Kryptographie II - Zusammenfassung

Klassische Chiffrierverfahren

MASC-Verschlüsselung

→ Monoalphabetische Substitutionschiffrierung

1. Verschlüsselung: Oben Alphabet, unten am Anfang Schlüsselwort eintragen. Doppelte Zeichen werden eliminiert, am Ende wird ausgehen vom letzten Buchstaben das Alphabet zirkulär fortgesetzt.
2. Entschlüsselung: Bijektive Umkehrfunktion.
3. Angriff: Häufigkeitsanalyse bzw. Textteile erraten und abhängig vom zirkulären Alphabet auffüllen.

TRANSSPA-Verschlüsselung

1. Schlüssel: Wort mit n Buchstaben. Tabelle anlegen, wobei pro Buchstaben eine Spalte existiert
2. Verschlüsselung: Text zeilenweise in die Tabelle eintragen und anschließend spaltenweise nach Reihenfolge der Buchstaben des Schlüssels ausgelesen. Bei gleichen Buchstaben geschieht dies von links nach rechts.
3. Entschlüsselung: Chiffretext spaltenweise in eine Tabelle mit n Spalten eintragen und mit richtiger Permutation zeilenweise auslesen.
4. Angriff: Aufüllzeichen betrachten oder Schlüssellänge raten

STROM-Chiffrierung

1. Es werden nur Großbuchstaben berücksichtigt!
2. Der Schlüssel besteht aus einer Folge von Großbuchstaben $k_1, k_2 \ldots$, die mindestens so lang ist wie der verschlüsselte Text.
3. Verschlüsselungsfunktion: $f(a_i, k_i) = a_i + k_i \mod 26$
4. Entschlüsselungsfunktion: $g(b_i, k_i) = b_i - k_i \mod 26$
5. AUTOKEY: Der key ist ein Schlüsselwort kleiner als der Text, an den der Klartext angehängt wird.

VIGENÈRE-Verschlüsselung

1. Schlüsselwort $k_1, k_2 \ldots, k_n$, welches periodisch auf den Text angewandt wird.
2. Verschlüsselungsfunktion: $b_i = a_i + k_i \ mod \ 26$.
3. Entschlüsselungsfunktion: $a_i = b_i - k_i \ mod \ 26$.
4. Angriff: Kasiski-Test, bei dem nach mehrfach auftretenden Zeichenketten gesucht wird. Aus der Differenz der Indizes kann die Länge des Schlüsselwortes bestimmt werden. ⇒ Tabelle anlegen, jede Spalte Caesar-Chiffriert, Häufigkeitsanalyse!
5. Periodisches Strom

PLAYFAIR-Verschlüsselung

1. Alphabet mit 25 Großbuchstaben, wobei J = I.
2. Eine Code-Wort wird in eine 5×5-Matrix zeilenweise eingetragen. Doppelte Buchstaben werden gedarlekt! Die restliche Matrix wird alphabetisch mit den noch fehlenden Buchstaben ergänzt.
3. Der Ausgangstext wird in Bigramme unterteilt, wobei der Buchstabe X eingefügt wird, falls ein Bigramm aus 2 gleichen Buchstaben besteht.
4. Verschlüsselung: Bigramm in Matrix suchen und nach folgenden 3 Regeln umschreiben:
   (a) Gleiche Zeile: Eins nach rechts (zirkulär).
   (b) Gleiche Spalte: Eins nach unten (zirkulär).
   (c) Sonst: Quadrat bilden und Ecktausch zeilenweise nach Bigrammreihenfolge.
5. Für die Entschlüsselung Regeln invers anwenden.

ADFGVX-Chiffrierung

1. Das Klartextalphabet besteht aus 36 Zeichen, den Buchstaben $A, \ldots, Z$ und den Ziffern $0, 1, \ldots, 9$. Das Chiffretextalphabet aus den Zeichen A, D, F, G, V, X.
2. Schlüssel: Erstellen einer 6×6 und Wahl eines Permutationsschlüssels der Länge n.
3. Verschlüsselung: Umwandlung des Klartextes in die Zeichenpaare des Chiffrealphabets. Anschließend wird der Chiffretext zeilenweise in n Spalten aufgeteilt und permutiert. Am Schluss wird der Chiffretext spaltenweise ausgelesen.
4. Entschlüsselung: Mit erstem Schlüssel Matrix erstellen. Mit dem zweiten Schlüssel Buchstabenreihenfolge bestimmen. Durch Anzahl Zeichen mod Schlüssellänge Füllbits berechnen, mit Teilen und Aufrunden Zeilenanzahl berechnen. Text Nach Reihenfolge der Spalten eintragen und zeilenweise mit Matrix entschlüsseln.

Drehraster-Chiffrierung für 6×6-Schablonen

1. key: Schablone mit 9 Löchern abhängig von Bahnmatrix
2. Verschlüsselung: Aufteilung des Ausgangstextes in Blöcke der Länge 36. Eintragen des Textes in Schablone mit 90° Drehungen im Uhrzeigersinn. Für jeden Block wiederholen.
3. Entschlüsselung: Blöcke der Länge 36 in Matrizen eintragen und mit Schablone entschlüsseln.
4. Angriff: Erste 36 Ziffern in Matrix. Buchstaben in Matrix vermuten und mit Bahnmatrix 3 Felder ausschließen. Verfahren mit jedem „bekannten“ Buchstaben wiederholen.

Zahlentheoretische Grundlagen

Der erweiterte euklidische Algorithmus
→ Berechnet ggT(a,b)!

Satz 0.1

Zu $a, b \in \mathbb{Z}$ gibt es $x, y \in \mathbb{Z}$ mit $ggT(a,b) = xa + yb$.

→ Initialisierung:
⇒ $a_0 = a,\ b_0 = b,\ x_0 = 1,\ x'_0 = 0,\ y_0 = 0,\ y'_0 = 1$
→ Inverse von a in x
→ Inverse von b in y
→ ggT(a,b) im Kästchen über 0

Beispiel:

Zeile	a	q	x	y	Berechnung
0	1077	-	1	0	Vorinitialisiert
1	1029	-	0	1	Vorinitialisiert
2	48	1	1	-1	x2 = x0 - (q2*x1)
3	21	21	-21	22	x3 = x1 - (q3*x2)
4	6	2	43	-45	x4 = x2 - (q4*x3)
5	3	3	-150	**157**	x5 = x3 - (q5*x4)
6	0	2	-	-	Berechnung von y äquivalent.
=> 1029 * 157 <=> 1 mod 1077 und -150*1077 + 157*1029 = 3					

Invertierbarkeit modulo n
Definition 0.2

Für $n \in \mathbb{N}$ und $a \in \mathbb{Z}$ ist a invertierbar modulo n, falls ein $b \in \mathbb{Z}$ existiert mit $ab \equiv 1\ mod\ n$.

⇒ a ist invertierbar modulo n, wenn $ggT(n,a) = 1$ ist.
⇒ Ist a invertierbar, so findet man mit dem EEA die Inverse y.

Satz 0.3 (*Kleiner Satz von Fermat*)

Für eine Primzahl p und eine ganze Zahl a gelten die Aussagen $a^p \equiv a\ mod\ p$ und $ggT(a,p) = 1 \Rightarrow a^{p-1} \equiv 1\ mod\ p$.

⇒ Für die ersten 200 Zahlen gilt auch die Umkehrung des Satzes!

Satz 0.4 (*Satz von Euler*)

Ist $n \in \mathbb{N}$ und $a \in \mathbb{Z}$ mit $ggT(a,n) = 1$, so gilt

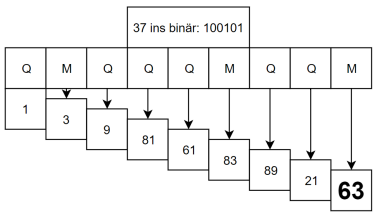
$$a^{\varphi(n)} \equiv 1\ mod\ n.$$

$\varphi(n)$: Gibt für eine Zahl n die Mächtigkeit an, wie viele Zahlen $a \in [0, \ldots, n-1]$ kein Teiler von n sind.
→ Bsp: $\varphi(10) = \#\{1,3,7,9\} = 4$

Die square-and-multiply-Methode

1. Gegeben: $a^b \mod n$
2. Definition: $Q \equiv x^2, M \equiv x \cdot a$.
3. $b_{(10)} \rightarrow b_{(2)}$ (Binärumwandlung des Exponenten).
4. $0 \in b \rightarrow$ Q bzw. $1 \in b \rightarrow$ QM.
5. Sequentielle Berechnung mit modulation nach jedem Schritt. Nicht vergessen, dass x mit 1 initialisiert wird!

