das Schulfach Informatik, als auch in den „Vorgaben zu den unterrichtlichen Voraussetzungen für die schriftlichen Prüfungen im Abitur in der gymnasialen Oberstufe“ für die Jahre 2013 bis 2015.

Das RSA-Verfahren, eingesetzt z. B. zur Verschlüsslung und Signierung von E-Mails, ist Thema dieser Arbeit. Ausgehend von den aktuellen Richtlinien und Lehrplänen und den Vorgaben zu schriftlichen Abiturprüfungen für das Fach Infor-

[1] Ministerium für Schule und Weiterbildung, Wissenschaft und Forschung des Landes Nordrhein-Westfalen, 1999

matik in Nordrhein-Westfalen, soll untersucht werden, ob und in welcher Form das RSA-Verfahren Teil des Informatikunterrichts der Sekundarstufe II ist. Ziel ist es zu ergründen, ob und in welcher Form das RSA-Verfahren Thema im Informatikunterricht ist, wo und welcher Art derzeit Probleme bestehen und es sollen Methoden zur unterrichtlichen Vermittlung des RSA-Verfahrens aufgezeigt werden.

1.1 Aufbau der Arbeit

Der Aufbau der Arbeit gestaltet sich wie folgt. In Abschnitt 2 werden die grundsätzlichen Eigenschaften von Public Key-Verfahren und das RSA-Verfahren in allen seinen Schritten vorgestellt. Daran anschließend wird in Abschnitt 3 analysiert und eingeordnet, welche Rahmenbedingungen und Vorgaben bezüglich des Themas RSA-Verfahren im Informatikunterricht der Sekundarstufe II für Nordrhein-Westfalen vorliegen. Abschnitt 4 fasst die zur Vermittlung des RSA-Verfahrens nötigen mathematischen Grundlagen zusammen und ordnet diese den einzelnen Schritten des Algorithmus zu. Um zu ergründen, ob und in welcher Form das RSA-Verfahren Teil des Informatikunterrichts ist, wird in Abschnitt 5 ein Fragebogen vorstellt und die gewonnenen Erkenntnisse aus dieser Umfrage unter Informatiklehrerinnen und Lehrern analysiert und eingeordnet. Abschnitt 6 beschäftigt sich mit verschiedenen Konzepten für Methoden der unterrichtlichen Vermittlung. Tools die in Zusammenhang mit der Vermittlung bzw. Vermittlung des RSA-Verfahrens genutzt werden können, werden in Abschnitt 7 vorgestellt, bevor in Abschnitt 8 die gewonnenen Erkenntnisse abschließend zusammengefasst werden.

2 Das RSA-Verfahren

Das RSA-Verfahren, benannt nach dessen Entwicklern Rivest, Shamir und Adleman, wurde 1977 vorgestellt. Es ist ein asymmetrisches kryptographisches Verfahren zur Verschlüsslung und Signierung von digitalen Dokumenten. Der Unterschied zu symmetrischen Verfahren besteht darin, dass die kommunizierenden Partner keinen gemeinsamen Schlüssel kennen müssen. Das heißt, ein vorheriger Schlüsseltausch ist bei diesen Verfahren nicht notwendig. Um mit dem RSA-Verfahren verschlüsselte Nachrichten auszutauschen, benötigt jeder Teilnehmer ein

Schlüsselpaar. Einen öffentlichen Schlüssel zum Verschlüsseln und einen privaten, geheimen Schlüssel zum Entschlüsseln der Nachrichten. Kryptographische Verfahren, die auf dem Prinzip eines öffentlichen Schlüssels basieren, werden auch Public-Key-Verfahren genannt.

2.1 Das Grundprinzip von Public-Key-Verfahren

Um ein Public-Key-Verfahren zu benutzen, benötigt jeder Teilnehmer ein Schlüsselpaar bestehend aus einem öffentlichen und einem privaten Schlüssel. Der öffentliche Schlüssel $E=E_T$ des Teilnehmers *T* wird zur Verschlüsslung (Encryption) benötigt. Der private Schlüssel $D=D_T$ zur Entschlüsselung (Decryption).
Ein Public-Key-Verfahren muss zwei Anforderungen erfüllen[2].

1. Die eindeutige Entschlüsselung: Für alle Nachrichten *m* muss gelten $D_T(E_T(m))=m$. E_T bezeichnet die Verschlüsslung der Nachricht unter Verwendung des öffentlichen Schlüssels des Teilnehmers *T*. Das Schlüsselpaar muss so zusammenpassen, dass das Anwenden des privaten Schlüssels D_T die Verschlüsslung mittels E_T wieder aufhebt.
2. Die Public-Key-Eigenschaft: Aus dem öffentlichen Schlüssel D_T darf der private Schlüssel E_T praktisch nicht, bzw. nur mit einem sehr hohem Aufwand, zu erschließen sein.

Bei einem Public-Key-Verfahren läuft die verschlüsselte Kommunikation zwischen zwei Teilnehmern *A* und *B* in der Regel so ab, dass sich der Sender *A* den öffentlichen Schlüssel E_B seines Kommunikationspartners *B* aus einem frei zugänglichen Verzeichnis sucht. *A* verschlüsselt die Nachricht *m* unter Anwendung des öffentlichen Schlüssels E_B und sendet die verschlüsselte Nachricht $E_B(m)$ an Empfänger *B*. Nachrichtenempfänger *B* entschlüsselt die übermittelte Nachricht $E_B(m)$ mit dem nur ihm bekannten privaten Schlüssel D_B : $D_B(E_B(m))=m$.

Um ein Public-Key-Verfahren zu definieren benötigt man demnach 3 Algorithmen. Einen zur Schlüsselerzeugung, einen zur Verschlüsslung und einen dritten zur Entschlüsselung[3]. Da privater und öffentlicher Schlüssel in einer engen Bezie-

[2] Beutelspacher, Neumann & Schwarzpaul, 2009 S. 107
[3] Beutelspacher, Neumann & Schwarzpaul, 2009 S. 107

hung zueinander stehen, ohne das es möglich sein soll den privaten aus dem öffentlichen Schlüssel zu berechnen, ist der Algorithmus zur Erzeugung der Schlüssel in der Regel etwas komplexer, als bei einem symmetrischen Verfahren. Die Vorteile werden aber schnell deutlich. Zum einen sind Sender und Empfänger nicht darauf angewiesen vor der vertraulichen Kommunikation einen Schlüssel zu vereinbaren. Zum anderen reduziert sich die Zahl der benötigten Schlüssel deutlich, da pro Teilnehmer nur 2 Schlüssel nötig und zu speichern sind.

2.2 Schlüsselerzeugung für das RSA-Verfahren

Wie oben beschrieben, benötigt man für jeden Teilnehmer im System ein Schlüsselpaar, um mit dem RSA-Verfahren verschlüsselte Nachrichten auszutauschen. Die Erzeugung dieser Schlüssel ist ein wichtiger Teil des Verfahrens, der allerdings pro Teilnehmer nur ein mal durchgeführt werden muss. Um die Schlüssel zu erzeugen wählt man zwei unterschiedliche Primzahlen p und q und bildet ihr Produkt, das RSA-Modul $n = p \cdot q$. Danach berechnet man die *Eulersche* φ*−Funktion* von n: $\varphi(n) = (p-1) \cdot (q-1)$. Anschließend wählt man eine zu $\varphi(n)$ teilerfremde Zahl e und sucht ein d für das gilt: $e \cdot d \ mod \ \varphi(n) = 1$. Die für die Ver- und Entschlüsselung nötigen Schlüssel bestehen nun jeweils aus einem Zahlenpaar. Der private Schlüssel aus (d, n), der öffentliche Schlüssel aus (e, n), wobei das RSA-Modul n bei beiden Schlüsseln gleich ist. Die zur Schlüsselerzeugung benötigten Primzahlen p und q sowie die *Eulersche* φ*−Funktion* aus deren Produkt n werden nicht mehr benötigt, sollten aber geheim bleiben, da der private Schlüssel mit diesen Informationen zu berechnen wäre.

2.3 Die Verschlüsslung mit dem RSA-Verfahren

Um eine Nachricht mit dem RSA-Verfahren zu verschlüsseln, muss diese aus einer oder mehreren natürlichen Zahlen oder Zahlenblöcken bestehen. In der Praxis werden verschiedene Verfahren angewandt, um Nachrichten wie z. B. E-Mails, die in der Regel aus verschiedenen Zahlen, Buchstaben und Sonderzeichen bestehen, so zu codieren, dass eine Verschlüsselung mit dem RSA-Verfahren möglich ist. Die Voraussetzung ist, dass die codierte Zahl oder die Zahlenblöcke kleiner sind

als das RSA-Modul n. Ist diese Voraussetzung erfüllt und kennt man den korrekten öffentlichen Schlüssel (e, n), kann die als Zahl codierte Nachricht m verschlüsselt werden, „indem sie mit *e* potenziert und dann mit *modulo n* reduziert wird“[4]. Formal ausgedrückt erhält man das Ergebnis, also den chiffrierten Text *c*, durch die Formel $c = m^e \mod n$.

2.4 Die Entschlüsselung mit dem RSA-Verfahren

Die Entschlüsselung einer Nachricht läuft nach dem gleichen Prinzip ab wie die Verschlüsselung. Notwendig ist hierfür der zweigeteilte private Schlüssel (d, n) und die verschlüsselte Nachricht *c*. Durch Potenzierung der verschlüsselten Nachricht *c* mit *d* und der anschließenden Reduzierung durch *modulo n* erhält man die entschlüsselte Nachricht *m*. Formal ausgedrückt berechnet man die dechiffrierte Nachricht *m* durch die Formel $m = c^d \mod n$. Nachdem die Nachricht decodiert wurde, nach dem gleichen Prinzip wie bei der Codierung, erhält man den lesbaren Klartext der ursprünglichen Nachricht.

2.5 Warum funktioniert das RSA-Verfahren?

Die wichtigste mathematische Grundlage für das RSA-Verfahren ist der Satz von Euler. Er garantiert, „dass der Empfänger genau die Nachricht entschlüsselt, die der Sender übermitteln wollte[5]“, die Nachricht also korrekt entschlüsselt wird.

Der Satz von Euler besagt, dass man bei der Division von $m^{\varphi(n)}$ durch *n* den Rest 1 erhält, wenn *m* und *n* zwei teilerfremde natürliche Zahlen sind. Formal ausgedrückt ist $m^{\varphi(n)} \mod n = 1$. Zusätzlich gilt für jede natürliche Zahl *k*, das $m^{1+k\varphi(n)} \mod n = m$ ergibt. In dem für das RSA-Verfahren relevanten Fall ist *n* eine so genannte Semiprimzahl, also das Produkt aus zwei verschiedenen Primzahlen. Aus dem Satz von Euler lässt sich folgern, dass falls *n* das Produkt aus zwei verschiedenen Primzahlen *p* und *q* ist und *m* eine natürliche Zahl < *pq*, für jede natürliche Zahl *k* gilt: $m^{k(p-1)(q-1)+1} \mod pq = m$. Das RSA-Verfahren ist somit „eine direkte Anwendung des Satzes von Euler.“[6] Zum sicheren ver- und entschlüsseln müssen allerdings die erforderlichen Parameter korrekt ausgewählt

[4] Beutelspacher, 2009 S. 110
[5] Beutelspacher, Neumann & Schwarzpaul, 2009. Seite 116
[6] Beutelspacher, 2009 S.102

werden. Hat man zwei verschiedene möglichst große Primzahlen p und q gefunden und $\varphi(pq) = \varphi(p-1)(q-1)$ berechnet, sucht man ein zu $\varphi(n)$ teilerfremdes e ($ggT(e, \varphi(n)) = 1$). Mit dem erweiterten euklidischen Algorithmus lässt sich dann relativ einfach das modulare Inverse d (als privater Schlüssel) zu e (öffentlicher Schlüssel) berechnen, so dass gilt: $e \cdot d \bmod \varphi(n) = 1$. Aus dem Satz von Euler und der Definition von e und d als $e \cdot d = 1 + k\varphi(n)$ ergibt sich nun $m^{e \cdot d} \bmod n = m^{1+k\varphi(n)} \bmod n = m$. Einfach beschrieben könnte man sagen, die Folgerung aus dem Satz von Euler für ein Modul aus zwei verschiedenen Primzahlen und die Existenz einer multiplikativen Inversen beim modularen Potenzieren wird ausgenutzt, um ein asymmetrisches Kryptosystem zu haben, bei dem eine Nachricht durch das modulare Potenzieren mit dem einen Exponenten (öffentlicher Schlüssel) m verschlüsselt und durch das modulare Potenzieren mit der modularen Inversen (privater Schlüssel) die verschlüsselte Nachricht wieder m zum Ergebnis hat.

2.6 Schwierigkeiten und Sicherheit des RSA-Verfahrens

Die Schwierigkeiten bei der Anwendung des RSA-Verfahrens liegen weder bei der eigentlichen Schlüsselerzeugung noch bei der Verschlüsselung oder Entschlüsselung. Kennt man die beiden Primzahlen p und q stellt die Berechnung von $\varphi(n)$ kein Problem dar. Auch die Berechnung von e und d bereitet keine größeren Schwierigkeiten. Hat man ein zu $\varphi(n)$ teilerfremdes e gefunden, lässt sich d mit Hilfe des erweiterten euklidischen Algorithmus einfach berechnen. Auch die Ver- und Entschlüsselung als mathematische Operationen ist, gerade mit Hilfe von Computern, relativ einfach zu berechnen.

Das eigentliche Problem stellt das Finden der benötigten Primzahlen p und q dar. Da die Sicherheit vom RSA-Verfahren unmittelbar von der Größe der genutzten Primzahlen abhängt, werden in der Praxis teilweise Primzahlen mit mehreren hundert Stellen genutzt. Da diese Primzahlen nur dem Schlüsselerzeuger bekannt sein dürfen, sollten sie zufällig gewählt werden. In der Praxis werden dafür Zufallszahlen erzeugt, die darauf getestet werden, ob sie wirklich eine Primzahl sind. Die dafür genutzten mathematischen Verfahren sind sogenannten Primzahltests. Bei Primzahlen der gesuchten Größenordnung kommt es auch auf die Effizienz

der eingesetzten Verfahren an. Aus diesem Grund setzt man häufig probabilistische Verfahren wie den Müller-Rabin-Test ein, die zwar nur mit einer „gewissen Wahrscheinlichkeit“[7] nachweisen ob die geprüfte Zufallszahl tatsächlich eine Primzahl ist, aber deutlich effizienter sind als genauere Primzahltests.

Die Idee, die hinter dem RSA-Algorithmus steht, ist die Variante einer sogenannten Einwegfunktion. Als Einwegfunktion bezeichnet man eine Funktion, die einerseits effizient zu berechnen, andererseits aber nicht effizient umzukehren ist. Sind diese beiden Eigenschaften gegeben, „kann die Einwegfunktion *f* zur effizienten Verschlüsselung verwendet werden“[8] und es ist sichergestellt, dass der Klartext nicht effizient aus dem Geheimtext zu berechnen ist. Da es für den Empfänger allerdings möglich sein soll, den Geheimtext effizient zu entschlüsseln, wird beim RSA-Verfahren eine Einwegfunktion mit Falltür benutzt. Diese Falltür ist beim RSA-Verfahren der private Schlüssel. Er garantiert die effiziente Entschlüsselung des Geheimtextes. Das modulare potenzieren, wie es im RSA-Verfahren praktiziert wird, ist so eine Einwegfunktion. Die Falltür zum Umkehren dieser Einwegfunktion ist die modulare Inverse zum Exponenten.

Die Sicherheit des RSA-Verfahrens hängt also maßgeblich davon ab, ob der private Schlüssel aus dem öffentlichen zu berechnen ist. Ob es möglich, bzw. wie einfach es ist, hängt von der Größe der zur Schlüsselerzeugung genutzten Zahlen ab. Denn „die Sicherheit des RSA-Algorithmus beruht auf der Schwierigkeit, große Zahlen zu faktorisieren“[9]. Falls es möglich ist, das aus dem öffentlichen Schlüssel bekannte *n* in seine Primfaktoren zu zerlegen und $\varphi(n)$ zu berechnen, stellt es kein Problem dar, den privaten Schlüssel unter Anwendung des erweiterten euklidischen Algorithmus zu finden. Da es derzeit keine effizienten Verfahren gibt, mit denen man sehr große Zahlen faktorisieren kann, gilt das RSA-Verfahren als sicher. Voraussetzung ist, man wählt hinreichend große Primzahlen.

[7] Beutelspacher, 2009 S.109
[8] Hromkovic, Freiermuth, Keller & Steffen, 2009 S. 246
[9] Beutelspacher, Neumann & Schwarzpaul, 2009 S.125

3 Vorgaben zum RSA-Verfahren im Informatikunterricht. Rahmenbedingung und Einordnung

In diesem Abschnitt soll untersucht werden, welche Vorgaben es bezüglich des RSA-Verfahrens als Thema im Schulfach Informatik gibt. Hauptaugenmerk wird dabei auf das Bundesland Nordrhein-Westfalen gelegt. Die Frage ist, inwieweit das Thema Kryptographie im Allgemeinen und das RSA-Verfahren im Speziellen Teil der Richtlinien und Lehrpläne für die Sekundarstufe II – Gymnasium/Gesamtschule in Nordrhein-Westfalen und den Vorgaben zu den unterrichtlichen Voraussetzungen für die schriftlichen Prüfungen im Abitur in der gymnasialen Oberstufe ist. Des Weiteren wird untersucht, ob und in welcher Form das RSA-Verfahren Bestandteil der Zentralabiturprüfungen im Schulfach Informatik in Nordrhein-Westfalen sind.

3.1 Richtlinien und Lehrpläne für die Sekundarstufe II – Gymnasium/Gesamtschule in Nordrhein-Westfalen. Informatik.

Eine Analyse des Lehrplans zeigt, dass grundsätzliche Bereiche des Faches und bestimmte fachliche Inhalte als obligatorisch für den Unterricht der Sekundarstufe II vorgeschrieben sind. Die einzelnen Inhalte werden aber wenig konkret beschrieben und lassen entsprechenden Freiraum. Das RSA-Verfahren als Teilthema der Kryptographie ist nicht direkt Teil der verpflichtenden Vorgaben. Einzuordnen wäre es am ehesten im Bereich „2.2.2 Lernen im Kontext der Anwendung“[10]. Hier sollen typische Anwendungsbereiche der Informatik beleuchtet werden, die exemplarischen Charakter haben und verschiedene Facetten des Faches aufzeigen. Als Beispiel für einen Anwendungsbereich wird hier unter anderem „Datenschutz und Datensicherheit“[11] genannt. Als Beschreibung dieses Anwendungsbereiches steht dort, dass „kryptologische Verschlüsselungsalgorithmen“[12] zu diesem Thema gehören.

[10] Ministerium für Schule und Weiterbildung, Wissenschaft und Forschung des Landes Nordrhein-Westfalen, 1999 S.19

[11] Ministerium für Schule und Weiterbildung, Wissenschaft und Forschung des Landes Nordrhein-Westfalen, 1999

[12] Ministerium für Schule und Weiterbildung, Wissenschaft und Forschung des Landes Nordrhein-Westfalen, 1999

Nach Maßgabe des Lehrplans ist es also durchaus möglich, das RSA-Verfahren als Thema im Unterricht der Sekundarstufe II zu behandeln. Verpflichtend vorgeschrieben ist es allerdings nicht.

