ingenieur
wissenschaften
htw saar

Hochschule für
Technik und Wirtschaft
des Saarlandes
University of
Applied Sciences

Projektbericht-Thesis

zur Erlangung des akademischen Grades
Projektarbeit
an der Hochschule für Technik und Wirtschaft des Saarlandes
im Studiengang Praktische Informatik
der Fakultät für Ingenieurwissenschaften

Implementierung elliptischer Kurven für die Kryptographie

vorgelegt von
Annick Aboa

betreut und begutachtet von
Prof. Dr. Peter Birkner
Prof. Dr. Thomas Kretschmer

Saarbrücken, 05. Februar 2021

Inhaltsverzeichnis

1 Einleitung

1.1 Motivation

In den letzten Jahren hat die dramatische Zunahme von elektronisch übertragenen Informationen zu einer zunehmenden Abhängigkeit von kryptographischen Verfahren geführt. In unserer modernen vernetzten Welt ermöglicht Kryptographie es Menschen, nicht nur geheimen Nachrichten über öffentlichen Kanälen auszutauschen, sondern auch Online-Banking, Online-Handel und Online-Einkäufe zu tätigen, ohne befürchten zu müssen, dass ihre persönlichen Informationen kompromittiert [5].

Daher ist unter dem Begriff **Kryptographie**, die Lehre mathematischer Techniken in Bezug auf Aspekte der Informationssicherheit wie Vertraulichkeit, Datenintegrität, Datensauthentifizierung, zu verstehen [3]. Die Dringlichkeit eines sicheren Austauschs digitaler Daten zu gewähren, hat daher in den letzten Jahren zu großen Mengen unterschiedlicher Verschlüsselungsverfahren geführt. Diese können in zwei Gruppen eingeteilt werden nähmlich: symmetrische (mit privaten Schlüsselalgorithmen) und asymmetrische Verschlüsselungsverfahren (mit öffentlichen Schlüsselalgorithmen) [3].

In dieser Arbeit wird der Fokus hauptsächlich auf eine asymmetrische Verschlüsselungstechnik liegen: die **Elliptische-Kurven-Kryptographie** (engl. **Elliptic Curve Cryptography: ECC**) aufgrund ihrer zahlreichen Vorteile gegenüber herkömmlichen kryptographischen Algorithmen. Gemäß den Richtlinien des Nationalen Instituts für Standards und Technologie (NIST) kann eine ECC-Schlüsselgröße von 163 Bit eine gleichwertige bzw. höhere Sicherheit wie ein 1024-Bit des RSA-Algorithmus gewährleisten. Mit ECC-Schlüsselgroße ist nur eine geringere Rechenleistung, sowie ein geringerer Speicher- und Stromverbrauch erforderlich ([10]; [31]). Zudem kann die Technologie in Verbindung mit den meisten Verschlüsselungsmethoden mit öffentlichen Schlüsseln wie RSA und Diffie-Hellman verwendet werden. ECC ist ideal für den Einsatz in eingeschränkten Umgebungen wie Personal Digital Assistenten, Mobiltelefone und Smartcards.

Im Allgemeinen lässt sich die Elliptische-Kurven-Kryptographie als ein Public-Key- bzw. ein asymmetrisches Verschlüssselungsverfahren definiert, das auf der elliptischen Kurventheorie basiert und zur Erstellung schnellerer, kleinerer und effizienterer kryptografischer Schlüssel verwendet werden kann. Im asymmetrischen Verfahren mit öffentlichen Schlüsseln verfügt jeder Benutzer oder das Gerät, das an der Kommunikation teilnimmt, über ein Schlüsselpaar: einen öffentlichen Schlüssel und einen privaten Schlüssel sowie eine Reihe von Operationen, die den Schlüsseln zugeordnet sind, um kryptographische Operationen wie die Verschlüsselung einer Nachricht auszuführen. Nur der bestimmte Benutzer kennt den privaten Schlüssel, während der öffentliche Schlüssel an alle an der Kommunikation beteiligten Benutzer verteilt wird. Zudem können die Daten, die mit öffentlichen Schlüsseln verschlüsselt sind, nur mit dem privaten Schlüssel entschlüsselt werden ([10]; [13]).

1.2 Problematik

Da ECC dazu beiträgt, eine gleichwertige Sicherheit bei geringerer Rechenleistung und geringerem Ressourcenverbrauch zu erreichen, hat sich ECC zu einem attraktiven und

sehr effizienten Public-Key-Krytosystem entwickelt [31]. Ihre Sicherheit basiert jedoch auf die Komplexität, das diskrete Logarithmusproblem in der Gruppe von Punkten auf einer elliptischen Kurve zu berechnen [11], da dieses Problem in nur exponentieller Zeit gelöst werden kann. Außerdem ist auch die Art der zu verwendeten elliptischen Kurven unter Betrachtung der verwendeten Parameter (z.B. den Koeffizienten der Kurve) optimal auszuwählen [16]. Es existiert bereits mehrere Kurven die vom amerikanischen Standardinstitut NIST festgelegt wurden, obwohl deren Erzeugung allerdings nicht vollständig nachvollziehbar ist, was in amerikanischen Krypto-Standards zu erheblicher Kritik geführt hat [1]. Zudem gibt es eine erhebliche Anzahl potenzieller Schwachstellen für elliptische Kurven, wie z. B. Seitenkanalangriffe und Twist-Security-Angriffe, die bedrohen, die Sicherheit von angebotenen ECC privaten Schlüsseln, ungültig zu machen [33]. Also unabhängig davon, wie sicher ECC theoretisch ist, muss der Algorithmus ordnungsgemäß implementiert werden, da fehlgeschlagene Implementierung von ECC-Algorithmen zu erheblichen Sicherheitslücken in der kryptografischen Software führen können [33].

