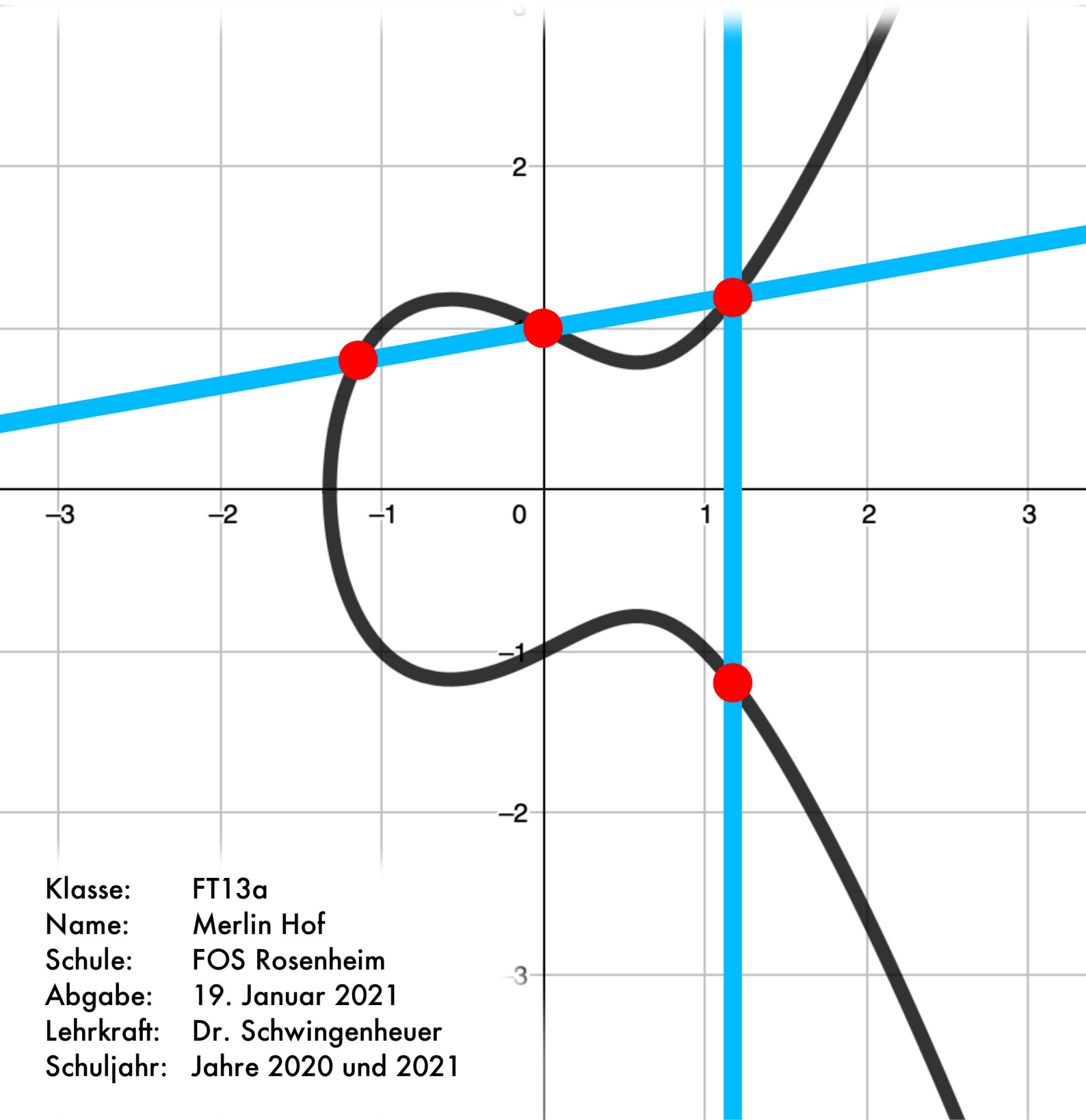
ELLIPTISCHE KURVEN
IN DER KRYPTOGRAPHIE



Klasse: FT13a
Name: Merlin Hof
Schule: FOS Rosenheim
Abgabe: 19. Januar 2021
Lehrkraft: Dr. Schwingenheuer
Schuljahr: Jahre 2020 und 2021

Inhalte

1 Einleitung

Diese Arbeit behandelt die sichere Verschlüsselung von Informationen mithilfe von elliptischen Kurven, die sich aufgrund ihrer besonderen mathematischen Eigenschaften sehr gut für die Anwendung in der Kryptographie eignen. Sensible Informationen können nur sicher über öffentliche Wege wie das Internet ausgetauscht werden, wenn sie vom Sender verschlüsselt und vom Empfänger später wieder entschlüsselt werden können, um zu verhindern, dass ein Anderer als der vorgesehene Empfänger Zugriff darauf bekommt. Die Kryptographie mit elliptischen Kurven bietet im Vergleich zu herkömmlichen Arten der Verschlüsselung wie beispielsweise dem sehr weit verbreiteten RSA Verfahren (benannt nach seinen Erfindern Rivest, Shamir und Adleman) den Vorteil, dass sie bei gleicher Länge der Schlüssel eine deutlich höhere Sicherheit ermöglicht, beziehungsweise dass bei gleicher Sicherheit nur ein sehr viel kürzerer Schlüssel benötigt wird. Das ist von großer Bedeutung, da ein längerer Schlüssel mit einem höheren Energieverbrauch für alle mit dem Schlüssel verbundenen Berechnungen einhergeht und vielen mobilen Geräten wie Smartphones oder Laptops nur eine begrenzte Menge an Energie, begrenzt durch die Kapazität ihres Akkus, zur Verfügung steht. Während beispielsweise bei RSA eine Schlüssellänge von 15.360 Bits benötigt wird, kann mit ECC (Elliptic Curve Cryptography) dasselbe Sicherheitslevel mit einem Schlüssel von nur 512 Bits Länge erreicht werden (Q6, "1.1 Motivation"). Da sich das Verhältnis der Länge des Schlüssels zwischen RSA und ECC mit steigendem Sicherheitsniveau sogar noch weiter vergrößert, erfreut sich die Verschlüsselung mit elliptischen Kurven trotz ihrer etwas aufwendigeren Implementation einer immer größer werdenden Beliebtheit.

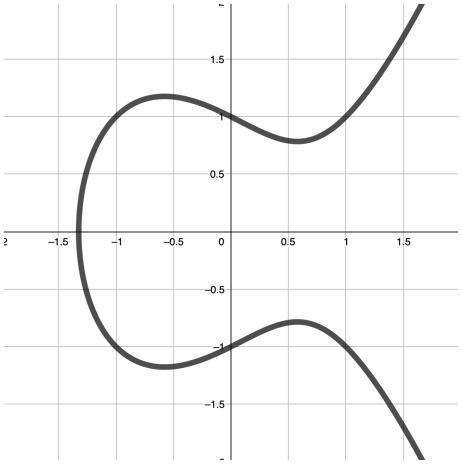
Im Verlauf dieser Arbeit werden zunächst elliptische Kurven allgemein und deren Mathematik, allem voran ihre für die Kryptographie sehr wichtigen gruppentheoretischen Eigenschaften beschrieben. Im darauffolgenden Abschnitt führt eine allgemeine Beschreibung in das Thema der Kryptographie ein, gefolgt von tiefgreifenden Konzepten der modernen Verschlüsselung. Zum Schluss befasst sich die Arbeit hauptsächlich mit der Effizienz und der Sicherheit der Kryptographie mit elliptischen Kurven, geht aber auch auf weitere wichtige Anwendungen in der Sicherheitsforschung ein.

Ziel dieser Arbeit ist das Erklären der Verschlüsselung von Informationen mithilfe elliptischer Kurven auf einem gut nachvollziehbaren, aber trotzdem professionell und wissenschaftlich gehaltenen Niveau, sodass sich die Leser mit Hintergründen in der Mathematik und Informatik als die Zielgruppe angesprochen fühlen. Trotzdem sollen aber auch geringe Vorkenntnisse ausreichen, um die theoretischen Prinzipien problemlos nachvollziehen zu können. Der Leser soll nach dem Lesen der Arbeit die Funktionsweise der Verschlüsselung mit elliptischen Kurven und alles Dazugehörige so umfangreich verstanden haben, dass er ein kryptographisches Verschlüsselungssystem mit elliptischen Kurven komplett nachbauen könnte und dabei keinerlei Fragen stellen müsste.

2 Elliptische Kurven

2.1 Allgemein

Elliptische Kurven sind Kurven dritter Ordnung der Form $y^2 = x^3 + ax + b$, wobei a und b Elemente des Körpers sind, über dem die Kurve definiert wurde. In der folgenden Abbildung ist die Kurve mit der Gleichung $y^2 = x^3 - x + 1$ über den reellen Zahlen $\mathbb{R}$ visualisiert.

