Karoline Moser, Leoben

Kryptographie

Die sichere Brücke im digitalen Zeitalter?!

Kryptographie ist eine seit Jahrhunderten perfektionierte Kunst und Wissenschaft zum Schutz von Informationen. Es ist eine Disziplin, die unsichtbar, aber essenziell ist für unser Leben im digitalen Zeitalter. Wenn wir heute online shoppen, Nachrichten verschicken oder Banking-Apps benutzen – überall ist Kryptographie im Spiel.

Sie ist die unsichtbare Brücke, die Vertrauen schafft, wo sich Menschen nie persönlich begegnen – von WhatsApp bis zu Bitcoin. Und diese Brücke muss sicher, stabil und effizient sein. Die moderne Kryptographie, vertreten in diesem Beitrag durch RSA- und Elliptische-Kurven-Kryptosysteme, befindet sich inmitten von signifikanten Herausforderungen und Entwicklungen.

Ver- & Entschlüsselung

Reinhart möchte Lisa eine verschlüsselte Nachricht schicken. Dazu verwendet er eine Verfahrensanleitung (Algorithmus) und einen Schlüssel, um den Klartext zu verschlüsseln. Der resultierende Chiffretext wird an Lisa gesendet, die die Nachricht mit einem Schlüssel entschlüsselt, um den ursprünglichen Klartext zu erhalten. Bei der Private–Key-Kryptographie wird derselbe Schlüssel zum Ver- und Entschlüsseln verwendet, während bei der Public-Key-Kryptographie zwei verschiedene Schlüssel zum Einsatz kommen. Diffie und Hellman haben die Public-Key-Kryptographie erstmals in "New directions in cryptography" beschrieben (Diffie und Hellman 1976). Basierend auf ihrer Arbeit schlugen Rivest, Shamir und Adleman in „A Method for Obtaining Digital Signatures and Public-Key Cryptosystems" eines der ersten Public-Key-Kryptosysteme vor, das heute als RSA bekannt ist – benannt nach den Autoren (Rivest et al. 1978). Der Fokus dieses Beitrags liegt auf der Public-Key-Kryptographie, insbesondere auf den Kryptosystemen RSA und Elliptische Kurven Kryptographie. Elliptische Kurven Kryptographie (ECC) ist ein Public-Key-Kryptosystem, das in den letzten Jahren an Aufmerksamkeit gewonnen hat. Neal Koblitz (Koblitz 1987) und Victor Miller (Miller 1986) haben unabhängig voneinander vorgeschlagen, elliptische Kurven in der Kryptographie zu verwenden.

Cäsar Verschlüsselung

Eine der ältesten Verschlüsselungen ist beschrieben in Suetons Cäsaren-Vita *Divus Iulius*. Der römische Feldherr Gaius Julius Cäsar wollte geheime Botschaften an seine Truppen senden, ohne dass sie vom Feind gelesen werden konnten. Seine Lösung war einfach, aber genial: Er verschob jeden Buchstaben im Alphabet um dieselbe Anzahl von Stellen. Ein solches Verfahren nennt man heute Private-Key-Kryptographie, weil der Sender und der Empfänger beide wissen müssen, um wie viele Stellen das Alphabet verschoben wurde. Diesen geheimen Schlüssel muss man sich im Vorhinein ausmachen oder mit einem vertrauenswürdigen Boten überbringen. Heutzutage können alle Möglichkeiten der Verschiebung des Alphabets mit Hilfe von Computern innerhalb von weniger als einer Sekunde geknackt werden (Katz und Lindell 2021).

Hier ein Beispiel für eine Cäsar-Verschlüsselung mit Verschiebung um 3 Stellen, wobei aus „A" → „D", aus „B" → „E", aus „C" → „F" usw. wird: Die Nachricht „LEOBEN" wird zu „OHREHQ".

RSA-Kryptographie

RSA ist ein bewährtes Verfahren, das auf einem simplen Prinzip basiert: Es ist sehr einfach, zwei große Zahlen miteinander zu multiplizieren – aber extrem schwer, das Ergebnis wieder zurückzurechnen, also die ursprünglichen Zahlen zu finden (Katz und Lindell 2021).

