Anwendungsbereiche von probabilistischen Primzahlentests in der Informatik

Technischer Bericht
Eastern Switzerland University of Applied Sciences

Christoph Landolt

Verfasser:	Christoph Landolt
	christoph.landolt@ost.ch
Institut:	ICE Institut für Compuational Engineering
Datum:	10.05.2021

1 Einleitung

Die Anwendungsbereiche der Informatik dringen immer mehr in Bereiche vor, in denen mit sensiblen Daten gearbeitet wird. Aus diesem Grund sind zuverlässige Verschlüsselungsverfahren unabdingbar.
Eines der wichtigsten asymmetrischen Verschlüsselungsverfahren ist der RSA-Algorithmus. Dieser basiert auf der Multiplikation von zwei Primzahlen. In der Praxis haben diese Primzahlen über 300 Stellen. Festzustellen, ob es sich bei so grossen Zahlen um eine Primzahl handelt, ist in der Praxis oft sehr ressourcenintensiv. Diese Arbeit beschäftigt sich mit der Frage, welche Algorithmen heute in der Informatik eingesetzt werden, um festzustellen, ob es sich bei einer gegebenen Zahl um eine Primzahl handelt und wie der aktuelle Stand der Forschung ist.
Zur Beantwortung dieser Frage dient die Beschreibung des RSA-Algorithmus auf Basis wissenschaftlicher Publikationen und wissenschaftlicher Werke aus den Fächern theoretische Informatik und mathematische Zahlentheorie.
Im Bericht werden als Einführung die Grundlagen von Primzahlen und der aktuelle Forschungsstand in der Verteilung von Primzahlen dokumentiert. Die wichtigsten Primzahlentests, welche in der Kryptografie zum Einzatz kommen, werden anschliessend dokumentiert. Zum Schluss werden die Nachteile der aktuellen Verfahren erläutert und mit dem AKS-Verfahren ein mögliches Verfahren der Zukunft identifiziert. Dieser Bericht wurde vom ICE Institut für Compuational Engineering der Ostschweizer Fachhochschule OST in Auftrag gegeben und dient als Grundlage für ein mögliches weiterführendes Forschungsprojekt.

2 Verschlüsselung

Eine der wichtigsten Klassen von Verschlüsselungssystemen stellen die asymmeterischen Kryptosysteme dar. Bei diesen Systemen brauchen die Sender und Empfänger von Daten keinen gemeinsamen geheimen Schlüssel zu kennen. Sowohl der Sender als auch der Empfänger haben je einen öffentlichen und einen privaten Schlüssel. Werden Daten übermittelt, verschlüsselt der Sender diese mit dem öffentlichen Schlüssel des Empfängers. Durch ein mathematisches Verfahren können diese Daten nur noch mit dem privaten Schlüssel des Empfängers in nützlicher Zeit entschlüsselt werden.
Eines der wichtigsten asymmetrischen Kryptosysteme stellt das RSA-Kryptosystem dar. Wie bereits in der Einleitung ausgeführt, basiert diese auf der Faktorisierung von Primzahlen (Beutelspacher, Neumann & Schwarzpaul, 2010). Dies stellt eine sehr hohe Sicherheit dar, da für dieses mathematische Problem noch keine effizienten Verfahren existieren.

3 Primzahlen

3.1 Definition von Primzahlen

Primzahlen sind wie folgt definiert:

Definition 1 *Eine natürliche Zahl p grösser als 1 heisst Primzahl, wenn p keinen positiven Teiler ausser 1 und sich selbst hat (Rosen, 2018).*

Für diese Arbeit sind folgende drei Eigenschaften von Primzahlen relevant. Der mathematische Beweis der folgenden drei Sätze wird in der Referenzliteratur erbracht.

Satz 1 *Jede natürliche Zahl kann eindeutig als Primzahl oder als Produkt von zwei oder mehr Primzahlen geschrieben werden (Rosen, 2018).*

Dieser Satz wird auch Fundamentalsatz der Arithmetik genannt und bildet die Grundlagen des RSA-Algorithmus. Da jede natürliche Zahl als Produkt von Primzahlen geschrieben werden kann, werden die Primzahlen auch Atome der Mathematik genannt. Beispiel: $100 = 2 * 2 * 5 * 5 = 2^2 * 5^2$

Satz 2 *Es gibt unendlich viele Primzahlen (Rosen, 2018).*

Dieser Satz sagt aus, dass die Menge der Primzahlen nicht beschränkt ist, also keine grösste Primzahl existiert. Dies kann mittels des Satzes 1 einfach veranschaulicht werden:
Angenommen es existiert eine grösste Primzahl p_n. Damit wäre die grösstmögliche natürliche Zahl $Q = p_1 * p_2 * ... * p_n$. Da dies nicht der Wahrheit entspricht, kann auch keine grösste Primzahl existieren.