Beispiel $3^{37} \mod 100$:



Die Ordnung
→ Ist $n \in \mathbb{N}$, $a \in \mathbb{Z}$ mit $ggT(n,a) = 1$, so gilt $a^{\varphi(n)} \equiv 1 \mod n$.
→ Die Ordnung $ord_n(a)$ von a modulo n ist der erste Exponent k, für den gilt:

$$a^k \equiv 1 \mod n$$

Eigenschaften der Ordnung
Sei $n \in \mathbb{N}$ und $a \in \mathbb{Z}$ mit $ggT(n,a) = 1$. Für $k, k_1, k_2 \in \mathbb{N}$ gilt dann:

1. $a^k \equiv 1\ mod\ n \Leftrightarrow ord_n(a) | k$.
2. $ord_n(a) | \varphi(n)$.
3. $a^{k_1} \equiv a^{k_2}\ mod\ n \Leftrightarrow k_1 \equiv k_2\ mod\ ord_n(a)$.
4. Für $k \in \mathbb{N}$ gilt $ord_n(a^k) = \frac{ord_n(a)}{ggT(ord_n(a),k)}$.

Bestimmung von $ord_n(a)$
Naive Methode:

1. Sei $n \in \mathbb{N},\ a \in \mathbb{Z}$ mit $ggT(n,a) = 1$.
2. Für $k = 1, \ldots, n$: Gebe k als $ord_n(a)$ zurück, falls $a^k \equiv 1\ mod\ n$.

Der chinesische Restsatz
Seien $m_r \in \mathbb{N}$ mit $ggT(m_i, m_j) = 1\ \forall\ i \neq j$ und $a_r \in \mathbb{Z}$ gegeben. Dann ist das Kongruenzgleichungssystem mit genau einer Lösung für $x \mod \prod_i m_i$ lösbar:

$$x \equiv a_i\ mod\ m_i$$

Lösung:

1. Tipp: Modulatoren verkleinern mit Primfaktorzerlegung.
2. M_i berechnen: Produkt aller m_l außer m_i.
3. EEA von M_i und m_i: Lösung z_i (Inverse von M_i).
4. Einsetzen: $x = a_1 \cdot z_1 \cdot M_1 + \cdots + a_r \cdot z_r \cdot M_r$
5. Ergebnis verkleinern:$x \mod \prod_i m_i$

Quadrate & Quadratwurzeln modulo n

Das Legendre-Symbol:

→ Gibt Lösbarkeit von $x^2 \equiv a \mod p$ mit p ungerade an.

→ $\left(\frac{a}{p}\right) = \begin{cases} +1 & \text{falls } ggT(a,p)=1, \exists\, x \in \mathbb{Z} : x^2 \equiv a \mod p \\ -1 & \text{falls } ggT(a,p)=1, \nexists\, x \in \mathbb{Z} : x^2 \equiv a \mod p \\ 0 & \text{falls } a \equiv 0 \mod p \end{cases}$

→ $\left(\frac{a}{p}\right) = 1$: a ist quad. Rest mod p (-1 quad. Nichtrest)

Eigenschaften:

→ Gilt $a \equiv b \mod p$ so folgt $\left(\frac{a}{p}\right) = \left(\frac{b}{p}\right)$

→ Besitzt den surjektiven Gruppenhomomorphismus:

$$\left(\frac{ab}{p}\right) \equiv \left(\frac{a}{p}\right)\left(\frac{b}{p}\right) \mod p$$

→ Jedes $g \in \mathbb{F}_p$ mit $ord_p(g) = p-1$ nennt man Primitivwurzel.

→ Ist $p > 2$ prim , g Primitivwurzel mod p , $m \in \mathbb{Z}$: dann gilt:

$$\left(\frac{g^m}{p}\right) = (-1)^m$$

⇒ Es gibt genau $\frac{p-1}{2}$ Quadrate und Nichtquadrate modulo p.

Satz von Euler:
Sei p eine ungerade Primzahl und $a \in \mathbb{Z}$. Dann gilt:

$$a^{\frac{p-1}{2}} \equiv \left(\frac{a}{p}\right) \; mod\; p.$$

⇒ Schnelle Berechenbarkeit über Square-and-Multiply.

→ Legendre-Symbol immer anhand von Euler berechnen

→ Gilt nicht für das Jacobi Symbol

Schnelle Berechenbarkeit von $a = -1$ und $a = 2$:

→ $\left(\frac{-1}{p}\right) = (-1)^{\frac{p-1}{2}} = \begin{cases} 1 & \text{falls } p \equiv 1 \; mod\; 4 \\ -1 & \text{falls } p \equiv -1 \; mod\; 4 \end{cases}$

→ $\left(\frac{2}{p}\right) = (-1)^{\frac{p^2-1}{8}} = \begin{cases} 1 & \text{falls } p \equiv 1,7 \; mod\; 8 \\ -1 & \text{falls } p \equiv 3,5 \; mod\; 8 \end{cases}$

Hilfreich für die Berechnung:

→ $(-1)^{\frac{ab-1}{2}} = (-1)^{\frac{a-1}{2}} \cdot (-1)^{\frac{b-1}{2}}$

→ $(-1)^{\frac{(ab)^2-1}{8}} = (-1)^{\frac{a^2-1}{8}} \cdot (-1)^{\frac{b^2-1}{8}}$

Satz 0.5 (*Gaußsches Reziprozitätsgesetz*)

Sind p und q verschiedene Zahlen, so gilt:

$$\left(\frac{p}{q}\right) = \begin{cases} \left(\frac{q}{p}\right) & \text{falls } p \equiv 1 \; mod\; 4 \text{ oder } q \equiv 1 \; mod\; 4 \\ -\left(\frac{q}{p}\right) & \text{falls } p \equiv q \equiv 3 \; mod\; 4 \end{cases}$$

→ Regel, welches Vorzeichen ich beim umdrehen des Legendre-Symbols brauche.

Das Jacobi-Symbol:

→ Verallgemeinerung des Legendre-Symbols.

⇒ Nicht zwangsweise Primzahlen im Nenner!

Definition 0.6 (*Jacobi-Symbol*)

Sei $a \in \mathbb{Z}$ und $b \in \mathbb{N}$ ungerade. Betrachte Primfaktorzerlegung $b = \prod p_i^{e_i}$. Dann ist das Jacobi-Symbol definiert als

$$\left(\frac{a}{b}\right) = \prod \left(\frac{a}{p_i}\right)^{e_i}$$

→ Achtung: Jacobi ist über Legendre definiert, nicht über die Lösbarkeit der Gleichung $x^2 \equiv a \; mod\; p$.

Eigenschaften:

→ Es gelten die gleichen Eigenschaften wie für das Legendre-Symbol.

→ Bemerke: $\left(\frac{a}{b}\right) = \left(\frac{a \bmod b}{b}\right)$

→ Sonderfall: $\left(\frac{0}{b}\right) = \begin{cases} 1 & \text{für } b = 1 \\ 0 & \text{für } b \geq 3 \end{cases}$

→ Anzahl der Schritte zur Berechnung des Jacobi-Symbols ist logarithmisch in b.

Algorithmus zur Berechnung des Jacobi-Symbols:

Geg: $a \in \mathbb{Z}, b \in \mathbb{N}$ ungerade.

Ges: $\left(\frac{a}{b}\right)$

1. Falls $a \geq b$, so berechne $\left(\frac{a \bmod b}{b}\right) = \left(\frac{a'}{b}\right)$.
2. Ist a' = 0, so tritt der Sonderfall ein und wir terminieren. Ansonsten zerlegen wir $a' = 2^e \tilde{a}$. Dann erhalten wir
$$\left(\frac{a'}{b}\right) = \left(\frac{2^e \cdot \tilde{a}}{b}\right) = \left(\frac{2}{b}\right)^e \cdot \left(\frac{\tilde{a}}{b}\right)$$
3. Nun lässt sich die Formel für a = 2 anwenden:
$$\left(\frac{a'}{b}\right) = (-1)^{\frac{b^2-1}{8} \cdot e} \cdot \left(\frac{\tilde{a}}{b}\right)$$
4. Da $\tilde{a}$ und b ungerade sind, lässt sich das quadratische Reziprozitätsgesetz anwenden:
$$\left(\frac{\tilde{a}}{b}\right) = (-1)^{\frac{\tilde{a}-1}{2}\frac{b-1}{2}} \left(\frac{b}{\tilde{a}}\right)$$
5. Gehe zu Schritt 1.

