Kann man RSA vertrauen?

Asymmetrische Kryptografie für die Sekundarstufe I

von Helmut Witten, Bernhard Esslinger, Andreas Gramm und Malte Hornung

Das Standard-Verfahren zur asymmetrischen Kryptografie wurde 1977 erfunden und mit den Initialen seiner Entwickler Rivest, Shamir und Adleman bezeichnet. Es ist seit über 30 Jahren führend – ohne RSA wäre der Online-Handel nicht denkbar. Dieses Verfahren steht aber in dem Ruf, dass eine Behandlung im Unterricht zu schwierig für die Sekundarstufe I sei. Man steht so vor einem Dilemma, denn die Frage, ob man im Internet seine Botschaften sicher verschlüsseln kann, geht nicht nur die künftigen Abiturienten an.

Wir wollen in diesem Beitrag deshalb Möglichkeiten aufzeigen, die Frage der Sicherheit der Kommunikation mithilfe des Internets im Allgemeinen und der von RSA im Besonderen bereits in der Mittelstufe zu thematisieren. Eine ausführlichere und mathematisch-informatisch fundiertere Behandlung des RSA-Verfahrens wird nach wie vor erst in der Oberstufe möglich sein, aber wesentliche Einblicke zu den Fragen

- ▷ Gibt es „unknackbare“ Verschlüsselungen?
- ▷ Was ist und wozu dient die asymmetrische Kryptografie?
- ▷ Worauf beruht die Sicherheit des nach wie vor am häufigsten verwendeten RSA-Verfahrens?

können bereits in der Mittelstufe gestellt und die Antworten darauf zumindest plausibel gemacht werden.

RSA (fast) ohne Mathematik?

Wenn man das Thema *RSA* in diesem Sinne unterrichten will, steht man vor den Fragen: Wie kann ich das Verfahren so aufbereiten, dass auch die Schülerinnen und Schüler es verstehen, die die mathematischen Hintergründe nicht vollständig durchdringen? Ab wann gilt das Verfahren als „verstanden“?

Eine Möglichkeit, diese Fragen anzugehen, besteht in der Entwicklung eines pragmatischen Kompetenzstufen-Modells (vgl. Meyer, ^{5}2007). Auf der untersten Kompetenzstufe steht beim vorliegenden Thema *RSA* die Fähigkeit zur pragmatischen Nutzung von Verschlüsselungssystemen. Am Anfang stehen die Entdeckung des Systems und das Üben des Verschlüsselns anhand konkreter E-Mails. Im letzten Schritt der Einheit – dem computergestützten Umgang mit großen Primzahlen – werden die Grenzen der Sicherheit des Systems durch die Schülerinnen und Schüler erforscht.

Ein pragmatisches Kompetenzstufenmodell zum Thema *RSA* schlagen wir wie in der Tabelle 1 vor.

Anhand der Tabelle 1 lässt sich erkennen, dass wir ein erstes Nachvollziehen des RSA-Systems auf die Handhabung des Verschlüsselungsalgorithmus und auf

Stufe 1	Die Schülerinnen und Schüler beschreiben Angriffsszenarien bei der Kommunikation über öffentliche Netzwerke und formulieren entsprechende Anforderungen (Authentizität, Integrität, Vertraulichkeit) an sichere Kommunikation. Sie setzen Verschlüsselung und digitale Signaturen in der E-Mail-Kommunikation auf der Anwendungsebene ein, erzeugen dazu Schlüsselpaare und tauschen öffentliche Schlüssel aus. Sie erklären, wie ein Computerprogramm prinzipiell mittels digitaler Signaturen die Integrität von Nachrichten und die Authentizität des Absenders verifizieren kann.
Stufe 2	Die Schülerinnen und Schüler beurteilen die Sicherheit des RSA-Verfahrens für verschiedene Schlüssellängen auf Grundlage von Experimenten zur Rekonstruktion privater Schlüssel mit dem Computer und einer Recherche zur RSA-Challenge.
Stufe 3	Die Schülerinnen und Schüler erzeugen per Hand ein eigenes, auf kleinen Primzahlen basierendes Schlüsselsystem und wenden die Algorithmen zum Ver- und Entschlüsseln per Hand auf kurze Nachrichten an. Sie rekonstruieren auf kleinen Primzahlen basierende private Schlüssel per Hand, erproben Verfahren zum Auffinden ausreichend großer Primzahlen und begründen den Zusammenhang zwischen Schlüssellänge und Sicherheit mit der Komplexität des mathematischen Problems der Faktorisierung von Semiprimzahlen. Sie beurteilen die Sicherheit des RSA-Verfahrens für verschiedene Schlüssellängen auf Grundlage einer mathematischen Argumentation.

Tabelle 1: Pragmatisches Kompetenzstufenmodell zum Unterrichtsthema *RSA*.

RSA – von der Greybox zur Whitebox

Für ein vollständiges mathematisches Durchdringen, damit das RSA-Verfahren zur „Whitebox“ wird, fehlen gegenüber der von uns vorgestellten Unterrichtsreihe die folgenden Schritte (vgl. Witten/Schulz, 2006ff.):

- ▷ Für das modulare Potenzieren werden einfache Regeln des modularen Rechnens sowie der „Square-and-Multiply“-Algorithmus zum schnellen Potenzieren verwendet (vgl. Witten/Schulz, 2006a).
- ▷ Zur Berechnung der Schlüssel benötigt man bei großen Zahlen den erweiterten Euklidischen Algorithmus. Außerdem kann man sich fragen, warum modulare Addition oder Multiplikation kein sicheres Verschlüsselungssystem liefert, warum man also Potenzieren muss (vgl. Witten/Schulz, 2006b).
- ▷ Für den Beweis der Korrektheit des RSA-Verfahrens verwendet man den kleinen Satz von Fermat bzw. den Satz von Euler-Fermat oder auch den Satz von Carmichael (vgl. Witten/Schulz, 2008).
- ▷ Wie man seit dem Altertum weiß, gibt es zwar unendlich viele Primzahlen, die sind aber mit zunehmender Größe immer dünner gesät. Gibt es bei der milliardenfachen Kommunikation überhaupt genügend Primzahlen für RSA? (vgl. Witten/Schulz, 2010a). Die Antwort auf diese Frage liefert der Gauß’sche Primzahlsatz, der eng mit der berühmten Riemann’schen Vermutung verknüpft ist, vielleicht das aktuell wichtigste ungelöste mathematische Problem.
- ▷ Bei der Größe der heute benötigten Schlüssel reichen klassische Methoden zum Auffinden von Primzahlen nicht mehr aus. Man verwendet hierfür den Miller-Rabin-Primzahltest. Doch wie funktioniert der? (vgl. Witten/Schulz, 2010b).
- ▷ Wie ist es um die Sicherheit von RSA bestellt? Antworten dazu erhält man im in den Beiträgen von Schulz/Witten (2010) sowie Witten/Schulz (in diesem Heft, S. 59ff.). Weitere mögliche Angriffe gegen RSA und die entsprechenden Vorkehrungen dagegen findet man in jedem Standardwerk zur Kryptologie (z. B. bei Klaus Schmeh, [4]2009).

Wie wir eingangs erwähnt haben, können die Jugendlichen zwar mit E-Mail-Programmen umgehen und tun dies auch häufig, ein Problembewusstsein für die damit verbundenen Fragen der Computersicherheit fehlt allerdings fast immer. Aber auch die Erwachsenen tun sich mit der sicheren Kommunikation per E-Mail schwer. Aus diesem Grund haben die Deutsche Post mit dem *E-Postbrief* und die Bundesregierung mit *De-Mail* kostenpflichtige Dienste ins Leben gerufen, die ein hohes Maß an Sicherheit versprechen.

E-POSTBRIEF
Ein Service der Deutschen Post

De-Mail

Um solche Angebote kritisch bewerten zu können, bedarf es einer Grundbildung in Fragen der Computersicherheit, die über das Verständnis und ggf. die Programmierung des Caesar-Verfahrens hinausgeht. Mit unserer Unterrichtsreihe wollen wir einen unseres Erachtens wichtigen Baustein zum Verständnis der asymmetrischen Kryptografie liefern, der schon in der Sekundarstufe I erarbeitet werden kann und somit potenziell allen Schülerinnen und Schülern zur Verfügung steht. Eine ausführlichere und mathematisch-informatisch fundiertere Behandlung des RSA-Verfahrens wird in der Regel erst in der Oberstufe möglich sein.

das Verständnis der Anforderungen Authentizität, Vertraulichkeit und Integrität reduzieren. Die Benutzung von RSA ohne Mathematik setzt voraus, dass die benötigte Mathematik in einer „Blackbox“ versteckt wird (Stufe 1). Entscheidend ist die Einsicht, dass sichere Kommunikation erst durch die Nutzung von Krptosystemen hergestellt werden kann. Die eigentliche Struktur des Kryptosystems bleibt dabei verborgen.

Im weiteren Verlauf wird in den in der Unterrichtsreihe vorgestellten Lernschritten mit der Lernsoftware *CrypTool* sowie einer eigens entwickelten Simulation das RSA-Verfahren zur „Greybox“. Als wichtigste Grundlage der Sicherheit der verwendeten Verschlüsselung wird die Länge der Schlüssel erkannt. Die Beurteilung der Sicherheit des Verfahrens kann dabei allein auf Experimenten zur Faktorisierung des Moduls (Stufe 2) oder – wenn auch der Algorithmus zur Erzeugung eines Schlüsselsystems erarbeitet wurde – auf den mathematischen Grundlagen der Zerlegung von Semiprimzahlen in ihre Primfaktoren beruhen (Stufe 3). Die Aufgaben und Forschungsfragen, vor die die Schülerinnen und Schüler im Verlauf der Unterrichtseinheit gestellt werden, ermöglichen ihnen, die jeweils nächsthöhere Kompetenzstufe weitgehend selbstständig zu erreichen. Insgesamt soll der vorgestellte Unterrichtsabschnitt die Lernenden nicht nur dazu befähigen, ihre Kommunikation über öffentliche Netzwerke sicher zu gestalten, sondern auch bestehende Kommunikationssysteme auf Basis der im Unterricht entwickelten Anforderungen (Authentizität, Integrität, Vertraulichkeit) zu beurteilen.

