# Bedeutung von starken Primzahlen für die heutige Zeit

*Eine Einführung in das RSA-Verschlüsselungssystem*

Winterer, Mathis Aaron

*4. Juli 2024*

# Inhaltsverzeichnis

# Abbildungsverzeichnis

# Tabellenverzeichnis

# 1. Einleitung

Kryptographie und Verschlüsselungssysteme wurden schon im Antiken Rom verwendet um Nachrichten zu verschlüsseln[Aic22], doch heutzutage wären Verschlüsselungssysteme wie der „Cäsar-Chiffre“ gegenüber eines Computers nutzlos. Die Rolle von Kryptographie ist aufgrund des leichten Zugangs der Allgemenheit zu leistungsstarken Computern, welche das Brechen von „schwachen“ Verschlüsselungssystemen automatisiert und beschleunigt haben um einiges angestiegen. Aufgrund der nun verfügbaren Rechenleistung wurde die Notwendigkeit für stärkere Verschlüsselungssysteme immer größer. Zur Zeit verwendete Verschlüsselungssysteme sind so konzipiert, dass sie mathematisch schlecht rückwärts zu berechnen sind. Hierzu werden standardmäßig große Primzahlen verwendet, da zur Zeit kein effizienter Algorithmus zur Berechnung von großen Primzahlfaktoren bekannt ist, welcher nicht die Verwendung eines Quantencomputers erfordern würde. Das „brute-forcing“ moderner Verschlüsselungssysteme ist so ineffizient, dass nur ein Teilschritt bis zu fünfzehnhundert Jahre dauern kann[Kle10]. „Für unsere Berechnungen waren mehr als $10^{20}$ Operationen erforderlich. Mit dem Äquivalent von fast 2000 Jahren Rechenzeit auf einem AMD Opteron mit einem Kern von 2,2 GHz […]“[Kle10]. Verschlüsselungssysteme sind aufgrund dieser hohen Anforderungen gegenüber der Resistenz gegen Angriffe äußerst komplexe Systeme welche in der Informationssicherheit eine entscheidende Rolle bei der Sicherung von sensiblen Daten sowohl in der Übertragung als auch bei der Speicherung spielen[Aic22]. Das Ziel dieser Facharbeit ist es den weit verbreiteten und etablierten RSA-Algorithmus zu Analysieren und zu Implementieren. Hierfür werden die einzelnen Teilschritte und verwendeten mathematischen Prinzipien und Funktionen erläutert und beispielhaft dargestellt sowie als Teil eines Programms in C# implementiert, so dass die Funktionen der Schlüsselerzeugung, der Verschlüsselung, der Entschlüsselung und der Signierung verfügbar sind.

## 1.1 Hintergrund der RSA-Verschlüsselung

$n - 1 = d \times 2^j$

MillerRabin.cs

```cs
private static (BigInteger, BigInteger) FirstPart(BigInteger n)
{
    // nMinusOne
    BigInteger nMo = n - 1;
    BigInteger j = 0;
    BigInteger d = nMo;

    // Divide $n-1$ By 2 Until The Result Is Odd, Incrementing `j`
    ↪ Everytime And Storing The Result In `d`, So That `$n-1=d*2^j$`
    while (d % 2 == 0)
    {
        d /= 2;
        j++;
    }

    return (d, j);
}
```

| Item | | |
|---|---|---|
| Animal | Description | Price ($) |
| Gnat | per gram | 13.65 |
| | each | 0.01 |
| Gnu | stuffed | 92.50 |
| Emu | stuffed | 33.33 |
| Armadillo | frozen | 8.99 |