Beispiel: Berechne $\left(\frac{24}{35}\right)$

$$\left(\frac{24}{35}\right) = \left(\frac{2^3 \cdot 3}{35}\right) = \left(\frac{2}{35}\right)^3 \cdot \left(\frac{3}{35}\right) \overset{35=3 \; mod\; 8}{=}$$
$$= (-1)^3 \cdot \left(\frac{3}{35}\right) \overset{3=35=3 \; mod\; 4}{=} (-1) \cdot \left(-\left(\frac{35}{3}\right)\right) \overset{35 \; mod 3=2}{=}$$
$$= \left(\frac{2}{3}\right) = -1$$

Primzahltests

Der Fermatsche Primzahltest

Definition 0.7

Eine zusammengesetzte natürliche Zahl n heißt Fermat-Pseudoprimzahl zur Basis a, wenn $ggT(n,a) = 1$ gilt und $a^{n-1} \equiv 1 \ mod \ n$.

Test: Berechne mit der square-and-multiply-Methode $b = a^{n-1} \ mod \ n$. Ist $b = 1$, so ist n eine Primzahl oder eine Fermat-Pseudoprimzahl zur Basis a.

Der Solovay-Strassen-Primzahltest

→ Basiert auf dem Satz von Euler bzgl. Legendre und Jacobi.
→ Idee: Ist n eine ungerade natürliche Zahl, so können wir das Jacobi-Symbol $\left(\frac{a}{n}\right)$ mit $a^{\frac{n-1}{2}} \ mod \ n$ vergleichen.

Definition 0.8

Sei $n \geq 3$ eine ungerade natürliche Zahl und $a \in \mathbb{Z}$ mit $ggT(a,n) = 1$. Gilt

$$\left(\frac{a}{n}\right) \not\equiv a^{\frac{n-1}{2}} \ mod \ n$$

so ist n zusammengesetzt, andernfalls besteht n den Solovay-Strassen-Primzahltest zur Basis a.

→ Besteht n den Test, so nennen wir n wahrscheinlich prim oder nennen sie Eulersche Pseudoprimzahl zur Basis a.
→ Bsp. $121 = 11^2$ ist eine Eulersche Pseudoprimzahl zur Basis 3.

Der Miller-Rabin-Test

Definition 0.9

Eine zusammengesetzte ungerade natürliche Zahl n heißt Miller-Rabin-Pseudoprimzahl zur Basis a oder eine starke Pseudoprimzahl zur Basis a, wenn n den Miller-Rabin-Test zur Basis a besteht.

Algorithmus zum Miller-Rabin-Test:

Gegeben: n ungerade

1. $x = n - 1 \Rightarrow x = 2^l \cdot Rest$
2. if($a^{Rest} \mod n == 1$) true
3. else if $(\forall i \in [0, l-1] : a^{Rest \cdot 2^i} \mod n == -1)$true
4. else false

Zusammenhang zwischen den Primzahltests

→ Gemeinsamkeit aller Tests:
⇒ $n \in \mathbb{N}$ ungerade und es soll $ggT(a,n) = 1$ für $a \in \mathbb{Z}$ gelten.
→ Besteht n den Solovay-Strassen-Test zur Basis a, so auch den Fermat-Test zur Basis a.
⇒ Logisch, da $\pm 1 = a^{\frac{n-1}{2} \cdot 2} \equiv a^{n-1} \ mod \ n = 1$
⇒ Solovay-Strassen-Test also besser als der Fermat-Primzahltest
→ Besteht n den Miller-Rabin-Test zur Basis a, so auch den Solovay-Strassen-Test zur Basis a.
⇒ Miller-Rabin-Test ist besser als der Solovay-Strassen-Test.

Quadratwurzelziehen modulo p

→ Ziel: Lösen der Gleichung wenn bekannt ist, dass $\left(\frac{a}{p}\right) = 1$ gilt, also a ein Quadrat modulo p ist.
⇒ Finden wir eine Lösung x_1, so auch eine Lösung $x_2 \equiv -x_1 \equiv p - x_1 \ mod \ p$.

Das Tonelli-Verfahren

Lemma 0.10

Sei p eine ungerade Primzahl, $\left(\frac{a}{p}\right) = 1$, $x, y \in \mathbb{N}_0$ und $n \in \mathbb{Z}$ mit $\left(\frac{n}{p}\right) = -1$.

1. *Es gilt $a^{\frac{p-1}{2}} n^0 \equiv 1 \ mod \ p$ wegen dem Satz von Euler.*
2. *Mit Umformung des Legendre-Symbols gilt: Ist $a^x n^y \equiv 1 \ mod \ p$, so ist y gerade.*
3. *Ist $a^x n^y \equiv 1 \ mod \ p$ und x gerade, so gilt $a^{\frac{x}{2}} n^{\frac{y}{2}} \equiv 1 \ mod \ p$ oder $a^{\frac{x}{2}} n^{\frac{y}{2}} \left(\frac{n}{p}\right) \equiv a^{\frac{x}{2}} n^{\frac{y}{2} + \frac{p-1}{2}} \equiv 1 \ mod \ p$.*
4. *Ist $a^x n^y \equiv 1 \ mod \ p$ und x ungerade, so gilt $a \equiv (a^{\frac{x+1}{2}} n^{\frac{y}{2}})^2 \ mod \ p$.*

Algorithmus zum Tonelli-Verfahren:

Geg. p ungerade Primzahl und $\left(\frac{a}{p}\right) = 1$.

Ges. w mit $w^2 \equiv a \ mod \ p$.

1. Suche ein $n \in \mathbb{F}_p$ mit $\left(\frac{n}{p}\right) = -1$.
2. Initialisiere $x = \frac{p-1}{2}$ und $y = 0$.
3. Falls $x \ mod \ 2 = 1$, terminiert der Algorithmus und w = $a^{\frac{x+1}{2}} \cdot n^{\frac{y}{2}}$ wird zurückgegeben.
4. Gilt $x \ mod \ 2 = 0$, so setze $x = \frac{x}{2}$ und $y = \frac{y}{2}$.
5. Falls $a^x n^y \not\equiv 1 \ mod \ p$, setze $y = y + \frac{p-1}{2}$. Goto 3.

⇒ Zerlegen wir $p - 1 = 2^l q$, so sehen wir, dass der Algorithmus l Schritte benötigen wird.

Lemma 0.11 (*Tonelli-Verfahren für $p \equiv 3 \ mod \ 4$*)

Sei p eine Primzahl mit $p \equiv 3 \ mod \ 4$ und $\left(\frac{a}{p}\right) = 1$. Dann ist

$$\pm w = a^{\frac{p+1}{2}} \ mod \ p$$

eine Quadratwurzel von a modulo p.

Beweis:
Sei $p \equiv 3 \ mod \ 4$, also $p = 3 + 4k$. Dann ist $p - 1 = 2 + 4k = 2 \cdot (1 + 2k)$. Beginnen wir das Tonelli Verfahren mit $x_1 = \frac{p-1}{2} = 1 + 2k, \ y_1 = 0$.
⇒ Da x_1 ungerade ist sind wir fertig und erhalten als Quadratwurzel $a^{\frac{p+1}{2}} \ mod \ p$.

□

Die Gleichung $p = x^2 + dy^2$ und der Cornacchia-Algorithmus

→ Eine Zahl lässt sich zu gegebenem $d \in \mathbb{N}$ genau dann in der Form $p = x^2 + dy^2$ mit ganzen Zahlen x,y schreiben, falls p eine Primzahl ist und $\left(\frac{-d}{p}\right) = 1$ gilt.
→ Gilt $\left(\frac{-d}{p}\right) = -1$, dann ist die Gleichung nicht in $\mathbb{Z}$ lösbar.
→ Gilt $p > d$ und gebe es u,v mit $p = u^2 + dv^2$. Sei $w \in \mathbb{N}$ mit $w^2 \equiv -d \ mod \ p$. Wendet man den EEA auf p und w an, erhält man a_i, x_i, y_i. Ist $a_{i+1} < \sqrt{p} < a_i$, so gilt $p = a_{i+1}^2 + dy_{i+1}^2$

Algorithmus:

Geg. $d \in \mathbb{N}$ und Primzahl $p > d$.

Ges. Lösung der Gleichung $p = x^2 + dy^2$.

