Staatliche Fachober- und Berufsoberschule München
Ausbildungsrichtung Technik und Agrarwirtschaft,
Bio- und Umwelttechnologie

Seminararbeit

Der RSA-Algorithmus

Seminar Kryptographie (Nr. 08)

Verfasser:
Suriyaa Sundararuban (13 B)

Seminarlehrkraft:
StR Landthaler

Schuljahr:
2017/2018

Abgabetermin:
15. Januar 2018

Angestrebter akademischer Grad

Die allgemeine Hochschulreife

Quelle: Erik Tews [1]

Abbildung 1: Adi Shamir, einer der drei Erfinder des RSA-Algorithmus

„Crypto will not be broken, it will be bypassed.“ [2]

Adi Shamir

[1] Tews, E. *Datei:Adi Shamir at TU Darmstadt (2013).jpg*. Abgerufen am: 13. Oktober 2017. url: https://commons.wikimedia.org/wiki/File:Adi_Shamir_at_TU_Darmstadt_(2013).jpg. [61]

[2] Shamir, A. „Crypto will not be broken, it will be bypassed“. Abgerufen am: 29. September 2017. url: http://www.azquotes.com/quote/1218393. [60]

Inhaltsverzeichnis

Quelle: RSA Security LLC. [3]

Abbildung 2: Das Logo des Unternehmens „RSA Security LLC.“.

„Crypto will not be broken, it will be bypassed.“[4] Durch dieses Zitat von dem RSA-Mitentwickler Adi Shamir wurde meine Interesse an Kryptographie geweckt, mich mit Krypto-Algorithmen ausführlich auseinanderzusetzen. Demzufolge wird sich meine Seminararbeit mit der Thematik des RSA-Algorithmus befassen. In dieser Seminararbeit werden neben mathematischen Grundlagen auch Bezüge auf praktische Anwendungen des RSA-Algorithmus gemacht, wie zum Beispiel die Verwendung des RSA-Algorithmus im Internet und in anderen Kommunikation-Infrastrukturen. Es wird hier deutlich gezeigt, dass der RSA-Algorithmus heutzutage eine durchaus wichtige Rolle in der Praxis spielt. Zum Abschluss meiner Arbeit werde ich ein Fazit aus der Problematik von RSA und dessen Bedeutung ziehen.

Alles begann im Jahre 1976: Der Kryptologe Martin Hellman publizierte zusammen mit dem Kryptographie-Experten Whitfield Diffie eine Arbeit über das Prinzip der Public-Key-Kryptographie, dass erst 1980 am United States Patent and Trademark Office (USPTO) als Patent angemeldet wurde. Bei diesem Verfahren sollt es nicht möglich sein den Entschlüsselungsschlüssel aus dem Verschlüsselungsschlüssel zu ermitteln.[5]

[3] RSA Security Inc. *customer-image-2-rsa.jpg*. Abgerufen am: 08. Dezember 2017. url: https://www.rsa.com/content/dam/images/11-2016/customer-image-2-rsa.jpg. [58]

[4] Shamir, A. „Crypto will not be broken, it will be bypassed“. Abgerufen am: 29. September 2017. url: http://www.azquotes.com/quote/1218393. [60]

[5] Hellman, M., Diffie, B. und Merkle, R. Cryptographic apparatus and method. US Patent 4,200,770. Abgerufen am: 29. September 2017. Apr. 1980. url: https://www.google.com/patents/US4200770. [41]

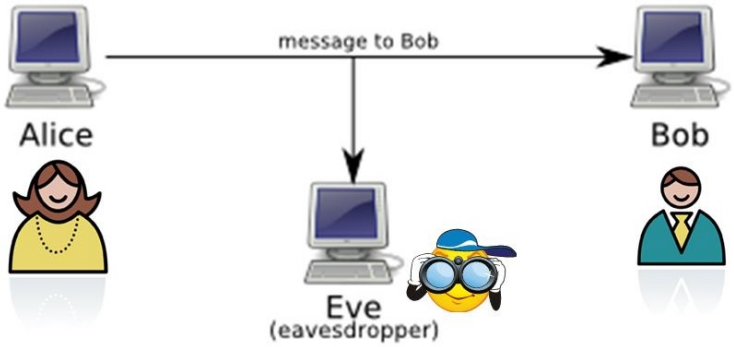
Das führt zu zwei wesentlichen Problemen:

1. das Schlüsselmanagementproblem:

   Jeder Teilnehmer muss mit jedem einen gemeinsamen Schlüssel vereinbaren. Kommt ein neuer Teilnehmer hinzu, muss dieser mit jedem bisherigen Anwender einen eigenen Schlüssel vereinbaren. Dies bedeutet einen großen Aufwand, wenn weitere Teilnehmer hinzukommen.[6]

2. Nachteil in der Symmetrie zwischen Ver- und Entschlüsselungsverfahren:

   Aus dem Verschlüsselungsschlüssel kann man mit Rechenaufwand den Entschlüsselungsschlüssel berechnen, da beide Schlüssel identisch sind beziehungsweise die gleichen Eigenschaften haben. Fängt ein Dritter - in dem Fall Eve (siehe Abbildung 3) - die verschlüsselte Nachricht C ab und kennt den Verschlüsselungsschlüssel, kann Eve die verschlüsselte Nachricht C auch entschlüsseln.[7]



Quelle: Wellesley-College [8]

Abbildung 3: Der Diffie-Hellman-Schlüsselaustausch, die Grundlage für den RSA-Algorithmus

[6] Busse, M., Schmitt, M. und Steeg, J. Der RSA-Algorithmus. Abgerufen am: 30. September 2017. url: http://www.zum.de/Faecher/Inf/RP/infschul/kr_rsa.html. [20]

[7] Rivest, R. L., Shamir, A. und Adleman, L. A Method for Obtaining Digital Signatures and Public-Key Cryptosystems. Communications of the ACM (Association for Computing Machinery) & Laboratory for Computer Science (Massachusetts Institute of Technology). url: https://people.csail.mit.edu/rivest/Rsapaper.pdf. Feb. 1978. [1]

[8] CS110-Team der Wellesley-College. *aliceBob.jpg*. Abgerufen am: 28. Oktober 2017. url: http://cs110.wellesley.edu/reading/cryptography-files/aliceBob.jpg. [27]

Kapitel 1

Was ist RSA?

Quelle: Massachusetts Institute of Technology[9]

Abbildung 1.1: Adi Shamir, Ronald Rivest und Leonard Adleman in den 1970er-Jahren (von links)

Daraufhin entwickelten Ronald Linn Rivest, Adi Shamir und Leonard Adleman (siehe Abbildung 1.1) 1977 am Massachusetts Institute of Technology (MIT) den RSA-Algorithmus. RSA ist ein asymmetrischer Public-Key-Algorithmus, um Klartexte M zu verschlüsseln beziehungsweise verschlüsselte Texte C wieder zu entschlüsseln. RSA wurde nach den drei Erfindern Rivest, Shamir und Adleman benannt. Sie stellten dieses Verfahren in ihrem Whitepaper unter dem Titel „A Method for Obtaining Digital Signatures and Public-Key Cryptosystems“der Öffentlichkeit vor, dass in der wissenschaftlichen Zeitschrift Sientific American[10] und in der Ausgabe der Association for Computing Machinery vom September 1977 herausgegeben wurde. [11]

[9] Massachusetts Institute of Technology (MIT). *rsa-photo.jpeg*. Abgerufen am: 30. Oktober 2017. url: http://people.csail.mit.edu/rivest/photos/rsa-photo.jpeg. [46]

[10] Greenemeier, L. Can't Touch This–New Encryption Scheme Targets Transaction Tampering. Abgerufen am 08. November 2017. Mai 2015. url: https://www.scientificamerican.com/article/can-t-touch-this-new-encryption-scheme-targets-transaction-tampering/. [39]

[11] Rivest, R. L., Shamir, A. und Adleman, L. A Method for Obtaining Digital Signatures and Public-Key Cryptosystems. Communications of the ACM (Association for Computing Machinery) & Laboratory for Computer Science (Massachusetts Institute of Technology). url: https://people.csail.mit.edu/rivest/Rsapaper.pdf. Feb. 1978. [1]

Es ist der erste Algorithmus der sowohl für digitale Signaturen als auch für Verschlüsselungen benutzt werden konnte. Er war einer der ersten großen Fortschritte in der Public-Key-Kryptographie. RSA gilt bei sehr langen Schlüsseln und bei korrekter Implementierungen als sicher. Der RSA-Algorithmus war seit 14. Dezember 1977 patentiert und ist seit September 2000 wieder frei, ohne Lizenzgebühren, verwendbar.[12]

Der RSA-Algorithmus erfüllt die von Whitfield Diffie und Martin Hellman geforderten Spezifikationen:

- Der Algorithmus muss eine grundlegende Verschiedenheit zwischen Ver- und Entschlüsselungsschlüssel aufzeigen.
- Der Verschlüsselungsschlüssel P wird öffentlich bekannt gegeben, der Entschlüsselungsschlüssel S bleibt geheim.

Der Algorithmus besteht aus zwei Teilen: Neben dem eigentlichen Verschlüsselungsalgorithmus ist die Schlüsselerzeugung von großer Wichtigkeit für die Sicherheit des Verfahrens.

[12] Rivest, R. L., Shamir, A. und Adleman, L. Cryptographic communications system and method. US Patent 4,405,829. Abgerufen am: 29. September 2017. Sep. 1983. url: https://www.google.com/patents/US4405829. [53]

Kapitel 2

Funktionsweise des RSA-Algorithmus

2.1 Grundlagen[13]

In einem „Public-Key-Kryptosystem“ stellt jeder Benutzer seinen öffentlichen Schlüssel P der Öffentlichkeit zur Verfügung. Jeder kann mit Hilfe des öffentlichen Schlüssels eine Nachricht M verschlüsseln. Dieser Verschlüsselungsverfahren wird als E bezeichnet. Der Benutzer hält die Details seines privaten Schlüssels S für den entsprechenden Entschlüsselungsverfahren D geheim.

(a) Die verschlüsselte Form einer Nachricht $E(M)$ wird zur Nachricht M entschlüsselt:

$$D(E(M)) = M \tag{2.1}$$

(b) Somit kann man mit E und D effizienter rechnen.

(c) Bei der öffentlichen Freigabe von E kann ein Nutzer daraus nicht D berechnen.