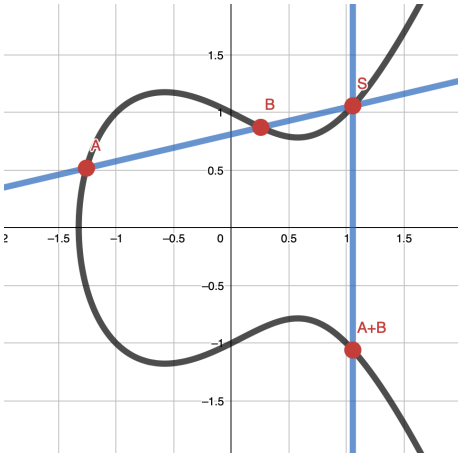


Durch das Quadrat in der Gleichung der Kurve muss zum Auflösen nach y die Wurzel gezogen werden, wobei es jeweils eine positive und eine negative Lösung gibt, was die Symmetrie zur x-Achse erklärt:
$y^2 = x^3 + ax + b \quad \Leftrightarrow$
$y = \pm\sqrt{x^3 + ax + b}$

Im Grunde ist eine elliptische Kurve die Wurzel des positiven Teils einer Funktion dritter Ordnung und ihre Spiegelung an der x-Achse. Die Graphen der elliptischen Kurven werden meist im zweidimensionalen Vektorraum $\mathbb{R}^2$ dargestellt. Für die Nützlichkeit in der Kryptographie ist es aber von entscheidender Bedeutung, elliptische Kurven in Vektorräumen über endlichen Körpern darstellen und darin mit ihnen rechnen zu können, worauf jedoch im Abschnitt "Endliche Körper" noch genauer eingegangen wird. Elliptische Kurven sind ursprünglich als Umkehrfunktion elliptischer Integrale entstanden (Q1, "Elliptische Kurve"), bevor 1985 ihre Nützlichkeit in der Kryptographie von Neal Koblitz und Victor Miller entdeckt wurde (Q2, "Introduction"). In der Kryptographie können nur jene elliptische Kurven verwendet werden, welche keine Singularitäten, also Punkte, an denen sich abrupt die Richtung ändert, haben. Das ist erfüllt, wenn ihre Diskriminante $-4a^3 - 27b^2 \neq 0$ ist.

2.2 Gruppentheorie

Auf elliptischen Kurven ist geometrisch eine Addition definiert (Q1, "Elliptische Kurve"), mittels welcher zwei Punkte auf einer elliptischen Kurve einem dritten Punkt zugeordnet werden können. In der folgenden Abbildung ist die Addition der beiden Punkte A und B illustriert.



Die Punkte A und B auf der Kurve werden addiert, indem der dritte Schnittpunkt der Sekante durch die beiden zu addierenden Punkte und der elliptischen Kurve an der x-Achse gespiegelt wird. Dieser Punkt entspricht der Summe von A und B. Die Steigung m der Sekante durch A und B ist nach der allgemeinen Form $m = \frac{\Delta y}{\Delta x}$ zu berechnen, jedoch nur, wenn $x_A \neq x_B$, um eine Division mit null zu verhindern.
$m = \frac{\Delta y}{\Delta x} = \frac{y_A - y_B}{x_A - x_B}$

Die Koordinaten des Punktes $A + B(x_{AB}|y_{AB})$ lassen sich mit den folgenden Formeln berechnen. Es werden keine Informationen zu der Kurve benötigt, auf der die Punkte liegen, da es für jede Kombination aus x- und y- Koordinaten zweier Punkte nur eine mögliche Kurve geben kann.
$x_{AB} = m^2 - x_A - x_B$
$y_{AB} = -y_A - m(x_{AB} - x_A)$

Ein Punkt auf einer elliptischen Kurve kann auch mit sich selbst addiert werden, was Punktverdoppelung genannt wird. Die Steigung einer Kurve in einem Punkt $P(x_P|y_P)$ ist die Steigung der Tangente in diesem Punkt, bei deren Berechnung der Wert a aus der Gleichung der elliptischen Kurve benötigt wird. Falls $y_P \neq 0$, so gilt für die Steigung m:
$m = \frac{3x_P^2 + a}{2y_P}$

Die Berechnung der Koordinaten des Verdoppelungspunktes $2P$ mithilfe der errechneten Steigung erfolgt auf gleichem Wege wie bei der Addition zweier unterschiedlicher Punkte.

Ein Punkt auf einer elliptischen Kurve kann aber auch mit einem ganzzahligen Faktor multipliziert werden, da die Multiplikation nur eine andere Art der Darstellung für eine wiederholte Addition ist. Soll ein Punkt P beispielsweise mit vier multipliziert werden, so wird das Ergebnis als $4P$ (oder $P + P + P + P$) bezeichnet. Für die Multiplikation eines Punktes mit vier wird zuerst der Punkt P verdoppelt, was den Punkt $2P$ ergibt, welcher daraufhin wieder mit P addiert wird und so weiter, bis am Schluss der gesuchte Punkt $4P$ erreicht ist. Eine ausschlaggebende Besonderheit elliptischer Kurven ist die, dass alternativ auch der Punkt $2P$ verdoppelt werden

kann, um auf das Ergebnis von $4P$ zu kommen. Diese Besonderheit spielt bei der Effizienz in der Kryptographie eine fundamentale Rolle, ohne die elliptische Kurven für die Verschlüsselung unbrauchbar wären.

Die Addition von Punkten auf einer elliptischen Kurve erfüllt alle drei Axiome der Gruppentheorie und hat somit eine Gruppenstruktur. Das erste Axiom, die Assoziativität, ist grundlegend für die Eignung elliptischer Kurven als Schlüsselelement in der Kryptographie. Liegt eine Assoziativität vor, so muss $P+(Q+W)$ zum selben Ergebnis führen wie $(P+Q)+W$. $P(x_P|y_P)$, $Q(x_Q|y_Q)$ und $W(x_W|y_W)$ sind Punkte auf einer elliptischen Kurve. Mit der Annahme, dass $x_P \neq x_Q \neq x_W$, wird im Folgenden der Beweis für die Richtigkeit der Aussage $P + (Q + W) = (P + Q) + W$ ausgeführt.

1. $R1 := Q + W$
$m_{QW} = \frac{y_Q - y_W}{x_Q - x_W}$
$x_{R1} = m_{QW}^2 - x_Q - x_W$
$y_{R1} = -y_Q + 2 * m_{QW} * x_Q - m_{QW}^3 + m_{QW} * x_W$

2. $R2 := P + Q$
$m_{PQ} = \frac{y_P - y_Q}{x_P - x_Q}$
$x_{R2} = m_{PQ}^2 - x_P - x_Q$
$y_{R2} = -y_P + 2 * m_{PQ} * x_P - m_{PQ}^3 + m_{PQ} * x_P$

3. $S1 := P + R1$
$m_{PR1} = \frac{y_P + y_Q - 2 * m_{QW} * x_Q + m_{QW}^3 - m_{QW} * x_W}{x_P - m_{QW}^2 + x_P + x_W}$
$x_{S1} = m_{PR1}^2 - m_{QW}^2 - x_P - x_Q - x_W$
$y_{S1} = -y_P + 2 * m_{PR1} * x_P - m_{PR1}^3 + m_{PR1} * m_{QW}^2 + m_{PR1} * x_Q + m_{PR1} * x_W$

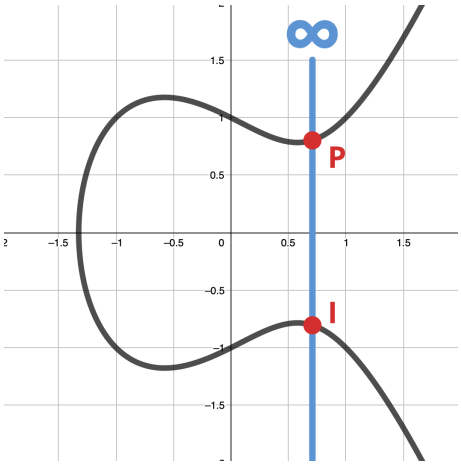
4. $S2 := W + R2$
$m_{WR2} = \frac{y_W + y_P - 2 * m_{PQ} * x_P + m_{PQ}^3 - m_{PQ} * x_P}{x_W - m_{PQ}^2 + x_P + x_Q}$
$x_{S2} = m_{WR2}^2 - m_{PQ}^2 - x_P - x_Q - x_W$
$y_{S2} = -y_W + 2 * m_{WR2} * x_W - m_{WR2}^3 + m_{WR2} * m_{PQ}^2 + m_{WR2} * x_P + m_{WR2} * x_Q$

5. $S1 = S2$
Wahre Aussage $\rightarrow$ Die Addition von unterschiedlichen Punkten auf elliptischen Kurven ist assoziativ.

Das additive neutrale Element elliptischer Kurven ist als ein Punkt im Unendlichen definiert. Das Ergebnis der Addition zweier Punkte auf einer elliptischen Kurve mit demselben x-Wert ist als ∞, also dem neutralen Element, definiert, da die Sekante durch die beiden zu addierenden Punkte eine Senkrechte mit unendlicher Steigung ist, die die elliptische Kurve in keinem dritten Punkt mehr schneidet. Selbiges gilt für die Verdoppelung eines Punktes P mit $y_P = 0$, da die Steigung der Kurve bei $y = 0$ undefiniert ist und die Berechnung der Steigung der Tangente aufgrund einer Division mit null nicht möglich ist. (Q1, "Addition zweier verschiedener Punkte")

Jeder Punkt $P(x_P|y_P)$ auf einer elliptischen Kurve hat einen zu ihm inversen Punkt $I(x_P| - y_P)$. Bei der Addition der zueinander inversen Punkte P und I ist das Ergebnis das neutrale Element,

also ein Punkt im Unendlichen, da das neutrale Element als das Ergebnis einer Addition zweier Punkte mit demselben x-Wert definiert ist.