1. Falls $\left(\frac{-d}{p}\right) \neq 1$: STOP, die Gleichung ist nicht lösbar.
2. Berechne mit Hilfe des Tonelli-Algorithmus ein $w \in \mathbb{N}$ mit $w^2 \equiv -d \ mod \ p$ und $0 < w < p$.
3. Setze $x = p, b = w$.
4. Solange $x^2 > p$, setze $x_{tmp} = x \ mod \ b$, $b = x \ mod \ b$ *und* $x = x_{tmp}$.
5. Setze $y = \lfloor \sqrt{\lfloor \frac{p-x^2}{d} \rfloor} \rfloor$.
6. Falls $p = x^2 + dy^2$, dann haben wir eine Lösung gefunden, ansonsten ist die Gleichung nicht lösbar.

Quadrate modulo N = pq

→ $a \in \mathbb{Z}$ mit $ggT(a, N)$ ist genau dann ein Quadrat modulo N, wenn $\left(\frac{a}{p}\right) = \left(\frac{a}{q}\right) = 1$ ist.
→ Für N kommt zur Basis a genau dann die Jacobi-Zahl 1 raus, falls $\left(\frac{a}{p}\right) = \left(\frac{a}{q}\right) = \pm 1$.
⇒ Nur Hälfte der Zahlen mit Jacobi-Symbol 1 sind Quadrate.
⇒ Man kennt bisher ohne die Faktorisierung von N keinen Weg zu sagen, ob a Quadrat ist oder nicht.

Die Gleichung $x^2 \equiv a \ mod \ N$ für RSA-Zahl N

→ Betrachte zusammengesetzte Zahl N mit unterschiedlichen ungeraden Primzahlen p,q mit ggT(a,N) = 1.
⇒ Die Gleichung ist genau dann lösbar, falls $\left(\frac{a}{p}\right) = \left(\frac{a}{q}\right) = 1$.

Quadratwurzel der Gleichung
Kennt man die Faktorisierung von N, so kann man die Gleichung wie folgt schnell lösen:

1. Geg: $x_p, x_q \in \mathbb{Z}$ mit $x_p^2 \equiv a \ mod \ p$, $x_q^2 \equiv a \ mod \ q$.
2. Berechne mit Hilfe des EEA $u, v \in \mathbb{Z}$ mit $up + vq = 1$.
3. Berechne mit $\epsilon_p, \epsilon_q \in \{1, -1\}$

$$x_{\epsilon_p, \epsilon_q} = (\epsilon_p x_p v q + \epsilon_q x_q u p) \ mod \ N$$

→ Formel ist abgeleitet vom Chinesischen Restsatz.
⇒ Man findet so alle vier Lösungen der Gleichung.

Faktorsierung von N
Kennt man die vier Lösungen der Gleichung, so kann man auf p und q schließen:

Lemma 0.12

Hat die Gleichung $x^2 \equiv a \ mod \ N$ vier verschiedene Lösungen $x_1, \ldots, x_4$, so gilt

$$\{ggT(x_1 - x_2, N), ggT(x_1 - x_3, N), ggT(x_1 - x_4, N)\} = \{1, p, q\}$$

→ Das Finden aller 4 Wurzeln ist genau so schwer wie das Faktorisieren von N.

Das Rabin-Verschlüsselungsverfahren

→ Sicherheit beruht auf der Tatsache, dass es schwierig ist die Quadratwurzeln einer zusammengesetzten Zahl N zu ziehen,
wenn man die Faktorisierung nicht kennt.

Verfahren:

1. Schlüssel: Wahl von zwei großen Primzahlen p,q mit $p \equiv q \equiv 3 \ mod \ 4$. Dann ist $N = pq$ der öffentliche Schlüssel. Mit dem EEA berechnet man zwei ganze Zahlen u,v mit $up + vq = 1$. Somit ist der private Schlüssel (p,q,u,v).
2. Verschlüsselung:
   (a) Umwandelung der Zeichenfolge in eine Zahlenfolge a_i.
   (b) Zu Verschlüsselung einer Zahlenfolge wird mit dem N die Folge b_i berechnet: $b_i \equiv a_i^2 \ mod \ N$.
3. Entschlüsselung: $c_{i,\epsilon_p,\epsilon_q} = \epsilon_p b_i^{\frac{p+1}{4}} vq + \epsilon_q b_i^{\frac{q+1}{4}} up \ mod \ N$. Nicht eindeutig, da 4 Lösungen pro Block!
   Als Hilfestellung kann man am Ende jedes Blocks den gleichen Buchstaben anhängen.

Angriff mit dem chinesischen Restsatz:
→ Voraussetzung: Der gleiche Text wurde mit zwei unterschiedlichen teilerfremden öffentlichen Schlüsseln verschlüsselt.
→ Dann lässt sich a^2 berechnen durch

$$a^2 \equiv \begin{cases} b_1 \ mod \ N_1 \\ b_2 \ mod \ N_2 \end{cases}$$

wobei gilt $0 \leq a^2 \leq N_1 N_2 - 1$.
⇒ Anschließend muss nur noch die Wurzel gezogen werden.

Vergleich mit RSA:
→ Verschlüsselung im RSA-Verfahren ähnlich, nur dass der Exponent nicht immer 2 war sondern als e im öffentlichen Schlüssel enthalten ist.
⇒ Angriff über den chinesischen Restsatz daher evtl. komplizierter, da e Kongruenzgleichungen gelöst werden müssen und nicht nur 2.
⇒ Angriff auf RSA schwieriger!

Rabin-Williams-Verschlüsselungsverfahren

→ Sicherheit beruht auf der Tatsache, dass es schwierig ist die Quadratwurzeln einer zusammengesetzten Zahl N zu ziehen, wenn man die Faktorisierung nicht kennt.

1. Schlüssel: Wähle Primzahlen $p \equiv 3 \ mod \ 8$, $q \equiv 7 \ mod \ 8$ berechne öffentlichen Schlüssel $N = pq$.
2. Verschlüsselung: Umwandelung der Zeichenfolge in eine Zahlenfolge a_i mit $0 \leq a_i \leq \lfloor \frac{N}{8} \rfloor$ und berechne

$$b_i = \begin{cases} 16(2a_i + 1)^2 \ mod \ N & falls \ \left(\frac{2a_i+1}{N}\right) = 1 \\ 4(2a_i + 1)^2 \ mod \ N & falls \ \left(\frac{2a_i+1}{N}\right) = -1 \end{cases}$$

→ Falls $\left(\frac{2a_i+1}{N}\right) = 0$, so gilt $ggT(2a_i + 1, N) \in \{p, q\}$

3. Entschlüsselung: Berechnung von $m = \frac{(p-1)(q-1)+4}{8}$ und $c_i = b_i^m \ mod \ N$. Damit berechnet man

$$a_i = \begin{cases} \frac{c_i - 4}{8} & falls \ c_i \equiv 0 \ mod \ 4 \\ \frac{N - 4 - c_i}{8} & falls \ c_i \equiv 1 \ mod \ 4 \\ \frac{c_i - 2}{4} & falls \ c_i \equiv 2 \ mod \ 4 \\ \frac{N - 2 - c_i}{4} & falls \ c_i \equiv 3 \ mod \ 4 \end{cases}$$

Goldwasser-Micali-Verschlüsselung

1. Schlüssel: Eine RSA-Zahl $N = pq$ und $k \in \mathbb{N}$ mit $\left(\frac{k}{p}\right) = \left(\frac{k}{q}\right) = -1$.
2. Verschlüsselung: Verschlüsselung einer Bitfolge a_i durch Wahl einer Zufallsfolge z_i mit $ggT(z_i, N) = 1$ und definiert $b_i = k^{a_i} z_i^2 \ mod \ N$.
3. Entschlüsselung: Ausgangsfolge a_i erhält man durch $a_i = \frac{1}{2}(1 - \left(\frac{b_i}{p}\right))$

→ Verschlüsselung hat starke Nachrichtenexpansion: m zu verschlüsselnde Bits liefern m Zahlen der Länge N.
→ Es lässt sich durch die Eigenschaften des Legendre-Symbols schnell bestimmen, ob $k = -1$ oder $k = 2$ wählbar ist.

Das Fiat-Shamir-Identifikationsprotokoll

→ Sicherheit beruht auf der Tatsache, dass man praktisch keine Wurzeln modulo einer RSA-Zahl N ziehen kann, wenn man ihre Faktorisierung nicht kennt.