(d) Wenn man zuerst eine Nachricht M entschlüsselt und dann verschlüsselt, ist M das Resultat:

$$E(D(M)) = M \tag{2.2}$$

RSA liefert öffentliche Schlüsseln, die an bestimmte private Schlüsseln gebunden sind. Verfügt Alice über den öffentlichen Schlüssel P von Bob, so kann sie eine Nachricht für ihn verschlüsseln und schicken, die er mit seinem zugehörigen privaten Schlüssel S entschlüsseln. Mittels RSA kann man mit einem privaten Schlüssels S auch Daten verschlüsseln, die jeder mit Hilfe des zugehörigen öffentlichen Schlüssels P wieder entschlüsseln kann. Das ermöglicht die Implementierung von digitaler Signaturen, worauf ich in Kapitel 4.2 eingehen werde. Diese stellen sicher, dass eine kryptographische Operation vom Besitzer eines bestimmten privaten Schlüssels S durchgeführt wurde.

[13] Rivest, R. L., Shamir, A. und Adleman, L. A Method for Obtaining Digital Signatures and Public-Key Cryptosystems. Communications of the ACM (Association for Computing Machinery) & Laboratory for Computer Science (Massachusetts Institute of Technology). url: https://people.csail.mit.edu/rivest/Rsapaper.pdf. Feb. 1978. [1]

2.2 Schlüsselerzeugung

Damit Alice eine RSA-verschlüsselte Nachricht C an Bob versenden kann, muss Bob zunächst sein eigenen RSA-Schlüsselpaar generieren. Die RSA-Schlüsselerzeugung läuft dabei wie folgt ab:

1. Man wählt zufällig zwei große Primzahlen p und q, die $p \neq q$ sein sollten. Die Länge der Primzahlen sollte mindestens 512 Bit betragen. In der Praxis werden Zufallszahlen mit dem Miller-Rabin-Algorithmus generiert und getestet.[14]

2. Im Folgenden berechnet man das Produkt der Primzahlen p und q:

$$n = p \cdot q \tag{2.3}$$

   n bezeichnet man als „RSA-Modul“ und hat eine Mindestlänge von 1024 Bit.

3. Mit den beiden Primzahlen p und q kann man den benötigten Wert der Eulerschen ϕ-Funktion für n berechnen[15]:

$$\phi(n) = (p-1) \cdot (q-1) \tag{2.4}$$

4. Man wählt eine kleine ungerade natürliche Zahl $e \in \mathbb{N}$ mit $1 < e < \phi(n)$, die zu $\phi(n)$ teilerfremd ist[16]. Das heißt beide Zahlen e und $\phi(n)$ können nur mit der Zahl 1 geteilt werden.[17] Der größter gemeinsamer Teiler (ggT) von e und $\phi(n)$ ist somit immer 1:

$$ggT(e, \phi(n)) = 1 \tag{2.5}$$

   Dieser sogenannte Verschlüsselungsexponent e bildet zusammen mit dem RSA-Modul den öffentlichen Schlüssel $P = (e, n)$.

5. Berechne d als Lösung der Gleichung:

$$e \cdot d \bmod \phi(n) = 1 \tag{2.6}$$

   Man kann also d mit dem erweiterten Euklidischen Algorithmus berechnen.

6. Das Schlüsselpaar $P = (e, n)$ von Bob wird nach der Schlüsselgenerierung als öffentlicher Schlüssel veröffentlicht.

[14] Wan Han, D. „Generating strong prime numbers for RSA using probabilistic Rabin-Miller algorithm“. Abgerufen am 17.11.2017. Electrical und Computer Engineering Department (George Mason University). url: http://ece.gmu.edu/coursewebpages/ECE/ECE646/F09/project/reports_1999/dong_report.pdf. [6]

[15] Heitkötter, H. „Public-Key-Verfahren: RSA/Rabin“. url: https://www.wi1.uni-muenster.de/pi/lehre/ws0708/seminar/Abgaben/RSA_Rabin.pdf. Westfälische Wilhelms-Universität Münster. [2]

[16] ebd.

[17] Die Serlo-Community. Teilerfremd. Abgerufen am 17. November 2017. url: https://de.serlo.org/mathe/zahlen-groessen/teiler-primzahlen/teiler-vielfache/teilerfremd. [29]

7. Das Schlüsselpaar $S = (d, n)$ von Bob wird nach der Schlüsselgenerierung als privater Schlüssel geheim gehalten.

Der öffentliche Schlüssel P besteht aus dem Verschlüsselungsexponenten e und dem RSA-Modul n. Man veröffentlicht P in einem Schlüsselverzeichnis zum Beispiel in einem „SKS Keyserver“. So ein Schlüsselserver bietet Zugang für Nutzer, um die öffentlichen Schlüssel der jeweiligen Person zu verifizieren.[18]

Quelle: Suriyaa Sundararuban (erstellt am 28. Dezember 2017 um 17:23:50)

Abbildung 2.1: Erzeugung des privaten RSA-Schlüssels in Microsoft Windows 10 [build 15063]

Der private Schlüssel S muss nach der Schlüsselgenerierung (siehe Abbildung 2.1) sicher gespeichert werden. Es empfiehlt sich den privaten Schlüssel S auf einem sicher aufbewahrten kryptographischen Hardwaregerät wie zum Beispiel auf einer Chipkarte zu speichern.[19]. In der Grundform besteht der private Schlüssel S nur aus dem RSA-Modul n und dem Entschlüsselungsexponenten d. Zur Beschleunigung der Entschlüsselung kann der private RSA-Schlüssel S weitere Elemente wie die Primzahlen p und q, aber auch die Funktion $\phi(n)$ enthalten. Diese Elemente dürfen auf keinesfalls veröffentlicht werden, da ein Angreifer aus diesen den privaten Schlüssel S ermitteln kann.[20]

[18] Fiskerstrand, K. sks-keyservers.net. Abgerufen am 11. Dezember 2017. url: https://sks-keyservers.net/. [34]

[19] GlobalSign, Inc. Schutz privater Schlüssel – So halten Sie Ihre Schlüssel geheim. Abgerufen am 11. Dezember 2017. url: https://www.globalsign.com/de-de/blog/schutz-des-private-keys/. [37]

[20] Lopes Gouvêa, C. P. Answer for 'What data is saved in RSA private key?' Abgerufen am 11. Dezember 2017. url: https://crypto.stackexchange.com/a/7964/25803. [44]

2.3 Verschlüsselung und Entschlüsselung[21]

- Verschlüsseln: Nun kann Alice ihren Klartext bzw. ihre Nachricht M mit Bobs öffentlichen Schlüssel P verschlüsseln:

$$E(M) = M^e \bmod n \tag{2.7}$$

  Sie erhält so aus ihrem Klartext M den Geheimtext C. Sie sendet den Geheimtext an Bob.

- Entschlüsseln: Bob bekommt den Geheimtext C von Alice. Die verschlüsselte Nachricht C kann durch modulares Potenzieren beim Entschlüsselungsverfahren D wieder zum lesbaren Klartext M entschlüsselt werden. Bob verwendet hier seinen privaten Schlüssel S, um den Geheimtext C zu entschlüsseln:

$$D(C) = C^d \bmod n \tag{2.8}$$

  Das modulares Potenzieren wird in der Mathematik auch als diskrete Exponentialfunktion[22] bezeichnet. Diese Funktion wird als Einwegfunktion in asymmetrischen Kryptosystemen wie in dem RSA-Kryptosystem verwendet. In dem Fall kennt nur Bob die Werte d und n, das in seinem privaten Schlüssel S (siehe Abbildung 2.1) vorhanden sind.

Listing 2.1: Inhalt eines zufällig generierten privaten RSA-Schlüssels mit der Größe von 1024 Bit

```
-----BEGIN RSA PRIVATE KEY-----
MIICXgIBAAKBgQuaj8MMO0vDvS5qUZVhDT5tIj35Xzl4Hu9rX4HRJD13gFU+WJvh
euh58tzDlAQ+Q6Oj/wDvkTo/VxG/QX4onTPVfMW0UsxnD/8wMzqAjCaI1bJnd0yn
d2AusFyyIFJ2utzdjSAEbEIQbDia00kmj9w0NZrGJCn2jaxSZDyoIPBdWQIDAQAB
AoGBAqTzdBiafraZqzUxaGBYm0xVjFgC+l7Edf07I6NcpeFQzpxJpyRoVQhNbz/m
wFzMH0OlPjpaFzMODY5PR4fL412KbXm0jAiknaevODRpLkqtF5knpVWWU0BsUrTc
lxUeS9TTEWLg86azMMEXJ5cxPdlo+8MgnNGNP0+tCTp8BOVxAkEDl+MKJlJwEVHB
jeMsMScaTRqvnEPzJl9LWSIihmsUFM+S80VhKmYwA6Z2GPLTtbskVvHcPxhTdvmo
VIDTWW8aVQJBAzqxVLEDwfMaK1GBtVCvHJxECCggJjCGLisSSy/EoOXnHdA9GKH1
hoHHBLowOwMLOvtrW86dfLnhzZ96tCxJgvUCQQG0Yj6EY4NKlBBQMT7bqoMtpMU+
SnXk9DwYcrsSLWX/YWGDCpzHsmaAtM09lKSeh8L7hA1J2U5vGIykWeDpgm1pAkEB
hfDZcvnDxWjFoBefr+Q5iZgphMvNV7wjENv7LRgBkRAYJv56nQKUJKj8mNphK4lj
k+5LCsWMbJ2PATe+vtkZ9QJAMO+1r4b2UV5LdHUj8bMNdHCRk6CC35vH6eqJJED1
HCHjSAmrQmtGQMtKCQx2lCCNotUYHnu/uCi4pC3iUOG5oA==
-----END RSA PRIVATE KEY-----
```

[21] Salomaa, A. Public-Key Cryptography. Berlin/Heidelberg: Springer-Verlag, 1990. Kap. 4, S. 125–157. isbn: 3-540-52831-8. [7]

[22] Die Wikipedia-Community. Diskrete Exponentialfunktion. Abgerufen am 07. November 2017. url: https://de.wikipedia.org/wiki/Diskrete_Exponentialfunktion. [30]

Kapitel 3

Vor- und Nachteile

3.1 Die Sicherheit des RSA-Algorithmus

Die Sicherheit von RSA basiert sich auf große Primzahlen, die man leicht miteinander multiplizieren kann. Die Zerlegung des Produkts bzw. des RSA-Moduls n in seine Primfaktoren p und q ist so gut wie unmöglich, da die Sicherheit des RSA-Verfahren auf dem Faktorisierungsproblem von ganzen Zahlen beruht. Aktuell ist kein mathematischer Algorithmus bekannt, der dieses Problem effizient bestimmen kann. Aus diesem Grund ist es bei großen Zahlen nicht möglich die Funktion $\Phi(n)$ zu berechnen. Also kann man damit nicht den privaten Schlüssel S aus dem öffentlichen Schlüssel P bestimmen. Das RSA-Modul n soll eine Mindestlänge von 2048 Bit haben. Des Weiteren sollten sich die Primzahlen p und q in ihrer Länge unterscheiden und nicht den gleichen Wert annehmen.[23] [24]