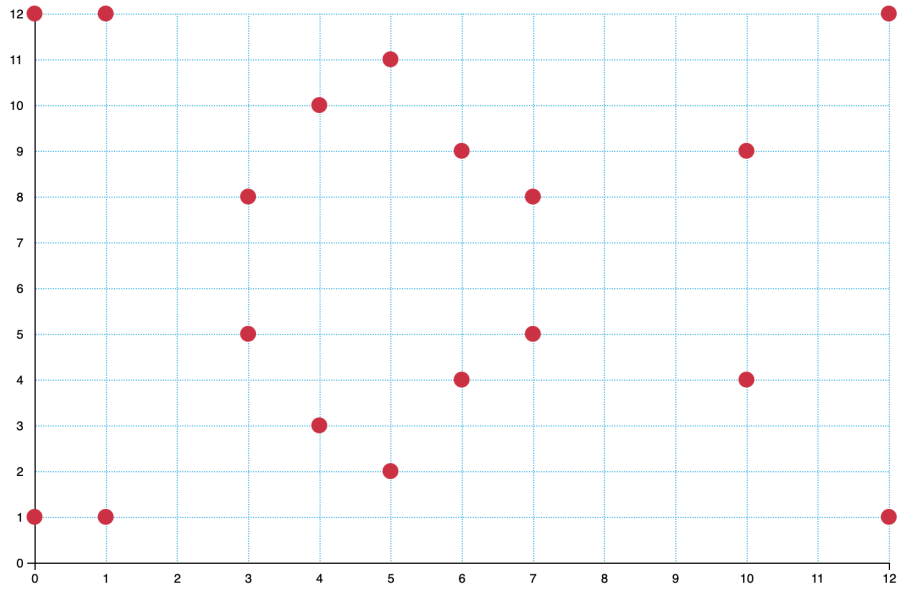
2.3 Endliche Körper

In der Kryptographie können ausschließlich elliptische Kurven über endlichen Körpern verwendet werden, da vermieden werden muss, dass in x- und y-Richtung unendlich viele Zahlen liegen, wie es bei elliptischen Kurven über dem Körper der reellen Zahlen $\mathbb{R}$ der Fall ist. Computer müssten mit unendlich hoher Präzision rechnen können, um in der Kryptographie elliptische Kurven über dem Körper der reellen Zahlen verwenden zu können, da sich kleine Rundungsfehler und andere Ungenauigkeiten sehr schnell aufsummieren und zu einem komplett falschen Ergebnis führen würden. Um das Auftreten dieses Problems zu verhindern, werden alle Berechnungen wie die Addition und Multiplikation von Punkten ausschließlich über endlichen Körpern $\mathbb{F}_p$ durchgeführt, da diese nur ganzzahlige Werte von null bis $p-1$ enthalten, sodass ein Vektorraum über dem endlichen Körper $\mathbb{F}_{13}$ beispielsweise den Punkt $E(2|5)$ enthält, den Punkt $N(0.2|0.9)$ jedoch nicht. Die Charakteristik p des endlichen Körpers ist eine Primzahl, sodass aus dem Körper ein Restklassenkörper wird. In der Praxis sind nur endliche Körper $\mathbb{F}_p$ mit sehr großen Primzahlen für p sinnvoll, da solche mit kleinen Werten nur eine geringe Anzahl an Punkten auf der über ihnen definierten elliptischen Kurve haben, wodurch keine sichere Verschlüsselung möglich wäre.

Ein weiterer Grund, elliptische Kurven über endlichen Körpern für die nötigen Berechnungen zu verwenden, ist neben der Gefahr von fatalen Rundungsfehlern der, dass es sehr ineffizient ist, Computer mit extrem großen Zahlen rechnen zu lassen, da diese mehr Arbeitsspeicher benötigen und die CPU deutlich stärker belasten, wodurch mehr Zeit benötigt wird, um das Ergebnis zu berechnen. Als Grund für die Verwendung endlicher Körper kann festgehalten werden, dass es schlicht ineffizient ist, Computer mit Zahlen mit sehr vielen gültigen Ziffern, also sehr großen oder sehr exakten Zahlen arbeiten zu lassen. Die Berechnung der Summe zweier Punkte auf einer elliptischen Kurve über einem endlichen Körper durchzuführen, löst dieses Problem, das es ansonsten bei elliptischen Kurven über $\mathbb{R}$ gäbe.

Ein Punkt auf einer elliptischen Kurve ist im Körper enthalten, wenn $(y^2 = x^3 + ax + b) \bmod p$ wahr ist, wobei $y^2 = x^3 + ax + b$ die Gleichung der Kurve und p die Charakteristik des Restklassenkörpers $\mathbb{F}_p$ ist. In der folgenden Abbildung ist die Kurve der Form $y^2 = x^3 - x + 1$

über dem endlichen Körper $\mathbb{F}_{13}$ visualisiert. Die ursprüngliche Symmetrie der Kurve ist noch immer klar erkennbar.



Beispiele:
für $x = 4:\ y^2 \bmod 13 = 4^3 - 4 + 1 \bmod 13 \rightarrow$ trifft für $y = 3$ und $y = 10$ zu
für $x = 5:\ y^2 \bmod 13 = 5^3 - 5 + 1 \bmod 13 \rightarrow$ trifft für $y = 2$ und $y = 11$ zu
für $x = 8:\ y^2 \bmod 13 = 8^3 - 8 + 1 \bmod 13 \rightarrow$ trifft für keine Werte für y zu

Um Punkte auf einer elliptischen Kurve über einem endlichen Körper zu addieren, können dieselben Formeln wie bei der Addition zweier Punkte auf elliptischen Kurven über $\mathbb{R}$ verwendet werden. Für die Berechnung der Steigung m sind jedoch weitere Schritte nötig, da die Steigung in den meisten Fällen keine ganze Zahl ist. Die Steigung muss jedoch auch eine ganze, im Körper enthaltene Zahl sein, um sinnvoll mit ihr rechnen zu können. Sollen beispielsweise die Punkte $H(3|5)$ und $J(7|8)$ auf einer elliptischen Kurve über $\mathbb{F}_{13}$ addiert werden, so ergibt sich für die Steigung:
$m = \frac{\Delta y}{\Delta x} \bmod p = \frac{5-8}{3-7} \bmod 13 = \frac{3}{4} \bmod 13 = (3 * 4^{-1}) \bmod 13.$

4^{-1} ist das multiplikative modulare Inverse von 4, worauf im nächsten Abschnitt genauer eingegangen wird. In diesem Fall ist 10 das gesuchte Inverse, da $(4 * 10) \bmod 13 = 1$. Für die Steigung der Sekante und die x- und y-Koordinaten des Punktes $L(x_L|y_L) = H + J$ ergeben sich somit:
$m = (3 * 10) \bmod 13 = 4$
$x_L = (m^2 - x_H - x_J) \bmod 13 = (16 - 3 - 7) \bmod 13 = 6$
$y_L = (-y_H - m(x_L - x_H)) \bmod 13 = (-5 - 4(6 - 3)) \bmod 13 = -17 \bmod 13 = 9$

Ein Blick auf die obige Grafik bestätigt, dass der Punkt $L(6|9)$ tatsächlich auf dem Graphen der Kurve liegt. Bei der Verdoppelung eines Punktes $M(x_M|y_M)$ ist ein ähnliches Vorgehen notwendig, da dieselben Formeln für die Koordinaten des Verdoppelungspunktes verwendet werden können und bei der Steigung wiederum das modulare Inverse des Nenners mit dem Zähler multipliziert werden muss, sodass gilt:

$m = ((3x_M^2 + a) * (2y_M)^{-1}) \bmod p$

Mithilfe dieser Methode können Punkte auf elliptischen Kurven über endlichen Körpern addiert, beziehungsweise verdoppelt werden, ohne dass dabei ihre Gruppenstruktur mit den für die Kryptographie unverzichtbaren Eigenschaften zerstört wird.

2.4 Modulare Inverse

Modulare Inverse müssen nicht nur wie im vorhergegangenen Abschnitt für die Berechnung der Summe zweier Punkte auf einer elliptischen Kurve über einem endlichen Körper bestimmt werden, auch in anderen Bereichen der Verschlüsselung besteht die Möglichkeit, dass das modulare Inverse einer Zahl gefunden werden muss. Multiplikative Inverse werden als die Zahl Z, deren Inverses gefunden werden soll, hoch -1 dargestellt. Eine Zahl multipliziert mit ihrem Inversen muss eins, das multiplikative neutrale Element, ergeben. So gilt für multiplikative Inverse ohne Modulo: $Z * Z^{-1} = 1$. Für beispielsweise $Z = 4$ ist $Z^{-1} = \frac{1}{4}$, da $4 * \frac{1}{4} = 1$.

In der Kryptographie mit elliptischen Kurven und auch in vielen anderen Anwendungsbereichen müssen jedoch meist multiplikative Inverse mit einem Modulo, sogenannte modulare Inverse, gefunden werden. Es gilt: $(Z * Z^{-1} = 1) \bmod m$. Z^{-1} muss eine ganze Zahl sein, da der Modulooperator nur bei ganzen Zahlen anwendbar ist. Für $Z = 4$ und $m = 7$ ist das modulare Inverse $Z^{-1} = 2$, da $(4 * 2) \bmod 7 = 1$.