Zwei wichtige mathematische Grundlagen, die in Lehrbüchern zur Arithmetik ausführlich abgehandelt sind, seien hier zum Verständnis kurz erläutert. Grundsätzlich wird im Folgenden mit positiven ganzen Zahlen gerechnet. Dabei ist die Teilbarkeit so definiert, dass eine Zahl ohne Rest durch eine andere teilbar ist. Bei der Modulo-Rechnung, auch bekannt als Division mit Rest, versucht man den Rest zu bestimmen, der bei einer Division stehen bleibt.

Ein Beispiel wäre 17 = 2 mod 5. Dies ist gleichbedeutend mit 17 = 5*3 + 2.

Weiters sei die Eulersche φ-Funktion eingeführt. Sie ist für jede positive ganze Zahl n definiert und gibt die Anzahl der positiven ganzen Zahlen kleiner als n an, die teilerfremd zu n sind. Das heißt, deren größter gemeinsamer Teiler mit n gleich 1 ist. Die Zahl 8 ist zu genau vier Zahlen von 1 bis 8 teilerfremd – nämlich zu 1, 3, 5 und 7. Daher gilt φ(8) = 4. Ein weiteres Beispiel ist die Zahl 11. Da sie eine Primzahl ist, ist sie zu jeder Zahl von 1 bis 10 teilerfremd. Es folgt φ(11) = 10. Für jede Primzahl p gilt φ(p) = p-1. Diese Eigenschaft ist im Zusammenhang mit der RSA-Kryptographie relevant. Zudem ist die φ-Funktion multiplikativ, das heißt φ(mn) = φ(m)φ(n). Wendet man diese Eigenschaft nun auf zwei Primzahlen p und q an, so ergibt sich φ(pq) = (p-1)(q-1).

Für die Schlüsselerzeugung werden nun verschiedene Parameter benötigt. Zuerst wählt die Empfängerin zwei Primzahlen p und q. Durch Multiplikation dieser beiden Primzahlen erhält sie die Zahl N. Sie wählt eine Zahl e, die teilerfremd zu φ(N) = φ(pq) = (p-1)(q-1) ist. Jede Primzahl, die größer ist als max(p, q) – also größer als die größere der beiden gewählten Primzahlen p und q –, ist teilerfremd zu (p-1)(q-1). Häufig wird als öffentlich bekannter Exponent e entweder 3 oder 65537 verwendet. Warum diese Werte besonders geeignet sind und welche Voraussetzungen erfüllt sein müssen, um sie verwenden zu können, ist zum Beispiel in Katz' und Lindells „Introduction to Modern Cryptography" (2021) erläutert. Sodann wird eine Zahl d so berechnet, dass e und d multipliziert und geteilt durch (p-1)(q-1) den Rest 1 ergeben. Der öffentliche Schlüssel ist (e, N) und der private Schlüssel (d, N) (Rivest et al. 1978).

Der Sender wandelt nun die Nachricht in eine Zahl M um – hierzu siehe unten – und berechnet die verschlüsselte Nachricht C mit dem öffentlichen Schlüssel des Empfängers, wobei M mit e potenziert und durch N geteilt wird, wobei der Rest als verschlüsselte Nachricht C dient.

Die Empfängerin entschlüsselt die Nachricht, indem sie C mit d potenziert und durch N teilt. Der verbleibende Rest gibt die ursprüngliche Nachricht M zurück.

Der grundsätzliche Ablauf der RSA-Kryptographie ist demnach wie folgt:

1. Schlüsselerzeugung

a. Lisa wählt zwei große Primzahlen **p** und **q**.

b. Lisa berechnet **N = p q**.

c. Lisa wählt eine Zahl **e**, die teilerfremd zu **(p-1)(q-1)** ist.

d. Lisa berechnet **d**, sodass **e d = 1 mod (p-1)(q-1).**

e. Der öffentliche Schlüssel ist somit **(e, N)** und der private Schlüssel von Lisa ist **(d, N)**.