Mehr Mathematik als Primzahlen, Sieb des Eratosthenes sowie einfaches modulares Rechnen einschließlich Potenzieren kommt in dieser Unterrichtsreihe nicht zum Einsatz. (Falls auf das manuelle Ver- und Entschlüsseln verzichtet wird, entfallen sogar diese letztgenannten Voraussetzungen.) Welche Schritte zum vollständigen Durchdringen des RSA-Algorithmus notwendig sind, findet man oben im Kasten „RSA – von der Greybox zur Whitebox“. Diese vertiefenden Einsichten können, wie bereits erwähnt, i. Allg. erst innerhalb der Oberstufe erarbeitet werden.

Kryptografie in der Sekundarstufe I

In unserer Unterrichtsreihe *E-Mail (nur?) für Dich* (vgl. Gramm u.a., 2011) wird das Thema Verschlüsselung im Kontext der „elektronischen Post“ behandelt. Die Entscheidung für diesen Kontext erfolgte aus mehreren Gründen. So zeigt die aktuelle JIM-Studie (vgl. mpfs, 2011, S.34), dass auch im Zeitalter von *Twitter* und *Facebook* dieser Kommunikationsweg sehr häufig genutzt wird. Wie man eine E-Mail schreibt, weiß mittlerweile also (fast) jeder Jugendliche. Welche informationstechnischen Systeme sich dahinter verbergen und welche Unsicherheiten elektronische Kommunikation mit sich bringt, bleibt dabei den meisten verborgen. Die Unterrichtseinheit *E-Mail (nur?) für Dich* hat sich deshalb zum Ziel gesetzt, den Schülerinnen und Schülern einen bewussten und sicheren Umgang mit dem Medium E-Mail zu ermöglichen.

Mit dem Internetdienst E-Mail wird ein immer noch aktueller Kommunikationsweg in den Vordergrund gestellt, der für die Schülerinnen und Schüler viele motivierende und handlungsorientierte Lernaufgaben im Zusammenhang mit Verschlüsseln und eventuellem unerwünschten Entschlüsseln („Knacken“) ermöglicht. Damit kann auch die Einstellung von „Ich habe nichts zu verbergen“ zu „Das ist doch ganz schön fies“ gewandelt werden. Der E-Mail-Verkehr wird in dieser Reihe in einem geschützten pädagogischen Bereich im Klassenzimmer durchgeführt (zu den technischen Einzelheiten siehe Koubek, 2007 ff., Schaltfläche: Email (nur?) für Dich).

Es ist aber auch möglich (und im bisherigen Informatikunterricht eher der übliche Weg), eine eigenständige Reihe zur Kryptologie durchzuführen, die dann durch Programmierübungen zu den bekannten elementaren Verschlüsselungsverfahren (Caesar, Vigenère, …) begleitet wird. Die Arbeitsbögen zur Kryptografie, die auf den Seiten zu der E-Mail-Reihe angeboten werden, lassen sich auch unabhängig von dem Kontext E-Mail verwenden. Sie finden sich gesammelt in dem PDF-Dokument, das bei der Internetpräsenz von *Informatik im Kontext* heruntergeladen werden kann (siehe Koubek, 2007 ff., Schaltflächen: Email (nur?) für Dich → >> direkt zu den Materialien → Arbeitsbögen und Materialien für sämtliche Stunden in einem PDF-Dokument, ab Seite 18, bzw. als URL bei Gramm u.a., 2012, angegeben).

Gibt es ein unknackbares Verschlüsselungsverfahren?

Die Antwort auf diese Frage ist überraschenderweise „Ja“. Das Verfahren hat den Namen *One-Time-Pad* und ist bereits seit etwa hundert Jahren bekannt (vgl. Wikipedia – Stichwort „One-Time-Pad“). Eine korrekt nach diesem Verfahren verschlüsselte Botschaft könnte auch nicht mit der vereinten Rechenkraft aller zurzeit auf der Erde existierender Computer gebrochen werden.

Um die Existenz dieser beweisbar sicheren Verschlüsselung plausibel zu machen, kann man im Unterricht der Sekundarstufe I in folgenden Schritten vorgehen:

1. Man beginnt mit dem *Caesar-Verfahren*, das sehr leicht geknackt werden kann. Man muss nur den häufigsten Buchstaben bestimmen, der bei längeren deutschen Texten mit hoher Wahrscheinlichkeit das „e“ ist, und schon hat man die Zahl der Verschiebungsschritte und damit den verwendeten Schlüsselbuchstaben gefunden. Alternativ kann man alle 25 möglichen Schlüssel mit dem Beginn der Chiffre testen und so ermitteln, welcher Schlüssel zu einem sinnvollen Klartext führt.
2. Darauf aufbauend wird das *Vigenère-Verfahren* eingeführt, das von seiner Entdeckung (ca. 1600) bis zum Ende des 19. Jahrhunderts als unknackbar galt. Hierbei handelt es sich gewissermaßen um ein Multi-Caesar-Verfahren: Jeder Buchstabe wird mit einem Buchstaben aus dem Schlüsselwort verschlüsselt, das zyklisch angewandt wird.
3. Es ist interessant, sich die verschiedenen Möglichkeiten, den Vigenère-Code dennoch zu knacken, vor Augen zu führen. Die Parallelstellen-Suche nach *Kasiski* (vgl. Wikipedia – Stichwort „Kasiski-Test“) kann auch von den Lernenden der Sekundarstufe I nachvollzogen werden; entsprechende, von unserer Berliner Kollegin Irmgard Letzner entwickelte Arbeitsbögen finden sich bei den Materialien unserer E-Mail-Reihe (siehe Gramm u.a., 2012, ab Seite 24).
   Der *Friedman-Test* ist zuverlässiger und für die Programmierung besser geeignet, sollte aber ebenso wie der mathematische Beweis der Sicherheit des One-Time-Pad-Verfahrens eher in der Sekundarstufe II behandelt werden (vgl. Witten u.a., 1998 und 1999). Der Friedman-Test ist auch im vielfach ausgezeichneten Programm *CrypTool 1* implementiert, sodass die *Anwendung* dieses Verfahrens für Schülerinnen und Schüler aus der Sekundarstufe I einfach möglich ist (Aufruf im CrypTool-Menübaum: Analyse → symmetrische Verschlüsselung (klassisch) → Ciphertext only → Vigenère).
4. Beiden Verfahren zur Kryptoanalyse von Vigenère ist gemeinsam, dass es ausreicht, die Schlüsselwort*länge* zu bestimmen. Mit dieser Information kann man den Geheimtext in Teiltexte zerlegen, die jeweils nur Caesar-verschlüsselt sind und wie unter Punkt 1 entschlüsselt werden können.
5. Der Schritt von Vigenère zum One-Time-Pad ist nach diesen Vorbereitungen einfach und logisch: Das Schlüsselwort muss genau so lang wie der Klartext sein. Darüber hinaus sollte das Schlüsselwort nicht einer natürlichen Sprache entstammen, sondern eine Zufallsfolge sein, die jeweils nur einmal verwendet wird (daher der Name *One-Time-Pad*: Agenten erhielten einen Block mit solchen Zufallsfolgen, siehe Bild 1, nächste Seite).
   Der jeweils verwendete Schlüssel sollte nach seiner Verwendung vernichtet werden. Das ähnelt der Verwendung von TAN-Briefen beim Online-Banking: TANs (***T**rans**a**ktions**n**ummern*) sind jeweils dem Kunden und dem Banksystem bekannt und werden nur einmal verwendet. Berühmte Beispiele der Anwendung des One-Time-Pad-Verfahrens werden von Witten u.a. (1999, S.52 ff.) geschildert. Erwähnenswert sind auch Dokumente aus dem VENONA-Projekt (vgl. NSA, 2009; Crowell, 2009; Wikipedia –

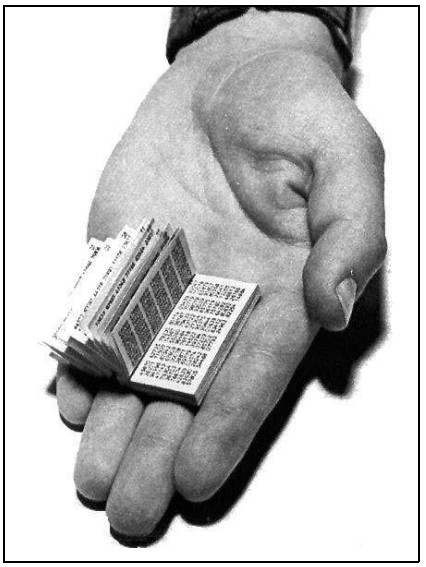


Bild 1: Ein sowjetrussisches One-Time-Pad.

http://www-ivs.cs.uni-magdeburg.de/bs/lehre/wise0102/progb/vortraege/choppe/choppe1.html

Stichwort „VENONA-Projekt“), in denen beschrieben wird, dass die sowjetischen Atomspione in den USA u.a. durch Fehler in der Anwendung der One-Time-Pad-Verschlüsselung überführt werden konnten.

6. Die wesentliche Eigenschaft des One-Time-Pad-Verfahrens, nicht knackbar zu sein, kann schon in der Sekundarstufe I durch ein einfaches Experiment eindrücklich veranschaulicht werden. Man wählt einen beliebigen Geheimtext, z.B. ICQFDBQDEYYNIGTR, und kann dann zu einem belieben Klartext gleicher Länge mithilfe des Vigenère-Quadrats den zughörigen Schlüssel konstruieren. Dies gelingt hier z.B. für folgende Klartexte: SCHULEMACHTSPASS, MATHEMATIKISTGUT und FERIENSINDBESSER. Damit wird deutlich, was mit „unknackbar“ gemeint ist: Man kann bei einem mit diesem Verfahren verschlüsselten Text aus dem Geheimtext keinerlei Rückschlüsse auf den Klartext ziehen. (Wer die Lösungen vergleichen möchte: Sie stehen in dem zitierten Artikel von Witten u.a., 1999, auf Seite 50.)

Leider ist das Verfahren nur sehr aufwendig umzusetzen, schließlich müsste jeder Kommunikationsteilnehmer für jeden möglichen Kommunikationspartner mit dem jeweils passenden One-Time-Pad ausgestattet werden – angesichts von abermillionen Internetnutzern ein Ding der Unmöglichkeit.