3.2 Vorgaben zu den unterrichtlichen Voraussetzungen für die schriftlichen Prüfungen im Abitur in der gymnasialen Oberstufe für das Fach Informatik der Jahre 2013 bis 2015 in NRW

Um die im Lehrplan für das Fach Informatik vorgegeben Inhalte im Hinblick auf die zentralen Abiturprüfungen zu konkretisieren und die für das Zentralabitur notwendigen inhaltlichen Voraussetzungen zu schaffen, werden vom Schulministerium NRW für jeden Abiturjahrgang entsprechende inhaltliche Vorgaben veröffentlicht. Die Vorgaben für die Abiturprüfungen der Jahre 2013 bis 2015 unterscheiden sich inhaltlich nicht und bestehen, auf „der Grundlage der Obligatorik des Lehrplans Informatik“[13], aus fünf inhaltlichen Schwerpunkten. Von diesen fünf Schwerpunkten sind die Punkte „I.1 Konzepte des objektorientierten Modellierens“[14] und „I.2 Algorithmen und Datenstrukturen“[15] obligatorisch. Zusätzlich zu den beiden obligatorischen Schwerpunkten müssen mindestens zwei der Schwerpunkte I.3, II oder III um Unterricht behandelt werden. Das Thema Kryptographie ist Teil des inhaltlichen Schwerpunktes „I.3 Modellieren und Implementieren kontextbezogener Problemstellungen als Netzwerkanwendungen“[16] und sieht im Teilbereich asymmetrische Verschlüsselungsverfahren das RSA-Verfahren als Thema vor. Anders als im Lehrplan von 1999 ist hier das RSA-Verfahren konkret als Teilthema im Bereich Kryptographie vorgesehen. Da es allerdings nicht den obligatorischen Schwerpunkten zugeordnet ist, muss es nicht verpflichtend Teil des Unterrichts der Sekundarstufe II sein.

[13] Ministerium für Schule und Weiterbildung des Landes Nordrhein-Westfalen, 2012
[14] Ministerium für Schule und Weiterbildung des Landes Nordrhein-Westfalen, 2012
[15] Ministerium für Schule und Weiterbildung des Landes Nordrhein-Westfalen, 2012
[16] Ministerium für Schule und Weiterbildung des Landes Nordrhein-Westfalen, 2012

3.3 Untersuchung der schriftlichen Abituraufgaben der Jahre 2008 bis 2012

Die Analyse des Lehrplans und der Vorgaben zu den schriftlichen Abiturprüfungen lässt deutlich werden, dass das RSA-Verfahren grundsätzlich Thema im Informatikunterricht der gymnasialen Oberstufe sein kann. Ein weiterer Indikator für den tatsächlichen Stellenwert des Themas sind die Aufgaben der Zentralabiturprüfungen in Nordrhein-Westfalen. Unter Umständen ist es möglich, anhand der bisher gestellten Abituraufgaben Rückschlüsse über die tatsächliche Verbreitung des RSA-Verfahrens als Thema im Informatikunterricht in Nordrhein-Westfalen zu ziehen. Die Untersuchung der Zentralabituraufgaben aus Nordrhein-Westfalen der Jahre 2008 bis 2012 und deren Bezüge zu den Vorgaben kommt da zu einem eindeutigem Ergebnis. Insgesamt gibt es in diesem Zeitraum zwei Aufgaben mit Bezug zum Thema Kryptographie. In beiden Aufgaben aus den Jahren 2012 und 2009 geht es ausschliesslich um symmetrische Verschlüsselungsverfahren. Asymmetrische Verschlüsselungsverfahren, insbesondere das RSA-Verfahren, waren bisher nicht Teil der schriftlichen Zentralabiturprüfungen.

Es stellt sich die Frage, inwieweit man durch diese Untersuchung Rückschlüsse auf den tatsächlichen Stellenwert des RSA-Verfahren im Informatikunterricht schließen kann. Es lässt sich allerdings vermuten, dass das RSA-Verfahren keine große Rolle im Informatikunterricht spielt. Bleibt zu analysieren, ob es tatsächlich so ist und welche Gründe es haben könnte.

4 Mathematische Grundlagen für das RSA-Verfahren im Informatikunterricht

Die RSA-Verfahren, als asymmetrisches kryptographisches Verfahren, beruht zu einem Großteil auf bestimmen Eigenschaften von natürlichen, ganzen Zahlen. Um den RSA-Algorithmus zu verstehen und ihn korrekt anwenden zu können, sind einige mathematische Grundlagen aus dem Bereich der elementaren Zahlentheorie erforderlich, die auch bei Schülerinnen und Schülern der gymnasialen Oberstufe nicht unbedingt vorausgesetzt werden können. In welchem Umfang der mathematische Hintergrund Teil einer Unterrichtsreihe zum Thema RSA-Verfahren ist oder

sein sollte, hängt von verschiedenen Faktoren ab. Ein wichtiger Faktor ist die Unterrichtszeit, die für das RSA-Verfahren im Unterricht zu Verfügung steht. Das bereits vorhandene Wissen der Schülerinnen und Schüler ist ein anderer. In Abhängigkeit von diesen Faktoren sollte man sich Gedanken darüber machen, wie weit man diese Unterrichtsreihe fassen möchte. Die Bandbreite, in der man sich beim RSA-Verfahren bewegen kann, reicht von der Vorstellung des Verfahrens, über die einfache Anwendung, für die man nur die grundlegende mathematische Operationen benötigt, bis zu Primzahlproblemen und einer genaueren Analyse des Faktorisierungsproblems und der Sicherheit von RSA.

4.1 Zahlentheoretische Grundlagen

Wie bereits beschrieben, ist für das Verständnis und die korrekte Anwendung des RSA-Verfahrens mathematisches Vorwissen im Bereich der elementaren Zahlentheorie vonnöten. Welches Wissen man benötigt ist davon abhängig, wie detailliert man das RSA-Vefahren behandeln will. Geht es nur um die einfache Vorstellung des Verfahrens und die Berechnung eines Beispiels der Ver- und Entschlüsselung, reichen Kenntnisse in modularer Arithmetik und des modularen Potenzierens.
Sollen zusätzlich selber Schlüssel erzeugt werden, benötigt man Wissen aus dem Bereich der Primzahlen und des erweiterten euklidischen Algorithmus. Um zu verstehen, warum das RSA-Verfahren funktioniert, wie sicher und warum es sicher ist und wie es in der Praxis eingesetzt wird, benötigt man zusätzliches Wissen wie den Satz von Euler, Kenntnisse über Primzahltests und Faktorsierungsverfahren. In den folgenden Abschnitten soll ein Überblick über die zum Verständnis der Schülerinnen und Schüler nötigen zahlentheoretischen Grundlagen und deren Zuordnung zu den einzelnen Phasen des RSA-Algorithmus gegeben werden.

4.1.1 Verschlüsselung und Entschlüsselung

Der Verschlüsselungs- und der Entschlüsselungsalgorithmus im RSA-Verfahren läuft nach dem gleichen Prinzip ab. Im Hinblick auf den mathematischen Hintergrund ist dementsprechend keine Unterscheidung notwendig. Als mathematische Operation ist hier nur eine vergleichsweise einfache modulare Exponentation notwendig. Wie bereits beschrieben, wird die zu chiffrierende oder die zu dechiffrie-

rende Zahl mit dem jeweiligen öffentlichen oder privaten Schlüssel potenziert und anschließend mit der Modulo Operation und dem entsprechenden Modul aus dem Schlüssel wieder reduziert. Zu beachten ist allerdings, dass ein zu chiffrierender Text bereits in Zahlen oder Zahlenblöcke codiert sein muss, die kleiner sein müssen als das Modul aus dem Schlüssel, da die Verschlüsslung nur mit natürlichen Zahlen möglich ist. Zur Ausführung des Algorithmus brauchen die Schülerinnen und Schüler prinzipiell nur grundlegende Kenntnisse der modularen Arithmetik, also eine einfache Division mit Rest. Für sehr kleine Beispiele funktioniert das relativ einfach. Werden die Zahlen größer, stößt man allerdings ohne weitere Hilfsmittel schnell an Grenzen. Zum besseren Verständnis des Algorithmus und des mathematischen Hintergrunds ist es sinnvoll und notwendig auch Begriffe wie Restklasse und Kongruenz zu klären.

4.1.2 Schlüsselerzeugung

Grundlage für die Funktion und die Sicherheit des RSA-Verfahrens sind Primzahlen. Die Schülerinnen und Schüler sollten wenigstens über Grundwissen aus dem Bereich der Primzahlen verfügen. Das heißt, was ist eine Primzahl bzw. wie ist eine Primzahl definiert, welche Eigenschaften haben Primzahlen und wo liegt der Unterschied zu zusammengesetzten Zahlen. Denn der wichtigste Schritt beim Erzeugen der Schlüssel ist das Finden von zwei geeigneten (und hinreichend großen) Primzahlen.Wie auch bei der Ver- und Entschlüsselung stellen Beispiele mit verhältnismäßig kleinen Primzahlen kein Problem dar. Sie lassen sich in überschaubarer Zeit einfach finden oder aus einer Primzahlentabelle ablesen. Weitere Verfahren zum Finden großer Primzahlen finden sich in Abschnitt 4.1.4.

Ein weiterer Begriff der geklärt werden sollte, ist die Teilerfremdheit. Betrachtet man den weiteren Verlauf der Schlüsselerzeugung, stellt man fest, dass die Teilerfremdheit Bestandteil von zwei weiteren Schritten der Schlüsselerzeugung ist. Zum einen muss die eulersche $\varphi-Funktion$ des aus den beiden Primzahlen berechneten RSA-Moduls berechnet werden, zum anderen muss der öffentliche Teil des Schlüssels wiederum teilerfremd zur errechneten $\varphi-Funktion$ sein. Laut Definition der eulerschen $\varphi-Funktion$ ist „ $\varphi(n)$ […] die Anzahl der zu *n* tei-

lerfremden natürlichen Zahlen, die kleiner oder höchsten gleich *n* sind“[17]. Da die beiden Primfaktoren bekannt sind, aus denen *n* berechnet wurde, ist die Lösung für $\varphi(n)$ leicht zu finden. Für zwei verschiedene Primzahlen *p* und *q* ist $\varphi(pq) = (p-1)(q-1)$. Da zwei natürliche Zahlen teilerfremd sind, wenn sie keine gemeinsamen Primfaktoren aufweisen, lässt sich die Teilerfremdheit für kleine Zahlen einfach durch die Primfaktorzerlegung, also die Darstellung einer natürlichen Zahl als Produkt aus Primzahlen, nachweisen. Da die Berechnung der Primfaktorzerlegung für große Zahlen sehr aufwändig ist, nutzt man im RSA-Verfahren im Allgemeinen den euklidischen Algorithmus zum Nachweis der Teilerfremdheit. Mit Hilfe des euklidischen Algorithmus lässt sich über eine wiederholte Division mit Rest sehr effizient der größte gemeinsame Teiler (ggT) von zwei natürlichen Zahlen bestimmen. Ist der ggT = 1 haben die untersuchten Zahlen keine gemeinsamen Primfaktoren und sind dementsprechend teilerfremd.

Die letzte Zahlentheoretische Grundlage, die zur Schlüsselerzeugung notwendig ist, ist das Finden des geheimen Schlüsselteils, also der modularen Inversen zum öffentlichen Schlüssel und des RSA-Moduls. Für zwei Teilerfremde Zahlen *a* und *n* existiert einen dritte ganze Zahl *b*, die mit *a* multipliziert und durch *n* reduziert den Rest 1 ergibt. Diese Zahl ist die modulare oder multiplikative Inverse von *a mod n*. Berechnen lässt sich diese Zahl durch die Erweiterung des euklidischen Algorithmus um die Vielfachsummendarstellung des größten gemeinsamen Teilers.

4.1.3 Warum funktioniert das RSA-Verfahren?

Warum funktioniert das RSA-Verfahren? Diese Frage muss eigentlich zweigeteilt beantwortet werden. Einerseits stellt sich die Frage, warum kann mit dem RSA-Algorithmus korrekt chiffriert und dechiffriert werden? Das heißt, auf welchem mathematischen Hintergrund basiert das RSA-Verfahren? Andererseits steht die Frage im Raum warum man, trotz öffentlich bekanntem Schlüssel, derzeit nicht der Lage ist, die Verschlüsselung zu knacken.

Um die Frage zu beantworten warum die Entschlüsselung gelingt, das heißt, dass der Empfänger tatsächlich die Nachricht bekommt, die der Sender ihm über-

[17] Beutelspacher, Schwenk & Wolfenstetter, 2010 S. 123

mitteln wollte, brauchen die Schülerinnen und Schüler die Kenntnis eines weiteren Satzes aus der Zahlentheorie. Die Antwort auf diese Frage liefert der Satz von Euler. Er „ist die wichtigste mathematische Grundlage für die Korrektheit des RSA-Algorithmus“[18]. Der Satz von Euler ist eine Verallgemeinerung des kleinen fermatschen Satzes. Dieser Satz besagt, dass wenn eine ganze Zahl *a* mit einer Primzahl *p* potenziert und durch *p* geteilt wird, der Rest *a* bleibt. Formal ausgedrückt: $a^p \bmod p = a$. Sind *a* und *p* teilerfremd und wird *a* mit *p-1* potenziert und durch *p* dividiert, ergibt sich ein Rest von *1* ($a^{p-1} \bmod p = 1$). Der Satz von Euler ist eine Verallgemeinerung des zweiten Falles in der Hinsicht, dass das Modul nicht notwendigerweise ein Primzahl sein muss. Ist *a* eine ganze Zahl, *n* eine natürliche Zahl und sind *a* und *n* teilerfremd, so erhält man den Rest 1, wenn *a* mit $\varphi(n)$ potenziert und durch *n* dividiert wird ($a^{\varphi(n)} \bmod n = 1$).

Eine für den RSA-Algorithmus entscheidende Folgerung aus dem Satz von Euler ergibt sich, wenn „*n* das Produkt von zwei verschiedenen Primzahlen ist“[19]. Laut dieser Folgerung gilt $m^{k(p-1)(q-1)+1} \bmod pq = m$, wenn *p* und *q* zwei verschiedene Primzahlen sind, *k* eine natürliche Zahl und *m* eine natürliche Zahl < *pq*. Anders als beim eigentlichen Satz von Euler müssen *m* und $n = pq$ nicht teilerfremd sein. „Die Aussage gilt für alle Zahlen $m \leq n$.“[20] Diese Folgerung und die Existenz einer multiplikativen Inversen für teilerfremde Module schaffen die Möglichkeit der asymmetrischen Verschlüsslung nach dem RSA-Verfahren.

4.1.4 Die Sicherheit vom RSA-Verfahren

Um den zweiten Teil der Eingangsfrage von Abschnitt 4.1.3 zu verstehen, brauchen die Schülerinnen und Schüler weiteres Wissen aus dem Bereich der Zahlentheorie. Zusätzlich ist zur Einordnung der gewonnen Erkenntnisse weiteres Wissen erforderlich. Damit die Schülerinnen und Schüler verstehen können, warum das RSA-Verfahren sicher ist und ob das RSA-Verfahren tatsächlich eine Einwegfunktion mit Falltür ist bzw. ob es überhaupt Einwegfunktionen gibt, brauchen sie Kenntnisse aus dem Bereich der Komplexitätstheorie. Genauer gesagt geht es um

[18] Beutelspacher, Neumann & Schwarzpaul, 2009 S. 121
[19] Beutelspacher, 2009 S.103
[20] Beutelspacher, 2009 S.103

das P-NP-Problem, also den Komplexitätsklassen P und NP und deren Beziehung zueinander.

Das RSA-Verfahren gilt als sicher, solange bestimmte Vorgaben eingehalten werden. Neben der korrekten Anwendung der Vorgaben zur Schlüsselerzeugung ist sind es unter anderem die Geheimhaltung der beiden Primzahlen, von $\varphi(n)$ und natürlich des privaten Schlüssels. Die wichtigste Vorgabe ist allerdings die Auswahl geeigneter Primzahlen. Das heißt, die Primzahlen sollen einerseits groß genug sein damit mit deren Produkt *n* nicht in polynomineller Zeit faktorisiert werden kann und andererseits möglichst zufällig gewählt werden. Für das grundlegende Verständnis beider Vorgaben ist zusätzliche Wissen aus dem Bereich der Effizienz von Algorithmen erforderlich.

Zum Verständnis, warum RSA ein sicheres Kryptographieverfahren ist und um den Aufwand der Faktorisierung selber abzuschätzen, brauchen die Schülerinnen und Schüler die Kenntnis von verschiedenen Faktorisierungsverfahren.

Die älteste und einfachste Methode zur Faktorisierung einer Zahl ist die Probedivision. Hier wird die zu faktorisierende Zahl durch jede Primzahl dividiert, bis alle Primfaktoren gefunden wurden, oder eine bestimme Schranke überschritten wurde. Das Prinzip ist das gleiche wie bei der Primfaktorzerlegung. Das Verfahren kann dementsprechend sowohl als Primzahltest, als auch als Faktorisierungsverfahren angewendet werden. Für kleine Primfaktoren lässt sich die Probedivision sehr gut nutzen. Für große Primfaktoren ist es nicht effizient genug. Weitere, effizientere Verfahren zur Faktorisierung sind zum Beispiel das quadratische Sieb und das Zahlkörpersieb.

Wie bereits beschrieben ist ein der Hauptprobleme des RSA-Verfahrens das Finden großer Primzahlen. Die Schülerinnen und Schüler werden sich irgendwann die Frage stellen, ob es genügend und ausreichend große Primzahlen gibt. Da die Sicherheit vom RSA-Algorithmus auch zu einem bestimmten Teil davon abhängt, dass möglichst viele Teilnehmer unterschiedliche Schlüssel, die aus unterschiedlichen Primzahlen erzeugt werden, nutzen. Des Weiteren sollen die Primzahlen zur RSA-Modul Erzeugung auch möglichst weit auseinander liegen.

Zwei Primzahltests haben die Schülerinnen und Schüler bisher im Prinzip schon kennengelernt. Zum einen die Probedivision und zum anderen den kleinen

Fermatschen Satz (als „Spezialisierung“ des Satzes von Euler). Durch die Umkehrung der Bedingung aus dem kleinen fermatschen Satz lässt sich prüfen, ob eine natürliche Zahl zusammengesetzt oder eine Primzahl ist. Beim fermatschen Primzahltest wählt man zu der zu testenden Zahl *n* eine teilerfremde natürliche Zahl $1<a<n$ als Basis, potenziert die Basis mit $n-1$ und dividiert das Ergebnis durch n. Erhält man den Rest 1 ist *n* eine Primzahl. Bei einem anderen Rest ist *n* zusammengesetzt. Um ein sicheres Ergebnis zu erhalten und Pseudoprimzahlen auszuschließen, muss man den Test für *n* mit mehreren Basen durchführen. Trotzdem ist es möglich, dass eine Zahl den Test besteht, obwohl sie zusammengesetzt sind. Häufig ist das bei den sogenannten Carmichael-Zahlen, also Zahlen, die aus mindestens drei Primzahlen zusammengesetzt sind, der Fall.