1.3 Zielsetzung

Ziel dieser Projektarbeit ist es einen Überblick über elliptischen Kurven in der Kryptographie zu geben. Zudem es werden basierend auf elliptische Modular- und Kurvenarithmetik, eine Implementierung von elliptischen Kurven für die Kryptographie bereitgestellt, die dann zur Erzeugung von Schlüsseln eines ECC-basierten Kryptosystems, das Diffie-Hellman-Kryptosystems verwendet wird. Es wurde in dieser Arbeit Java als bevorzugte Sprache für die Implementierung von elliptischen Kurven für die Kryptographie gewählt, weil die Verwendung von Java bei der Entwicklung einer Vielzahl von Internetanwendungen diese zur geeigneten Sprache für das Schreiben dieser Programme macht. Allerdings können die meisten Programme für Umgebungen mit eingeschränkten Ressourcen erweitert und optimiert werden.

1.4 Projektaufbau

Die vorliegende Arbeit ist wie folgt aufgebaut: Nach diesem einleitenden Abschnitt, gibt der zweite Abschnitt einen Überblick über das Thema Im Abschnitt 2 bietet eine Einfuhrung in elliptische Kurven und deren Arithmetik. //TODO

2 Grundlagen von elliptischen Kurven

2.1 Begriffsabgrenzung

Nach Dietmer ist unter dem Begriff **Kryptographie** „die Wissenschaft von geheimen Schreiben zu verstehen." Grundkonzept eines kryptografischen Systems ist also die Verschlüsselung von Informationen bzw. von Daten, um die Vertraulichkeit der Informationen zu gewährleisten [36].

Daten, die über einen unsicheren Kanal wie das Internet übertragen werden, werden mit Hilfe moderner Kryptosysteme so verschlüsselt, dass Unbefugte in einem Szenario, in dem sie auf die Informationen zugegriffen haben, nicht verstehen, wonach sie suchen[21]. Der unverschlüsselte Information wird als *Klartext* (Plaintext, Cleartext), der verschlüsselte als *Chiffretext* (Ciphertext, Cryptotext) bezeichnet und der Verschlüsselungsprozess des Klartextes wird *Chiffrieren* (engl. Encryption) gennant. Umgekerht wird unter *Dechriffrieren* (engl. Decryption), den Entschlüsselungsprozess eines Chiffretextes, verstanden.

Im Allgemeinen werden die beiden Prozesse durch eine Reihe von Regeln erreicht, sogennante Verschlüsselungs- und Entschlüsselungsalgorithmen. Der Verschlüsselungsprozess basiert auf einem Schlüssel , der dann zusammen mit den Informationen als Eingabe an einen Verschlüsselungsalgorithmus übergeben wird. Danach können unter Verwendung eines Entschlüsselungsalgorithmus die Informationen mit dem entsprechenden Entschlüsselungsschlüssels abgerufen werden. Wer einen geheimen Schlüssel besitzt, kann die Informationen in Klartext entschlüsseln[21]. Die folgende Abbildung 2.1 verdeutlicht die Zusammenhänge.

In den 70er und 80er Jahren war die Kryptographie vor allem auf dem militären und diplomatischen Sektor beschränkt. Um geheime Nachrichten zu verschlüsseln, wurden sogenannte symmetrische Verschlüsselungsverfahren (z.B. Caesar-Verschlüsselung ([21])), wo nur ein gemeinsamer geheimer Schlüssel sowohl für die Ver- als auch für die Entschlüsselung benötigt wird, verwendet.

Aus diesem Grund ist es bei der symmetrischen Verschlüsselung sehr wichtig, dass der geheime Schlüssel auf einem sicheren Übertragungsweg an den Empfänger weitervermittelt wird, bevor die verschlüsselten Nachrichten übermittelt werden können. Früher wurde der Schlüssel meist persönlich, in Form eines Botens, übergeben. Da das persönliche Übergeben des Schlüssels sehrumständlich ist und es besteht das Risiko, dass der Schlüssel belauscht oder gestohlen werden konnte, wurden weitere Verschlüsslungsverfahren vorgeschlagen und zwar asymmetrisches Verschlüsselungsverfahren (auch Public-Key-

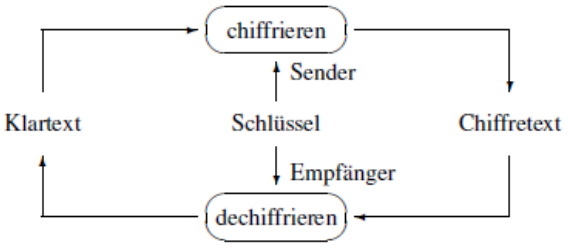

Abbildung 2.1: Grundkonzept der Kryptographie

Verfahren gennant) [5].

Im Gegensatz zu einem symmetrischen Verschlüsselungsverfahren erfordern asymmetrische Verschlüsselungsverfahren, nicht nur einen Schlüssel, sondern ein Schlüsselpaar bestehend aus einem öffentlichen Schlüssel und einem privaten Schlüssel [36]. Die Kommunikation erfolgt hier ohne vorhergehenden Schlüsselaustausch und ist jedoch viel langsamer als die Kryptografie mit privaten Schlüsseln. Mit dem privaten Schlüssel werden Daten entschlüsselt oder digitale Signaturen erzeugt, während mit dem öffentlichen Schlüssel Daten verschlüsselt und die Authentizität von erzeugten Signaturen überprüft werden([29]; [36]). Die Abbildung 2.2 veranschaulicht diese Schlüssel.

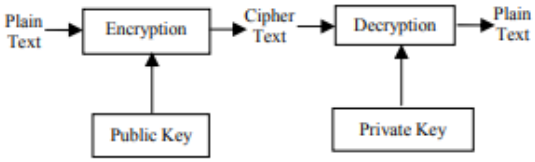

Abbildung 2.2: Public-Key Verschlüsselung

Das erste asymmetrische Kryptosystem, Rivest-Shamir-Adleman-Verfahren (RSA-Verfahren) wurde im Jahr 1977 von Ronald Rivest, Adi Shamir und Leonard Adleman gefunden, danach wurden weitere Kryptosysteme wie Rabin, Elgamal, Diffie Hellman Schlüsselaustausch (DH) und Elliptische Kurven-Kryptogrphie vorgestellt. Für unsere Arbeit liegt jedoch der Schwerpunkt auf der Kryptographie mit elliptischen Kurven und die Erzeugung von Schlüsseln, die für das Diffie-Hellman-Verfahren notwendig sind.