Asymmetrische Kryptografie

Anhand des One-Time-Pad-Verfahrens kann den Lernenden immerhin die Existenz sicherer symmetrischer Verschlüsselungsverfahren relativ einfach plausibel gemacht werden, auch wenn die in der Kommunikation mithilfe des Internets tatsächlich verwendeten symmetrischen Verfahren (DES, IDEA, AES) nicht besprochen werden können, weil sie für die Sekundarstufe I zu schwer sind.

Damit ist aber das Problem sicherer Kommunikation über das Internet noch keineswegs gelöst. Zentral dafür sind die Möglichkeiten, die die asymmetrische Kryptografie für den sicheren Schlüsselaustausch und die Authentifizierung der Nutzer bietet. In den meisten Fällen wird dazu – wie bereits erwähnt – das RSA-Verfahren verwendet. Zum Verständnis dieses Verfahrens wird etwas elementare Zahlentheorie benötigt, die unseren Lernenden in der Sekundarstufe I leider nicht zur Verfügung steht. Für die vorliegende Unterrichtseinheit haben wir daher nach Möglichkeiten gesucht, das Prinzip der RSA-Verschlüsselung mit einem Minimum von Mathematik zumindest plausibel zu machen. Die Mittel dafür sind einerseits eine Animation zur asymmetrischen Verschlüsselung, die von Andreas Gramm (2010) entwickelt wurde, andererseits die RSA-Demo aus dem vielfach ausgezeichneten *CrypTool*-Lernprogamm (siehe CrypTool-Portal). Weitere Anregungen erhielten wir aus der von Bernhard Esslinger u.a. (2010) erstellten Schritt-für-Schritt-Anleitung *Asymmetrische Kryptologie*, die allerdings in ihrer Gesamtheit zu umfangreich für unser Unterrichtsvorhaben war.

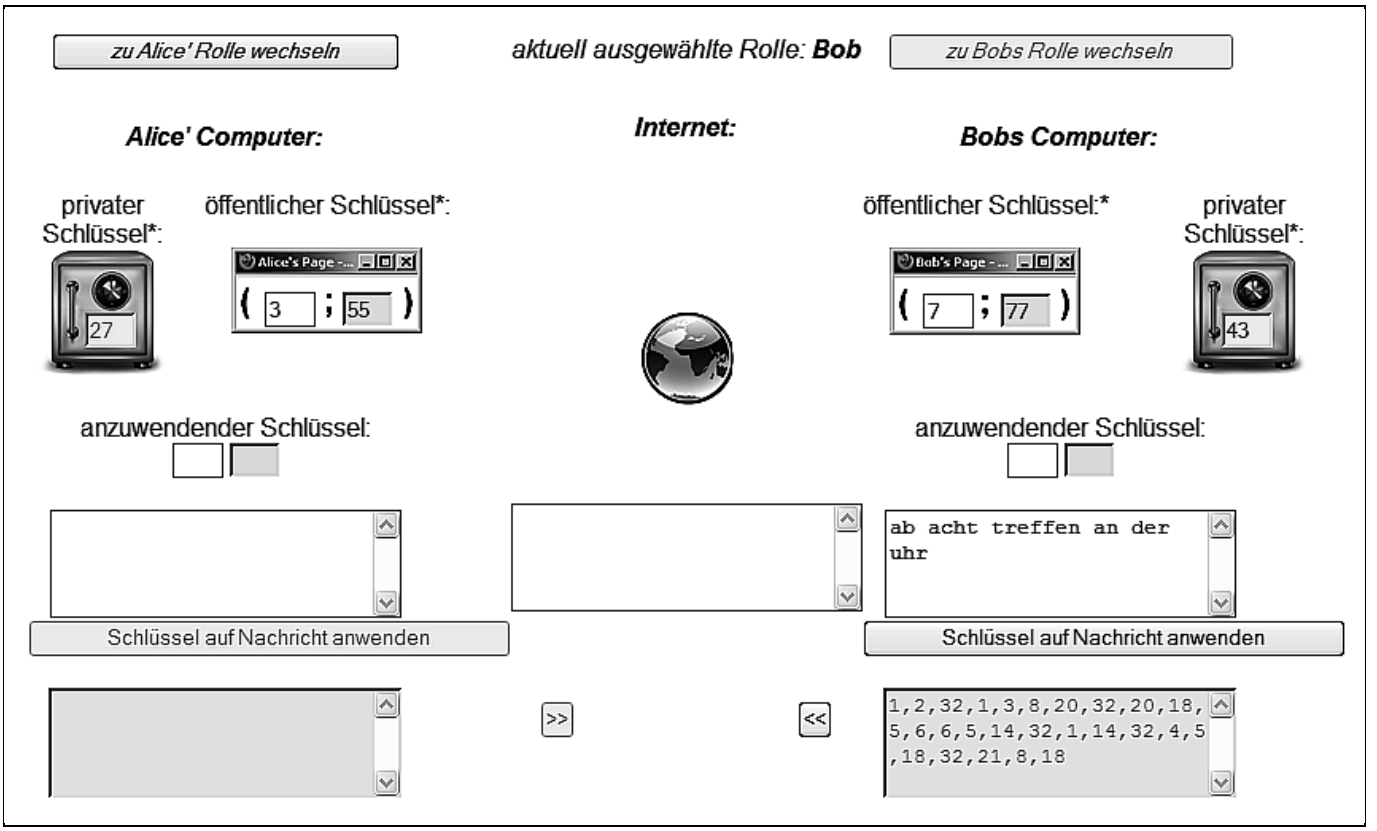
Um das Anspruchsniveau für die Sekundarstufe I zu verdeutlichen, sollen im Folgenden einige der in der E-Mail-Reihe verwendeten Unterrichtsmaterialien aus diesem Unterrichtsabschnitt aufgeführt werden (vgl. auch Gramm u.a., 2011 und 2012).

Einstieg

Der Einstieg in die asymmetrischen Verschlüsselungsverfahren, die den Austausch von Schlüsseln auf einem geheimen Kanal überflüssig machen, findet anhand des *Diffie-Hellman-Verfahrens* statt. Für die Erarbeitung des Verfahrens im Unterricht genügen eine Box und zwei Vorhängeschlösser: Die Schülerinnen und Schüler erhalten die Aufgabe, eine Nachricht an ihre Mitschüler zu senden, ohne dass die Box von einer dritten Partei auf dem Transportweg geöffnet werden kann.

Die Lernenden finden i. Allg. schnell die Lösung: Alice (Sender) packt ihr Geheimnis in die Kiste und verschließt sie mit ihrem Vorhängeschloss; den Schlüssel behält sie. Dann schickt sie die verschlossene Kiste zu Bob (Empfänger), der sie ein zweites Mal mit seinem Vorhängeschloss verschließt. Anschließend wandert die Kiste mit beiden Schlössern zurück zu Alice, die jetzt ihr Schloss mit ihrem Schlüssel öffnet und dann – nur noch mit Bobs Schloss gesichert – zu Bob zurückschickt, der sie jetzt ohne Weiteres öffnen und das Geheimnis entnehmen kann. Alternativ zu diesem Vorgehen kann Bob auch geöffnete Vorhängeschlösser *ohne* Schlüssel an seine Kommunikationspartner zum späteren Gebrauch verteilen. Da offene Vorhängeschlösser von jedermann verschlossen (zugedrückt), aber nur mit dem Schlüssel problemlos wieder geöffnet werden können, sind sie ein Beispiel für Einwegfunktionen mit Falltür (kurz: Falltürfunktionen; vgl. auch Müller, 2011).

Das Vorhängeschloss steht in diesem Beispiel für den öffentlichen Schlüssel, der Schlüssel dazu für den privaten Schlüssel. Der Vorteil dieser Demonstration liegt in der Anschaulichkeit, ein Nachteil besteht darin,



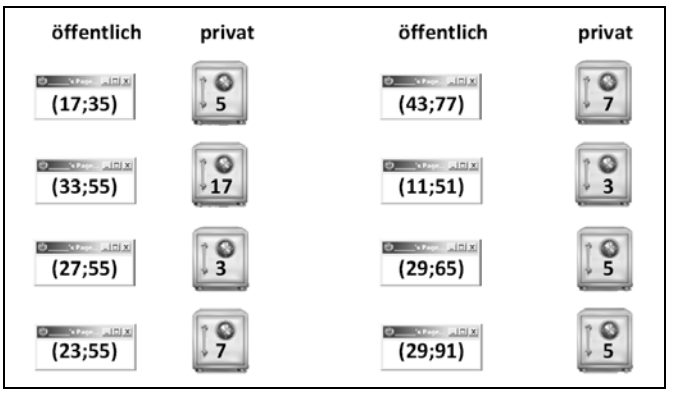


Bild 2: Animation zur asymmetrischen Kryptografie – Vertraulichkeit herstellen.

* Die beim Start der Animation eingetragenen Schlüsselpaare sind nur Vorschläge. Es gibt verschiedene Schlüsselpaare, wie die links nebenstehenden Beispiele zeigen. Wenn ein anderes Schlüsselpaar verwendet werden soll, sind einfach die Einträge in den Textfeldern zu ändern, die von den Symbolen für den öffentlichen und den privaten Schlüssel umgeben sind.

dass man damit das Signieren einer Botschaft nicht simulieren kann. Ohne die Nachricht zusätzlich zu signieren, besteht die Gefahr eines „Man-in-the-Middle"-Angriffs. Damit ist gemeint, dass sich die böse Eve (von engl.: *evil*, deutsch: böse, schlecht) einschaltet, die Nachrichten abfängt und durch ihre eigenen Botschaften ersetzt.

Signieren einer Botschaft

Die asymmetrische Verschlüsselung gestattet aber nicht nur, die Vertraulichkeit herzustellen, indem Nachrichten ohne Austausch geheimer Schlüssel chiffriert werden können. Mit dem *Signieren* einer Botschaft kann auch sichergestellt werden, dass diese authentisch ist, d.h. wirklich von Alice und nicht etwa von Eve stammt. Durch den Einsatz einer Hashfunktion, die gewissermaßen einen „digitalen Fingerabdruck" eines Dokuments erzeugt, kann zusätzlich überprüft werden, ob das Dokument eventuell verändert wurde oder nicht, d.h. die Korrektheit bzw. Unversehrtheit der Daten – ihre Integrität – wird damit geprüft.