Der für das RSA-Verfahren wichtigste Primzahltest ist der Miller-Rabin-Test. Er ist eine Erweiterung des Fermat Tests. Beim Miller-Rabin-Test wird eine weitere Eigenschaft von Primzahlen ausgenutzt. Wenn die zufällig ausgewählte Zahl tatsächlich eine Primzahl ist, „hat die Kongruenz $x^2 \equiv 1(mod\ p)$ nur die Lösungen $x \equiv 1(mod\ p)$ und $x \equiv -1(mod\ p)$.“[21] Da alle Primzahlen außer der 2 ungerade sind, wird dieser Test nur mit zufällig gewählten ungeraden Zahlen durchgeführt. Für den Test der Zufallszahl *n* wird zunächst *n-1* berechnet und in der Form $n-1 = 2^s \cdot d$ zerlegt, wobei *d* ungerade sein muss. Für jede Testrunde wird ein *a* aus dem Intervall *2* bis *n-1* als Basis gewählt und durch fortgesetztes Quadrieren von $a^d\ mod\ p$ eine Folge erstellt. Ist in der Folge eine 1 oder -1 enthalten, ist *n* sehr wahrscheinlich eine Primzahl. Je mehr Durchgänge mit verschiedenen Basen durchgeführt werden, desto wahrscheinlicher ist *n* tatsächlich eine Primzahl. Der Müller-Rabin-Test ist ein probabilistisches Verfahren und es kann daher nicht 100 prozentig sichergestellt, dass *n* tatsächlich eine Primzahl ist. Trotzdem gilt das Verfahren als sehr verlässlich und liefert bezogen auf die Länge der getesteten Zahl ein Ergebnis in exponentieller Laufzeit. Neuere Verfahren, wie der AKS-Primzahltest und dessen Verbesserungen, schaffen es mittlerweile in polynomineller Laufzeit zu testen ob eine natürliche Zahl zu den Primzahlen gehört oder eine zusammengesetzte Zahl ist.

[21] Witten & Schulz, 2010

4.2 Zusammenfassung der mathematischen Grundlagen

Betrachtet man die letzten Abschnitte stellt man fest, dass für das umfassende Verständnis des RSA-Verfahrens einiges an mathematischen oder genauer gesagt zahlentheoretischen Kenntnissen erforderlich ist. Allerdings müssen nicht unbedingt alle Grundlagen innerhalb oder für eine Unterrichtseinheit über das RSA-Verfahren neu gelegt werden. Ausgehend vom Ziel und vom Umfang der Einheit könnte entweder auf dem Vorwissen der Schülerinnen und Schüler aufgebaut werden oder Schritt für Schritt das benötigte Wissen erarbeitet werden. Einiges, wie zum Beispiel grundlegende Kenntnisse der Moduloarithmetik, Primzahlen und Primfaktorzerlegung und Verfahren zum Finden des größten gemeinsamen Teilers, dürften den Schülerinnen und Schüler der Sekundarstufe II bereits bekannt sein. Einerseits durch den Mathematikunterricht, andererseits durch den vorhergegangenen Informatikunterricht. Der folgende Abschnitt soll Ideen und Konzepte für Unterrichtsmethoden zu einer Unterrichtseinheit zum RSA-Verfahren aufzeigen, bei dem die zahlentheoretischen Grundlagen je nach Bedarf eingefügt oder ausgelassen werden können.

5 Das RSA-Verfahren im Informatikunterricht

Wie Abschnitt 3 zu entnehmen ist, sind die Rahmenbedingungen das RSA-Verfahren als Thema im Informatikunterricht zu behandeln, wenn auch nicht verpflichtend, gegeben. Die sich anschließenden Frage ist, ob das RSA-Verfahren tatsächlich Bestandteil des Informatikunterrichts ist und welche Verbreitung es hat. Zusätzlich ist es interessant zu erfahren, in welcher Form das Verfahren vermittelt wird und wo eventuell Probleme bei der Vermittlung bestehen. Ist das RSA-Verfahren nicht Bestandteil des Informatikunterrichts, wäre es sinnvoll zu wissen, welche Gründe dafür vorliegen.

Um Antworten auf diese Fragen zu finden, wurde ein Fragebogen zum Thema RSA-Verfahren im Informatikunterricht erstellt. Dieser Fragebogen befindet sich im Anhang dieser Arbeit und besteht aus 17 Fragen, die dazu dienen sollen, Antworten auf die eingangs formulierten Fragestellungen zu finden. Durchgeführt

wurde diese Umfrage als Online-Fragebogen mit dem Fragebogenprogramm GrafStat, das sowohl zur Datenerfassung als auch zur Auswertung der erfassten Daten genutzt wurde. Anhand des Philologen-Jahrbuchs Schuljahr 2011/12 Landesausgabe NRW für Gymnasien und Gesamtschulen[22] wurden insgesamt 41 Lehrerinnen und Lehrer von 25 Gymnasien und 6 Gesamtschulen ausgewählt, die das Fach Informatik unterrichten. Um Lehrerinnen und Lehrer aus ganz Nordrhein-Westfalen zu erreichen, erfolgte die Auswahl in etwa gleich verteilt aus allen fünf Regierungsbezirken in NRW. Vor Beginn der Online-Umfrage erhielten die ausgewählten Lehrerinnen und Lehrer ein Anschreiben an die jeweilige Schuladresse, in dem das Anliegen des Fragebogens erläutert wurde und mit der Bitte, sich an der Umfrage zu beteiligen. Der Zeitraum zur Teilnahme lag bei ca. 3 Wochen. Die Quote der Rückmeldungen, bzw. der ausgefüllten Fragebögen liegt bei ca. 20 Prozent der zuvor angeschriebenen Lehrerinnen und Lehrer. Die Rückmeldeqoute könnte unter anderem darin begründet, dass die angeschriebenen Lehrerinnen und Lehrer nicht mehr an den im Philologen-Jahrbuch angegebenen Schulen tätig sind oder derzeit das Fach Informatik nicht oder nicht regelmäßig unterrichten. Die Ergebnisse des Fragebogens sollen im folgenden Abschnitt vorgestellt und eingeordnet werden.

5.1 Auswertung und Einordnung der Ergebnisse des Fragebogens zum RSA-Verfahren im Informatikunterricht

Auf die Frage, ob das RSA-Verfahren Teil ihres Unterrichts ist, antworten 75 Prozent der Teilnehmer mit ja. In Anbetracht der Tatsache, dass das RSA-Verfahren nicht verpflichten für den Informatikunterricht der Sekundarstufe II in NRW ist, ist das ein relativ hoher Wert, der allerdings bei einer Umfrage unter einer größeren Anzahl an Teilnehmern oder einer höheren Rückmeldequote auch niedriger ausfallen könnte. Die Nichtbehandlung des RSA-Verfahrens wird auf verschiedene Arten begründet. Einerseits wird der Themenbereich Kryptographie in der Sekundarstufe zwar behandelt, aber „nicht bis in die Tiefen von RSA“. Begründet wird diese Aussage nicht weiter, allerdings lässt sich vermuten, dass der zur Vermittlung nötige zeitliche Rahmen nicht gegeben ist. Anderseits deutet sich an, dass das RSA-Verfahren in einigen Schulen nicht Teil Unterrichts ist, weil genau die

[22] *Philologen-Jahrbuch Schuljahr 2011/12. Kunzes Kalender im 111. Jahrgang/Landesausgabe NRW für Gymnasien und Gesamtschulen*,

Schwerpunkte der unterrichtlichen Vorgaben zum Zentralabitur gewählt werden, in denen das Thema Kryptographie und das RSA-Verfahren nicht enthalten sind (siehe Abschnitt 3.2). Es ist allerdings anzumerken, dass das RSA-Verfahren nicht grundsätzlich für zu anspruchsvoll für den Informatikunterricht gehalten wird.

Weiterhin lässt sich anhand der Ergebnisse des Fragebogens feststellen, dass das RSA-Verfahren sowohl Thema in Grund- und Leistungskursen, als auch im Wahlbereich der Sekundarstufe I ist und es von den Teilnehmern durchweg als relevant für den Informatikunterricht angesehen wird.

Sehr aufschlussreich sind Antworten auf die Frage, wo die Lehrerinnen und Lehrer konkrete Probleme bei der Vermittlung des RSA-Verfahrens sehen. Bei dieser Frage mit Mehrfachauswahl steht das fehlende allgemeine Vorwissen im Bereich Mathematik der Schülerinnen und Schüler mit 75 Prozent der Antworten neben den speziellen mathematischen Problemen mit 50 Prozent an erster Stelle. Konkret werden die Probleme unter anderen bei der Modulo-Rechnung, dem euklidischen Algorithmus und allgemein im Bereich der Zahlentheorie gesehen, da dieser Bereich im Mathematikunterricht im Allgemeinen nicht behandelt wird. Es wird allerdings auch ausgeführt, dass die Probleme sich in Grenzen halten, wenn die mathematischen Beweise bei der Vermittlung des RSA-Verfahren außen vor gelassen werden.

Des Weiteren wird das Fehlen von automatisierten Tools für die notwendigen Berechnungen und die Problematik der notwendigen Langzahlarithmetik zur Schlüsselerzeugung angesprochen. Den Antworten lässt sich entnehmen, dass das Verfahren im Unterricht teilweise nur allgemein angesprochen wird, ohne näher auf die Funktionsweise und den mathematischen Hintergrund einzugehen.

Bezüglich der Frage ob zu viel mathematisches Vorwissen nötig ist, lässt sich keine Eindeutige Meinung, aber durchaus eine Richtung erkennen. 62,5 Prozent der Befragten beantworten die Frage, ob das Thema RSA zu viel mathematischen Vorwissen voraussetzt, mit nein. Bei den Fragen, wie intensiv auf den mathematischen Hintergrund eingegangen wird und für wie wichtig die befragten Lehrerinnen und Lehrer den mathematischen Hintergrund halten, lässt sich kein eindeutiges Ergebnis erkennen. In der Tendenz wird der mathematische Hintergrund allerdings als eher wichtig bzw. sehr wichtig angesehen.

In der Antwort auf die Frage, ob sich die Lehrerinnen und Lehrer aus persönlicher Sicht, also unabhängig von Vorgaben, dafür aussprechen das RSA-Verfahren im Unterricht zu behandeln, antworten fast 90 Prozent der Befragten mit ja. Das deutet darauf hin, dass dem RSA-Verfahren durch die fehlende Verpflichtung (siehe Abschnitt 3) in den Richtlinien, Vorgaben und unter Umständen schulinternen Curricula, ein zu niedriger Stellenwert eingeräumt wird.

Weitere Fragen beziehen sich auf das genutzte Unterrichtsmaterial und die Unterrichtzeit, die für das RSA-Verfahren aufgewendet wird. Anhand der Antworten lässt sich feststellen, dass größtenteils eigenes Unterrichtsmaterial eingesetzt wird, dass unter Zuhilfenahme von Fachliteratur und Internetquellen erstellt wird und (bis auf eine Ausnahme) für das Thema RSA keine Schulbücher speziell für den Informatikunterricht eingesetzt werden.

Die eingesetzte Unterrichtszeit unterscheidet sich allerdings deutlich, was Rückschlüsse auf die Intensität der Behandlung zulässt. Hier lässt sich eine Spanne von 2 Unterrichtsstunden, wahrscheinlich für eine nur grundlegende Vorstellung des Verfahrens, bis zu 12 Unterrichtsstunden, für die Behandlung inklusive mathematischen Hintergrund, erkennen. Teil von Klausuren ist das RSA-Verfahen in der Regel nicht. Zur Verbesserung des Unterrichts werden vor allem Hilfestellungen zu den mathematischen Grundlagen und Arbeitsblätter gewünscht.

Der komplette Fragebogen inklusive der abgegeben Antworten findet sich im Anhang I.

6 Konzepte für Vermittlungsmethoden für das RSA-Verfahren im Informatikunterricht

Betrachtet man das RSA-Verfahren in seiner kompletten Bandbreite, lassen sich verschieden Ansatzpunkte finden, es als Thema im Unterricht zu behandeln. Bevor man eine Unterrichtsreihe zum RSA-Verfahren plant, sind aber einige grundsätzliche Vorüberlegungen notwendig. In Abhängigkeit verschiedener Faktoren, wie der zur Verfügung stehenden Unterrichtszeit, der jeweiligen Lerngruppe und dem bereits vorhandenen Kenntnissen der Schülerinnen und Schüler bieten sich verschiedene Methoden an das RSA-Verfahren im Unterricht zu vermitteln. Wie

bereits beschrieben, sind für ein umfassendes Verständnis des Verfahrens umfangreiche Kenntnisse aus der Zahlentheorie notwendig. Zum umfassenden Verständnis gehört allerdings nicht nur die reine Anwendung der Algorithmen zur Schlüsselerzeugung und der Ver- und Entschlüsselung oder sogar nur das Durchführen eines einfachen Beispiels mit bereits vorhandenen Schlüsseln und kleinen Zahlen. Zusätzlich zur Kenntnis und dem Verständnis des eigentlichen RSA-Algorithmus gehören zum Beispiel Überlegungen, warum die Verschlüsselung mit RSA sicher ist, wie und unter welchen Voraussetzungen es sicher ist und ob und wie lange es nach derzeitigen Wissens- und Technikstand sicher bleiben kann.

Auch Kenntnisse verschiedener Vorteile und Nachteile gegenüber anderen kryptographischen Verfahren trägen zu einem umfassenden Verständnis bei. Hinter all dem steht die Frage, wie viel Mathematik wirklich notwendig ist. Betrachtet man nur die eigentliche Anwendung des Verfahrens, also die Erzeugung der Schlüssel und die Ver- und Entschlüsselung, ist die Durchführung, gerade unter Zuhilfenahme von Computern und den passenden Werkzeugen, ohne tiefer gehende mathematische Kenntnisse einfach möglich. Das erworbene Wissen beruht dann allerdings nur auf der reinen Kenntnis des Verfahrens, das Verständnis der Funktionsweise fehlt. Trotzdem würde es sich, auch ohne den die eigentlichen Gedanken hinter dem RSA-Verfahren zu verstehen, von den Schülerinnen und Schüler als Programm umsetzten lassen.

Wie bereits beschrieben, sollte also die Anwendung des Verfahrens für den jede Schülerin und jeden Schüler nachvollziehbar sein. Will man aber verstehen, warum das RSA-Verfahren funktioniert und die Schülerinnen und Schüler das Verfahren sich im Unterricht selber erschließen lassen, ist das nicht ohne Kenntnisse aus der Zahlentheorie möglich.

Abhängig von den bereits beschrieben Faktoren haben sicherlich mehrere Methoden und Wege das Thema zu vermitteln ihre Berechtigung. Ist die Entscheidung gefallen das RSA-Verfahren im Unterricht zu behandeln, stellt sich die grundsätzliche Frage, wie mathematisch soll und muss dieses Verfahren im Informatikunterricht behandelt werden. Kann man „das Verfahren so aufbereiten, dass auch die Schülerinnen und Schüler es verstehen, die die mathematischen Hintergründe

nicht vollständig durchdringen?“[23] Ist es möglich, in Zeiten von Zentralabitur und verkürzter Schulzeit, die nötigen Hintergründe im Informatikunterricht zu vermitteln? Eine weitere sich anschließende Frage ist, in weit es nötig und sinnvoll ist, den zahlentheoretischen Hintergrund gemeinsam mit den Schülerinnen und Schülern zu erarbeiten oder ihn sich selbständig erarbeiten zu lassen.
Zusammenfassend lässt sich festhalten, dass bedingt durch den nötigen mathematischen Hintergrund eine ausführliche „und mathematisch-informatische fundierte Behandlung des RSA-Verfahrens […] erst in der Oberstufe möglich“[24] ist. Da sich das Verfahren, mit gewissen Abstrichen, auch schon in der Sekundarstufe I vermitteln lässt [25], könnte man in Abhängigkeit von Faktoren wie Unterrichtszeit und Vorwissen auch eine weniger mathematisch fundierte Behandlung des Themas in Betracht ziehen. Die Schülerinnen und Schüler würden das Verfahren kennen lernen und hätten, trotz eingeschränktem mathematischen Hintergrund, die Möglichkeit generelle Aussagen über asymmetrische Kryptographie und die Sicherheit des RSA-Verfahrens zu treffen.

Ziel des folgenden Abschnittes ist, es einen Überblick über verschieden Methoden zu geben, das RSA-Verfahren im Unterricht zu vermitteln und Möglichkeiten für Beispiele und Aufgaben zum Verfahren darzustellen.

6.1 Grundsätzliche Überlegungen zu Arbeitsformen und Methoden

Stellt man grundsätzliche Überlegungen zu Vermittlungsformen des RSA-Verfahrens im Informatikunterricht an, bieten sich mehrere Optionen. Neben der Vermittlung im Klassenverband unter Berücksichtigung von verschiedenen Vermittlungsmethoden und in Abhängigkeit von den bereits beschriebenen Faktoren, bieten sich weitere Vermittlungsmethoden an. Eine Möglichkeit wäre ein über mehrere Unterrichtsstunden angelegtes Projekt zum RSA-Verfahren, bei dem sich wieder mehrere Vermittlungsoptionen anbieten würden. Ein anderer Ansatz ist, das Thema asymmetrische Kryptographie und RSA-Verfahren in einer Art Workshop zu behandeln, der auf verschieden Arten ausgestaltet werden könnte. Neben der Mög-

[23] Witten, Esslinger, Gramm & Hornung, 2012 S.9
[24] Witten, Esslinger, Gramm & Hornung, 2012 S.79
[25] Vgl. Witten, Esslinger, Gramm & Hornung, 2012

lichkeit einzelne Bereiche des RSA-Algorithmus durch einführende Referate der Schülerinnen und Schüler vorbereiten zu lassen, bietet sich in der Sekundarstufe II auch eine Facharbeit an, die sich entweder mit dem Thema RSA als ganzem oder mit detaillierterer Bearbeitung von Teilbereichen des RSA-Verfahrens beschäftigen könnte.

Vor der Entscheidung, wie das RSA-Verfahren erfolgreich vermittelt werden kann, steht immer die im vorherigen Abschnitt gestellte grundsätzliche Frage, wie detailliert möchte und muss die Lehrkraft den mathematischen Hintergrund in den Unterricht integrieren. Wie bereits beschrieben, kann diese Frage nur in Abhängigkeit von den bereits ausgeführten Faktoren getroffen werden. Da die zahlentheoretischen Einzelheiten der verschiedenen Phasen des Verfahrens bereits beschrieben wurden, soll hier nicht näher darauf eingegangen werden. Aber prinzipiell bieten sich auch hier mehrere Optionen. Es besteht zum Beispiel die Möglichkeit, nicht auf alle zahlentheoretischen Einzelheiten einzugehen. So könnte unter anderem auf detaillierte mathematische Herleitungen, wie zum Beispiel die der eulerschen $\varphi-Funktion$ für ein Produkt aus zwei Primzahlen, und Beweisführungen verzichtet werden.