Die elliptische Kurvenkryptographie (ECC) ist Verschlüsselungstechnik mit öffentlichem Schlüssel, die auf der algebraischen Struktur elliptischer Kurven über endlichen Körper basiert [19] und zur Erstellung schnellerer, kleinerer und effizienterer kryptografischer Schlüssel verwendet werden kann [6].

Die Verwendung einer elliptischen Kurve in der Kryptographie wurde im Jahr 1985 unabhängig voneinander von Miller [34] und Koblitz [23] vorgeschlagen. In den späten 1990er Jahren wurde ECC von einer Reihe von Organisationen wie ANSI [25], IEEE [32], ISO [14], NIST [2] standardisiert und erhielt kommerzielle Akzeptanz [12]. Das bekannteste Verschlüsselungsschema ist das Elliptic Curve Integrated Encryption Scheme (ECIES), das in IEEE- und auch in SECG SEC 1-Standards enthalten ist[27]. Beispiele für die Anwendung von ECC sind unter anderen Mehrzweck-Smartcards, die neuen deutschen Personaldokumente[16],

Im Allgemeinen basieren Kryptosysteme mit öffentlichem Schlüssel auf die Schwierigkeit der Lösung bestimmten mathematischen Probleme, z.B. beruht RSA-Verfahren auf dem Integer Factorizierungsproblem und DH sowie ECC basieren auf dem Diskreten Logarithmus-Problem. Es besteht jedoch ein Problem bei herkömmlichen Kryptosystemen mit öffentlichem Schlüsseln wie RSA.

Das Hauptproblem besteht darin, dass die Schlüsselgröße ausreichend groß sein muss, um die Sicherheitsanforderungen auf hohem Niveau zu erfüllen, was zu einer geringeren Geschwindigkeit und einem höheren Bandbreitenverbrauch führt. Dies ist nicht den Fall, wenn elliptische Kurven (EC) für das Public Key Verfahren eingesetzt sind, da ECC im Vergleich zu RSA für den Benutzer eine gleichwertige Sicherheit bei kleineren Schlüsselgrößen bietet und für Angreifer schwerer exponentieller Zeitherausforderung, um in das System einzudringen[12].

Laut einigen Forschern kann ECC mit einem 164-Bit-Schlüssel ein Sicherheitsniveau erreichen, für dessen Erreichung andere Systeme einen 1.024-Bit-Schlüssel benötigen[6].

Es stellte sich heraus, dass ECC das effizienteste Kryptosystem mit öffentlichem Schlüssel ist [22]. Der Grund dafür ist, dass nach Miller und Koblitz das Diskreter-Logarithmus-Problem für elliptische Kurven schwerer sei als das klassische Diskreter-Logarithmus-Problem ist und zudem auch schnellere Laufzeiten ermögliche. Bevor das Diskreter-Logarithmus-Problem vorgestellt wird, ist es wichtig zu definieren, was unter einer elliptischen Kurven zu verstehen ist.

Im Allgemeinen ist eine elliptische Kurve eine projektive, algebraische Kurve **C(K)**, auf der sich ein bestimmter Punkt O befindet, der als Punkt unendlich oder Nullpunkt bezeichnet wird [12].

Eine elliptische Kurve E über ein Körper K der Charakteristik $\neq 2$ oder 3, lässt sich formal definieren als die Menge der Punkte $(x, y) \in K$ der Weierstraß-Gleichung:

$$y^2 = x^3 + ax + b \quad \text{mit a, b} \in K \text{ [23].} \tag{2.1}$$

Elliptische Kurven in Form von Gleichungen können in singuläre und nicht-singuläre Gruppen unterteilt werden. In der ECC werden nicht-singuläre Kurven bevorzugt, damit eine Kurve frei von Spitzen oder Selbstüberschneidungen sein kann[13].

Eine elliptische Kurve wird als *nicht-singulär* bezeichnet, wenn sie keinen Punkt $P = (x, y)$ enthält, an dem ein mathematisches Objekt nicht definiert ist [5] und für die Werte a und b folgende Bedingung $\Delta = 4a + 27b \neq 0$ erfüllt, um eine endliche Abelsche Gruppe zu bilden.

In der abstrakten Algebra ist eine abelsche Gruppe, eine Gruppe, in der das Ergebnis der Anwendung der Gruppenoperation auf zwei Gruppenelemente nicht von ihrer Reihenfolge abhängt.

Die **Charakteristik** eines Körpers **K** ist die kleinste positive Zahl **p** mit

$$\underbrace{1 \cdot 1 \cdot 1 \cdot \ldots \cdot 1}_{k \text{ mal}} = 0 \quad \text{, falls sie existiert.}$$

Dies ist beispielsweise für die Körper der rationalen Zahlen $\mathbb{Q}$, der reelen Zahlen $\mathbb{R}$ und der komplexen Zahlen $\mathbb{C}$ der Fall.Für jeden endlichen Körper ist die Charakteristik immer eine Primzahl [36].

K kann ein beliebiger Körper, also etwa $\mathbb{R}$, $\mathbb{Q}$, $\mathbb{C}$ oder ein endlicher Körper $\mathbb{F}$ sein[5]. Die obige Gleichung entspricht die Definition einer elliptischen Kurve über reelen Zahlen. Obwohl eine elliptische Kurve über den reellen Zahlen ein guter Ansatz ist, um die Eigenschaften einer elliptischen Kurve zu verstehen, erfordert sie eine höhere Rechenzeit, um verschiedene Operationen auszuführen, und ist manchmal aufgrund von Rundungsfehlern ungenau. Kryptographie-Schemata erfordern jedoch eine schnelle und präzise Arithmetik[13]. Folglich werden in kryptografischen Anwendungen zwei Arten von elliptischen Kurven verwendet:

- Primzahlen über einem Feld $\mathbf{Z_p}$, wobei p eine Primzahl und p > 3 ist. Alle Variablen und Koeffizienten werden aus einer Menge von ganzen Zahlen von 0 bis p - 1 entnommen und Berechnungen werden über Modulo p durchgeführt [5].
- Binäre Kurve über dem Galois-Feld 2^m, auch bekannt als $GF(2^m)$, wobei alle Variablen und Koeffizienten in GF sind und Berechnungen über $GF(2^m)$ durchgeführt werden.