Satz 3 *Ist n eine zusammengesetzte natürliche Zahl, dann sind die Primteiler von n kleiner oder gleich* $\sqrt{n}$ *(Rosen, 2018).*

Dieser Satz ist wichtig für die Generierung von Primzahlen und einfache Primzahlentests. Wird geprüft, ob es sich bei einer Zahl um eine zusammenhängende Zahl handelt, müssen die Primteiler nur bis $\sqrt{n}$ geprüft werden. Beispielsweise müssen bei der Zahl $100 = 2^2 * 5^2$ die Primteiler nur bis $\sqrt{100} = 10$ geprüft werden. Dadurch lassen sich viele Algorithmen frühzeitig abbrechen.

3.2 Generieren von Primzahlen

Bis zum jetzigen Zeitpunkt existiert noch keine geschlossene Form, um Primzahlen zu generieren (Rosen, 2018). Daher werden zwei Strategien eingesetzt um Primzahlen zu finden:

Strategie 1: Sieb des Erathostenes
Bei dieser Methode werden alle natürlichen Zahlen aufgeführt. Danach werden alle Vielfachen dieser Zahl durchgestrichen. Als Abbruchbedingung dieses Algorithmus wird gemäss Satz 3 die Quadratwurzel der grössten Zahl in der Liste verwendet. Dieser Algorithmus kann wie folgt formalisiert werden:

Algorithm 1: Sieb des Erathostenes (Rosen, 2018)

Input Eine natürliche Zahl $n > 1$;
Output Alle Primzahlen von 2 bis n;
while $i < \sqrt{n}$ **do**
  Füge i zur Liste der Primzahlen und streiche alle vielvachen von i;
end

Dieser Algorithmus ist sehr gut zum Aufspüren von kleinen Primzahlen. Jedoch für die Suche von grossen Primzahlen, welche in der Kryptografie verwendet werden, ungeeignet. Daher wird auf der Suche nach grossen Primzahlen vor allem auf Klassen von Zahlen zurückgegriffen, in welchen Primzahlen häufig vorkommen. Die Wichtigste dieser Klassen sind die Mersenne-Zahlen.

Strategie 2: Mersenne-Zahlen

Definition 2 *Eine Zahl* $2^n - 1$ *wird Mersenne-Zahl genannt. Eine Mersenne Zahl ist höchstens dann eine Primzahl, wenn n eine Primzahl ist (Forster, 2015).*

Zahlen aus der Klasse der Mersenne-Primzahlen brechen regelmässig Rekorde auf der Suche nach der grössten Primzahl. Jedoch ist nicht jede Zahl, welche die Bedingungen gemäss Definition 2 erfüllt, eine Primzahl. Um festzustellen, ob es sich bei einer Mersenne-Zahl um eine Primzahl handelt, werden Primzahlentests benötigt.

4 Primzahlentests

Festzustellen, ob es sich bei einer gegebenen Zahl um eine Primzahl handelt, ist eines der zentralen Probleme der Kryptografie. Dabei steht nicht nur die Zuverlässigkeit der Verfahren im Vordergrund, sondern auch die Geschwindigkeit der Algorithmen. Kein Benutzer möchte mehrere Minuten warten, bis eine sichere Verbindung aufgebaut ist.

4.1 Einfache Algorithmen

Das einfachste Verfahren wäre, alle Primzahlen in einer Liste zu speichern und eine neue Zahl mit dieser Liste zu vergleichen. So könnte sehr schnell und mit absoluter Sicherheit festgestellt werden, ob es sich bei dieser Zahl um eine Primzahl handelt oder nicht. Bei genauer Betrachtung scheint dieses Verfahren jedoch nicht praktikabel, da mit Satz 2 aufgezeigt werden konnte, dass es unendlich viele Primzahlen gibt. Aus diesem Grund ist es nicht möglich ein solches Verfahren auf einem Computer zu implementieren.

Vielversprechender erscheint hingegen die Probedivision (Forster, 2015). Bei diesem Verfahren werden ähnlich wie beim Sieb des Eratosthenes alle möglichen Zahlen bis $\sqrt{n}$ durchprobiert, auf der Suche nach einem Teiler. Dieses Verfahren kommt in der Zahlentheorie mangels besserer Alternativen auch wirklich zum Einsatz, ist aber in der Praxis für die Kryptografie zu langsam (Forster, 2015).