Das prinzipielle Vorgehen zum Erlangen von Vertraulichkeit, Authentizität und Integrität bei der Kommunikation über öffentliche Netze kann relativ einfach mit den Arbeitsbögen „Vertraulichkeit mit RSA herstellen" sowie „Integrität und Authentizität mit digitaler Unterschrift sicherstellen" durch die Lernenden selbstständig erarbeitet werden (vgl. Gramm, 2010). Im Hintergrund arbeitet bei diesen Animationen der RSA-Algorithmus mit kleinen Zahlen als „Blackbox". Die Schlüsselpaare von Alice und Bob werden jeweils voreingestellt, sodass weder die Erzeugung eines Schlüsselpaares noch der Algorithmus zur Ver- und Entschlüsselung an dieser Stelle bereits problematisiert werden.

Vertraulichkeit mit RSA herstellen

Bei der Erarbeitung dieser Animation lernen die Schülerinnen und Schüler die Funktion von öffentlichem und privatem Schlüssel kennen (siehe Bild 2).

Der geheime Schlüssel wird dabei durch einen kleinen Tresor symbolisiert; der Inhalt ist immer nur für denjenigen sichtbar, der gerade die Rolle von Alice bzw. Bob einnimmt. Der öffentliche Schlüssel wird durch eine kleine Webseite dargestellt. Die Weltkugel soll verdeutlichen, dass die öffentlichen Schlüssel weltweit einsehbar sind.

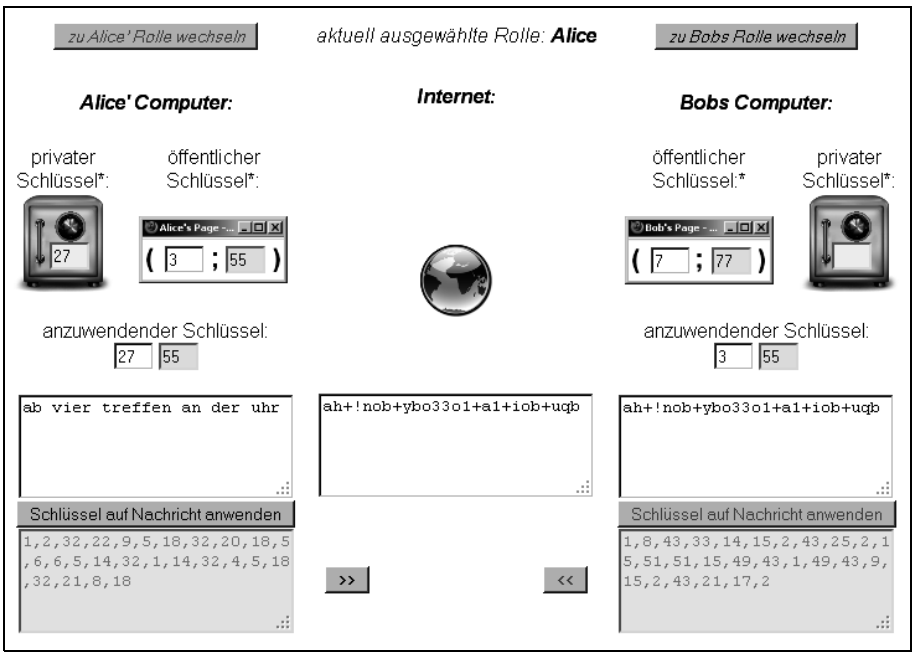
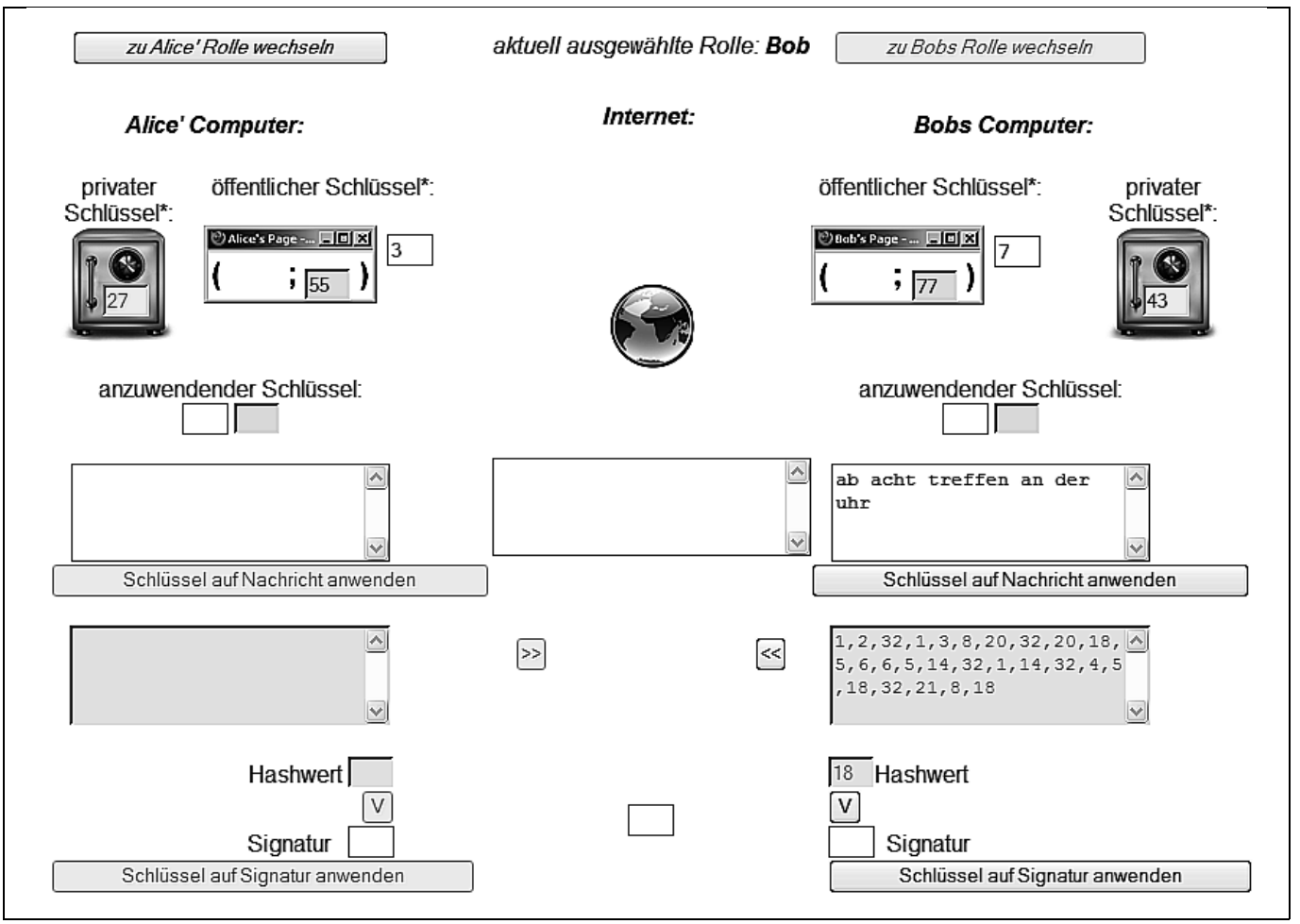


Bild 3: Ausschnitte aus dem Video „Vertraulichkeit durch asymmetrische Kryptologie herstellen“ (rechts oben: Beispiel des Vorhängeschlosses; rechts: Alice hat Bob eine verschlüsselte Nachricht geschickt, die er noch entschlüsseln muss).

http://www.youtube.com/watch?v=nAXp7xbsAHE

Bild 4 (unten): Erweiterte Animation zur asymmetrischen Kryptografie – Integrität und Authentizität sicherstellen.

http://ods3.schule.de/informatik/material/asym/Integritaet-und-Authentizitaet-mit-digitaler-Unterschrift-sicherstellen.html



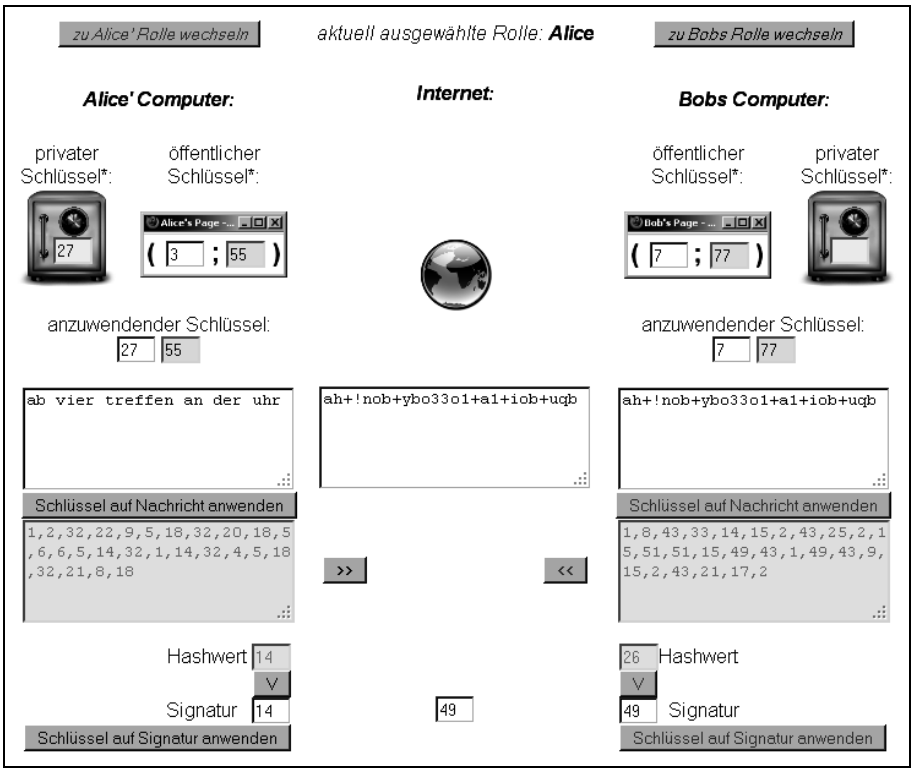


Bild 5:
Ausschnitt aus dem Video „Integrität und Authentizität mit digitaler Unterschrift sicherstellen“ (Bob hat Alices verschlüsselte Nachricht mit dem ebenfalls verschlüsselten Hashwert erhalten).

http://www.youtube.com/watch?v=j84kC7p4les

Außerdem geht es bei E-Mail um weltweite Kommunikation, auch das soll durch die Weltkugel angedeutet werden.