Da Primkurven analog zur Binärkurve keine erweiterte Bit-Fidding-Operation haben, sind sie für die Software-Implementierung geeignet[13].

In dieser Arbeit wurde die Implementierung von elliptischen Kurven über endliche Körper mit $\mathbf{K} = \mathbf{Z_p}$ berücksichtigt. Dazu wird mehr im Kapitel 4 erläutert. Eine elliptische

Kurve über dem endlichen Feld $\mathbf{Z_p}$ enthält alle Punkte (x, y) in der $\mathbf{Z_p} \times \mathbf{Z_p}$ Matrix, die die folgende elliptische Kurvengleichung erfüllt:

$$y^2 = x^3 + ax + b \quad (mod p) \tag{2.2}$$

Dabei sind x und y Zahlen in $\mathbf{Z_p}$ und ähnlich wie im realen Fall ist $\Delta \neq 0$. Alle Punkte (x, y), die die obige Gleichung erfüllen, liegen auch auf der elliptischen Kurve. Der öffentliche Schlüssel ist ein Punkt in der Kurve und der private Schlüssel ist eine Zufallszahl. Das Hinzufügen von Punkten auf der elliptischen Kurve ist jedoch kein einfacher Prozess, sondern an ein auf polynomiale Zeit zu lösendem Problem gebunden [20]: Das diskrete Logarithmusproblem.

2.2 Diskretes Logarithmusproblem

Nehmen wir nun an, dass (G, $\times$) eine multiplikative zyklische Gruppe mit Domainparametern g, h und n ist. Das **diskrete Logarithmusproblem** ist explizit definiert, als das Problem der Bestimmung einer eindeutigen Ganzzahl x, die zufällig aus dem Intervall [1, p - 1] ausgewählt wird, so dass es gilt:

$$g^x = h mod p$$

vorausgesetzt, dass eine solche ganze Zahl existiert. Der Parameter g ist die Basis des Logarithmus bzw. ein Erzeuger der Gruppe **G** ; n die Anzahl der Elemente in **G** ; der private Schlüssel bzw. das diskrete Logarithmusproblem von h zur Basis g ist die Ganzzahl x, und der öffentliche Schlüssel ist $h = g^x$ [8].

Das **Elliptic Curve Diskrete Logarithmus Problem (ECDLP)** ist ähnlich definiert, aber betrachtet die Weierstraß-Gleichung für eine elliptische Kurve $\mathbf{E} : y^2 = x^3 + ax + b$ über $\mathbb{Z}$ und zwei Punkte P und $Q \in \mathbf{F_p}$. Zu bestimmen ist, eine Zahl k $\in \mathbf{Z}$ mit

$$Q = kP \quad \text{, falls so ein k existiert.}$$

Die Primzahl p, die Gleichung der elliptischen Kurve E und der Punkt P und seine Ordnung n sind die Domainparameter. Der privater Schlüssel ist die ganze Zahl k, die gleichmäßig zufällig aus dem Intervall [1, n - 1] ausgewählt wird, und der entsprechende öffentliche Schlüssel ist $Q = kP$.

Daher ist die Hauptoperation bei der ECC die Punktmultiplikation, also die Multiplikation eines Skalars k mit einem beliebigen Punkt P auf der Kurve, um einen anderen Punkt Q auf der Kurve zu erhalten.

Das Lösen von ECDLP ist viel schwieriger als DLP, da die Komplexität der Punktarithmetik in ECDLP im Vergleich zur Ganzzahlarithmetik in DLP zunimmt. Aus diesem Grund ist ECC in der Lage, ein ähnliches Sicherheitsniveau wie RSA bereitzustellen, jedoch mit einer kürzeren Schlüsselgröße, was es wiederum zur beliebtesten Wahl macht[20].

Damit ein auf $\mathbf{F_p}$ basierendes diskretes Logarithmus-System effizient ist, sollten schnelle Algorithmen zur Berechnung der Gruppenoperation bekannt sein. Aus Sicherheitsgründen sollte das Problem des diskreten Logarithmus in $\mathbf{Z_p}$ unlösbar sein[8]. Es gibt viele Algorithmen zur Lösung von **ECDLP**, aber der erfolgreichste ist die Kombination von Pollards Rho- und Pohlig-Hellman-Angriffen mit vollständig exponentieller Laufzeit ([20];[8]). Angriffe resultieren normalerweise aus den Schwächen bei der Auswahl der elliptischen Kurve und des endlichen Feldes (siehe Kapitel 4 und 5), aber die meisten Angriffe können durch korrekte Auswahl der Parameter der elliptischen Kurve vereitelt werden [20]. Daher sollten diese öffentlichen Parameter sicher ausgewählt werden, um alle erkannten Angriffe zu vermeiden[4]. Der nächste Abschnitt beschäftigt sich näher mit diesen Parametern.

2.3 Domänenparameter für elliptischen Kurven

Vor der Implementierung eines ECC-Systems müssen mehrere Entscheidungen getroffen werden. Dazu gehören die Auswahl von Domänenparametern für elliptische Kurven (zugrunde liegendes endliches Feld, Felddarstellung, elliptische Kurve) sowie Algorithmen für Feldarithmetik, elliptische Kurvenarithmetik und Protokollarithmetik. Die Auswahl kann durch Sicherheitsaspekte, Anwendungsplattform (Software oder Hardware), Einschränkungen der jeweiligen Computer- und Kommunikationsumgebung (z. B. Speichergröße, Bandbreite) beeinflusst werden [7].