Es gibt viele unterschiedliche Verfahren, um ein modulares Inverses einer Zahl zu finden, ohne dabei jede Möglichkeit ausprobieren zu müssen, was bei sehr kleinen Zahlen aufgrund der Effizienz und des Aufwands für die Implementierung jedoch durchaus eine sinnvolle Methode sein kann. Eine sehr effiziente Methode, welche bei beliebig großen Zahlen anwendbar ist, ist ein Verfahren nach Carl Friedrich Gauß. Wichtig anzumerken hierbei ist, dass der Algorithmus nur funktioniert, wenn es sich bei m um eine Primzahl handelt, was jedoch praktisch keine Einschränkung darstellt, da die Addition von Punkten über endlichen Körpern auch nur das Finden eines modularen Inversen mit einer Primzahl erfordert. Um diese Methode anzuwenden, wird zuerst ein Bruch mit einer eins im Zähler und der Zahl Z, deren modulares Inverses gefunden werden soll, im Nenner aufgestellt. Dieser Bruch wird solange mit einem Faktor F multipliziert, bis der Bruch eine ganze Zahl ergibt, bei welcher es sich dann um das gesuchte Inverse handelt. F ist der kleinste Wert, der mit dem Nenner multipliziert größer oder gleich m ist. Nach jeder Multiplikation mit F werden Zähler und Nenner jeweils durch mod m reduziert und anschließend so weit wie möglich gekürzt. (Methode angelehnt an Q3 und Q4)

Ein Beispiel für die Berechnung von Z^{-1} für $Z = 7$ und $m = 31$:
$\frac{1}{7} \equiv \frac{5}{4} \equiv \frac{9}{1}$

Im ersten Schritt wurde $\frac{1}{7}$ mit $F = 5$ multipliziert, da 5 der kleinste ganzzahlige Wert für F ist, sodass $F * 7 \geq 31$ wahr ist. Nenner und Zähler des Ergebnisses $\frac{5}{35}$ werden daraufhin jeweils mit mod 31 reduziert, was in den Bruch $\frac{5}{4}$ resultiert. Der nächste Wert für F ist 8, sodass sich der Bruch $\frac{40}{32}$ ergibt, woraus reduziert $\frac{9}{1}$ wird. Nun steht im Nenner eine eins, weshalb die Berechnung abgeschlossen ist und der Wert im Zähler das Ergebnis Z^{-1} für $Z = 7$ und $m = 31$ ist. Eine kurze Gegenrechnung bestätigt die Richtigkeit des Ergebnisses:

$7 * 9 \bmod 31 = 63 \bmod 31 = 1$

Der Faktor F kann einfach berechnet werden, indem m durch die Zahl im Nenner geteilt, und das Ergebnis auf eine ganze Zahl aufgerundet wird. Im ersten Schritt beim obigen Beispiel wäre das $\frac{31}{7} \approx 4.43$, was aufgerundet 5 ergibt, beim zweiten Schritt $\frac{31}{4} = 7.75$, was aufgerundet 8 ergibt.

3 Kryptographie

3.1 Allgemein

Verschlüsselung oder Kryptographie ist im Allgemeinen die Umwandlung von Text in eine für Menschen unlesbare, zufällig wirkende Anordnung von Zeichen, die nur mit dem richtigen Zugangsschlüssel wieder leserlich gemacht werden kann. Verschlüsselung ist aus der heutigen Zeit kaum mehr wegzudenken, weshalb es umso wichtiger ist, auf ein Verschlüsselungssystem zurückgreifen zu können, welches schnell, sicher und effizient ist. Anwendungen moderner Verschlüsselung sind unter anderem Apps zur Kommunikation wie WhatsApp oder Signal, Computerprogramme, die sichere Verbindungen zu Servern über das Internet aufbauen müssen wie Onlinespiele, aber auch bei digitalen Bezahlsystemen wie Apple Pay oder PayPal kann auf eine Verschlüsselung nicht verzichtet werden. Ohne eine Verschlüsselung wäre die große Mehrheit aller Internetanwendungen komplett transparent und jeder könnte unerlaubt Zugriff auf alle öffentlich gespeicherten Nutzerdaten wie Passwörter, Bilder oder Textnachrichten bekommen und diese lesen, verändern und im schlimmsten Fall auch missbrauchen.

Wird beispielsweise eine Nachricht in Signal verschickt, so wird diese vor dem Senden auf dem Gerät des Senders mit einem Schlüssel verschlüsselt und somit unleserlich gemacht. Diese unlesbare Nachricht wird dann für prinzipiell jeden öffentlich einsehbar über das Internet zum Empfänger weitergeleitet. Der Empfänger ist im Besitz des gleichen Schlüssels wie der Sender und kann somit die Nachricht wieder entschlüsseln und auf auf dem Display anzeigen. Beide Chatpartner haben denselben Schlüssel, mit dem sie ihre Nachrichten gegenseitig ver- und entschlüsseln. Das große Problem, das es zu lösen gilt, ist, wie beide Chatpartner anfangs an den exakt gleichen Schlüssel kommen, ohne dass der Schlüssel beim Schlüsselaustausch abgefangen werden kann, wodurch die Nachrichten für denjenigen, der den Schlüssel abgefangen hat, problemlos zu entschlüsseln wären. Für dieses Problem gibt es mehrere Lösungsansätze, aber der wohl bekannteste und eleganteste ist der Schlüsselaustausch nach Diffie-Hellman, auf den im Abschnitt "Diffie-Hellman Schlüsselaustausch" genauer eingegangen wird. Elliptische Kurven werden in der Kryptographie hauptsächlich zum sicheren Austausch der Schlüssel verwendet. Auch wenn elliptische Kurven zum Verschlüsseln selbst verwendet werden können, übernehmen die eigentliche Verschlüsselung und Entschlüsselung einer Nachricht meist sogenannte Cipher, die aufgrund ihrer Wichtigkeit einen eigenen Abschnitt "Cipher" in dieser Arbeit bekommen haben. Zuerst müssen jedoch einige wichtige Grundlagen geklärt werden, unter anderem, wie Hashfunktionen funktionieren und wie Computer überhaupt an "zufällige" Zahlen kommen, die beim Diffie-Hellman Schlüsselaustausch fundamental für den sicheren Austausch der Schlüssel sind.

3.2 Hashfunktionen

Ein wichtiger Bestandteil der Kryptographie sind kryptographische Hashfunktionen. Eine Hashfunktion weist ihrem Argument eine spezielle Folge an Zeichen mit fester Länge zu, die sich bei jeder noch so kleinen Veränderung des Arguments von Grund auf komplett verändert. Eine weit verbreitete Hashfunktion ist die MD5-Funktion, die bereits in einen Großteil aller UNIX-basierten Betriebssysteme wie macOS, Linux, iOS oder Android integriert ist. In folgendem Beispiel ist zu sehen, wie die Hashfunktion für zwei minimal differierte Argumente komplett unterschiedliche Hashs ausgibt.
MD5("Seminararbeit.2020") = ad2f07f6098f01fcebc45eeb6c770876
MD5("Seminararbeit.2021") = 29b5ee9d441aacb4ad4845fbdb38a5e8

Gute Hashfunktionen müssen mehrere Anforderungen erfüllen, um in der Kryptographie sinnvoll eingesetzt werden zu können. Zum einen müssen sie konsistent sein, sie müssen also für dasselbe Argument immer denselben Hash produzieren, zum anderen dürfen sie nur einen sehr geringen Prozentsatz an Kollisionen haben, was heißt, dass sie nur sehr selten bis gar nicht für unterschiedliche Argumente denselben Hash ausgeben dürfen. Die dritte Bedingung, die kryptographische Hashfunktionen erfüllen müssen, ist die, dass aus dem Hash keine Rückschlüsse auf das Argument gezogen werden können dürfen, weshalb sich der Hash auch bei einer geringen Änderung des Arguments grundlegend ändern sollte.

In folgendem Szenario wird die Sinnhaftigkeit solcher Hashfunktionen ersichtlich: Ein Nutzer erstellt ein Konto auf einer Webseite, indem er einen Nutzernamen und ein zugehöriges Passwort eingibt. Diese Informationen müssen auf dem Server gespeichert werden, um bei einer zukünftigen Anmeldung auf der Webseite die Richtigkeit der eingegebenen Daten überprüfen zu können. Nun wäre es jedoch ein großes Sicherheitsrisiko, alle Nutzernamen mit den zugehörigen Passwörtern zusammen in einer Datenbank zu speichern, da ein Hacker, dem es gelungen ist, Daten aus dieser Datenbank zu extrahieren, uneingeschränkten Zugriff auf alle Nutzerkonten hätte. Um das zu verhindern, werden in der Datenbank lediglich der Nutzername und der Hash des zugehörigen Passworts gespeichert, weshalb die Website bei einer Anmeldung das eingegebene Passwort hashen und es dann mit dem gespeicherten Hash in der Datenbank vergleichen kann.

Hashfunktionen haben neben der im Beispiel beschriebenen noch viele andere Einsatzbereiche, in denen sie für eine höhere Sicherheit benötigt werden. Oft ist ihr Einsatz nicht essentiell für die Funktion eines Systems, aber essentiell für ein hohes Level an Sicherheit.