Ziel einer Unterrichtsreihe zum RSA-Verfahren sollte nicht nur die Kenntnis eines (weiteren) kryptographischen Verfahrens sein. Unabhängig vom umfassenden Verständnis des mathematischen Hintergrunds, soll den Schülerinnen und Schülern die Möglichkeit gegeben werden, die Sicherheit des Verfahrens und deren Abhängigkeit von bestimmten Faktoren selber einschätzen zu können.

In den Abschnitten 6.2 bis 6.5 werden verschiedene Konzepte für Vermittlungsmethoden aufgezeigt und in den Abschnitten 6.6 und 6.7 Möglichkeiten für verschiedene Aufgaben und Beispiele zum RSA-Verfahren dargestellt. In Kapitel 7 werden verschiedene Tools und Literaturempfehlungen vorgestellt, die zur Vermittlung im Unterricht eingesetzt werden können.

6.2 Das RSA-Verfahren als Projekt im Informatikunterricht

Neben dem Unterricht im Klassenverband bietet sich das RSA-Verfahren auch als Thema für Projekte im Informatikunterricht an. Hier sind verschieden Ansätze denkbar, die sich in Schwierigkeit und Zeitanspruch deutlich unterscheiden kön-

nen. Einige Möglichkeiten für Projekte zum RSA-Verfahren soll hier angeschnitten werden.

6.2.1 Programmierprojekt

Gerade im Informatikunterricht bietet es sich an, die Schülerinnen und Schüler das RSA-Verfahren selbständig implementieren zu lassen. Hierzu müssen die Grundlagen allerdings vor dem Start des Projekts vorhanden sein. Das heißt, die Schülerinnen und Schüler müssen die mathematischen Grundlagen beherrschen und den Ablauf des eigentlichen RSA-Algorithmus bereits kennen und anwenden können.

Als Projektaufgabe könnte ein Programm entstehen, das alle Schritte des RSA-Verfahren vereint. Es soll also ein Benutzerprogramm erstellt werden, das folgende Grundoperationen beherrscht:

- Schlüsselerzeugung
- Verschlüsselung eines einzugebenden Klartextes
- Entschlüsselung eines einzugebenden Geheimtextes

Abhängig von diesen Grundoperationen sich zusätzliche Funktionen zu implementieren, die für die einzelne Schritte notwendig sind. Als Teil der Schlüsselerzeugung ist es notwendig, einen Algorithmus zum Finden großer Primzahlen zu implementieren. Zusätzlich ist ein Algorithmus nötig, der ein zu $\varphi(n)$ teilerfremde *e* sucht oder erzeugt und ein passendes modulares Inverses als privaten Schlüssel findet.

Im Rahmen der Verschlüsselung ist es nötig einen Algorithmus zu implementieren der den eingegeben Text in den zur Verschlüsselung notwendigen Zahlencode übersetzt und für die eigentliche Verschlüsselung eine Funktion, die Modulo-Exponentation vornimmt. Für die Entschlüsselung wird wiederum die Funktion zu Modulo-Exponentation und eine Rückübersetzung des Zahlencodes in den Klartext benötigt.

Zusätzlich könnte das Programm eine Verwaltung für einen privaten und mehrere öffentliche Schlüssel von Kommunikationspartnern bieten.

Da für ein Programm in dieser oder ähnlicher Form viele einzeln zu implementierende Algorithmen nötig sind, bietet sich besonders ein Gruppenprojekt an.

6.2.2 Vorbereitung der Grundlagen durch die Schülerinnen und Schüler

Ein weiteres Projekt könnte die Vorbereitung des RSA-Verfahrens durch die Schülerinnen und Schüler sein. In Form von Präsentationen könnten die Schülerinnen und Schüler in Einzel-, Gruppen- oder Partnerarbeit Präsentationen zu einzelnen Bereichen des RSA-Verfahren erstellen. Durch die Vorführung der Präsentationen wäre eine grundlegende Einführung in Teilbereiche des Verfahrens möglich. Auch hier sind verschiedene Schwierigkeitsgrade möglich. Für Präsentationen mit einfacheren Themen könnte man Bereiche wie die Geschichte des RSA-Verfahrens, die allgemeine Vorstellung von asymmetrischen Kryptographieverfahren oder einfache Beispiele für Einwegfunktionen wählen. Sollen die Themen etwas anspruchsvoller sein, bieten sich Themen zu mathematischen oder zahlentheoretischen Grundlagen an wie z. B. die Vorstellung der Modulo-Division oder ein Vortrag über Primzahlen.

6.2.3 Projekt: RSA-Challenge

Ein für Schülerinnen und Schülerinnen sehr interessantes Projekt wäre eine Art RSA-Challenge. Um ein Projekt dieser Art durchzuführen müssen allerdings, ähnlich wie dem in Abschnitt 5.2.1 beschriebenen Programmierprojekt, die mathematischen Grundlagen und die Kenntnis des Ablaufs des RSA-Algorithmus bereits bekannt sein. Anbieten würde sich ein Projekt in der Form, dass eine Gruppe oder einzelne Schülerinnen und Schüler zuerst einen bestimmten Vorgaben entsprechenden Text oder eine Nachricht verschlüsseln. Eine andere Gruppe hat dann die Aufgabe diese Nachricht möglichst effizient zu knacken. Je nachdem, wie viel Zeit in dieses Projekt investiert werden soll, kann die Vorgabe sein, die zur Ver- und Entschlüsselung und zum Knacken der Nachricht nötigen Programme selber zu implementieren oder die Schülerinnen und Schüler ein Programm wie CrypTool verwenden zu lassen. Die Nutzung eigener Programme bietet sich natürlich genau dann an, wenn diese schon im Rahmen der Vermittlung des RSA-Verfahrens erstellt wurden.

Eine weitere Möglichkeit wäre, wenn die einzelnen Gruppen eine vom Lehrer vorgegebene Nachricht in der Form eines Wettbewerbs, einer sogenannten RSA-Chipper-Challange, knacken sollen. Hier sollte es nicht nur um die reine Faktori-

sierung des RSA-Moduls gehen, sondern der komplette Ablauf des Verfahrens durchgespielt werden. Der Weg zum Ziel, also die Nachricht ohne privaten Schlüssel zu Entschlüsseln, bleibt dabei den Schülerinnen und Schülern überlassen.

6.3 Das RSA-Verfahren für den Unterricht im Klassenverband

6.3.1 Schritt für Schritt Erarbeitung der zahlentheoretischen Grundlagen

Eine Möglichkeit, wie sich die Schülerinnen und Schüler den Zugang zum RSA-Verfahren über die elementare Zahlentheorie selber erarbeiten können, beschreibt Dr. Hermann Puhlmann in seinem Manuskript „Kryptographie verstehen – Ein schülergerechter Zugang zum RSA-Verfahren“[26], das allerdings eher auf den Unterricht in der Sekundarstufe I ausgerichtet ist. Ein ähnlichen, sehr auf den zahlentheoretischen Hintergrund ausgerichteten Ansatz das RSA-Verfahren zu vermitteln, entstammt einer Kooperation zwischen dem Fraunhofer-Institut Algorithmen und Wissenschafliches Rechnen SCAI und dem mathematischen Institut der Universität zu Köln. Das Unterrichtsmodul „RSA – Primzahlen zur Verschlüsslung von Nachrichten“[27] ist auf den Mathematikunterricht ab Klasse 9 ausgerichtet. Einen weiteren Ansatz zur Schritt für Schritt Erarbeitung der notwendigen zahlentheoretischen Grundlagen bietet das Manuskript „Lauschen zwecklos! Oder Wie aus einer seltsamen Erkenntnis über Zahlen die beste Geheimsprache aller Zeiten wurde“[28] von Kirchgraber und Kramer. Anhand dies Skriptes, das nicht direkt für den Unterricht entwickelt wurde, lassen sich die mathematischen Grundlagen mit Hilfe von Definitionen und Aufgaben im Selbststudium erarbeiten, um sie dann in Zusammenhang mit dem RSA-Verfahren anzuwenden.

Das Unterrichtsmodul von Puhlmann führt nur in den eigentlichen RSA-Algorithmus ein und verzichtet auf mathematische Beweise, den euklidischen Algorithmus und den Satz von Euler. Die Sicherheit des RSA-Vefahrens und worauf diese gründet, sowie Details zum Finden großer Primzahlen bleiben außen vor.

Das Unterrichtsmodul „RSA - Primzahlen zu Verschlüsslung von Nachrichten“ verzichtet zwar auch auf den Satz von Euler, beinhaltet aber den erweiterten eukli-

[26] Puhlmann, 1998
[27] Schüller, Trottenberg, Wienands, Koziol & Schneider, 2013
[28] Kirchgraber & Kramer, 2005

dischen Algorithmus und führt Begriffe wie Restklassen und den der Einwegfunktion ein. Die Sicherheit vom RSA-Verfahren und Details zu Primzahlen und Primzahltests sind nicht Bestandteil des Moduls. Die zur umfassenden Behandlung des Verfahrens nötigen fehlenden Details und Eigenschaften lassen sich damit begründen, dass beide Unterrichtsmodule für den Mathematikunterricht der Sekundarstufe I konzipiert wurden.

Plant man in der Sekundarstufe II einen Einstieg ins RSA-Verfahren über die Zahlentheorie, kann man trotzdem auf Konzepte und Methoden zurückgreifen, die an diese Unterrichtsmodule für die Sekundarstufe I angelehnt sind.

Im Hinblick auf die Motivation der Schülerinnen und Schüler für das Thema und in Abgrenzung zum reinen Mathematikunterricht, erscheint es sinnvoll, die Zahlentheorie nicht direkt in den Vordergrund zu stellen. Eine schrittweise Erarbeitung der zum Verständnis des RSA-Verfahrens nötigen Grundlagen, zum Beispiel durch eine Verknüpfung mit anderen informatischen Fragestellungen oder eine Erarbeitung im direkten Zusammenhang mit dem RSA-Algorithmus scheint dem Informatikunterricht eher angemessen.

Für die Vermittlung des RSA-Verfahren über die Zahlentheorie, ohne diese direkt in den Vordergrund zu stellen, bieten sich verschiedene Konzepte an.

6.3.1.1 Von der Verschlüsselung durch Addition zur Verschlüsslung durch Potenzierung oder von der Caesar-Chiffre zum RSA-Verfahren.

Ein weiterer Ansatz das RSA-Verfahren zu vermitteln, wird unter anderem von Puhlmann[29] und Witten/Schulz[30] beschrieben. Hier soll es nicht um „einen Kurs zur elementaren Zahlentheorie mit RSA als krönenden Abschluss gehen“[31], sondern es sollen alle Schritte mit einer kryptologischen Fragestellung verknüpft werden. Die Idee ist, dass die Schülerinnen und Schüler sich durch den Weg über die Verschlüsslung durch Addition und die Verschlüsslung durch Multiplikation das RSA-Verfahren als asymmetrische Verschlüsslung durch Potenzierung selber erschließen können. Wie bereits dargestellt, ist das Skript von Puhlmann für den Mathematikunterricht ausgelegt und verfolgt den Ansatz, vordergründig die Ma-

[29] Puhlmann, 1998
[30] Witten & Schulz, 2006
[31] Witten & Schulz, 2006 S.50

thematik hinter dem RSA-Verfahren zu vermitteln. Der Artikel von Witten/Schulz verknüpft die mathematischen und kryptographischen Ansätze zusätzlich mit Konzepten aus der Informatik. Für die Vermittlung des RSA-Verfahrens im Informatikunterricht der Sekundarstufe II scheint daher ein Vermittlungskonzept, dass an die Ideen von Witten/Schulz angelehnt ist, sinnvoller. Wie ein solches Konzept vom Grundsatz her aussehen könnte, wird im folgenden Abschnitt beschrieben. Diese Methode soll in Form von Phasen beschrieben werden, die bis zum Verständnis des eigentlichen RSA-Verfahrens durchlaufen werden sollen.

Ausgangspunkt für eine an dieses Konzept angelehnte Vermittlungsmethode ist die den Schülerinnen und Schülern unter Umständen bereits bekannte Caesar-Chiffre, die sich mathematisch auch als eine Verschlüsselung mittels modularer Addition darstellen lässt.

Phase 1: Falls bisher nicht bekannt, lernen die Schülerinnen und Schüler in Phase 1 zum Einstieg die Caesar-Chiffre kennen, bei der jeder Buchstabe des Klartext-Alphabets durch die Verschiebung um eine bestimmte Anzahl an Stellen auf einen Buchstaben im Geheimtext-Alphabet abgebildet wird. Die Caesar-Chiffre dient in diesem Fall dazu, in die modulare Arithmetik einzuführen. Codiert man das Alphabet in Zahlen von 0 bis 26, wobei die 0 als Leerzeichen dient. Verschiebt das Klartext-Alphabet um k Zeichen, lässt sich die Nachricht m mit der Formel $(m+k) \, mod \, 26$ in den Geheimtext c verschlüsseln. Die Entschlüsselung gelingt mit der Formel $(c-k) \, mod \, 26$.

Phase 2: Die zweite Phase dient dazu, den Schülerinnen und Schülern die Existenz einer Inversen zu vermitteln, mit der die Verschlüsslung, anstatt mit der oben angegeben Formel, wieder aufgehoben werden kann. Aus der symmetrischen Caesar-Chiffre wird so eine (unsichere) asymmetrische Verschlüsselungsmethode.

Phase 3: Wenn eine Verschlüsslung durch modulare Addition gelingt, werden sich die Schülerinnen und Schüler auch die Frage stellen, ob eine Verschlüsslung durch modulare Multiplikation möglich ist. Ausgehend von der Caesar-Chiffre könnte man zusammen mit den Schülerinnen und Schülern eine Verschlüsselungs-

methode entwickeln, bei der nicht der Schlüssel addiert wird, sondern mit einen Schlüssel multipliziert und das Ergebnis anschließend mit der Restwertdivision reduziert wird. Die Schülerinnen und Schüler werden feststellen, dass die Umkehrung dieser Verschlüsselung nicht so einfach möglich ist wie bei der Caesar-Chiffre. Mit Hilfe von Verknüpfungstafeln und Multiplikationstabellen für bestimme Module, können die Schülerinnen und Schüler allerdings feststellen, welche mathematischen Voraussetzungen für bestimme Module und Schlüssel gegeben sein müssen, um die für die Entschlüsselung benötigte modulare Inverse zu finden.

Phase 4: Wenn die Schülerinnen und Schüler erkannt, dass das ausgewählte Modul und der Schlüssel teilerfremd sein müssen, damit die Entschlüsselung korrekt funktioniert, besteht die Möglichkeit den euklidischen Algorithmus zu behandeln. Da die Verschlüsselung mittels modularer Multiplikation und kleinem Modul nicht sicherer ist als die Caesar-Chiffre, sind große Module oder lange Schlüssel nötig. Durch die Kenntnis des euklidischen Algorithmus ist nun auch die Möglichkeit gegeben, mit großen Schlüsseln und Modulen die Verschlüsslung vorzunehmen, da sich die Teilerfremdheit von Schlüssel und Modul so einfach überprüfen lassen. Neben der Anwendung per Hand könnten die Schülerinnen und Schüler den euklidischen Algorithmus auch als eigenes Programm umsetzten.

Phase 5: Das nächste Problem, was es für die Schülerinnen und Schüler zu lösen gilt, wäre eine effiziente Möglichkeit die modulare Inverse zum Entschlüsseln der Nachricht zu finden. Sie werden schnell feststellen, dass das Finden der modularen Inversen durch ausprobieren (auch mit Hilfe des Computers) unter bestimmten Voraussetzungen sehr lange dauern kann. Durch die Erweiterung des euklidischen Algorithmus um die Vielfachsummendarstellung bekommen die Schülerinnen und Schüler die Möglichkeit, die modulare Inverse zu Schlüssel und Modul effizient zu ermitteln. Das Problem ist allerdings, dass durch die Existenz einer effektiven Methode auch die Sicherheit des Verschlüsslung durch modulare Multiplikation gering bzw. nicht vorhanden ist. In dieser Phase bietet es sich an, dass die Schülerinnen und Schüler ihr Programm zum euklidischen Algorithmus zum erweiterten euklidischen Algorithmus ergänzen.

Phase 6: Die Schülerinnen und Schüler werden festgestellt haben, dass der Aufwand der Umkehrung der Verschlüsselungsfunktion von der Verschlüsselung mittels modularer Addition zur Verschlüsslung mittels modularer Multiplikation deutlich zugenommen hat. Daher liegt der Schritt nahe, eine Verschlüsselung mittels modularem Potenzieren in Betracht zu ziehen. Aufbauend auf dem verschlüsseln durch modulare Addition und durch modulare Multiplikation, bei denen die Entschlüsselung durch die Kenntnis eines modularen Inversen möglich ist, besteht jetzt die Möglichkeit die Existenz einer Einwegfunktion mit Falltür zu vermitteln. Der RSA-Algorithmus dient somit als Beispiel für eine Einwegfunktion mit Falltür und als Verschlüsselungsmethode durch modulare Potenzierung. Die nötigen zahlentheoretischen Grundlagen zur Berechnung der Schlüssel und zur Ver- und Entschlüsselung sind in den vorherigen Phasen bereits gelegt worden. Zur Einführung des RSA-Algorithmus eignet sich hier unter anderem ein Beispiel mit kleinen Zahlen, das per Hand zu berechnen ist und als Erweiterung ein Beispiel mit größeren Zahlen, für das auch der Computer zu Hilfe genommen werden kann. Verschiedene Möglichkeiten werden in Abschnitt 6.6 dargestellt.

Je nach Unterrichtszeit und Motivation der Schülerinnen und Schüler sind weitere Phasen möglich. Es wäre zum Beispiel sinnvoll, die Schülerinnen und Schüler den RSA-Algorithmus als Programm umzusetzen zu lassen. Des Weiteren fehlen in diesem Ansatz noch tiefer gehende Überlegungen zur Sicherheit des Verfahrens und zum Finden von großen Primzahlen. Möglichkeiten dazu werden in Abschnitt 6.5 aufgezeigt.

6.3.1.2 Über die Caesar-Chiffre zum RSA-Verfahren

Einen Einstieg in die modulo Arithmetik über die Caesar-Chiffre bietet auch das für den Mathematikunterricht entwickelte Unterrichtsmodul von Schüller et al. [32]. Der Ablauf dieser Einheit ist allerdings konkreter auf die mathematischen Grundlagen des RSA-Verfahrens ausgelegt und verzichtet auf die in Abschnitt 6.3.1.1 beschriebene Entwicklung der Verschlüsslung mittels modularer Addition und Multiplikation als Zwischenstufen. Ein vom grundsätzlichem Ablauf her an

[32] Schüller, Trottenberg, Wienands, Koziol & Schneider, 2013

dieses Unterrichtsmodul angelehnte Methodenkonzept für den Informatikunterricht, das direkter auf den RSA-Algorithmus hinführt und den mathematischen Hintergrund mit informatischen Fragestellungen verknüpft, könnte in folgendermaßen aussehen.

Phase 1: Falls den Schülerinnen und Schülern die Caesar-Chiffre bisher nur als reine Verschiebechiffre, z. B. mittels Rad oder Scheibe, bekannt ist, könnte in Phase 1 die Vorcodierung mittels Zahlen und die Verschlüsslung durch Addition und modulo Division eingeführt werden. Ein weiterer Weg zur modulo Division einzuführen, wäre auch das Berechnen der Uhrzeit mittels modulo 24.