2. Verschlüsselung

a. Reinhart wandelt die Nachricht, z.B. "Leoben", in eine Zahl **M** um.

b. Die verschlüsselte Nachricht **C**, die Reinhart an Lisa schickt, ist **C = M^e mod N**.

3. Entschlüsselung

a. Lisa berechnet **M= C^d mod N**, um **M** zu erhalten.

RSA-Kryptographie: ein Beispiel

Die Empfängerin, Lisa, muss ein Public-Private Key Paar generieren. Dafür wählt sie zwei große Primzahlen und errechnet durch deren Multiplikation die Zahl N. Weiters berechnet sie das Produkt (p-1)(q-1) aufgrund der Eulerschen φ-Funktion. Lisa wählt zum Beispiel:

$$p = 4294900427,$$
$$q = 4294901243,$$
$$N = p\,q = 18446173182483530761,$$
$$(p-1)(q-1) = 18446173173893729092.$$

Lisa wählt weiters eine Zahl e nach den oben im Schritt 1.c. genannten Kriterien und berechnet daraus d (Schritt 1.d.). So entstehen ihr öffentlicher und privater Schlüssel.

Lisas öffentlicher Schlüssel lautet:

$$(e, N) = (65537, 18446173182483530761).$$

Ihr privater Schlüssel, den nur sie alleine kennt, würde dann so lauten:

$$(d, N) = (15756804043836592185, 18446173182483530761).$$

Der öffentliche Schlüssel kann allseits bekannt sein, da man aus seinen Parametern nur sehr schwer auf die ursprünglichen Primzahlen p und q schließen kann. Dies wird auch als Faktorisierungsproblem bezeichnet.

Damit Reinhart die Nachricht „Leoben" verschlüsseln kann, muss er diese zuerst in eine Zahl umwandeln. Hierfür verwendet er das ASCII-System. Jeder Buchstabe wird anhand des ASCII-Codes in eine 8-Bit-Binärzahl umgewandelt und hintereinandergeschrieben. Diese Binärzahl wird als eine große Zahl interpretiert und ins Dezimalsystem umgerechnet. Das ergibt die dezimale Darstellung der Nachricht, die bereit ist, verschlüsselt zu werden. Für die Nachricht „Leoben" würde die zahlenmäßige Repräsentation wie folgt lauten:

$$M = (01001100\ 01100101\ 01101111\ 01100010\ 01100101\ 01101110)_2$$
$$= (83998544127342)_{10}.$$

Als Sender benötigt Reinhart nun den öffentlichen Schlüssel von Lisa und die zahlenmäßige Repräsentation der Nachricht:

$$(e, N) = (65537, 18446173182483530761),$$
$$M = 83998544127342.$$

In der RSA-Kryptographie wird die Nachricht mit dem Parameter e des öffentlichen Schlüssels potenziert und durch N dividiert. Der Rest ist die verschlüsselte Nachricht, die sicher an Lisa übermittelt werden kann (siehe Schritt 2.b. oben). In unserem Beispiel ist folgende Berechnung durchzuführen, um die verschlüsselte Nachricht C zu erhalten:

$$C = M^e \bmod N,$$
$$C = 83998544127342^{65537} \bmod 18446173182483530761,$$
$$C = 18080932581028888918.$$

Diese verschlüsselte Botschaft sendet Reinhart nun an Lisa. Selbst wenn Christian die Nachricht abfängt, kann er sie kaum entschlüsseln, da ihm der zugehörige private Schlüssel fehlt.

Sobald Lisa die verschlüsselte Nachricht erhält, kann sie diese mithilfe ihres privaten Schlüssels entschlüsseln. Dabei potenziert sie den Chiffretext mit der Potenz d und berechnet anschließend den Rest bei Division durch N. Lisa berechnet wie folgt:

$$M = C^d \bmod N,$$
$$M = 18080932581028888918^{15756804043836592185} \bmod 18446173182483530761,$$
$$M = 83998544127342.$$

M entspricht der ursprünglichen zahlenmäßigen Darstellung der Nachricht „Leoben" (Katz und Lindell 2021).