1. Schlüssel: Eine vertrauenswürdige Zentrale wählt verschiedene große Primzahlen p und q und berechnet den öffentlichen Schlüssel $N = pq$. Jeder Teilnehmer A wählt sich anschließend geheim ein e_A mit $1 \leq e_A \leq N - 1$ und $ggT(N, e_A) = 1$ (siehe RSA!) und berechnet $f_A = e_A^2 \ mod \ N$.
   Der Wert f_A wird veröffentlicht.
2. Identifikationsprotokoll: A will B von seiner Identität überzeugen, indem er eine Wurzel e_A von $f_A \ mod \ N$ kennt. Dazu werden folgenden Schritte hinreichend oft wiederholt:
   (a) A wählt zufällig ein $1 \leq a_i \leq N - 1$ und schickt $b_i = a_i^2 \ mod \ N$ an B.
   (b) Challenge: B schickt zufällig ein $e_i \in \{0, 1\}$ an A.
   (c) Response: A schickt an B
   $$c_i = \begin{cases} a_i & falls \ e_i = 0 \\ e_A a_i & falls \ e_i = 1 \end{cases}$$
   (d) Test: B akzeptiert den Schritt, wenn
   $$c_i^2 \equiv \begin{cases} b_i \ mod \ N & falls \ e_i = 0 \\ f_A b_i \ mod \ N & falls \ e_i = 1 \end{cases}$$
   Wurden genug Schritte akzeptiert gilt die Identität von A als bestätigt.

→ Bei Täuschung muss der Teilnehmer sich im vornherein festlegen, welches e_i B wählen wird. Daher hat er nur eine 50/50 Chance.

Blum-Goldwasser-Verschlüsselung

→ Sicherheit beruht auf der Tatsache, dass man praktisch keine Wurzeln modulo einer RSA-Zahl N ziehen kann, wenn man ihre Faktorisierung nicht kennt.

1. Schlüssel: Wahl zweier Primzahlen $p \equiv 3 \ mod \ 4$, $q \equiv 3 \ mod \ 4$ und Veröffentlichung von $N = pq$.
2. Verschlüsselung:
   (a) Umwandelung der Zeichenfolge in eine Zahlenfolge $a_0, \ldots, a_{l-1}$.
   (b) Wahl einer Zufallszahl z mit $ggT(z, N) = 1$ und Berechnung von $x_0 = z^2 \ mod \ N$. Rekursive Berechnung von $x_i = x_{i-1}^2 \ mod \ N$ und $z_i = x_i \ mod \ N \quad \forall i = 1, \ldots, l$.
   (c) Berechne Chiffretext durch $b_i = a_i + z_i \ mod \ N$ und versende $b_0, \ldots, b_{l-1}, x_l$.
3. Entschlüsselung: Mit der erhaltenen Zahlenfolge lässt sich
   $$x_0 = x_l^{\frac{(p-1)(q-1)+4}{8}^l \ mod \ (p-1)(q-1)}$$
   und dann rekusiv $z_i = x_i \ mod \ N$, $a_i = b_i - z_i \ mod \ N$, $x_{i+1} = x_i^2 \ mod \ N$ berechnen.

→ Blum-Goldwasser-Verschlüsselung ist eine Stromchiffrierung, wobei aus der rekursiven Verschlüsselung eine Pseudozufallszahlenfolge erzeugt wird.

Lucas-Folgen

Definition 0.13 (*Lucas-Folge*)

Zu vorgegebenen ganzen Zahlen P und Q werden die Lucas-Folgen $(U_i(P,Q))_{i\geq 0}$ und $(V_i(P,Q))_{i\geq 0}$ rekursiv durch die Formeln

$$U_0(P,Q) = 0, \quad U_1(P,Q) = 1$$
$$U_i(P,Q) = P \cdot U_{i-1}(P,Q) - Q \cdot U_{i-2}(P,Q)\ \forall i \geq 2$$

und

$$V_0(P,Q) = 2, \quad V_1(P,Q) = P$$
$$V_i(P,Q) = P \cdot V_{i-1}(P,Q) - Q \cdot V_{i-2}(P,Q)\ \forall i \geq 2$$

→ Beispiel: Für $U_n(1,-1)$ erhält man die Fibonacci-Folge.
Formeln:

- $U_i(-P,Q) = (-1)^{i-1}U_i(P,Q)$
- $V_i(-P,Q) = (-1)^i V_i(P,Q)$
- $U_i(0,Q) = \begin{cases} 0 & \text{für i gerade} \\ (-Q)^{\frac{i-1}{2}} & \text{für i ungerade} \end{cases}$
- $V_i(0,Q) = \begin{cases} 2\cdot(-Q)^{\frac{i}{2}} & \text{für i gerade} \\ 0 & \text{für i ungerade} \end{cases}$
- $U_i(P,0) = P^{i-1} \quad \forall i \geq 2$
- $V_i(P,0) = P^i \quad \forall i \geq 1$
- Gilt $P^2 - 4Q = 0$, so ergibt sich

$$U_i(P,Q) = i \cdot \frac{P}{2}^{i-1} \quad \text{für } i \geq 2$$
$$V_i(P,Q) = 2 \cdot \frac{P}{2}^{i} \quad \text{für } i \geq 1$$

- $U_{i+j} = U_iV_j - Q^jU_{i-j}$ bzw. $V_{i+j} = V_iV_j - Q^jV_{i-j}$
- $U_{2i} = U_iV_i,\ U_{2i+1} = U_{i+1}V_i - Q^i$
- $V_{2i} = V_i^2 - 2Q^i,\ V_{2i+1} = V_{i+1}V_i - PQ^i$

Darstellung der Lucas-Folge mit Hilfe des Nullstellenpolynoms $x^2 - Px + Q$

Seien $P,Q \in \mathbb{Z}$ und $D = P^2 - 4Q$ und betrachte die Nullstellen $\alpha = \frac{P+\sqrt{D}}{2}$ und $\beta = \frac{P-\sqrt{D}}{2}$ des Polynoms $x^2 - Px + Q$. Dann gilt:

- $\alpha + \beta = P$
- $\alpha\beta = Q$
- $\alpha - \beta = \sqrt{D}$

und die Lucas-Folgen lassen sich darstellen als:

$$U_i(P,Q) = \begin{cases} i\alpha^{i-1} & \text{für } \alpha = \beta \\ \frac{\alpha^i - \beta^i}{\alpha - \beta} & \text{für } \alpha \neq \beta \end{cases}$$
$$V_i(P,Q) = \alpha^i + \beta^i$$

Berechnungsmethoden für $U_i(P,Q),\ V_i(P,Q)\ mod\ n$

Variante 1: Mit Square-and-multiply (für $i \geq 1$)

$$\begin{pmatrix} U_{i-1} & V_{i-1} \\ U_i & V_i \end{pmatrix} = \begin{pmatrix} 0 & 1 \\ -Q & P \end{pmatrix}^{i-1} \begin{pmatrix} 0 & 2 \\ 1 & P \end{pmatrix}$$

Variante 2: Trick über Formeln

$$U_{2i} = U_iV_i,\ U_{2i-1} = U_iV_{i-1} - Q^{i-1}$$
$$V_{2i} = V_i^2 - 2Q^i,\ V_{2i-1} = V_iV_{i-1} - PQ^{i-1}$$

Ziel: Berechnung von $U_j,\ V_j$ für große j.
Vorgehen:

1. Bilde eine Folge der Form $j \rightarrowtail \lceil \frac{j}{2} \rceil$, bis j=1.
2. Berechne mit den obigen Formeln von unten nach oben die einzelnen j, bis der gewünschte Wert berechnet ist.

j	$U_j(1,-1)$	$V_j(1,1)$
1	1	1
2	1	3
4	3	7
7	13	29
13	233	521
25	75025	167761
50	12586269025	28143753123
99	218922995834555169026	489526700523968661124

Was ist $U_{p-\frac{D}{P}}(P,Q)$ mod p (mit $D = P^2 - 4Q$)?
→ p ungerade Primzahl
→ $ggT(p,2QD) = 1$
⇒ $\forall k \in \mathbb{N} : U_{k(p-\frac{D}{p})}(P,Q) \equiv 0\ mod\ p$

Lucas-Primzahltest und Lucas-Pseudoprimzahl
Betrachte nun für natürliche, zusammengesetzte Zahlen n mit $ggT(n,2QD) = 1$ die Formel $U_{n-\frac{D}{n}}(P,Q)\ mod\ n$:
→ Voraussetzungen: $D = P^2 - 4Q \neq 0$, $Q \neq 0$ und $ggT(p,2QD) = 1$
→ Wir haben genau dann eine Lucas-Pseudoprimzahl gefunden, wenn $U_{n-\frac{D}{n}}(P,Q) \equiv 0\ mod\ n$ gilt und sagen, n besteht den Lucas-Test zum Parameter (P,Q).
→ Gilt $U_{n-\frac{D}{n}}(P,Q) \not\equiv 0\ mod\ n$, so ist n zusammengesetzt.