Phase 2: Die zweite Phase dient der Einführung der Begriffe Restklassen und Kongruenz. Nachdem die modulo-Funktion bzw. der modulo-Operator der im Unterricht genutzten Programmiersprache einführt wurde, sollen sich die Schülerinnen und Schüler über Experimente mit selber implementierten Funktionen erschließen, dass verschiedene natürliche Zahlen bei Division mit gleichem Modul die gleichen Reste als Ergebnis haben. Haben die Schülerinnen und Schüler das erkannt, kann man den Begriff der Kongruenz einführen.

Phase 3: In Phase 3 soll die Existenz von multiplikativen Inversen im Zusammenhang mit der modulo Division geklärt werden. Hierzu sollen die Schülerinnen und Schüler mit Hilfe einer eigenständig entwickelten Funktion Multiplikationstabellen für verschiedene Zahlen/Restklassen und Module ausgeben lassen. Nach dem man den Bergriff der multiplikativen Inversen eingeführt und definiert hat, dass das multiplikative Inverse einer Zahl a die Zahl ist, die mit a multipliziert und durch ein bestimmtes Modul wieder reduziert als Ergebnis 1 hat, sollen die Schülerinnen und Schüler ihre erstellten Multiplikationstabellen auf die Voraussetzungen untersuchen, unter denen ein multiplikativ Inverses existiert. Sie werden zu dem Ergebnis kommen, dass das Modul eine Primzahl sein muss oder die zu dividierende Zahl zum Modul teilerfremd sein muss.

Phase 4: Ein weiterer Schritt in Richtung RSA-Algorithmus, ist die Kenntnis des euklidischen bzw. erweiterten euklidischen Algorithmus. Da das Ablesen der multiplikativen Inversen aus Multiplikationstabellen für große Zahlen nicht mehr praktikabel ist, soll die Schülerinnen und Schüler in dieser Phase lernen, dass es effektivere Methode zur Bestimmung der Teilerfremdheit durch Primfaktorzerlegung und zum Finden eines multiplikativen Inversen gibt.

Da die modulare Division den Schülerinnen und Schülern bereits bekannt ist, sollte es für sie kein Problem darstellen, den euklidischen Algorithmus, nach Vorstellung durch den Lehrer und anschließend anhand von einigen Beispielen mit geeigneten Zahlen, selber durchzuführen. Nachdem die Schülerinnen und Schüler den euklidischen Algorithmus als Pseudocode beschrieben haben, sollen sie ihn als Aufgabe selbständig als Funktion umsetzen.

Als letzten Schritt dieser Phase, stellt der Lehrer den Schülerinnen und Schülern den erweiterten euklidischen Algorithmus zur Berechnung des modularen Inversen vor. Wie beim euklidischen Algorithmus sollen die Schülerinnen und Schüler zuerst Beispiele mit geeigneten Zahlen durchführen und anschließend den Algorithmus in Pseudocode beschreiben. Als weitere Aufgabe bietet sich die Implementierung des erweiterten euklidischen Algorithmus an.

Phase 5: In Phase 5 soll den Schülerinnen und Schüler der Unterschied zwischen der durch die Caesar-Chiffre bekannte symmetrische Verschlüsselung und der bisher unbekannten asymmetrischen Verschlüsselung erläutert werden.
Die Schülerinnen und Schüler sollen darlegen, was ihrer Meinung nach die Nachteile von symmetrischen Verschlüsselungsverfahren sind und welche Verbesserungen nötig und möglich wären. Ein Punkt wird sein, dass der Schlüsseltausch, bzw. die Schlüsselübergabe ein Unsicherheitsfaktor der symmetrischen Kryptosysteme sein kann. Eine mögliche Verbesserung ist also, auf diesen gemeinsamen Schlüssel zu verzichten und eine Möglichkeit zu suchen, ein asymmetrisches Verfahren zu entwickeln. Die Schülerinnen und Schüler sollten die Möglichkeit haben, sich über Vor- und Nachteile von asymmetrischen Verschlüsselungsverfahren auszutauschen, bevor der Hinweis auf Einwegfunktionen und deren speziellen Variante, den Einwegfunktionen mit Falltür, kommt.

Als Beispiel für eine Einwegfunktion dient die Multiplikation von zwei großen Primzahlen, die einfach durchzuführen, aber nur schwer umzukehren ist. Die Schülerinnen und Schüler werden erkennen, dass der öffentliche Schlüssel die Einwegfunktion ist und der private Schlüssel die zur Entschlüsselung nötige Falltür. Als wichtige Anforderung sollte erkannt werden, dass der private Schlüssel sich nicht aus dem öffentlichen Schlüssel erschließen lassen darf.

Phase 6: Die Suche nach einer Einwegfunktion mit Falltür führt dann direkt zum RSA-Verfahren. Da die mathematischen Grundlagen für den eigentlichen RSA-Algorithmus größtenteils behandelt wurden, ist es nach der Vorstellung für die Schülerinnen und Schüler bereits möglich, ein Beispiel für eine Verschlüsselung mit einer kleinen Zahl und bereits erstellten Schlüsseln zu berechnen. Gibt man den Schülerinnen und Schülern den genauen Ablauf zur Schlüsselerzeugung und zur Ver- und Entschlüsselung vor, sollte es auch möglich sein, unter Nutzung der bisher implementierten Funktionen, das RSA-Verfahren für die Verschlüsslung von kleinen Zahlen zu implementieren.

Phase 7: Um zu verstehen warum und wie genau die Verschlüsselung mittels RSA funktioniert, sind jedoch noch weitere Kenntnisse nötig. Elementar für das RSA-Verfahren als Einwegfunktion mit Falltür ist der Satz von Euler. Da den Schülerinnen und Schülern bereits bekannt ist, dass unter bestimmten Voraussetzungen eine modulare Inverse für eine modulo Multiplikation existiert, lässt sich zeigen, dass die Ausnutzung dieser in Zusammenhang mit einer Folgerung aus dem Satz von Euler eine Einwegfunktion mit Falltür ist, die als asymmetrisches Verschlüsselungsverfahren genutzt werden kann. Voraussetzung dafür ist, dass die Vorgaben für die Schlüsselerzeugung korrekt umgesetzt werden.

Phase 8: Um das RSA-Verfahren umfassend zu behandeln fehlen noch weitere Überlegungen zur Sicherheit des Verfahrens. Aus Phase 5 kennen die Schülerinnen und Schüler das Problem der Faktorisierung eines Produktes aus zwei Primzahlen als Einwegfunktion. Nach Vorstellung des Algorithmus wird also schnell

klar werden, dass die Sicherheit vom RSA-Verfahren genau von diesem Problem abhängt.
Um ein Gefühl für die Sicherheit des Verfahrens zu schaffen, bieten sich Laufzeitexperimente mit verschieden großen zusammengesetzten Zahlen und Faktorisierungsverfahren an. Ein Beispiel für eine einfaches aber ineffizientes Verfahren wäre die Probedivision, die auch relativ einfach zu implementieren ist. Weitere Möglichkeiten werden Abschnitt 6.4 beschrieben. Die Schülerinnen und Schüler werden schnell erkennen, dass die Größe der Primzahlen zu Schlüsselerzeugung der entscheidende Punkt für die Sicherheit des Verfahrens ist. Die sich anschließende Frage wäre, wie man effizient so große Primzahlen finden kann. Möglichkeiten für Experimente zu Primzahltests finden sich in Abschnitt 6.4.

6.3.2 Konzept Mathematikunterricht: Zuerst die Zahlentheorie, dann das RSA-Verfahren

Ein eher an den Mathematikunterricht angelehntes und für den Informatikunterricht eher weniger geeignetes Konzept wäre es, den Schülerinnen und Schülern nach und nach mit Hilfe von Beispielen und Übungsaufgaben die nötigen Grundlagen beizubringen und als Abschluss den RSA-Algorithmus vorzustellen.
Eine Einheit nach diesem Konzept könnte so aufgebaut sein, dass innerhalb von 3 bis 6 Phasen zuerst das nötige Wissen in Bereich der Zahlentheorie aufbaut wird, um in den letzten beiden Phasen den RSA-Algorithmus selber zu thematisieren und ihn dann anhand eines Beispiels mit kleineren Zahlen selber zu berechnen. Da für Beispiele mit größeren Zahlen die Berechnung per Hand oder Taschenrechner nicht mehr praktikabel ist, wäre es eine Möglichkeit, die Schülerinnen und Schülern Beispiele anhand der online Version „Sage Cell Servers“ oder mit einer eventuelle an der Schule vorhandenen Mathematica Installation berechnen zu lassen. Aufbauen darauf könnten die Schülerinnen und Schüler auch eigene Funktionen oder den kompletten RSA-Algorithmus programmieren. Einzelheiten der Phasen oder auch der Inhalt einer kompletten Phase könnte den Schülerinnen und Schülern durchaus schon bekannt sein. In diesem Fall könnten einzelne Teilbereiche

ausgelassen oder zu Wiederholung genutzt werden. Der Inhalt der einzelnen Phasen soll im folgenden Abschnitt beschrieben werden.

Phase 1: In Phase 1 werden Grundlagen behandelt, die Schülerinnen und Schülern der Sekundarstufe II vom Grundsatz her bekannt sein dürften. Deshalb würde es sich anbieten, diese Phase etwas kürzer zu fassen und zur Wiederholung zu nutzen. Vorausgesetzt wird die Kenntnis der Menge der natürlichen Zahlen $\mathbb{N}$ und deren Erweiterung, die Menge der geraden Zahlen $\mathbb{Z}$. Inhalte der Phase 1 während dann die Division mit Rest und der Modulo-Operator, der größte gemeinsame Teiler und zur Bestimmung des ggT, der euklidische Algorithmus und die Definition von Primzahlen.

Phase 2: Da für das RSA-Verfahren die modulare Potenzierung benötigt wird, wird in Phase 2 Rechnen mit Resten eingeführt. Themen innerhalb dieser Phase wären also, Restklassen, modulare Addition, modulare Multiplikation und Kongruenzen.

Phase 3: Da ein Großteil der für den RSA-Algorithmus benötigten Grundlagen bereits gelegt ist, bietet es sich an in Phase 3 den kleinen Satz von Fermat und den Satz von Euler als dessen Verallgemeinerung einzuführen. In Abhängigkeit von der Gruppe könnten auch die Beweise der beiden Sätze Teil der Phase 3 sein.

Phase 4: Phase 4 beinhaltet die Theorie des RSA-Algorithmus. Also die Konstruktion der Schlüssel und des ver- und entschlüsseln. Im theoretischen Teil der Schlüsselerzeugung bietet sich die Möglichkeit, den euklidischen Algorithmus zum erweiterten euklidischen Algorithmus zum Finden der modularen Inversen zu erweitern bzw. die Existenz einer modularen Inversen zu vermitteln.

Phase 5: In dieser Phase sollen die Schülerinnen und Schüler das bereits gelernte umsetzen und den RSA-Algorithmus anhand eines Beispiels selber durchführen. Um den Ablauf und die Berechnung per Hand nachzuvollziehen, wäre die einfachste Form, eine einstellige Zahl zu verschlüsseln.

Damit die Berechnung per Hand bzw. mit dem Taschenrechner durchgeführt werden kann, sollten die Primzahlen zur Schlüsselerzeugung relativ klein gewählt werden. Die Aufgabe soll dazu dienen, den Schülerinnen und Schülern den Ablauf des Verfahrens zu verdeutlichen. Weitere Aufgaben wären die Verschlüsslung einer kurzen Textnachricht (z. B. ein Satz) und die Schlüsselerzeugung mit größeren Primzahlen z. B. mittels des *Sage Cell Servers* oder mittels CrypTool (siehe Abschnitt 7). Verschiedene Möglichkeiten für Beispiele zum Berechnen per Hand werden in Abschnitt 5.7 aufgeführt.

Phase 6: In der sechsten Phase sollen die Schülerinnen Schüler den RSA-Algorithmus oder Einzelteile des Algorithmus selber als Programm umsetzen.

6.3.3 Schnelles modulares Potenzieren mit Square & Multiply

Um mit dem RSA-Verfahren zu arbeiten, bzw. es zu implementieren oder ein Beispiel zu berechnen, müssen die Schülerinnen und Schüler neben der modularen Arithmetik, das Rechnen mit Potenzen beherrschen. Da beim Verschlüsseln mit dem RSA-Algorithmus sehr große Exponenten genutzt werden, mit denen das Rechnen sehr schnell ineffizient bzw. bei Nutzung eines Taschenrechners unmöglich wird, brauchen die Schülerinnen und Schüler ein Verfahren, mit dem sie einfach und effizient das modulare Potenzieren umsetzen können. Eine Methode die sich anbietet, ist das Square and Multiply Verfahren. Bei diesem Verfahren wird schrittweise der Exponent halbiert und eventuell abgerundet und die Basis quadriert. Um das Ergebnis zu erhalten werden alle geraden Exponenten gestrichen und die Basen der ungeraden Exponenten aufmultipliziert. Für modulares Potenzieren wird zusätzlich nach jedem Quadrieren und Multiplizieren die modulo Division durchgeführt. Das Square und Mulltiply-Verfahren eignet sich sowohl für die Berechnung per Hand, als auch zur Implementierung.

Beispiel für die Berechnung per Hand:

$4^{13} \; mod \; 25$	
13	4
~~6~~	~~16~~
3	6 (256 mod 25)
1	11 (65536 mod 25)
Ergebnis	14 (264 mod 25)

6.4 Experimente zum RSA-Verfahren

Experimente bieten sich vor allem im Zusammenhang mit der Sicherheit des RSA-Verfahrens an. Die Schülerinnen und Schüler können zum Beispiel damit experimentieren, welche Faktoren welchen Einfluss auf die Sicherheit des Verfahrens haben. Denkbar sind hier mehrere Szenarien, wie die eigentliche Größe des Schlüssels oder die Auswahl der zur Schlüsselerzeugung nötigen Primzahlen. Auch Experimente im Bereich des Findens von Primzahlen sind sinnvoll. Die Schülerinnen und Schüler können so ein Gefühl dafür bekommen, wie die Sicherheit des RSA-Verfahrens gewährleistet werden kann. Der folgende Abschnitt soll einige Konzepte zu Experimenten zum RSA-Verfahren darstellen, die in die Bereiche Experimente zu Faktorisierung und Experimente zum Finden von Primzahlen / Primzahltests unterteilt sind.

6.4.1 Experimente zur Faktorisierung

Um Experimente im Bereich der Faktorsierung durchzuführen ist es sinnvoll, dass den Schülerinnen und Schülern zumindest ein oder mehrere Faktorisierungsverfahren bekannt sind. Abhängig von der zu Verfügung stehenden Zeit reicht es unter Umständen nur oberflächlich darauf einzugehen, damit die Schülerinnen und Schülern durch die Experimente lernen, wie hoch der zeitliche Aufwand zu Faktorisierung unter Umständen sein kann. Grundsätzlich ist es ein Bereich, der wichtig ist, um Aussagen über die Sicherheit des RSA-Verfahrens treffen zu können. Durch Experimente zur Faktorisierung können sich die Schülerinnen eine Vorstel-

lung davon verschaffen, wie schnell mit heutigen Rechenkapazitäten relativ große Zahlen faktorisiert werden können und welche Verfahren zu Faktorisierung von bestimmen Zahlen besonders effektiv sind.

Für Experimente zu Faktoriserung eignet sich besonders die Software CrypTool. In Cryptool sind verschiedene Faktorisierungsmethoden, wie Brute Force, das Quadratische Sieb oder die Methode nach Brent implementiert, die mit selbstgewählten Primzahlen getestet werden können.

Cryptool bietet auch die Möglichkeit Primzahlen einer bestimmten Länge erzeugen zu lassen. Nach erfolgreicher Faktorisierung kann man sich eine detaillierte Auswertung bezüglich der benötigten Zeit und der Durchläufe der gewählten Methode anzeigen lassen. Die Schülerinnen und Schüler können so Experimente bezüglich der Laufzeit und der Effektivität der einzelnen Methoden durchführen und diese auswerten. Es bieten sich auch Experimente bezüglich des Unterschieds zwischen einfachen zusammengesetzten Zahlen und den für das RSA-Verfahren gebräuchlichen Semiprimzahlen an.

Ein weiteres Experiment, das für die Schülerinnen und Schüler sehr motivierend sein kann, ist der Vergleich bezüglich der Laufzeit und der Effektivität von selber implementierten Faktorisierungmethoden. Hier könnten in Gruppenarbeit verschiedene Faktorisierungsmethoden implementiert werden, mit denen dann Experimente zur Laufzeit durchgeführt werden können.

Ein interessantes Experiment, das nicht direkt nicht der Faktorisierung zu hat, aber zum Bereich Sicherheit des RSA-Verfahrens gehört, wird im Cryptool Skript[33] beschrieben. Die Schülerinnen und Schüler könnten Überlegungen anstellen, wie viele private Schlüssel es in einem bestimmen Bereich von Primzahlen gibt und ob es sinnvoll bzw. effizient wäre, das RSA-Verfahren über diesen Ansatzpunkt zu knacken.

6.4.2 Experimente zu Finden von Primzahlen / Primzahltests

Ähnlich wie bei den Faktorisierungsmethoden bieten sich auch für Primzahltests Experimente an. Hier sind es allerdings nicht nur Experimente zu Effektivität der

[33] Esslinger, 2013 S.183f

Primzahltests, sondern es stellt sich auch die Frage, mit welcher Wahrscheinlichkeit werden echte Primzahlen generiert.

Bevor Experimente zu und mit Primzahltests gemacht werden können, sollten den Schülern einige Verfahren bekannt gemacht werden. Neben der bereits bekannten Probedivision, die sowohl zur Faktorisierung als auch zum Testen von Primzahlen verwendet wird, bietet sich das Sieb Eratosthenes an. Die Schülerinnen und Schüler werden allerdings schnell erkennen, dass diese beide Verfahren für Primzahlen der Größenordnung, die im das RSA-Verfahren gebräuchlich sind, zu ineffizient arbeiten. Das Sieb des Eratosthenes bietet allerdings den Vorteil, dass es sehr einfach zu verstehen und gut zu visualisieren ist.

Effizienter sind da die Primzahltests die auf dem den Schülerinnen und Schülern bereits bekannten kleinen fermatschen Satz beruhen. Deshalb bietet es sich an, die Schülerinnen und Schülern mit dem Primzahltest nach Fermat und dessen Weiterentwicklung, dem probabilistischen Miller-Rabin-Test bekannt zu machen.

Es besteht nun die Möglichkeit die Schülerinnen und Schüler selber einen Primzahltest implementieren zu lassen, der dann durch Experimente mit verschiedenen Zahlen auf Effizienz und Korrektheit getestet werden kann.

Eine weitere Möglichkeit besteht darin, die Schülerinnen und Schülern Experimente mittels Cryptool und Cryptool 2 durchführen zu lassen. In Cryptool 2 sind unter anderem das Sieb des Eratosthenes und der Miller-Rabin-Test so implementiert, dass nicht nur das Ergebnis angezeigt wird, sondern auch eine Visualisierung des Tests. So lassen sich die Schritte der Tests für die Schülerinnen und Schüler gut nachvollziehen. Gerade beim Miller-Rabin-Test ist es sinnvoll, die gleiche zu testenden Zahl mit verschiedenen Basen und unterschiedlicher Rundenzahl zu testen, um Aussagen über die Effizienz und Leistungsfähigkeit bezüglich der Korrektheit des Ergebnisses treffen zu können. Um die Tests zu „überlisten“ und die Korrektheit bei verschiedener Anzahl von Durchläufen zu testen, bieten sich bekannte Carmichael-Zahlen an.