Elliptische Kurven Kryptographie

Elliptische Kurven Kryptographie nutzt die Eigenschaften elliptischer Kurven. Eine solche Kurve über die Reellen Zahlen wird in einem durch die Achsen x und y definierten Diagramm durch die allgemeine Form y^2 = x^3 + Ax + B dargestellt, wobei A und B bestimmte Bedingungen erfüllen müssen, die zu erläutern an dieser Stelle zu weit führen würde. Diese Kurven sind symmetrisch bezüglich der x-Achse, und jede nicht senkrechte Gerade schneidet sie genau dreimal, wobei die Berührungspunkte bei Tangenten doppelt gezählt werden.

Die Grundlagen der Elliptische Kurven Kryptographie sind in „Fundamentals of Cryptography" (Buell 2021) und in „Introduction to Modern Cryptography" (Katz und Lindell 2021) dargestellt, die auch als weiterführende Literatur zum Thema dienen können.

Unter Nutzung der Eigenschaften elliptischer Kurven lässt sich so etwas wie die Addition von Punkten („Punktaddition") definieren: Zieht man eine Gerade durch zwei Punkte P und Q auf der Kurve, schneidet man einen dritten Punkt (-R). Spiegelt man -R an der x-Achse, erhält man den Punkt R, das Ergebnis der Addition.

In der Elliptischen Kurven Kryptographie wird ein Generatorpunkt G mit einem Skalar multipliziert, um Punkte wie 2G, 3G, 4G usw. zu erhalten, wobei auch die Punktaddition und die doppelte Zählung von Berührungspunkten einer Tangente genutzt werden. Diese Punkte können überall auf der Kurve landen, und nach einer bestimmten Anzahl von Additionen können sie sogar zum Ausgangspunkt zurückkehren und einen neuen Zyklus beginnen. Selbst wenn man den Generatorpunkt und einen weiteren Punkt auf der Kurve kennt, ist es sehr schwierig zu bestimmen, wie oft der Generatorpunkt mit sich selbst addiert wurde, um diesen Punkt zu erhalten. Dieses Prinzip ist fundamental für die Elliptische Kurven Kryptographie und als das Diskrete Logarithmusproblem der Elliptischen Kurven bekannt (Buell 2021).

Nach dem Legen einer Tangente durch den Punkt G erhält man durch den Schnittpunkt mit der elliptischen Kurve und Spiegelung entlang der Symmetrieachse den Punkt 2G. Der Punkt 3G wird durch Addition von G und 2G ermittelt, indem man eine Gerade durch diese Punkte legt. Diese schneidet die Kurve bei -3G, und der gespiegelte Punkt ist der gesuchte 3G. Die Punktaddition kann fortgeführt werden, wobei es so erscheint, als ob die Bewegung auf der Kurve fast zufällig erfolgt.

Man betrachtet einen Punkt auf der Kurve und fragt sich, wie oft G multipliziert wurde, um diesen Punkt zu erhalten. Der Multiplikationsfaktor ist der private Schlüssel. Der Punkt, also dG, kann öffentlich bekannt sein, aber das d, der Multiplikationsfaktor, muss geheim bleiben.

Eine elliptische Kurve wird über einem endlichen Körper (Finite Field) definiert, eine algebraische Struktur, die grundlegende arithmetische Operationen ermöglicht. Bei einer gegebenen Primzahl p sind die Elemente des Körpers die Zahlen von 0 bis p-1, und die Kurvengleichung wird modulo p berechnet: y^2 = x^3 + Ax + B mod p. Die Auswahl von A, B und p ist entscheidend für die Sicherheit der Kurve.

In elliptischen Kurven repräsentiert das Neutralelement, auch "Point-At-Infinity" genannt, das Äquivalent zur Null in der Arithmetik und bleibt bei der Addition unverändert. Es ist integraler Bestandteil der durch Punktaddition gebildeten endlichen Gruppe auf der Kurve. Der Parameter n, die Ordnung des Generatorpunktes, ist die kleinste positive Zahl, bei deren multipler Addition des Punktes G das Neutralelement resultiert. Dieser Parameter ist daher entscheidend für die Struktur und Eigenschaften der Punktgruppe auf der Kurve (Buell 2021).