Um die Lernenden bei der selbstständigen Erarbeitung des Umgangs mit öffentlichem und privatem Schlüssel zu unterstützen, steht zudem unter http://it-lehren.de/asym ein Lehrvideo zur Verfügung (siehe Bild 3, vorige Seite, und LOG-IN-Service, Seite 123). Die Arbeitsaufträge finden sich einerseits unterhalb der Animation, können aber in einem gesonderten Fenster ein zweites Mal geöffnet werden, damit nicht ständig auf- und abgescrollt werden muss.

Integrität und Authentizität mit digitaler Unterschrift sicherstellen

Für den nächsten Lernschritt steht die Animation in einer erweiterten Fassung zur Verfügung (siehe Bild 4, vorige Seite). Hier sind zusätzlich zu den aus Bild 2 bekannten Bildschirmelementen jeweils ein Feld für Hashwert und Signatur zu erkennen.

Beim Eintippen einer Nachricht kann man beobachten, dass sich der Hashwert mit jedem eingetippten Buchstaben verändert (digitaler „Fingerabdruck“). Die Grundidee der digitalen Signatur besteht nun darin, dass jeder diesen Hashwert ebenfalls berechnen kann (natürlich die gleiche Hashfunktion vorausgesetzt). Bob verschlüsselt den Hashwert seiner Nachricht mit seinem privaten Schlüssel. Alice berechnet den Hashwert der empfangenen Nachricht und entschlüsselt Bobs Signatur mit seinem öffentlichen Schlüssel. Wenn diese Werte übereinstimmen, kann sie sicher sein, dass die Nachricht nicht verändert wurde (Integrität) und auch wirklich von Bob stammt (Authentizität), da die Signatur nur mit Bobs privatem Schlüssel erzeugt worden sein kann. Die digitale Signatur funktioniert unabhängig davon, ob Bob seine Botschaft verschlüsselt übermittelt oder nicht – es sind nämlich Fälle denkbar, dass die Integrität und Authentizität wichtiger sind als die Vertraulichkeit (z.B. bei verbindlichen Rechtsgeschäften, die über das Internet abgewickelt werden sollen).

Auch zu dieser Animation gibt es ein Lehrvideo (siehe Bild 5); die Erläuterung sowie die Arbeitsaufträge finden sich wieder unterhalb der Animation.

Die Sicherheit von RSA – Faktorisierung und das Sieb des Eratosthenes

Die Falltürfunktion, auf der die Sicherheit von RSA in erster Linie beruht, ist verbunden mit dem Faktorisierungsproblem (vgl. auch Wikipedia – Stichwort „Faktorisierungsverfahren“). Damit ist gemeint, dass es mit dem Computer einfach und schnell möglich ist, auch sehr große Zahlen zu multiplizieren, die Faktorisierung des Produkts aber bei hinreichend großen Zahlen sehr aufwendig ist. Produkte aus genau zwei Primzahlen heißen *Semiprimzahlen*. So ist es z.B. bis zum heutigen Tag nicht gelungen, die Semiprimzahl RSA-1024 zu faktorisieren – eine Zahl mit 309 Dezimalstellen (vgl. RSA Inc., 2012). Experten erwarten, dass dies jedoch bis zum Jahr 2020 gelingen wird.

Wenn die Faktorisierung gelungen ist, werden die Faktoren veröffentlicht, und jeder kann durch einfache Multiplikation überprüfen, ob die Faktoren die gegebene Zahl liefern. Da die Faktorisierung bis auf die Reihenfolge der Faktoren eindeutig ist, kann es auch keine zweite Lösung geben.

Im Arbeitsbogen „Primzahlen finden mit dem Sieb des Eratosthenes“ (siehe Bild 6, nächste Seite) werden die zugehörigen Definitionen geliefert und einige einfache Aufgaben dazu gestellt. Für die praktische Arbeit mit dem Sieb bietet sich die Animation von Hans-Bernhard Meyer an, die es den Lernenden gestattet, die Arbeitsweise des Siebs selbstständig zu erforschen (vgl. Meyer, 2012).

Die auf dem Arbeitsbogen bei der Aufgabe 4 angegebenen Prim- und Semiprimzahlen können mit der Meyer’schen Animation aber nur zum kleinen Teil ge-

Primzahlen finden mit dem Sieb des *Eratosthenes*

Für die asymmetrische Kryptographie benötigen wir mathematische Funktionen, deren Anwendung mit einer Information (dem öffentlichen Schlüssel) sich durch Anwendung mit einer anderen Information (dem privaten Schlüssel) rückgängig machen lässt. Hier spielen **Primzahlen** eine wichtige Rolle, die einige besondere Eigenschaften aufweisen:

Definition:
Primzahlen sind alle natürlichen Zahlen größer als 1, die nur durch 1 und sich selber teilbar sind.
Alle natürlichen Zahlen größer als 1, die keine Primzahlen sind, heißen **zusammengesetzte Zahlen**.
Die 1 ist weder eine Primzahl noch ist sie zusammengesetzt!

Aufgabe 1:
Nenne 10 Beispiele für Primzahlen!

Aufgabe 2:
Überlege Dir eine Begründung, warum man diese Zahlen zusammengesetzt nennt!
Nenne 10 Beispiele für zusammengesetzte Zahlen!

Wie findet man Primzahlen? Eine sehr effektive Methode ist das **Sieb des *Eratosthenes***.

Aufgabe 3:
Informiere Dich unter der Adresse http://www.hbmeyer.de/eratosib.htm über die Funktionsweise des Primzahlsiebs! Bearbeite die auf dieser Seite genannte Aufgabe!

Zur Auswertung dieser Experimente überlege Dir Antworten auf die folgenden Fragen:

- Warum erhält man bereits alle Primzahlen ≤ 400, wenn man nur mit den Primzahlen ≤ 20 „siebt"?
- Wieso kann man sicher sein, dass wirklich nur noch Primzahlen in der Tabelle stehen?
- Warum nannte *Eratosthenes* das Verfahren, das er vermutlich gar nicht selber erfunden hat, „Sieb"?

Wichtig für die moderne Kryptologie im Allgemeinen und das RSA-Verfahren im Besonderen sind die so genannten **Semiprimzahlen**. Das sind natürliche Zahlen n, die genau zwei unterschiedliche Primfaktoren p und q haben, so dass $n = p \cdot q$ gilt. Für die asymmetrische Kryptographie ist es wichtig, dass man aus dem öffentlichen Schlüssel e den privaten Schlüssel d nicht berechnen kann. Dies wird beim RSA-Verfahren dadurch abgesichert, dass es praktisch unmöglich ist, riesige Semiprimzahlen mit hunderten von Dezimalstellen in ihre beiden Primfaktoren zu zerlegen. Umgekehrt ist es sehr einfach, aus zwei großen Primzahlen durch Multiplikation eine Semiprimzahl zu erzeugen.

Aufgabe 4:
Welche der folgenden Zahlen sind Primzahlen, welche sind Semiprimzahlen?
Falls es Semiprimzahlen sind: Gib die Faktoren an!
23, 55, 113, 119, 841, 1829, 3109, 9847, 10807, 13121, 14603, 15551, 16061, 16199,
1522605027922533360535618378132637429718068114961380688657908494580122963258952897654000350692006139

Aufgabe 5:
Recherchiere:
1. Wer war eigentlich *Eratosthenes*?
2. Informiere dich über die *RSA Factoring Challenge*!

Freiwillige Zusatzaufgabe für Interessierte:
Eratosthenes ist auch als Begründer der wissenschaftlichen Geographie bekannt, er konnte bereits vor fast 2000 Jahren ziemlich genau den Erdumfang bestimmen. Informiere dich über *Eratosthenes'* Methode zur Bestimmung des Erdumfangs!

Bild 6: Arbeitsbogen zu Primzahlen und Semiprimzahlen.

löst werden. Eine sehr raffinierte Version des Siebs von Eratosthenes wird bei den Materialien zur oben erwähnten E-Mail-Reihe angeboten und wurde dem Buch von Armin Leutbecher (1996, Ausklapptafel) entnommen. Das Verständnis der Arbeitsweise mit dieser Tabelle ist nicht trivial und muss mit den Lernenden entsprechend geübt werden. Da dieses Sieb dank seiner geschickten Darstellung für alle Zahlen kleiner 16200 entweder anzeigt, dass sie *prim* sind oder den jeweils kleinsten Primfaktor angibt, können damit alle Aufgaben zu den Zahlen aus der ersten Reihe von Aufgabe 4 des Arbeitsbogens gelöst werden.

Für die Schülerinnen und Schüler ist es aber einfacher, mit *CrypTool 1* zu arbeiten und mit dem Aufruf Einzelverfahren → RSA-Kryptosystem → Faktorisieren einer Zahl das Fenster für die Faktorisierung zu aktivieren (siehe auch Arbeitsbogen „RSA knacken mit *CrypTool*“, S. 90). Es ist praktisch, wenn die Lernenden den Arbeitsbogen zu Eratosthenes online zur Verfügung haben, da sie dann die Zahlen nicht abtippen müssen, sondern einfach durch „Kopieren und Einfügen“ übertragen können. Sie erhalten auf diesem Weg unmittelbar die Ergebnisse, z.B. dass 16199 eine Semiprimzahl mit den Faktoren 97 und 167 ist und dass 16061 eine „richtige“ Primzahl ist. Außerdem üben sie auf diese Weise den Umgang mit *CrypTool 1*.

Mit der letzten in Aufgabe 4 angegebenen Zahl

1522605027922533360535618378132637429718068114961380688657908494580122963258952897654000350692006139

ist auch *CrypTool 1* überfordert (*CrypTool 2* dagegen schafft die Faktorisierung auf einem modernen Rechner in ca. zwei Stunden, vgl. Schulz/Witten, 2010), immerhin kann man mit dem Rabin-Miller-Primzahltest sehr schnell feststellen, dass dieser Zahlenwurm keine Primzahl, also vermutlich eine Semiprimzahl ist. Geschickter ist es, diese Zahl in ein Suchfenster z.B. von *Google* einzugeben. Man erhält dann als erstes Ergebnis der Suche, dass es sich um die Zahl RSA-100 handelt. Wenn man in der *englischen* Wikipedia (http://en.wikipedia.org/) „RSA-100“ eingibt, wird man fündig und erhält unter http://en.wikipedia.org/wiki/RSA-100#RSA-100 die Zahlen

3797522793694367392280887275544562785456553663638199

und

4009469095092088103068373529276146838921489972406 1

sowie die Information, dass diese Faktoren von RSA-100 am 1. April 1991 von Arjen K. Lenstra gefunden wurden.