6.5 Möglichkeiten für verschiedene Beispiele des RSA-Verfahrens im Unterricht

Nachdem die grundsätzliche Funktionsweise vermittelt wurde, kann man den Schülerinnen und Schülern anhand von Beispielen mit konkreten Zahlen den RSA-Algorithmus vorführen oder sie selber als Aufgabe selber berechnen lassen.
Für Beispiele mit konkreten Zahlen bieten sich mehrere Möglichkeiten, die sich in Schwierigkeit und Berechnungsaufwand unterscheiden. Grundsätzlich bieten sich folgende Möglichkeiten an, die ähnlicher Art im CrypTool-Script[34] beschrieben sind:

6.5.1 Beispiel mit kleinen Primzahlen und einer kleinen Zahl als Nachricht

Für die einfachste Variante der Durchführung des kompletten RSA-Algorithmus ist die Auswahl von sehr kleinen, also einstelligen oder niedrigen zweistelligen, Primzahlen erforderlich. Da die zu verschlüsselnde Nachricht kleiner sein muss als das RSA-Modul damit die Ver- und Entschlüsselung korrekt funktioniert, ist man auch bezüglich der Nachricht eingeschränkt. Um das Beispiel möglichst übersichtlich zu halten, soll als Nachricht nur eine natürliche Zahl verschlüsselt werden. Falls das Beispiel per Hand oder per Taschenrechner berechnet werden soll ist darauf zu achten, dass die Potenzierung der Nachricht mit dem errechneten Schlüssel nur solche Ergebnisse liefert, die mit dem Taschenrechner korrekt dargestellt werden können. Erfolgt die Berechnung in einer Programmiersprache mit Langzahlarithmetik oder einem Tool wie dem *Sage Cell Server,* sind diesbezüglich keine Einschränkungen nötig. Der Vorteil dieser Variante ist, dass die Primzahlen leicht zu finden oder aus einer Primzahlentabelle abzulesen sind und das bei einer kleinen Zahl als Nachricht auf eine Vorcodierung oder Blockbildung der Nachricht verzichtet werden kann.

Eine Aufgabe für die Schülerinnen und Schüler könnte folgendermaßen aussehen: „ Konstruiere aus den Primzahlen $p = 5$ und $q = 11$ das für das RSA-Verfahren benötige Schlüsselpaar und führe die Verschlüsslung und die Entschlüsselung der Nachricht $m = 2$ durch.“

[34] Esslinger, 2013

Bei einer Aufgabe dieser Form wäre ein teilerfremder öffentlicher Schlüssel leicht zu finden und das modulare Inverse einfach zu bestimmen. Natürlich kann auch bei einer solchen Aufgabe der erweiterte euklidische Algorithmus zu Bestimmung genutzt werden. Auch etwas größere Primzahlen und eine zu verschlüsselnde Zahl sind als Alternative möglich.

6.5.2 Beispiel mit etwas größeren zweistelligen Primzahlen und einem kurzen Text in Großbuchstaben als Nachricht

Eine etwas anspruchsvollere und realitätsnähere Möglichkeit wäre ein Beispiel mit zweistelligen Primzahlen und einen Wort oder kurzen Text aus Großbuchstaben als Nachricht. Bei der modularen Potenzierung der Nachricht mit zweistelligen Schlüsseln und einem vierstelligen Modul ergeben sich schon relativ große Zahlen, die für die Berechnung per Hand nicht mehr praktikabel sind. Die zu verschlüsselnden Textnachricht macht es erforderlich, diese Nachricht mit Zahlen zu codieren. Da die Nachricht nur aus Großbuchstaben und Leerzeichen besteht, kommt man allerdings mit wenigen Zeichen aus. Eine Tabelle zum Codieren der Nachricht könnte zum Beispiel so aussehen:

Zeichen	Zahlencode	Zeichen	Zahlencode	Zeichen	Zahlencode
Leerzeichen	00	J	10	T	20
A	01	K	11	U	21
B	02	L	12	V	22
C	03	M	13	W	23
D	04	N	14	X	24
E	05	O	15	Y	25
F	06	P	16	Z	26
G	07	Q	17		
H	08	R	18		
I	09	S	19		

Nachdem die Schülerinnen und Schüler die Schlüssel erzeugt haben, müssen sie im nächsten Schritt zuerst die Textnachricht codieren um diese verschlüsseln zu können. Je nach Größe des RSA-Moduls wäre es auch möglich die codierten Buchstaben in Blöcke von z. B. zwei direkt aneinander gehängten Zahlencodes zusammen zu fassen, um Berechnungsschritte beim Ver- und Entschlüsseln einzusparen. Das RSA-Modul müsste dazu größer als 2626, den Code für zwei aufein-

anderfolgende *Z*, sein. Die Nachricht muss dementsprechend nach der Entschlüsselung wieder zum eigentlichen Text anhand der Tabelle decodiert werden. Eine Aufgabe für die Schülerinnen und Schüler mit vorgegeben Primzahlen und einer zu verschlüsselnden Nachricht könnte folgendermaßen aussehen: „Konstruiere aus den Primzahlen p = 47 und q = 79 die für die Verschlüsselung mit dem RSA-Verfahren nötigen Schlüssel. Codiere anschließend die Nachricht „RSA FUNKTIONIERT“ anhand der Codetabelle und führe mit den Codes die Verschlüsselung durch.“

Für ein umfangreicheres Beispiel ist es auch möglich, die Schülerinnen und Schüler Primzahlen aus einem vorgegebenen Bereich selbständig suchen zu lassen. Als Erweiterung der Aufgabe ist auch die Erstellung einer eigenen Tabelle zur Codierung und natürlich die Verschlüsselung eines eigenen Textes, der innerhalb von bestimmten Vorgaben bleibt, möglich. Für die Berechnung der einzelnen Schritte bietet sich wiederum der *Sage Cell Server* oder ein von den Schülerinnen und Schülern selber implementiertes Programm für den RSA-Algorithmus an.

6.5.3 Beispiel mit großen Primzahlen und einem Text der mittel ASCII-Tabelle codiert wird

Da in der Praxis Nachrichten die mittels RSA-Verfahren verschlüsselt werden mit dem ASCII-Code codiert werden, ist sinnvoll, die Schülerinnen und Schülern ein Beispiel dieser Art berechnen zu lassen. Die Einzelzeichen der Nachricht werden dazu in 8-Bit lange Zahlen codiert. Je nachdem, ob man die Zeichen in Blöcke zusammenfassen möchte oder nicht, braucht man ein entsprechend großes RSA-Modul. Denn zwei zusammengefasste Zeichen aus dem ASCII-Code sind 16-Bit lang und ergeben umgerechnet in das für das RSA-Verfahren gebräuchliche Dezimalsystem einen Wert von bis zu 65535. Das Modul müsste dementsprechend mindestens 65526 sein.

Für ein Beispiel dieser Art wären dann z. B. dreistellige Primzahlen sinnvoll. Bezüglich einer Aufgabe wären dann mehrere Varianten denkbar. Wenn man keine Zusammenfassung der einzelnen Buchstaben der Nachricht möchte, kann man direkt den ASCII-Dezimalcode für den entsprechende Zeichen verwenden. Sollen

Buchstaben als Blöcke zusammenfassen, nimmt man entweder den ASCII-Binärcode oder den ASCII-Dezimalcode.

Auszug aus der ASCII-Tabelle		
Dezimal	Binär	Zeichen
82	01010010	R
114	01110010	r
32	00100000	Leerzeichen
57	00111001	9

Zwei im ASCII-Binärcode codierte Zahlen kann man als Block aneinanderhängen und ins Dezimalsystem umrechnen. Die Dezimalzahl kann man anschließend verschlüsseln.

Beispiel:

- zu codierende Zeichen „R“ und „!“
- R ist als ASCII-Binärcode 01010010
- ! ist als ASCII-Binärcode 00100001
- als Block zusammengesetzter Code: 0101001000100001
- in Dezimalschreibweise 21025

Nach erfolgreicher Ver- und Entschlüsselung kann die Dezimalzahl wieder in den Binärcode umgewandelt werden, der dann wieder in zwei 8-Bit Zeichen getrennt und in die eigentlichen Zeichen umgewandelt werden kann.

Eine weitere Variante wäre es, den ASCII-Dezimalcode von 2 oder mehr Zahlen aneinanderzuhängen. Je nach Größenordnung des Modul ergeben sich verschiedene Möglichkeiten. Fasst man zwei im dreistelligen ASCII-Dezimalcode codierte Zeichen zusammen, ist ein Modul > 255255 notwendig, wenn alle Zeichen des ASCII-Codes genutzt werden können.

Beispiel:

- zu codierende Zeichen „u“ und „z“
- u ist als ASCII-Dezimalcode 117
- z ist als ASCII-Dezimalcode 122
- als Block zusammengesetzter Codes: 117122
- dieser Code kann direkt verschlüsselt werden

Nach erfolgreicher Ver- und Entschlüsselung kann der Block wieder in zwei dreistellige Zahlen getrennt werden und anhand der ASCII-Tabelle in die ursprüngliche Nachricht decodiert werden.

Ein etwas umfangreichere Aufgabe für die Schülerinnen und Schüler könnte mehrere Fragestellungen enthalten. Wird den Schülern den Schülerinnen und Schülern ein Satz wie z. B. „RSA ist genial!“ zur Verschlüsselung vorgegeben und sonst keine weiteren Vorgaben gemacht, müssen sie sich selber Gedanken zur Wahl der Primzahlen, des nötigen Moduls und der Art der Vorcodierung und eventueller Blockbildung machen. Es wäre auch möglich, den Schülerinnen und Schülern Vorgaben zu machen, mit denen die Verschlüsselung nicht korrekt funktioniert (z. B. zu kleine Primzahlen). So können sie analysieren warum das Verfahren nicht korrekt funktioniert, die Fehler finden und beheben.

6.5.4 Kryptoanalyse

Für ein Beispiel oder eine Aufgabe aus dem Bereich der Kryptoanalyse sind verschiedene Szenarien denkbar. Eine einfache Möglichkeit wäre es, den Schülerinnen und Schülern in einer Aufgabe einen verschlüsselten Text und den passenden öffentlichen Schlüssel zur Verfügung zu Stellen. Anhand der ausgehändigten Informationen sollen sie versuchen diesen Text zu entschlüsseln. Das Programm Cryptool und das CrypTool-Skript[35] beschreiben mehrere mögliche Optionen mit unterschiedlichen Schwierigkeitsgraden. Eine Möglichkeit ist das Knacken des Codes durch Faktorisierung des RSA-Moduls. Eine Aufgabe dieser Art könnte entweder mit einer selbständig implementierten Faktorisierungsfunktion oder mit Hilfe der in CrypTool vorhandenen Funktionen gelöst.

Verschiedene Schwierigkeitsgrade können sich auch daraus ergeben, ob die zu verschlüsselnde Nachricht nur aus einer Zahl oder mehreren Zahlen besteht oder aus Buchstaben, die in Zahlen codiert wurden. Eine zusätzliche Hürde wäre es, die korrekte Codierungstabelle zu finden, z. B. durch eine Häufigkeitsanalsyse der Buchstaben.

[35] Esslinger, 2013

Zwei fertige Beispiele für so eine so genannte Chiper-Challenge finden sich im CrypTool Skript[36].

7 Tools und Literaturempfehlungen für die Vermittlung des RSA-Verfahrens im Informatikunterricht

7.1 Cryptool

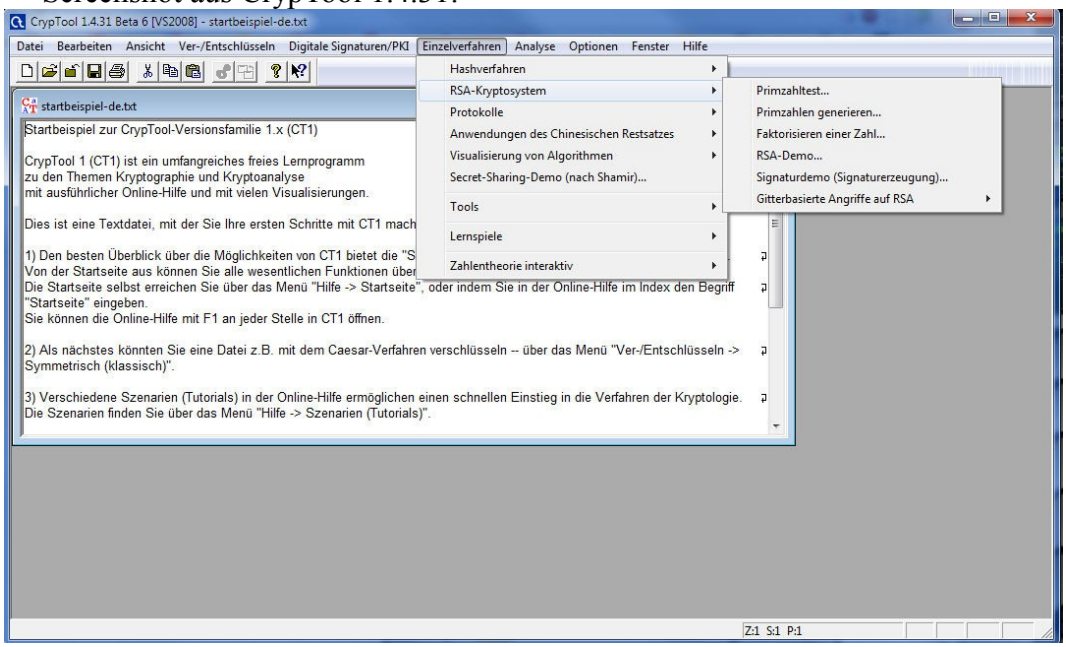
CrypTool ist eine freie E-Learning-Software für Kryptographie und Kryptoanalyse, die mittlerweile in mehreren Versionen erschienen ist. Im folgenden Abschnitt sollen die beiden Versionen, die für zur Vermittlung des RSA-Verfahrens am geeignetsten sind und deren Merkmale kurz aufgeführt werden.

7.1.1 CrypTool 1

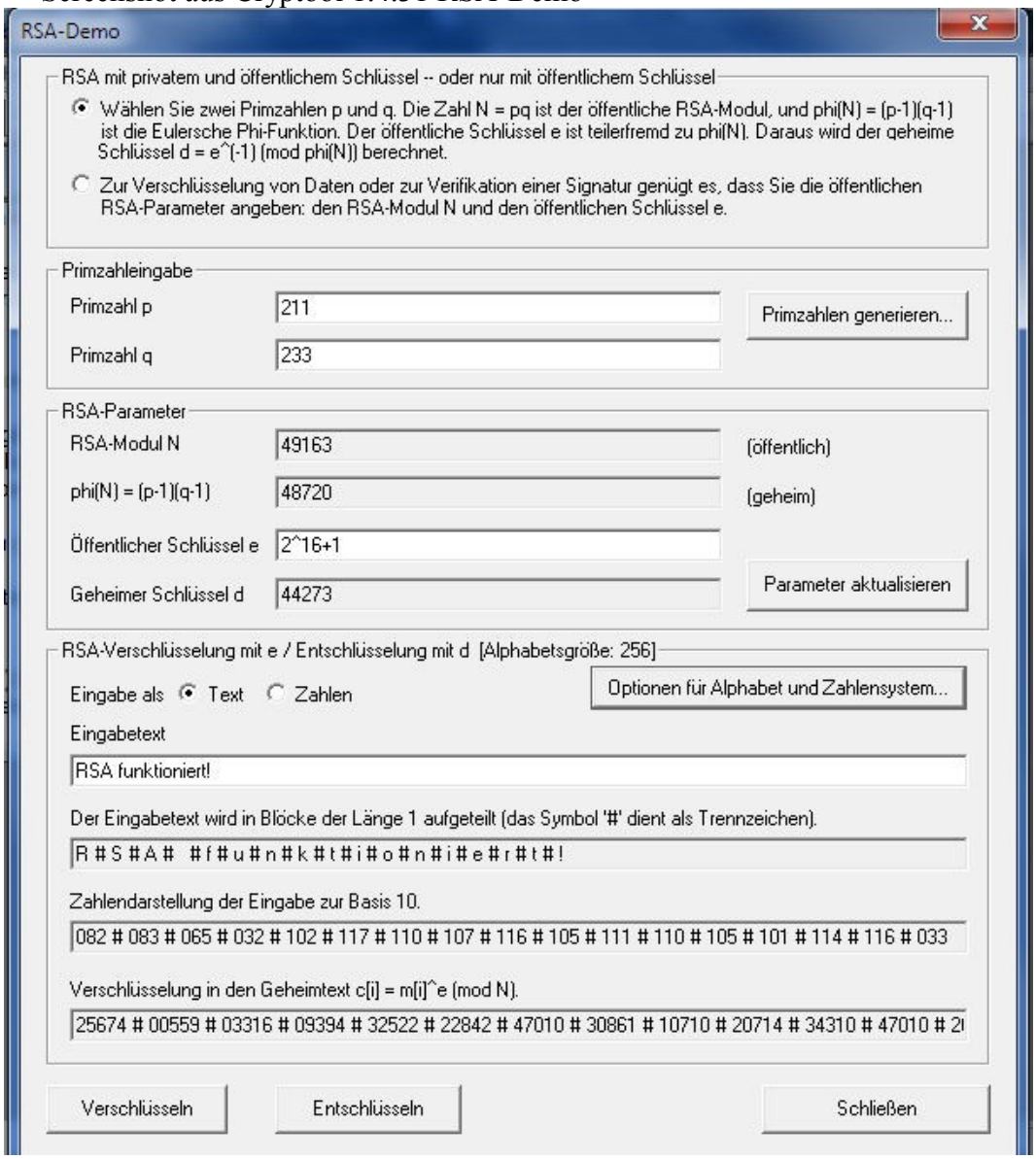
CrypTool 1 ist ein kostenloses Open-Source-Programm für Windows-Betriebssysteme, das erstmals im Jahr 1998 erschienen ist. Die aktuelle Version CrypTool 1.4.31 Beta 6 bietet zahlreiche Informationen zu klassischen und modernen kryptographischen Verfahren, einige Visualisierungen zu kryptographischen Verfahren, eine umfangreiche Online-Hilfe und ein Lernprogramm zur Zahlentheorie. Für den Bereich RSA-Verfahren sind einige interessante Funktionen implementiert. Zu den wichtigsten gehören hier verschiedene Primzahltests, wie der Fermat-Test und der Miller-Rabin-Test und die Implementierung von einigen Methoden zur Faktorisierung und deren Analyse. Des Weiteren gibt es ein RSA-Demo mit dem die Schülerinnen und Schüler den kompletten RSA-Algorithmus, von der Suche nach geeigneten Primzahlen bis zur Ver- und Entschlüsselung, durchführen können. Zusätzlich können verschieden Szenarien für Angriffe auf das RSA-Verfahren durchgeführt werden. Die Entwicklung von CrypTool 1 ist abgeschlossen, es werden nur noch Bugfix-Updates durchgeführt. Weitere Information und die aktuelle Version von CrypTool 1 finden sich unter http://www.cryptool.org/de/cryptool1.