Bevor der Verschlüsselungsprozess gestartet und der verschlüsselte Text transformiert wird, müssen Domänenparameter von beiden Parteien vereinbart werden, die an der sicheren und vertrauenswürdigen Kommunikation über ECC beteiligt sind [31]. Sie werden von einer Gruppe von Benutzern gemeinsam genutzt und können in einigen Anwendungen jedoch für jeden Benutzer spezifisch sein [17]. Es können zwei Arten von elliptischen Kurvendomänenparametern verwendet werden [28]:

- elliptische Kurvendomänenparameter über $\mathbf{Z_p}$ und
- elliptische Kurvendomänenparameter über $\mathbf{F_2^m}$.

2.3.1 Parameter über Z_p

Die Domänenparameter für die elliptische Kurve über $\mathbf{F_p}$ sind ein Sextuple

$$(p, a, b, G, n, h)$$

bestehend aus einer ganzen Zahl $\mathbf{p}$, die das endliche Feld $\mathbf{Z_p}$ spezifiziert; der Kurve aus 2.2, den Parametern a und b, einem Basispunkt $G = (x_G, y_G)$ auf $E(\mathbf{Z_p})$, $\mathbf{n}$ die Ordnung der elliptischen Kurve und dem Cofaktor h, wobei

$$h = \frac{\#E(F_p)}{n}$$

und $\#E(\mathbf{F_p})$ die Anzahl der Punkte auf einer elliptischen Kurve ist.([28]; [12])

2.3.2 Parameter über F_2^m

Die Domänenparameter für die elliptische Kurve über $\mathbf{F_2^m}$ sind ein Septuple

$$T = (m, f(x), a; b; G; n; h)$$

bestehend aus einer ganzen Zahl m, einem irreduziblen binären Polynom $f(x)$ vom Grad m, Parameter a, b $\in \mathbf{F_2^m}$ der durch die Gleichung der elliptischen Kurve $\mathbf{E}(\mathbf{F_2^m})$:

$$y^2 + xy = x^3 + ax^2 + b \quad \in F_2^m \tag{2.3}$$

definiert sind, einem Basispunkt $G = (xG; yG)$ auf $E(F_2^m)$, eine Primzahl n, die in der Größenordnung von G liegt und die Cofaktor h mit

$$h = \frac{\#E(F_p)}{n}$$

und $\#E(\mathbf{F_p})$ die Anzahl der Punkte auf einer elliptischen Kurve ist. ([28]; [12])

Um einen Angriff auf ECDLP zu vermeiden, muss $|E|$ eine ausreichend große Primzahl sein. Zumindest wird vorgeschlagen, dass $n > 2^{160}$ ist [17].

Für diese Arbeit wurden jedoch nur Parameter über $\mathbf{Z_p}$ bevorzugt. Die Verwendung der elliptischen Kurvenarithmetik macht ECC unter allen vorhandenen kryptografischen Schemata einzigartig. Je nach gewähltem Feld verwendet ECC für seine Operationen modulare Arithmetik oder Polynomarithmetik. Dies wird im nächsten Kapitel präsentiert.

3 Zahlentheoretische Grundlagen

Hoher Durchsatz und geringe Ressourcen sind in vielen Anwendungen die entscheidenden Entwurfsparameter des ECC-Prozessors [26]. Da die Effizienz von ECC-Prozessoren hauptsächlich von modularen arithmetischen Operationen wie modularer Addition, Subtraktion und Multiplikation abhängt, ist der effiziente Entwurf modularer Arithmetik eine sehr anspruchsvolle Aufgabe für die Implementierung eines Hochleistungs-ECC-Prozessors [26]. ist es sehr wichtig Grundlagen der Zahlentheorie, zu verstehen. In diesem Kapitel werden die wichtigsten Arithmetikoperationen über dem Primfeld $\mathbf{Z_p}$, die für die Kryptographie von Bedeutung sind, zusammen mit Beispielen vorgestellt und beschrieben.

3.1 Modulararithmetik

Die modulare Arithmetik ist:

> „ein Prozess zum Reduzieren einer Zahl modulo einer anderen Zahl, wobei dieser Prozess zahlreiche Multi-Präzisions-Gleitkomma-Divisionsoperationen umfassen kann. “[9]

Die modulare Artihmetik bietet endliche Strukturen, die alle üblichen arithmetischen Operationen der ganzen Zahlen aufweisen und die mit vorhandener Computerhardware problemlos implementiert werden können [30]. Eine wichtige Eigenschaft dieser Strukturen ist, dass sie durch Operationen wie Exponentiation zufällig permutiert zu sein scheinen, aber die Permutation kann oft leicht durch eine andere Exponentiation umgekehrt werden [30]. In entsprechend ausgewählten Fällen ermöglichen diese Vorgänge die Ver- und Entschlüsselung oder die Generierung und Überprüfung von Signaturen [30]. Im Allgemeinen lässt sich eine modulare Reduktion durch die folgende Gleichung definieren: $r = xmodp = x^{\smile}\left\lfloor \frac{x}{p} \right\rfloor \cdot p$, wobei x die zu reduzierende Anzahl modulo p ist, der gefundene Rest r, der im Bereich von [0, p-1] liegt[9].

Daraus lässt sich die Äquivalenzrelation definieren und man sagt, dass r kongruent zu x modulo p ist. Eine Zahl r ist **kongruent** bzw. **äquivalent** zu einer Zahl x modulo p, ausgedrückt durch $r \equiv x(modp)$, falls

$$p|(r-x)$$

, also wenn r und x den gleichen Rest haben, wenn sie durch p geteilt werden und $r = k \cdot p + x$ mit $k \in \mathbf{Z}$.

In dieser Arbeit bildet die Klasse *BasicTheoreticMethod.java*, das Grundgerüst der Implementierung von elliptischen Kurven über das Primfeld $\mathbf{Z_p}$. Sie enthält zahlentheoretische Methoden, die eine wichtige Bedeutung für die Kryptographie haben.

