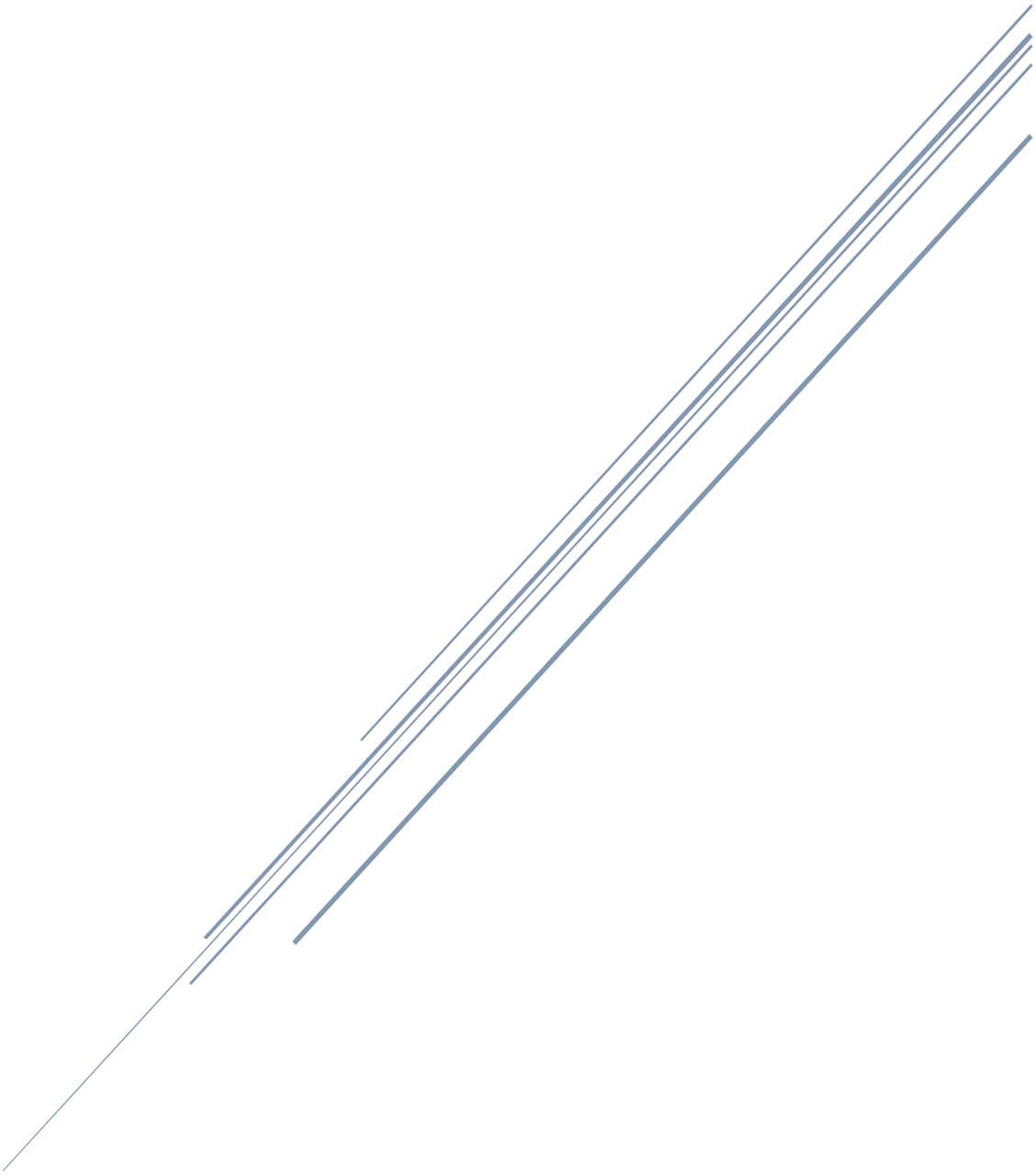
RSA-VERFAHREN

Inwiefern ist die RSA-Verschlüsselung sicher anwendbar oder ist sie lösbar und unsicher?

Von Victor Jaenisch

Betreuer: Frau Preker und Frau Ohrt

Abgabetermin: 25.05.2021



Lessing-Stadtteilschule
Mathematik/Informatik

Zeichen (Arbeit mit Formalitäten): 30.579

Wörter(Arbeit mit Formalitäten): 3957

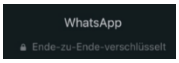
Jahr: 2021

Inhalt

1. Einleitung Gesamtgesellschaftliche Relevanz und persönliche Motivation

Im Büro, in der Schule und in unserer Freizeit benutzen wir Geräte, über die wir auf das Internet zugreifen. Die Verschlüsselung unserer Daten begegnet uns dabei jeden Tag, wenn auch unbewusst. Jeder Mensch versendet Nachrichten über sein Handy, verfasst E-Mails über seinem Computer und benutzt Online-Banking. Dabei ist nur den wenigsten bewusst, dass die Verschlüsslung einem fundamentalen mathematischen Problem unterliegt, die die Sicherheit oder Unsicherheit der Verschlüsslung garantiert.

Die meisten Menschen versenden Nachrichten über Kommunikationsdienste wie WhatsApp oder Telegram, dabei wird bei WhatsApp am Anfang eines Chats der Hinweis auf die Verschlüsselung angegeben. Die Sicherheit der Nachrichten wird vor allem dann wichtig, wenn es um persönliche Informationen geht. Wenn diese Nachrichten unsicher verschlüsselt werden, könnten sie relativ einfach wieder entschlüsselt und im Darknet verkauft werden.



Jede Person, die über das Internet kommuniziert, sollte die Funktionsweise der Verschlüsselung kennen, damit sie sich im Klaren ist, wie sicher es ist.