3.2 Zeitdauer der Rechenvorgänge

Das RSA-Verfahren hat auch einen Nachteil: Da die Nachricht M aus Sicherheitsgründen sehr groß sein muss, dauern die Rechenvorgänge, die zum Verschlüsseln und Entschlüsseln benötigt werden relativ lange. Außerdem ist der zu übermittelnde verschlüsselte Nachricht $E(M)$ wesentlich umfangreicher als der entsprechende Klartext. Trotzdem kommt RSA zum Beispiel bei Banken oder zur gesicherten Dateiübertragung im Internet sehr häufig zum Einsatz.[25]

[23] Rivest, R. L. und Silverman, R. D. „Are 'Strong' Primes Needed for RSA“. urls: http://eprint.iacr.org/2001/007 und https://people.csail.mit.edu/rivest/RivestSilverman-AreStrongPrimesNeededForRSA.pdf. IACR Cryptology ePrint Archive, Report paper 2001/007 (Version 1998-12-01 Submitted January 30, 2001) & MIT, 30. Jan. 2001. [5]

[24] Litzel, A. „Das RSA-Verfahren“. url: https://www7.in.tum.de/um/courses/seminar/krypto/SS09/litzel/zusammenfassung.pdf. Fakultät „Informatik“ an der Technischen Universität München (TUM), 2009. [3]

[25] Guggenberger, W. (Humboldt-Gymnasium Vaterstetten). Das RSA-Verfahren. Abgerufen am 10. November 2017. url: http://www.humboldt-gym.de/fileadmin/faecher/mathematik/krypto/rsatxt.html. [40]

3.3 Attacken auf den RSA-Algorithmus

3.3.1 Brute-Force-Angriffe auf RSA

Der erste Schritt zum Knacken des privaten RSA-Schlüssels ist die Bestimmung der beiden Primzahlen p und q, deren Produkt das RSA-Modul n liefert. Man nennt dieses Verfahren „Faktorisierung“. Die Faktorisierung spielt eine wichtige Rolle beim Brute-Force-Angriff. Man verwendet diese Methode, um alle möglichen Primzahlen eines RSA-Schlüssels mit einer bestimmten Schlüssellänge bis zur letzten Möglichkeit durchzuprobieren. Bisher hatte man nur eine 512 Bit-Variante des RSAs analysiert und erfolgreich geknackt.[26] Für die Faktorisierung hatte man 5,2 Monate gebraucht.[27] Das zeigt, dass das Knacken von RSA sehr rechenintensiv ist. Die Sicherheit von RSA hängt also vom Problem der Faktorisierung großer Zahlen ab, da sonst ein privater Schlüssel S mit einem Brute-Force-Angriff zur Faktorisierung von n knacken lässt.[28]

3.3.2 Faktorisierungs-Angriffe auf RSA

1977 erläuterten Riverst, Shamir und Adleman ihren Algorithmus gegenüber Martin Gardner von der Zeitschrift Scientific American, was zur ersten öffentlichen Beschreibung führte.[29] Die Erfinder waren sich der Tatsache bewusst, dass die Faktorisierung des RSA-Moduls eine Schwachstelle darstellt. Sie glaubten fest daran, dass die Faktorisierung ein schwieriges Problem bleiben würde. Sie sagten voraus, dass eine Zahl mit 129 Dezimalstellen jede verfügbare Computerleistung auch in näherer Zukunft überfordern würde.[30] [31]

3.3.3 RSA-Timing-Angriff

Es wurde festgestellt, dass der RSA-Algorithmus je nach Wert des Schlüssels unterschiedlich lange für die Durchführung seiner kryptographischen Operationen braucht.[32] Man kann inzwischen Schätzungen über den Wert eines privaten Schlüssels S abgeben und den Schlüssel rekonstruieren, wenn ein Angreifer die Zeit zur Anwendung des

[26] RSA Security, Inc. RSA-155 is factored! Abgerufen am 07. Dezember 2017. url: https://web.archive.org/web/20060616162727/http://www.rsasecurity.com/rsalabs/challenges/factoring/rsa155.html. [59]

[27] Clayton, R. (The Computer Laboratory at University of Cambridge). Brute force attacks on cryptographic keys - RSA. Abgerufen am 07. Dezember 2017. url: https://www.cl.cam.ac.uk/~rnc1/brute.html#RSAsection. [25]

[28] Smith, R. E. Internet-Kryptographie. Informationssicherheit. Bonn: Addison-Wesley, 1998. isbn: 3-8273-1344-9. [9]

[29] Greenemeier, L. Can’t Touch This–New Encryption Scheme Targets Transaction Tampering. Abgerufen am 08. November 2017. Mai 2015. url: https://www.scientificamerican.com/article/can-t-touch-this-new-encryption-scheme-targets-transaction-tampering/. [39]

[30] ebd.

[31] fgrieu (Stack Overflow). Answer for ’Is RSA vulnerable to possible PRNG + Miller Rabin test weaknesses?’ Abgerufen am 01. Dezember 2017. url: https://crypto.stackexchange.com/a/52729/25803. [33]

[32] Kocher, P. C. „Timing Attacks on Implementations of Diffie-Hellman, RSA, DSS, Other Systems“. In: CRYPTO 1996, Band 1109 in LNCS. url: https://link.springer.com/chapter/10.1007/3-540-68697-5_9. Abgerufen am 08.11.2017. Springer, 1996, S. 104–113. isbn: 978-3-540-61512-5. [11]

privaten Schlüssels auf eine bestimmte Information kennt. Ein entsprechender Angriff wird um so wahrscheinlicher, je näher ein Angreifer an den Prozess gelangt, der die kryptographischen Operationen durchführt. Beispiele dafür wären Hosting-Anbietern oder Clouds. Der Angriff ist nur dann durch durchführbar, wenn der Angreifer die Verarbeitungszeit genau messen kann. Falls dies der Fall ist, sollten Warteschleifen eingebaut werden.[33] Dadurch wird sichergestellt, dass der Angreifer aus der Berechnungszeit des RSA-Schlüssels keine Informationen über den privaten Schlüssel S ableiten kann.

3.3.4 Angriff auf kleine private Schlüssel „S“

Das RSA-Verfahren ist nach dem aktuellen Stand der Wissenschaft als sicher einzustufen. Bei der Wahl der Parameter sind allerdings einige Fallen zu vermeiden; insbesondere sollte als Schlüssellänge kurzfristig mindestens 2048 Bit, mittelfristig 4096 Bit gewählt werden.[34] Bei kleinen privaten Schlüsseln kann man sonst d und n leicht und schnell berechnen, weil die Werte zu klein sind.

3.3.5 Angriff auf kleinen Klartext „M“[35]

Hat der öffentliche Schlüssel P den Wert 3 und beträgt die Nachrichtenlänge weniger als ein Drittel der Länge des RSA-Moduls n, so gibt es Abkürzungen zur Ermittelung der Nachricht M. Ist die Nachricht kürzer als die Kubikwurzel von n, so kann die Nachricht M bestimmt werden, indem man einfach die Kubikwurzel von n berechnet. Die Datenformatierung gemäß PKCS (siehe Kapitel 4.3.2) verhindern diesen Angriff, indem immer eine Nachrichtenlänge größer als n erzwingt.

3.3.6 Angriff auf kleine RSA-Verschlüsselungsexponenten[36] [37] [38] [39]

Wird eine bestimmte Nachricht mit einem öffentlichen Schlüssel P mit einem kleinen Wert verschlüsseln, so gibt es einen Angriff, der den Klartext liefert. Dieser Angriff lässt sich jedoch verhindern, indem alle Nachrichten so mit Zufallsdaten aufgefüllt

[33] Smith, R. E. „Internet-Kryptographie“. In: Informationssicherheit. Bonn: Addison-Wesley, 1998. Kap. 9.2.3 - Technische Sicherheitsanforderungen, S. 225. isbn: 3-8273-1344-9. [9]

[34] Pommerening, K. Kryptoanalyse des RSA-Verfahrens. Abgerufen am 22. Dezember 2017. Mai 2000. url: https://www.staff.uni-mainz.de/pommeren/Kryptologie/Asymmetrisch/2_RSAanalyse/. [51]

[35] Alrasheed, A. und Fatima (The University of Tennessee at Chattanooga). RSA Attacks. Abgerufen am 10. November 2017. url: https://www.utc.edu/center-information-security-assurance/pdfs/course-paper-5600-rsa.pdf. [18]

[36] ebd.

[37] Wiener, M. J. Cryptanalysis of Short RSA Secret Exponents. Abgerufen am 10. Dezember 2017. url: https://www.cits.ruhr-uni-bochum.de/imperia/md/content/may/krypto2ss08/shortsecretexponents.pdf. [62]

[38] Boneh, D. (Computer science at Stanford University). Twenty Years of Attacks on the RSA Cryptosystem. Abgerufen am 10. Dezember 2017. url: https://www.ams.org/notices/199902/boneh.pdf. [19]

[39] Rivest, R. L. und Kaliski, B. „RSA Problem“. url: https://people.csail.mit.edu/rivest/RivestKaliski-RSAProblem.pdf. MIT Laboratory for Computer Science & RSA Laboratories, Dez. 2003. [4]

werden, dass keine zwei Nachrichten ähnlich sind. PKCS schreibt daher ein zufälliges Auffüllen von Zufallsdaten für alle mit RSA verschlüsselten Nachrichten vor.

3.3.7 Gewählter Chiffretext (chosen ciphertext)[40]

Dieser Angriff erlaubt dem Angreifer, eine ausgewählte Nachricht zu entschlüsseln, wenn er den Inhaber des privaten Schlüssels dazu bringen kann, eine mathematisch verwandte Nachricht zu ver- oder zu entschlüsseln. Eve, die Schnüfflerin, fängt eine Nachricht M von Alice an Bob ab, die mit Bobs öffentlichen Schlüssel P verschlüsselt ist. Bob kann die Nachricht M mit seinen privaten Schlüssel S entschlüsseln. Eve besitzt Bobs privaten Schlüssel S nicht. Sie kann Bob aber dazu bringen, seinen privaten Schlüssel S auf eine andere Nachricht anzuwenden, die nicht von Alice stammt, sondern von Eve. Eve konstruiert also eine Nachricht, die mathematisch mit der Nachricht von Alice verwandt ist. Bob entschlüsselt diese Nachricht für Eve und stellt dabei fest, dass er nur Unsinn enthält, bevor er die entschlüsselte Nachricht an Eve sendet. Eve kombiniert den vermeintlichen Unsinn mathematisch mit Alices ursprünglicher Nachricht an Bob. Dies liefert eine Klartextkopie von Alices Nachricht M.