Elliptische Kurven Kryptographie: Ablauf

Das Verfahren zur Umwandlung der zahlenmäßigen Repräsentation einer Nachricht – zum Beispiel einer Dezimalzahl – in einen Punkt auf einer elliptischen Kurve umfasst mehrere Schritte, wie sie beispielsweise in „Fundamentals of Cryptography" (Buell 2021) beschrieben werden. Schematisch läuft die Elliptische Kurven Kryptographie wie folgt ab:

1. Schlüsselerzeugung

a. Lisa wählt eine elliptische Kurve und einen Punkt **G** auf dieser Kurve.

b. Lisa wählt eine zufällige Zahl **d** als privaten Schlüssel.

c. Der öffentliche Schlüssel von Lisa beinhaltet die gewählte elliptische Kurve und den Generatorpunkt **G**, sowie **Q = dG**, einen weiteren Punkt auf der elliptischen Kurve.

2. Verschlüsselung

a. Reinhart wandelt die Nachricht, z.B. "Leoben", in einen Punkt **M** auf der elliptischen Kurve um.

b. Reinhart wählt eine zufällige Zahl **k**.

c. Reinhart berechnet $\mathbf{R_1 = kG}$ und $\mathbf{R_2 = M + kQ}$.

d. Die verschlüsselte Nachricht, die Reinhart an Lisa schickt, ist $\mathbf{(R_1, R_2)}$.

3. Entschlüsselung

a. Lisa berechnet $\mathbf{dR_1 = kQ}$.

b. Die entschlüsselte Nachricht ist $\mathbf{M = R_2 - dR_1 = R_2 - kQ = M + kQ - kQ}$.

Sicherheitsanalyse

Die Sicherheit kryptographischer Systeme kann durch das Lösen der zugrundeliegenden mathematischen Probleme oder durch Designfehler gefährdet werden. Lokale Kopien von Nachrichten oder sensiblen Daten, die während der Ver- oder Entschlüsselung erstellt werden, stellen eine besondere Gefahr dar und müssen sicher gelöscht werden.

Für Verfahren wie RSA und Elliptische Kurven Kryptographie sind „Side-Channel Attacks" riskant. Hierbei werden Informationen aus Nebenkanälen wie Stromverbrauch oder elektromagnetische Emissionen analysiert, um Rückschlüsse auf Schlüssel zu ziehen. Gegenmaßnahmen wie die Hinzufügung von zufälligen Daten (Maskierung) können die Vorhersagbarkeit reduzieren.

Ein kryptographisches System ist nur so sicher wie sein schwächstes Glied. Der menschliche Faktor, etwa der unvorsichtige Umgang mit privaten Schlüsseln, kann oft das größte Risiko darstellen, wodurch selbst robuste und sichere Systeme kompromittiert werden können (Buell 2021).

Vergleichsanalyse

Die moderne Kryptographie, insbesondere durch RSA- und Elliptische-Kurven-Kryptosysteme repräsentiert, befindet sich inmitten von Herausforderungen und Entwicklungen. RSA stützt seine Sicherheit auf die Komplexität der Faktorisierung großer Zahlen und benötigt daher große Primzahlen, um robust zu sein. Elliptische Kurven hingegen, die auf dem Diskreten Logarithmusproblem basieren, ermöglichen kürzere Schlüssellängen, was in puncto Geschwindigkeit und Effizienz von Vorteil ist.

Die Algorithmen zum Lösen bzw. Brechen von kryptographischen Systemen lassen sich hinsichtlich ihrer Leistungsfähigkeit (Laufzeit) grundsätzlich in drei Kategorien einteilen:

Polynomieller Algorithmus: Hier ist die Laufzeit proportional zu einer Potenz der Eingabegröße. Die Zeitkomplexität wächst also polynomiell mit zunehmender Eingabegröße an.

Exponentieller Algorithmus: Die Laufzeit lässt sich hier nicht mit einem polynomiellen Ausdruck beschränken. Die Zeitkomplexität nimmt exponentiell mit der Eingabegröße zu. Bei großen Eingaben sind exponentielle Algorithmen langsamer als polynomielle.