Ich persönlich bin auf die Kryptografie, ein Teilgebiet der Mathematik, aufmerksam geworden, als ich mich mit NFTs und Kryptowährungen beschäftigt habe. Immer mehr Menschen investieren in Kryptowährungen und dabei wäre es sinnvoll zu wissen, wie eine Verschlüsselung funktioniert.

2. Begründete Wahl der Problemfrage

Ich habe für das Thema meiner Seminararbeit die RSA-Verschlüsslung ausgewählt, da sie eine der ersten asymmetrischen Verschlüsselungsmethoden war. Darüber hinaus wird sie heute immer noch, wenn auch abgeändert, verwendet, da die RSA-Verschlüsselung einem sehr schweren mathematischen Problem unterliegt. Ich werde mich problemorientiert und präzise mit der Leitfrage „**Inwiefern ist die RSA-Verschlüsselung sicher anwendbar oder ist sie lösbar und unsicher?**“ beschäftigen.

Die Entwicklung des RSA-Verfahrens war enorm wichtig, da es die Kryptografie, die wir heute kennen, sonst womöglich nicht geben würde. Ich habe mich entschlossen die Sicherheit der RSA-Verschlüsselung zu überprüfen, da es die Hauptfunktion der Verschlüsselung ist.

In meiner Seminararbeit werde ich mit den Grundlangen anfangen und immer weiter ins Detail gehen, damit die Frage der Sicherheit am Ende der Arbeit evaluiert werden kann.

3. Begriffe bzw. Definitionen und mathematische Grundlagen

Beantwortung der Problemstellung aus Sicht der Fächer

Im folgenden Teil werden die mathematischen Grundlagen, die für das Verständnis des RSA-Verfahrens notwendig sind, skizziert. Dabei werde ich nur auf die essenziellen Bestandteile eingehen, da der Umfang der Seminararbeit sonst zu groß wäre.

3.1 Primzahlen

Definition: Eine Primzahl ist eine natürliche Zahl, die größer als 1 und ausschließlich durch sich selbst und durch 1 teilbar ist.[1]

3.2 Eulersche Phi-Funktion

Bei der eulerschen Phi-Funktion handelt es sich um eine zahlentheoretische Funktion. Sie ordnet jeder natürlichen Zahl n die Anzahl der natürlichen Zahlen a von 1 bis n zu, die zu n teilerfremd sind, für die also $ggT(a,n) = 1$ ist.

$$\varphi(n) := |\{a \in \mathbb{N} \mid 1 \leq a \leq n \wedge ggT(a,n) = 1\}|$$

Eine Primzahl p ist nur durch 1 und sich selbst teilbar, deshalb ist sie zu den Zahlen 1 bis $p-1$ teilerfremd.

Es gilt daher $\varphi(p) = p - 1$.

Des Weiteren ist die Phi-Funktion eine multiplikative zahlentheoretische Funktion, sodass für teilerfremde Zahlen $\varphi(m * n) = \varphi(m) * \varphi(n)$ gilt.[2]

3.3 Euklidischer Algorithmus

Der euklidische Algorithmus wurde von dem griechischen Mathematiker Euklid entwickelt. Mit dem euklidischen Algorithmus kann der größte gemeinsame Teiler – kurz ggT – zweier Zahlen effizient berechnet werden.

Satz 1: $Seien\ a \in \mathbb{N}_0, b \in \mathbb{N}.\ Dann\ gilt\ ggT(a,b) = ggT(b, a\ mod\ b)$

Zwei Zahlen haben immer einen gemeinsamen Teiler ((-1) und 1) und keiner kann größer sein als die kleinere der beiden Zahlen.[3]

$a > b$

$b = r_0$

---

[1] Witt, Kurt-Ulrich: *Algebraische Grundlagen der Informatik, 3*. Auflage, 2007, S. 162 verfügbar unter: https://link.springer.com/content/pdf/10.1007/978-3-322-91825-3.pdf, eingesehen am 10.05.2021.

[2] (Für Unterkapitel 3.2) Eulersche Phi-Funktion, Wikipedia, Die freie Enzyklopädie, verfügbar unter https://de.wikipedia.org/w/index.php?title=Eulersche_Phi-Funktion&oldid=211167258, eingesehen am 15.5.2021.

[3] Weitz, Edmund: Konkrete Mathematik (nicht nur) für Informatiker. Wiesbaden: Springer Fachmedien Wiesbaden, 2018, S. 53-54, verfügbar unter: https://link.springer.com/content/pdf/10.1007/978-3-662-62618-4.pdf, eingesehen am 13.05.2021.

$$a = q_1 * r_0 + r_1$$

$$r_0 = q_2 * r_1 + r_2$$

$$r_1 = q_3 * r_2 + r_3$$

.

.

.

$$r_{n-1} = q_{n+1} * r_n + 0$$

Bei dem euklidischen Algorithmus handelt es sich um einen rekursiven Algorithmus, da in jedem weiteren Schritt mit dem Divisor und dem Rest des vorhergehenden Schritts eine erneute Division mit Rest durchgeführt wird, bis der Rest 0 ist.[4]

$$ggT(a, b) = r_n$$

In der Programmiersprache Python kann der Algorithmus wie in der Abbildung implementiert werden.

```
def ggT(a, b):
    while b != 0:
        a, b = b, a % b
    return a
```

Beispiel 1: $ggT(544, 391)$

```
print(ggT(544, 391))
```

$ggt(544{,}391) = 17$

3.4 Erweiterter Euklidischer Algorithmus

Mit dem erweiterten euklidischen Algorithmus kann eine Linearkombination des größten gemeinsamen Teilers zweier Zahlen a und b bestimmen werden.