[36] Esslinger, 2013 S. 153ff

Screenshot aus CrypTool 1.4.31:



Screenshot aus Cryptool 1.4.31 RSA-Demo

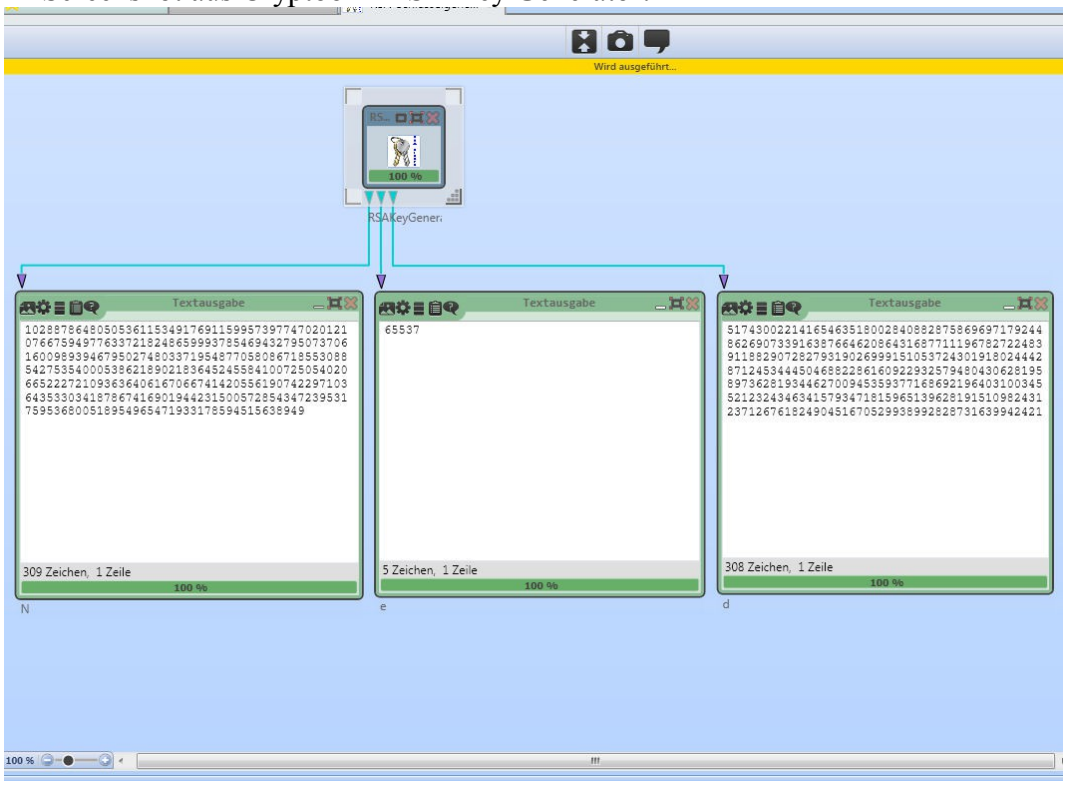


7.1.2 CrypTool 2

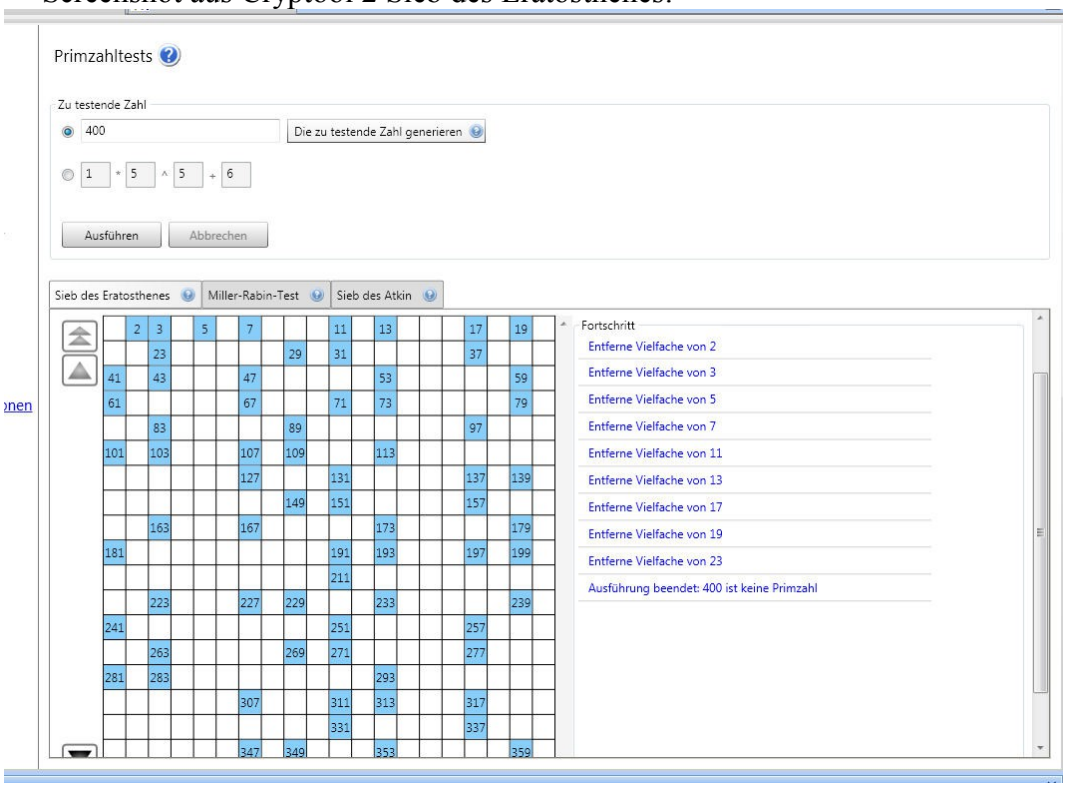
CrypTool 2 ist ein ebenfalls kostenloses Open-Source-Programm und die aktuelle und etwas zeitgemäßer gestaltete Version von CrypTool für Windows-Betriebsysteme. Es enthält derzeit über 100 Funktionen zur Kryptographie und Kryptoanalyse. Diese Version bietet neben visueller Programmierung und der Visualisierung von Algorithmen vielfältige Möglichkeiten sich mit dem Thema Kryptographie auseinanderzusetzen. Wie in CrypTool 1 ist eine umfangreiche Online-Hilfe inte-

griert. Das sogenannte Kryptotutorium bietet im Bereich „Die Welt der Primzahlen“ sehr viele Informationen aus dem Bereich der Primzahlen, die besonders für die Vermittlung des RSA-Verfahrens im Informatikunterricht nützlich sind. In diesem Tutorium lassen sich Faktorisierungsverfahren wie die Probedivision und das Quadratische Sieb und Primzahltests wie das Sieb des Eratosthenes und der Miller-Rabin-Test nutzen und visualisieren. Im Abschnitt Zahlentheorie lassen sich einige zahlentheoretische Funktionen wie z. B. die Anzahl der Primzahlen in einem bestimmten Bereich oder die eulersche Phi-Funktion berechnen. Weitere Informationen und die aktuelle Version von CrypTool 2 finden sich unter http://www.cryptool.org/de/cryptool2.

Screenshot aus Cryptool 2 RSA Key Generator:



Screenshot aus Cryptool 2 Sieb des Eratosthenes:



7.2 Sage

Sage ist ein kostenloses Open-Source Computer-Algebra-System das sich auf verschiedene Weise nutzen lässt. Es kann lokal installiert oder in Form des Sage-Notebooks online genutzt werden. Zusätzlich gibt es die Möglichkeit, den im Funtionsumfang eingeschränkten Sage Cell Server online zu nutzen. Sage lässt sich z. B. als funktionsreicher Taschenrechner und Programmierumgebung nutzen. Im Zusammenhang mit dem RSA-Verfahren lässt sich Sage nutzen um Zahlen zu faktorisieren oder die modulare Potenzierung durchzuführen, da diese und andere Funktionen bereits implementiert sind. Die Schülerinnen und Schüler können so mit dem RSA-Verfahren experimentieren ohne die nötigen Funktionen selber umsetzten zu müssen. Im CrypTool-Skript finden sich einige Beispiele. Weitere Infomationen finden sich unter http://www.sagemath.org/.

Screenshot der Funktion factor() im Sage Cell Server [37]:

7.3 Übersicht über Literaturempfehlungen zur Vermittlung des RSA-Verfahrens

In diesem Abschnitt wird eine kurze Übersicht über verschiedene Unterrichtsmodule gegeben, die zur Vermittlung des RSA-Verfahren im Informatikunterricht genutzt werden können. Einige dieser Empfehlungen wurden bereits in den vorherigen Abschnitten angesprochen.

Sehr interessant und umfangreich ist die Reihe „RSA & Co. In der Schule“ von Helmut Witten und Ralph-Hardo Schulz, die mittlerweile aus 6 neuen Folgen besteht und in der Zeitschrift LOG IN erschienen sind.

Ideen zur Faktorisierung enthalten die ebenfalls von Witten/Schulz verfassten in LOG IN erschienen Artikel „Zeit-Experimente zu Faktorisierung“[38] und „Faktorisieren mit dem Quadratischen Sieb“[39]

Sehr umfassend im Bereich des mathematischen Hintergrunds und der Bedienung von Sage und CrypTool ist das „CrypTool-Skript“[40] von Esslinger.

Weitere Empfehlungen:

- Kann man RSA-Vertrauen?[41]
- RSA – Primzahlen zu Verschlüsselung von Nachrichten[42]

[37] Esslinger, 2013
[38] Ralph-Hardo Schulz & Witten, 2010
[39] Witten & Schulz, 2012
[40] Esslinger, 2013
[41] Witten, Esslinger, Gramm & Hornung, 2012
[42] Schüller, Trottenberg, Wienands, Koziol & Schneider, 2013

- Kryptographie verstehen – Ein schülergerechter Zugang zum RSA-Verfahren[43]
- Lauschen zwecklos! oder Wie aus einer seltsamen Erkenntnis über Zahlen die beste Geheimsprache aller Zeiten wurde[44]

8 Fazit

Abschließend lässt sich festhalten, dass das RSA-Verfahren durchaus Teil heutigen Informatikunterrichts ist. Es scheint allerdings auch so zu sein, dass das RSA-Verfahren nicht immer mit der Intensität behandelt wird, die zu einem umfassenden Verständnis des Algorithmus und der sich anschließenden Fragestellungen zur Sicherheit eigentliche nötig wäre. Die Gründe hierfür können vielfältig sein. Mit großer Wahrscheinlichkeit ist die zur umfassend Vermittlung fehlende Unterrichtszeit einer der Hauptgründe. Die Schaffung der nötigen mathematischen Grundlagen, die größtenteils nicht im Mathematikunterricht gelegt werden, ist allerdings durch die Konzentration auf das Wesentliche und durch sorgfältig auf die jeweilige Lerngruppe angepasste Methoden, durchaus auch durch den Informatikunterricht zu leisten. Es gilt hier die richtige Balance zu finden, die die Mathematik nicht offensichtlich in den Vordergrund stellt bzw. mit Fragestellungen die dem Informatikunterricht angemessen sind, verknüpft. Auch die Schülerinnen und Schüler die den mathematischen Hintergrund unter Umständen nicht gänzlich verstehen, sollen die Möglichkeit haben, den RSA-Algorithmus zu verstehen und Aussagen über seine Sicherheit treffen können.

Es lässt sich auch festhalten, dass durchaus einige sehr gute mehr oder weniger gut ausgearbeitete Unterrichtseinheiten und Module zum RSA-Verfahren vorhanden sind. Diese Einheiten sind allerdings größtenteils stark auf den mathematischen Hintergrund ausgerichtet. Als Anregung für eigene Einheiten, die an die Be-

[43] Puhlmann, 1998
[44] Kirchgraber & Kramer, 2005

dürfnisse, Kenntnisse, dem Vorwissen der jeweiligen Lerngruppe und der zur Verfügung stehenden Unterrichtszeit angepasst sind, können einige dieser Module gut verwendet werden. Gerade die Reihe RSA & Co von Witten/Schulz und CrypTool 1 und 2 bieten viele Möglichkeiten und Ideen für eigene, an die jeweilige Lerngruppe angepasste Einheiten. Um das Thema RSA umfassend zu behandeln, ist allerdings sehr viel Zeit und Motivation der Schülerinnen und Schüler und des unterrichtenden Lehrers nötig. Bleibt festzuhalten, dass das RSA-Verfahren ein wichtiges Thema für den Informatikunterricht ist, dass die Lebenswelt der Schülerinnen und Schüler direkt betrifft und an dem viele der Inhalts- und Prozessbereiche der Informatik vermittelt werden können.

9 Anhang

Anhang I: Fragebogen zum Thema RSA-Verfahren im Informatikunterricht

Fragen und Grundauswertung der Antworten (unbearbeitet):

1) Bitte tragen Sie hier Ihre Schulnummer ein.

2) Bitte tragen Sie die Folgenummer aus dem Anschreiben ein.

3) Falls Sie für Rückfragen zu Verfügung stehen, tragen Sie hier Ihre E-Mail Adresse ein.

4) Ist das RSA-Verfahren Gegenstand Ihres Unterrichts?

Ja	6	(75,00%)
Nein	2	(25,00%)
Summe	8	
ohne Antwort	0	

5) Falls Ihre Antwort bei Frage 4 "Nein" lautet: Aus welchen Gründen ist es nicht Teil Ihres Unterrichts?

<4/14> Das Thema Kryptographie wird nicht bis in die Tiefen von RSA behandelt

<6/16> SEK2: Wir behandeln Datenbanken und Komplexitätstheorie. Deshalb wird das Thema Netzwerke (welches auch Kryptographie beinhaltet) weggelassen.

SEK1: Zu kompliriert aus meiner Sicht. Ich bin immernoch auf der Suche nach einem einfacheren asymmetrischen Verfahren, dass nicht trivial zu knacken ist.

6) In welchen Kursen behandeln Sie das RSA-Verfahren? (Mehrfachauswahl möglich)

Leistungskurs	2	(33,33%)
Grundkurs	6	(100,00%)
Wahlbereich SI	2	(33,33%)
Sonstige	1	(16,67%)
Nennungen (Mehrfachwahl möglich!)	11	
geantwortet haben	6	
ohne Antwort	2	

7) Wie wichtig ist es, Ihrer Ansicht nach, das Thema Kryptographie im Informatikunterricht zu behandeln?

unwichtig	0	(0,00%)
	0	(0,00%)
	0	(0,00%)
	6	(75,00%)
sehr wichtig	2	(25,00%)
Summe	8	
ohne Antwort	0	
Mittelwert	1,25	
Median	1	

8) Für wie relevant halten Sie das Thema RSA im Informatikunterricht?

irrelevant	0	(0,00%)
	0	(0,00%)
	3	(37,50%)
	5	(62,50%)
sehr relevant	0	(0,00%)

Summe	8
ohne Antwort	0
Mittelwert	0,63
Median	1

9) Wo sehen Sie konkrete Probleme bei der Vermittlung des RSA-Verfahrens im Informatikunterricht? (Mehrfachauswahl möglich)

Verständnisschwierigkeiten der Schüler	3(37,50%)
Verständnisschwierigkeiten des Lehrers	1(12,50%)
fehlendens mathematisches Vorwissen der Schüler (allg.)	6(75,00%)
spezielle mathematische Probleme	4(50,00%)
fehlendes/unzureichendes Unterrichtsmaterial	3(37,50%)
Vermittlungsprobleme	0(0,00%)
sonstige	1(12,50%)

Nennungen (Mehrfachwahl möglich!)	18
geantwortet haben	8
ohne Antwort	0

Sonstige: <8/18> Ich sehe nicht mehr Probleme beim RSA-Verfahren als bei einigen anderen Themengebieten im Informatikunterricht. Ich hoffe, dass das Feld \"Verständnisschwierigkeiten des Lehrers\" nicht ernst gemeint war. Als Lehrer sollte man doch die Zeit investieren.

10) Welche Bereiche der Mathematik erachten Sie als schwierig? Welche mathematischen Probleme sehen Sie konkret? (Bezogen auf das RSA-Verfahren)

<1/11> Modulo-Rechnung wird nicht in der Schule gelehrt, ebenso wird der erweiterte euklidische Algorithmus nicht gelehrt, keine automatisierten Tools für die erforderlichen Rechnungen verfügbar.

Zu 11: Das Wissen kann im Fach Informatik vermittelt werden, braucht aber sehr viel zeit und könnte daher andere Kollegen abschrecken.

<2/12> Potenzrechnung
Modulo-Rechnung

<3/13> Der gesamte, hier benötigte, Bereich Zahlentheorie wird im Mathematikunterricht in der Regel nicht behandelt, muss also im Informatikunterricht mitbehandelt werden.

<5/15> Da Beweise ausgelassen werden (z.B. Satz von Euler), halten sich die Probleme in Grenzen.

<6/16> Zahlentheorie

<7/17> Ich erkläre das Verfahren nur allgemein, damit die SuS allgemeine Vorstellung bekommen haben. Sonst soll man zuerst mod( ) definieren, dann eulersche Funktion und so weiter.

<8/18> Am schwierigsten im Themengebiet des RSA-Verfahrens ist vielleicht die Programmierung des erweiterten Euklidischen Algorithmus mit Langzahlarithmetik für die Schlüsselerzeugung. Das schwierigste mathematische Problem in diesem Themenkomplex ist wahrschenlich die Begründung/der Beweis, dass a^(k * \\phi(n)+1)=a mod n für ALLE 0<a<n gilt. Allerdings ist dies auch nicht schwierieger als der Beweis der Hauptsätze der Differential- und Integralrechnung, die im Grundkurs Mathematik in der Q1 behandelt werden. Wieso sollte ein Informatik-Grundkurs leichter sein als ein Mathematik-Grundkurs? Zudem kann ein Schüler diesen Zusammenhang/Satz auch mit Zahlenbeispielen entdecken und anschließend anwenden, ohne seinen Beweis/seine Begründung vollständig zu erfassen, genauso wie die Hauptsätze der Differential- und Integralrechnung im Mathematikunterricht oder andere Beispiele aus dem Mathematikunterricht.