Subexponentieller Algorithmus: Die Laufzeit wächst schneller als bei polynomiellen Algorithmen, aber langsamer als bei exponentiellen Algorithmen.

Die Laufzeit eines polynomiellen Lösungsalgorithmus wächst somit langsamer als die eines subexponentiellen Algorithmus und diese wiederum

langsamer als die eines exponentiellen Algorithmus (Hankerson et al. 2004).

Um dies genauer zu betrachten: RSA verwendet große, zufällige Primzahlen, um Widerstand gegen Angriffe zu bieten. Es ist zu beachten, dass für das zugrundeliegende Faktorisierungsproblem subexponentielle Lösungsalgorithmen existieren. Im Gegensatz dazu sind für das Diskrete Logarithmusproblem auf elliptischen Kurven nur Lösungsalgorithmen mit exponentieller Laufzeit bekannt, was zusätzliche Sicherheit bietet (Hankerson et al. 2004).

Bei der Anwendung der Elliptischen Kurven Kryptographie ist die Auswahl einer sicheren Kurve und eines geeigneten Generatorpunkts entscheidend. Es wird geraten, standardisierte und bewährte Kurven zu verwenden. Dank der höheren Effizienz dieses Systems können kürzere Schlüssel verwendet werden, was zu Einsparungen bei Rechenleistung, Bandbreite und Speicher führt (Barker 2020).

Die Empfehlungen des US National Institute of Standard and Technology (NIST; siehe Barker 2020) bieten Richtlinien zur Bestimmung des Sicherheitsniveaus kryptographischer Verfahren. Die Tabelle, die diese Empfehlungen darstellt, ist wie folgt strukturiert:

Sicherheitslevel	RSA – Zahl N	ECC – Ordnung n des Generatorpunktes G
112	2048	224 - 255
128	3072	256 - 383
192	7680	384 - 511
256	15360	512 +

Tabelle 1: Die notwendigen Schlüssellängen für ein bestimmtes Sicherheitsniveau der RSA- und ECC-Algorithmen laut NIST in (Barker 2020).

Spalte 1 zeigt die Sicherheitsstärke des Algorithmus in Bits an. Diese Stärke repräsentiert den rechnerischen Aufwand, der benötigt wird, um ein kryptographisches System zu brechen.

Spalte 2 spezifiziert die Größe von N für die RSA-Kryptographie in Bits.

Spalte 3 definiert den Bereich für die Größe von n in der Elliptische Kurven Kryptographie, wobei n die Ordnung des Basispunktes G in Bits ist.

Bei einem Sicherheitsniveau von 128 Bits ist auffällig, dass die empfohlene Schlüssellänge für RSA etwa 12-mal größer ist als für Elliptische-Kurven Kryptographie. Dieser Unterschied in der Schlüssellänge kann insbesondere in Systemen mit eingeschränkten Ressourcen zu erhöhtem Speicherbedarf und höheren Kosten führen (Barker 2020).

Es ist wichtig zu betonen, dass die NIST-Empfehlungen dynamisch sind und regelmäßig aktualisiert werden, um mit technologischen Fortschritten und neu entdeckten Angriffsmethoden Schritt zu halten.

Während beide Systeme ihre eigenen Stärken und Schwächen aufweisen, ist es bei beiden essentiell, standardisierte und sichere Parameter zu wählen. Mit dem Aufkommen von Quantencomputern können beide Systeme in polynomieller Zeit gebrochen werden (polynomieller Lösungsalgorithmus). Daher wird die Entwicklung der Post-Quanten-Kryptographie immer relevanter (Shor 1994).

Derzeit wird intensiv an der Entwicklung leistungsfähiger Quantencomputer geforscht. Die Aussicht, dass solche Systeme in naher Zukunft Realität werden könnten, zwingt die Kryptographie schon heute zum Umdenken. Von zentraler Bedeutung ist dabei die Entwicklung der Post-Quanten-Kryptographie – also von Verschlüsselungsverfahren, die auch

Angriffen mit Quantencomputern standhalten können.