Die erste wichtigste Methode ist die Methode der Berechnung vom Modulo: **X** mod **P**

3.1.1 Berechnung von X mod Y

Eine naivste Methode wurde implementiert, in dem das Finden von $r = xmodp$ zuerst durch wiederholte Berechnung des Quotients $q = \frac{x}{p}$ und dann durch wiederholtes Subtrahieren von x mit dem Ergebnis aus der Multiplikation von p mit q, bis das Ergebnis im

Bereich von $[0, p-1]$ liegt. Die Leistung der Modulo-Operation hängt vor allem hier von der Leistung des Divisionsalgorithmus ab.

Die naivste Art eine ganzzahlige Division zu implementieren, besteht darin, solange den Divisor p von der Dividende x zu subtrahieren, bis die Dividende x negativ wird, und dabei den Quotienten q hochzuzählen. Wenn die Dividende x negativ wird, wird die Dividende vom Divisor addiert und der Quotient um eins verringert[15].

Die implementierte Methode ist jedoch sehr langsam und kostspielig, wenn die Anzahl der Iterationen berücksichtigt wird, wenn eine große Zahl durch eine kleine Zahl geteilt wird. [24]

Aber Methoden wie die **Barrett-Reduktion** können verwendet werden, um die Modulo-Operation zu optimieren.([18], [35])

Eine Barrett-Reduktion ist ein Verfahren zum Reduzieren einer Zahl modulo einer anderen Zahl, wobei die Division durch Multiplikationen und Verschiebungen ersetzt werden können, da die beiden letzteren Operationen viel billiger sind.[9]. Also der Quotient $q = \frac{x}{p}$ wird unter Verwendung kostengünstigerer Operationen mit Potenzen einer geeignet gewählten Basis b geschätzt. Für die Barett-Reduktion wird b häufig als Potenz von 2 gewählt werden.

Eine modulabhängige Größe $m = \left\lfloor \frac{b^{2k}}{p} \right\rfloor$ muss berechnet werden, wodurch der Algorithmus für den Fall geeignet ist, dass viele Reduktionen mit einem einzigen Modul durchgeführt werden[8].

Dies erklärt sich aus der Tatsache, dass beispielsweise jede RSA-Verschlüsselung für eine Entität ein Reduktionsmodul für den öffentlichen Schlüsselmodul dieser Entität erfordert [3]. q kann nun einmal nur für jedes Modul gleich dem Kehrwert von p berechnet werden [24].

Das Modulares Reduktonsverfahren umfasst[9]:

- die Eingabe die zu reduzierenden Wert x und des Modulos p, wobei dies eine spezielle Form (z.B. NIST-Primzahl) hat,
- das Durchführen der modularen Reduktion von x mod p, wobei die modulare Reduktion umfasst:
  - Berechnen eines genäherten Basisquotienten m,
  - Berechnen einer Quotienten-Näherung q,
  - Berechnen eines genäherten Rests r, und
  - Berechnen einer geschätzten, reduzierten Form des Wert x auf der Basis der Quotienten-Näherung und des genäherten Rests.

Die Tabelle 3.1 veranschaulicht den Barett-Reduktion-Algortihmus.

Weitere grundlegende und wesentliche Operationen für die Kryptographie sind die modulare Addition, modulare Subtraktion und modulare Multiplikation. Die Berechnung erfolgt genauso wie bei der normalen Arithmetik und dann wird das Ergebnis durch den Modulo geteilt, bis der kleinste positive Rest gefunden wird.

3.1.2 Modulare Addition

Die modulare Addition über $\mathbf{Z_p}$ kann mathematisch geschrieben werden als:

$$x + y(mod p)$$

, wobei x und y die gegebene Zahlen und p die Primzahl sind. In dieser Arbeit wurde für die Implementierung der modularen Addition zuerst x und y addiert, dann wird das

Algorithmus: Modulo-Berechnung
Input: b > 3, p, $k = \lfloor log_b p \rfloor + 1, 0 \leq x < b^{2k}, m = \left\lfloor \frac{b^{2k}}{p} \right\rfloor$
Output: $xmodp$
1. Berechne $q = \left\lfloor \left\lfloor \frac{x}{b^{k-1}} \right\rfloor \frac{m}{b^{k+1}} \right\rfloor \cdot p$;
2. $r = (x \bmod b^{k+1}) - (q \cdot p \bmod b^{k+1})$;
3. If r < 0 then $r = r + b^{k+1}$;
4. While $r \geq p$ do r = r - p;
5. Return r.

Tabelle 3.1: Barett-Reduktion[2]

Ergebnis modulo p berechnet, wenn außerhalb des Bereichs von [0, p-1] liegt, sonst wird die Summe direkt ausgegeben.

Seien zum Beispiel x = 17, y = 4, p = 5, dann ist

$$x + y(modp) = \quad 17 + 4(mod5) = \quad 21 \equiv 1mod5$$

Ein kostengünstiger bzw. schnellerer Algorithmus wäre zuerst x und y binär darzustellen und in einem Array von t-Bit zu speichern. Danach sind x und y wortweise bzw. bitweise zu addieren und das Ergebnis dann p subtrahieren, solange es p - 1 überschreitet. Jede Wortaddition erzeugt zum Beispiel in einer 32-Bit Plattform-Architektur eine 32-Bit-Summe und eine 1-Bit-Übertragsziffer , die zur nächsthöheren Summe addiert werden[2]. Die einfache Addition und die *Addition mit Übertrag* können schnelle Einzeloperationen sein. [26]. Die Tabellen 3.2 und 3.3 veranschaulichen die beiden Algorithmen.