11) Setzt das Thema RSA Ihrer Meinung nach zu viel mathematisches Wissen voraus?

ja	3	(37,50%)
nein	5	(62,50%)

Summe	8

ohne Antwort 0

12) Wie intensiv gehen Sie in Ihrem Unterricht auf den mathematischen Hintergrund ein?

nicht intensiv	2	(33,33%)
	1	(16,67%)
	0	(0,00%)
	2	(33,33%)
sehr intensiv	1	(16,67%)
Summe	6	
ohne Antwort	2	
Mittelwert	2,83	
Median	2	

13) Für wie wichtig halten Sie den mathematischen Hintergrund?

unwichtig	0	(0,00%)
	2	(28,57%)
	1	(14,29%)
	1	(14,29%)
sehr wichtig	3	(42,86%)
Summe	7	
ohne Antwort	1	
Mittelwert	3,71	
Median	4	

14) Welche Literatur nutzen Sie zur Vorbereitung und im Unterricht (bezogen auf das RSA-Verfahren)? (Fachliteratur/Schulbücher/Internetquellen)

<1/11> Eigenes Material, Computer Science unplugged: Public-Key-Verfahren mit Maps
<2/12> Informatik macchiato (Pearson Studium)
<3/13> Internetquellen: unter anderem Wikipedia
<5/15> S. Singh, Codes
B. Schriek, Informatik mit Java, Band III
div. Netzquellen, z.B. http://www.matheprisma.uni-wuppertal.de/Module/RSA/index.htm
<7/17> Internetquellen und mathematische Fachliteratur zum Thema: Primzahlentheorie und ihre Anwendungen.
<8/18> - Einführung in die Kryptographie, Johannes Buchmann, Springer-Verlag
- Elliptische Kurven in der Kryptographie, Annette Werner, Springer-Verlag

15) Sollte man das RSA-Verfahren Ihrer Ansicht nach im Unterricht behandeln? (Eine Antwort der Form "Nein", "Ja, in Jahrgangsstufe ..."Ja, weil..." ist ausreichend.)

<1/11> Ja, in Sek II, da relevant für das Zentralabitur im Bereich Netzwerke
<2/12> Ja, aber nur kurz (eine Doppelstunde)
<3/13> Ja in der Oberstufe. Aber es hat leider eine geringere Priorität als andere Themen.
<4/14> Nein
<5/15> Ja, in 13 (G9)
<6/16> Ja, im Mathematik-LK oder Informatik-LK.
<7/17> Ja, entweder in SI oder SII
<8/18> Ja, denn:
Das RSA-Verfahren eignet sich gut als ergänzendes Thema, um zur Ausgewogenheit des Bildes beizutragen, das den Schülern und Schülerinnen ein Jahr vor ihrer Studienfachwahl vom Fach Informatik vermittelt wird. Insbesondere die enge Beziehung und Verflechtung der beiden Fächer Mathematik und Informatik kann auch hier gut aufgezeigt werden. Gerade das RSA-Verfahren ist gut geeignet, exemplarisch auch die Idee asymmetrischer Kryptosysteme vorzustellen.

16) Welches Unterrichtsmaterial nutzen Sie, wenn Sie RSA unterrichten? Nutzen Sie eigenes Material oder eine vollständig ausgearbeitete Unterrichtseinheit?

<1/11> Eiegenes Material, vor allem Tafelanschriebe, keine ausgearbeitete Unterrichtseinheit

<2/12> Alle Schülerinnen und Schüler halten in der Reihe Netzwerke und Kryptografie Vorträge entweder mit Präsentationsprogramm oder mit Tafel und Arbeitsblättern. Sowohl RSA als auch Diffie Hellmann gehören zu den Themen für Vorträge (Cäsar wird in der EF und Vigenere wird vorher selbst implementiert). Die Schülerinnen und Schüler erhalten als Einstieg Literatur (siehe Punkt 14) und suchen weiteres Material selbst.

<3/13> Eigenes Material mit einigen Beispielen zu Ver- und -Entschlüsselung.

<5/15> Teils, teils., siehe auch unter 14

<7/17> eigenes

<8/18> Ich nutze eigenes Material, das eine vollständig ausgearbeitete Unterrichtseinheit ist (inklusive kleinem Programmierprojekt).

17) Ist das RSA-Verfahren in Ihrem Informatikunterricht eine eigene Unterrichtsreihe oder Teil einer Unterrichtsreihe? Als Antwort würde z.B. reichen: "Eigene Reihe mit ca. XX Unterrichtsstunden" oder "Teil der Reihe XXX mit ca. XX Unterrichtsstunden für

<1/11> Teil der Reihe Kryptologie mit etwa 12 Stunden für das RSA-Verfahren inkl. math. Hintergrund

<2/12> Teil der Reihe Netzwerke und Kryptographie

<3/13> Teil der Reihe zu Kryptographie. mit nur 2 Unterrichtsstunden für das RSA-Verfahren.

<5/15> Teil der Reihe Kryptographie mit ca. 6 Ustd.

<7/17> als Teil der Kriptographie

<8/18> Das RSA-Verfahren ist in den Vorgaben für das Zentralabitur NRW im Bereich der Kryptographie und diese im "Inhaltlichen Schwerpunkt" "Modellieren und Implementieren kontextbezogener Problemstellungen als Netzwerkanwendungen" angesiedelt. Dort wird es an unserem Gymnasium auch unterrichtet.

18) Ist RSA-VerfahrenTeil von Klausuren? In welcher Form/in welchem Umfang?

<1/11> Ja. Erstellen eines Schlüsselpaars, Erläuterung des Verfahrens zum Verschlüsseln oder zum Signieren

<2/12> Nein

<3/13> Nein

<5/15> Nein

<7/17> Nein.

Es wird am Ende des Jahres unterrichtet.

<8/18> Wählt ein Gymnasium "Modellieren und Implementieren kontextbezogener Problemstellungen als Netzwerkanwendungen" als inhaltliches Gebiet für die Oberstufen-Informatik aus, so werden die inhaltlichen Schwerpunkte, die dort aufgeführt sind, also u. a. das RSA-Verfahren, im Unterricht behandelt. Damit ist es möglich, dass die enthaltenen Inhalte in Klausuren vorkommen können, also auch das RSA-Verfahren.

19) Was für Unterrichtsmaterialien wünschen Sie sich für das RSA-Verfahren im Informatikunterricht (z.B. vollständig ausgearbeitete Einheit, Hilfestellungen zu konkreten Bereichen, Literaturhinweise, Arbeitsblätter)?

<1/11> Ausgearbeitete Einheit inkl. des mathematischen Hintergrunds und Beispielen zur praktischen Anwendung des Verfahrens in Übungsaufgaben, nicht nur als Aufzählung

<2/12> Arbeitsblätter

<3/13> Hilfestellung zu den mathematischen Grundlagen mit Arbeitsblättern z.B. zum erweiterten euklidischen Algorithmus.

<4/14> vollständig ausgearbeitete Einheit

<5/15> vollständig ausgearbeitete Einheit ist immer gut, Arbeitsblätter

<6/16> vollständig ausgearbeitete Einheit, gerade mit Übungen zu den mathematischen Grundlagen

<7/17> Interessante AB, wo entdeckendes Lernen im Mittelpunkt steht.

<8/18> Für ein allseits so gut dokumentiertes Verfahren wie das RSA-Verfahren benötigt man nicht unbedingt weitere Unterrichtsmaterialien. Man kann viele Arbeitsblätter und Zusammenfassungen finden, die von Kolleginnen und Kollegen im Internet auf ihren Web-Seiten zum RSA-Verfahren angeboten werden. Diese können gute Anregungen für eigene Arbeitsmaterialien geben. Auch Fachliteratur wie die oben genannte ist hervorragend geeignet, um nicht nur das Verfahren mit Rechenbeispielen nachzulesen sondern sich auch zusätzliche Hintergrundinformationen zu beschaffen.

20) Wie ist der konkrete Ablauf/Inhalt Ihrer Unterrichtseinheit zum RSA-Verfahren? Welche Methoden nutzen Sie?

<1/11> Im Wesentlichen Unterrichtsgespräch, Lehrervortrag, eigene Arbeitsblätter und Übungsphasen für die SuS, da ich kaum geeignetes Selbstlernmaterial kenne

1. Erweiterer Euklidischer Algorithmus
2. Modulo-Rechnung
3. Idee des Public-Key-Verfahrens
4. Erzeugung eines Schlüsselpaars
5. Verschlüsselung, Entschlüsselung von Texten

<2/12> siehe Punkt 16

<3/13> Das RSA-Verfahren war nur ein kurzer Exkurs in der Reihe zu Kryptographie.
Es wurden leidglich einige Beispiel im Unterricht durchgeführt.

<5/15> Der Theorieteil der Einheit Kryptographie wird bei mir "traditionellerweise" in GA durchgeführt.

<7/17> in den ersten drei Stunden lernen die SuS Caeser-Verfahren und Vegenere-Verfahren kennen. Dann kommen sie zum Schluss, dass diese zwei Verfahren nicht ideal sind. Danach fragen sie sich, ob es eine moderne Methode gibt, die von "Verbrechern" geknackt werden kann. Einige erzählen anschließend, dass sie etwas über Verschlüsselung von e-Mails gehört usw.

<8/18> Ablauf: (1) Rechnen mit Kongruenzen (2) RSA-Verfahren (3) Schlüsselerzeugung
Methoden im weitesten Sinne und Materialien, die an unserem Gymnasium zum : Arbeitsblätter, unterschiedliche Sozialformen, programmieren ausgewählter Teile, Verwendung fertiger Kryptographie-Programme, Tabellenkalkulation

Anhang II: Der RSA-Algorithmus in der Übersicht (mit Beispiel)

Der RSA-Algorithmus im Überblick	
Schritte des Algorithmus	Beispiel
<u>Schlüsselerzeugung:</u> • Zu wählen sind zwei Primzahlen *p* und *q* • Das Modul *n* ist $n = p \cdot q$ • $\varphi(n) = (p-1)\cdot(q-1)$ • Zu bestimmen sind zwei Zahlen *e* und *d*, so dass gilt: $e \cdot d \mod \varphi(n) = 1$	<u>Schlüsselerzeugung:</u> • $p = 5; q = 11$ • $n = 5 \cdot 11 = 55$ • $\varphi(n) = 40$ • $e = 7; d = 23$ • $7 \cdot 23 \mod 40 = 1$
Öffentlicher Schlüssel: *(e,n)* Privater Schlüssel: *(d,n)*	Öffentlicher Schlüssel: (7,55) Privater Schlüssel:(23,55)
<u>Verschlüsselung:</u> • Nachricht *m* • Öffentlicher Schlüssel *(e,n)* • Geheimtext *c* • $c = m^e \mod n$	<u>Verschlüsselung:</u> • Nachricht m = 2 • Öffentlicher Schlüssel (7,55) • Geheimtext c • $c = 18 = 2^7 \mod 55$
<u>Entschlüsselung:</u> • Geheimtext c • Privater Schlüssel (d,n) • Nachricht m' • $m' = c^e \mod n$	<u>Entschlüsselung:</u> • Geheimtext c = 18 • Privater Schlüssel (23,55) • Nachricht m' • $m' = 2 = 18^{23} \mod 55$

10 Literaturverzeichnis

[1] Ertel, Wolfgang: *Angewandte Kryptographie*. München : Hanser, 2012 — ISBN 3446427562 9783446427563

[2] Witten, Helmut ; Esslinger, Bernhard ; Gramm, Andreas ; Hornung, Malte: Asymmetrische Kryptographie für die Sek I - RSA (fast) ohne Mathematik? (2012)

[3] Esslinger, Bernhard: Asymmetrische Kryptologie am Beispiel RSA entdecken (2010)

[4] Borys, Thomas: *Codierung und Kryptologie Facetten Einer Anwendungsorientierten Mathematik im Bildungsprozess.* : Vieweg + Teubner Verlag, 2011 — ISBN 9783834817068 3834817066

[5] Esslinger, Bernhard: Das Cryptool Script: Kryptographie, Mathematik und mehr (2013)

[6] Humbert, Ludger: *Didaktik der Informatik: mit praxiserprobtem Unterrichtsmaterial*. 2., überarb. u. erw. Aufl. 2006. Aufl. : Vieweg+Teubner Verlag, 2006 — ISBN 3835101129

[7] Hromkovic, Juraj ; Freiermuth, Karin ; Keller, Lucia ; Steffen, Bjoern: *Einführung in die Kryptologie. Lehrbuch für Unterricht und Selbststudium*. Wiesbaden : Vieweg + Teubner in GWV Fachverlage GmbH, 2009 — ISBN 9783834810052 3834810053

[8] Kultusministerkonferenz: Einheitliche Prüfungsanforderungen Informatik (2004)

[9] Witten, Helmut ; Schulz, Ralph-Hardo: Faktorisieren mit dem Quadratischen Sieb. Ein Beitrag zur Didaktik der Algebra und Kryptologie. In: *LOG IN* (2012), Nr. Heft Nr. 172/173, S. 70–78

[10] Hartmann, Werner ; Näf, Michael ; Reichert, Raimond: *Informatikunterricht planen und durchführen*. 1. Aufl. 2006. Korr. Nachdruck 2007. Aufl. : Springer, 2006 — ISBN 3540344845

[11] Witten, Helmut ; Esslinger, Bernd ; Gramm, Andreas ; Hornung, Malte: Kann man RSA vertrauen? Asymmetrische Kryptographie für die Sekundarstufe I. In: *LOG IN* (2012), Nr. Heft Nr. 172/173, S. 79–92

[12] Beutelspacher, Albrecht ; Neumann, Heike B ; Schwarzpaul, Thomas: *Kryptografie in Theorie und Praxis : mathematische Grundlagen für Internetsicherheit und Mobilfunk und elektronisches Geld*. Wiesbaden : Vieweg + Teubner, 2009 — ISBN 9783834809773 3834809772

[13] Puhlmann, Dr. Hermann: Kryptographie verstehen - Ein schülergerechter Zugang zum RSA-Verfahren (1998)

[14] Beutelspacher, Albrecht: *Kryptologie : Eine Einführung in die Wissenschaft vom Verschlüsseln, Verbergen und Verheimlichen ; ohne alle Geheimniskrämerei, aber nicht ohne hinterlistigen Schalk, dargestellt zum Nutzen und Ergötzen des allgemeinen Publikums*. Wiesbaden : Vieweg + Teubner, 2009 — ISBN 9783834807038 3834807036 9783834802538 3834802530 9783834896063 3834896063

[15] Karpfinger, Christian ; Kiechle, Hubert: *Kryptologie. Algebraische Methoden und Algorithmen*. Wiesbaden : Vieweg + Teubner, 2010 — ISBN 9783834808844 3834808849

[16] Kirchgraber, U. ; Kramer, J.: Lauschen zwecklos! oder Wie aus einer seltsamen Erkenntnis über Zahlen die beste Geheimsprache aller Zeiten wurde (2005)

[17] Klüver, Christina ; Klüver, Jürgen: *Lehren, Lernen und Fachdidaktik Theorie, Praxis und Forschungsergebnisse Am Beispiel der Informatik.* : Vieweg + Teubner Verlag, 2011 — ISBN 9783834815477 3834815470

[18] Beutelspacher, Albrecht ; Schwenk, Jörg ; Wolfenstetter, Klaus-Dieter: *Moderne Verfahren der Kryptographie : Von RSA zu Zero-Knowledge*. Wiesbaden : Vieweg + Teubner, 2010 — ISBN 9783834812285 3834812285

[19] *Philologen-Jahrbuch Schuljahr 2011/12. Kunzes Kalender im 111. Jahrgang/Landesausgabe NRW für Gymnasien und Gesamtschulen*. Münster : Aschendorff-Verlag — ISBN 978-3-402-77858-6

[20] Ministerium für Schule und Weiterbildung, Wissenschaft und Forschung des Landes Nordrhein-Westfalen (Hrsg.): *Richtlinien und Lehrpläne für die Sekundarstufe II, Gymnasium, Gesamtschule in Nordrhein-Westfalen, Informatik*. Frechen : Ritterbach, 1999 — ISBN 3893146121 9783893146123

[21] Schüller, Anton ; Trottenberg, Ulrich ; Wienands, Roman ; Koziol, Michael ; Schneider, Rebekka: RSA - Primzahlen zur Verschlusselung von Nachrichten (2013)

[22] Witten, Helmut ; Schulz, Ralph-Hardo: RSA & Co. in der Schule. Moderne Kryptologie, alte Mathematik, raffinierte Protokolle. Neue Folge Teil 1: RSA für Einsteiger. In: *LOG IN* (2006), Nr. Heft Nr. 140, S. 45–52

[23] Witten, Helmut ; Schulz, Ralph-Hardo: RSA & Co. in der Schule. Moderne Kryptologie, alte Mathematik, raffinierte Protokolle. Neue Folge Teil 2: RSA für große Zahlen. In: *LOG IN* (2006), Nr. Heft Nr. 143, S. 50–58

[24] Witten, Helmut ; Schulz, Ralph-Hardo: RSA & Co. in der Schule. Moderne Kryptologie, alte Mathematik, raffinierte Protokolle. Neue Folge Teil 3: RSA und die elementare Zahlentheorie. In: *LOG IN* (2008), Nr. Heft Nr. 152, S. 60–70

[25] Witten, Helmut ; Schulz, Ralph-Hardo: RSA & Co. in der Schule. Moderne Kryptologie, alte Mathematik, raffinierte Protokolle. Neue Folge Teil 4: Gibt es genügend Primzahlen für RSA? In: *LOG IN* (2010), Nr. Heft Nr. 163/164, S. 97–103

[26] Witten, Helmut ; Schulz, Ralph-Hardo: RSA & Co. in der Schule. Moderne Kryptologie, alte Mathematik, raffinierte Protokolle. Neue Folge Teil 5: Der Miller-Rabin-Primzahltest oder: Falltüren für RSA mit Primzahlen aus Monte Carlo. In: *LOG IN* (2010), Nr. Heft Nr. 166/167, S. 92–106

[27] Witten, Helmut ; Schulz, Ralph-Hardo: RSA & Co. in der Schule. Moderne Kryptologie, alte Mathematik, raffinierte Protokolle. Neue Folge Teil 6: Das Faktorisierungsproblem oder: Wie sicher ist RSA? In: *LOG IN* (2012), Nr. Heft Nr. 172/173, S. 59–69

[28] Vorgaben zu den unterrichtlichen Voraussetzungen für die schriftlichen Prüfungen im Abitur in der gymnasialen Oberstufe im Jahr 2013. Vorgaben für das Fach Informatik. In: Ministerium für Schule und Weiterbildung des Landes Nordrhein-Westfalen (Hrsg.) (2010)

[29] Vorgaben zu den unterrichtlichen Voraussetzungen für die schriftlichen Prüfungen im Abitur in der gymnasialen Oberstufe im Jahr 2014. Vorgaben für das Fach Informatik. In: Ministerium für Schule und Weiterbildung des Landes Nordrhein-Westfalen (Hrsg.) (2011)

[30] Vorgaben zu den unterrichtlichen Voraussetzungen für die schriftlichen Prüfungen im Abitur in der gymnasialen Oberstufe im Jahr 2015. Vorgaben für

das Fach Informatik. In: Ministerium für Schule und Weiterbildung des Landes Nordrhein-Westfalen (Hrsg.) (2012)

[31] Ralph-Hardo Schulz ; Witten, Helmut: Zeit-Experimente zur Faktorisierung. In: *LOG IN* (2010), Nr. 166/167

11 Erklärung

Ich versichere, dass ich die schriftliche Hausarbeit selbstständig verfasst und keine anderen als die angegebenen Quellen und Hilfsmittel benutzt habe. Alle Stellen der Arbeit, die anderen Werken dem Wortlaut oder Sinn nach entnommen wurden, habe ich in jedem Fall unter Angabe der Quelle als Entlehnung kenntlich gemacht. Das gleiche gilt auch für die beigegeben Zeichnungen, Kartenskizzen und Darstellungen.

Rheine, den 25. Mai 2013

_____

Thimo Engelsmeyer