Eine zentrale Herausforderung besteht darin, diese neuen Verfahren so zu gestalten, dass sie sich möglichst reibungslos in bestehende Infrastrukturen integrieren lassen. Die Zukunft der Kryptographie hängt daher nicht nur von der technischen Realisierbarkeit neuer Verfahren ab, sondern ebenso von deren Standardisierung, Praxistauglichkeit und Kompatibilität mit vorhandenen Systemen.

Wir stehen an der Schwelle zu einer neuen Ära der Kryptographie, in der die Balance zwischen Sicherheit, Effizienz und Anwendbarkeit neu definiert wird. Die unsichtbare Brücke der Kryptographie muss dabei sicher, stabil und effizient bleiben.

Literaturverzeichnis

Barker 2020 = Elaine Barker. SP 800-57 Part 1 Rev. 5 Recommendation for Key Management: Part 1 - General. Tech. rep. Gaithersburg, MD, USA, May 2020. DOI: 10.6028/NIST.SP.800-57pt1r5.

Buell 2021 = Duncan Buell. Fundamentals of Cryptography. Springer Nature Switzerland AG, 2021. ISBN: 978-3030734916

Diffie und Hellman 1976 = Whitfield Diffie and Martin E. Hellman. "New directions in cryptography". In: IEEE Transactions on Information Theory 22.6 (1976), pp. 644–654. DOI: 10.1109/TIT.1976.1055638.

Hankerson et al. 2004 = Darrel Hankerson, Alfred Menezes, and Scott Vanstone. Guide to Elliptic Curve Cryptography. Springer-Verlag New York, 2004. ISBN: 978-1441929297

Katz und Lindell 2021 = Jonathan Katz and Yehuda Lindell. Introduction to Modern Cryptography. CRC Press, 2021. ISBN: 978-1-4665-7027-6

Koblitz 1987 = Neal Koblitz. "Elliptic Curve Cryptosystems". In: Mathematics of Computation 48.177 (1987), pp. 203–209. ISSN: 00255718.

Miller 1986 = Victor S. Miller. "Use of Elliptic Curves in Cryptography". In: Advances in Cryptology — CRYPTO '85 Proceedings. Ed. by Hugh C. Williams. Berlin, Heidelberg: Springer Berlin Heidelberg, 1986, pp. 417–426. ISBN: 978-3-540-39799-1.

Rivest et al. 1978 = Ron L. Rivest, Adi Shamir, and Len Adleman. "A Method for Obtaining Digital Signatures and Public-Key Cryptosystems". In: Commun. ACM 21.2 (Feb. 1978), 120–126. ISSN: 0001-0782.
DOI: 10.1145/359340.359342.

Shor 1994 = Peter W. Shor. "Algorithms for quantum computation: discrete logarithms and factoring". In: Proceedings 35th Annual Symposium on Foundations of Computer Science. Ed. by Shafi Goldwasser. IEEE Computer Society Press, 1994, pp. 124–134.
DOI: 10.1109/SFCS.1994.365700.

KAROLINE MOSER hat im Wintersemester 2020/21 das damals neu eingeführte Bachelorstudium Industrial Data Science an der Montanuniversität Leoben begonnen. Den darauf aufbauenden Master hat sie vor Kurzem erfolgreich abgeschlossen. Derzeit arbeitet sie als wissenschaftliche Mitarbeiterin am Lehrstuhl für Zerstörungsfreie Prüfung der Technischen Universität München – mit dem Ziel, dort auch ihr Doktoratsstudium zu beginnen. Besonders viel Freude bereitet ihr die Lehrtätigkeit: zunächst als Studienassistentin in Leoben und nun als wissenschaftliche Mitarbeiterin in München. Karoline ist es ein großes Anliegen, das akademische Leben aktiv mitzugestalten. So engagierte sie sich in der ersten Studienvertretung von Industrial Data Science, im Senat der Montanuniversität und derzeit als Jahressprecherin bei PRO SCIENTIA – eine Aufgabe, die ihr besonders viel Freude bereitet. Karoline Moser ist seit 2023 PRO SCIENTIA Stipendiatin.