3.1.3 Modulare Subtraktion

Mathematisch kann die modulare Subtraktion als $(x - y)modp$ geschrieben werden, wobei x und y die gegebenen Zahlen und p die Primzahl sind. Ein einfachster Weg diese Operation durchführen, besteht darin, die beiden Zahlen x und y zuerst zu subtrahieren und dann der kleinste positive Rest im Bereich von [0, p-1] zu berechnen, in dem das Ergebnis der Subtraktion von (x - y) durch p geteilt wird. Der Algorithmus sieht genauso wie den Algorithmus der modularen Addition, wobei im ersten Schritt anstatt eine Addition, eine Subtraktion durchgeführt wird.

Seien zum Beispiel x = 15, y = 23, p = 5, dann ist

$$(x - y)modp = \quad (15 - 23)mod5 = \quad -8 \equiv 2mod5$$

In der effizienteren Art, Modulo-Subtraktion durchzuführen, wird das Übertragungsbit nicht mehr als `Carry-Bit`, sondern als `Borrow-Bit` (Ausleih-Bit) interpretiert.[2]. Die Operation wird dann ähnlich wie die modulare Addition implementiert.

3.1.4 Modulare Multiplikation

Die modulare Multiplikation ist eine der teuersten und zeitaufwändigsten Operationen der Kryptographie über Z_p. Aber um ein höheres, leistungsstarkes Kryptosystem zu entwickeln, muss eine effiziente Implementierung der modularen Multiplikation erfolgen

Klassischer Algorithmus: Modulare Addition
Input: Modulo p und Zahlen x, y $\in$ [0, p - 1]
Output: $r = (x + y) mod p$
1. Berechne r = x + y;
2. If $r \neq 0$, then finde den Rest r = r mod p else r = 0;
3. return r.

(a) 1.Tabelle

Tabelle 3.2: Modulare Addition aus der Klasse *BasicTheoreticMethods*

Algorithmus: Modulare Addition
Input: Modulo p und Zahlen x, y $\in$ [0, p - 1]
$t = \lceil m/32 \rceil$ und $m = \lceil log_2 p \rceil$
Output: $c = (x + y) mod p$
1. $c_0 = x_0 + y_0$;
2. For i from 1 to t-1 do: $c = Add\_With\_Carry(x_i, y_i)$;
3. If the carry bit is set, then subtract p from $c = (c_{t\_1}, ..., c_2, c_1, c_0)$;
4. If $c \geq p$ then $c = c - p$;
5. Return c.

(a) 2.Tabelle

Tabelle 3.3: Empfohlene modulare Addition von amerikanischen Standard NIST für eine 32-Bit Architektur-Plattform[2].

[26]. Mathematisch kann die modulare Multiplikation-Operation ausgedrückt werden als

$$(x \cdot y) mod p.$$

Um Modulo-Multiplikationen durchzuführen, wurde in dieser Arbeit die Zahlen x und y einfach multipliziert und dann die ganzzahlige Division durch p zu verwendet, um den Rest zu erhalten. Dies bedeutet auch, dass die Leistung des ganzen Algorithmus, genauso wie bei der modularen Addition und Subtraktion, von Modulo-Operationen bzw. von der Leistung des Divisionsalgorithmus abhängt. Seien zum Beispiel: x = 35, y = 7, p = 5, dann gilt:

$$(x \cdot y) mod p = \quad (35 \cdot 7) mod 5 = \quad 245 \equiv 0 mod 5$$

Neben der Barett-Reduktion Methode gibt es auch eine andere Methode, die die Modulo-Operationen viel schneller als die reguläre klassische Methode durchführen kann: Die **Montgomery-Multiplikation**. Die **Montgomery-Multiplikation** ist ein Verfahren zur modularen Multiplikation, bei der die traditionelle Teilung-Operation vermeidet wird und anschließend nur Multiplikationen, Additionen und Verschiebungen verwendet werden[29]. Der modulare multiplikative Algorithmus ist in Tabelle 3.4 angegeben.

Algorithmus: Modulare Multiplikation

Input: p, x und y zwei positive k-Bit Ganzzahlen, x_i, y_i: i-te Bit in x und y
Output: $m = x \cdot y \bmod p$

1. m = 0;
2. For i = 0 to k-1
   2.1 $m = m + (x \cdot y_i)$;
   2.2 If $(m_0 = 1)$ then $m = m/2$ else $m = (m + p)/2$;
3. Return m.

Tabelle 3.4: Montgomery-Multiplikation[29]

Das Verfahren ist für eine einzelne modulare Multiplikation nicht effizient, kann jedoch effektiv bei Berechnungen wie der modularen Exponentiation verwendet werden, bei denen viele Multiplikationen für eine gegebene Eingabe durchgeführt werden[8].

3.1.5 Modulare Exponentiation

3.2 Modulare Inverse

3.3 Polynomarithmetik

4 Endliche Koerper

5 Elliptische Kurven

5.1 Darstellungen von Punkten

5.2 Punktmultiplikation

5.3 Punktaddition

5.4 Punktverdopplung

6 Elliptic Curve Diffie Hellman

7 Fazit

Literatur

[1] .

[2] *Advanced Encryption Standard*.

[3] Menezes Alfred, Oorschot Paul und Vanstone Scott. *Handbuch of Applied Cryptography*. CRC-Press, 1997.

[4] Nada Ali und Esaa Kawther. „Security Improvement in Elliptic Curve Cryptography“. In: *International Journal of Advanced Computer Science and Applications* 9 (Jan. 2018), S. 122–133.

[5] Werner Annette. *Elliptische Kurven in der Kryptographie*. Hrsg. von Springer. Springer-Verlag Berlin Heidelberg GmbH, 2002.

[6] Khan Arif. *Comparative Analysis of Elliptic Curve Cryptography*. Jan. 2017. ISBN: 978-3-330-01788-7.

[7] Hankerson Darrel, López Julio und Menezes Alfred. „Software Implementation of Elliptic Curve Cryptography over Binary Fields“. In: Jan. 2000, S. 1–24. ISBN: 978-3-540-41455-1. DOI: 10.1007/3-540-44499-8_1.

[8] Hankerson Darrel, Vanstone Scott und Menezes Alfred. „Guide to Elliptic Curve Cryptography“. In: *Springer Professional Computing*. 2004.