4.2 Probabilistische Tests

Bei der Betrachtung der einfachen Verfahren für die Primzahlentests konnte festgestellt werden, dass diese für die Anwendungen der Kryptografie zu langsam oder zu speicherintensiv wären. Daher setzt man in diesen Anwendungen auf die probabilistischen Primzahltests, welche zu jeder untersuchten Zahl eine Wahrscheinlichkeit angeben, mit welcher es sich bei der getesteten Zahl um eine Primzahl handelt. Diese Verfahren liefern zwar keine akkuraten Primzahlen, jedoch reichen die hohen Wahrscheinlichkeiten für die Anwen-

dungen der Kryptografie aus.
Alle probabilistischen Tests basieren auf dem kleinen Satz von Fermat:

Satz 4 *Wenn p eine Primzahl ist und a eine natürliche Zahl, welche sich nicht durch p teilen lässt, dann gilt:*
$a^{p-1} \equiv 1(modp)$ *(Rosen, 2018).*

Der fermatsche Primzahlentest läuft wie folgt ab: Für eine gegebene Zahl n werden verschiedene Basen a durchprobiert. Wird eine Basis a gefunden, zu der n nicht teilerfremd ist, handelt es sich bei der Zahl n um keine Primzahl. Umso mehr Basen a durchprobiert werden, umso genauere Resultate liefert der Primzahlentest.

Eine Weiterentwicklung des fermatschen Primzahlentests ist der Miller-Rabin-Test. Bei diesem Test wird die Basis a nicht mehr zufällig gewählt, sondern systematisch abgearbeitet. Daher können beispielsweise vielfache von a ausgeschlossen werden und der Test liefert bessere Resultate (Forster, 2015). Das Miller-Rabin-Verfahren wird aktuell in den meisten Programmiersprachen wie MATLAB oder Java standardmässig verwendet.

4.3 Verfahren der Zukunft

Im Jahre 2004 wurde der sogenannte AKS-Primzahlentest von mehreren indischen Mathematikern veröffentlicht. Der AKS-Primzahlentest basiert auf dem kleinen Satz von Fermat(Satz 4) und ist eine Erweiterung dieses Satzes für Polynome:

Satz 5 *Wenn n eine natürliche Zahl ist und a eine ganze Zahl, welche sich nicht durch n teilen lässt, dann ist n eine Primzahl, wenn gilt:*
$(X + a)^n \equiv X^n + a(modn)$
Dabei ist X eine Unbestimmte. (Manindra, Neeraj & Nitin, 2002).

Obwohl dieser Test theoretisch die beste Performance aufweisen würde, ist dieser für praktische Anwendungen zu langsam. Dies liegt daran, dass die Polynome aus dem Satz 5 mit einem grossen Exponenten n ebenfalls einen hohen Grad erhalten, $p(x) = a_n x^n + a_{n-1} x^{n-1} + ... + a_1 x + a_0$ mit $a_n \neq 0$ (Oliver & Sebastian, 2009). Diese Polynome sind in der Praxis sehr schwer und meist nur durch einsetzen von Zahlen zu lösen (Forster, 2015). Da in der Kryptografie jedoch gerade grosse Primzahlen n von Interesse sind, stellt dies ein Hindernis für den AKS-Primzahlentest dar.
Trotz diesen praktischen Schwierigkeiten scheint gerade dieser Algorithmus theoretisch betrachtet viel Potenzial aufzuweisen und es wird aktiv an Lösungen für eine praktische Implementierung geforscht (Folkmar, 2002).

5 Schlussfolgerungen und Ausblick

Durch immer schnellere Computer werden auch immer höhere Anforderungen an die Verschlüsselung gestellt. In der Praxis bedeutet dies, dass immer grössere Primzahlen benötigt werden. Dazu werden auch immer bessere Primzahlentests benötigt. Diese Arbeit konnte aufzeigen, dass in der Theorie bereits bessere Verfahren existieren, diese jedoch noch nicht erfolgreich in der Praxis eingesetzt werden können. In einem weiterführenen Forschungsprojekt könnte festgestellt werden, welche Methoden in der Praxis das höchste Potenzial haben, die probabilistischen Primzahlentests zu ersetzen.

Literatur

Beutelspacher, A., Neumann, H. B. & Schwarzpaul, T. (2010). *Kryptografie in theorie und praxis* (2., überarbeitete Auflage Aufl.). Vieweg + Teubner. doi: 10.1007/978-3-8348-9631-5

Folkmar, B. (2002). *Ein durchbruch für jedermann.* Zugriff auf `https://www-m3.ma.tum.de/foswiki/pub/M3/Allgemeines/FolkmarBornemannPublications/aks.pdf`

Forster, O. (2015). *Algorithmische zahlentheorie* (2., überarb. u. erw. Aufl. Aufl.). Springer Spektrum. doi: 10.1007/978-3-658-06540-9

Manindra, A., Neeraj, K. & Nitin, S. (2002). *Primes is in p.* Zugriff auf `https://www.cse.iitk.ac.in/users/manindra/algebra/primality_v6.pdf`

Oliver, B. & Sebastian, S. (2009). *Der aks-primzahltest.* Zugriff auf `http://www.math.rwth-aachen.de/~Gabriele.Nebe/Vorl/pros/AKS.pdf`

Rosen, K. H. (2018). *Discrete mathematics and its applications* (Eigth edition, international student edition Aufl.). McGraw-Hill.