Satz 2: $(Lemma\ von\ Bézout)$

$Seien\ a, b\ \in \mathbb{N}_0.\ Der\ größte\ gemeinsame\ Teiler\ ggT(a, b)$
$lässt\ sich\ als\ ganzzahlige\ Linearkombination\ von\ a\ und\ b\ darstellen$:

$$ggT(a, b) = a * x + b * y\ mit\ \ x, y\ \in\ \mathbb{Z}$$

[4] Koc, Aygül: Einführung in die Kryptologie und ihre Vermittlung im Schulunterricht, Innsbruck, 2019, S.11, verfügbar unter: https://www.uibk.ac.at/mathematik/algebra/media/teaching/diplomarbeit_koc.pdf eingesehen am 10.05.2021.

Der Algorithmus wird in der Kryptografie verwendet um das multiplikative Inverse einer Zahl $a\ Modulo\ n$ zu bestimmen.

$$ggT(a,b) = 1 = a * x + b * y$$

$$a^{-1} * a = 1\ mod\ n$$

In der Programmiersprache Python lässt sich der Algorithmus wie folgt implementieren.[5]

```
def erggT(a, b):
    u, v, s, t = 1, 0, 0, 1
    while b!=0:
        q=a//b
        a, b = b, a-q*b
        u, s = s, u-q*s
        v, t = t, v-q*t
    return a, u, v
```

3.5 Einwegfunktionen

Definition: $Eine\ Funktion\ f: X \rightarrow Y\ heißt\ Einwegfunktion,\ falls\ für\ alle\ x \in X\ das\ f(x)\ leicht\ zu\ berechnen\ ist,\ und\ für\ fast\ alle\ y \in Y\ das\ f^{-1}\ (y)\ schwierig.$

Eine Einwegfunktion ist eine Funktion, die sich nur sehr schwer umkehren lässt. Ein Beispiel ist die Multiplikation zweier Primzahlen, dessen Berechnung leicht möglich ist, wohingegen die Primfaktorzerlegung schwierig ist.[6]

Leicht = In Polynomialzeit lösbar

Schwer = Nicht in Polynomialzeit

4. Verschlüsselung

Die Verschlüsselung wird zum Schutz von Daten eingesetzt. Dabei werden die Daten in ein unleserliches verschlüsseltes Format konvertiert, das nur nach einer Entschlüsselung wieder lesbar ist. Im Internet und in der Welt der elektronischen Geräte wird die Verschlüsselung eingesetzt, um die Sicherheit der Daten zu gewährleisten. Des Weiteren wird die Verschlüsselung für die Geheimhaltung, Integrität, Authentizität und Verbindlichkeit verwendet.

Das Programm „Main.py“ kann nun ausgeführt werden und während der Seminararbeit als Hilfe oder Anwendungsbeispiel dienen.

[5] (Für Unterkapitel 3.4) Erweiterter euklidischer Algorithmus, Hochschule Flensburg, https://www.inf.hs-flensburg.de/lang/krypto/algo/euklid.htm, eingesehen am 14.05.2021.

[6] (Für Unterkapitel 3.5)Frettlöh, Dirk: Technische Fakultät Universität Bielefeld, Universität Bielefeld, 2020, S.18, verfügbar unter https://www.math.uni-bielefeld.de/~frettloe/teach/krypto/krypto2020.pdf, eingesehen am 11.05.2021.

4.1 Sicherheit

Definition: “
Ein Verfahren gilt als effizient und sicher, falls fur alle m das f(e, m) einfach zu berechnen ist; f ∗ (d, c) ebenso, falls man das d kennt; und f ∗ (d, c) soll fur fast alle c schwer zu berechnen sein, falls man das d nicht kennt.“[7]

5. Symmetrische Verschlüsselung

Die Menschen benutzten die symmetrische Verschlüsselung bereits in der römischen Republik. Gaius Julius Caesar hat sie benutzt, um seinen Generälen geheime militärische Nachrichten zu übermitteln.

Bei einem symmetrischen Verschlüsselungsverfahren wird derselbe Schlüssel für die Ver- und Entschlüsselung verwendet. Diese Verfahren eignen sich gut zur Verschlüsselung von Datenspeichern, jedoch nicht zur Kommunikation im Internet. Die Problematik entsteht, wenn zwei Akteure kommunizieren wollen, da das Austauschen des Schlüssels nicht sicher über das Internet möglich ist. Der Schlüssel müsste ebenfalls chiffriert werden, damit er nicht zum Dechiffrieren der Nachricht verwendet werden kann.

Das bedeutet, dass sich beide Personen, die sicher verschlüsselt kommunizieren wollen, treffen müssen, um den Schlüssel in der realen Welt austauschen. Die Übergabe des Schlüssels ist nicht nur risikoreich, sondern auch unpraktisch, da die physische Entfernung sehr hoch sein kann. [8]

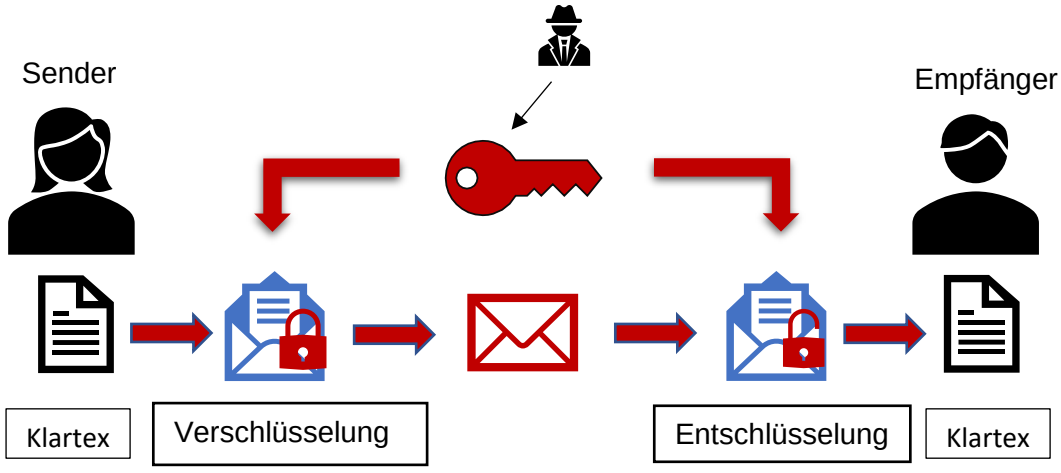


Abb. 1

[7] Frettlöh, Dirk: Technische Fakultät Universität Bielefeld, Universität Bielefeld, 2020, S.5, verfügbar unter https://www.math.uni-bielefeld.de/~frettloe/teach/krypto/krypto2020.pdf, eingesehen am 11.05.2021.