3.4 Sicherheitsanforderungen bei der Schlüsselgenerierung[41]

Um die technischen Sicherheitsanforderungen für den sicheren Einsatz von RSA in der Praxis zu gewährleisten, muss man folgendes beachten:

1. Ausreichend lange Schlüssel: Es ist wichtig den öffentlichen Schlüssel P so lang wie möglich zu wählen, dass das Knacken genauso viel Aufwand erfordert wie beim geheimen Schlüssel S.

2. Effektive Schlüsselgenerierung: Die Schlüssel müssen mit einem wirklich zufälligen Startwert generiert werden. Bei der Primzahlerzeugung sollten mehrere Tests angewendet werden, um sicherzustellen, dass es sich um Primzahlen handelt.

3. Begrenzte Zugang zu den kryptografischen Funktionen: Der Zugang zu den Ver- und Entschlüsselungsoperationen E und D sollte auf sorgfältig geplante Anwendungsfunktionen beschränkt sein, die keine beliebigen Daten durchreichen. Dadurch werden Angriffe mit gewähltem Klartext oder gewähltem Chiffretext verhindert.

4. Einhaltung der PKCS: Die Einhaltung dieses Standards wehrt eine Vielzahl mathematischer Angriffe auf RSA und auf andere Public-Key-Algorithmen ab.

[40] Rivest, R. L. und Kaliski, B. „RSA Problem“. url: https://people.csail.mit.edu/rivest/RivestKaliski-RSAProblem.pdf. MIT Laboratory for Computer Science & RSA Laboratories, Dez. 2003. [4]

[41] Smith, R. E. „Internet-Kryptographie“. In: Informationssicherheit. Bonn: Addison-Wesley, 1998. Kap. 9.2.3 - Technische Sicherheitsanforderungen, S. 225. isbn: 3-8273-1344-9. [9]

Kapitel 4

Anwendungsgebiete

RSA ist wohl das am weitesten verbreitete kryptografische Verfahren. Der RSA-Algorithmus wird in vielen Bereichen des Alltags verwendet. Vor allem wird RSA…

- …in digitalen Signaturen verwendet z. B. in Pretty Good Privacy (PGP).
- …in Versionsverwaltungsystemen für die Softwareentwicklung verwendet, z. B. in GNU Privacy Guard (GnuPG) für Git und Mercurial.[42] [43] [44] [45]
- …im Electronic Banking verwendet. Das deutsche Banking-System „Homebanking Computer Interface (HBCI/FinTS)“ nutzt das RSA-Verfahren.[46]
- …für die Verschlüsselung von Verbindungen zwischen Server und Browser (S-HTTP) mit X.509-Zertifikaten und dem Übertragungsprotokoll Secure Sockets Layer (SSL) bzw. Transport Layer Security (TLS) verwendet.[47]
- …für eine sichere E-Mail-Verschlüsselung und E-Mail-Signierung verwendet. Einer der bekanntesten Standards zur Absicherung von E-Mails sind OpenPGP, Privacy Enhanced Mail (PEM) und S/MIME.
- …für andere Übertragungsprotokolle verwendet wie zum Beispiel in IPsec, und Secure Shell (SSH).

[42] 156 Mitwirkende am Open Source Project „git-scm.com“, siehe Liste der Mitwirkenden unter https://github.com/git/git-scm.com/graphs/contributors. 7.4 Git Tools - Signing Your Work. Abgerufen am: 29. Oktober 2017. url: https://git-scm.com/book/en/v2/Git-Tools-Signing-Your-Work. [17]

[43] GnuPG-Team. The GNU Privacy Guard. Abgerufen am: 29. Oktober 2017. Okt. 2017. url: https://gnupg.org/. [38]

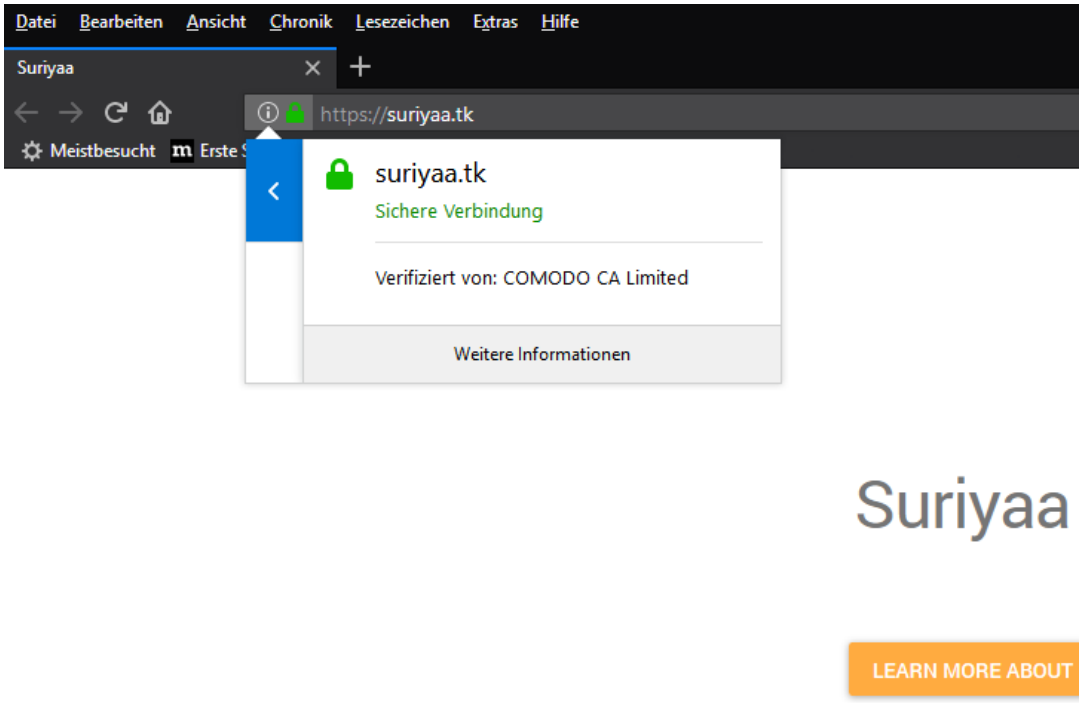
[44] GitHub, Inc. About GPG. Abgerufen am: 29. Oktober 2017. url: https://help.github.com/articles/about-gpg/. [35]

[45] Lopez, B. (GitHub, Inc.). GPG signature verification. Abgerufen am: 29. Oktober 2017. Apr. 2016. url: https://github.com/blog/2144-gpg-signature-verification. [45]

[46] Deutsche Kreditwirtschaft (DK). Die Deutsche Kreditwirtschaft HBCI FinTS Spezifikation. Seite 9. Abgerufen am: 29. Oktober 2017. url: http://www.hbci-zka.de/dokumente/spezifikation_deutsch/fintsv3/FinTS_3.0_Security_Sicherheitsverfahren_HBCI_Rel_20130718_final_version.pdf. [28]

[47] Digicert, Inc. What is an SSL certificate? Abgerufen am: 24. Dezember 2017. url: https://www.digicert.com/ssl/. [31]

Die wichtigste und am bedeutendste Anwendung von RSA ist immer noch SSL/TLS: Der Aufruf von SSL/TLS erfolgt über die Protokollangabe `https://` in der Adresszeile des Browsers. Die SSL/TLS-Verschlüsselung wird üblicherweise durch ein geschlossenes Vorhängeschloss - wie in Abbildung 4.1 - symbolisiert.



Quelle: Suriyaa Sundararuban (erstellt am 29. Oktober 2017 um 16:53:22)

Abbildung 4.1: Eine mit SSL/TLS abgesicherte Webseite in Mozilla Firefox v57.0b12 (Developer Edition).

4.1 Der RSA-Algorithmus als asymmetrisches Verschlüsselungsverfahren

RSA wird als asymmetrisches Verschlüsselungsverfahren für den E-Mail-Versand oder bei der Erzeugung des privaten Schlüssels für einen Webserver verwendet. Unter asymmetrisches Verschlüsselungsverfahren versteht mein ein kryptographisches Verfahren wie dem RSA, bei dem man keinen gemeinsamen Schlüssel benötigt. Jeder Nutzer erzeugt somit sein eigenes RSA-Schlüsselpaar.
Dabei muss beim korrekten Entschlüsseln folgendes beachten: Das Verschlüsselungsverfahren E wird als Verschlüsseln interpretiert und Entschlüsselungsverfahren D als Entschlüsseln interpretiert. Somit garantiert der Satz von Euler, dass die Nachricht korrekt entschlüsselt wurde[48]:

Verschlüsselung: $f_E(M) := M^E \bmod n$

[48] Beutelspacher, A., Schwenk, J. und Wolfenstetter, K.-D. „Von RSA zu Zero-Knowledge“. In: Moderne Verfahren der Kryptographie. 7. Auflage. Wiesbaden: Vieweg, 2010. Kap. 2.8 - Der RSA-Algorithmus, S. 19–22. isbn: 978-3-8348-1228-5. [10]

Entschlüsselung: $f_D(C) := C^D \bmod n$

Korrektheit: $f_D(f_E(M)) = (M^E)^D \bmod n = m$

Public-Key-Eigenschaft: Wenn man nur die Zahl des RSA-Moduls n kennt (und nicht etwa die Faktoren p und q oder die Zahl der Eulerschen $\phi(n)$-Funktion $\phi(n)$, so kann man aus dem öffentlichen Schlüssel nicht den privaten Schlüssel berechnen.

4.2 Digitale Signaturen mit dem RSA-Algorithmus

Unter einer digitalen Signatur darf man sich keine Unterschrift von Hand vorstellen, die eingescannt und digital gespeichert wird. Eine solche könnte schließlich leicht kopiert und missbraucht werden. Vielmehr versteht man darunter eine spezielle, schlüsselabhängige Prüfsumme, die im Zusammenhang mit einem digitalen Dokument ähnliche Eigenschaften aufweist wie eine Unterschrift von Hand.[49] Damit die Signatur gerechtfertigt ist, muss diese Prüfsumme folgende Voraussetzungen erfüllen:

- Sie darf nicht gefälscht werden.
- Ihre Echtzeit muss überprüfbar sein.
- Sie darf nicht unbemerkt von einem Dokument zum anderen übertragen werden können.
- Das dazugehörige Dokument darf nicht unbemerkt verändert werden können.