Wie kann man diese Information überprüfen, d.h. wie kann man diese beiden 50-stelligen Faktoren miteinander multiplizieren? Ein normaler Taschenrechner scheitert an dieser Aufgabe, weil er bei der Darstellung der Zahlen in die (ungenaue) Exponentialform umschaltet. Auf diese Weise erhält man kein exaktes Ergebnis.

Man muss ein System verwenden, das Langzahlarithmetik beherrscht, z.B. der PYTHON-Interpreter (vgl. Python Programming Language) oder das Computer-Algebra-System SAGE (vgl. SAGE). Mit einem solchen Programm kann man leicht nachprüfen, dass die Zerlegung korrekt ist.

Mit RSA-100 ist man bereits bei der *RSA-Challenge* angekommen (Aufgabe 5.2 des Arbeitsbogens). Leider sind die Informationen dazu auf den deutschen Wikipedia-Seiten nicht ganz zuverlässig, besser ist man mit der englischen Version beraten, die allerdings für viele Schülerinnen und Schüler nicht so einfach zu lesen ist wie die deutsche (http://en.wikipedia.org/wiki/RSA_Factoring_Challenge). Alle noch ungelösten RSA-Challenges finden sich mit einer Beschreibung auch auf der Krypto-Wettbewerbsseite MysteryTwisterC3 (http://www.mysterytwisterc3.org/).

Bei der Frage nach Eratosthenes (Aufgabe 5.1 des Arbeitsbogens) soll herausgefunden werden, dass dieser geniale und vielseitige Forscher als Begründer der wissenschaftlichen Geografie gilt. Seine auf sorgfältigen Messungen beruhende Bestimmung des Erdumfangs gehört zu den bekanntesten wissenschaftlichen Leistungen des Altertums (vgl. z.B. Wikipedia – Stichwort „Eratosthenes“).

Die Sicherheit von RSA – RSA knacken mit der RSA-Demo von CrypTool 1

An dieser Stelle des Unterrichtsverlaufs muss man sich entscheiden, ob der RSA-Algorithmus behandelt werden soll oder nicht. Wir empfehlen, dies von der Leistungsfähigkeit der Lerngruppe abhängig zu machen. Wenn der Algorithmus behandelt wird, hat es sich bewährt, das RSA-Verfahren zunächst mit kleinen Zahlen per Hand anzuwenden. Das methodische Vorgehen dazu wird in einem Artikel von Witten/Schulz (2006a) beschrieben; Arbeitsbögen finden sich in der Sammlung der Materialien zur E-Mail-Reihe ab Seite 30 (siehe Gramm u.a., 2012).

Der Einsatz der RSA-Demo von *CrypTool 1* wird in zweifacher Hinsicht motiviert: Bei mathematisch schwachen Lerngruppen arbeiten die Algorithmen aus diesem Programm zum schnellen, modularen Potenzieren („square and multiply“) sowie der Algorithmus zur Berechnung der modularen Inversen (erweiterter Euklidischer Algorithmus) als „Blackbox“ im Hintergrund, ohne dass man dies problematisieren muss. Wenn der Algorithmus aber vorher besprochen wurde, kann das Verständnis durch die Anwendung in einer neuen Programmumgebung vertieft werden.

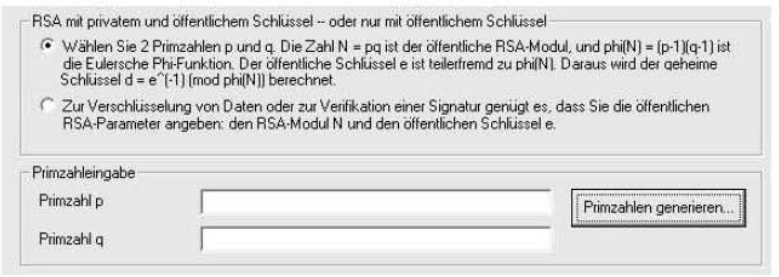
Das nächste Argument gilt für beide Typen von Lerngruppen: Man kann bei zu klein gewähltem Modul N mit der RSA-Demo das Verfahren brechen, indem der Modul faktorisiert und aus der Kenntnis der Faktoren p und q der geheime Schlüssel berechnet wird.

In dem Arbeitsbogen „Anleitung: Ver- und Entschlüsseln mit der RSA-Demo von *CrypTool*“ (siehe Bild 7, nächste und übernächste Seite) wird Schritt für Schritt erklärt, wie man ein Schlüsselpaar mit diesem Programmsystem generieren und zur Verschlüsselung verwenden kann. Da auch die Möglichkeit besteht, Texte in Zahlenfolgen und umgekehrt zu verwandeln, kann man so einfach Botschaften übermitteln. Dazu müssen die Kommunikationsteilnehmer lediglich ihren

Anleitung: Ver- und Entschlüsseln mit der RSA-Demo von *CrypTool*

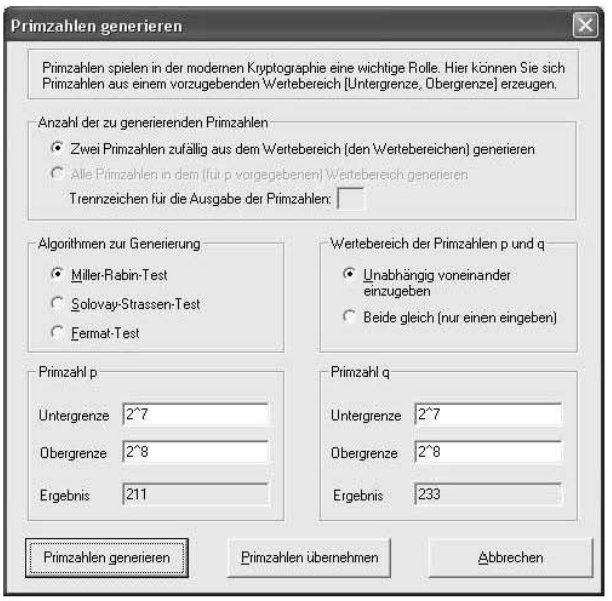
1. RSA-Schlüsselpaar generieren:

Starte *CrypTool* und rufe im Menü *Einzelverfahren* → *RSA-Kryptosystem* → *RSA-Demo...* auf! Es erscheint ein (auf den ersten Blick etwas unübersichtliches) Fenster. Betrachte zunächst nur den oberen Ausschnitt:



Es gibt zwei Möglichkeiten, die Schlüssel **(e; N)** und **d** zu konstruieren:

- zwei Primzahlen in die Felder für p und q eintragen oder
- die Primzahlen von *CrypTool* wie folgt erzeugen lassen: Knopf „Primzahlen generieren" drücken. Ein neues Fenster erscheint mit den voreingestellten Werten:



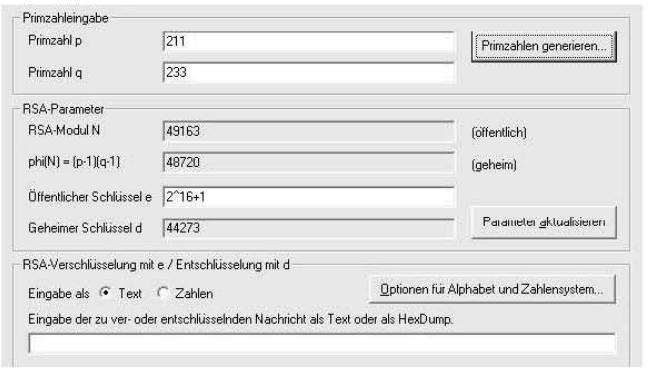
Wir haben also zwei Primzahlen zwischen $2^7 = 128$ und $2^8 = 256$ erhalten. Wenn uns diese Zahlen nicht gefallen sollten, drücken wir ggf. mehrfach „Primzahlen generieren" und erhalten dann andere Primzahlen aus diesem Bereich, z. B. 227 und 251. *CrypTool* benutzt Pseudozufallszahlen, die stets in der gleichen Reihenfolge auftreten, es macht also Sinn, mehrmals auf den Knopf zu drücken!

Klicke nun auf „Primzahlen übernehmen". Es erscheint wieder der Ausgangsschirm, aber neben den Primzahlen p und q sind bereits der **RSA-Modul N** und **phi(N) = (p-1)(q-1)** eingetragen. Als öffentlicher Schlüssel e ist immer $2^{16}+1 = 65537$ voreingestellt[1]. Wem diese Zahl nicht gefällt, kann auch hier eine andere eintragen. Diese muss aber teilerfremd zu phi(N) sein!

[1] Wer wissen will, warum gerade diese Zahl bevorzugt wird, sollte sich z. B. mit Hilfe des Windows-Taschenrechners ihre Darstellung im Dualsystem anschauen

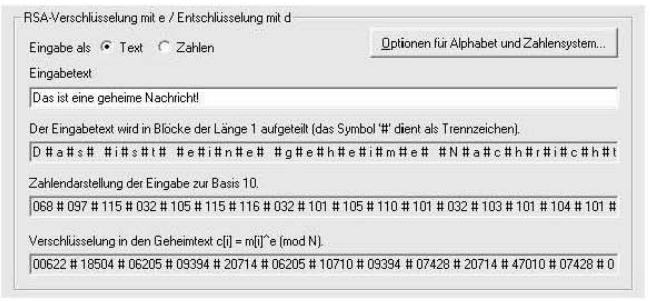
Fortsetzung nächste Seite

Der zugehörige geheime Schlüssel wird ebenfalls automatisch erzeugt:



2. Verschlüsseln:

- Gib einen Text in das untere Eingabefeld ein!
- Klicke auf „Verschlüsseln“ um ihn zu **verschlüsseln**!