3.3 Pseudozufällige Zahlen

Die Möglichkeit zur Generierung zufälliger Zahlen bildet nicht nur in Bezug auf elliptische Kurven die Basis für eine sichere Verschlüsselung, sondern auch für viele andere kryptographische Verfahren, in die zufällige Zahlen in sicherheitsrelevante Berechnungen einfließen. Deshalb ist es für die Sicherheit extrem wichtig, eine Reihe an "zufälligen" Zahlen generieren zu können, aus denen nur sehr schlecht ein Muster abgeleitet werden kann, mithilfe dessen Rückschlüsse auf die nächste Zahl gezogen werden kann. Eine schlechte Reihe an zufälligen Zahlen wäre $\{1, 1, 2, 3, 5, 8, 13, 21, 34\}$, da das Muster hier einfach als Summe der vorherigen zwei Zahlen zu erkennen ist und mit nur zwei aufeinanderfolgende Zahlen aus dieser Reihe die nächste

Zahl bestimmt werden kann. Da es in der Physik keinen *echten* Zufall gibt, sind alle von Computern generierten zufälligen Zahlen nur das Ergebnis von komplexen Gleichungen, die ihre Ergebnisse in Abhängigkeit ihrer vorherigen Ergebnisse bestimmen, oder aber auch gemessene Werte chaotischer physikalischer Systeme, die zwar theoretisch deterministisch sind und berechnet werden können, praktisch jedoch nicht ohne einen unermesslich großen Aufwand. Jene zufälligen Zahlen, welche mithilfe rein mathematischer Gleichungen berechnet werden, sind um einiges einfacher vorherzusagen und werden deshalb pseudozufällig genannt, woher auch die Abkürzung für die Algorithmen stammt, die pseudozufällige Zahlen generieren: **PRNG** für **P**seudo **R**andom **N**umber **G**enerator.

Die unten gezeigte iterative Folge ist eine einfache, aber effektive Methode zur Berechnung einer Reihe an pseudozufälligen Zahlen w:
$w_{i+1} = (a * w_i + c)\ mod\ m$

w_0 wird zu Anfang festgelegt und ist der Startwert, welcher oft auch als "Seed" bezeichnet wird. Viele pseudozufällige Zahlengeneratoren verwenden die Zahl der vergangenen Sekunden seit dem 01.01.1970, auch bekannt als Unixzeit, als Startwert. Es können aber auch gemessene, physische Eingänge wie "zufällige" Schwankungen der Radiowellen als Startwert verwendet werden, wozu zwar spezielle Hardware benötigt wird, die aber viele mobile Computer wie Smartphones oder Laptops schon für den Radioempfang besitzen. a, c und m sind Konstanten, der Modulo grenzt die Werte der generierten Zahlen zwischen null und $m - 1$ ein. Bei geeigneten Konstanten (hier: $a = 12,\ c = 21,\ m = 50$) und dem Startwert $w_0 = 7$ ergeben sich für w_1 bis w_{10} die folgenden Werte:
$w = \{5, 31, 43, 37, 15, 1, 33, 17, 25, 21\}$

Diese Zahlen wirken komplett zufällig und sind daher für die meisten Anwendungen sicher sehr gut geeignet. Die Gleichung zur Berechnung der Werte und die Konstanten der meisten populären Zufallszahlengeneratoren sind jedoch teilweise oder sogar komplett öffentlich, weshalb aus einer Reihe der von der Gleichung erzeugten Zahlen auf die nächste "zufällige" Zahl geschlossen werden kann. Aus diesem Grund sollte bei dieser Methode der Startwert relativ oft erneuert werden. Für eine etwas höhere Sicherheit in der Kryptographie können deshalb auch "echte" zufällige Zahlen verwendet werden, aus denen auch aus einer Reihe an Werten praktisch keine Schlüsse auf die nächste Zahl gezogen werden können. Der Computer kann dafür beispielsweise äußere Umwelteinflüsse wie die eben genannten Radiowellen verwenden, aber auch der Luftdruck, die Farben bestimmter Pixel von der Kamera oder Bewegungen der Maus, beziehungsweise Änderungen der Neigung von mobilen Geräten können einbezogen werden, um Zahlen zu generieren, die von einem Angreifer nicht vorhersagbar sind. In der Theorie *könnten* natürlich auch diese berechnet und vorhergesagt werden, mit unserer aktuellen Technologie und unserem Wissen über die Physik ist das jedoch praktisch nicht umsetzbar. Intel, AMD und auch andere CPU-Hersteller verbauen in ihren Prozessoren Hardware-Zufallsgeneratoren, die mit auf Entropiequellen basierenden Systemen unvorhersagbar zufällige Zahlen versprechen.

Das alles hindert jedoch Schadsoftware nicht daran, diese Werte einfach auf dem Gerät selbst auszulesen und demjenigen, der die Verschlüsselung umgehen möchte, mitzuteilen. Auch kann Schadsoftware so direkt die Schlüssel für beispielsweise WhatsApp-Chatverläufe auslesen und

so die Nachrichten auf dem Gerät entschlüsseln und an den Angreifer versenden. Apple geht gegen diese Art der Schadsoftware vor und verwendet seit einigen Jahren in iOS sogenannte App-Groups, wodurch Apps desselben Entwicklers untereinander Daten austauschen können, jedoch unter keinen Umständen auf die Daten wie eben Schlüssel oder Chatverläufe einer App eines anderen Entwicklers zugreifen können. So kann zwar theoretisch die FaceBook-App auf die WhatsApp Daten zugreifen, da WhatsApp von FaceBook vertrieben wird, nicht aber nicht von FaceBook entwickelte Software. (Ideen und Konzepte dieses Abschnittes angelehnt an Q7)

3.4 Diffie-Hellman Schlüsselaustausch

Der Diffie-Hellman Schlüsselaustausch ist benannt nach Whitfield Diffie und Martin Hellman. Die beiden Mathematiker entwickelten das Verfahren zusammen mit Diffies Studenten Ralph Merkle im Jahr 1976 (Q5, "Martin Hellman").

Der wichtigste Bestandteil des Diffie-Hellman Schlüsselaustausches ist eine sogenannte Falltür- oder Einwegfunktion. Das ist eine Funktion, die zwar leicht zu berechnen, aber fast unmöglich umzukehren ist, es also keine effiziente Methode dafür gibt, vom Ergebnis wieder zurück auf das Argument zu schließen. Ein Beispiel für eine relativ einfache Falltürfunktion ist der Modulo. $31415 \; mod \; 12 = 11$ ist leicht zu berechnen, es ist aber nicht möglich, aus dem Ergebnis 11 und mod 12 die eingesetzte Zahl 31415 eindeutig zu bestimmen, da beispielsweise 23, 35 und 47 zum selben Ergebnis führen würden. Im Gegensatz dazu kann man bei der Addition $3 + 9 = 12$ aus 12 und 9 die eingesetzte Zahl 3 eindeutig bestimmen, weshalb die Addition zweier Zahlen keine Falltürfunktion darstellt.

Das Prinzip von Diffie-Hellman ist der Schlüsselaustausch mit einem öffentlichen und einem privaten Schlüssel. Wollen beispielsweise zwei Messengerdienste wie WhatsApp oder Signal auf zwei verschiedenen Geräten an denselben Schlüssel zum Verschlüsseln und Entschlüsseln der Nachrichten kommen, so können sie diesen nicht einfach über das Internet austauschen, da alles, was über das Internet gesendet wird, öffentlich und für potentielle Angreifer sichtbar ist. Um dieses Problem zu lösen, erstellen die Apps jeweils einen privaten Schlüssel, aus dem sie mithilfe einer Falltürfunktion einen öffentlichen Schlüssel generieren, der dann öffentlich über das Internet mit der anderen App ausgetauscht werden kann. Von diesem öffentlichen Schlüssel können keine Rückschlüsse auf den privaten Schlüssel gezogen werden, da der öffentliche Schlüssel eben mit einer Falltürfunktion erzeugt wurde, deren Stärke ja genau darin liegt, ihr Argument in ihrem Ergebnis unkenntlich zu machen. Aus dem öffentlichen Schlüssel des Anderen und dem eigenen privaten Schlüssel kann ein gemeinsamer, gleicher Schlüssel berechnet werden. Die privaten Schlüssel werden nie geteilt und bleiben immer auf dem Gerät, auf dem sie generiert wurden, da jeder, der an den privaten Schlüssel kommt, die verschlüsselten Nachrichten ohne Probleme entschlüsseln kann. Aufgrund der Verwendung einer Falltürfunktion kann ein Angreifer aus den öffentlich geteilten Information keine Rückschlüsse auf die privaten Schlüssel der Chatpartner ziehen und kommt somit auch nicht an den gemeinsamen Schlüssel zum Verschlüsseln und Entschlüsseln der Nachrichten. Zum besseren Verständnis des Diffie-Hellman Schlüsselaustauschs wird im Folgenden kurz die Funktionsweise einer einfacheren Methode zum Schlüsselaustausch erklärt:

1. Beide Seiten, beispielsweise zwei Nachrichtenapps auf zwei Smartphones, kreieren jeweils zufällig ihren privaten Schlüssel n, in diesem Beispiel: $n_1 = 13$ und $n_2 = 25$.
2. Nun kann jeweils der öffentliche Schlüssel o mit der Falltürfunktion $o(n) = a^n \ mod \ b$ mit willkürlich gewählten Werten für a und b, die jedoch auf beiden Seiten identisch sein müssen, berechnet werden. So ergibt sich mit $a = 5$ und $b = 46$ für $o_1 = 21$ und $o_2 = 33$.
3. Diese öffentlichen Schlüssel werden nun öffentlich über das Internet mit der jeweils anderen App ausgetauscht.
4. Mit dem eigenen privaten Schlüssel und dem öffentlichen Schlüssel des Anderen kann der gemeinsame Schlüssel S berechnet werden, der auf beiden Seiten exakt gleich ist, ohne dass dieser direkt über das Internet ausgetauscht wurde und ohne dass er aus den öffentlich zugänglichen Daten effizient und eindeutig berechnet werden kann. $S_1 = o_2^{n_1} \ mod \ 46 = 33^{13} \ mod \ 46 = 15$, $S_2 = o_1^{n_2} \ mod \ 46 = 21^{25} \ mod \ 46 = 15$

Die hier gewählten Werte für n, a und b wurden zu Demonstrationszwecken gewählt und würden bei einer realen Anwendung in der Kryptographie nur eine sehr geringe Sicherheit bieten. Mit deutlich größeren Werten, hauptsächlich für n und b, kann die Sicherheit dieser Methode drastisch erhöht werden, was jedoch auch die RAM-Belegung und die CPU-Auslastung bedeutend erhöht. Es muss also ein guter Kompromiss zwischen Systembelastung und Sicherheit gefunden werden. In der Kryptographie mit elliptischen Kurven ist die Falltürfunktion deutlich effizienter, wodurch der gesamte Algorithmus effizienter wird und kein Kompromiss zwischen Systembelastung und Sicherheit mehr gefunden werden muss, da selbst mit relativ geringem Rechenaufwand ein hohes Level an Sicherheit erzielt werden kann.