4.2.1 Funktionsweise

Der RSA-Algorithmus lässt sich auch für digitale Signaturen anwenden, da RSA-Signaturen auf der RSA-Verschlüsselung basieren.[50] Für digitale Signaturen mit dem RSA-Verfahren müssen Alice und Bob die Verwendung der Schlüssel vertauschen.[51] Das heißt, wenn Alice eine Nachricht signieren will, so entschlüsselt sie diese mit ihrem geheimen Schlüssel S, obwohl die Nachricht unverschlüsselt ist. Der resultierende Text $D(M)$ ist die digitale Signatur. Bob verifiziert diese, indem er $D(M)$ mit den öffentlichen Schlüssel P verschlüsselt. Erhält er so die ursprüngliche Nachricht M zurück, dann ist die digitale Signatur echt.

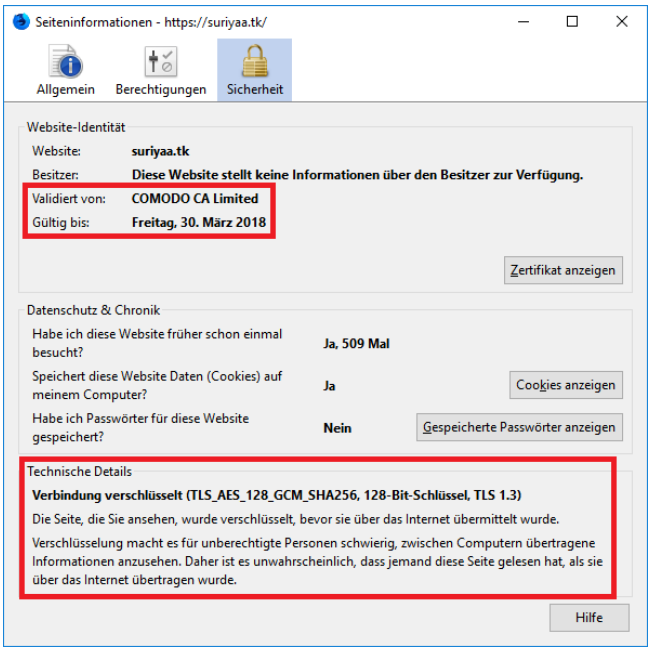
[49] Schmeh, K. Kryptografie. 9. Edition. Heidelberg: dpunkt-Verlag, 2007. isbn: 978-3-89864-435-8. [8]

[50] ebd.

[51] Rivest, R. L., Shamir, A. und Adleman, L. A Method for Obtaining Digital Signatures and Public-Key Cryptosystems. Communications of the ACM (Association for Computing Machinery) & Laboratory for Computer Science (Massachusetts Institute of Technology). url: https://people.csail.mit.edu/rivest/Rsapaper.pdf. Feb. 1978. [1]

4.3 SSL/TLS & Co.

Das Kürzel „SSL/TLS“ steht für das bislang erfolgreichste Sicherheitskonzept im Internet. Jede Webanwendung und fast jede Webseite bietet heute jeden Nutzern die Möglichkeit, vertrauliche Daten über eine SSL/TLS-geschützte Verbindung an den Server zu übertragen (siehe Abbildung 4.2).[52] Allein dadurch wird SSL/TLS zu mit Abstand am häufigsten eingesetzten kryptographischen Protokoll überhaupt. Darüber hinaus werden aber noch andere Webprotokolle wie FTP, IMAP oder POP3 über SSL/TLS mit RSA abgesichert, aber auch in andere Netzwerkschichten verwendet wie in WLAN-Netzwerken und in IPSec.[53]



Quelle: Suriyaa Sundararuban (erstellt am 29. Oktober 2017 um 17:26:43)

Abbildung 4.2: Das abgebildete Fenster erhält man, wenn man in Abbildung 4.1 auf „Weitere Informationen“ klickt.

Es gibt keine großen Unterschiede zwischen dem SSL-Protokoll und TLS. Das SSL-Protokoll wurde von Netscape entwickelt, um Informationen sicher zu versenden und die Serveridentität zu verifizieren. Es wird von allen Webbrowsern unterstützt. Die Internet Engineering Task Force (IETF) hat das TLS-Protokoll als den Nachfolger von Secure Sockets Layer (SSL) festgelegt. Transport Layer Security (TLS) verwendet stärkere Verschlüsselungsalgorithmen als SSL und wird hauptsächlich in E-Mail-Programmen und in Browser-Server-Übertragungen verwendet. Beide beinhalten neben anderen Krypto-Algorithmen auch den RSA-Algorithmus.[54]

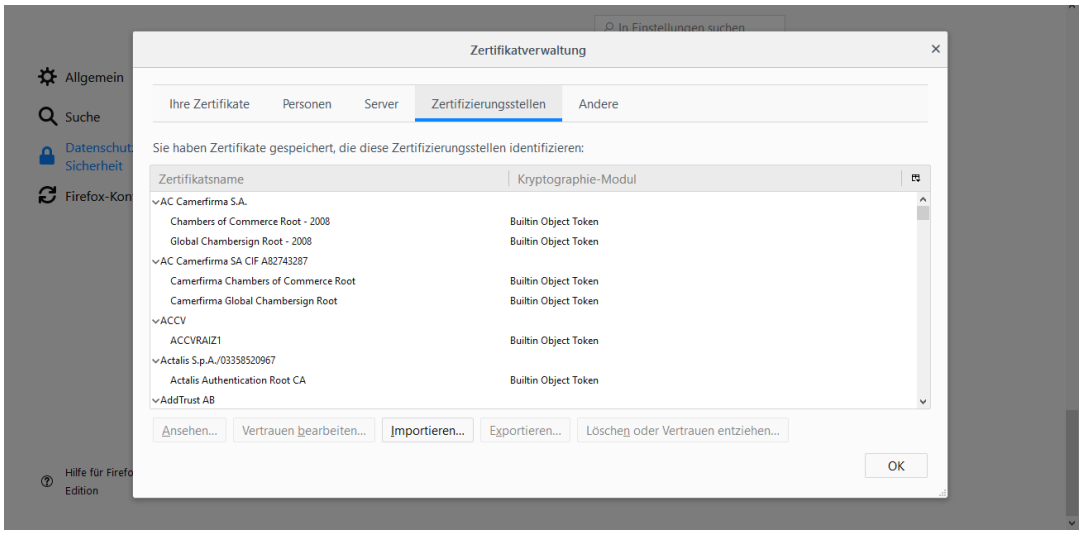
[52] Schwenk, J. In: Sicherheit und Kryptographie im Internet - Theorie und Praxis. 4. Auflage. Wiesbaden: Springer Verlag, 2014. Kap. 1.8 - Zertifikate, S. 30–35. isbn: 978-3-658-06543-0. [13]

[53] Schwenk, J. In: Sicherheit und Kryptographie im Internet - Theorie und Praxis. 4. Auflage. Wiesbaden: Springer Verlag, 2014. Kap. 7 - WWW-Sicherheit mit SSL, S. 145–172. isbn: 978-3-658-06543-0. [14]

[54] Indiana University Knowledge Base. What is the difference between SSL and TLS? Abgerufen am 28. Dezember 2017. url: https://kb.iu.edu/d/anjv. [42]

4.3.1 Public-Key-Infrastrukturen (PKI)

RSA und andere asymmetrisches Verfahren haben die Kryptografie revolutioniert. Sie sind jedoch noch keine Garantie für ein fehlerfreies und sichere Kommunikationsmöglichkeit für das Internet. Dies liegt daran, dass es beim praktischen Einsatz von RSA und Co. zahlreiche Probleme entstehen.
Die bayerische Landesamt für Digitalisierung, Breitband und Vermessung - IT-DLZ - Bayerische Verwaltungs-PKI erklärt das Problem so: „Mit Hilfe [von RSA kann man] Nachrichten im Internet signier[en] und verschlüssel[en]. Das Signieren garantiert, dass die Nachricht in dieser Form wirklich vom angegebenen Absender stammt. Allerdings benötigt man hierzu den Public-Key des Absenders. Dieser könnte z. B. per E-Mail versendet werden. Es stellt sich genau an diesem Punkt aber die Frage, wie sichergestellt werden kann, dass es sich tatsächlich um den Schlüssel des Absenders handelt und nicht um die Fälschung eines Betrügers. Hierzu kann der zu verschickende Schlüssel selbst wieder mit einem vertrauenswürdigen Schlüssel signiert sein. Auf diese Weise lässt sich eine Hierarchie aus vertrauenswürdigen Institutionen aufbauen. Auf die Echtheit der Schlüssel der obersten Institutionen dieser Hierarchie muss man sich aber verlassen können.“ [55]



Quelle: Suriyaa Sundararuban (erstellt am 28. Dezember 2017 um 17:04:52)

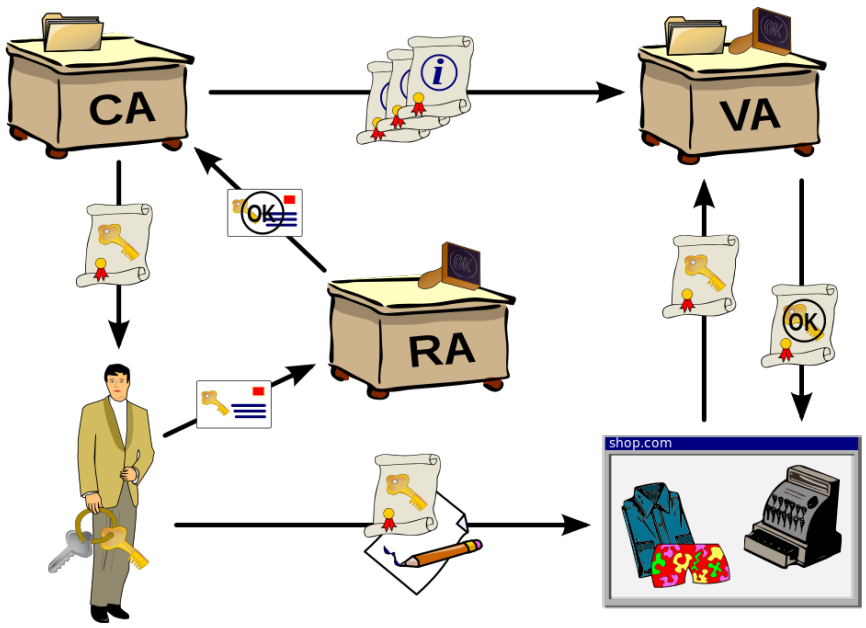
Abbildung 4.3: Zertifikatsverwaltung der vertrauenswürdigen PKIs in Mozilla Firefox 58.0b12 (Developer Edition).