Zusatz: Gilt zusätzlich noch $P = 0$ oder $P^2 - kQ = 0$ für ein k=1,2,3, so ist jede natürliche Zahl n eine Lucas-Pseudoprimzahl.

Der Baillie-PSW-Primzahltest

1. Es soll untersucht werden, ob eine natürliche Zahl n zusammengesetzt ist oder nicht.
2. Zunächst wird auf kleine Primteiler getestet. (Oft ist man dann schon fertig)
3. Führe den Miller-Rabin-Test zur Basis 2 durch. Besteht n den Test nicht, so ist n zusammengesetzt.
4. Suche ein D mit $\left(\frac{D}{n}\right) = -1$. Dafür muss ausgeschlossen werden, dass n eine Quadratzahl ist. Allerdings sind bisher nur zwei Quadratzahlen bekannt, die den Miller Rabin Test bestehen. Deren Basen sind Primzahlen. Geeignet als D sind ±Primzahlen ab 5.
5. Findet man hierbei zufällig ein D mit $\left(\frac{D}{n}\right) = 0$ und $ggT(n, D) > 1$, so ist n zusammengesetzt. (sehr unwahrscheinlich!)
6. Setze $P = 1,\ Q = \frac{1-D}{4}$ und teste $U_{n-\frac{D}{n}}(P, Q)\ mod\ n = 0$. Wenn ja ist n wahrscheinlich prim.

⇒ Bisher wurde noch keine zusammengesetzte Zahl gefunden, die den BPSW-Test besteht.

Primzahlbeweise

Variante 1: Kenntnis der Faktorisierung von n-1
→ Sei Zerlegung von $n - 1 = q_1^{e_1} \dots q_r^{e_r}$ bekannt.
⇒ Findet man nun eine Zahl a mit $a^{n-1} \equiv 1\ mod\ n$ und $a^{\frac{n-1}{q_i}} \not\equiv 1\ mod\ n\ \forall i = 1, \dots, r$, so ist n eine Primzahl.

Variante 2: Kenntnis der Faktorisierung von n+1
→ Sei Zerlegung von $n + 1 = q_1^{e_1} \dots q_r^{e_r}$ bekannt.
→ Findet man P,Q und $D = P^2 - 4Q$ mit $ggT(n, 2QD) = 1$ und $U_{n+1}(P, Q) \equiv 0\ mod\ n$ und $ggT(n, U_{\frac{n+1}{q_i}}(P, Q)) = 1$ $\forall i = 1, \dots, r$, so ist n eine Primzahl.
→ Zahlen n, bei denen n+1 eine besonders einfache Primfaktorzerlegung hat, sind die Mersenne-Zahlen $n = 2^k - 1$.
⇒ Ist dabei n eine Primzahl, so auch k!

Lucas-Lehmer-Test für Mersenne-Zahlen
→ Für $l \geq 3$ und $n = 2^l - 1$ gilt:

$$n\ ist\ eine\ Primzahl\ \Leftrightarrow V_{2^{l-2}}(4, 1) \equiv 0\ mod\ n$$

Alternative Formulierung im Lucas-Lehmer-Test:
Für $l \geq 3$ und $n = 2^l - 1$ wird rekursiv eine Folge v_i definiert durch:
$v_0 = 4$ und $v_{i+1} = (v_i^2 - 2)\ mod\ n$
Dann gilt:

$$n\ ist\ eine\ Primzahl\ \Leftrightarrow v_{l-2} = 0$$

Lucas-RSA-Verschlüsselung
In Anlehnung an RSA betrachte folgenden Zusammenhang:
Sei $N = pq$ eine RSA-Zahl und seien $ed \equiv 1\ mod\ (p^2-1)(q^2-1)$. Dann gilt

$$b \equiv V_e(a, 1)\ mod\ N \Rightarrow a \equiv V_d(b, 1)\ mod\ N$$

LUC-Verschlüsselung:

1. Schlüssel: Wähle $p_A,\ q_A$ ungerade Primzahlen und berechne $N_A = p_A q_A$. Konstruiere anschließend mit dem EEA Zahlen $e_A,\ d_A$ mit $e_A d_A \equiv 1\ mod\ (p_A^2 - 1)(q_A^2 - 1)$. → Öffentlicher Schlüssel ist (N_A, e_A), privater Schlüssel ist (N_A, d_A).
2. Verschlüsselung: Wandel Nachricht nach vereinbarten Schema in Zahlenfolge um und schicke $b_i = V_{e_A}(a_i, 1)\ mod\ N_A$
3. Entschlüsselung: Berechne Klartext mit $a_i = V_{d_A}(b_i, 1)\ mod\ N_A$.

→ Wie auch bei RSA sind e und d ungerade natürliche Zahlen.
→ $e = 3$ kommt als öffentlicher Schlüssel nicht in Frage, da für jede Primzahl $p \geq 5$ gilt: $ggT(3, p^2 - 1) = 3$

Angriff auf LUC mit dem chinesischen Restsatz:

1. Voraussetzung: Eine Nachricht a geht an min. 5 Personen, die jeweils öffentliche Schlüssel (N_i,5) haben.
2. Abgefangen wird Chiffretext $b_i = V_5(a, 1)\ mod\ N_i$.
3. Mit dem chinesischen Restsatz berechnet man

$$c \equiv \begin{cases} b_1\ mod\ N_1 \\ b_2\ mod\ N_2 \\ b_3\ mod\ N_3 \\ b_4\ mod\ N_4 \\ b_5\ mod\ N_5 \end{cases} \quad \text{mit } 0 \leq c \leq N_1 N_2 N_3 N_4 N_5 - 1$$

4. a ist Nullstelle des Polynoms $f(x) = x^5 - 5x^3 + 5x - c$

Alternativer privater Schlüssel:
→ Vorteil: Es kann bei der Entschlüsselung auf den chinesischen Restsatz verzichtet werden.
→ Privater Schlüssel besteht aus $d_{1,1},\ d_{1,-1},\ d_{-1,1},\ d_{-1,-1}$.
Berechnung:

$$ed_{1,1} \equiv 1\ mod\ (p-1)(q-1)$$
$$ed_{1,-1} \equiv 1\ mod\ (p-1)(q+1)$$
$$ed_{-1,1} \equiv 1\ mod\ (p+1)(q-1)$$
$$ed_{-1,-1} \equiv 1\ mod\ (p+1)(q+1)$$

Entschlüsselung:

$$a_i = \begin{cases} V_{d_{1,1}}(b_i, 1)\ mod\ N & falls\ \frac{b_i^2-4}{p} = 1,\ \frac{b_i^2-4}{q} = 1 \\ V_{d_{1,-1}}(b_i, 1)\ mod\ N & falls\ \frac{b_i^2-4}{p} = 1,\ \frac{b_i^2-4}{q} = -1 \\ V_{d_{-1,1}}(b_i, 1)\ mod\ N & falls\ \frac{b_i^2-4}{p} = -1,\ \frac{b_i^2-4}{q} = 1 \\ V_{d_{-1,-1}}(b_i, 1)\ mod\ N & falls\ \frac{b_i^2-4}{p} = -1,\ \frac{b_i^2-4}{q} = -1 \end{cases}$$

Diffie-Hellman-Schlüsselaustausch mit Lucas-Folgen
Erinnerung: Wir nennen x einen diskreten Logarithmus von a zur Basis g mod p, wenn gilt: $a \equiv g^x \ mod\ p$
→ Diskrete Logarithmen lassen sich im Allgemeinen schlecht berechnen und müssen nicht existieren.

DH-Schlüsselaustausch mit Lucas-Folgen:

Geg: Ungerade Primzahl p und g mit $ggT(p,g) = 1$.

