Kurzdarstellung

Aufgrund der allgegenwärtigen Präsenz von eingebetteten Systemen, z.B. basierend auf Sensornetzwerken und RF-ID-Tag, ist wieder eine zunehmende Nachfrage nach kostengünstigen Geräten, wie z.B. 8-Bit Mikroprozessoren, zu beobachten. Dabei stellt die kryptographische Sicherheit dieser Geräte eine größe Hürde für ihre breite Akzeptanz dar. Asymmetrische Kryptosysteme (engl. public-key cryptosystems, PKC) wie RSA und DSA sind nicht für den Einsatz auf solchen beschränkten, eingebetteten Geräten geeignet, weil sie im Allgemeinen für arithmetische Operationen Langzahlen (1024-2048 Bit Operanden) verwenden.

Die Elliptische Kurven Kryptographie (engl. Elliptic Curve Cryptography, ECC) hat sich als geeignete Alternative für eingebettete Systeme herausgestellt, weil sie mit kleinen Schlüssellängen, kleineren Operanden und vergleichsweise geringen arithmetischen Anforderungen auskommt. Die Implementierung eines standardisierten ECC Algorithmus auf 8-Bit Prozessoren stellt indes immer noch eine große Herausforderung dar, die als nicht praktikabel für rechen- und speicherbeschränkte Geräte angesehen wird.

In dieser Dissertation wird die Umsetzbarkeit von asymmetrischen Verfahren, insbesondere der Elliptische Kurven Kryptographie, auf Geräten mit beschränkten Resourcen behandelt. Die Elliptische Kurven Kryptographie ermöglicht ein großes Maß an Flexibilität aufgrund möglicher Freiheitsgrade bzgl. der Wahl verschiedenener Parameter und Algorithmen, welche in dieser Arbeit diskutiert und effizient implementiert werden. So wird zuerst gezeigt, dass es möglich ist, einen sicheren Schlüsselaustausch basierend auf ECC auf einem kostengünstigen, für drahtlose Anwendungen ausgelegten Prozessor (vergleichbar mit dem dem weit verbreiteten 8-Bit 8051 Mikroprozessor) zu implementieren. Diese gänzlich auf Software basierende Implementierung des Diffie-Hellman Protokolls mit Elliptischen Kurven (Elliptic Curve Diffie-Hellman, ECDH) führt arithmetische Berechnungen in einem optimalen 131-Bit Erweiterungskörper (optimal extension field, OEF) durch. Eine kryptografisch sichere Verbindung zwischen zwei Endteilnehmern wird auf einem solchen Gerät ohne kryptographischen Co-Prozessor innerhalb von drei Sekunden hergestellt.

Desweiteren untersuchen wir die Möglichkeiten von Software/Hardware Co-Design Ansätzen um mittels Architekturmodifikationen, z.B. Befehlssatzerweiterungen (Instruction Set Extensions, ISE) für körperarithmetische Basisoperationen wie sie bei ECC zum Einsatz kommen, die Performanz zu steigern. Es wird gezeigt, dass eine standardisierte 163-Bit Punkt-Multiplikation mit minimalen zusätzlichen Hardware-

Kosten auf einem 8-Bit AVR Mikro-Controller (ein typischer, kostengünstiger Prozessor), der mit 4 MHz getaktet ist, in 0,113 Sekunden ausgeführt werden kann. Dieses Design bringt im Vergleich zu einer rein Software-basierten Implementierung einen Geschwindigkeitsgewinn um mehr als das 30-fache, während die Größe des Quelltext verringert und weniger Arbeitspeicher verbraucht wird. Zusätzlich werden zwei neue Befehle für den MIPS 32-Bit Prozessor vorgeschlagen, die Reduktionen modulo Pseudo-Mersenne Primzahlen beschleunigen. Desweiteren wird gezeigt, dass für Multiplikationen in einem OEF ein vergrösserter Akkumulator in der ALU von Vorteil ist. Die vorgestellte Architektur führt zu einem Geschwindigkeitszuwachs um 180

Darüber hinaus werden architektonische Verbesserungen sowie optimale Parameter für Least Significant Digit (LSD) Multiplizierer für Binärkörper vorgestellt. Die architektonischen Verbesserungen basieren auf einem Double Accumulator Multiplier (DAM) und N-Accumulator Multiplier (NAM), welche beide klassische LSD Multiplizierer bzgl. der Geschwindigkeit übertreffen.

Im Anschluß wird eine effiziente ECC-Prozessorarchitektur (für 169-bit, 289-bit und 361-bit OEF) vorgestellt, die alle arithmetischen Operationen im Frequenzbereich durchführt. So wird eine optimierte 169-Bit OEF ECC Implementierung mit 24K Logikgattern für einen 0.35um CMOS Prozess präsentiert.

Schließlich wird eine flächenoptimierte ECC ASIC Implementierung für Binärköper mit standardisierte 133 bis 193 Bits vorgestellt. Es wird gezeigt, dass lediglich 10K bis 18K Logikgatter für eine 0.35um CMOS Implementierung benötigt werden. Daher eignet sich diese ECC Architektur insbesondere für kostengünstige Implementierungen, wie sie z.B. in drahtlosen Netzwerken zum Einsatz kommen.