Nur durch den Aufbau einer geeigneten Infrastruktur kann dieses Problem gelöst werden. Eine solche Infrastruktur wird als Public-Key-Infrastruktur (PKI) bezeichnet. Dabei stellt ein PKI-System digitale Zertifikate aus.[56] Außerdem verteilt es die

[55] Landesamt für Digitalisierung, Breitband und Vermessung - IT-DLZ - Bayerische Verwaltungs-PKI. Was ist PKI? Abgerufen am 04. Dezember 2017. url: https://www.pki.bayern.de/vpki/allg/pki/index.html. [43]

[56] DoD Public Key Infrastructure Program Management Office (U.S. Defense Information Systems Agency). X.509 Certificate Policy for the United States Department of Defense. Abgerufen am 04. Dezember 2017. Feb. 2005. url: http://jitc.fhu.disa.mil/projects/pki/documents/dod_x509_certificate_policy_v9_0_9_february_2005.pdf. [32]

Zertifikate und prüft sie auf Korrektheit und Vertrauenswürdigkeit. Eine Anzahl von verschiedenen PKIs sind in Webbrowsern vorinstalliert (siehe Abbildung 4.3).
Die innerhalb einer PKI ausgestellten Zertifikate werden zur Absicherung rechnergestützter Kommunikation verwendet.[57] Dabei wird häufig auf RSA zurückgegriffen.[58]
Der Aufbau von Public-Key-Infrastrukturen gilt in vielen Fälle als Voraussetzung für den sinnvollen Einsatz asymmetrischer Kryptografie wie dem RSA, insbesondere innerhalb größerer Nutzergruppen.[59]



Quelle: Wikimedia Commons [60]

Abbildung 4.4: Aufbau einer Public-Key-Infrastruktur (PKI)

Die Abbildung 4.4 zeigt den vereinfachten Aufbau einer Public-Key-Infrastruktur (PKI). Es wird gezeigt, wie sich ein Nutzer mit seinen RSA-Schlüsselpaar gegenüber einer Registrierungsstelle (RA) authentifiziert. Diese bestätigt den öffentlichen Schlüssel des Teilnehmers gegenüber der Zertifizierungsstelle (CA). Die Zertifizierungsstelle (CA) stellt zum Beispiel dem Teilnehmer ein X.509-Zertifikat aus. Die Zertifizierungsstelle (CA) übergibt alle gesammelten und überprüften Zertifikate dem Validierungsdienst (VA). Der Teilnehmer kann sich damit gegenüber einer anderen

[57] Landesamt für Digitalisierung, Breitband und Vermessung - IT-DLZ - Bayerische Verwaltungs-PKI. Was ist PKI? Abgerufen am 04. Dezember 2017. url: https://www.pki.bayern.de/vpki/allg/pki/index.html. [43]

[58] Xenitellis, S. „The Open-source PKI Book“. In: Sourceforge. Kap. 4.2.1 - Generate the RSA key–pair for the CA. url: https://sourceforge.net/projects/ospkibook/. Abgerufen am 08.11.2017. S. 20. [16]

[59] Schmeh, K. „Kryptografie“. In: Informationssicherheit. Heidelberg: dpunkt.verlag, 2007. Kap. 24 - Public-Key-Infrastrukturen (Teil 4 - Public-Key-Infrastrukturen), S. 440–470. isbn: 978-3-89864-435-8. [8]

[60] Chris (Wikimedia Commons). *Public-Key-Infrastructure.svg*. Abgerufen am: 04. Dezember 2017. url: https://commons.wikimedia.org/wiki/File:Public-Key-Infrastructure.svg. [24]

Person mit dem Zertifikat zum Beispiel mit einer Website authentifizieren. Die andere Person kann das Zertifikat von einem Validierungsdienst (VA) überprüfen lassen.

4.3.2 Public-Key Cryptography Standards (PKCS)[61]

„Public-Key Cryptography Standards (PKCS)“ ist eine Sammlung von Spezifikationen für asymmetrische Kryptosysteme, darunter auch RSA. Diese Standards wurden von IT-Unternehmen RSA Security, Inc. ab 1991 entwickelt, um die Verbreitung asymmetrischer Kryptosysteme zu beschleunigen.[62] Davon gibt es 15 PKCS-Varianten. Einige der Spezifikationen sind nun genormte Standards. Uns interessiert nur das erste PKCS. „PKCS #1 (Version 2.2)“[63] [64] definiert das Format der RSA-Verschlüsselung.[65]

4.3.3 Der Zertifikatstandard X.509[66] & Gültigkeit von Zertifikaten

Einen signierten Datensatz mit diesen Eigenschaften wie es in Abbildung 4.5 gezeigt wird, wird digitales Zertifikat (Public-Key-Zertifikat) genannt. Die digitalen Zertifikate wurden standardisiert, damit eine PKI einheitlich arbeitet. Das Maß aller Dinge in Sachen digitaler Zertifikate gibt es in mehreren Versionen und zahlreichen Profilen. X.509 ist einer der ältesten Krypto-Standards. Die erste Version von X.509 erschien 1988. Zertifikate, die dem darin beschriebenen Format entsprechen, werden X.509v1-Zertifikate genannt. 1993 folgte X.509v2. Die aktuelle Version ist X.509v3. [67] [68] [69]

Bis März 2017 durfte man die SSL/TLS-Zertifikate beziehungsweise Public-Key-Zertifikate länger als zwei Jahre festlegen. Das änderte sich im Nachhinein, als das CA/Browser Forum die maximale Gültigkeit der SSL/TLS-Zertifikate änderte. Zur neu festgelegten Gültigkeit von Zertifikaten nach der Entscheidung schrieb GlobalSign, eine weltweit führende Zertifizierungsstelle (CA), folgenden Kommentar: „Das

[61] RSA Laboratories. Public-Key Cryptography Standards (PKCS). Abgerufen am 28. Dezember 2017. url: https://web.archive.org/web/20160414135330/http://www.emc.com/emc-plus/rsa-labs/standards-initiatives/public-key-cryptography-standards.htm. [56]

[62] RSA Laboratories. What is PKCS? Abgerufen am 28. Dezember 2017. url: https://web.archive.org/web/20170510092137/https://www.emc.com/emc-plus/rsa-labs/standards-initiatives/pkcs.htm. [57]

[63] RSA Laboratories. PKCS #1 v2.2: RSA Cryptography Standard. Abgerufen am 28. Dezember 2017. url: https://web.archive.org/web/20160320063514/http://www.emc.com/emc-plus/rsa-labs/pkcs/files/h11300-wp-pkcs-1v2-2-rsa-cryptography-standard.pdf. [55]

[64] RSA Laboratories. PKCS #1: RSA Cryptography Standard. Abgerufen am 28. Dezember 2017. url: https://web.archive.org/web/20160422081449/https://www.emc.com/emc-plus/rsa-labs/standards-initiatives/pkcs-rsa-cryptography-standard.htm. [54]

[65] Network Working Group (IETF). PKCS #1: RSA Cryptography Specifications Version 2.2. Abgerufen am 28. Dezember 2017. url: https://tools.ietf.org/html/rfc8017. [50]

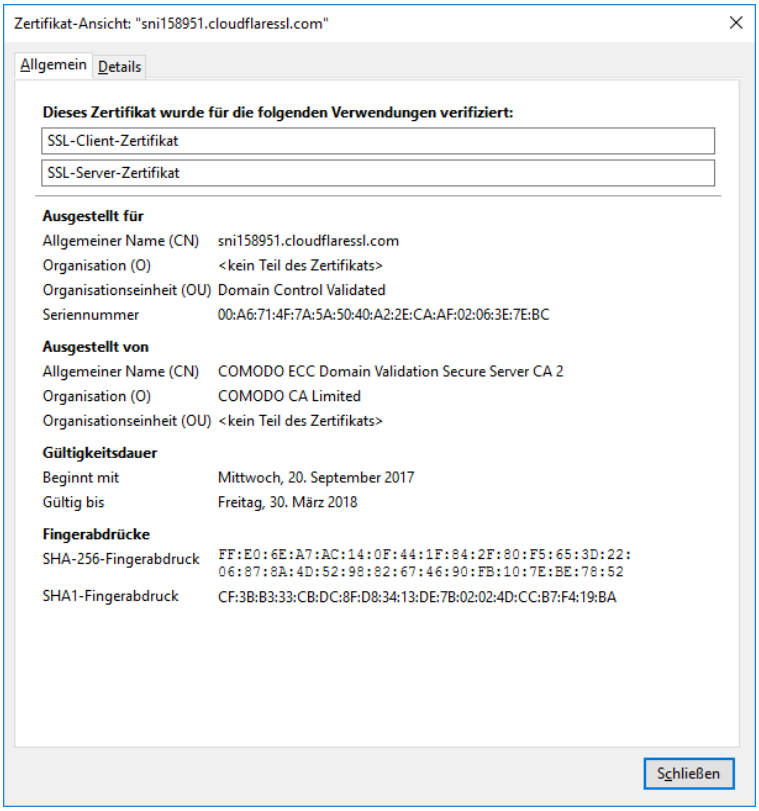
[66] Python Cryptographic Authority. X.509. Abgerufen am 04. Dezember 2017. url: https://cryptography.io/en/latest/x509/. [52]

[67] Network Working Group (IETF). Internet X.509 Public Key Infrastructure Certificate and Certificate Revocation List (CRL) Profile. Abgerufen am 04. Dezember 2017. url: https://tools.ietf.org/html/rfc5280.html. [47]

[68] Network Working Group (IETF). Internet X.509 Public Key Infrastructure - Certificate and CRL Profile. Abgerufen am 04. Dezember 2017. url: https://tools.ietf.org/html/rfc2459.html. [49]

[69] Network Working Group (IETF). Internet X.509 Public Key Infrastructure - Certificate and Certificate Revocation List (CRL) Profile. Abgerufen am 04. Dezember 2017. url: https://tools.ietf.org/html/rfc3280.html. [48]

CA/Browser Forum, ein Branchengremium bestehend aus Zertifizierungsstellen (CAs), Webbrowsern und Betriebssystemen, hat kürzlich in der Wahl 193 entschieden, [70] [71] [72] dass die maximale Gültigkeit von SSL/TLS Zertifikaten auf zwei Jahre (825 Tage, wenn man es genau wissen will) reduziert wird.“[73] Nur selbst-generierte Zertifikate und kostenlos-angebotene Zertifikate wie zum Beispiel die SSL/TLS Zertifikate von Cloudflare[74] (siehe Abbildung 4.5) dürfen zum Beispiel für kleine Webseiten maximal drei Jahre gelten.