3.5 Schlüsselaustausch mit elliptischen Kurven

Die Multiplikation eines Punktes auf einer elliptischen Kurve stellt eine Falltürfunktion dar, da es einfach ist, $n * P$ zu berechnen, es aber keine effiziente Methode gibt, n aus nP und P zu berechnen. Der Diffie-Hellman Schlüsselaustausch im Bezug auf elliptische Kurven heißt *ECDH* für *Elliptic Curve Diffie-Hellman*. Dieser Algorithmus ist sehr beliebt, da er sowohl ausgesprochen effizient als auch sehr sicher arbeitet. Der private Schlüssel n ist eine "zufällig" generierte Zahl, die mit einem gemeinsamen Startpunkt P auf einer elliptischen Kurve über einem endlichen Körper multipliziert wird. Alle Konstanten wie der Startpunkt, die Gleichung der elliptischen Kurve oder die Charakteristik des endlichen Körpers sind auf beiden Seiten identisch. Der aus der Multiplikation resultierende Punkt nP ist der öffentliche Schlüssel O, für den gilt:
$O_1 = n_1 P$
$O_2 = n_2 P$

Diese öffentlichen Schlüssel können jetzt über das Internet mit dem Gegenüber ausgetauscht werden. Anschließend kann der gemeinsame Schlüssel berechnet werden, indem der öffentliche Schlüssel des jeweils Anderen mit dem eigenen privaten Schlüssel multipliziert wird. So gilt für den gemeinsamen Schlüssel S:
$S = n_1 O_2 = n_2 O_1$

Beide Seiten kommen so an denselben Schlüssel, ohne diesen direkt über das Internet ausgetauscht zu haben und ohne dass er aus den ausgetauschten Daten eindeutig und effizient berechnet

werden kann. Es ist beinahe unmöglich, aus der Gleichung der Kurve, dem Startpunkt, dem Endpunkt, also dem öffentlichen Schlüssel, und der Charakteristik des Restklassenkörpers den privaten Schlüssel und damit den gemeinsamen Schlüssel zu berechnen. Die einzige Methode, aus allen Daten auf den privaten Schlüssel zu schließen, wäre ein sogenannter BruteForce-Angriff, bei dem alle Möglichkeiten ausprobiert werden müssen, was in der Praxis jedoch nicht umsetzbar ist (darauf wird im Abschnitt "Effizienz" genauer eingegangen). Bei sehr großen Werten für die privaten Schlüssel und besonderen elliptischen Kurven ist es für Angreifer praktisch unmöglich, aus allen öffentlichen und ausgetauschten Informationen den gemeinsamen Schlüssel S zu berechnen. Mithilfe dieses Schlüssels können Nachrichten verschlüsselt, entschlüsselt und sicher über das Internet versandt werden. So können sensible Informationen wie Kontodaten oder private Nachrichten sicher über unsichere Wege wie das Internet ausgetauscht werden, ohne dass dabei die Gefahr besteht, dass die reinen Daten einem Angreifer in die Hände fallen.

3.6 Cipher

Zwar gibt es unterschiedliche Methoden, einen Text mithilfe eines Schlüssels zu verschlüsseln, die hier beschriebene erfreut sich aber aufgrund ihrer Einfachheit und trotzdem hohen Sicherheit sehr großer Beliebtheit. Ein Cipher wird in der Regel dafür verwendet, eine in Klartext vorliegende Nachricht mithilfe eines Schlüssels in den sogenannten Ciphertext, die verschlüsselte Nachricht, umzuwandeln und später auch wieder zu entschlüsseln, also den Ciphertext wieder in den Klartext umzuwandeln. Dazu wird jedes Zeichen der Nachricht und des Schlüssels einer Zahl zugeordnet und diese anschließend einzeln miteinander addiert. Um bestimmten Zeichen bestimmte Zahlen zuzuordnen, wird meist die UTF-8-Repräsentation der Zeichen verwendet. UTF-8 ist eine Zuordnungstabelle, die jedem Zeichen eine Zahl zwischen 0 und 255 zuordnet, um Computern das Speichern, Verarbeiten und Darstellen von Text zu ermöglichen. Mit acht Bit, also einem Byte, können 256 Zahlen (0 - 255) dargestellt werden, was auch der Grund dafür ist, dass ein Textzeichen im Normalfall ohne Formatierung ein Byte an Speicherplatz belegt. Die Zeichen der Nachricht N "Das ist eine Nachricht" und des Schlüssels S "Seminararbeit-2020" werden mithilfe dieser Tabelle durch die folgenden Dezimalzahlen repräsentiert:

$N = \{68; 97; 115; 32; 105; 115; 116; 32; 101; 105; 110; 101; 32; 78; 97; 99; 104; 114; 105; 99; 104; 116\}$
$S = \{83; 101; 109; 105; 110; 97; 114; 97; 114; 98; 101; 105; 116; 45; 50; 48; 50; 48\}$

Für den Ciphertext C an der Stelle i gilt:
$C_i = (N_i + S_{i \ mod \ |S|}) \bmod m$

So ergibt sich für C mit $m = 256$:
$C = \{151, 198, 224, 137, 215, 212, 230, 129, 215, 203, 211, 206, 148, 123, 147, 147, 154, 162, 188, 200, 213, 221\}$

Würde man den Ciphertext jetzt wieder als reinen Text darstellen, würde er wie folgt aussehen: Æà×Ôæ×ËÓÍ{¢¼ÈÕÝ. Um aus dem Ciphertext und dem Schlüssel wieder die originale Nachricht zu errechnen, muss der Schlüssel vom Ciphertext, bzw. seiner numerischen Darstellung abgezogen werden. So gilt für die Nachricht N an der Stelle i:
$N_i = (C_i - S_{i \ mod \ |S|}) \ mod \ m$

Bei der Umwandlung in Ciphertext werden alle Informationen des ursprünglichen Klartextes zwischen Schlüssel und Ciphertext aufgeteilt, sodass mit jeweils nur einer Information von entweder dem Ciphertext oder dem Schlüssel der Klartext nicht bestimmt werden kann, da die Informationen dazu schlicht nicht vorhanden sind. Veranschaulicht werden kann dies an folgendem Beispiel: Die Nachricht "Hallo Welt" soll mit dem Schlüssel $\{50, 25, 14, 14, 11, 90, 35, 21, 14, 6\}$ verschlüsselt werden. Der Ciphertext sieht dann wie folgt aus: "zzzzzzzzzz". Daran sieht man eindeutig, dass der Ciphertext allein nicht genügend Informationen beinhaltet, um daraus die originale Nachricht zu extrahieren. Auch hat der Schlüssel allein nicht alle benötigten Informationen, sodass die ursprüngliche Nachricht wirklich nur in Kombination aus Schlüssel und Ciphertext bestimmt werden kann.

Soll als Schlüssel kein Text sondern ein Punkt $P(x_P|y_P)$ auf beispielsweise einer elliptischen Kurve verwendet werden, so kann auch damit die Nachricht verschlüsselt und entschlüsselt werden, wobei x_P^{-1} das multiplikative modulare Inverse von x_P ist.
$C_i = (x_P * N_i + y_P)\ mod\ m$
$N_i = (x_P^{-1} * (C_i - y_P))\ mod\ m$

Bei dieser Methode ergibt sich jedoch das Problem, dass gleiche Zeichen immer gleich verschlüsselt werden, weshalb die Sicherheit hier nicht besonders hoch ist. Um die Sicherheit zu erhöhen, kann entweder jedes Zeichen zusätzlich noch in Abhängigkeit der vorherigen Zeichen berechnet werden oder die spezielle Kombination der Koordinaten des Schlüsselpunktes wird einfach mithilfe einer Hashfunktion einem Text zugeordnet und der Klartext dann mithilfe der oben beschriebenen Methode verschlüsselt.