[8] Küsters, Ralf und Thomas, Wilke: Moderne Kryptographie. Vieweg + Teubner, 2011, S.7, verfügbar unter: https://link.springer.com/content/pdf/10.1007%2F978-3-8348-8288-2.pdf, eingesehen am 11.05.2021.

Das Schlüsselaustauschproblem wird in Abb. 1 deutlich. Wenn eine symmetrisch verschlüsselte Nachricht über das Internet verschickt wird, dann müsste der Schlüssel, wie bereits erwähnt, ebenfalls versendet werden, da der Empfänger die verschlüsselte Nachricht sonst nicht entschlüsseln könnte. Der Schlüssel könnte zwar über das Internet versendet werden, jedoch könnten Unbefugte diesen Schlüssel abfangen und die Nachricht dechiffrieren oder selbst unbemerkt Nachrichten schreiben. Das macht die Kommunikation anhand symmetrischer Verschlüsselung über das Internet unsicher. Des Weiteren müsste jeder Sender einen neuen Schlüssel erstellen und mit dem Empfänger austauschen, wenn er mit einer anderen Person kommunizieren will, da seine anderen Kontakte diese Nachrichten sonst entschlüsseln könnten.

Das sichere Aufrufen einer Internetseite wäre mit einem symmetrischen Verfahren unmöglich, da sich der Computer und der Server der Internetseite nicht im geheimen auf ein Passwort einigen könnten.

6. Asymmetrische Verschlüsselung

Die Lösung des Problems ist die asymmetrische Verschlüsslung, das sogenannte Public-Key-Verfahren.

Die Menschen gingen bis zum Jahr 1976 davon aus, dass Sender und Empfänger einen gemeinsamen Schlüssel benötigen, damit sie Nachrichten vertraulich übermitteln können. Doch dann erschien der Artikel „New Directions in Cryptography“ von Whitfield Diffie und Martin Hellman, in dem das erste Public-Key-Verfahren vorgestellt wurde, die Diffie-Hellman-Schlüsselvereinbarung. Die Mathematiker Ron Rivest, Adi Shamir und Leonard Adleman haben bei dem Versuch, die Annahme von Diffie und Hellman zu widerlegen, das bekannteste asymmetrische Verschlüsselungsverfahren, die RSA-Verschlüsslung im Jahr 1978 erfunden.[9]

[9] Schwenk, Jörg: Sicherheit und Kryptographie im Internet. Vieweg + Teubner Verlag, 2005, S.21, verfügbar unter: https://link.springer.com/content/pdf/10.1007/978-3-658-29260-7.pdf, eingesehen am 10.05.2021.

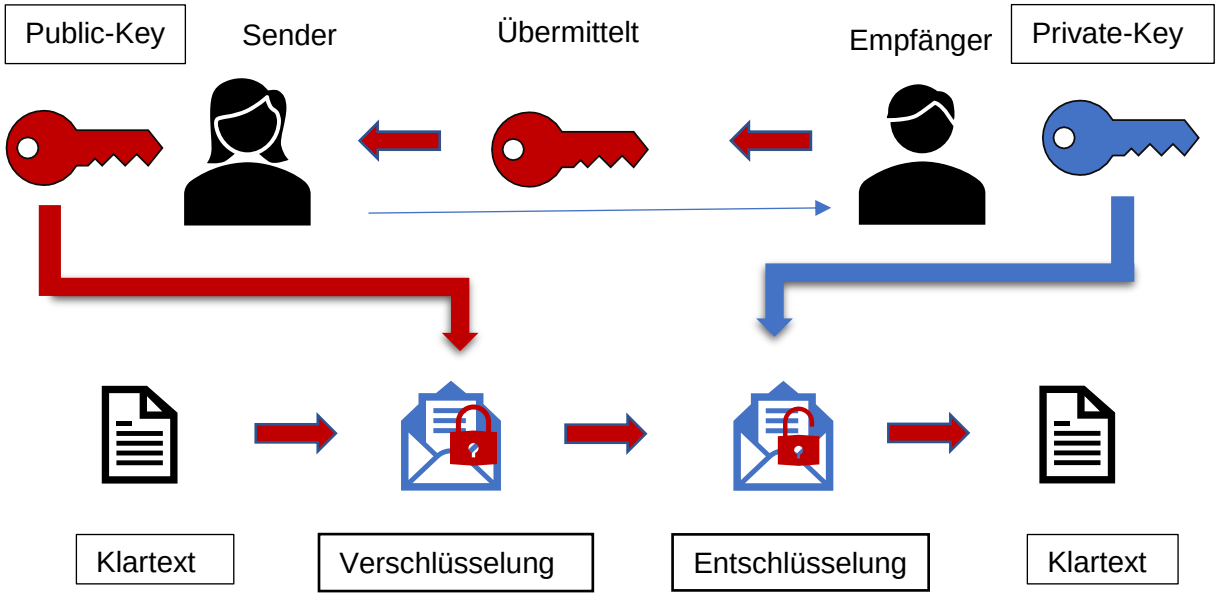


Abb. 2

Bei der asymmetrischen Verschlüsselung existieren zwei Schlüssel, die zusammen ein Schlüsselpaar ergeben, einen, um die Nachricht zu verschlüsseln und einen um die Nachricht zu entschlüsseln. In Abb. 2 wird der Public-Key rot dargestellt und der Private-Key blau. Der Sender fragt den Empfänger an und dieser übermittelt den Public-Key. Der Sender verschlüsselt seine Nachricht mit dem Public-Key und schickt die verschlüsselte Nachricht an den Empfänger, der sie mit seinem Private-Key entschlüsselt.[10] Der öffentliche Schlüssel (Public-Key) kann mit jeder beliebigen Person geteilt werden, da die verschlüsselte Nachricht nur mit dem privaten Schlüssel dechiffriert werden kann.