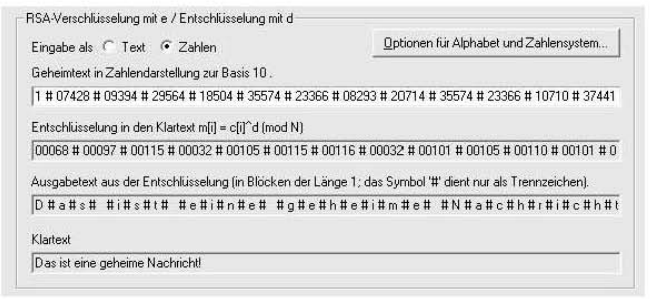


Wegen des relativ kleinen Moduls **N** wird der Text in Blöcke der Länge 1 unterteilt[2] und als Zahlen dargestellt (die entsprechenden ASCII-Nummern). Diese werden dann Block für Block (bei Blocklänge 1 also Zeichen für Zeichen) verschlüsselt.

3. Entschlüsseln:

Trotzdem wollen wir uns überzeugen, dass sich dieser „Geheimtext“ wieder korrekt **entschlüsseln** lässt:

- Kopiere den Geheimtext in die Eingabezeile!
- Klicke auf den Knopf „Zahlen“! (Wenn Du das vergisst, weißt dich *CrypTool* darauf hin.)
- Klicke auf den Knopf „Entschlüsseln“! Ergebnis:



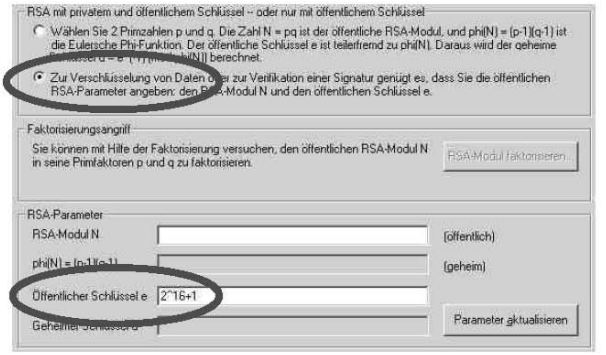
[2] Bei einem so kleinen Modul N handelt es sich um eine schlichte monoalphabetische Verschlüsselung, die ein Knacken per Häufigkeitsanalyse erlaubt. Bei größeren Primzahlen werden jedoch stets mehrere Zeichen in einem Block zusammengefasst.

Quelle: Gramm u. a., 2012, S. 36–37

Bild 7 (links und oben): Arbeitsbogen „Anleitung: Ver- und Entschlüsseln mit der RSA-Demo von *CrypTool*“.

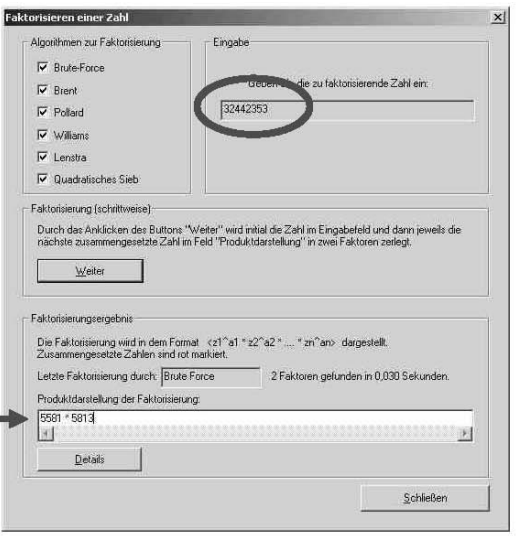
Anleitung: RSA knacken mit *CrypTool*

1. Starte CrypTool und rufe im Menü *Einzelverfahren → RSA-Kryptosystem → RSA-Demo...* auf!
   Wähle diesmal den zweiten Radioknopf „Zur Verschlüsselung von Daten..."!
   In diesem Fall können nur der RSA-Modul N und der öffentliche Schlüssel e eingegeben werden (e ist wieder auf $2^{16}+1$ voreingestellt, das kann aber geändert werden).



2. Gib einen RSA-Modul N ein!

3. Klicke nun den Knopf „RSA-Modul faktorisieren"! Wenn die eingegebene Zahl ein gültiger RSA-Modul (und nicht zu groß) ist, werden die beiden Primfaktoren p und q in dem Faktorisierungsfenster von CrypTool gefunden. Falls N keine Semiprimzahl ist, wird eine entsprechende Fehlermeldung ausgegeben.



Erfolgreiche Faktorisierung des RSA-Moduls N= 32442353.

Nach Schließen dieses Fensters werden die Primfaktoren p und q im normalen Fenster vom RSA-Demo eingetragen, außerdem werden sogleich phi(N) und der geheime Schlüssel d berechnet. Man hat damit wieder ein voll funktionsfähiges RSA-System, obwohl nur der öffentliche Schlüssel bekannt war!

Merke: **Die Sicherheit von RSA wird ganz wesentlich von der Schlüssellänge des gewählten RSA-Moduls bestimmt!**

Forschungsauftrag

Wieviel Bit muss ein RSA-Modul haben, damit er nicht mehr innerhalb von wenigen Sekunden mit *CrypTool* in der oben beschriebenen Weise geknackt werden kann?

Hinweis: Wie kann ich z. B. einen RSA-Modul mit 64 Bit erzeugen? Dafür werden zwei Primfaktoren p und q mit jeweils 32 Bit Länge benötigt. Man gebe also im Fenster „Primzahlen generieren..." als Untergrenze jeweils 2^{31} und als Obergrenze 2^{32} ein. Nach den Klicks auf „Primzahlen generieren" und danach „Primzahlen übernehmen" hat man zwei Primzahlen p und q mit 32 Bit Länge und einen Modul N = p·q mit 64 Bit Länge erzeugt!

Quelle: Gramm u. a., 2012, S. 38

Bild 8: Arbeitsbogen „Anleitung: RSA knacken mit *CrypTool*".

öffentlichen Schlüssel bekannt geben. Hier bedarf es wiederum einiger Übungen durch die Lernenden, um die nötige Sicherheit im Umgang mit der RSA-Demo zu gewinnen.

Im nächsten Schritt wird mit einem weiteren Arbeitsbogen (siehe Bild 8, vorige Seite) ergründet, wie sich der RSA-Algorithmus „knacken“ lässt. Die Anleitung aus dem letzten Arbeitsbogen führt dazu, dass die Lernenden einen relativ kleinen Modul *N* generieren, der sich mit *CrypTool* sehr einfach faktorisieren lässt.

Die RSA-Demo ist so konstruiert, dass der Aufruf des Faktorisierungs-Moduls automatisch angeboten wird, wenn lediglich der öffentliche Schlüssel eingetragen wird. Ruft man die Faktorisierung auf, erscheint das entsprechende Fenster (siehe die Abbildung auf dem Arbeitsbogen in Bild 8, vorige Seite). In einem Forschungsauftrag werden die Lernenden angehalten, systematisch auszutesten, bis zu welcher Schlüsselgröße der Modul *N* von *CrypTool 1* noch in angemessener Zeit (höchstens ein paar Minuten) zerlegt werden kann.

Wenn man die mit *CrypTool 1* möglichen Zerlegungen genauer eingrenzen will, ergeben sich auf einem handelsüblichen Laptop folgende Werte: Eine Semiprimzahl mit 315 Bit kann nicht zerlegt werden, ein RSA-Schüssel mit 235 Bit wird mit dem quadratischen Sieb in knapp 41 Minuten zerlegt, die anderen implementierten Methoden haben bei dieser Länge keine Chance (vgl. Schulz/Witten, 2010).

Für ein vertieftes Verständnis der Fragen rund um die Sicherheit von RSA verweisen wir auf die aktuelle Folge der Beitragsserie von Helmut Witten und Ralph-Hardo Schulz in diesem Heft (S. 59 ff.).

Die Unterrichtsreihe sollte durch praktische Übungen z.B. mit *Enigmail* (vgl. Wikipedia – Stichwort „Enigmail“) – einem auf *PGP* (vgl. Wikipedia – Stichwort „Pretty Good Privacy“) aufsetzenden Erweiterungsmodul für das E-Mail-Programm Mozilla Thunderbird (vgl. Wikipedia – Stichwort „Mozilla Thunderbird“) – abgerundet werden. Hierzu sind wieder in den PDF-Materialien von Gramm u.a. (2012) entsprechende Anleitungen bereitgestellt.

Grundbildung in Fragen der Computersicherheit für alle!

Anfang Mai 2012 erschien in der allgemein als seriös geltenden Wochenzeitung *DIE ZEIT* ein eher reißerischer Artikel unter dem Titel „Zufällige Sicherheit“ (vgl. Strassmann, 2012). Hier einige Zitate daraus: „Skandal“ – „Das Internet wackelt“ – „Die Sicherheit durch Verschlüsselung ist nicht garantiert“ – „Unser Vertrauen in die Geschäftsgrundlage des Webs als Marktplatz ist möglicherweise nicht mehr zu rechtfertigen. Denn einem Team europäischer und amerikanischer Mathematiker und Kryptografiespezialisten ist es gelungen, die bislang beliebteste, als sicher geltende Web-Verschlüsselung RSA zu attackieren und teilweise zu knacken.“

Die Leserinnen und Leser reagierten zum Teil verunsichert: „Der Artikel klärt nicht wirklich auf, sondern lässt den Leser mit dem Gefühl *Ich verwende besser kein Online-Banking mehr* alleine“ schrieb *Chandler81* in den Leserkommentaren von *ZEIT-ONLINE*. Der Leser *rumblebelly* – mit offenbar gründlicheren Kryptografie-Kenntnissen – schrieb zu Recht: „Zu viel Halbwissen steckt in diesem Artikel.“ (Zu den wirklichen Hintergründen dieses „Skandals“ vgl. Witten/Schulz, S. 59 f. in diesem Heft, oder Esslinger u.a., 2012.)

Letztlich stützt sich der *ZEIT*-Artikel nur auf eine im Februar veröffentlichte Untersuchung, die zeigte, dass es in seltenen Fällen (bei ca. 10000 von 11 Millionen untersuchten Zertifikaten) vorkam, dass die Schlüssel nicht gemäß den bekannten Standards generiert wurden und sich dadurch brechen ließen. Das RSA-Verfahren als solches ist davon gänzlich unberührt.

Wenn man den Anspruch ernst nimmt, die Jugendlichen im Informatikunterricht über Chancen und Risiken der Kommunikation mithilfe des Internets aufzuklären, ist die Frage nach der Sicherheit der dabei verwendeten Protokolle und Algorithmen zentral. Stimmt es z.B., dass „jede Verschlüsselung der Internetkommunikation […] geknackt werden [kann], sogar die bislang zuverlässigste“, wie Herr Strassmann gleich am Anfang seines Artikels behauptet? Diese Aussage ist schlicht falsch, weil es ja das beweisbar sichere Verfahren One-Time-Pad gibt (s.o.) – was auch dem Leser *rumblebelly* aufgefallen ist.