Es ist praktisch unmöglich, ohne den richtigen Schlüssel aus dem Ciphertext die originale Nachricht zu berechnen, da die Nachricht auch jede andere mit der gleichen oder einer geringeren Länge, wie zum Beispiel "Es ist gerade 23 Uhr" oder "Es gibt keinen Gott" sein kann. Aber auch "0pELHek1mLz=12-AsM1" ist eine der $256^{22} \approx 9,6 * 10^{52}$ Möglichkeiten. Zum Größenvergleich: Um alle diese Möglichkeiten zu berechnen, würde selbst ein Matrioshka Brain, ein hypothetischer Computer, der sämtliche von einem Stern abgestrahlte Energie effizient für seine Berechnungen nutzt, approximativ drei Jahre brauchen. Bei einer Nachricht mit nur drei Zeichen mehr wären es schon über 50 Millionen Jahre, um alle möglichen Kombinationen zu berechnen.

Um die Sicherheit beim Verschlüsseln weiter zu erhöhen, kann die Länge des Ciphertexts verändert werden, sodass es schwieriger wird, Rückschlüsse auf die Länge der originalen Nachricht zu ziehen. Zumindest bei kurzen Nachrichten spielt deren Länge eine signifikante Rolle bei der Zeit, die es dauert, alle Möglichkeiten auszuprobieren, weshalb eine Längenmodifikation hier durchaus sinnvoll sein kann. Bei langen Nachrichten ist es nicht mehr besonders wichtig, die Länge zu verändern, da es praktisch egal ist, ob ein Angreifer 500 Milliarden Jahre oder 10 Trillionen Jahre brauchen würde, um die Verschlüsselung durch Ausprobieren aller Möglichkeiten zu umgehen. Wenn Quantencomputer jedoch weiterhin so massive Fortschritte erleben, spielt das womöglich in Zukunft doch eine Rolle, da die Stärke von Quantencomputern eben genau darin liegt, sehr viele Rechenoperationen gleichzeitig durchführen zu können, weshalb ein zukünftiger Quantencomputer wahrscheinlich alle möglichen Kombinationen der Nachricht auf einmal ausprobieren kann.

4 Effizienz und Sicherheit

4.1 Effizienz

Der einfachste Algorithmus zum Berechnen des Ergebnisses von nP ist das Addieren von P mit sich selbst, und zwar n-mal. Soll beispielsweise das Ergebnis von $97P$ berechnet werden, so wird zuerst $P + P = 2P$ berechnet, dann $2P + P = 3P$, danach $3P + P = 4P$ und so weiter, bis das Ergebnis von $97P$ erreicht ist.

Jetzt könnte jedoch die Annahme getroffen werden, dass ein möglicher Angreifer zu allen möglichen n von nP das Ergebnis berechnen kann, bis das Ergebnis mit dem öffentlichen Schlüssel übereinstimmt. Ist das der Fall, so hat der Angreifer den privaten Schlüssel n und kann die Nachrichten entschlüsseln. Um diese Art des Angriffs zu verhindern, muss der private Schlüssel n so immens hoch gewählt werden, dass die Zeit, die der Angreifer brauchen würde, um alle Möglichkeiten auszuprobieren, selbst mit extremer Rechenleistung beinah unendlich hoch wäre. Damit jedoch Smartphones oder sonstige Endgeräte für die Berechnung des öffentlichen Schlüssels nicht auch mehrere Milliarden Jahre sondern nur einige Millisekunden benötigen, gibt es effiziente Methoden, das Ergebnis von nP innerhalb kürzester Zeit zu berechnen. Eine ebensolche Methode ist die sogenannte "double-and-add" Methode. Sie funktioniert für Angreifer nicht, da mit dieser Methode direkt auf das angepeilte Ziel hingearbeitet werden kann, wohingegen der Angreifer dieses Ziel eben nicht kennt und daher alle Möglichkeiten auf dem Weg dahin ausprobieren muss.

Dieses Verfahren basiert auf der Tatsache, dass jede ganze Zahl als eine Summe von Zweierpotenzen dargestellt werden kann. Die Zahl 97 beispielsweise lässt sich durch $2^6 + 2^5 + 2^0$, bzw. durch $64 + 32 + 1$ darstellen. Im Fall von elliptischen Kurven heißt das, es werden alle Verdoppelungen des Startpunktes kleiner oder gleich dem privaten Schlüssel berechnet und die jeweils richtigen addiert, bis das Ergebnis erreicht ist. Um beim Beispiel von $97P$ zu bleiben, wäre das $64P + 32P + 1P$. So sind nicht wie beim ursprünglichen Algorithmus 96 Rechenoperationen nötig, sondern nur 8. Diese 8 Operationen bestehen aus dem 6-maligen Verdoppeln des Startpunktes bis einschließlich $64P$ und die darauffolgenden 2 Additionen $64P + 32P = 96P$ und $96P + 1P = 97P$. Bei $n = 970.624.379.105.268.705$ werden nur 94 Rechenoperationen benötigt, für deren Berechnung ein modernes Smartphone weniger als zehn Millisekunden braucht. Für einen Angriff mit der obigen, ineffizienten Methode müsste das Smartphone im für den Angreifer schlechtesten Fall jedoch ganze 31 Jahre lang rechnen, um alle Möglichkeiten auszuprobieren und den gemeinsamen Schlüssel zu berechnen. In der Praxis ist n so hoch gewählt, dass selbst der stärkste Supercomputer der Welt um ein vielfaches länger brauchen würde, als das Universum alt ist, um den privaten Schlüssel durch das Ausprobieren aller Möglichkeiten zu bestimmen.

4.2 Sicherheit

Die Verschlüsselung mit elliptischen Kurven lässt eine größere Schlüssellänge bei selbiger Effizienz wie beim herkömmlichen RSA-Verfahren zu, wodurch die Sicherheit bei gleicher Länge des Schlüssels drastisch erhöht werden kann. Laut NIST (National Institute of Standards and Technology) wird für ein gewisses Level an Sicherheit beim RSA-Verfahren ein Schlüssel mit 2048 Bits benötigt, bei der Verschlüsselung mit elliptischen Kurven nur einer mit 224 Bits, das

sind knapp 11% der Schlüssellänge von RSA. Für ein höheres Sicherheitsniveau ist die Länge des Schlüssels der Kryptographie mit elliptischen Kurven nur noch 3,3% von der des RSA Schlüssels. (Q6, "1.1 Motivation")

Wie bereits im Abschnitt für pseudozufällige Zahlen erwähnt, werden in die meisten CPUs von ihren Herstellern schon Hardwarezufallsgeneratoren eingebaut. Die zufällig generierten Zahlen dieser Generatoren werden in ihrer Reinform nur selten für kryptographische Anwendungen verwendet, oft bilden sie jedoch mit zusätzlicher Randomisierung die Grundlage für sichere und unvorhersagbar zufällige Zahlen. In einigen integrierten Zufallsgeneratoren werden Hintertüren vermutet, mit denen Geheimdienste wie die NSA Verschlüsselungen, die auf diesen Zufallsgeneratoren basieren, umgehen können, um so an bestimmte Information bestimmter Personen zu gelangen.

Um die Sicherheit in sensiblen Bereichen wie der Internetkommunikation weiter zu erhöhen wird der Schlüssel außerdem häufig erneuert, sodass ein Angreifer, der wie auch immer an den Schlüssel gekommen ist, nur einen kleinen Block an Informationen entschlüsseln kann und nicht die ganze Datenbank beziehungsweise die gesamte Konversation. Beim Verschlüsseln selbst wird der Schlüssel vor dem Ver- und Entschlüsseln häufig mit einer Hashfunktion gehashed, sodass die Nachricht auch mit einem fast komplett richtigen Schlüssel nicht entschlüsselt werden kann, da eine gute Hashfunktion für zwei sich auch nur minimal unterscheidende Argumente komplett unterschiedliche Hashs produziert.

Das Problem, auf elliptischen Kurven n aus nP und P zu berechnen, heißt diskretes Logarithmusproblem in elliptischen Kurven, kurz **ECDLP** für **E**lliptic **C**urve **D**iscrete **L**ogarithm **P**roblem. Es ist bis jetzt kein mathematisches Verfahren bekannt, dass das ECDLP effizient lösen kann. Sollte irgendwann eine Lösung gefunden werden, wären sämtliche Verschlüsselungsmethoden basierend auf elliptischen Kurven mit einem Schlag angreifbar und unsicher. Selbiges kann aber auch bei den anderen asymmetrischen Verfahren passieren, das ist also keine Schwäche speziell bei elliptischen Kurven. Sobald jedoch kommerzielle Quantencomputer verfügbar sind, werden sich viele der aktuellen Verschlüsselungsmethoden sowieso als unsicher herausstellen, da das Diskreter-Logarithmus-Problem von diesen vermutlich effizient gelöst werden kann. Bei der Verschlüsselung mit elliptischen Kurven muss sich erst noch herausstellen, wie sie sich in der Ära der Quantencomputer im Vergleich zu ihrem Konkurrenten RSA schlagen wird.