1. A wählt privaten Schlüssel e_A und veröffentlicht $f_A = V_{e_A}(g,1)\ mod\ p$.
2. B wählt privaten Schlüssel e_B und veröffentlicht $f_B = V_{e_B}(g,1)\ mod\ p$.
3. Gemeinsamer Schlüssel berechnet sich dann durch

$$k_{AB} = V_{e_A e_B}(g,1) \equiv V_{e_A}(f_B,1) \equiv V_{e_B}(f_A,1)\ mod\ p$$

Lucas-ElGamal-Verschlüsselung

1. Schlüssel: Wähle Primzahl p und $g \in \mathbb{Z}$. Wähle geheime Zahl e_A und veröffentliche $f_A = V_{e_A}(g,1)\ mod\ p$.
2. Verschlüsselung: Wandel Text in Zahlenfolge a_i um. Wähle anschließen Zufallszahl z_i und gib als Chiffretext Zahlenpaarfolge $b_i = V_{z_i}(g,1)\ mod\ p$ und $c_i = a_i \cdot V_{z_i}(f_A,1)\ mod\ p$.
3. Entschlüsselung: Berechne $a_i = \frac{c_i}{V_{e_A}(b_i,1)}\ mod\ p$

Zur Erschwerung von Angriffen können bei der Schlüsselerstellung noch folgende Bedingungen gestellt werden:

1. Die Faktorisierung von p+1 ist bekannt.
2. p+1 sollte mindestens einen großen Primteiler haben.
3. $\frac{g^2-4}{p} = -1$
4. $V_{\frac{p+1}{q_i}}(g,1) \not\equiv 2\ mod\ p$

Elliptische Kurven mit Additionsgesetz

Definition 0.14 (*Elliptische Kurve (a)*)

1. *Sei K ein Körper der Charakteristik ≠ 2,3*
2. *Sei eine elliptische Kurve E über K definiert*
3. *Sei a,b ∈ K und* $\Delta = 4a^3 + 27b^2 \neq 0$
4. *Somit gibt es folgende Weierstraß-Gleichung für E:*

$$y^2 = x^3 + ax + b$$

→ Elliptische Kurven skizziert man mit reellen Bilder
→ Charakteristiken 2,3 startet man mit anderer Gleichung
→ Charakteristik: Kleinste Zahl, so dass bei Addition des gleichen Elements das neutrale Element entsteht.
→ Für Kryptographie ist Charakteristik 2 noch interessant
→ Aus der Bedingung $\Delta = 4a^3 + 27b^2 \neq 0$ folgt:
⇒ Die Kurve besitzt in jedem Punkt eine Tangente

Definition 0.15 (*E(K) (b)*)

1. *Sei E eine elliptische Kurve über K*
2. *Sei E gegeben durch:* $y^2 = x^3 + ax + b$
3. *Sei O ein unendlich ferner Punkt*
4. *Definition der K-Rationalen Punkte:*

$$E(K) = \{(x,y) \in K \times K : y^2 = x^3 + ax + b\} \cup \{O\}$$

Definition 0.16 (*Addition algebraisch (c)*)

1. *Sei E eine Kurve nach obiger Definition*
2. *Sei* $P_1 = (x_1,y_1), P_2 = (x_2,y_2) \in E(K)$
3. *Addition auf K-Rationalen Punkten E(K):*

$$\forall P \in E(K) : P + O = O + P = P$$

$$falls:\ x_1 = x_2, y_1 + y_2 = 0 \Rightarrow P_1 + P_2 = O$$

$$sonst:\ P_1 + P_2 = (x_3, y_3)$$

$$mit:\ x_3 = m^2 - x_1 - x_2, y_3 = m(x_1 - x_3) - y_1$$

$$wobei\ m:\ m = \begin{cases} \frac{y_1 - y_2}{x_1 - x_2}, falls\ x_1 \neq x_2 \\ \frac{3x_1^2 + a}{2y_1}, falls\ x_1 = x_2, y_1 = y_2 \neq 0 \end{cases}$$

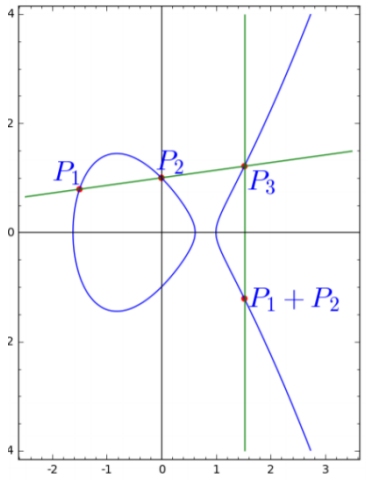
4. *Inverse:*

$$P = (x,y) \Rightarrow -P = (x,-y)$$

5. *Trick: Tangente aus* P_1, P_2 *schneidet Kurve in P3:*

$$P_3 = -(P_1 + P_2)$$

→ O ist als Körpereigenschaft das neutrale Element
→ Geometrische Idee hinter Addition:
⇒ Seien zwei Punkte P_1, P_2 gegeben, so schneidet die Verbindungsgeraden der beiden Punkte die Kurve im Allgemeinen in einem weiteren Punkt $P_3 = (x_3, y_3)$. Dabei ist die Summe der Punkte der an der x-Achse gespiegelte Punkt: $P_1 + P_2 = (x_3, -y_3)$
⇒ Gibt es keinen weiteren Punkt, so lösen wir dieses Problem mit Hilfe des unendlich fernen Punktes O.



Definition 0.17 (*Berechnung von $n \cdot P$ (d)*)

1. *Es handelt sich hierbei um eine n-fache Addition von P*
2. *Verwendet wird ein additives Square-and-multiply*
3. *$P \in E(K)$ und $n \in \mathbb{N}_0$*
4. *Initialisierung: result = O, R = P*
5. *while $n > 0$ do:*

   Falls n ungerade: result = result + R
   $R = R + R$
   $n = \lfloor n/2 \rfloor$
6. *Ergebnis von $n \cdot P$ steht in result*

Beispiel:

1. E über $\mathbb{F}_5$ durch $y^2 = x^3 + x + 1$
2. E($\mathbb{F}_5$)=$\{O, (0,1), (0,4), (2,1), (2,4), (3,1), (3,4), (4,2), (4,3)\}$
3. P = (0, 1), 2P = (4, 2), 3P = (2, 1), 4P = (3, 4)
4. 5P = (3, 1), 6P = (2, 4), 7P = (4, 3), 8P = (0, 4), 9P = O
5. E($\mathbb{F}_5$) ist zyklische Gruppe der Ordnung 9

Definition 0.18 (*Teilungspunkte (e)*)

1. *Sei $n \in \mathbb{N}$ und $P \in E(K)$*
2. *Definition eines n-Teilungspunktes: $n \cdot P = O$*
3. *$P = (x, y) \in E(K) \backslash \{O\}$ ist 2-Teilungspunkt wenn:*

   $y = 0 \Rightarrow x^3 + ax + b = 0$
4. *Eigenschaften: 2P = O, P = -P, (x,y) = (x,-y)*
5. *Berechnung der 2-Teilungspunkte:*
   (a) *Polynom besitzt keine Nullstellen in K:*
   $\Rightarrow E(K)[2] = \{O\}$
   (b) *Polynom besitzt eine Nullstelle c in K:*
   $\Rightarrow E(K)[2] = \{O, (c, 0)\}$
   (c) *Polynom besitzt drei Nullstellen in K:*
   $\Rightarrow E(K)[2] = \{O, (c_1, 0), (c_2, 0), (c_3, 0)\}$

Definition 0.19 (*Punkte in $E(\mathbb{F}_p)$ finden (f)*)

1. *Sei E eine über $\mathbb{F}_p$ definierte elliptische Kurve*
2. *Sei $p >= 5$ eine Primzahl und $x_0 \in \mathbb{F}_p$*
3. *Sei $z = \left(\frac{x_0^3 + ax_0 + b}{p}\right)$*
4. *Falls z = -1: Es gibt keine solchen Punkte*
5. *Falls z = 0: Koordinaten des Punktes: $(x_0, 0)$*
6. *Falls z = 1: Seit $w = \sqrt{x_0^3 + ax_0 + b}$*

   Erster Punkt: (x_0, w)
   Zweiter Punkt: $(x_0, p - w)$

Definition 0.20 (*Anzahl von $\#E(\mathbb{F}_p)$ bestimmen (g)*)

1. *Sei E eine über $\mathbb{F}_p$ definierte elliptische Kurve*
2. *Für die Anzahl der $\mathbb{F}_p$-rationalen Punkte gilt:*

$$\#E(\mathbb{F}_p) = p + 1 + \sum_{x \in \mathbb{F}_p} \left(\frac{x^3 + ax + b}{p}\right)$$

Definition 0.21 (*Satz von Hasse (h)*)

1. *Sei E eine über $\mathbb{F}_p$ definierte elliptische Kurve*
2. *Somit gilt:*

$$p + 1 - 2\sqrt{p} < \#E(\mathbb{F}_p) < p + 1 + 2\sqrt{p}$$

3. *Vereinfacht: $|\#E(\mathbb{F}_p) - (p+1)| < 2\sqrt{p}$*

Definition 0.22 (*$\#E(\mathbb{F}_p)$ für b=0 schnell bestimmen*)

1. *Sei E eine elliptische Kurve über $\mathbb{F}_p$ mit $(a \neq 0)$*
2. *Fall 1: $p \equiv 3$ mod 4: $\#E(\mathbb{F}_p) = p + 1$*
3. *Fall 2: $p \equiv 1$ mod 4: $\exists\, m, n \in \mathbb{N}$ mit $p = m^2 + n^2$ (Cornacchia-Algorithmus):*