Man kann dem RSA-Verfahren also weiterhin vertrauen. Voraussetzung dafür war schon immer, dass man die Schlüssel „richtig“ generiert (also genügend lang und genügend zufällig).

Und falls ein genialer Mathematiker doch ein neuartiges Faktorisierungsverfahren findet (wie z.B. im Film „Sneakers – Die Lautlosen“ aus dem Jahr 1992; siehe auch die Seiten 63 und 64 in diesem Heft), dann wird es mit den sogenannten Post-Quantum-Algorithmen auch dafür Ersatz geben.

Wichtig ist uns, dass die Schülerinnen und Schüler die richtigen Software-Werkzeuge (wie z.B. *Enigmail*) kennen und auch keine Scheu vor deren Einsatz haben. Außerdem sollte ihre gesellschaftliche Kompetenz gestärkt werden, so dass sie kritisch hinterfragen können, wenn Gesetzesinitiativen mit positiven Begründungen vorgelegt werden, diese aber letztlich unausgereift bis gefährlich sind (wie z.B. im Juni 2012 das inzwischen gestoppte Meldegesetz oder das inzwischen ad acta gelegte Elena-Verfahren, die beide den Datenschutz der Bürger erheblich verletzten).

Helmut Witten
Brandenburgische Straße 23
10797 Berlin

E-Mail: helmut@witten-berlin.de

Prof. Bernhard Esslinger
Universität Siegen
Institut für Wirtschaftsinformatik
Hölderlinstraße 3
57076 Siegen

E-Mail: esslinger@fb5.uni-siegen.de

Andreas Gramm
1. Schulpraktisches Seminar
Charlottenburg-Wilmersdorf
Otto-Suhr-Allee 100
10585 Berlin

E-Mail: gramm@gymnasium-tiergarten.de

Malte Hornung
Freie Universität Berlin
Didaktik der Informatik
Königin-Luise-Straße 24–26
14195 Berlin

E-Mail: hornung@inf.fu-berlin.de

Anmerkung: Dieser Artikel ist eine stark überarbeitete und erweiterte Fassung des Beitrags „Asymmetrische Kryptographie für die Sek I – RSA (fast) ohne Mathematik" aus dem von M. Weigend, M. Thomas und F. Otte herausgegebenen Tagungsband „Informatik mit Kopf, Herz und Hand – Praxisbeiträge zur INFOS 2011" (Zentrum für Lehrerbildung, Münster, 2011, S. 225–235).

Im **LOG-IN-Service** (siehe Seite 123) steht eine PDF-Datei mit Bildschirmfotos zum „Signieren und Verschlüsseln einer Nachricht in fünf Schritten" und entsprechenden Erläuterungen zum Herunterladen bereit, die dem Video „Vertraulichkeit durch asymmetrische Kryptologie herstellen" entnommen sind.

Literatur und Internetquellen

Crowell, W.P.: Remembrances of Venona. 15. Januar 2009.
http://www.nsa.gov/public_info/declass/venona/remembrances.shtml

CrypTool 1:
http://www.cryptool.org/de/cryptool1

CrypTool-Portal:
http://www.cryptool.org/de/

Esslinger, B. u. a.: Asymmetrische Kryptologie am Beispiel RSA entdecken. Januar 2010.
https://www.cryptoportal.org/data/Asymmetrische%20Kryptologie%20am%20Beispiel%20RSA%20entdecken_v1.1.pdf

Esslinger, B.; Simon, V.; Schneider, J.: RSA-Sicherheit in der Praxis – Fehler in der Anwendung des RSA-Algorithmus. In: <kes> – Die Zeitschrift für Informations-Sicherheit, 28. Jg. (2012), Heft 2, S. 22–27.
http://www.cryptool.org/images/ctp/documents/kes_2012_RSA_Sicherheit.pdf

Gramm, A.: Animationen zur asymmetrischen Kryptographie. 2010.
http://it-lehren.de/asym

Gramm, A.; Hornung, M.; Witten, H.: E-Mail (nur?) für Dich – Eine Unterrichtsreihe des Projekts Informatik im Kontext. In: LOG IN, 31. Jg. (2011), Heft 169/170, Beilage.

Gramm, A.; Hornung, M.; Witten, H.: E-Mail (nur?) für Dich – Alle Arbeitsmaterialien (2012):
http://medienwissenschaft.uni-bayreuth.de/informatik-im-kontext/assets/Entwuerfe-Material/E-Mail-nur-fuer-dich/alleMaterialien.pdf

Koubek, J.: Informatik im Kontext – IniK für alle. 2007 ff.
http://www.informatik-im-kontext.de/

Leutbecher, A.: Zahlentheorie – Eine Einführung in die Algebra. Berlin u. a.: Springer, 1996. – Ausklapptafel: Sieb des Eratosthenes.
http://medienwissenschaft.uni-bayreuth.de/inik/email_nur_fuer_dich/3_verschluesseln/3.3_asymmetrisch_verschluesseln/Sieb%20des%20Eratosthenes.pdf

Meyer, H.: Leitfaden Unterrichtsvorbereitung. Berlin: Cornelsen Verlag Scriptor, ^{5}2007.

Meyer, H.-B.: Das Sieb des Eratosthenes. 2012.
http://www.hbmeyer.de/eratosib.htm

mpfs – Medienpädagogischer Forschungsverbund Südwest (Hrsg.): JIM-Studie 2011 – Jugend, Information, (Multi-)Media. Basisstudie zum Medienumgang 12- bis 19-Jähriger in Deutschland. Stuttgart: Medienpädagogischer Forschungsverbund Südwest, 2011.
http://www.mpfs.de/fileadmin/JIM-pdf11/JIM2011.pdf

Müller, J.: Einwegfunktionen. In: LOG IN, 31. Jg. (2011), Heft 169/170, S. 106–111.

NSA – National Security Agency: VENONA Chronology. 2009.
http://www.nsa.gov/public_info/declass/venona/chronology.shtml

Python Programming Language – Official Website:
http://www.python.org/

RSA Inc.: RSA Faktorisierungs-Challenge – RSA-1024. Januar 2012.
http://www.mysterytwisterc3.org/images/challenges/mtc3-rsa-15-de.pdf

SAGE-Computer-Algebra-System:
http://www.sagemath.org/

Schmeh, K.: Kryptografie – Verfahren, Protokolle, Infrastrukturen. Heidelberg: dpunkt.verlag, ^{4}2009.

Schulz, R.-H.; Witten, H.: Zeit-Experimente zur Faktorisierung – Ein Beitrag zur Didaktik der Kryptologie. In: LOG IN, 30. Jg. (2010), Heft 166/167 S. 107–114.

Strassmann, B.: Zufällige Sicherheit. In: DIE ZEIT, 67. Jg., Nr. 19 vom 3. Mai 2012, S. 37, und ZEIT-ONLINE – Datenschutz (05.05.2012).
http://www.zeit.de/2012/19/N-zufaellige-Zahlenreihen/komplettansicht

Wikipedia – Stichwort „Enigmail":
http://de.wikipedia.org/wiki/Enigmail

Wikipedia – Stichwort „Eratosthenes":
http://de.wikipedia.org/wiki/Eratosthenes

Wikipedia – Stichwort „Faktorisierungsverfahren":
http://de.wikipedia.org/wiki/Faktorisierungsverfahren

Wikipedia – Stichwort „Kasiski-Test":
http://de.wikipedia.org/wiki/Kasiski-Test

Wikipedia – Stichwort „One-Time-Pad":
http://de.wikipedia.org/wiki/One-Time-Pad

Wikipedia – Stichwort „Pretty Good Privacy":
http://de.wikipedia.org/wiki/Pretty_Good_Privacy

Wikipedia – Stichwort „Mozilla Thunderbird":
http://de.wikipedia.org/wiki/Mozilla_Thunderbird

Wikipedia – Stichwort „VENONA-Projekt":
http://de.wikipedia.org/wiki/VENONA-Projekt

Witten, H.; Letzner, I.; Schulz, R.-H.: RSA & Co. in der Schule – Moderne Kryptologie, alte Mathematik, raffinierte Protokolle. Teil 2: Von Cäsar über Vigenère zu Friedman. In: LOG IN, 18. Jg. (1998), Heft 5, S. 31–39.

Witten, H.; Letzner, I.; Schulz, R.-H.: RSA & Co. in der Schule – Moderne Kryptologie, alte Mathematik, raffinierte Protokolle. Teil 3: Flußchiffren, perfekte Sicherheit und Zufall per Computer. In: LOG IN, 19. Jg. (1999), Heft 2, S. 50–57.

Witten, H.; Schulz, R.-H.: RSA & Co. in der Schule – Moderne Kryptologie, alte Mathematik, raffinierte Protokolle. Neue Folge – Teil 1: RSA für Einsteiger. In: LOG IN, 26. Jg. (2006a), Heft 140, S. 45–54.

Witten, H.; Schulz, R.-H.: RSA & Co. in der Schule – Moderne Kryptologie, alte Mathematik, raffinierte Protokolle. Neue Folge – Teil 2: RSA für große Zahlen. In: LOG IN, 26. Jg. (2006b), Heft 143, S. 50–58.

Witten, H.; Schulz, R.-H.: RSA & Co. in der Schule – Moderne Kryptologie, alte Mathematik, raffinierte Protokolle. Neue Folge – Teil 3: RSA und die elementare Zahlentheorie. In: LOG IN, 28. Jg. (2008), Heft 152, S. 60–70.

Witten, H.; Schulz, R.-H.: RSA & Co. in der Schule – Moderne Kryptologie, alte Mathematik, raffinierte Protokolle. Neue Folge – Teil 4: Gibt es genügend Primzahlen für RSA? In: LOG IN, 30. Jg. (2010a), Heft 163/164, S. 97–103.

Witten, H.; Schulz, R.-H.: 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, 30. Jg. (2010b), Heft 166/167, S. 92–106.

Witten, H.; Schulz, R.-H.: 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, 31. Jg. (2011), Heft 172/173, S. 59–69 *(in diesem Heft)*.

Alle Internetquellen wurden zuletzt am 31. August 2012 geprüft.