„RSA: Implementation and Security“ (Sommersemester 2014)

Thomas Stadler

Lehrstuhl für Technische Informatik

Zusammenfassung Sicherheitsaspekte spielen bei Hardware-Schaltungen eine immer stärkere Rolle. Im Rahmen des Seminars werden neue Themen aus diesem Bereich anhand des Buches [1] ausgearbeitet.

1 Einleitung

Der Wunsch nach gesicherter Übermittlung von Nachrichten ist wohl so alt wie die Kommunikation selbst. In der Geschichte etablierten sich mannigfache Möglichkeiten Texte zu Ver- und Entschlüsseln. Die Grundlage jeder Verschlüsselung ist in der Mathematik zu finden und der Berechnung eines Geeigneten Schlüssels. Je Komplexer die Berechnung, desto schwieriger ist es die Nachricht wieder zu dechiffrieren. Der große Sprung in der Versicherungstechnik kam mit dem Anfang des zwanzigsten Jahrhunderts und der Computertechnologie, die es ermöglichte die Komplexität an eine Maschine weiterzugeben. Dabei ist die Art und Weise der Verschlüsselung noch immer die selbe. Ein Klartext wird durch ein bestimmtes Schema in einen nicht lesbaren Text Übersetzt werden und ebenfalls durch ein festgelegtes Schema wieder zurück-gelesen werden. Das Geheimnis dabei ist recht simpel. Die beiden Kommunikationspartner brauchen einen Schlüssel für die jeweilige Übersetzung. Seit dem Siegeszug der Computer Mitte der 1900er Jahre sind diese Schlüssel Zahlen, die in die Übersetzungs-Algorithmen eingespeist werden.

2 Schlüsseltypen

Man unterscheidet bei der Verschlüsselung zwei Kategorien. Die symmetrische und die asymmetrische Verschlüsselung. Der unterschied der Typen liegt in den Verwendeten Schlüssel.

2.1 Symmetrische Verschlüsselung

Bei dieser Verschlüsselung wird ein Schlüssel erstellt, der sowohl zur Ver- als auch zur Entschlüsselung verwendet wird. Der Algorithmus, der verwendet wird um Klartext zu chiffrieren benötigt also den gleichen Schlüssel um den Chiffretext wieder in Klartext zu Übersetzen. Ein großer Nachteil dieser Technik ist also der Schlüssel selbst. Dieser muss auch gesichert den Kommunikationspartnern übermittelt werden. Sollte der Schlüssel nach Außen gelangen ist die Verschlüsselung nichtig und kann jederzeit gelesen werden.

2.2 Asymmetrische Verschlüsselung

In diesem Typ werden zwei voneinander verschieden Schlüssel erzeugt. Ein öffentlicher und ein privater Schlüssel. mit dem öffentlichen Schlüssel kann jeder einen Text für den Inhaber des privaten Schlüssels chiffrieren. Dabei ist sichergestellt, das der verschlüsselte Text nur vom Kommunikationspartner entziffert werden kann. Selbst der Inhaber des öffentlichen Schlüssels, kann den Chiffretext mit diesem nicht mehr entschlüsseln. Trotzdem muss der private Schlüssel stets geheimgehalten werden, denn nur mit ihm können Nachrichten entschlüsselt werden. Durch die Verwendung eines gesonderten Schlüssels zur Verschlüsselung stellt diese Art ein Plus an Sicherheit dar, da nicht mehr der eine Schlüssel aus dem symmetrischen Verfahren offen verwendet werden muss um Texte zu chiffrieren. Ein weiterer Aspekt der Sicherheit ist es, dass der private Schlüssel nicht aus dem öffentlichen berechnet werden kann. Sollte also der öffentliche Schlüssel in fremde Hände gelangen ist es weiterhin nicht möglich Chiffretexte zu entschlüsseln. Der im Nachfolgenden genauer erklährte RSA Algorithmus zählt zu den asymmetrischen Verfahren.

3 RSA

Der RSA Algorithmus ist sehr bekannt und weit verbreitet. Doch trotz seines Bekanntheitsgrades gilt diese Verschlüsselung als sehr sicher. Ein Grund für die Verbreitung des Verfahrens ist wohl die einfache mathematische Herleitung des Algorithmus. In diesem Kapitel wird auf die Herleitung und die hardwarespezifische Umsetzung eingegangen.

3.1 mathematische Herleitung

Der erste Schritt der Berechnung ist die Wahl von zwei voneinander verschiedenen, großen und zufälligen Primzahlen, im Weiteren werden diese als P und Q bezeichnet. Die beiden Primzahlen werden verwendet um zwei weitere Zahlen für die Rechnung zu ermitteln, $N = P \cdot Q$ und $\psi = (P-1) \cdot (Q-1)$. Als nächstes wird eine Zahl E ermittelt für die gilt: $1 < E < \psi$ und Eist teilerfremd zu ψ. Durch E und ψ wird eine weitere Zahl berechnet, für die gilt: $D \cdot E = 1 mod \psi$. Nach der Ermittlung aller benötigten Zahlen, N E und D, können die Schlüssel erzeugt werden. Dabei werden die Zahlen N und E für den öffentlichen Schlüssel verwendet, N und D für den privaten.
Um nun einen Klartext in einen Chiffretext zu überführen muss die Nachricht in eine natürliche Zahl I umgewandelt werden. Unter Verwendung von E und N kann die Umrechnung wie folgt ausgeführt werden:
$C = I^E mod N$
Das Dechiffrieren funktioniert sehr ähnlich zum Verschlüsseln. Mit den Zahlen N und D des privaten Schlüssels wird die Nachricht folgendermaßen zurückgerechnet:
$I = C^D mod N$

Interessanterweise gestaltet sich die mathematische Herleitung des Algorithmus sehr einfach und die Ver- und Entschlüsselung können mit jeweils einem mathematischen Ausdruck ermittelt werden. Um die Schwierigkeiten der Implementation des RSA zu verstehen, müssen die Herleitungsschritte genauer betrachtet werden.

Sicherheit der Herleitung Schon der erste Schritt der Herleitung entpuppt sich als der vielleicht schwierigste und zugleich wichtigste. Aus zwei großen, voneinander verschiedenen und zufälligen Primzahlen wird durch Multiplikation die Zahl N erzeugt, die sowohl teil des privaten als auch des öffentlichen Schlüssels ist. Sollte der öffentliche Schlüssel eines Kommunikationspartners bekannt werden, wäre es für einen Angreifer ein Leichtes alle anderen Zahlen zu ermitteln und den Algorithmus zu knacken, sofern er die beiden Primzahlen errechnen kann. Doch hier zeigt eine Barriere, die Teil des Fundamentalsatzes der Arithmetik is: Es existiert kein Verfahren das die Primfaktorzerlegung einer Zahl in polynomialer Zeit errechnen kann.
Jedoch kann man mit modernen Computern einfach alle Möglichkeiten der Faktorisierung durchprobieren. Um dieser einfachen Methode entgegenzuwirken, müssen die Zahlen P und Q sehr groß gewählt werden. Damit kann sichergestellt werden, dass es unrealistisch ist, dass ein Brute-Force Anschlag in ab-wartbarer Zeit zum Erfolg führt. Zur gegenwärtiger Zeit ist eine Größenordnung von 1024 oder 2048 Bit für die Primzahlen gängig.

Komplexität der Herleitung Test für Git

Literatur

1. Tehranipoor, M., Wang, C.: Introduction to Hardware Security and Trust. Springer (2012)