   $\#E(\mathbb{F}_p) \in M = \{p+1-2m, p+1+2m, p+1-2n, p+1+2n\}$
4. *Die Menge auf ein Element reduzieren:*

   Sei nun $P \in \#E(\mathbb{F}_p)$ und $N \in M$

   Es gilt $\#E(\mathbb{F}_p) \cdot P = O$

   Ist $N \cdot P \neq O$ so gilt: $N \neq \#E(\mathbb{F}_p)$

   Ist $N \cdot P = O$ so gilt: $N \neq \#E(\mathbb{F}_p)$ oder $N = \#E(\mathbb{F}_p)$

   Mit der Wahl von verschieden Punkten P wird die Menge M schnell auf 1 Element reduziert
5. *Algorithmus funktioniert für Primzahlen $p > 233$ immer, bei kleineren evtl Endlosschleife.*
6. *Heute: Verwendung von Schoof-Algorithmus und SEA-Algorithmus.*

Definition 0.23 (*Punktkompression (i)*)

1. *Sei E eine über $\mathbb{F}_p$ definierte elliptische Kurve*
2. *Sei $P = (x, y) \in E(\mathbb{F}_p)$ und $-P = (x, -y) = (x, p - y)$*
3. *Unterscheidung von P und -P, gegeben: (x,y)*

   Berechne y' aus Funktion von E mit x
   P, wenn $y' \equiv y \bmod 2$
   -P, wenn $y' \equiv (p - y) \bmod 2$

Definition 0.24 (*Diffie-Hellman-Schlüsselaustausch (j)*)

1. *x ist diskreter Logarithmus von Q zur Basis P in Gruppe $E(\mathbb{F}_p)$, wenn gilt:*

$$Q = x \cdot P$$

2. *Sicherheit hängt von der Berechnung dieses diskreten Logarithmus ab, was im Allgemeinen schwierig ist*
3. *Primzahl p, $E: y^2 = x^3 + ax + b$ über $\mathbb{F}_p$ und $P \in E(\mathbb{F}_p)$*
4. *A wählt privaten Schlüssel e_A*
5. *A berechnet öffentlichen Schlüssel $Q_A = e_A \cdot P$*
6. *B wählt privaten Schlüssel e_B*
7. *B berechnet öffentlichen Schlüssel $Q_B = e_B \cdot P$*
8. *Gemeinsamer Schlüssel:*

$$(x_k, y_k) = e_A e_B \cdot P = e_A \cdot Q_B = e_B \cdot Q_A$$

Definition 0.25 (*ElGamal-Verschlüsselung (k)*)

1. *Sicherheit: Sicherheit hängt von der Berechnung des diskreten Logarithmus und den gewählten Zufallszahlen ab*
2. *Schlüssel: Man einigt sich auf Primzahl $p > 3$, elliptische Kurve $E(\mathbb{F}_p)$ und $P \in E(\mathbb{F}_p)$ Jeder Teilnehmer wählt privaten Schlüssel e_A und berechnet daraus den öffentlichen Schlüssel $Q_A = e_A P$*
3. *Verschlüsselung: B will Nachricht an A schicken*
   → B besorgt sich den Schlüssel Q_A von A
   → Nachricht wird in Zahlen $a_i \in \mathbb{F}_p$ übersetzt
   → Für alle a_i wählt B eine Zufallszahl $z_i \in \mathbb{N}$
   Nun berechnet B nacheinander:

$$R_i = z_i \cdot P = (x_{R_i}, y_{R_i})$$
$$S_i = z_i \cdot Q_A = (x_{S_i}, y_{S_i})$$
$$t_i = a_i + x_{S_i}$$

   Verschlüsselte Nachricht: (x_{R_i}, y_{R_i}, t_i)
4. *Entschlüsselung: A berechnet*

$$S_i = e_A \cdot R_i = (x_{S_i}, y_{S_i})$$
$$a_i = t_i - x_{S_i}$$

   Wandel a_i noch in den Ausgangstext um.
5. *Angriff:*
   → e_A ist diskreter Logarithmus von Q_A zu P
   → z_i ist diskreter Logarithmus von R_i zu P
   → z_i erraten

Definition 0.26 (*Isomorphie (m)*)

1. *Seien $E = y^2 = x^3 + ax + b$, $E' = y^2 = x^3 + a'x + b'$*
2. *E, E' isomorph über K, wenn ein $u \in K^*$, sodass:*

$$a' = u^4 a \text{ und } b' = u^6 b$$

3. *$(x, y) \rightarrowtail (u^2 x, u^3 y)$ liefert Gruppenisomorphismus*

Definition 0.27 (*j-Invariante (n / o)*)

1. *Sei E eine über $\mathbb{F}_p$ definierte elliptische Kurve*
2. *Berechnung der j-Invariante:*

$$j = j(E) = 1728 \cdot \frac{4a^3}{4a^3 + 27b^2}$$

3. *$j = 0$, wenn $y^2 = x^3 + b$ (mit $b \neq 0$)*
4. *$j = 1728$, wenn $y^2 = x^3 + ax$ (mit $a \neq 0$)*
5. *Sind E und E' isomorph über K so gilt: j(E) = j(E')*
6. *Gilt j(E) = j(E'), so sind E und E' isomorph über $\bar{K}$*

Definition 0.28 (*j-Invariante und Primzahlen*)

1. *Sei $p >= 5$ eine Primzahl*
2. *Anzahl $\mathbb{F}_p$-Isomorphieklassen ellip. Kurven über $\mathbb{F}_p$:*

$$2p + \begin{cases} 6 \text{ für } p \equiv 1 \ mod\ 12 \\ 2 \text{ für } p \equiv 5 \ mod\ 12 \\ 4 \text{ für } p \equiv 7 \ mod\ 12 \\ 0 \text{ für } p \equiv 11 \ mod\ 12 \end{cases}$$

Faktorisieren mit elliptischen Kurven (p)
→ Bekannt ist (p-1)-Methode von Pollard
⇒ Mit $ggT(2^{kgV(1,\ldots,K)} - 1, n)$ kann man evtl. nichttriviale Teiler über die Gruppe $\mathbb{F}_p^*$ finden.
Idee von Lenstra: Ersetze die Gruppe durch die elliptische Kurve $E_{a,b}(\mathbb{F}_p)$ mit der Gleichung $y^2 = x^3 + ax + b$.
Grundidee:
Sei $n > 1$ eine zusammengesetzte Zahl mit $ggT(n, 6) = 1$.

1. Wähle eine natürliche Zahl K.
2. Wähle $a, x, y \in \mathbb{Z}$ und berechne $b = (y^2 - x^3 - ax)\ mod\ n$.
3. Berechne auf elliptischer Kurve: $kgV(1, \ldots, K) \cdot P$
4. keinen nichttrivialen Teiler gefunden: ändere entweder $a, x, y \in \mathbb{Z}$ oder K.

Das Silver-Pohling-Hellman-Verfahren (s)
→ Annahme: N hat glatte Gruppenordnung
→ Glatte Gruppenordnung: $N = \#E(\mathbb{F}_p)$ kann mit kleinen, teilerfremden Zahlen d_i faktorisiert werden.
→ Ziel: Lösen der Gleichung $xP = Q$, wobei $P, Q \in E(\mathbb{F}_p)$
→ Gilt $xP = Q$, so auch $x\frac{N}{d_i}P = \frac{N}{d_i}Q$
→ Es gilt NP = O.
⇒ $\{x \in \mathbb{Z} \mid x\frac{N}{d_i}P\} = \{0 \leq x \leq d_i - 1 \mid x\frac{N}{d_i}P\}$
⇒ Ist d_i klein, so kann man durch ausprobieren ein x_i mit

$$x_i \frac{N}{d_i} P = \frac{N}{d_i} Q \quad 0 \leq x_i \leq d_i - 1$$

finden oder die Unlösbarkeit der Gleichung feststellen.
Es gibt zwei Fälle:

1. Es gibt ein i, sodass obige Gleichung nicht lösbar ist, so existiert auch keine Lösung der Gleichung $xP = Q$.
2. Man findet für alle i eine Zahl x_i, welches die obige Gleichung erfüllt. Dann berechnet man mit dem chinesischen Restsatz ein $\tilde{x}$ mit $\tilde{x} \equiv x_i\ mod\ d_i$ für alle i.
   ⇒ Dann gilt $\tilde{x}P = Q$.