[9] Michel Douguet und Vincent Dupaquis. „Modulare Reduktion unter Verwendung einer speziellen Form des Modulo“. Deutsch. Pat. DE112009000152T5. 2008. URL: https://patents.google.com/patent/DE112009000152T5/de.

[10] Kossi D. Edoh. „Elliptic curve cryptography: Java implementation“. In: *Information Security Curriculum Development Conference, InfoSecCD 2004*. 2004, S. 88–93.

[11] *Elliptic Curve Crytography*. Technical Guideline. Bonn, Germany: Federal Office for Information Security, Juni 2018.

[12] Samta Gajbhiye, Monisha Sharma und Samir Dashputre. „A Survey Report On Elliptic Curve Cryptography“. In: *International Journal of Electrical and Computer Engineering (IJECE)* 1 (Okt. 2011). DOI: 10.11591/ijece.v1i2.86.

[13] Rahman Hazifur, Azad Saiful und Khan Pathan Al-Sakib. *Practical Cryptography Algorithms and Implementations using C++*. CRC Press, 2015. ISBN: 13: 978-1-4822-2890-8.

[14] *Information Technology Security Techniques- Digital Signatures., with Appendix- part 3: Certificatebased mechanism*.

[15] Steffen Daniel Jensen, Brian Melin Iversen, Rasmus Feldthaus und Markus Neubrand. *Cryptography: Fast Modular Arithmetic*. Aarhus University, 2009.

[16] Merkle Johannes und Lochter Manfred. „Ein neuer Standard für Elliptische Kurven“. In: Mai 2009.

[17] Sheetal Kalra und Sandeep Sood. „Elliptic curve cryptography: Survey and its security applications“. In: *Proceedings of the International Conference on Advances in Computing and Artificial Intelligence, ACAI 2011* (Jan. 2011), S. 102 –106. DOI: 10.1145/2007052.2007073.

[18] Anoop MS. „Elliptic curve cryptography-an implementation guide“. In: (2007).

[19] Mihailescu Marius und Nita Stefania. „Elliptic-Curve Cryptography“. In: Jan. 2021, S. 189–223. ISBN: 978-1-4842-6585-7. DOI: 10.1007/978-1-4842-6586-4_9.

[20] Mohamad Afendee Mohamed. „A survey on elliptic curve cryptography“. In: *Applied mathematical sciences* 8 (2014), S. 7665–7691.

[21] Abdalbasit Mohammed und Nurhayat Varol. „A Review Paper on Cryptography“. In: Juni 2019, S. 1–6. DOI: 10.1109/ISDFS.2019.8757514.

[22] Nayak und Rajput. „Cryptography Algorithms – The Science of Information Security: Review Paper“. In: 2017.

[23] Koblitz Neal. „Elliptic Curve Cryptosystems“. In: Bd. 48. 177. 1987. Kap. Mathematics of Computation, S. 203–209.

[24] Barett Paul. „Implementing the Rivest Shamir and Adleman Public Key Encryption Algorithm on a Standard Digital Signal Processor“. In: 1986. Kap. Advances in Cryptology — CRYPTO' 86, S. 311 –323. ISBN: ISBN 978-3-540-18047-0. DOI: doi:10.1007/3-540-47721-7_24.

[25] *Public Key Cryptography for the financial Services Industry: The Elliptic curve Digital SignatureAlgorithm (ECDSA)*. ANSI X9.62, 1999.

[26] Hossain Rownak und Hossain Selim. „Efficient FPGA Implementation of Modular Arithmetic for Elliptic Curve Cryptography“. In: *International Conference on Electrical, Computer and Communication Engineering (ECCE)* (2019), S. 1–6.

[27] Markan Ruchika und Kaur. „Literature Survey on Elliptic Curve Encryption Techniques“. In: 2013.

[28] *STANDARDS FOR EFFICIENT CRYPTOGRAPHY SEC 2: Recommended Elliptic Curve Domain Parameters*. Jan. 2010. URL: https://www.secg.org/sec2-v2.pdf.

[29] Sushanta Sahu und Manoranjan Pradhan. „Implementation of Modular Multiplication for RSA Algorithm“. In: Juli 2011, S. 112 –114. DOI: 10.1109/CSNT.2011.30.

[30] Contini Scott, Çetin Kaya Koç und Colin Walter. „Modular Arithmetic“. In: *Encyclopedia of Cryptography and Security*. Hrsg. von Henk C. A. van Tilborg und Sushil Jajodia. Boston, MA: Springer US, 2011, S. 795–798. ISBN: 978-1-4419-5906-5. DOI: 10.1007/978-1-4419-5906-5_49. URL: https://doi.org/10.1007/978-1-4419-5906-5_49.

[31] Ankita Soni und Nisheeth Saxena. „Elliptic Curve Cryptography: An Efficient Approach for Encryption and Decryption of a Data Sequence“. In: *International Journal of Science and Research (IJSR)* 2 (2013), S. 203–208.

[32] *Standard Specifications for Public Key Cryptography*.

[33] Stolbikova Veronika. „Can Elliptic Curve Cryptography Be Trusted? A Brief Analysis of the Security of a Popular Cryptosystem“. In: *ISACA Journal* 3 (Mai 2016), S. 1–5.

[34] Miller Victor. „Use of elliptic curves in cryptography“. In: LNCS 218, Springer Verlag, 1986. Kap. Advancesin Cryptography-Crypto '85, S. 417 –426.

[35] Hasenplaugh William, Gaubatz Gunnar und Gopal Vinodh. „Fast Modular Reduction“. In: 2007. DOI: doi:10.1109/ARITH.2007.18..

[36] Dietmar Wätjen. *Kryptographie Grundlagen, Algorithmen, Protokolle*. Bd. 3. Springer-Verlag, 2018. DOI: https://doi.org/10.1007/978-3-658-22474-5.

Abbildungsverzeichnis

Tabellenverzeichnis

Listings

Abkürzungsverzeichnis