Eine Unsicherheit der Kryptographie mit elliptischen Kurven ist der sogenannte Man-In-The-Middle-Angriff (kurz MITMA), bei dem ein Computer sämtlichen Internetverkehr zwischen seinen "Opfern" abfängt und so tun kann, als wäre er der andere Chatpartner, um beim Beispiel mit Nachrichtenapps zu bleiben. So tauscht er mit den beiden Gesprächspartnern jeweils Schlüssel aus und vermittelt auch zwischen ihnen, um nicht aufzufallen. So kann er trotz allen Sicherheitsvorkehrungen alle verschlüsselten Nachrichten mitlesen, ohne dass die beiden Chatpartner etwas davon mitbekommen würden. Um diese Art des Angriffs zu verhindern, gibt es die sogenannte Zwei-Faktor-Authentifizierung, bei der eine Anmeldung über einen zweiten Weg, wie eine SMS oder E-Mail bestätigt werden muss. Über diesem Wege erhält das Gerät eine zusätzliche Information, die zum Verschlüsseln und Entschlüsseln der Nachrichten benötigt wird, die jedoch ein Angreifer "In The Middle" nicht besitzt. Geheime Chats in Telegram bieten zusätzlich die

Möglichkeit, ihre Verbindung auf eine "MITMA" hin überprüfen zu lassen, indem jeder Nutzer seinen gemeinsamen Schlüssel in dem jeweiligen Chat einsehen kann. Ist der Schlüssel beim Gegenüber derselbe, so kann kein "Man In The Middle" sein und den Chat mitlesen.

4.3 Digitale Signaturen

Elliptische Kurven können in der Sicherheitsforschung nicht nur dazu verwendet werden, den Schlüssel für ein asymmetrisches Verschlüsselungsverfahren auszutauschen, sondern auch, um digitale Signaturen anzufertigen. Digitale Signaturen werden dazu benutzt, den Absender einer Datei zu identifizieren, beziehungsweise deren Ursprung zu bestätigen. Wird beispielsweise eine JPG-Datei über das Internet an einen Arbeitskollegen gesendet, so kann die Datei auf dem Weg zum Empfänger unbemerkt manipuliert worden sein. Um die Echtheit digitaler Dateien zu beweisen, wird die Datei vor dem Verschicken vom Sender digital signiert, wodurch der Empfänger feststellen kann, ob die Datei unterwegs modifiziert wurde oder nicht. Dazu wird die Datei vor dem Senden mit einer Hashfunktion gehashed. Der MD5-Hash des Bildes aus dem Abschnitt "Endliche Körper" sieht beispielsweise so aus: d321349d7c2eaf752db6987efeca600b. Der Sender kann diesen Hash mit seinem privaten Schlüssel verschlüsseln und die Datei, seinen öffentlichen Schlüssel und den verschlüsselten Hash, welcher die digitale Signatur darstellt, an den Empfänger schicken. Nun kann der Empfänger die empfangene Signatur mit dem empfangenen öffentlichen Schlüssel entschlüsseln und mit dem Hashwert der empfangenen Datei vergleichen. Stimmen beide überein, so kann der Empfänger sicher sein, dass die Datei nur von der Person stammen kann, die den privaten Schlüssel besitzt und somit unterwegs nicht modifiziert wurde. Algorithmen, die digitale Signaturen mithilfe von elliptischen Kurven anfertigen, werden kurz **ECDSA** für **E**lliptic **C**urve **D**igital **S**ignature **A**lgorithm genannt.

4.4 Anwendungen

Obwohl die Kryptographie mit elliptischen Kurven im Vergleich zu RSA erst vor relativ kurzer Zeit populär wurde, wird sie trotzdem schon in sehr vielen Systemen wie der Verschlüsselung von Internetseiten oder der digitalen Signatur von Bitcoin genutzt. Die Mehrheit der Verschlüsselungen basiert jedoch noch immer auf RSA, da dieses Verfahren schon deutlich länger bekannt ist und sich somit in vielen Bereichen als der Standard durchgesetzt hat. Abgesehen von der etwas einfacheren Implementierbarkeit des RSA-Verfahrens gibt es keinen rationalen Grund für die breite Verwendung, vor allem im Vergleich zur elliptische-Kurven-Verschlüsselung. Aufgrund der deutlich höheren Sicherheit und Effizienz der ECC wird sich diese Art der Kryptographie langfristig jedoch durchsetzen und RSA langsam aber sicher ersetzen. Beispiele für Bereiche, in denen elliptische Kurven in der Kryptographie schon jetzt eingesetzt werden, sind unter anderem Smartcards wie Kreditkarten oder Girocards, da hier die Effizienz und die geringe Schlüssellänge von ECC aufgrund der sehr geringen physischen Größe eine sehr große Rolle spielen. Auch basiert ein Großteil der Verschlüsselung von Webseiten mittlerweile auf elliptischen Kurven und nicht mehr auf RSA, was noch vor einigen Jahren der Standard war. Immer, wenn in einem Browser wie Safari oder Chrome oben ein Schloss angezeigt wird, beziehungsweise die URL der Website mit https:// (ausschlaggebend ist das "s" am Ende) beginnt, weist das auf eine verschlüsselte Webseite hin. Außerdem ist das gesamte TOR Netzwerk, das seinen Nutzern einen absolut anonymen Internetverkehr ermöglicht, komplett mit elliptischen Kurven verschlüsselt.

Aber auch digitale Signaturen mit elliptischen Kurven werden häufig benutzt, unter anderem in Apples iMessage, WhatsApp und Telegram. Aber auch der Besitznachweis von Bitcoin basiert auf Signaturen mit elliptischen Kurven.

5 Schluss

Weite Teile unserer Gesellschaft würden ohne die Möglichkeit zur sicheren Verschlüsselung von Informationen nicht so funktionieren, wie sie es heute tun. Das gesamte Internet würde und könnte in der Form nicht existieren, die wir kennen. Die technologische und digitale Infrastruktur wäre noch lange nicht so weit entwickelt und auch lange nicht so nützlich wie sie es heute ist. Der technologische Fortschritt der Menschheit hätte sich vermutlich nie so entwickelt, dass die Welt entstanden wäre, die es heute gibt. Ohne die Möglichkeit, Informationen sicher verschlüsseln zu können, gäbe es keine Instant-Messengerdienste wie Telegram, Signal oder iMessage, es könnten keine kritischen Informationen öffentlich auf einen Server in das Internet gestellt und es könnte erst recht keinen Informationen aus dem Internet getraut werden, da sie ohne Probleme hätten manipuliert werden können. Es gäbe keine Wikipedia, kein Google, keine Onlineshops, keine Onlinezeitungen und keine sozialen Netzwerke. Menschen würden die Informationen, die nicht persönlich ausgetauscht werden können, per Telefon oder Fax austauschen. Sensible Informationen könnten nur persönlich ausgetauscht werden. Es könnten keine Raketen oder Satelliten in den Orbit geschossen werden, da sie nicht vor einer Fremdübernahme geschützt wären. Der Stand der zivil nutzbaren und lebenserweiternden Technologien wäre auf dem gleichen Level wie in der Mitte des 20. Jahrhunderts.

Zu unserem großen Glück gibt es jedoch sichere Verschlüsselungsmethoden, die sich zudem ständig weiterentwickeln, um den immer neuen Herausforderungen unserer Welt gewachsen zu bleiben. In Zukunft wird aufgrund der zunehmenden Globalisierung, Digitalisierung und auch der Kolonialisierung des Weltalls noch viel mehr von der Sicherheit kryptographischer Verschlüsselungsmethoden abhängen als heute. Jegliche zwischenmenschliche Kommunikation wird über ein globales oder sogar interplanetares Netzwerk ablaufen, der gesamte Informationsfluss und das gesamte Wissen der Menschheit wird in Zukunft einzig und allein durch sichere Arten der Verschlüsselung gespeichert, übertragen und geschützt werden können. Extrem sichere Methoden der Verschlüsselung bilden das Fundament für eine vernetzte und digitale Zukunft. Die Kryptographie mit elliptischen Kurven ist ein Teil davon und wird uns dabei unterstützen, in eine schönere, bessere und sicherere Zukunft vorzudringen.

6 Sonstiges

6.1 Literaturverzeichnis

Q1: https://de.m.wikipedia.org/wiki/Elliptische_Kurve [Stand: 19.11.2020]
Q2: Meshram, Suchitra: "Overview of History of Elliptic Curves and its use in cryptography", April 2014, https://www.ijser.org/researchpaper/Overview-of-History-of-Elliptic-Curves-and-its-use-in-cryptography.pdf [Stand: 04.01.2021]
Q3: Magidin, Arturo: "How to find the inverse modulo m?", März 2011, https://math.stackexchange.com/questions/25390/how-to-find-the-inverse-modulo-m [Stand: 07.01.2021]
Q4: Dubuque, Bill: "Solving linear congruences by hand: modular fractions and inverses", Juli 2012, https://math.stackexchange.com/questions/174676/solving-linear-congruences-by-hand-modular-fractions-and-inverses/174687#174687 [Stand: 07.01.2021]
Q5: https://de.wikipedia.org/wiki/Martin_Hellman [Stand: 17.11.2020]
Q6: Kultinov, Kirill: "Software Implementations and Applications of Elliptic Curve Cryptography", 2019, https://corescholar.libraries.wright.edu/cgi/viewcontent.cgi?article=3288&context=etd_all [Stand: 08.01.2021]
Q7: Hoffman Chris, "How Computers Generate Random Numbers", November 2019, https://www.howtogeek.com/183051/htg-explains-how-computers-generate-random-numbers/ [Stand: 12.12.2020]

Abbildungen erstellt mithilfe von:
- https://www.geogebra.org/calculator
- https://graui.de/code/elliptic2/
- Gravit Designer

6.2 Erklärung

Ich erkläre, dass ich diese Seminararbeit selbstständig verfasst und keine anderen als die angegebenen Hilfsmittel benutzt habe. Insbesondere versichere ich, dass ich alle wörtlichen und singemäßen Übernahmen aus anderen Werken als solche kenntlich gemacht habe.

Merlin Hof — Rosenheim, 19.01.21

Unterschrift — Datum