Quelle: Suriyaa Sundararuban (erstellt am 30. Oktober 2017 um 18:27:10)

Abbildung 4.5: Dieses X.509-Zertifikat für die Domain `suriyaa.tk` wird von Cloudflare, Inc. und von COMODO CA Ltd. bereitgestellt (Das abgebildete Fenster erhält man, wenn man in Abbildung 4.2 auf „Zertifikat anzeigen“ klickt.)

[70] CA/Browser Forum. Ballot 193 – 825-day Certificate Lifetimes. Abgerufen am 23. Dezember 2017. url: https://cabforum.org/2017/03/17/ballot-193-825-day-certificate-lifetimes/. [22]

[71] CA/Browser Forum. Baseline Requirements for the Issuance and Management of Publicly-Trusted Certificates (v.1.2.3). Abgerufen am 23. Dezember 2017. url: https://cabforum.org/wp-content/uploads/BRv1.2.3.pdf. [23]

[72] CA Security Council, Inc. SSL Certificate Validity Periods Limited to 39 Months Starting in April. Abgerufen am 23. Dezember 2017. url: https://casecurity.org/2015/02/19/ssl-certificate-validity-periods-limited-to-39-months-starting-in-april/. [21]

[73] GlobalSign, Inc. Gültigkeit von SSL/TLS Zertifikaten auf maximal zwei Jahre beschränkt. Abgerufen am 23. Dezember 2017. url: https://www.globalsign.com/de-de/blog/gueltigkeit-von-ssl-zertificaten-auf-maximal-zwei-jahre-beschraenkt/. [36]

[74] Cloudflare, Inc. Cloudflare One-Click SSL. Abgerufen am 23. Dezember 2017. url: https://www.cloudflare.com/ssl/. [26]

Fazit

Die Auseinandersetzung mit den Themengebieten der Funktionsweise, der Sicherheit, der Attacken und der praktischen Anwendungsgebiete vom RSA-Algorithmus schaffte mir einen ersten, tiefen Einblick in die mathematische und technische Welt der Kryptographie.
RSA ist für mich nicht einfach nur ein einfacher kryptographischer Algorithmus, der beliebig verwendet werden kann und dann dauerhafte Sicherheit bietet. Das Verfahren muss sehr sorgfältig eingesetzt werden, damit kein Angreifer die Schwachstellen mit den angesprochenen Attacken ausnutzen kann. Eine RSA-Implementierung muss den veröffentlichten Empfehlungen der zuvor erwähnten PKCS folgen.
Das Erstellen dieser Arbeit erwies sich zu Beginn als sehr herausfordernd und aufwendig, zumal die Definitionen der IT-Begriffe und die mathematischen Veranschaulichung der RSA-Algorithmus, aber auch die stark übergreifende Verbindung von Kryptographie, Informatik und Mathematik. Auch das Arbeiten mit LaTeX war zunächst ein große Herausforderung, da es ein, mir zu der Zeit noch, unbekanntes Programm war und der Umgang mit LaTeX erst erlernt werden musste.
Insgesamt aber, lässt sich sagen, dass die Anfertigung dieser Seminararbeit, Kenntnisse in Bezug auf das mathematische Verständnis über den RSA-Algorithmus, das Lösen der Sicherheitsprobleme und Attacken gegenüber RSA, das Wissen über technische Zusammenhänge, das Erlernen des Programmierens mit dem Textsatzsystemsoftware LaTeX , dem Literaturangaben-Erstellungssoftware „BibTeX“ und die Fertigung einer fachlichen - mathematisch und technisch orientierten Arbeit, erbrachte.
Demzufolge war dies eine gute Vorbereitung für das spätere Studium und ein Anreiz zur näheren Beschäftigung mit den Formalien der Mathematik und der Kryptographie.

Literaturverzeichnis

[1] Rivest, R. L., Shamir, A. und Adleman, L. A Method for Obtaining Digital Signatures and Public-Key Cryptosystems. Communications of the ACM (Association for Computing Machinery) & Laboratory for Computer Science (Massachusetts Institute of Technology). url: https://people.csail.mit.edu/rivest/Rsapaper.pdf. Feb. 1978.

Wissenschaftliche Publikationen

[2] Heitkötter, H. „Public-Key-Verfahren: RSA/Rabin“. url: https://www.wi1.uni-muenster.de/pi/lehre/ws0708/seminar/Abgaben/RSA_Rabin.pdf. Westfälische Wilhelms-Universität Münster.

[3] Litzel, A. „Das RSA-Verfahren“. url: https://www7.in.tum.de/um/courses/seminar/krypto/SS09/litzel/zusammenfassung.pdf. Fakultät „Informatik“ an der Technischen Universität München (TUM), 2009.

[4] Rivest, R. L. und Kaliski, B. „RSA Problem“. url: https://people.csail.mit.edu/rivest/RivestKaliski-RSAProblem.pdf. MIT Laboratory for Computer Science & RSA Laboratories, Dez. 2003.

[5] Rivest, R. L. und Silverman, R. D. „Are 'Strong' Primes Needed for RSA“. urls: http://eprint.iacr.org/2001/007 und https://people.csail.mit.edu/rivest/RivestSilverman-AreStrongPrimesNeededForRSA.pdf. IACR Cryptology ePrint Archive, Report paper 2001/007 (Version 1998-12-01 Submitted January 30, 2001) & MIT, 30. Jan. 2001.

[6] Wan Han, D. „Generating strong prime numbers for RSA using probabilistic Rabin-Miller algorithm“. Abgerufen am 17.11.2017. Electrical und Computer Engineering Department (George Mason University). url: http://ece.gmu.edu/coursewebpages/ECE/ECE646/F09/project/reports_1999/dong_report.pdf.

Bücher

[7] Salomaa, A. Public-Key Cryptography. Berlin/Heidelberg: Springer-Verlag, 1990. Kap. 4, S. 125–157. isbn: 3-540-52831-8.

[8] Schmeh, K. Kryptografie. 9. Edition. Heidelberg: dpunkt-Verlag, 2007. isbn: 978-3-89864-435-8.

[9] Smith, R. E. Internet-Kryptographie. Informationssicherheit. Bonn: Addison-Wesley, 1998. isbn: 3-8273-1344-9.

Verwendete Bücherkapiteln

[10] Beutelspacher, A., Schwenk, J. und Wolfenstetter, K.-D. „Von RSA zu Zero-Knowledge“. In: Moderne Verfahren der Kryptographie. 7. Auflage. Wiesbaden: Vieweg, 2010. Kap. 2.8 - Der RSA-Algorithmus, S. 19–22. isbn: 978-3-8348-1228-5.

[11] Kocher, P. C. „Timing Attacks on Implementations of Diffie-Hellman, RSA, DSS, Other Systems“. In: CRYPTO 1996, Band 1109 in LNCS. url: https://link.springer.com/chapter/10.1007/3-540-68697-5_9. Abgerufen am 08.11.2017. Springer, 1996, S. 104–113. isbn: 978-3-540-61512-5.

[12] Schmeh, K. „Kryptografie“. In: Informationssicherheit. Heidelberg: dpunkt.verlag, 2007. Kap. 24 - Public-Key-Infrastrukturen (Teil 4 - Public-Key-Infrastrukturen), S. 440–470. isbn: 978-3-89864-435-8.

[13] Schwenk, J. In: Sicherheit und Kryptographie im Internet - Theorie und Praxis. 4. Auflage. Wiesbaden: Springer Verlag, 2014. Kap. 1.8 - Zertifikate, S. 30–35. isbn: 978-3-658-06543-0.

[14] Schwenk, J. In: Sicherheit und Kryptographie im Internet - Theorie und Praxis. 4. Auflage. Wiesbaden: Springer Verlag, 2014. Kap. 7 - WWW-Sicherheit mit SSL, S. 145–172. isbn: 978-3-658-06543-0.

[15] Smith, R. E. „Internet-Kryptographie“. In: Informationssicherheit. Bonn: Addison-Wesley, 1998. Kap. 9.2.3 - Technische Sicherheitsanforderungen, S. 225. isbn: 3-8273-1344-9.

[16] Xenitellis, S. „The Open-source PKI Book“. In: Sourceforge. Kap. 4.2.1 - Generate the RSA key–pair for the CA. url: https://sourceforge.net/projects/ospkibook/. Abgerufen am 08.11.2017. S. 20.

World Wide Web (WWW)

[17] 156 Mitwirkende am Open Source Project „git-scm.com“, siehe Liste der Mitwirkenden unter https://github.com/git/git-scm.com/graphs/contributors. 7.4 Git Tools - Signing Your Work. Abgerufen am: 29. Oktober 2017. url: https://git-scm.com/book/en/v2/Git-Tools-Signing-Your-Work.

[18] Alrasheed, A. und Fatima (The University of Tennessee at Chattanooga). RSA Attacks. Abgerufen am 10. November 2017. url: https://www.utc.edu/center-information-security-assurance/pdfs/course-paper-5600-rsa.pdf.

[19] Boneh, D. (Computer science at Stanford University). Twenty Years of Attacks on the RSA Cryptosystem. Abgerufen am 10. Dezember 2017. url: https://www.ams.org/notices/199902/boneh.pdf.

[20] Busse, M., Schmitt, M. und Steeg, J. Der RSA-Algorithmus. Abgerufen am: 30. September 2017. url: http://www.zum.de/Faecher/Inf/RP/infschul/kr_rsa.html.

[21] CA Security Council, Inc. SSL Certificate Validity Periods Limited to 39 Months Starting in April. Abgerufen am 23. Dezember 2017. url: https://casecurity.org/2015/02/19/ssl-certificate-validity-periods-limited-to-39-months-starting-in-april/.

[22] CA/Browser Forum. Ballot 193 – 825-day Certificate Lifetimes. Abgerufen am 23. Dezember 2017. url: https://cabforum.org/2017/03/17/ballot-193-825-day-certificate-lifetimes/.

[23] CA/Browser Forum. Baseline Requirements for the Issuance and Management of Publicly-Trusted Certificates (v.1.2.3). Abgerufen am 23. Dezember 2017. url: https://cabforum.org/wp-content/uploads/BRv1.2.3.pdf.

[24] Chris (Wikimedia Commons). *Public-Key-Infrastructure.svg*. Abgerufen am: 04. Dezember 2017. url: https://commons.wikimedia.org/wiki/File:Public-Key-Infrastructure.svg.

[25] Clayton, R. (The Computer Laboratory at University of Cambridge). Brute force attacks on cryptographic keys - RSA. Abgerufen am 07. Dezember 2017. url: https://www.cl.cam.ac.uk/~rnc1/brute.html#RSAsection.

[26] Cloudflare, Inc. Cloudflare One-Click SSL. Abgerufen am 23. Dezember 2017. url: https://www.cloudflare.com/ssl/.