Im Gegensatz zu den symmetrischen Verfahren gibt es kein Schlüsselaustauschproblem aufgrund des Faktes, dass der Schlüssel ohne bedenken über das Internet übermittelt werden kann, da die Nachricht, wenn sie verschlüsselt wurde, nur mit dem Private-Key entschlüsselt werden kann. So wäre es ebenfalls möglich, Nachrichten von mehreren Personen, die denselben Schlüssel benutzen, zu erhalten, ohne dass Unbefugte die Nachrichten lesen könnten.

Die Sicherheit bei den asymmetrischen Verschlüsselungsverfahren gewährleisten die Einwegfunktionen.

Um dies weitgehend zu erklären, wird nun ausgeführt, wie die RSA-Verschlüsslung funktioniert.

7. RSA-Verfahren

[10] Rivest, Ronald L., Adi Shamir, und Leonard Adleman: A method for obtaining digital signatures and public-key cryptosystems., Communications of the ACM 21.2 (1978): 120-126, S. 1-4, verfügbar unter: https://apps.dtic.mil/sti/pdfs/ADA606588.pdf, eingesehen am 10.05.2021.

7.1 Konstruktion des Schlüssels

1. Der Empfänger wählt zwei Primzahlen p und q (geheim)
2. Der Empfänger berechnet $N = p * q$ und $\varphi(N)\ als\ (\mathrm{p}-1)*(\mathrm{q}-1)$
3. Der Empfänger wählt $e \neq 1$ mit $ggT(e, \varphi(N)) = 1$ und d mit $e * d \equiv 1\ mod\ \varphi(N)$ (Damit e ein inverses hat $mod\ \varphi(N)$)
4. Der Empfänger gibt N und e öffentlich bekannt. $\varphi(N)$ und d sind geheim.

- Der Sender kann nun die Nachricht blockweise in eine Zahl m verwandeln und m verschlüsseln als $(m^e)^d \equiv m^{ed} \equiv m\ mod\ N$.[11]

7.2 Korrektheit

Die Frage ist nun warum $m^{ed} \equiv m\ mod\ N$ funktioniert. Aufgrund des Satzes von Euler-Fermat: Ist $ggT(m, N) = 1$, dann ist $m^{\varphi(N)} \equiv 1\ mod\ N$. Da $ed \equiv 1\ mod\ \varphi(N)$ ist $ed = k\varphi(N) + 1$ für ein $k \in \mathbb{N}$.

$c^d = (m^e)^d \equiv m^{ed} \equiv m^{1+k\varphi(N)} \equiv m\ mod\ N.$

e und d sind invers zueinander $mod\ \varphi(N)$, damit ist $e * d$ ein Vielfaches von $\varphi(N) + 1$. Wegen des Satzes von Euler-Fermat ist $m^{k*\varphi(N)+1} = m\ mod\ N$, da beim Verschlüsseln und Entschlüsseln polynomiell durch modulares Potenzieren ($mod\ N$) gerechnet wird.[12]

Dieser Beweis funktioniert nur für $ggT(m, N) = 1$.[13]

7.3 Sicherheit

Die Sicherheit des RSA-Verfahrens basiert auf der Schwierigkeit der Faktorisierung eines Produktes. Eine Variante der Einwegfunktionen sind Trapdoor-Einwegfunktionen, die beim RSA-Verfahren verwendet werden. Das Produkt N lässt sich effizient aus zwei großen Primzahlen (2048 – 4096 Bit) berechnen, jedoch ist die Umkehrung - nach aktuellen Wissenstand – nur mit exorbitanter Rechenleistung und extrem viel Zeit möglich. Es ist eine Trapdoor, da d die Lösung der Gleichung $e * x = 1\ mod(\varphi(N))$ polynomiell berechenbar ist. Dadurch wird deutlich, dass die Sicherheit des RSA-Verfahrens auf Einwegfunktionen basiert und die Definition von

[11] Frettlöh, Dirk: Technische Fakultät Universität Bielefeld, Universität Bielefeld, 2020, S.18, verfügbar unter https://www.math.uni-bielefeld.de/~frettloe/teach/krypto/krypto2020.pdf, eingesehen am 11.05.2021.

[12] Frettlöh, Dirk: Technische Fakultät Universität Bielefeld, Universität Bielefeld, 2020, S.18, verfügbar unter https://www.math.uni-bielefeld.de/~frettloe/teach/krypto/krypto2020.pdf, eingesehen am 11.05.2021.

[13] Frettlöh, Dirk: Technische Fakultät Universität Bielefeld, Universität Bielefeld, 2020, S.18, verfügbar unter https://www.math.uni-bielefeld.de/~frettloe/teach/krypto/krypto2020.pdf, eingesehen am 11.05.2021.

Sicherheit erfüllt (vgl.1.4), wenn jeweils mehr als 512 bis 2048 Bit für p und q verwendet werden. [14]

Es ist derzeit unbekannt, ob es einen Polynomzeit-Algorithmus gibt, der die Primfaktorzerlegung einer großen Zahl berechnet. Es wird angenommen, dass ein solcher Algorithmus nicht existiert, jedoch gibt es einen Algorithmus, der auf Quantencomputern (Shor-Algorithmus) basiert, mit dem es möglich wäre, alle verschiedenen Möglichkeiten einer Verschlüsselung gleichzeitig zu berechnen. Die Technik ist jedoch noch nicht soweit und es bleibt fraglich, ob es jemals einen funktionsfähigen Quantencomputer geben wird.[15]

Das RSA-Verfahren ist deshalb aktuell sicher, da die Faktorisierung eines 4096 Bit Schlüssels Hunderte Jahre dauern würde, da es 2^{4096} verschiedene Möglichkeiten gibt. Des Weiteren ist die Größe des Schlüssels nicht begrenzt, da es unendliche Primzahlen und damit auch unendlich viele Schlüssel gibt. Wenn die Rechenleistung und die Energieeffizienz der Prozessoren in den nächsten Jahren steigen würde, dass erwartet wird, dann könnten die Schlüssel verlängert werden. Dies wäre bereits jetzt theoretisch möglich, in der Praxis aber aufwendig, da das Erhöhen der Primzahlen der Schlüssel, die Server stärker auslasten würde.

„Can the reader say what two numbers multiplied together will produce the number 8616460799? I think it unlikely that anyone but myself will ever know." – 1874, William Stanley Jevons[16]

Der Philosoph William Stanley Jevons hat vor fast hundert Jahren eine Zahl aufgeschrieben und behauptet, dass niemals jemand wissen wird, was die Faktorisierung der Zahl ist. Zu dieser Zeit gab es noch keine Computer und deshalb war es unvorstellbar, alle möglichen Kombinationen auszuprobieren, jedoch ist die Faktorisierung der Zahl 8616460799 mittlerweile relativ einfach mit einem ineffizienten Algorithmus möglich ($89681 * 96079 = 8616460799$). Das Problem dabei ist, dass die Daten, die mit kurzen Schlüsseln aus der Vergangenheit verschlüsselt wurden, mittlerweile entschlüsselt werden können. Es kann nicht vorausgesagt werden, ob es einen Polynomzeit-Algorithmus, einen vollständig funktionierenden Quantencomputer oder exorbitante Leistungsanstiege und Effizienzanstiege geben wird und damit das Entschlüsseln vieler Teile des Internets erleichtert.

Im Vergleich zu den symmetrischen Verfahren hat die RSA-Verschlüsselung kein Schlüsselaustauschproblem, da der öffentliche Schlüssel, wie der Name es auch schon sagt, öffentlich mit jeder Person geteilt werden kann, da nur mit dem privaten Schlüssel die mit dem öffentlichen Schlüssel chiffrierte Nachricht dechiffriert werden

[14] Frettlöh, Dirk: Technische Fakultät Universität Bielefeld, Universität Bielefeld, 2020, S.20, verfügbar unter https://www.math.uni-bielefeld.de/~frettloe/teach/krypto/krypto2020.pdf, eingesehen am 11.05.2021.

[15] Küsters, Ralf, und Thomas Wilke: Moderne Kryptographie. Vieweg+ Teubner, 2011, S. 149, verfügbar unter https://link.springer.com/content/pdf/10.1007%2F978-3-8348-8288-2.pdf, eingesehen am 16.05.2021.

[16] Golomb, Solomon W: ON FACTORING JEVONS'NUMBER, Cryptologia, 1996, S. 243, verfügbar unter https://www.tandfonline.com/doi/pdf/10.1080/0161-119691884933?casa_token=5Gg06c4i2dIAAAAA:j7xS7n0ZdTlin5zAsLkcyaZXgpxyXbHHvoXVoGkwHjldCqNjU2iv3KsdKuhUO74a-T_9TGsdzbayoQ, eingesehen am 20.05.2021.

kann. Ein Sicherheitsproblem, das dabei entsteht, ist das Authentifizierungsproblem. Die Problematik besteht darin, den Herausgeber des öffentlichen Schlüssels zu identifizieren, da dieser jemand anderes sein kann, der selbst Schlüssel generiert, um den Sender zu täuschen. Damit wäre es möglich, dass der Sender der Nachricht den falschen Schlüssel benutzt und dem Betrüger die Möglichkeit gibt, die geheime Nachricht zu entschlüsseln. Es gibt viele Möglichkeiten, um dieses Problem zu lösen. Ein Beispiel wäre, die Telefonnummer und die dazugehörige SIM-Karte als Authentifikation zu benutzen, um die Identität zu bestätigen.

Es ist wichtig, dass der private Schlüssel nicht geteilt wird, da die Nachrichten sonst entschlüsselt werden könnten. Um dies Vorzubeugen, ist der private Schlüssel dem Benutzer einer App oder Webseite in den meisten Fällen nicht bekannt. Dies wird besonders wichtig, wenn es sich um eine Hybridverschlüsselung handelt, da dabei der symmetrische Schlüssel übertragen wird.

7.4 Beispiel

Das Beispiel wird mit kleinen Zahlen durchgeführt, um das Verständnis zu verbessern. Normalerweise werden p und q mit jeweils mehr als 512 bis 2048 Bit gewählt.

1. Der Empfänger wählt zwei Primzahlen $p = 23$ und $q = 13$.
2. Der Empfänger berechnet das Produkt $N = 23 * 13 = 299$ und die Anzahl der teilerfremden Zahlen zu N, mit der eulerschen Phi-Funktion $\varphi\ (\ N\ ) = (23 - 1)(13 - 1) = 264$.
3. Der Empfänger wählt $e = 173\ \big(ggT(e, 264)\big) = 1$ und berechnet d mit $173 * d\ \equiv 1\ mod\ 264$ das multiplikative Inverse $mod\ \varphi\ (\ N\ )$, also $d = 29$ mit dem erweiterten euklidischen Algorithmus.
4. Der Empfänger übermittelt $N = 299$ und $e = 173$ über das Internet.
5. Der Sender kann nun eine Botschaft m verschlüsseln. Zum Beispiel A in ASCII[17] m $= 65$.

   Privat$(d = 29, N = 299)$ und Öffentlich$(e = 173, N = 299)$

   Verschlüsseln $= m^e\ mod\ N = 65^{173}\ mod\ 299 = c = 221$

   Entschlüsseln $= c^d\ mod\ N = 221^{29}\ mod\ 299 = m = 65$, da $(65^{(173*29)})\ mod\ 299 = m = 65$ ist.

6. Der Sender verschlüsselt die Nachricht 65 und sendet sie an den Empfänger.
7. Der Empfänger entschlüsselt die Nachricht und kann sie lesen.

7.4 Test

Der Versuch, eine Zahl mit der Länge von 2048 Bit zu faktorisieren ist wie erwartet fehlgeschlagen.

Die Faktorisierung dieser Zahl ist derzeitig nur mit einer exorbitanten Rechenleistung und viel Zeit möglich.

[17] ASCII = American Standard Code for Information Interchange

Diese Zahl kann als Beweis einer Identität verwendet werden. Zum Beispiel wäre es möglich zu beweisen, dass ich diese Zahl berechnet habe, da ich die Primfaktoren kenne.

112948360001851820047393271350743380237027435955089965356053861371131318581149804140260787045818349507206564138583562553826241567629803505323817133988371638616116599482670022747519021680033619826066372227894518742547067376919262336027558822847678050481774316699993486950012826519199426282079546069323885234795366981628134514013250244954039639221647087829569684300455313950735004977547853874242880934788073220670582250721479358744660989500354308729427968048230847395842096384673584150031695623102675574135766571836038211110037347884466023602322199769899502096620890317481722710193287801895267336678350892940670291086 61

8. Resümee

Aber inwiefern ist die RSA-Verschlüsselung jetzt sicher anwendbar oder lösbar und unsicher?“

Die RSA-Verschlüsselung wird heutzutage, wenn auch verändert, in fast allen digitalen Bereichen verwendet, um die Sicherheit der Daten zu gewährleisten. Des Weiteren funktioniert das Verfahren und ist mathematisch korrekt, wie am Beispiel oder am Beweis zu erkennen ist. Es ist sogar zur Selbstverständlichkeit geworden, dass die Nachrichten von niemanden gelesen werden können. Das RSA-Verfahren ist im Vergleich zu symmetrischen Verfahren adäquat für die Kommunikation im Internet geeignet, da das Schlüsselaustauschproblem von den asymmetrischen Verfahren gelöst wurde. Das System ist öffentlich bekannt und dies ist in der Welt der Kryptografie als wichtig angesehen, da sich das Verfahren auf die Mathematik verlässt. Die Trapdoor-Einwegfunktionen machen das Verfahren sicher und effizient. Die Sicherheit des RSA-Verfahrens basiert auf der Schwierigkeit der Faktorisierung eines Produktes. Das Produkt N lässt sich effizient aus zwei großen Primzahlen (2048 – 4096 Bit) berechnen, jedoch ist die Umkehrung - nach aktuellen Wissenstand – nur mit exorbitanter Rechenleistung möglich. Es ist eine Trapdoor, da d die Lösung der Gleichung $e * x = 1\, mod(\varphi\ (\ N\ )\ )$ polynomiell berechenbar ist. Dadurch wird deutlich, dass die Sicherheit des RSA-Verfahrens auf Einwegfunktionen basiert und die Definition von Sicherheit erfüllt, wenn jeweils mehr als 512 bis 2048 Bit für p und q verwendet werden, wie am Test eindeutig wird. Es ist damit theoretisch lösbar, aber nur ineffizient mit Brute-Force Attacken.

Die Sicherheit des RSA-Verfahrens ist damit momentan gewährleistet, wenn das Verfahren richtig implementiert wurde, jedoch kann am Zitat vom Philosophen William Stanley Jevons erkannt werden, dass die Technik voranschreitet und es möglicherweise in der Zukunft Methoden geben wird, sehr große Zahlen effizient in ihre Primfaktoren zu zerlegen. Dies ist jedoch noch nicht der Fall und im Normalfall verbessern sich Prozessoren nur um ein paar Prozent pro Jahr. Um dies vorzubeugen, wäre es sinnvoll, die Länge der Schlüssel zu erhöhen.

Anschließen lässt sich sagen, dass das RSA-Verfahren deshalb aktuell sicher ist, wenn die Schlüssellänge über 2048 Bit beträgt und die Primzahlen stochastisch

unabhängig berechnet werden, da die Faktorisierung eines 4096 Bit Schlüssels Hunderte Jahre dauern würde, da es 2^{4096} verschiedene Möglichkeiten gibt. Darüber hinaus ist die Größe des Schlüssels nicht begrenzt, da es unendliche Primzahlen und damit auch unendlich viele Schlüssel gibt.

9. Zusammenfassung

Das RSA-Verfahren wird in vielen Bereichen des digitalen Lebens angewendet, um die Sicherheit der Daten zu gewährleisten. Es ist mathematisch korrekt und wenn es richtig angewendet wird sicher. Ich bin bei der Auswertung meines Tests auf dasselbe Ergebnis gekommen, wie es in den Quellen beschrieben wurde. Die Sicherheit des Verfahrens ermöglicht sichere Kommunikation im Internet, jedoch bin ich bei meiner Recherche darauf gekommen, dass das RSA-Verfahren bei richtiger Implementierung zwar sicher ist, es aber extra manipuliert oder ungewollt schachstellen im System des jeweiligen Dienstes enthalten kann. Damit sind die Daten für ausgewählte Personen oder Hacker über eine Hintertür erreichbar. Ich bin darauf nicht explizit in meiner Beantwortung eingegangen, da es nichts mit der Sicherheit des RSA-Verfahrens zu tun hat, sondern lediglich mit der Implementierung. Ich will jedoch darauf hinweisen, dass die Daten in diesen Fällen natürlich nicht Geschütz sind und mit der Sicherheit der Server des jeweiligen Dienstleisters zusammenhängt.

10. Reflexion

Ich habe die im Unterricht erlernten Fähigkeiten wie das richtige Zitieren von Quellen angewendet. Dazu haben mir die mathematischen Grundlagen aus der Schulzeit extrem weitergeholfen, um die Logik zu verstehen. Die Programmierfähigkeiten, die ich teilweise in der Schule und zu Hause gelernt habe, haben mir beim Verständnis der komplexen mathematischen Themen weitergeholfen.

Ich habe bei der Bearbeitung dieser Seminararbeit sehr viel dazugelernt. Es ist mir im Laufe der Bearbeitung immer leichter gefallen, an schweren Problemen zu arbeiten. Des Weiteren hat das intensive Arbeiten mit Quellen dazu geführt, dass ich mich mit höherer Mathematik beschäftigt und dadurch gelernt habe, wie man mathematische Formeln liest und soweit es geht Beweise durchführt. Darüber hinaus habe ich gelernt, wie komplexe Algorithmen in Python implementiert und angewendet werden. Ich bin nun in der Lage, meine Daten selbst zu verschlüsseln und damit, falls ich ein Projekt zum Thema Internet starten würde, die Sicherheit gewährleisten könnte.

Die Bearbeitung der Seminararbeit hat mir insgesamt gut gefallen, da ich viele neue Fähigkeiten empirisch gelernt habe. Ich muss jedoch sagen, dass die Bearbeitung etwas nervig war, da ich viel von meiner Freizeit opfern musste. Ich würde beim nächsten Mal die Bearbeitung über mehrere Wochen aufteilen und nicht in den Ferien machen. Ich finde meine Ausarbeitung präzise für die geringe Anzahl an Seiten, die wir verwenden durften. Ich wusste vorher nicht wie sicher eine RSA-Verschlüsselung wirklich ist.

11. Ausblick

Zukünftig muss die Entwicklung der Technik betrachtet werden, da die Weiterentwicklung dazu führen könnte, dass die Schlüssellänge erhöht werden muss, da die Entschlüsselung sonst mithilfe schnellerer Leistung und Effizienz in höherer Geschwindigkeit und mit weniger Ressourcen möglich wäre.

12. Literaturverzeichnis

Buchquellen/Fachzeitschriften

1. Witt, Kurt-Ulrich: Algebraische Grundlagen der Informatik, 3. Auflage, 2007, S. 162 verfügbar unter: https://link.springer.com/content/pdf/10.1007/978-3-322-91825-3.pdf, eingesehen am 10.05.2021.

2. Weitz, Edmund: Konkrete Mathematik (nicht nur) für Informatiker. Wiesbaden: Springer Fachmedien Wiesbaden, 2018, S. 53-54, verfügbar unter: https://link.springer.com/content/pdf/10.1007/978-3-662-62618-4.pdf, eingesehen am 13.05.2021.

3. Koc, Aygül: Einführung in die Kryptologie und ihre Vermittlung im Schulunterricht, Innsbruck, 2019, S.11, verfügbar unter: https://www.uibk.ac.at/mathematik/algebra/media/teaching/diplomarbeit_koc.pdf eingesehen am 10.05.2021.

4. Frettlöh, Dirk: Technische Fakultät Universität Bielefeld, Universität Bielefeld, 2020, S.18, verfügbar unter https://www.math.unibielefeld.de/~frettloe/teach/krypto/krypto2020.pdf, eingesehen am 11.05.2021.

5. Küsters, Ralf und Thomas, Wilke: Moderne Kryptographie. Vieweg + Teubner, 2011, S.7, verfügbar unter: https://link.springer.com/content/pdf/10.1007%2F978-3-8348-8288-2.pdf, eingesehen am 11.05.2021.

6. Schwenk, Jörg: Sicherheit und Kryptographie im Internet. Vieweg + Teubner Verlag, 2005, S.21, verfügbar unter: https://link.springer.com/content/pdf/10.1007/978-3-658-29260-7.pdf, eingesehen am 10.05.2021.

7. Rivest, Ronald L., Adi Shamir, und Leonard Adleman: A method for obtaining digital signatures and public-key cryptosystems., Communications of the ACM 21.2 (1978): 120-126, S. 1-4, verfügbar unter: https://apps.dtic.mil/sti/pdfs/ADA606588.pdf, eingesehen am 10.05.2021.

8. Golomb, Solomon W: ON FACTORING JEVONS'NUMBER, Cryptologia, 1996, S. 243, verfügbar unter https://www.tandfonline.com/doi/pdf/10.1080/0161-119691884933?casa_token=5Gq06c4i2dIAAAAA:j7xS7n0ZdTlin5zAsLkcyaZXgpxyXbHHvoXVoGkwHjldCqNjU2iv3KsdKuhUO74a-T_9TGsdzbayoQ, eingesehen am 20.05.2021.

Internetquellen

1. Eulersche Phi-Funktion, Wikipedia, Die freie Enzyklopädie, verfügbar unter: https://de.wikipedia.org/w/index.php?title=Eulersche_PhiFunktion&oldid=211167258 , eingesehen am 15.5.2021.

2. Erweiterter euklidischer Algorithmus, Hochschule Flensburg, https://www.inf.hsflensburg.de/lang/krypto/algo/euklid.htm, eingesehen am 14.05.2021.

Bildquelle:

1. By Unknown author - Popular Science Monthly Volume 11, Public Domain, https://commons.wikimedia.org/w/index.php?curid=11022925

13. Unterschrift

Ich versichere, dass die Seminararbeit von mir selbstständig erarbeitet wurde und ich keine anderen als die angegebenen Hilfsmittel benutzt habe. Diejenigen Teile der Seminararbeit, die anderen Werken im Wortlaut oder dem Sinn nach entnommen wurden, sind als solche kenntlich gemacht.