Tabelle 1.1: Tabelle

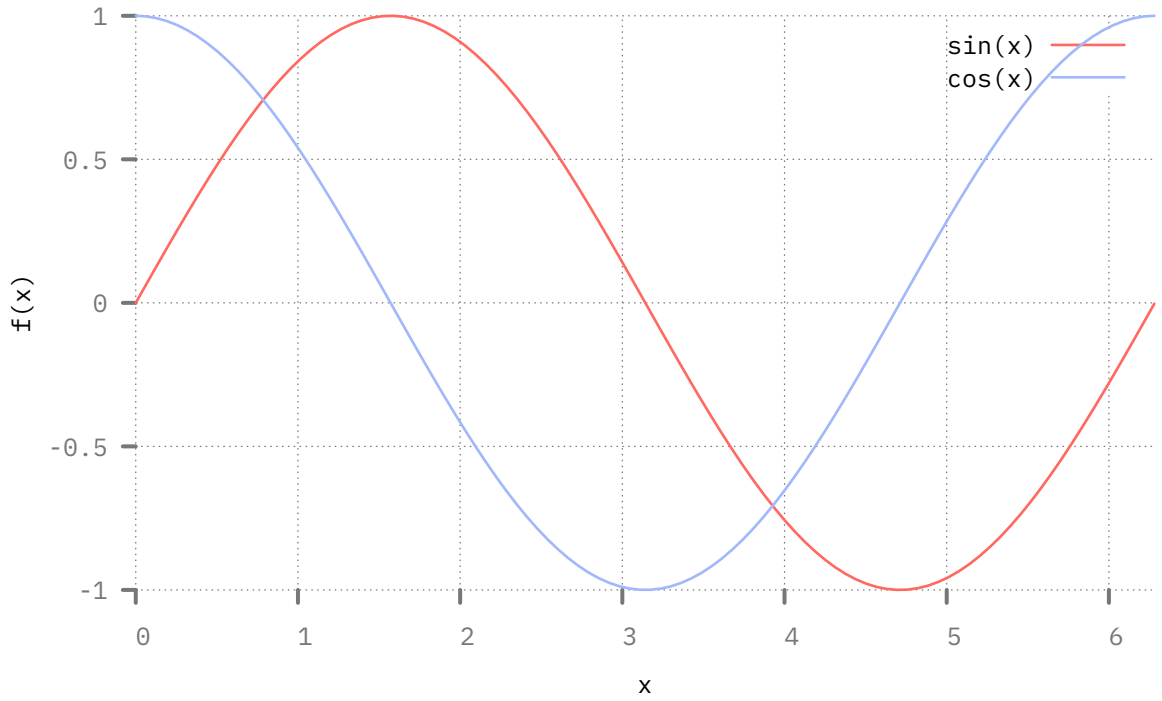


Abbildung 1.1: sin(x) über [0; 6,28]

## 1.2 Anwendungsbereiche der RSA-Verschlüsselung

# 2. Mathematische Grundlagen der RSA-Verschlüsselung

## 2.1 Starke Primzahlen

## 2.2 Eulersche Phi-Funktion

## 2.3 Carmichael-Funktion

## 2.4 Euklidischer Algorithmus

### 2.4.1 Erweiterter Euklidischer Algorithmus

## 2.5 Hamming-Gewicht

## 2.6 Chinesischer Restsatz

## 2.7 Kleiner fermatscher Satz

# 3. Anwendungsbeispiel

## 3.1 Schlüsselerzeugung

Für die Erzeugung eines Schlüsselpaares werden zwei zufällige große Primzahlen $p$ und $q$ gewählt. Um das Errechnen von $p$ und $q$ durch Faktorisierung von $n$ zu Erschweren sollten Primzahlen für $p$ und $q$ gewählt werden, welche beide „groß“[1] und „weit“ auseinander liegen. Beide Primfaktoren $p$ und $q$ sind durch die Bitgröße von $n$, für welche in der Regel eine Zweierpotenz gewählt wird, beschränkt, wobei $p$ und $q$ normalerweise entweder gleich groß oder annähernd gleich groß sind. Für jeden Wert $n$ ist die Bytegröße $nB$:

$$\{nB \in \mathbb{N} \mid 6 \leq nB \leq 512\} \tag{3.1}$$

$$nB = \lfloor \log_2 n \rfloor + 1 = \lfloor \frac{\log n}{\log 2} \rfloor + 1 \tag{3.2}$$

So ergibt sich für $nB = 512$ $p$, $q$ und $n$:

$$\{p \in \mathbb{N} \mid 5 < p \leq 2^{256} - 1\} \tag{3.3}$$

$$\{q \in \mathbb{N} \mid 11 < q \leq 2^{256} - 1\} \tag{3.4}$$

$$\{n \in \mathbb{N} \mid 55 \leq n \leq 2^{512} - 1\} \tag{3.5}$$

[1] Derzeitig sichere RSA-Schlüssel besitzen zwischen $1024$-und $4096$-Bits beziehungsweise $128$-und $512$-Bytes

## 3.2 Schlüsselverteilung

## 3.3 Verschlüsselung

## 3.4 Entschlüsselung

## 3.5 Signatur

# Literatur


[Aic22] Daniela Aichner. „Primzahlen und ihre Bedeutung in der Kryptographie“. [Online; accessed 28-June-2024]. Diplomarbeit. Innsbruck, Österreich: Universität Innsbruck, Feb. 2022. URL: https://www.uibk.ac.at/mathematik/algebra/media/teaching/diplomarbeit.pdf.

[con24a] Wikipedia contributors. *Fermat primality test — Wikipedia, The Free Encyclopedia*. [Online; accessed 28-June-2024]. 2024. URL: https://en.wikipedia.org/w/index.php?title=Fermat_primality_test%5C&oldid=1227031378.

[con24b] Wikipedia contributors. *Miller–Rabin primality test — Wikipedia, The Free Encyclopedia*. [Online; accessed 28-June-2024]. 2024. URL: https://en.wikipedia.org/w/index.php?title=Miller%E2%80%93Rabin_primality_test%5C&oldid=1222212331.

[con24c] Wikipedia contributors. *RSA (cryptosystem) — Wikipedia, The Free Encyclopedia*. [Online; accessed 28-June-2024]. 2024. URL: https://en.wikipedia.org/w/index.php?title=RSA_(cryptosystem)%5C&oldid=1221958777.

[Kle10] Thorsen Kleinjung. „Factorization of a 768-Bit RSA modulus“. In: (Feb. 2010). [Online, accessed 28-June-2024]. URL: https://eprint.iacr.org/2010/006.pdf.

[Rob99] Eric Roberts. *Prime Generation & Testing for primality*. [Online; accessed 28-June-2024]. 1999. URL: https://www-cs-faculty.stanford.edu/people/eroberts/courses/soco/projects/1998-99/randomized-algorithms/applications/primality.html.

[Sha49] C. E. Shannon. „Communication theory of secrecy systems“. In: *The Bell System Technical Journal* 28.4 (1949). [Online; accessed 28-June-2024], S. 656–715. DOI: 10.1002/j.1538-7305.1949.tb00928.x. URL: https://ieeexplore.ieee.org/document/6769090.