[27] CS110-Team der Wellesley-College. *aliceBob.jpg*. Abgerufen am: 28. Oktober 2017. url: http://cs110.wellesley.edu/reading/cryptography-files/aliceBob.jpg.

[28] Deutsche Kreditwirtschaft (DK). Die Deutsche Kreditwirtschaft HBCI FinTS Spezifikation. Seite 9. Abgerufen am: 29. Oktober 2017. url: http://www.hbci-zka.de/dokumente/spezifikation_deutsch/fintsv3/FinTS_3.0_Security_Sicherheitsverfahren_HBCI_Rel_20130718_final_version.pdf.

[29] Die Serlo-Community. Teilerfremd. Abgerufen am 17. November 2017. url: https://de.serlo.org/mathe/zahlen-groessen/teiler-primzahlen/teiler-vielfache/teilerfremd.

[30] Die Wikipedia-Community. Diskrete Exponentialfunktion. Abgerufen am 07. November 2017. url: https://de.wikipedia.org/wiki/Diskrete_Exponentialfunktion.

[31] Digicert, Inc. What is an SSL certificate? Abgerufen am: 24. Dezember 2017. url: https://www.digicert.com/ssl/.

[32] DoD Public Key Infrastructure Program Management Office (U.S. Defense Information Systems Agency). X.509 Certificate Policy for the United States Department of Defense. Abgerufen am 04. Dezember 2017. Feb. 2005. url: http://jitc.fhu.disa.mil/projects/pki/documents/dod_x509_certificate_policy_v9_0_9_february_2005.pdf.

[33] fgrieu (Stack Overflow). Answer for 'Is RSA vulnerable to possible PRNG + Miller Rabin test weaknesses?' Abgerufen am 01. Dezember 2017. url: https://crypto.stackexchange.com/a/52729/25803.

[34] Fiskerstrand, K. sks-keyservers.net. Abgerufen am 11. Dezember 2017. url: https://sks-keyservers.net/.

[35] GitHub, Inc. About GPG. Abgerufen am: 29. Oktober 2017. url: https://help.github.com/articles/about-gpg/.

[36] GlobalSign, Inc. Gültigkeit von SSL/TLS Zertifikaten auf maximal zwei Jahre beschränkt. Abgerufen am 23. Dezember 2017. url: https://www.globalsign.com/de-de/blog/gueltigkeit-von-ssl-zertificaten-auf-maximal-zwei-jahre-beschraenkt/.

[37] GlobalSign, Inc. Schutz privater Schlüssel – So halten Sie Ihre Schlüssel geheim. Abgerufen am 11. Dezember 2017. url: https://www.globalsign.com/de-de/blog/schutz-des-private-keys/.

[38] GnuPG-Team. The GNU Privacy Guard. Abgerufen am: 29. Oktober 2017. Okt. 2017. url: https://gnupg.org/.

[39] Greenemeier, L. Can't Touch This–New Encryption Scheme Targets Transaction Tampering. Abgerufen am 08. November 2017. Mai 2015. url: https://www.scientificamerican.com/article/can-t-touch-this-new-encryption-scheme-targets-transaction-tampering/.

[40] Guggenberger, W. (Humboldt-Gymnasium Vaterstetten). Das RSA-Verfahren. Abgerufen am 10. November 2017. url: http://www.humboldt-gym.de/fileadmin/faecher/mathematik/krypto/rsatxt.html.

[41] Hellman, M., Diffie, B. und Merkle, R. Cryptographic apparatus and method. US Patent 4,200,770. Abgerufen am: 29. September 2017. Apr. 1980. url: https://www.google.com/patents/US4200770.

[42] Indiana University Knowledge Base. What is the difference between SSL and TLS? Abgerufen am 28. Dezember 2017. url: https://kb.iu.edu/d/anjv.

[43] Landesamt für Digitalisierung, Breitband und Vermessung - IT-DLZ - Bayerische Verwaltungs-PKI. Was ist PKI? Abgerufen am 04. Dezember 2017. url: https://www.pki.bayern.de/vpki/allg/pki/index.html.

[44] Lopes Gouvêa, C. P. Answer for 'What data is saved in RSA private key?' Abgerufen am 11. Dezember 2017. url: https://crypto.stackexchange.com/a/7964/25803.

[45] Lopez, B. (GitHub, Inc.). GPG signature verification. Abgerufen am: 29. Oktober 2017. Apr. 2016. url: https://github.com/blog/2144-gpg-signature-verification.

[46] Massachusetts Institute of Technology (MIT). *rsa-photo.jpeg*. Abgerufen am: 30. Oktober 2017. url: http://people.csail.mit.edu/rivest/photos/rsa-photo.jpeg.

[47] Network Working Group (IETF). Internet X.509 Public Key Infrastructure Certificate and Certificate Revocation List (CRL) Profile. Abgerufen am 04. Dezember 2017. url: https://tools.ietf.org/html/rfc5280.html.

[48] Network Working Group (IETF). Internet X.509 Public Key Infrastructure - Certificate and Certificate Revocation List (CRL) Profile. Abgerufen am 04. Dezember 2017. url: https://tools.ietf.org/html/rfc3280.html.

[49] Network Working Group (IETF). Internet X.509 Public Key Infrastructure - Certificate and CRL Profile. Abgerufen am 04. Dezember 2017. url: https://tools.ietf.org/html/rfc2459.html.

[50] Network Working Group (IETF). PKCS #1: RSA Cryptography Specifications Version 2.2. Abgerufen am 28. Dezember 2017. url: https://tools.ietf.org/html/rfc8017.

[51] Pommerening, K. Kryptoanalyse des RSA-Verfahrens. Abgerufen am 22. Dezember 2017. Mai 2000. url: https://www.staff.uni-mainz.de/pommeren/Kryptologie/Asymmetrisch/2_RSAanalyse/.

[52] Python Cryptographic Authority. X.509. Abgerufen am 04. Dezember 2017. url: https://cryptography.io/en/latest/x509/.

[53] Rivest, R. L., Shamir, A. und Adleman, L. Cryptographic communications system and method. US Patent 4,405,829. Abgerufen am: 29. September 2017. Sep. 1983. url: https://www.google.com/patents/US4405829.

[54] RSA Laboratories. PKCS #1: RSA Cryptography Standard. Abgerufen am 28. Dezember 2017. url: https://web.archive.org/web/20160422081449/https://www.emc.com/emc-plus/rsa-labs/standards-initiatives/pkcs-rsa-cryptography-standard.htm.

[55] RSA Laboratories. PKCS #1 v2.2: RSA Cryptography Standard. Abgerufen am 28. Dezember 2017. url: https://web.archive.org/web/20160320063514/http://www.emc.com/emc-plus/rsa-labs/pkcs/files/h11300-wp-pkcs-1v2-2-rsa-cryptography-standard.pdf.

[56] RSA Laboratories. Public-Key Cryptography Standards (PKCS). Abgerufen am 28. Dezember 2017. url: https://web.archive.org/web/20160414135330/http://www.emc.com/emc-plus/rsa-labs/standards-initiatives/public-key-cryptography-standards.htm.

[57] RSA Laboratories. What is PKCS? Abgerufen am 28. Dezember 2017. url: https://web.archive.org/web/20170510092137/https://www.emc.com/emc-plus/rsa-labs/standards-initiatives/pkcs.htm.

[58] RSA Security Inc. *customer-image-2-rsa.jpg*. Abgerufen am: 08. Dezember 2017. url: https://www.rsa.com/content/dam/images/11-2016/customer-image-2-rsa.jpg.

[59] RSA Security, Inc. RSA-155 is factored! Abgerufen am 07. Dezember 2017. url: https://web.archive.org/web/20060616162727/http://www.rsasecurity.com/rsalabs/challenges/factoring/rsa155.html.

[60] Shamir, A. „Crypto will not be broken, it will be bypassed“. Abgerufen am: 29. September 2017. url: http://www.azquotes.com/quote/1218393.

[61] Tews, E. *Datei:Adi Shamir at TU Darmstadt (2013).jpg*. Abgerufen am: 13. Oktober 2017. url: https://commons.wikimedia.org/wiki/File:Adi_Shamir_at_TU_Darmstadt_(2013).jpg.

[62] Wiener, M. J. Cryptanalysis of Short RSA Secret Exponents. Abgerufen am 10. Dezember 2017. url: https://www.cits.ruhr-uni-bochum.de/imperia/md/content/may/krypto2ss08/shortsecretexponents.pdf.

Abbildungsverzeichnis

Danksagung

Zunächst möchte ich mich an dieser Stelle bei all denjenigen bedanken, die mich während der Anfertigung dieser Seminararbeit in der 12. und in der 13. Klasse von Juni 2017 bis Januar 2018 unterstützt und motiviert haben.

Ganz besonders gilt dieses Dank meinem Seminiarlehrer Herr Landthaler, der meine Arbeit und somit auch mich und meine Seminargruppe „Seminar Kryptographie (Nr. 08)“ betreut hat. Er gaben mir immer wieder Verbesserungsvorschläge, hilfreiche Hinweise für die Seminararbeit und für die Präsentationen und kontinuierliche Motivation am Thema „Kryptographie“. Er wies außerdem auch auf meine Schwächen hin und konnte als Fachbetreuer immer wieder zeigen, wo noch Erklärungsbedarf zum Thema „Der RSA-Algorithmus“ bestand. Er haben mich dazu gebracht, über meine bisherigen mathematischen Grenzen hinaus zu denken.

Nicht zuletzt gebührt mein Dank an meine Eltern und an meinen Mitschülern in der Seminargruppe, ohne welche dieses ganze Arbeit und Anstrengung schon im Vorhinein niemals zustande gekommen wäre.

Vielen Dank für all diejenigen für die Geduld und Mühen.

Suriyaa Sundararuban

Eidesstattliche Erklärung

Ich, Suriyaa Sundararuban, erkläre hiermit, dass ich die Seminararbeit „Der RSA-Algorithmus“ selbständig verfasst und keine anderen als die angegebenen Quellen und Hilfsmittel benutzt habe:

- Die Stellen der Arbeit, die anderen Werken im Wortlaut oder dem Sinn nach entnommen sind, habe ich durch Quellen und zusätzliche Angaben deutlich kenntlich gemacht.
- Dies gilt auch für Zitate, bildliche Darstellungen und dergleichen.
- Alle Quellen, die aus dem Internet entnommen oder in einer digitalen Form verwendet wurden, sind im Literaturverzeichnis unter „World Wide Web (WWW)“ beigefügt.
- Textstellen ohne Quellangaben und alle Screenshots habe ich selbständig verfasst beziehungsweise selber erstellt.

Unterschrift:

Ort und Datum:
