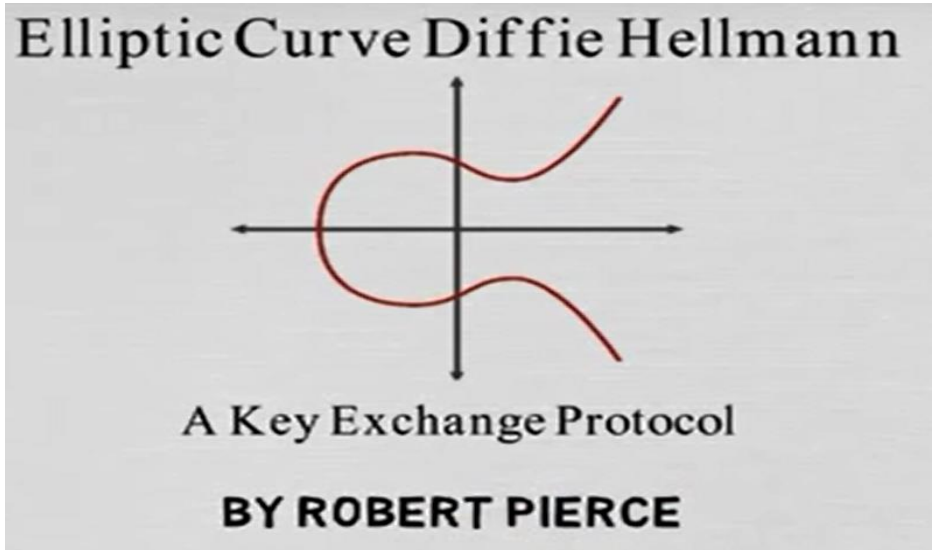
J ∫ Skripte

Pr 9. am Di. 16. April
vorgeholt für Mo.
22. April

Kryptologie ICS.KRYPTO

Folien zu den Semesterwochen 9 & 10 «Public Key Kryptographie III & IV: ECC und Weiteres», FS 24, V6.2

In Präsenz 9 bis Kap. 5 (ohne Kap. 3.5), dann als *HA* auf Präsenz 10, Kap. 6.1 & 6.2. In Präsenz 10, Kap. 3.5 & Kap. 6.3 ff.



https://ch.video.search.yahoo.com/search/video?fr=sfp&p=video+elliptische+kurven#id=7&vid=5b6dcda602db2cab748fdbf99c603a30&action=view

©Josef Schuler, dipl. math., dipl. Ing. NDS ETHZ, MSc Applied IT-Security, Feldhof 25, 6300 Zug, j.schuler@bluewin.ch resp. josef.schuler@hslu.ch

Inhaltsübersicht

- Die Elliptischen Kurven
  - Eine Einführung in die ECC
  - Div. Berechnungen mit ECC
  - Der Zusammenhang mit der Gruppentheorie
  - Verschlüsselung (nach Volker Müller) mit ECC
  - DH-Schlüsselaustausch mit ECC
- Weiteres zu Signaturen
  - Doppelunterschriften, allgemein und mit RSA
  - Blinde Unterschriften, allgemein und mit RSA
- Kurze Betrachtung von Laufzeiten RSA-ECC

Kap. 3.5 → Pr. 10

Pr 10

- ***Bemerkung:***
  - In diesem Thema werden nun die Aspekte der abstrakten Algebra (Einführung in die Gruppen, Körper usw.) aus den Folien von Präsenz 6, Kap. 6 **«Mathematische Grundlagen der asymmetrischen Kryptographie»**, relevant.
  - Bitte beachten Sie dazu auch das Kap. 8.1 in JS Skript „Elliptische Kurven Kryptosysteme, ECCS und weitere Aspekte zu Signaturen, speziell von RSA“.

Verweise zur Literatur

- ***Für die Theorie:***
  - JS Skript „Elliptische Kurven Kryptosysteme, ECCS und weitere Aspekte zu Signaturen, speziell von RSA“. Im Folgenden jeweils mit JS Skript "ECCS" abgekürzt.
  - Im JS Skript „Einführung in die Kryptologie“, Kap. 11 ist nur eine rudimentäre Einführung ins Thema Elliptische Kurven enthalten, daher Spezialskript "ECCS".
  - In [CP-D] die Kap. 8.2, 9, 10.1 & 10.2.
- ***Aufgaben***
  - Aufgaben und Beispiele in den Folien & JS Skript „ECCS“.
  - JS Skript „Aufgaben und Lösungen zum Modul KRYPT an der HSLU-I“, Kap. 2.4, „Aufgaben zu den Präsenzen 9 & 10.
- ***Beachten Sie, dass…***
  - ***… die Aufgabennummerierung im JS Skript „ECCS“ und in den vorliegenden Folien stimmen nicht überein!***
  - … die Kapitelnummerierung in den Folien derjenigen im JS Skript „ECCS“ entspricht.
  - … nur das Bearbeiten und Lernen der Folien nicht genügt. Das Durcharbeiten des oben erwähnten JS Skripte „ECCS“ ist absolut zentral zum Bestehen der MEP, dies gilt insbesondere für die Aufgaben und Beispiele in „ECCS“.

Lernziele

- Ich kann bei einer gegebenen Gleichung überprüfen, ob sie eine Elliptische Kurve darstellt.
- Ich kann die Nichtsingularitätsbedingung einer EC überprüfen.
- Ich kann entscheiden, ob ein gegebener Punkt auf einer gegebenen Elliptischen Kurve liegt oder nicht.
- Ich kann eine Punktverdoppelung einer EC berechnen.
- Ich kann eine Kurvenpunktaddition einer EC berechnen.
- Ich kann einen "double and add" Algorithmus durchführen.
- Ich kann mit einer EC eine Verschlüsselung nach Volker-Müller durchführen.
- Ich kann den DH-Schlüsselaustausch auf Basis von EC durchführen.
- Ich kann alle Punkte einer EC bestimmen.
- Ich kann die Anzahl der Punkte einer EC bestimmen (Theorem von Hasse).
- Ich kann die Ordnung eines Punktes einer EC bestimmen.
- Ich kann eine Codebuchanalyse mit dem RSA durchführen.
- Ich kann eine RSA Verschlüsselung verfälschen.
- Ich kann das RSA Signaturverfahren erklären.
- Ich kann eine Meldung mit RSA signieren.
- Ich kann den Unterschied zwischen RSA und DSA sowie Schnorr erklären.
- Ich kann bei einer Meldung das blinde Signaturverfahren anwenden.
- Ich kann beide Typen Doppelunterschrift mit dem RSA durchführen.
- Ich kann das Protokoll einer blinden Signatur mit RSA durchführen.

Agenda

- Elliptische Kurven EC
  - Einleitungen
  - Was sind Elliptische Kurven?
  - Berechnungen (Punktverdoppelung, Punktaddition)
  - Berechnung aller Punkte
  - Der "double and add" Algorithmus
  - Der Zusammenhang mit der Gruppentheorie
  - Sicherheit von EC
  - Schlüsselverteilung mit EC (Diffie-Hellman)
  - Verschlüsselung mit EC (nach Volker-Müller)
- Weiteres zu Signaturen
  - Weitere RSA Spezialitäten
  - Blinde Signaturen
- Kurze Laufzeitbetrachtung RSA – ECC
- Lösungen der Aufgaben
- Danksagung

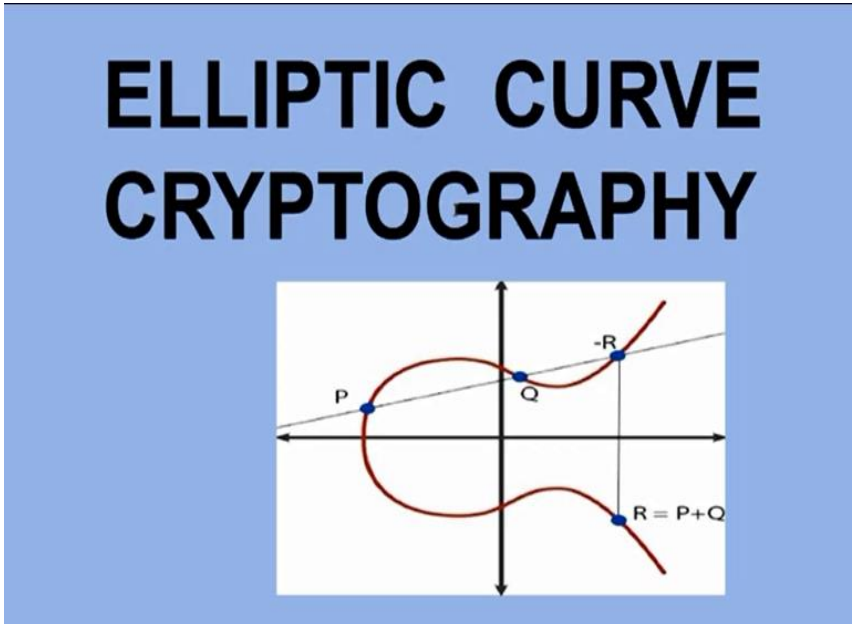
Die Slogans zu dieser Präsenz

Zu ECC: Wie bitte? Wie soll Verschlüsseln mit einer Kurve gehen?

Zu RSA ff.: Was ist der Nutzen einer blinden Unterschrift?

Teil 1

Elliptische Kurven Kryptographie



https://www.youtube.com/watch?v=igmCqyjiS_U

Kap. 1

EINLEITUNG

Einleitung

- Wir behandeln elliptische Kurven.
  - Was sind Elliptische Kurven?
  - Warum Elliptische Kurven?
  - Grundaufgaben zu den Elliptischen Kurven.
- Terminologie
  - EC = Elliptische Kurven = Elliptic Curve
  - ECC = Elliptic Curve Cryptography
  - ECCS = Elliptic Curve Cryptography Systems

- Anwendungen der ECC wie bei RSA, nämlich
  - Digital Signature (wird in diesem Modul nicht weiter besprochen!)
  - Key Exchange
  - (Hybrid) Encryption

Was sind ECC? Warum ECC?

- Was sind nun ECC?
  - In aller Kürze gesagt: EC (Elliptic Curve oder Elliptische Kurven) sind mathematische Objekte, die man als Public Key Algorithmen (ECC) verwenden kann.

- Warum ECC?
  - Performance
    - Kürzere Schlüssellängen
    - Schnellere Operation, d.h. kürzere Rechenzeiten, Grund: Es werden Multiplikationen statt Exponentiationen verwendet.
    - ***Kurzum:*** Wenige und schnelle Operationen, statt viele und langsame wie beim RSA.
  - Weniger Speicherplatz (z.B. für die Signatur)
    - Ist z.B. bei Chipkarten ein wichtiges Kriterium.
  - ECC sind „sicher“ → siehe aber auch Kap. 2.3.5 und 2.3.6 im JS Skript "ECCS".
  - Es gibt Standards (inkl. standardisierte Kurven)
  - Patentfreie Algorithmen

Kap. 2

THEORIE DER ELLIPTISCHEN KURVEN

Kap. 2.1 Um was geht es?
&
Kap. 2.2 Mathematische Grundlagen

Was sind Elliptische Kurven?

Allgemeine Form:

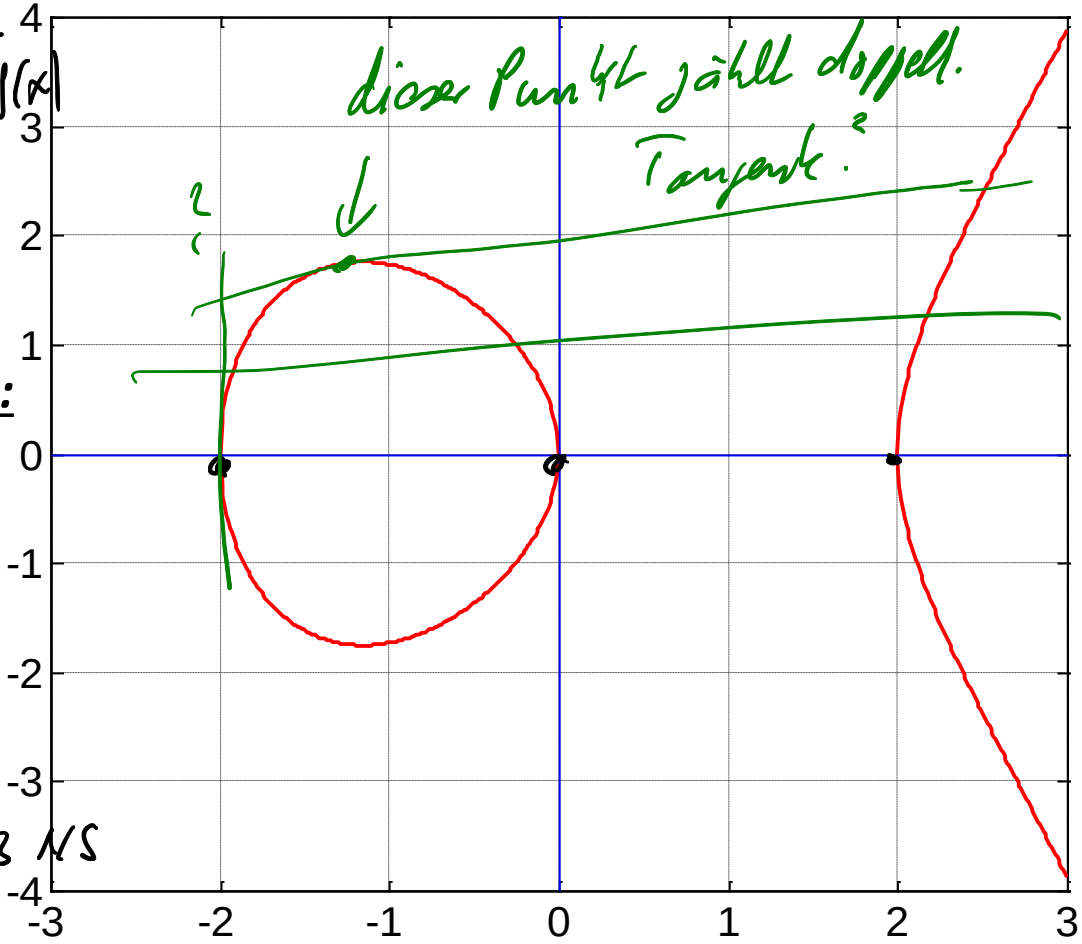
$$y^2 = x^3 + ax + b$$

Nichtsingularitätsbedingung:

$$4a^3 + 27b^2 \neq 0 \ (*)$$

Beispiel:

$$y^2 = x^3 - 4x = x(x^2 - 4)$$
$$= x(x-2)(x+2)$$



Die Bedingung (*) muss erfüllt sein, ansonsten gibt es mehrfache Wurzeln (Lösungen).
In diesem Fall wäre die Eindeutigkeit nicht mehr gegeben.
Die allg. Form zusammen mit (*) garantiert, dass die Verbindung zwischen zwei Punkten genau einen dritten Punkt hat. Bei Tangenten wird der Berührpunkt doppelt gezählt.

Vergleich zur Kreisgleichung $x^2 + y^2 = r^2$

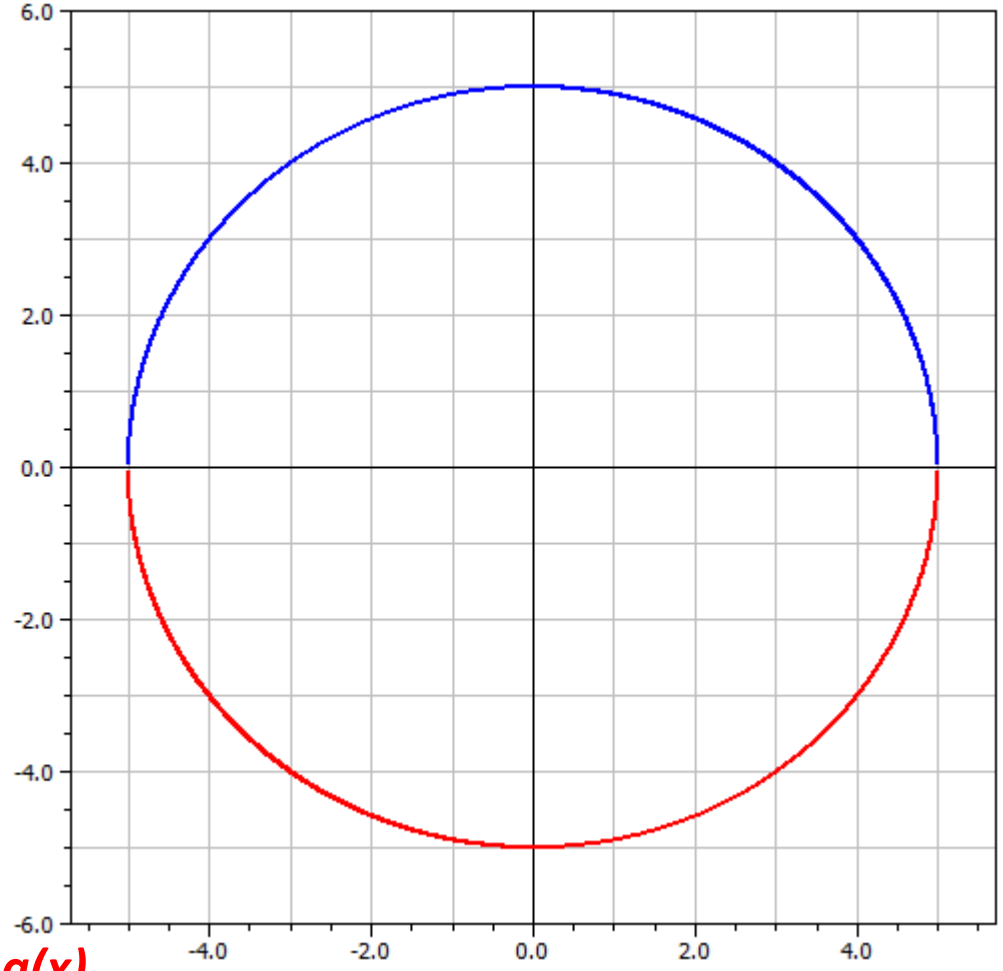
Gleichung des Kreises

mit Radius r und

Mittelpunkt (0; 0) ist

nicht von der Form y = f(x)

$$y^2 = r^2 - x^2$$



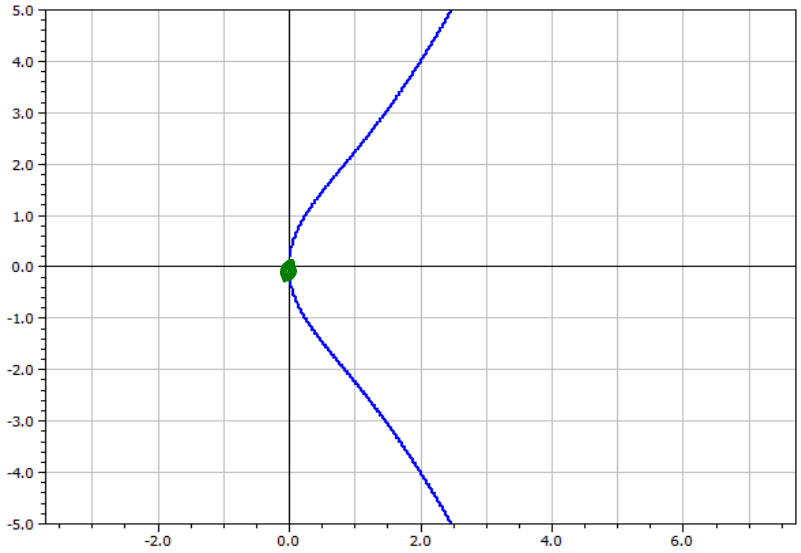
Der obere Halbkreis, y = f(x)

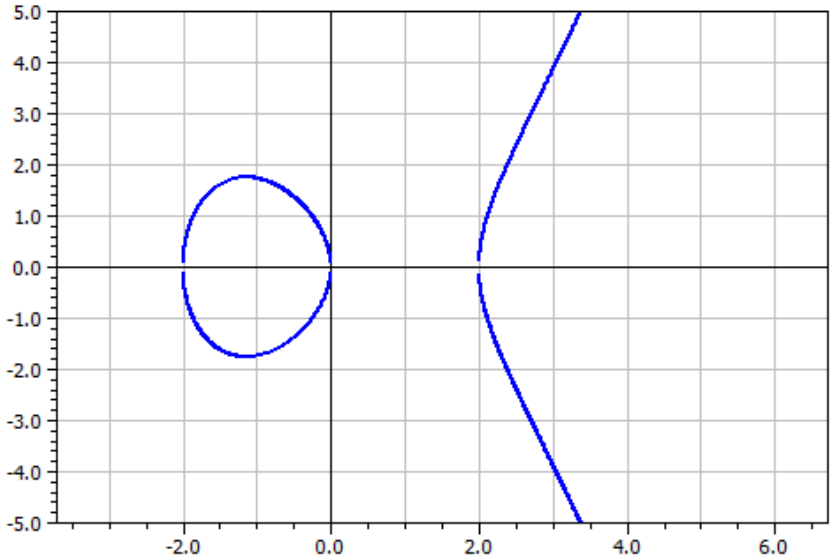
$$y = f(x) = \sqrt{r^2 - x^2}$$

Der untere Halbkreis, y = -f(x) = g(x)

$$y = g(x) = -\sqrt{r^2 - x^2}$$

2 nicht singuläre Kurven, d.h. mit $4a^3 + 27b^2 \neq 0$





$$y^2 = x^3 + 4x = x(x^2 + 4)$$

x=0 ist NS

$$y^2 = x^3 - 4x = x(x^2 - 4)$$
$$= x(x-2)(x+2)$$

Berechnung von $4a^3 + 27b^2$

$$y^2 = x^3 + 4x \Rightarrow 4a^3 + 27b^2 = 256 \neq 0$$

$a=4$ & $b=0$

$y^2 = x^3 + 0x + b$

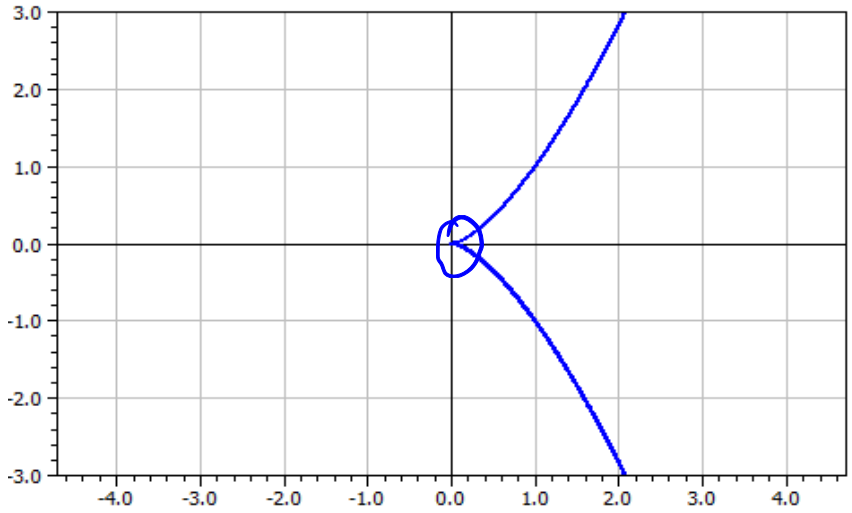
Aufgabe 1 Berechnen Sie analog für $y^2 = x^3 - 4x$

a = 4 b = 0

a = -4 b = 0

für 18446

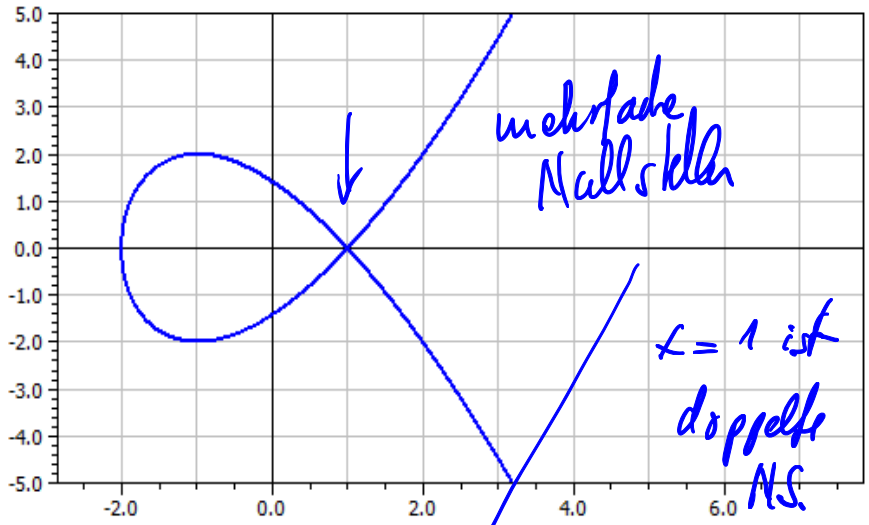
2 singuläre Kurven, d.h. mit $4a^3 + 27b^2 = 0$



$$y^2 = x^3$$

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

$a = 0 \quad b = 0$



$$y^2 = x^3 - 3x + 2$$
$$= (x-1)^2(x+2)$$

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

$a = -3 \quad b = 2$

Berechnung von $4a^3 + 27b^2$

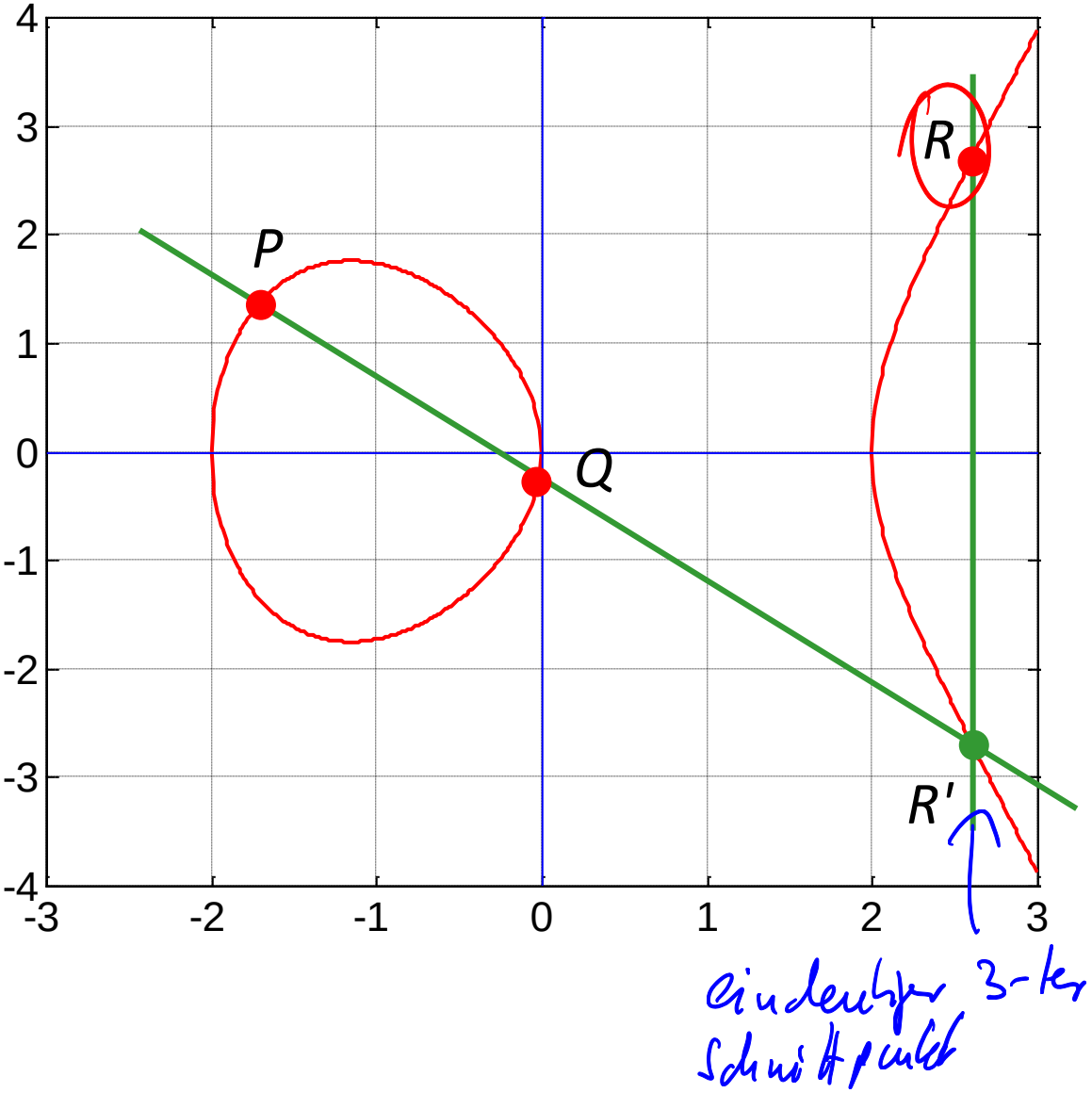
$$y^2 = x^3 \Rightarrow 4a^3 + 27b^2 = 0 + 0 = 0$$
$$y^2 = x^3 - 3x + 2 \Rightarrow 4a^3 + 27b^2 = -108 + 108 = 0$$

Kurve enthält Spitze oder doppelte Nullstellen → Solche Kurven heissen singulär und dürfen nicht gebraucht werden. Daher ist die Bedingung zu überprüfen.

Addition von zwei Punkten graphisch dargestellt

Zwei Punkte P und Q der Kurve werden «addiert». Das Ergebnis ist wieder ein eindeutiger Punkt der Kurve.

$$R = P + Q$$

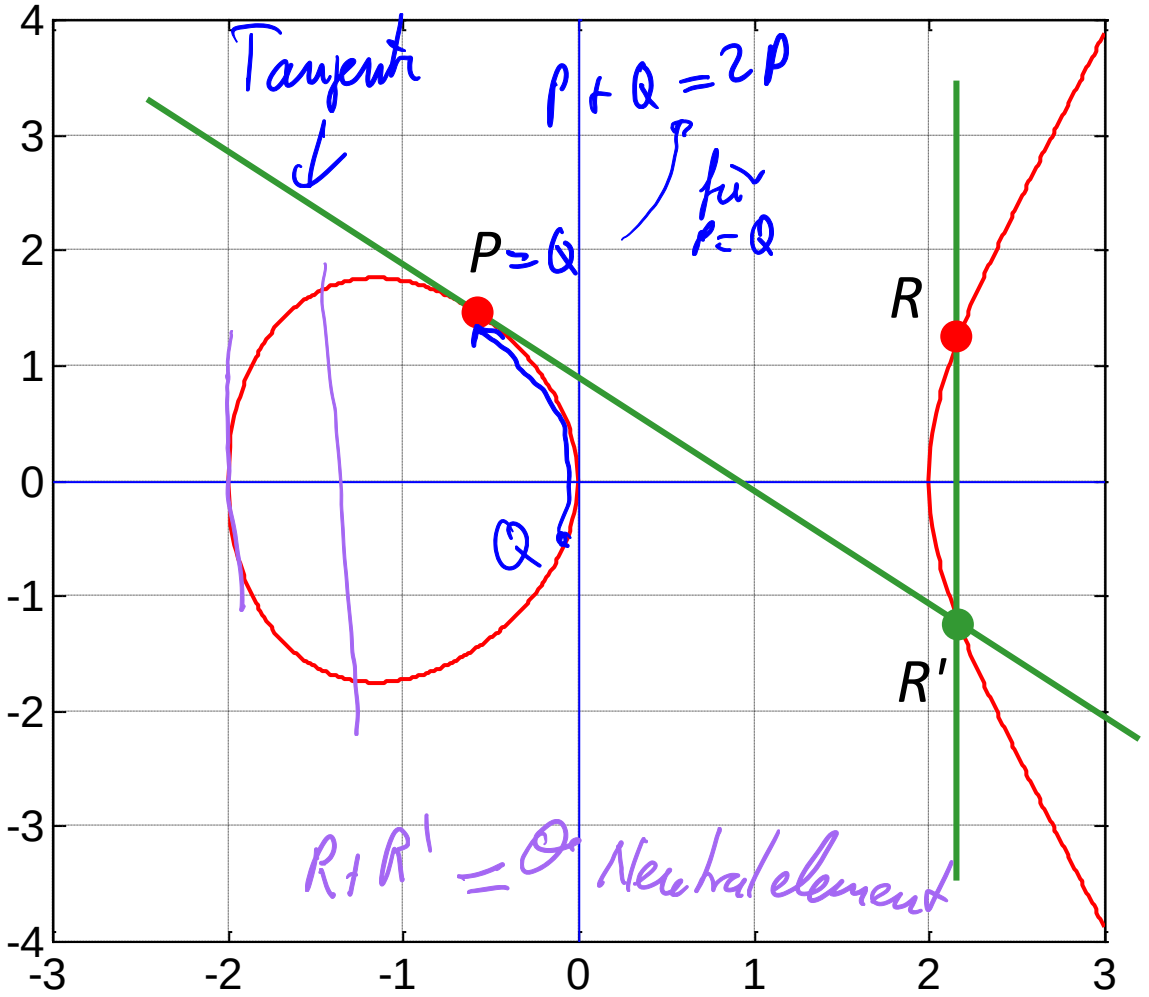


Verdoppelung Punkt graphisch dargestellt

Punkt Q geht zum Punkt P

Für P = Q gibt es eine Punktverdoppelung.
Das Ergebnis ist wieder ein eindeutiger Punkt der Kurve.

R = P + P = 2P



Die zentrale Eigenschaft bei der Addition

Die zentrale Eigenschaft bei der oben definierten Addition von zwei Punkten P + Q = R, resp. P + P = 2P = R ist, dass es ***genau einen*** (dritten) Punkt R gibt. D.h. es gibt genau einen dritten Punkt R, so dass R = P + Q, resp. R = 2P = P + P gilt. D.h. im Konkreten:

- Geht eine Gerade durch die Punkte P & Q gibt es genau einen dritten Schnittpunkt R' und damit genau einen gespiegelten Punkt R (***Grund:*** wegen dem y^2 ist die Kurve an der x-Achse spiegelbildlich. Resp. wenn wir nachher mit **mod p** reduzieren, werden wir sehen, dass die Gerade $y = \frac{p}{2}$ die Symmetrieachse sein wird.)
- Dasselbe gilt für die Tangente durch den Punkt P (bei P + P = 2P wird ja die Tangente gezeichnet). Der Berührpunkt P zählt einfach doppelt → in diesem Sinne gilt es genau wie oben bei R = P + Q.
- Es bleibt noch offen, was bei senkrechten Geraden passiert. Also R' + R = ? → dort ist ja die Verbindung eine senkrechte Gerade. Die analoge Frage stellt sich bei den Tangenten an den Nullstellen (an den Nullstellen sind die Tangenten ja senkrechte Geraden).
- ***Antwort:*** Wir müssen einen neuen Punkt $\mathcal{O}$ – der unendlich ferne Punkt – dazu nehmen → siehe dazu die nächste Folie.

Neutralelement der Addition

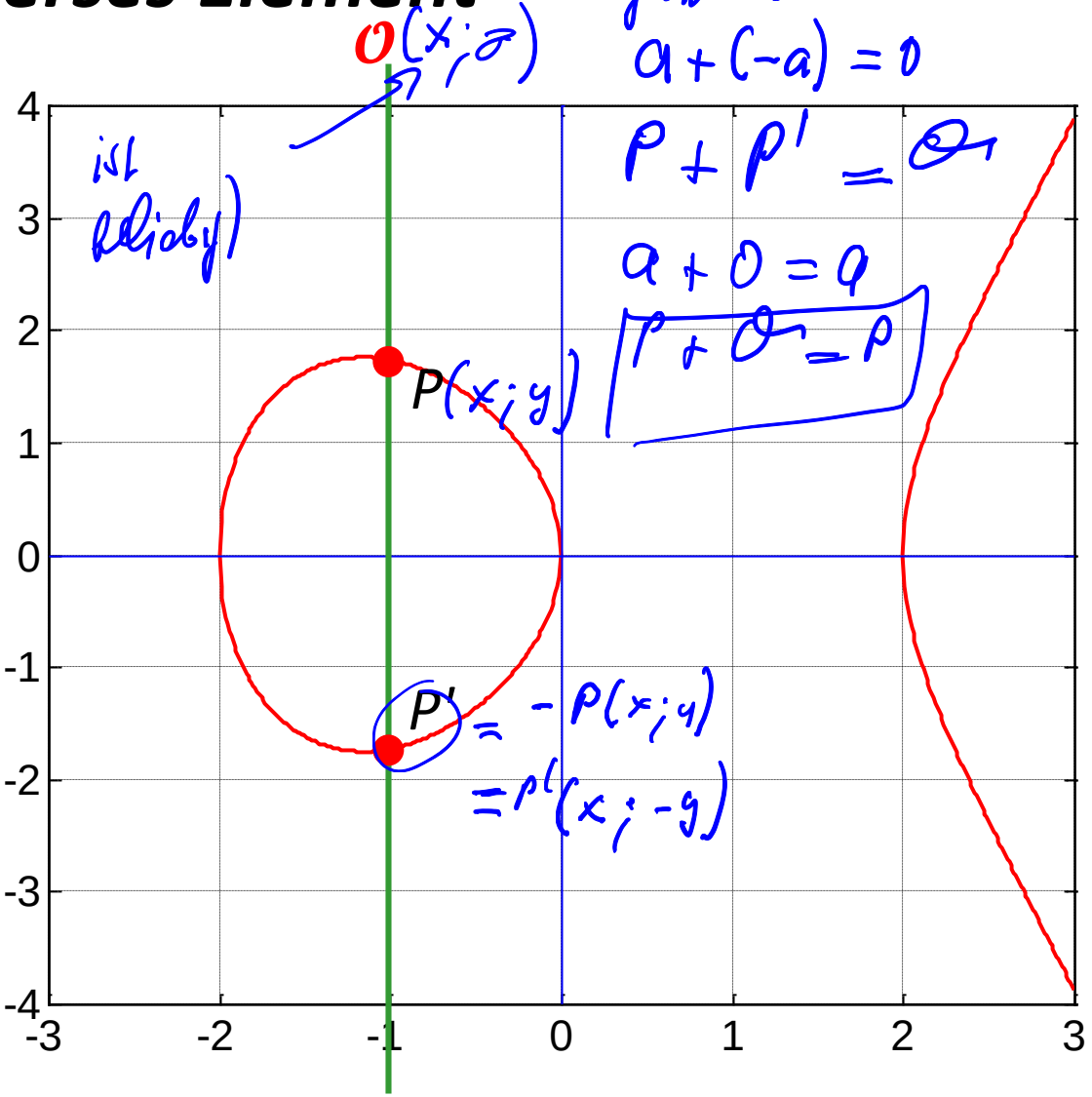
Neutrales und Inverses Element

Inverses Element:
P'(x; -y) = -P(x; y)
Spiegelung an der x-Achse

Punktaddition mit dem Inversen Element:
P + P' = P' + P = $\mathcal{O}$
Ergibt das Neutrale Element
$\mathcal{O}(x; \infty)$ = der Punkt im Unendlichen

Für das Neutralelement gilt:
P + $\mathcal{O}$ = $\mathcal{O}$ + P = P

Bemerkung 1: Das geht auch wunderbar mit der graphischen Vorstellung auf: Die Gerade durch P und $\mathcal{O}$ hat in P' den dritten Schnittpunkt. Dieser wird nun an der x-Achse gespiegelt und es resultiert P.



Bemerkung 2: Nun sieht man auch, warum die Spiegelung wichtig ist. Würde man nur den dritten Schnittpunkt nehmen, dann wäre P + $\mathcal{O}$ = P' und damit wäre $\mathcal{O}$ nicht mehr das Neutralelement, resp. es gäbe kein solches Element und somit wäre die Bedingung (3) in Folie 24 nicht erfüllt.

*k*P = Addition des Punktes (k – 1) mal mit sich*

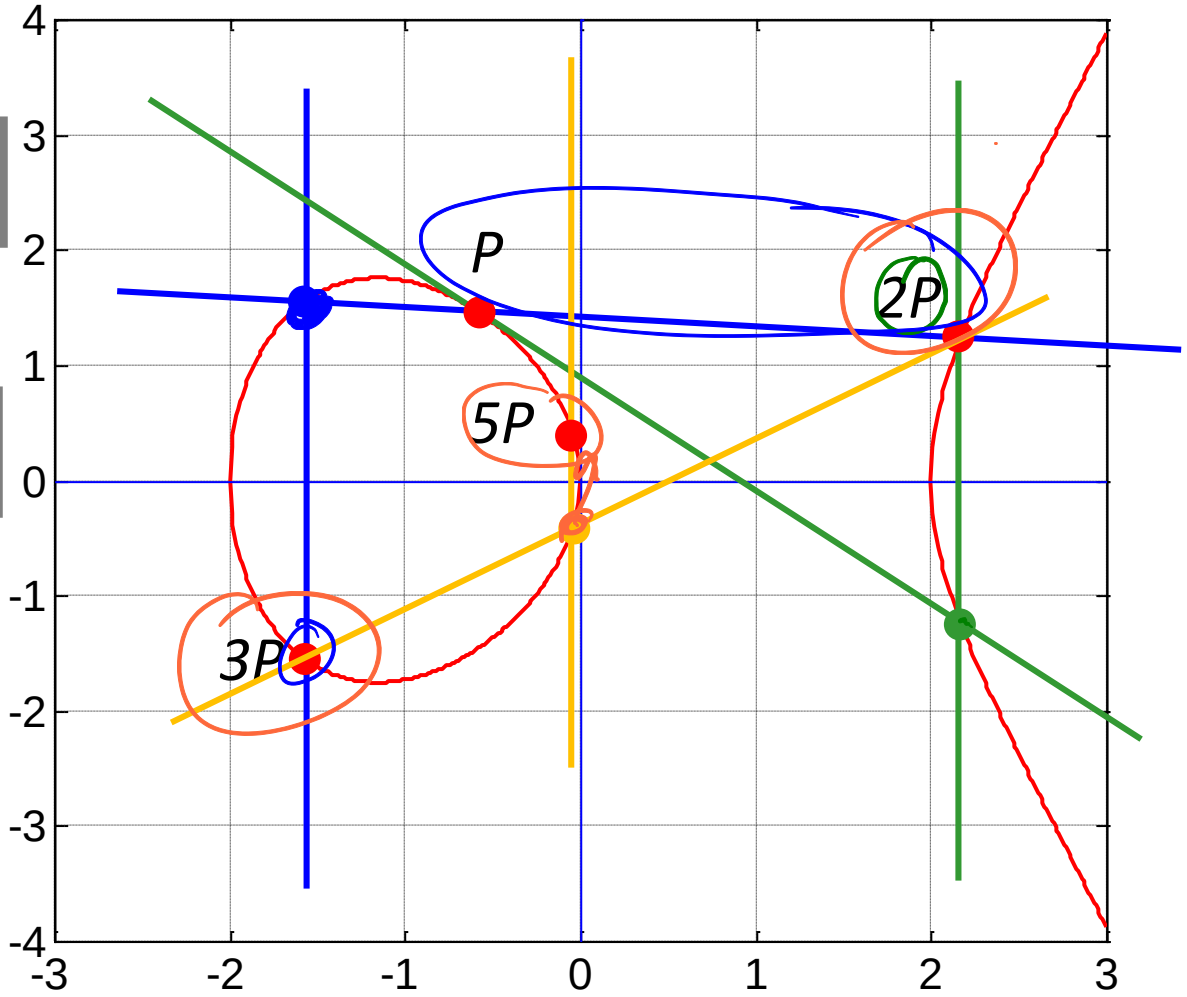
Punkt Iteration:

$$kP = P + P + \ldots + P$$

$$P + P = 2P$$

$$P + 2P + 3P$$

$$3P + 2P = 5P$$



Nun das Ganze mit mod p, p eine Primzahl

Allgemeine Form:

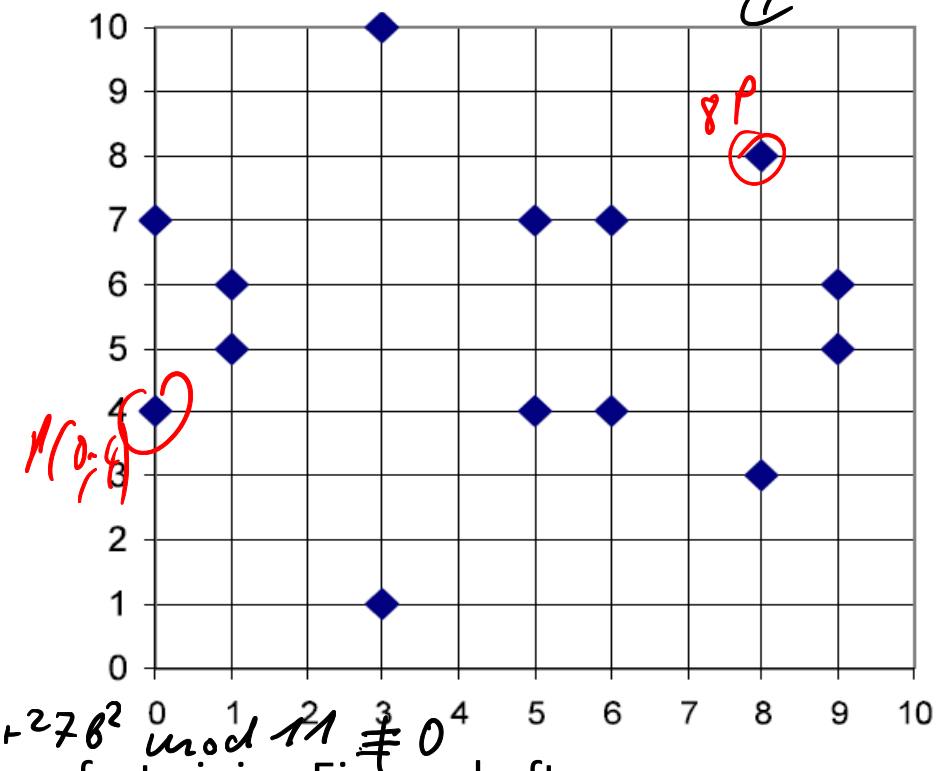
$$y^2 \equiv x^3 + ax + b \bmod p$$

Nichtsingularitätsbedingung:

$$4a^3 + 27b^2 \not\equiv 0 \bmod p$$

Beispiel:

$$y^2 \equiv x^3 + 8x + 5 \bmod 11$$



Aus dem Diagramm sehen wir sofort einige Eigenschaften:

- Die Gerade y = 5,5 ist Symmetrieachse.
- Jede vertikale Gerade hat genau 2 Punkte (gilt immer, ausser für die Punkte auf der x-Achse → in diesem Bsp. hat es keine P. auf der x-Achse).
- Horizontale Geraden haben 1, 2 oder 3 Schnittpunkte → Details betrachten wir in den Beispielen.

$y^2 \equiv x^3 + 8x + 5 \bmod 11$; Punkte und erste Berechnungen

Ohne z.Z. auf die Details einzugehen, erhalten wir alle Punkte der Kurve indem wir k*P mit P(0; 4) für k = 1, 2, … durchrechnen. Ein Punkt P, mit dem wir alle Punkte erzeugen können, nennen wir Basispunkt. In diesem Falle hat die «Kurve» – mit dem unendlich fernen Punkt, resp. der Null $\mathcal{O}$ – insgesamt 15 Punkte. Das sehen wir auch im Bild. Im Weiteren sehen wir – siehe auch die Tabelle – dass 16*P = P, 17*P = 2P usw. ist.

k	1	2	3	4	5	6	7	8
k*P	(0; 4)	(1; 6)	(3; 1)	(9; 5)	(5; 4)	(6; 7)	(8; 3)	(8; 8)

k	9	10	11	12	13	14	15	16
k*P	(6; 4)	(5; 7)	(9; 6)	(3; 10)	(1; 5)	(0; 7)	$\mathcal{O}$	(0; 4)

k	17	18	19	20	21	22	23	24
k*P	(1; 6)	(3; 1)	(9; 5)	(5; 4)	(6; 7)	(8; 3)	(8; 8)	(6; 4)

Wichtig:

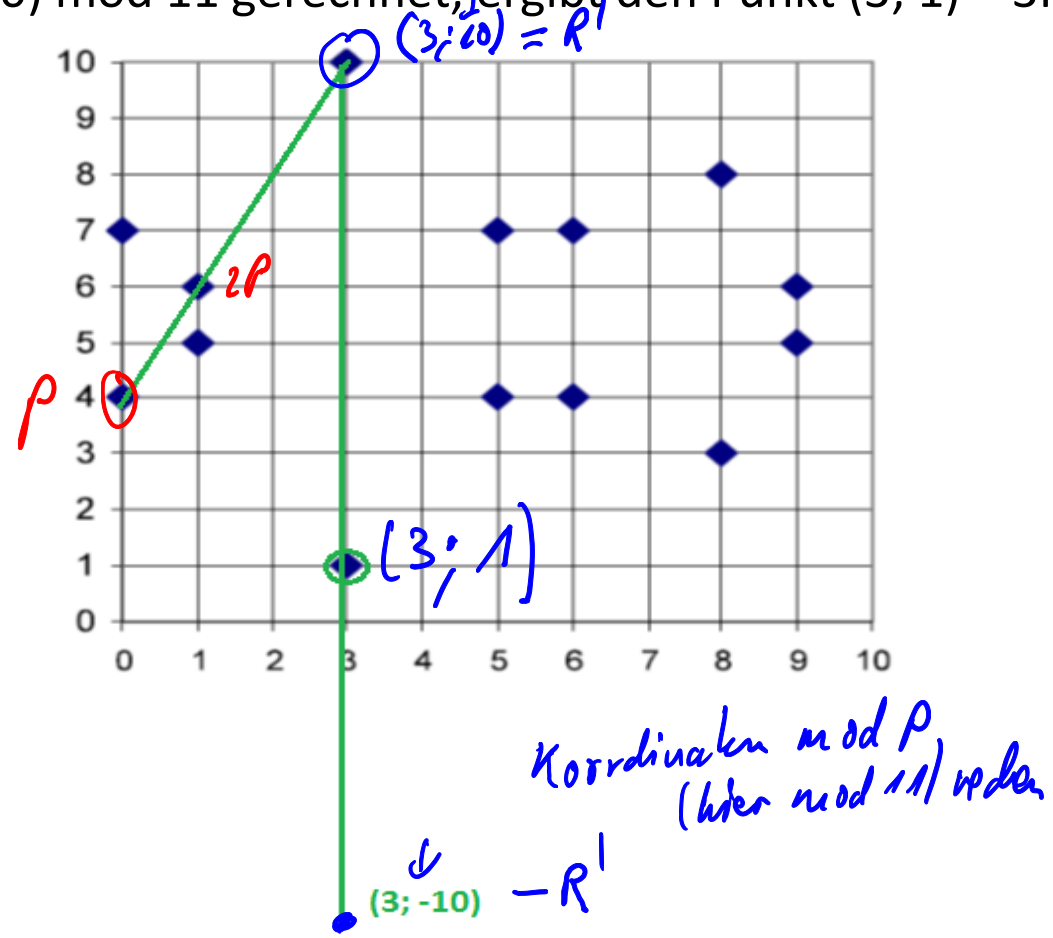
I) Die Koordinaten muss man mod p (hier p = 11) rechnen.

II) Das k bei $k \cdot P$ muss mod Gruppenordung (hier 15) reduziert werden.

$y^2 \equiv x^3 + 8x + 5 \bmod 11$; Punkte und erste Berechnungen

Beispiel 1: P + 2P = 3P, D.h. (0; 4) + (1; 6) = (3; 1)

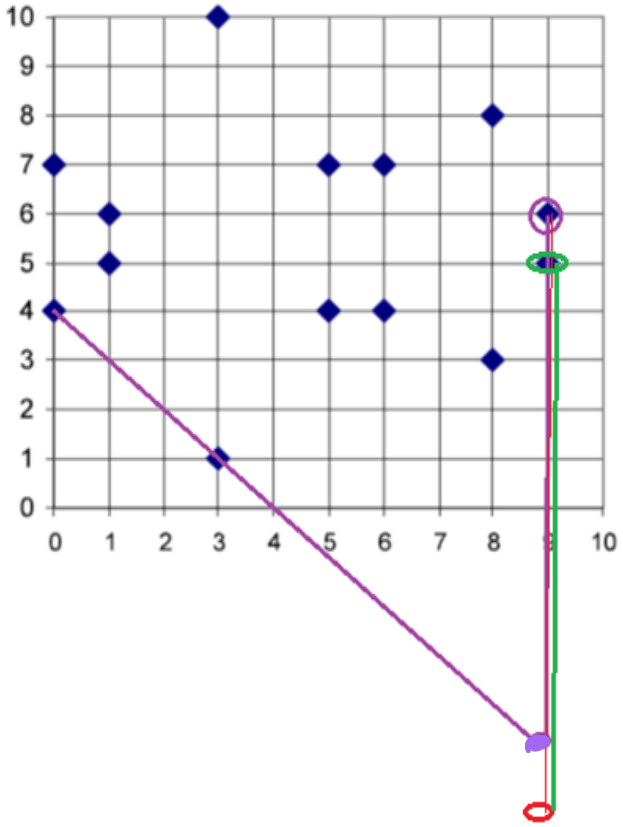
- Durch Verbinden der beiden Punkte kommt man auf den Punkt (3; 10)
- Spiegeln des Punktes (3; 10) ergibt (3; -10)
- Die Koord. des Punktes (3; -10) mod 11 gerechnet, ergibt den Punkt (3; 1) = 3P



$y^2 \equiv x^3 + 8x + 5 \bmod 11$; Punkte und erste Berechnungen

Beispiel 2: P + 3P = 4P, D.h. (0; 4) + (3; 1) = (9; 5)

- Durch Verbinden der beiden Punkte kommt man auf den Punkt (9; -5) = (9; 6)
- Spiegeln des Punktes (9; 6) ergibt (9; -6) = (9; 5) da mit mod 11 gerechnet wird.
- Punkt (9; 5) = 4P



(9; 6)

(9; 5) = P(9; -6) aber mit mod 11 gerechnet = (0; 4) + (3; 1)

(9; -5) = (9; 6) wegen mod 11

(9; -6) = gespiegelter P(9; 6)

Siehe Beispiele 2.5 & 2.6: Im Skript "ECCS", p. 14 ff.

Erste Zusammenfassung

Allgemeine Form:

$$E = \{(x; y) \mid y^2 \equiv x^3 + ax + b \bmod p\} \cup \{\mathcal{O}\}$$

Nichtsingularitätsbedingung:

$$4a^3 + 27b^2 \not\equiv 0 \bmod p$$

Gruppenstruktur: Die Punkte mit der P.addition bilden eine abelsche Gruppe

(1) Abgeschlossenheit: $P + Q = S \in E$

(2) Assoziativgesetz AG: $P + (Q + R) = (P + Q) + R$

(3) Neutralelement $\mathcal{O}$: $P + \mathcal{O} = \mathcal{O} + P = P$

(4) Inverses Element in E: $P + (-P) = (-P) + P = \mathcal{O}$

(5) Kommutativgesetz KG: $P + Q = Q + P$

Wahl der Parameter:

- Die Grösse der Primzahl p muss je nach Sicherheitslevel zw. 256 und 512 Bit sein.
- Die Parameter a und b können in einem gewissen Rahmen frei gewählt werden, sie müssen die Nichtsingularitätsbedingung erfüllen.
- Weitere Bedingungen und Bemerkungen siehe Skript "ECCS", Kap. 2.2.

Nicht.sing. Bed & Anzahl Punkte einer Kurve

Beispiel: $y^2 \equiv x^3 + 8x + 5 \bmod 11$ $a=8$ $b=5$

Nichtsingularitätsbedingung:

$$4a^3 + 27b^2 \not\equiv 0 \bmod p$$

Beispiel: $4 \cdot 8^3 + 27 \cdot 5^2 \equiv 2723 \equiv 6 \not\equiv 0 \bmod 11$

 # Punkte = Theorem von Hasse: (inkl $\mathcal{O}$)

$$p + 1 - 2\sqrt{p} \leq |E| \leq p + 1 + 2\sqrt{p}$$

hängt von der Primzahl p ab

Beispiel: $11 + 1 - 2\sqrt{11} \leq |E| \leq 11 + 1 + 2\sqrt{11}$ $|E| = p+1 \pm 2\sqrt{p}$

Also: $5 \leq |E| \leq 19$ **Erfüllt, denn:** $|E| = 15$

Bemerkung:

Die genaue Anzahl der Punkte wird dann durch die Wahl der Parameter a und b bestimmt, mit $mod\ p$ ist „nur“ der mögliche Range der Anzahl Punkte bestimmt.

Reale Anzahl bei Primzahl p mit 512 Bit:

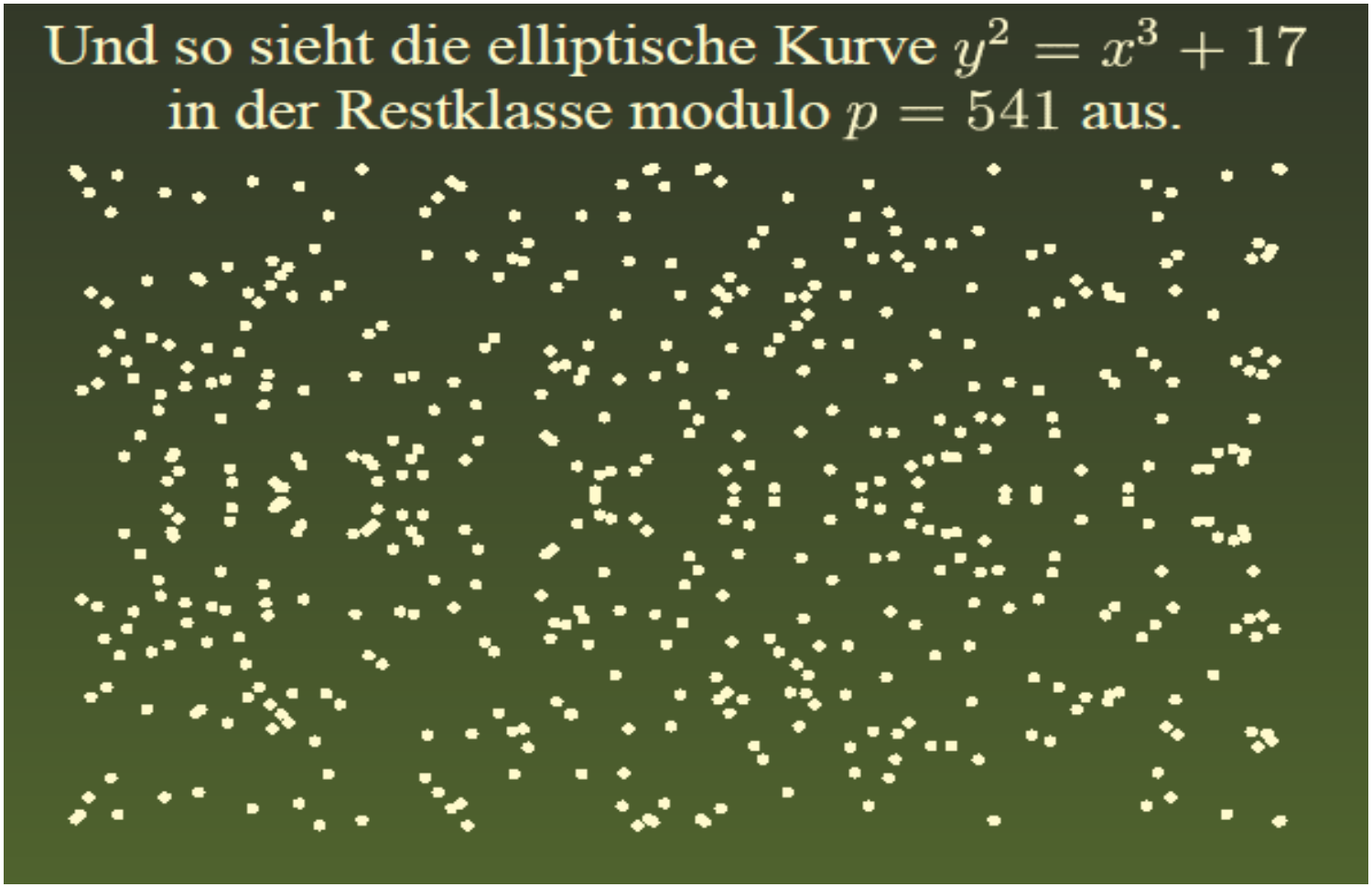
$p \approx 2^{512} \rightarrow |E| \approx 2^{512} \pm 2^{257} \approx 2^{512}$

$p+1 \pm 2\sqrt{p}$ $2^{512} \pm 2 \cdot \sqrt{2^{512}} = 2^{512} \pm 2 \cdot 2^{256} = 2^{512} \pm 2^{257}$

Vergleich: # Atome im Weltall $\approx 10^{77} \approx 10^{3*26} = (10^3)^{26} \approx (2^{10})^{26} \approx 2^{260}$

Wobei $2^{512} \approx$ Anzahl Atome, wenn auf jedem Atom im Weltall ein Weltall läge.

$y^2 \equiv x^3 + 17 \ mod \ 541$



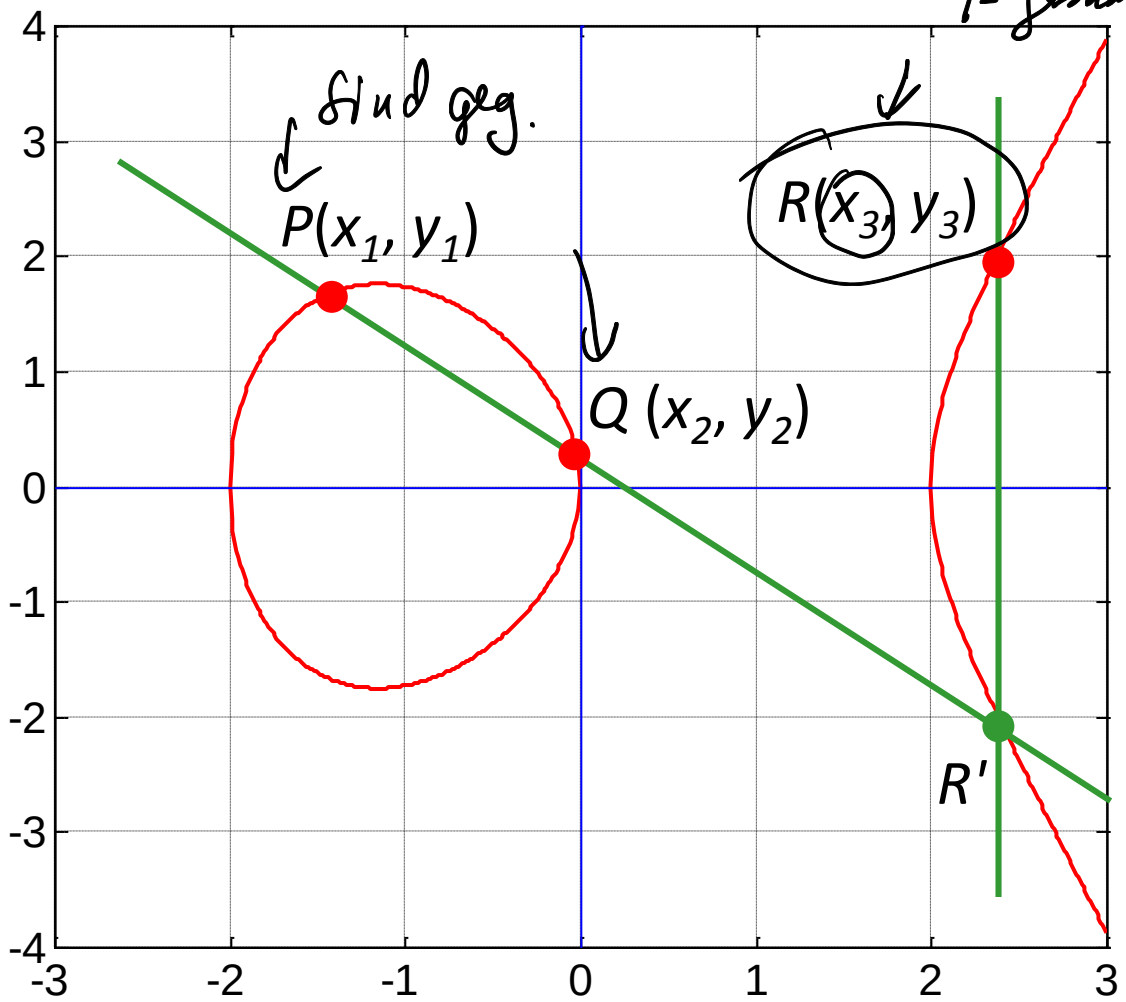
Aufgabe 2

a) Checken Sie die Nichtsingularitätsbedingung.

b) Im welchem Bereich liegen die Anzahl der Punkte?

bis 19432

Formeln zur Addition von zwei Punkten R = P + Q



Steigung der Geraden

$$s \equiv \frac{y_2 - y_1}{x_2 - x_1} \; mod \; p$$

$$x_3 \equiv s^2 - x_1 - x_2 \; mod \; p$$

$$y_3 \equiv s(x_1 - x_3) - \; y_1 \; mod \; p$$

Präsenzbeispiel zur Addition von zwei Punkten R = S + T

$y^2 \equiv x^3 + 8x + 5 \bmod 11$ gemäss Tabelle S(6; 7) und T(8; 3)

6P + 7P = 13P ≡ (1; 5)

Erwartetes Resultat:

$$s \equiv \frac{y_2 - y_1}{x_2 - x_1} \bmod p \equiv \frac{3 - 7}{8 - 6} \bmod 11 \equiv \frac{-4}{2} \equiv (-4) \cdot (2^{-1}) \equiv 7 \cdot 6 \equiv 42 \equiv 9 \bmod 11$$

$2^{-1} \bmod 11 = 6$, da $2 \cdot 6 \equiv 12 \equiv 1 \bmod 11$

$$x_3 \equiv s^2 - x_1 - x_2 \bmod p \equiv 9^2 - 6 - 8 \equiv 67 \equiv 1 \bmod 11$$

$$y_3 \equiv s(x_1 - x_3) - y_1 \bmod p \equiv 9(6 - 1) - 7 \equiv 45 - 7 \equiv 38 \equiv 5 \bmod 11$$

Resultat: R = S + T = (1; 5)

Beispiel zur Addition von zwei Punkten R = P + Q

$y^2 \equiv x^3 + 8x + 5 \bmod 11$ *P(0; 4) und Q(3; 1)*

Erwartetes Resultat: *Da Q = 3P, ist R = P + Q = P + 3P = 4P = (9; 5)*

$$s \equiv \frac{y_2 - y_1}{x_2 - x_1} \bmod p \equiv \frac{1-4}{3-0} \bmod 11 \equiv \frac{-3}{3} \bmod 11 \equiv (-3) \cdot 3^{-1} \bmod 11 \ (*)$$

$$\equiv [(-3) \bmod 11 \cdot 3^{-1} \bmod 11] \bmod 11 \equiv 8 \cdot 4 \bmod 11 \equiv 32 \bmod 11 = 10$$

$$x_3 \equiv s^2 - x_1 - x_2 \bmod p \equiv 10^2 - 0 - 3 \bmod 11 \equiv 97 \bmod 11 = 9$$

$$y_3 \equiv s(x_1 - x_3) - y_1 \bmod p \equiv 10 \cdot (0 - 9) - 4 \bmod 11 \equiv -94 \bmod 11 = 5$$

Resultat: *R = P + Q = (9; 5)*

(*) Kürzen wäre hier möglich -1 mod 11 = 10, aber !!!!! beim Kürzen

Aufgabe 3 *a) Berechnen Sie nun Q + P.*

b) Wählen Sie andere Punkte aus der Tabelle.

Bis 13h50

Paul bis 20h00

Formeln zur Verdoppelung R = 2P

$y^2 \equiv x^3 + ax + b \mod p$

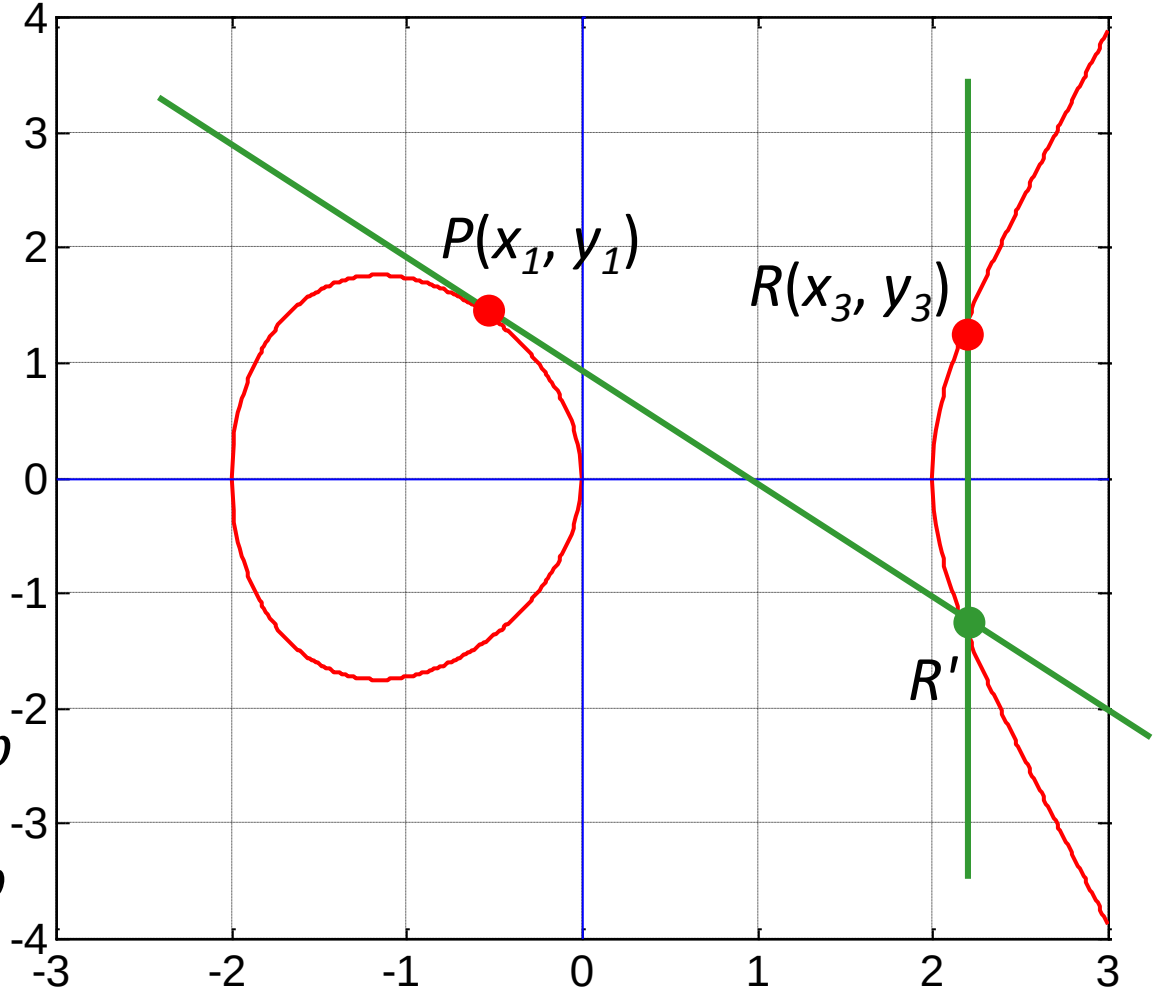
Steigung wird neu berechnet

$$s \equiv \frac{3 \cdot x_1^2 + a}{2 \cdot y_1} \; mod\; p$$

$$x_3 \equiv s^2 - x_1 - x_2 \; mod\; p$$

$$\equiv s^2 - x_1 - x_1 \; mod\; p$$

$$y_3 \equiv s(x_1 - x_3) - y_1 \; mod\; p$$



Präsenzbeispiel zur Verdoppelung U = 2R

Tabelle = 13P

$y^2 \equiv x^3 + 8x + 5 \bmod 11$ (9)

mod 15

R(1; 5) (= 13P)

Erwartetes Resultat: 2P = 26P = 11P = (9; 6) (**)

$$s \equiv \frac{3 \cdot x_1^2 + a}{2 \cdot y_1} \bmod p \equiv \frac{3 \cdot 1^2 + 8}{2 \cdot 5} \equiv \frac{11}{10} \equiv \frac{0}{10} \bmod 11 = 0$$

$(10^{-1} \bmod 11 \equiv 10)$

$$x_3 \equiv s^2 - x_1 - x_1 \bmod p \equiv 0^2 - 1 - 1 \equiv -2 \equiv 9 \bmod 11$$

$$y_3 \equiv s(x_1 - x_3) - y_1 \bmod p \equiv 0(1 - 9) - 5 \equiv -5 \equiv 6 \bmod 11$$

Bis 20h15

Resultat: U = 2R = (9; 6)

(**) Für k·P muss k mod 15 genommen werden (Gruppenordg).

Beispiel zur Verdoppelung R = 2P resp. 2Q

$y^2 \equiv x^3 + 8x + 5 \bmod 11$ *Q(6; 4) (= 9P)*

Erwartetes Resultat: *Siehe Tabelle: 2Q = 18P = 3P = (3; 1) (**)*

$$s \equiv \frac{3 \cdot x_1^2 + a}{2 \cdot y_1} mod\ p \equiv \frac{3 \cdot 6^2 + 8}{2 \cdot 4} mod\ 11 \equiv \frac{116}{8}\ mod\ 11 \equiv 116 \cdot 8^{-1}\ mod\ 11$$

$$\equiv [116\ mod\ 11 \cdot 8^{-1}\ mod\ 11]\ mod\ 11 \equiv 6\ \cdot 7\ mod\ 11 \equiv 42\ mod\ 11 = 9$$

$$x_3 \equiv s^2 - x_1 - x_1\ mod\ p \equiv 9^2 - 6 - 6\ mod\ 11 \equiv 69\ mod\ 11 = 3$$

$$y_3 \equiv s(x_1 - x_3) - y_1\ mod\ p \equiv 9 \cdot (6 - 3) - 4\ mod\ 11 \equiv 23\ mod\ 11 = 1$$

Resultat: *R = 2Q = (3; 1)*

(**) Für t*P muss t mod 15 genommen werden, darum ist 18P = 3P.

Aufgabe 4 *Verdoppeln Sie andere Punkte aus der Tabelle.*

Wie merkt man, dass $\mathcal{O}$ herauskommen sollte?

Das ist eine ganz zentral Frage. Betrachten wir unsere Beispielkurve, so sehen wir, dass wir bei $r \cdot P + s \cdot P = (r + s) \cdot P = 15 \cdot P$ automatisch $\mathcal{O}$ erhalten. Wenn r + s = 15 (allgemein: r + s = die Anzahl der Punkte resp. r + s = Ordnung der Punkte), dann liegen sie auf der gleichen senkrechten Geraden. Damit haben sie die gleiche x-Koordinate. Wenn diese Punkte nun addiert werden, so muss $\mathcal{O}$ resultieren.

Fazit: Bei Division durch Null → wir haben $\mathcal{O}$ erhalten.

In Kap. 5.1.4 im JS Skript „ECCS“ hat es entsprechende Beispiele.

Und an dieser Stelle noch weitere Besonderheiten → siehe auch Kap. 5.1 im JS Skript „ECCS“.

- Addieren wir zwei Punkte mit gleichen x-Koordinaten (siehe oben), dann ist die Summe der y-Koordinaten dieser zwei Punkte gleich p, d.h. es gilt dann: $y_1 + y_2 \equiv p \equiv 0 \; mod \; p$.
- Wenn eine Kurve zwei Punkte auf der x-Achse hat, dann muss sie notwendigerweise noch einen dritten haben. Denn zwei solche Punkte darf man ja addieren.
- Verdoppelt man einen Punkt P, der auf der x-Achse liegt, dann resultiert der unendlich ferne Punkt $\mathcal{O}$, also $2P = \mathcal{O}$.

Der Double and Add Algorithmus (daa-Algorithmus)

Analog dem „square and multiply“ (sam) kann mit dem „double and add“ (daa)-Algorithmus effizient gerechnet werden.

Algorithmus:

- Dezimalzahl in Bitdarstellung schreiben.
- Mit der ersten „1“ macht man nichts, d.h. P → P
- Nach der ersten Eins macht man von links nach rechts nun Folgendes:
  - Wenn eine „0“ kommt, dann verdoppelt man, also n*P → 2n*P
  - Bei einer „1“ wird verdoppelt und danach mit P addiert, d.h. n*P → 2n*P → 2nP + P = (2n+1)*P.

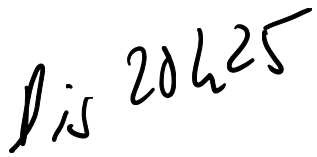
Präsenzbeispiel: 28*P

- $28 = 16+8+4 = 2^4+2^3+2^2 = (11100)_2$
- Mit der ersten „1“ macht man nichts, d.h. $P \Rightarrow P$
- Nach der ersten Eins macht man von links nach rechts nun Folgendes:
  - Wegen 1 $P \Rightarrow 2P \Rightarrow 2P+P = 3P$ DA
  - Wegen 1 $3P \Rightarrow 6P \Rightarrow 6P+P = 7P$ DA
  - Wegen 0 $7P \Rightarrow 14P$ D
  - Wegen 0 $14P \Rightarrow 28P$ D

Der Double and Add Algorithmus, Beispiel

Beispiel: 25*P

- $25 = 16 + 8 + 1 = 2^4 + 2^3 + 2^0 = (11001)_2$
- Mit der ersten „1“ macht man nichts, d.h. P → P
- Nach der ersten Eins macht man von links nach rechts nun Folgendes:
  - Wegen 1, P → 2P und 2P → P + 2P = 3P
  - Wegen 0, 3P → 6P
  - Wegen 0, 6P → 12P
  - Wegen 1, 12P → 24P und P + 24P → 25P



Tipps zur Prüfung

In der Prüfung werden i.d.R. die folgenden 2 Grundaufgaben separat geprüft.

Typ 1:

Gegeben eine Dezimalzahl, gesucht der Ablauf des daa-Algorithmus. Cf. Beispiel oben.

Typ 2:

Gegeben die Parameter einer EC, sowie die Koordinaten von R = 4*P & S = 7*P.
Gesucht: T = 15*P.
Also: T = 15*P = 2*R + S = 2*(4*P) + 7*P → konkret die Rechenop. durchführen.

Kap. 2.4 Berechnen aller Punkte
&
Kap. 2.5 Zusammenhang mit der Gruppentheorie

Präsenzbeispiel mit $y^2 \equiv x^3 + 3x + 2 \bmod 7$

A) Bestimmung aller Punkte

x	0	1	2	3	4	5	6
$x^3 + 3x + 2 \bmod 7$	2	5	2	3	1	2	5

y	0	1	2	3	4	5	6
$y^2 \bmod 7$	0	1	4	2	2	4	1

3 kommt nicht vor

Für x = 0: gibt es (0; 3) und (0; 4)

Für x = 1: gibt es keine Punkte da $y^2 \bmod 7 \neq 5$

Für x = 2: gibt (2; 3) und (2; 4)

Für x = 3: gibt es keine Punkte

Für x = 4: gibt es die Punkte (4; 1) und (4; 6)

Für x = 5: gibt es die Punkte (5; 3) und (5; 4)

Für x = 6: gibt es keine Punkte

Somit gibt es 8 Punkte + „Nullpunkt“ $\mathcal{O}$ = 9 Punkte. Damit ist die Gruppenordnung (= Anzahl Elemente resp. Punkte) = 9.

B) Allg. Zusammenhang mit der Gruppentheorie

Definitionen:

(1) Die Ordnung d. Kurve = Gruppenordnung = Anz. Punkte der Kurve inkl. $\mathcal{O}$.
(2) Die Ordnung eines Punktes P ist das kleinste k, so dass $k \cdot P = \mathcal{O}$
(3) Eine Gruppe heisst ***zyklisch***, wenn ein Punkt P mit $k \cdot P$ alle weiteren Punkte erzeugt. Ein solcher Punkt heisst ***erzeugend*** oder ***primitiv***.

Sätze:

(1) Die Ordnung eines Punktes P muss die Gruppenordnung teilen.
(2) Ist die Gruppenord. prim $\Leftrightarrow$ alle Elemente – ausser $\mathcal{O}$ – sind erzeugend.
(3) Ist die Gruppenord. nicht prim, so kann es erzeug. Eleme. haben o. nicht.
(4) Wenn es erzeugende El. hat, dann hat es genau $\varphi(Grupp.ord)$ erz. El.
(5) Wenn es erzeugende El. hat, dann hat es genau $\varphi(Elementeord)$ Elemente von jeder möglichen Ordnung, die ein Element haben kann.
(6) Die Ordnung der Untergruppen muss auch die Gruppenordnung teilen.
(7) Jede Gruppe G hat G (also sich selber) und $\{e\}$ hier $= \{\mathcal{O}\}$ als sogenannte triviale Untergruppe.

Für unsere Elliptischen Kurven gibt es 3 Varianten (cf. Skript "ECCS", Kap. 2.5):

a) Die Gruppenordnung ist eine Primzahl (cf. Beispiel 2.9) → Gr. ist autom. zyklisch.
b) Die Gruppenord. ist nicht prim und es hat erzeug. Elemente (cf. Beispiel 2.8).
c) Die G.ord. ist nicht prim und es hat keine erzeug. Elemente (cf. Beispiel 2.10).

Präsenzbeispiel mit $y^2 \equiv x^3 + 3x + 2 \bmod 7$

C) Zusammenhang mit der Gruppentheorie

Für unsere Elliptischen Kurven gibt es 3 Varianten:

a) Die Gruppenordnung ist eine Primzahl (cf. Beispiel 2.9).

b) Die Gruppenord. ist nicht prim, es hat erzeug. Elemente (cf. Beispiel 2.8).

c) Die G.ord. ist nicht prim, es hat keine erzeug. Elemente (cf. Beispiel 2.10).

⇒ alle Elemente außer $\mathcal{O}$ sind erzeugend

Die Gruppenordnung ist 9, somit können nur die Fälle b) & c) vorkommen.

Präsenzbeispiel: Wir bestimmen die Ordnung des Punktes P(0; 3)

Lösung:

Da die Gruppenordnung 9 ist, kann ein Element nur die Ordnung 1 (nur $\mathcal{O}$ kann die Ord. 1 haben), 3 oder 9 haben.

- Mit einer Punktverdopplung (mit den Formeln nachrechnen) hat man: 2P = (2; 3).
- Mit Punktadd. (selber rechnen) hat man: 3P = P + 2P = (0; 3) + (2; 3) = (5; 4) $\neq \mathcal{O}$.
- Also, (0; 3) hat nicht die Ordnung 3 und damit muss (0; 3) die Ordnung 9 haben.
- Damit hat die Gruppe:
  - $\varphi(9) = \varphi(3^2) = 3^2 - 3^1 = 6$ erzeugende Elemente (also mit Ord 9).
  - $\varphi(3) = 2$ Elemente mit Ordnung 3.
  - 1 Element (die $\mathcal{O}$) mit Ordnung 1.
  - Total sind es 9 Elemente.

Präsenzbeispiel mit $y^2 \equiv x^3 + 3x + 2 \bmod 7$

D) Für alle P das k*P und alle Untergruppen

k	1	2	3	4	5	6	7	8	9
k*(0; 3)	(0; 3)	(2; 3)	(5; 4)	(4; 6)	(4; 1)	(5; 3)	(2; 4)	(0; 4)	$\mathcal{O}$

k	1	2	3	4	5	6	7	8	9
k*(0; 4)	(0; 4)	(2; 4)	(5; 3)	(4; 1)	(4; 6)	(5; 4)	(2; 3)	(0; 3)	$\mathcal{O}$

k	1	2	3	4	5	6	7	8	9
k*(2; 3)	(2; 3)	(4; 6)	(5; 3)	(0; 4)	(0; 3)	(5; 4)	(4; 1)	(2; 4)	$\mathcal{O}$

k	1	2	3	4	5	6	7	8	9
k*(2; 4)	(2; 4)	(4; 1)	(5; 4)	(0; 3)	(0; 4)	(5; 3)	(4; 6)	(2; 3)	$\mathcal{O}$

k	1	2	3	4	5	6	7	8	9
k*(4; 1)	(4; 1)	(0; 3)	(5; 3)	(2; 3)	(2; 4)	(5; 4)	(0; 4)	(4; 6)	$\mathcal{O}$

k	1	2	3	4	5	6	7	8	9
k*(4; 6)	(4; 6)	(0; 4)	(5; 4)	(2; 4)	(2; 3)	(5; 3)	(0; 3)	(4; 1)	$\mathcal{O}$

k	1	2	3
k*(5; 3)	(5; 3)	(5; 4)	$\mathcal{O}$

k	1	2	3
k*(5; 4)	(5; 4)	(5; 3)	$\mathcal{O}$

erzeugen alle Punkte

2 Punkte haben die Ordg 3

Damit ist $\{\mathcal{O};\ (5;3);\ (5;4)\}$ die einzige nicht triviale UG; sie hat die Ord. 3.

Bemerkung: Wäre die Gruppenordnung prim, so gäbe es nur die trivialen UG.

Kap. 2.3 Sicherheit von EC

Sicherheit von ECC

- Grundlagen
  - Punkte P und Q = k*P sind bekannt
  - k ist geheim → es ist schwierig k zu finden, obwohl P und Q bekannt sind (→ das diskrete Log. Problem für Elliptische Kurven).
- Public Key
  - Kurvengleichung $y^2 = x^3 + ax + b \; mod \; p$, die Parameter a und b sowie der Modulus p (p prim).
  - Punkte P und Q = k*P (k ist geheim)
- Secret Key
  - k = Anzahl Iterationen

darf man Standards nehm.

→ obwohl die Kurve sowie P und Q bekannt sind kann das "k" nicht berechnet werden!

- Es gibt aber auch unsichere Spezialfälle → cf. Kap. 2.3.5 im JS Skript "ECCS".
- Es gibt auch kritische Stimmen zur Sicherheit von ECC
  - Dies insbesondere auch, weil man standardisierte Kurven benutzen kann. Nur viele davon sind von NIST definiert worden → cf. Kap. 2.3.6 im JS Skript "ECCS".

Warum ECC, wenn es den RSA gibt?

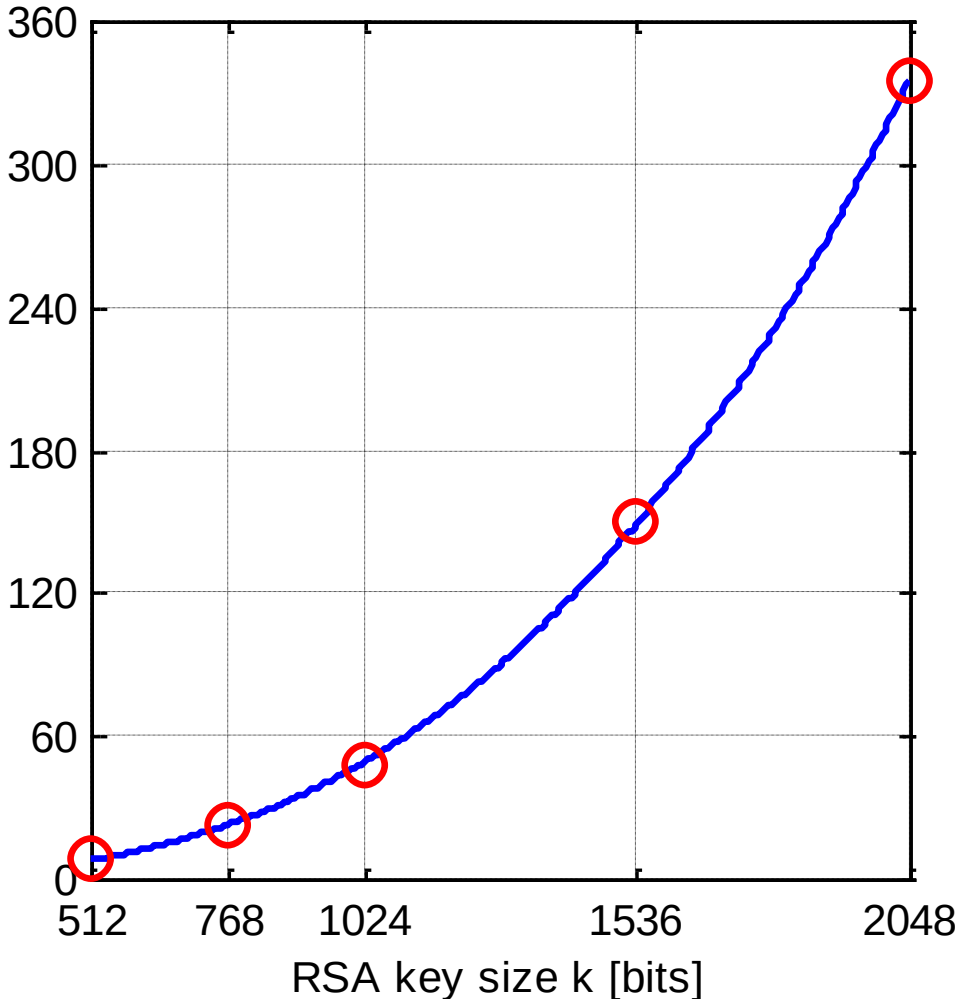
Die Gründe können wie folgt zusammengefasst werden:

- Das ECDLP Problem ist schwerer zu lösen als das Faktorisierungsproblem und das DLP Problem in $\mathbb{Z}_p^*$, deshalb braucht es viel kleinere Schlüsselgrössen.
- Wegen den kleineren Schlüsselgrössen sind auch die Signaturen viel kleiner, was in Chipkarten von grossem Vorteil ist.
- Die Berechnungen in ECC sind viel effizienter (wenige schnelle Multiplikationen statt viele langsame Exponentiationen).
- Pro Kurve gibt es viele Gruppen, die benutzt werden können. D.h. zu den Parameter a, b und p können viele Punkte P genommen werden.

Die Kryptografische Stärke im Vergleich (Quantencomputer nicht berücksichtigt!)

Symmetric	**56**	**80**	**112**	**128**	**192**	**256**
RSA n	512	1024	2048	3072	7680	15360
ECC p	112	160	224	256	384	512
Key size ratio	5:1	6:1	9:1	12:1	20:1	30:1

RSA Berechnung $y = x^e \bmod n$; Exp. zunahme



RSA key size k [bits]	Processing time t [s]
512	8
768	22
1024	48
1536	150
2048	335

ECC, RSA und Quantencomputer (QC) → betrachten wir in Präs 13

- Qubits
  - Bei Quantencomputer wird mit Qubits gerechnet.
  - Für k Bits im klassischen System braucht es
    - Für die Faktorisierung von k Bits mit QC bei RSA $K \approx 2k$ Qubits
    - Für das DL-Probl. von k Bits mit QC bei EC $K \approx 5k + 8\sqrt{k} + 5log_2 k$ Qubits
  - Quantencomputer mit wie viel Qubits gibt es aktuell?
    - 2001 → 7 Qubits
    - Aktuell 512 Qubits? → Wissenschaftlich nicht bestätigt. Und noch wichtig: Aber bei -273 Grad Celsius!!
    - Forecast 2015: 2048 Qubits → wurde nicht erreicht → es gibt z.Z. keine bekannten Forecasts!!
- Beispiel
  - 3072 Bit RSA ist im Rahmen der klassischen Methoden ungefähr gleich stark wie 256 Bit ECC.
- Es braucht nun:
  - Faktorisierung von k = 3072 Bits bei RSA $K \approx 2k = 2 \cdot 3072 = 6144$ Qubits
  - DL bei ECC, k = 256 Bits $K \approx 5 \cdot 256 + 8\sqrt{256} + 5log_2 256 = 1468$ Qubits
  - Für QC bräuchte es für ECC p = 1164 Bit um so stark wie ein RSA 3072 zu sein. D.h. das Sicherheitslevel von ECC und RSA ist mit/ohne QC unterschiedlich.

Basis-Test Kap. 2, Theorie der EC

Aussage	Richtig oder falsch?	Begründung
Mit EC kann man – ähnlich zu RSA – signieren, Schlüsselaustauschen & (hybrid) verschlüsseln.	☐ richtig ☐ falsch	
Die EC werden den RSA kaum je ablösen.	☐ richtig ☐ falsch	
Bei EC wird – analog zu RSA – immer mit mod N ($N = p \cdot q$, p, q grosse Primzahlen – gerechnet.	☐ richtig ☐ falsch	
Bei EC wird – im Gegensatz zu RSA – addiert und verdoppelt, anstatt multipliziert und quadriert.	☐ richtig ☐ falsch	
Es können beliebige EC genommen werden.	☐ richtig ☐ falsch	
Bei zwei gegebenen Punkten, z.B. P(2; 4), Q(3; 5) können für P + Q einfach die jeweiligen Koordinaten zusammengezählt werden, also: P(2; 4) + Q(3; 5) = S(5; 9).	☐ richtig ☐ falsch	
Die Anzahl der Punkte können mit einer Formel abgeschätzt werden.	☐ richtig ☐ falsch	
Die Anzahl der Punkte einer EC können die Anzahl der Atome im Weltall weit übertreffen.	☐ richtig ☐ falsch	
Die Sicherheit von EC basiert auf einer Einwegfunktion mit Trapdoor.	☐ richtig ☐ falsch	
Die Sicherheit von EC liegt im diskreten Logarithmen Problem für EC.	☐ richtig ☐ falsch	

Basis-Test Kap. 2, Theorie der EC, Fortsetzung

Aussage	Richtig oder falsch?	Begründung
Die Sicherheit von EC liegt, dass bei gegebenen Punkten P, Q mit $Q = i \cdot P$ das „i“ nicht berechnet werden kann.	☐ richtig ☐ falsch	
$Q = i \cdot P$ wird – analog zum RSA mit dem SaM – mit „double and add“ gerechnet.	☐ richtig ☐ falsch	
Der Schnittpunkt des Koordinatensystems ist der Punkt (0; 0)	☐ richtig ☐ falsch	
Der Punkt (0; 0) kann nie Punkt einer Elliptischen Kurve sein.	☐ richtig ☐ falsch	
Der Punkt (0; 0) ist das Nullelement der abelschen Gruppe mit der Punkteaddition.	☐ richtig ☐ falsch	
Alle Punkte der EC bilden eine abelsche Gruppe, sie hat aber kein Nullelement.	☐ richtig ☐ falsch	
Alle Punkte der EC bilden mit der Punktaddition eine abelsche Gruppe, das Nullelement ist der unendlich ferne Punkt $\mathcal{O}$.	☐ richtig ☐ falsch	
Alle Punkte der EC bilden mit der Punktaddition eine abelsche Gruppe, die Gruppenordnung ist immer p, weil mod p gerechnet werden muss.	☐ richtig ☐ falsch	
Es gibt immer $\varphi(p) = p - 1$ primitive Elemente.	☐ richtig ☐ falsch	
Die Anzahl der Punkte der Kurve ist auch gleichzeitig die Ordnung der Gruppe.	☐ richtig ☐ falsch	

Basis-Test Kap. 2, Theorie der EC, Fortsetzung

Aussage	Richtig oder falsch?	Begründung
Die Gruppe kann auch Untergruppen enthalten.	☐ richtig ☐ falsch	
Für EC werden nur Gruppen genommen, die zyklisch sind.	☐ richtig ☐ falsch	
Für EC werden nur Gruppen genommen, deren Ordnung prim ist.	☐ richtig ☐ falsch	
Im Rahmen der klassischen Kryptoanalyse braucht es für EC eine deutlich kleinere Schlüsselgrösse als bei RSA, um die gleiche Sicherheit zu erreichen.	☐ richtig ☐ falsch	
Beim Einsatz von Quantencomputer QC braucht es viel weniger Qubits um einen RSA zu knacken, als beim EC.	☐ richtig ☐ falsch	

Kap. 3

KRYPTOGRAPHISCHE ANWENDUNGEN MIT EC

Kap. 3.1 Einleitung

Kap. 3.2

Übersicht akt. Schlüsselgrössen

Als Nachbearbeitung sind auf Präsenz 10 die Kap. 3.1 & 3.2 im JS Skript "ECCS" durchzuarbeiten, sowie die Aufgaben in Kap. 3.3 zu lösen.

Kap. 3.3 Disk. Log. bei ECC

Disk. Log. Problem anhand von $y^2 \equiv x^3 + 8x + 5 \bmod 11$

Cf. Diffie-Hellman, dort besteht das mathematisch schwer zu berechnende Problem, dass die Gleichung $z \equiv g^{x} mod\ p$ nicht auf x lösbar ist, obwohl z, g und p bekannt sind. Dies unter der Bedingung, dass p eine mindestens 600-stellige Primzahl, g ein sogenannter Generator (einer genügend grossen zyklischen Untergruppe) und x zufällig gewählt ist.
Bei EC ist das Lösen der Gleichung $Q = k \cdot P$ ebenfalls schwer lösbar, obwohl die Kurve (mit den Parametern a, b und p) sowie die Punkte P und Q gegeben sind.

k	1	2	3	4	5	6	7	8
k*P	(0; 4)	(1; 6)	(3; 1)	(9; 5)	(5; 4)	(6; 7)	(8; 3)	(8; 8)

k	9	10	11	12	13	14	15	16
k*P	(6; 4)	(5; 7)	(9; 6)	(3; 10)	(1; 5)	(0; 7)	$\mathcal{O}$	(0; 4)

k	17	18	19	20	21	22	23	24
k*P	(1; 6)	(3; 1)	(9; 5)	(5; 4)	(6; 7)	(8; 3)	(8; 8)	(6; 4)

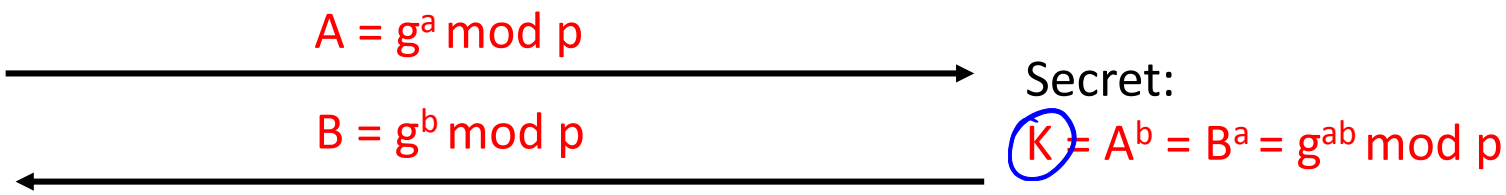
Bei kleinen Werten ist das natürlich kein Problem.
Gegeben: a = 8, b = 5, p = 11, P = (0; 4), dann ist Q(8; 3) $= 7 \cdot P = 7 \cdot (0; 4)$.
Wichtig: das k wird immer mod Gruppenordnung (hier = 15) genommen.

Kap. 3.4

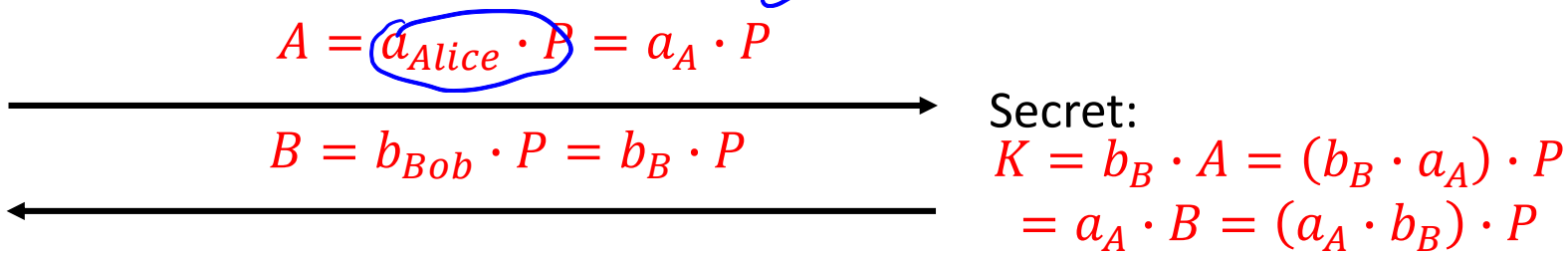
DHEC Diffie-Hellman mit EC

Krypto Anwendung – Secret Key Exchange

- Diffie-Hellman in $(\mathbb{Z}_p^*; \odot)$: Basis g (= Generator der Gruppe, genauer Generator einer zyklischen Untergruppe von Primordnung) und p prim. Der Generator g und die Primzahl p sind öffentlich bekannt.



- Elliptic Curve (a, b, p) und Basispunkt P (= Generator der additiven Gruppe, genauer Generator einer zyklischen Gruppe oder Untergruppe von Primordnung). Generator P, die Primzahl p und die Kurvenparameter a und b sind öffentlich bekannt. ***Wichtig:*** Im Gegensatz zum klassischen DH, gibt es beim ECC-DH Gruppen mit Primordnung; man ist in diesem Fall nicht auf Untergruppen angewiesen!



Untervarianten zum EC-DH und EC-MQV: Cf. Skript "ECCS", Kap. 3.4.2 ff. Nicht Prüfungsstoff!!

Präsenzbsp. EC-DH $y^2 \equiv x^3 + 8x + 5 \; mod \; 11$ mit P(0; 4)

Also öffentlich bekannt sind: a = 8, b = 5, p = 11 und P(0; 4).
Damit wir einfach rechnen können, beachten Sie die Punktetabelle weiter vorne. Für den letzten Schritt bitte im Skript "ECCS" Anhang 2 Bsp. 2.8 Fortsetzung nachschauen.

Alice	unsichere Leitung	Bob
Erzeugt Zufallszahl a = 14, rechnet & schickt A = aP = 14*(0; 4) = (0; 7)		
	A = aP = (0; 7) ---------->	
		Erzeugt b = 13, K = 13*(0; 7) = (1; 6) er rechnet & schickt B = 13*(0; 4) = (1; 5)
	B = bP = (1; 5) <----------	
a(bP) = 14*(1; 5) = (1; 6) = K		

Kontrolle: K = $14 \cdot (13(0;4)) \equiv (14 \cdot 13) P = 182 P = 2P = 2(0;4) = (1;6)$ mod 45

Wichtig: Bitte die Parameter a = 8 & b = 5 der Kurve ***nicht*** mit den zufällig gewählten Werten a = 14 (von Alice) und b = 13 (von Bob) vermischen!!

Achtung: Man kann nur die x-Koordinate von K als Secret Key nehmen! Cf. Kap. 3.4.1.
Weitere Bemerkungen: Im Skript "ECCS", Kap. 3.4.1.

Beispiel EC-DH $y^2 \equiv x^3 + 8x + 5 \; mod \; 11$ mit P(8; 3)

Also öffentlich bekannt sind: a = 8, b = 5, p = 11 und P(8; 3).
Damit wir einfach rechnen können, beachten Sie den Anhang 2, Bsp. 2.8 im Skript "ECCS".

Alice	unsichere Leitung	Bob
Erzeugt Zufallszahl a = 4, rechnet & schickt A = aP = 4*(8; 3) = (1; 5)		
	A = aP = (1; 5) ---------->	
		Erzeugt b = 2, K = 2*(1; 5) = (9; 6) (#) er rechnet & schickt B = 2*(8; 3) = (0; 7)
	B = bP = (0; 7) <----------	
a(bP) = 4*(0; 7) = (9; 6) = K (##)		

(#) Die Berechnung von 2*(1; 5) = (9; 6) kann man in Anhang 2, Beispiel 8.1 nachschauen oder direkt berechnen.
(##) Die Berechnung von 4*(0; 7) = (9; 6) kann man in Anhang 2, Beispiel 8.1 nachschauen oder direkt berechnen.

Kontrolle: Insbesondere gilt 4*(0; 7) = 4*(2*(8; 3)) = 4*2*(8; 3) = 8*(8; 3) = (9; 6).

Achtung: Man kann nur die x-Koordinate von K als Secret Key nehmen! Cf. Kap. 3.4.1.
Weitere Bemerkungen: Im Skript "ECCS", Kap. 3.4.1.

Präs 10

Kap. 3.5

Verschlüsselung mit EC nach VM

Verschlüsselung mit EC nach Volker Müller [VM98]

A. Schlüsselgenerierung (Empfänger Bob)
   a. Wähle eine Elliptische Kurve E mit
      i. Primzahl p mit ca. 256, 384 oder 512 Bit (d.h. ca. 77-, 115- oder 154-stellige Dezimalzahl).
      ii. Koeffizienten a und b
      iii. Einen Punkt P, der eine zyklische Untergruppe der Primordnung q generiert. Die Ordnung q ist also ein grosser Primteiler der Gruppenordnung, die mit dem Theorem von Hasse abgegrenzt ist. Im Wesentlichen verlangt man, dass q ebenfalls ca. 256, 384 resp. 512 Bit gross ist. Oder anders gesagt, der Kofaktor h = |E|/q < 10.
   b. Wähle zufällig d so, dass 0 < d < q gilt, wobei d auch ca. die Grösse von q hat (256, 384 o. 512 Bit).
   c. Berechne $Q = d \cdot P$

   Damit haben wir die Schlüssel erzeugt:
   K_{pub} = (p, a, b, q, P, Q)
   K_{pr} = (d)

B. Verschlüsselung (Sender Alice)
   a. Wähle i mit 1 < i < q – 1.
   b. Berechne Einmal-Key K_E mit $K_E = i \cdot P$
   c. Berechne Masking-Key K_M mit $K_M = i \cdot Q$
   d. Verschlüssle die Meldung T mit $Y = T \oplus xKoord\ von\ K_M$
   Verschickt wird die verschlüsselte Meldung Y mit dem Einmal-Key, also (Y, K_E)

C. Entschlüsseln (Empfänger Bob)
   a. Berechne Masking-Key K_M mit $K_M = d \cdot K_E$
   b. Entschlüssle die Meldung Y mit $T = Y \oplus xKoord\ von\ K_M$

Beweis der Richtigkeit erfolgt durch „simples“ Ausrechnen:

$$K_M = d \cdot K_E = d \cdot (i \cdot P) = i \cdot (d \cdot P) = i \cdot Q = K_M$$

Präsenzbeispiel einer Verschlüsselung mit EC n. [VM98]

Es seien $\boldsymbol{E}$: $\boldsymbol{y^2 \equiv x^3 + x + 6}$ über GF(11), d.h. p = 11 und P(3; 6) der Basispunkt.

Abmachung:

In einer Aufgabenstellung im Rahmen der EC sind die folg. Ausdrücke synonym.

- "mod p"; "in $\mathbb{Z}_p$"; "in $(\mathbb{Z}_p, +)$"; "über $\mathbb{Z}_p$"; "über GF(p)".
- Dabei bedeutet GF = Galois Field, oder endliche Körper. Ein endlicher Körper ist eine algebraische Struktur, die endlich viele Elemente enthält, ansonsten aber die gleichen Eigenschaften wie die reellen Zahlen haben. → Nicht Prüfungswissen!!

A. Schlüsselgenerierung (Empfänger Bob)
   a. Wähle eine Elliptische Kurve E mit
      i. Primzahl p = 11
      ii. Koeffizienten a = 1 und b = 6
      iii. Die Ordnung von E ist 13, d.h. E ist eine zyklische Gruppe von Primordnung und damit ist jeder Punkt – ausser der Nullpunkt – ein erzeugendes Element der Gruppe. Und somit ist q = Ord G = 13; d.h. der Kofaktor h = 1. Wir wählen den Punkt P(3; 6) als Basispunkt.

   b. Wähle d = 10, somit ist 0 < d < q erfüllt.
   c. Berechne $Q =$ d * P = 10 (3;6) = (5;9) -> siehe Tabelle

   Damit haben wir die Schlüssel erzeugt:

   K_{pub} = (p, a, b, q, P, Q) = 11; 1; 6; 13; (3,6); (5,9)

   K_{pr} = (d) = 10

Präsenzbeispiel einer Verschlüsselung, Fortsetzung

Beachten Sie die Tabellen im Anhang 2, Bsp. 2.9 im Skript "ECCS".

B. Verschlüsselung (Sender Alice)
   a. Wähle i = 5, somit ist 1 < i < q – 1 erfüllt.
   b. Berechne Einmal-Key K_E mit $K_E = 5 \cdot P = 5 \cdot (3; 6) = (2; 4)$
   c. Berechne Masking-Key K_M mit $K_M = i \cdot Q = 5 \cdot (5; 9) = (8; 3)$
   d. Verschlüssle die Meldung T = 7 mit $Y = T \oplus xKoord\ von\ K_M = 7 \oplus 8 = F$

   Verschickt wird die verschlüsselte Meldung Y mit dem Einmal-Key, also (Y, K_E) = (F, (2; 4))

C. Entschlüsseln (Empfänger Bob)
   a. Berechne Masking-Key K_M mit $K_M = d \cdot K_E = 10 \cdot (2; 4) = (8; 3)$
   b. Entschlüssle die Meldung Y mit $T = Y \oplus xKoord\ von\ K_M = F \oplus 8 = 7$

Alice	unsichere Leitung	Bob
		Siehe oben A.
Erhält K_{pub} = (p, a, b, q, P, Q) = (11, 1, 6, 13, (3; 6), (5; 9))	<----------	
Verschlüsselt Meldung T = 7, siehe B		
	(Y, K_E) = (F, (2; 4)) ---------->	
		Entschlüssle Y, siehe C.

Bemerkungen:

1) P ist Teil des Public Keys und $K_E = i \cdot P$ wird über die Leitung geschickt. Wegen dem disk. Log. Problem der EC, kann Eve – trotz allen Informationen – das „i“ nicht berechnen.
2) Nur bei der XOR-Operation müsste die HEX-Darstellung einer Zahl > 9 berücksichtigt werden!

Beispiel einer Verschlüsselung mit EC nach [VM98]

Es seien $\boldsymbol{E}$: $\boldsymbol{y^2 \equiv x^3 + x + 6}$ über GF(11), d.h. p = 11 und P(7; 2) der Basispunkt.
Beachten Sie die Tabellen im Anhang 2, Bsp. 2.9 im Skript "ECCS".

A. Schlüsselgenerierung (Empfänger Bob)
   a. Wähle eine Elliptische Kurve E mit
      i. Primzahl p = 11
      ii. Koeffizienten a = 1 und b = 6
      iii. Die Ordnung von E ist 13, d.h. E ist eine zyklische Gruppe von Primordnung und damit ist jeder Punkt ausser der Nullpunkt ein erzeugendes Element der Gruppe. Und somit ist q = Ord G = 13; d.h. der Kofaktor h = 1. Wir wählen den Punkt P(7; 2) als Basispunkt.
   b. Wähle d = 8, somit ist 0 < d < q erfüllt.
   c. Berechne $Q = d \cdot P = 8 \cdot (7; 2) = (10; 2)$

   Damit haben wir die Schlüssel erzeugt:
   K_{pub} = (p, a, b, q, P, Q) = (11, 1, 6, 13, (7; 2), (10; 2))
   K_{pr} = (d) = 8

Beispiel einer Verschlüsselung, Fortsetzung

Beachten Sie die Tabellen im Anhang 2, Bsp. 2.9 im Skript "ECCS".

B. Verschlüsselung (Sender Alice)
   a. Wähle i = 3, somit ist 1 < i < q – 1 erfüllt.
   b. Berechne Einmal-Key K_E mit $K_E = i \cdot P = 3 \cdot (7; 2) = (3; 5)$
   c. Berechne Masking-Key K_M mit $K_M = i \cdot Q = 3 \cdot (10; 2) = (2; 4)$
   d. Verschlüssle die Meldung T = 5 mit $Y = T \oplus xKoord\ von\ K_M = 5 \oplus 2 = 7$

   Verschickt wird die verschlüsselte Meldung Y mit dem Einmal-Key, also (Y, K_E) = (7, (3; 5))

C. Entschlüsseln (Empfänger Bob)
   a. Berechne Masking-Key K_M mit $K_M = d \cdot K_E = 8 \cdot (3; 5) = (2; 4)$
   b. Entschlüssle die Meldung Y mit $T = Y \oplus xKoord\ von\ K_M = 7 \oplus 2 = 5$

Alice	unsichere Leitung	Bob
		Siehe oben A.
Erhält K_{pub} = (p, a, b, q, P, Q) = (11, 1, 6, 13, (7; 2), (10; 2))	<----------	
Verschlüsselt Meldung T = 5, siehe B		
	(Y, K_E) = (7, (3; 5)) ---------->	
		Entschlüssle Y, siehe C.

Weiteres zur Verschlüsselung mit EC

- Die Verschlüsselung ist also eine hybride Verschlüsselung.
- Weitere Verschlüsselungsarten (Nicht Prüfungsstoff!!):
  - Schlüsselaustausch und Verschlüsselung analog zu Elgamal, siehe [MS04].
  - Schlüsselaustausch & Verschlüsselung def. von Menezes-Vanstone, cf. [MS04].
  - Der ECIES (Elliptic Curve Integrated Encryption Scheme) und
  - der PSEC (Provably Secure Encryption Curve Scheme) sind Weiterentwicklungen des Elgamal.
  - In der Vollausgabe des Skripts „Elliptische Kurven Kryptosysteme, ECCS und weitere Aspekte zu Signaturen, speziell von RSA“ (für das Modul wurde ja „nur“ ein Auszug hochgeladen) sind diese Themen drin. Bei Interesse kann diese Vollausgabe bei mir angefragt werden.

- Nachteil:
  - Ciphertext Y || K_E ist 3-mal so lang wie der Klartext T („||“ = Konkatenation = anhängen)

- Vorteile:
  - Probabilistische Verschlüsselung (im Gegensatz zum Schulbuch RSA)
    - Wegen der zufälligen Wahl des Faktors «i» wird der gleiche Klartext jedes mal in einen anderen Chiffretext verschlüsselt.
    - Wird der geheime Schlüssel k resp. d bekannt, dann kann man einen neuen Punkt Q als Public Key und eine neue Zahl k resp. d als Private Key wählen. Im Gegensatz zum RSA, wo alle Parameter geändert werden müssen.

Kap. 3.6

Signaturen mit EC

Signaturen mit EC

- Diverse Signaturalgorithmen werden mit EC umgesetzt.
- Nicht Prüfungsstoff!
- Die klassischen Signaturverfahren RSA, DSA, Nyberg-Rueppel und Schnorr können – mit Ausnahme des RSA – mit Elliptischen Kurven umgesetzt werden.
  - Signaturverfahren analog zu DSA → EC-DSA (cf. z.B. [CP10])
  - Signaturverfahren analog zu Nyberg-Rueppel → EC-NR (cf. z.B. [BH03])
  - Signaturverfahren analog zu Schnorr → EC-Schnorr (cf. z.B. [BSI_TR03])
  - Varianten zum EC-DSA
- In der Vollausgabe des Skripts „Elliptische Kurven Kryptosysteme, ECCS und weitere Aspekte zu Signaturen, speziell von RSA“ (für das Modul wurde ja „nur“ ein Auszug hochgeladen) sind diese Themen drin. Bei Interesse kann diese Vollausgabe bei mir angefragt werden.

Kap. 3.7

Eine (echte) EC

Eine echte EC

In der BSI Richtlinie [BSI_TR02] wird der rfc5639 zitiert: M. Lochter, J. Merkle, RFC 5639: Elliptic Curve Cryptography ECC Brainpool Standard Curves and Curve Generation, 2010, http://tools.ietf.org/html/rfc5639,
Hier werden konkrete Parameter angegeben.
Z.B. für eine 256-Bit Kurve:
p = A9FB57DBA1EEA9BC3E660A909D838D726E3BF623D52620282013481D1F6E5377
a = 7D5A0975FC2C3057EEF67530417AFFE7FB8055C126DC5C6CE94A4B44F330B5D9
b = 26DC5C6CE94A4B44F330B5D9BBD77CBF958416295CF7E1CE6BCCDC18FF8C07B6
x = 8BD2AEB9CB7E57CB2C4B482FFC81B7AFB9DE27E1E3BD23C23A4453BD9ACE3262
y = 547EF835C3DAC4FD97F8461A14611DC9C27745132DED8E545C1D54C72F046997
Dabei sind (x, y) die Koordinaten des Basispunktes P.

Bemerkung:
Es gibt aber viele weitere Kurven und Standardisierungsgremien, z.B. die NIST-Kurven, die in den FIPS-Standards eingegangen sind. Oder die Koblitzkurven usw. Cf. [CP-D], Kap. 9.6.

Kap. 3.8

Unterschiede EC und RSA

Unterschiede zur Verschlüsselung mit EC und RSA

- Bei RSA viel grössere Bitgrössen.
- Operationen bei RSA viel aufwändiger.
- EC ist probabilistisch, der (Schulbuch-)RSA nicht.
- Bei ECC gibt es Standards für die Public Parameter a, b, p & P.
- Beim RSA kann die Reihenfolge bei der Erzeugung der private und public Exponenten d & e beliebig gewählt werden, bei EC nicht. OK, vielfach auch nicht nötig, da eben bei EC standardisierte Public Parameter a, b, p & P genommen werden.
- Beim RSA werden beim Verschlüsseln und Signieren im Wesentlichen die Reihenfolge der Potenzierungen mit e resp. d getauscht. Bei EC sind Verschlüsselungen und Signaturen vollkommen verschieden.
- Bei EC gibt es viele Arten, wie man verschlüsseln und signieren kann.
- Der Angriff (die sog. Bellcore Attacke) mit dem Chinesischen Restsatz (CRT) auf den geheimen Exponenten d funktioniert nur beim RSA, nicht aber bei ECC.

Kap. 3.9

Angriffe auf EC

Angriffe auf EC, allgemein

Es funktionieren die gleichen Angriffe wie auf den Diffie-Hellman, cf. Kap. 8.3.3 in [CP-D]
→ nicht Prüfungsstoff.

Die folgende Tabelle gibt Auskunft, wie weit EC schon geknackt sind.

Table 13.2 Record discrete log computations in elliptic curves

Bits	Announced
112	July 2009
113	April 2014
114	21 August 2017
114	June 2020

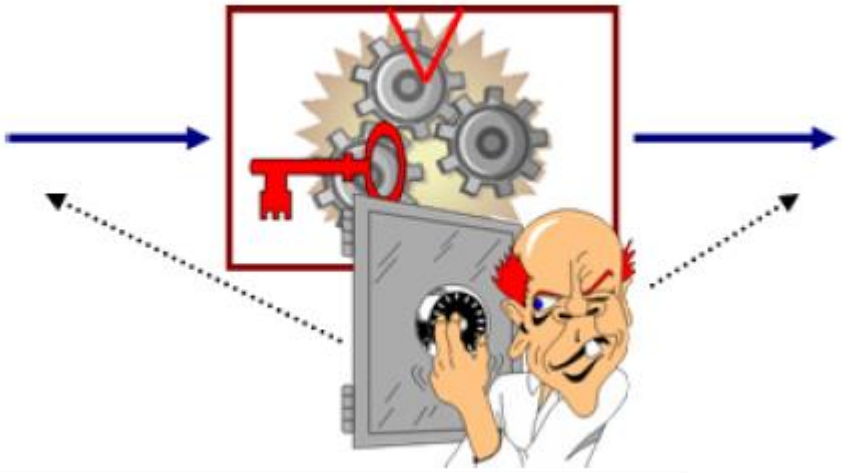
Angriffe auf EC, Side Chanel Attacks

Es funktionieren auch die Seitenkanalangriffe.

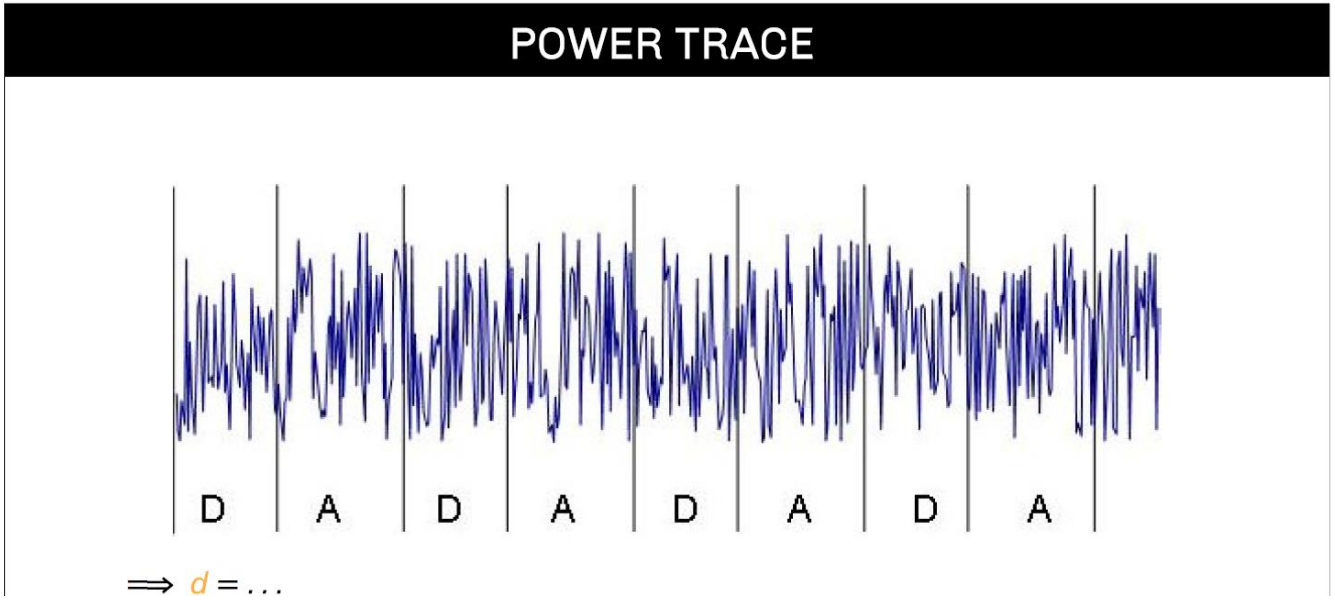
Präsenzbeispiel:

Die Regeln:

- Die Binärdarstellung beginnt mit einer 1 → mit dieser 1 muss nichts gemacht werden.
- Danach wird bei einer 0 ein Double gemacht, bei einer 1 ein Double und zusätzlich ein Add.



- Ergo: d = _____



Basis-Test Kap. 3, Anwendungen der EC

Aussage	Richtig oder falsch?	Begründung
Um eine analoge Sicherheit von EC wie bei Blockchiffren haben, muss die Schlüsselgrösse etwa doppelt so gross wie bei Blockchiffren sein.	☐ richtig ☐ falsch	
Von einer EC sind die Parameter a, b und p sowie die Punkte P und Q gegeben. Daher kann die Gleichung $Q = k \cdot P$ recht schnell auf „k“ aufgelöst werden, denn $k = \frac{P}{Q}$.	☐ richtig ☐ falsch	
Es gibt einen Diffie-Hellman mit EC, genannt DHEC.	☐ richtig ☐ falsch	
Nach einem DHEC haben beide Partner zwei gleiche Schlüssel, da man vom ausgetauschten Punkt sowohl die x- wie die y-Koordinate als symmetrischen Schlüssel verwenden kann.	☐ richtig ☐ falsch	
Es gibt weitere Varianten und Untervarianten zu einem Schlüsselaustausch mit EC.	☐ richtig ☐ falsch	
Es gibt – analog zu RSA – nur eine Variante, wie man mit EC verschlüsseln kann.	☐ richtig ☐ falsch	
Die Verschlüsselung mit EC ist – analog zu RSA – deterministisch.	☐ richtig ☐ falsch	
Wird der geheime Schlüssel bei EC bekannt, so müssen – analog wie beim RSA – sämtliche Parameter neu berechnet werden.	☐ richtig ☐ falsch	
Bei EC darf ich standardisierte Kurven verwenden. M.a.W. ev. verwenden Millionen von Teilnehmern die gleiche Kurve.	☐ richtig ☐ falsch	
Es gibt keine wesentlichen Unterschiede zw. RSA und EC, denn mit beiden kann man verschlüsseln, signieren und Schlüsselaustauschen.	☐ richtig ☐ falsch	
EC sind – im Gegensatz zu RSA – immun gegen Side Channel Attacken.	☐ richtig ☐ falsch	

Kap. 4

FORMELSAMMLUNG ZU DEN EC BERECHNUNGEN

Formelsammlung für ECC Berechnungen

Siehe Kap. 4 im Skript "ECCS"

Es sind alle wichtigen Formeln zusammengestellt:

- Definition der Elliptischen Kurve.
- Die Nichtsingularitätsbedingung
- Die Punktaddition
- Die Punktverdoppelung
- Anzahl der Punkte einer Kurve (Theorem von Hasse)
- Der Double and Add Algorithmus

Kap. 5

BEISPIEL UND AUFGABEN ZU DEN EC BERECHNUNGEN

Beispiele für EC Berechnungen

Siehe Kap. 5.1 Beispiele im Skript "ECCS"

Es ist nochmals ein sehr ausführliches Beispiel mit allen Facetten aufgeführt.

Siehe Kap. 5.2 Aufgaben im Skript "ECCS"

Und nochmals ausführliche Aufgaben.

Wichtig:

Die Basisaufgaben zu Kap. 4 & 5 sind die diversen, grundlegenden Rechenaufgaben. Daher sind keine Basisaufgaben im bisherigen Sinne mit «richtig» oder «falsch» vorhanden.

Teil 2

Weitere RSA Spezialtäten und blinde Signaturen

Kap. 6.1
Repetitionen zum RSA
& 6.2
Signatur mit RSA

Die Kap. 6.1 & 6.2 in „Elliptische Kurven Kryptosysteme, ECCS und weitere Aspekte zu Signaturen, speziell von RSA“ sind vorgängig durchzuarbeiten.

Kap. 6.3

Doppelunterschrift oder verteilte Signatur mit RSA

"BUT YOU SAID ON THE PHONE I COULD HAVE MY BEST FRIEND AS A CO-SIGNER."

Mathematische Grundlagen

Beim Signieren einer Meldung m (resp. m ist der Hash einer Meldung) wird $m^d mod\ N$ berechnet. Wobei d der geheime Exponent ist.
Um nun Doppelsignaturen zu realisieren gibt es die Möglichkeiten den geheimen Exponenten additiv oder multiplikativ aufzuteilen.

Additive Aufteilung des Exponenten

Es gilt: $s \equiv m^d mod\ N \equiv m^{d_1+d_2} mod\ N \equiv \left(m^{d_1} \cdot m^{d_2}\right) mod\ N$
$\equiv \left(m^{d_1} mod\ N \cdot m^{d_2} mod\ N\right) mod\ N \equiv (s_1 \cdot s_2) mod\ N,$
mit $d = (d_1 + d_2) mod\ \varphi(N)$

Multiplikative Aufteilung des Exponenten

Es gilt: $s \equiv m^d mod\ N \equiv m^{d_1 \cdot d_2} mod\ N \equiv \left(m^{d_1}\right)^{d_2} mod\ N$
$\equiv (s_1)^{d_2} mod\ N$, mit $d = (d_1 \cdot d_2) mod\ \varphi(N)$

Präsenzbeispiel „die additive Aufteilung des geheimen Exponenten“, cf. "ECCS", Kap. 6.3.1.

Ziel: Zwei Banken müssen gegenüber einem Kunden je den Vertrag m unterschreiben, damit er gültig ist.

Die Grundformel für die Signatur s:

$$s \equiv m^d mod\ N \equiv m^{d_1+d_2} mod\ N \equiv \left(m^{d_1} \cdot m^{d_2}\right) mod\ N \equiv s_1 \cdot s_2\ mod\ N,$$

mit $d = (d_1 + d_2) mod\ \varphi(N) \Rightarrow d_2 \equiv (d - d_1) mod\ \varphi(N)$

Die Parameter des Signatursystems:

(N, e) = (91, 5) und (p, q, d) = (7, 13, 29) seien der öffentliche resp. geheime Schlüssel.

Kontrolle der Parameter des Signatursystems & weitere Daten:

$$\varphi(N) = \varphi(7 \cdot 13) = \varphi(7) \cdot \varphi(13) = 6 \cdot 12 = 72 = (2 \cdot 2 \cdot 2 \cdot 3 \cdot 3)$$

$e = 5$ ist teilerfremd zu 72.

$$d = 5^{-1} mod\ 72 \equiv 29, denn\ 5 \cdot 29 \equiv 145 \equiv 2 \cdot 72 + 1 \equiv 1\ mod\ 72$$

Sei m = 4, wir wählen zufällig $d_1 = 20$ und berechnen

$d_2 \equiv 29 - 20$ mod 72 = 9

Präsenzbeispiel „die add. Auf. …“, Fortsetzung

Kunde kennt Public Key (91, 5)	**Meldung/ Vertrag(*)**	**Bank 1 kennt Secret Exp. $d_1 = 20$ Bank 2 kennt Secret Exp. $d_2 = 9$**
	m = 4, s_1 = 16 <----------	Bank 1 berech. Signatur s_1 von m = 4. s_1 = 4^2 mod 91 = 16
	m = 4, s_2 = 64 <----------	Bank 2 berech. Signatur s_2 von m = 4. s_2 = 4 mod 91 = 64
Berechnet s = 16 * 64 = 23 mod 91 Verifi. der Signatur: s^e mod N = 23 mod 91 = 4		**Kontrolle mit direkter Berechnung:** s = m^d mod N = 4^2 mod 31 = 23

Präsenzbeispiel „die add. Auf. …“, Fortsetzung

Frage: Funktioniert es auch mit $d_1 > d$?
Antwort: Ja, da $d_2 \equiv (d - d_1) mod\ \varphi(N)$

Fortsetzung Präsenzbeispiel mit $d_1 > d$:

Sei m = 4, wählen zufällig d_1 = 40 & berechnen
$d_2 \equiv 29 - 40 \bmod 72 = 61$

Das ergibt nun die folg. Werte/Resultate:

- $s_1 \equiv 4^{40} mod\ 91 = 74$
- $s_2 \equiv 4^{61} mod\ 91 = 4$
- $s \equiv 74 \cdot 4\ mod\ 91 = 23$
- Damit haben wir die gleiche Signatur erhalten wie vorhin!
- Es muss natürlich für eine beliebige additive Aufteilung von d = 29 immer die gleiche Signatur für m = 4 geben.

Beispiel „die add. Aufteilung des geheimen Exp.“

Die Parameter des Signatursystems:

(N, e) = (667, 9) und (p, q, d) = (23, 29, 137) seien der öffentliche resp. geheime Schlüssel.

Kontrolle der Parameter des Signatursystems & weitere Daten:

$\varphi(N) = \varphi(23 \cdot 29) = \varphi(23) \cdot \varphi(29) = 22 \cdot 28 = 616 = (2^3 \cdot 7 \cdot 11)$

$e = 9$ ist teilerfremd zu 616.

$d = 9^{-1} mod\ 616 \equiv 137, denn\ 9 \cdot 137 \equiv 1233 \equiv 1232 + 1 \equiv 1\ mod\ 72$

$\equiv 2 \cdot 616 + 1 \equiv 1\ mod\ 616$

Sei m = 38, wir wählen zufällig d_1 = 105 und berechnen

$d_2 \equiv 137 - 105 \bmod 616 = 32$

Beispiel „die add. Auf. …“, Fortsetzung

Kunde kennt Public Key (667, 9)	**Meldung/ Vertrag(*)**	**Bank 1 kennt Secret Exp. $d_1 = 105$ Bank 2 kennt Secret Exp. $d_2 = 32$**
	m = 38, s_1 = 608 <----------	Bank 1 berech. Signatur s_1 von m = 38 $s_1 \equiv 38^{105} mod\ 667 = 608$
	m = 38, s_2 = 210 <----------	Bank 2 berech. Signatur s_2 von m = 38 $s_2 \equiv 38^{32} mod\ 667 = 210$
Berechnet $s \equiv 608 \cdot 210\ mod\ 667 = 283$ Verifi. der Signatur: $283^9 mod\ 91 = 38$		**Kontrolle mit direkter Berechnung:** $\mathbf{s \equiv 38^{137} mod\ 667 = 283}$

Beispiel „die add. Auf. …“, Fortsetzung

Frage: Funktioniert es auch mit $d_1 > d$?
Antwort: Ja, da $d_2 \equiv (d - d_1) mod\ \varphi(N)$

Fortsetzung Präsenzbeispiel mit $d_1 > d$:

Sei m = 38, wählen zufällig d_1 = 203 & berechnen
$d_2 \equiv 137 - 203 \bmod 616 = 550$

Das ergibt nun die folg. Werte/Resultate:

- $s_1 \equiv 38^{203} mod\ 667 = 260$
- $s_2 \equiv 38^{550} mod\ 667 = 645$ (direkte Berechnung mit TI-89 geht nicht, aber z.B. mit www.wolframalpha.com).
- $s \equiv 260 \cdot 645\ mod\ 667 = 283$
- Damit haben wir die gleiche Signatur erhalten wie vorhin!
- Es muss natürlich für eine beliebige additive Aufteilung von d = 137 immer die gleiche Signatur für m = 38 geben.

Präsenzbeispiel „die multiplikative Aufteilung des geheimen Exp.“, cf. "ECCS" Skript, Kap. 6.3.2.

Ziel: Ein Dokument m (z.B. Arbeitsvertrag) muss unterzeichnet (Partner 1) und gegengezeichnet (Partner 2) werden, damit er gültig ist.

Die Grundformel für die Signatur s:

$$s \equiv m^d mod\ N \equiv m^{d_1 \cdot d_2} mod\ N \equiv \left(m^{d_1}\right)^{d_2} mod\ N \equiv (s_1)^{d_2} mod\ N$$

mit $d \equiv (d_1 \cdot d_2) mod\ \varphi(N) \Rightarrow d_2 \equiv (d \cdot d_1^{-1}) mod\ \varphi(N)$

Die Parameter des Signatursystems:

(N, e) = (91, 5) und (p, q, d) = (7, 13, 29) seien der öffentliche resp. geheime Schlüssel.

Kontrolle der Parameter des Signatursystems & weitere Daten:

Erster Teil wie vorher

d_1 = 20 von vorher funktioniert nicht mehr! Grund: Es muss die Teilerfreiheit von $\varphi(N) = 72$ sichergestellt sein.
Sei m = 6, wählen d_1 = ___ & berech. d_2 =

Es muss gelten: $d \equiv (d_1 \cdot d_2) mod\ \varphi(N)$, also _____

Präsenzbeispiel „die mult. Auf. …“, Fortsetzung

Belieb. Verifyer kennt Public Key (91, 5)	**Meldung/ Vertrag(*)**	**Partner 1 kennt Secret Exponent (23) Partner 2 kennt Secret Exponent (67)**
		Partner 1 berech. Signatur s_1 von m = 6. $s_1 =$ _____
	m = 6, s = ____ <----------	Part. 2 berech. Signatur s von $s_1 =$ ____ $s =$ _____
Verifi. der Signatur: _____		

Beispiel „die mult. Aufteilung des geheimen Exp.“

Die Parameter des Signatursystems:

(N, e) = (667, 9) und (p, q, d) = (23, 29, 137) seien der öffentliche resp. geheime Schlüssel.

Kontrolle der Parameter des Signatursystems & weitere Daten:

Erster Teil wie vorher

$d_1 = 105 = 3 \cdot 5 \cdot 7$ von vorher funktioniert nicht mehr! Grund: Es muss die Teilerfreiheit von $\varphi(N) = 616 = (2^3 \cdot 7 \cdot 11)$ sichergestellt sein.

Sei m = 38, und wir wählen $d_1 = 201 = 3 \cdot 67$

und berechnen $d_2 = 137 \cdot \underbrace{201^{-1}}_{=521} mod\ 616 \equiv 537$

Es muss gelten:

$d \equiv (d_1 \cdot d_2) mod\ \varphi(N)$, also $137 \equiv (201 \cdot 537) mod\ 616$

Beispiel „die mult. Auf. …“, Fortsetzung

Bel. Verifyer kennt Public Key (667, 9)	Meldung/ Vertrag(*)	Part. 1 kennt Secret Exponent (201) Part. 2 kennt Secret Exponent (537)
		Part. 1 berech. Signatur s_1 von m = 38 $s_1 = m^{d_1} = 38^{201} mod\ 667 = 63$
	m = 38, s = 283 <----------	Part. 2 berech. Signatur s von $s_1 = 63$ $s = (s_1)^{d_2} = 63^{537} mod\ 667 = 283$
Verifi. der Signatur: $283^9 mod\ 667 = 38$		

Kurzanalyse der zwei Formen

- Additive Aufteilung, Vorteile:
  - Beide Partner unterschreiben die Meldung m.
  - Die Reihenfolge der Signaturschritte ist unabhängig.
  - d_1 kann völlig zufällig gewählt werden.
- Multiplikative Aufteilung:
  - Die obigen Vorteile sind bei der multiplikativen Aufteilung nicht mehr gegeben.
  - Hier muss einer der Partner zuerst die Meldung m (Teil-)signieren und dann der zweite die Teilsignatur unterschreiben, grundsätzlich müsste der zweite dazu die Meldung m nicht sehen, die er unterschreibt. Er kann aber nach der Unterschrift die Signatur verifizieren und sieht dann, ob er die richtige Meldung m unterschrieben hat.
  - d_1 kann nur im Rahmen der Teilerfreiheit zu $\varphi(N)$ zufällig gewählt werden (d.h. $ggt(d_1, \varphi(N)) = 1$).

Problemzonen der zwei Formen

- In beiden Aufteilungen gilt, dass das Protokoll keine Antwort darauf gibt, wie der Kunde bemerken oder kontrollieren kann, ob wirklich zwei Institutionen (z.B. Banken, Personen o.a.) unterschrieben haben.
- Ebenfalls gilt in beiden Aufteilungen, dass die zweite Institution jeweils den Wert d_1 kennen muss, um den eigenen Wert d_2 rechnen zu können.
- ***Lösung:***
  Beide kennen den Secret Key nicht und eine Trusted Third Party (was diese TTP auch immer sein könnte) müsste jeweils die Werte d_i den Institutionen geheim zustellen. Somit könnte auch die eine Institution nicht mehr im Namen der anderen unterschreiben.

- Im Falle der multiplikativen Aufteilung muss das Protokoll ggf. der wirklichen Situation angepasst werden. So ist das Unterzeichnen und Gegenzeichnen eines Arbeitsvertrages im Beispiel 6.3 nicht der Situation angepasst. In diesem Falle ist erst recht wichtig aufzuzeigen, wie der Arbeitnehmer zum Wert d_2 kommt. Ihr geht es kaum ohne TTP.

Aufgabe 5

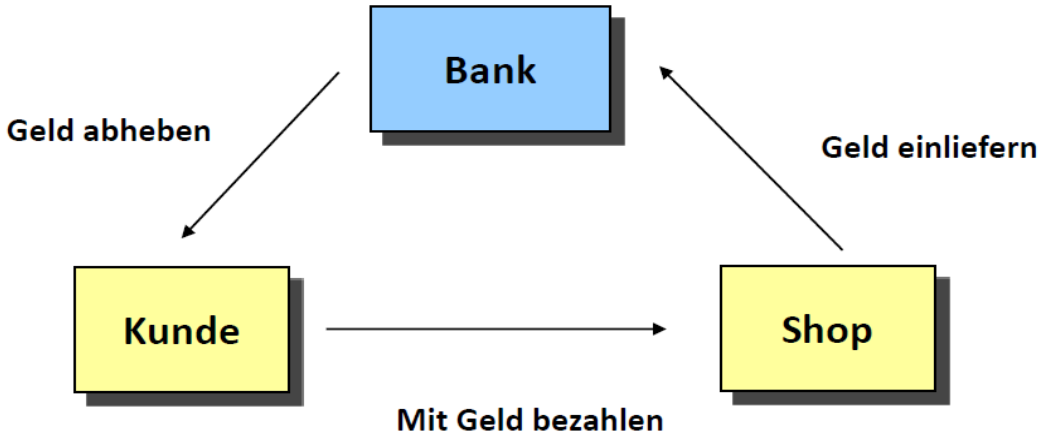
Betten Sie nun das Unterzeichnen und Gegenzeichnen eines Arbeitsvertrages in ein Protokoll ein.

Kap. 7

Blinde Signaturen

Der typische Ablauf bei einer Zahlung

Zunächst der typische (vereinfachte) Ablauf des Geldflusses bei einer Zahlung.



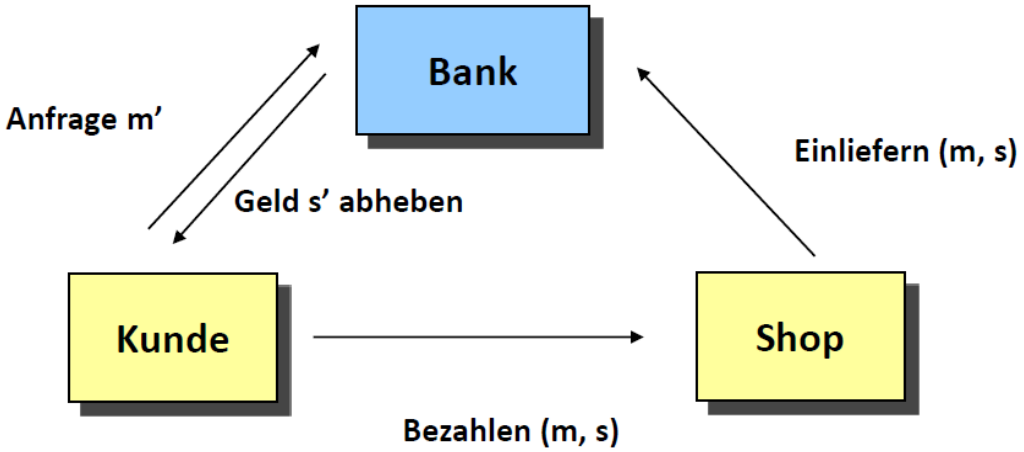
➔ Der Kunde hebt Geld von seinem Bankkonto ab.
➔ Der Kunde bezahlt mit diesem Geld irgendeine Ware.
➔ Der Shopbesitzer überweist dieses Geld auf sein Konto bei der gleichen Bank.

Idee der blinden Signaturen

Das Verwenden und die Grundidee von blinden Signaturen kann sehr gut an anonymen, digitalen Geld gezeigt werden.

Annahmen:

- Kunde und Shop sind bei der gleichen Bank.
- Es gibt nur z.B. 1.- Stücke.
- Weitere Werte würden mit dem gleichen Ablauf, aber unterschiedlichen Schlüsseln realisiert.



Ablauf Abheben & Bezahlen mit digitalem Geld

- ➔ Der Kunde bezeichnet (weltweit eindeutig) seine digitale Münze mit einem grossen zufällig (z.B. mit Zufallsgenerator → Wsk. dass zwei Münzen die Id.nummer hat muss verschwindend klein sein) gewähltem Wert m, m ist dann die Identifikationsnummer dieser Münze.
- ➔ Der Kunde „blindet" m zum Wert m' und frägt die Bank an, ob sie das Geld signiert und vom seinem Konto abbucht.
- ➔ Die Bank rechnet die digitale Signatur s' über den Wert m', bucht den Wert vom Konto des Kunden ab und schickt s' zurück.
- ➔ Der Kunde berechnet aus s' die Signatur s für den Wert m (K. entblindet).
- ➔ Der Kunde gibt das Geld mit der Identifikationsnummer m und der Signatur s aus.
- ➔ Der Shop kann mit dem Public Key und der Signatur kontrollieren, ob m von der Bank signiert wurde und dementsprechend eine echte Münze ist.
- ➔ Der Shop liefert das Geld mit der Identifikationsnummer m und der Signatur s der Bank ein.
- ➔ Die Bank verifiziert ebenfalls die Signatur. Die Bank weiss nun, dass sie irgendeinmal dieses Geld von einem Konto abgebucht hat, aber sie weiss nicht von welchem.

Gewünschte Eigenschaften des digitalen Geldes

Anonymität:
Der Kunde soll gegenüber der Bank ***anonym*** bleiben (vorausgesetzt der Shop liefert der Bank nicht den Namen und die Adresse des Kunden!!).

Unverfälschbarkeit (in klassischen Sinne „Insertion"): Es soll nicht möglich sein, dass falsches Geld erzeugt wird.

Unlinkbarkeit:
Selbst wenn (die Bank) von k Geldstücke m_i resp. m_i' sowie s_i' und s_i bekannt sind, kann man keinen Zusammenhang zu den Geldstücken und/oder Bezahler machen.

Bemerkung:
Auf weitere Eigenschaften wie „Double Spending", Anonymität gegenüber dem Shop usw. gehe ich an dieser Stelle nicht weiter ein.

Definition:
Ein **blindes Signaturverfahren** ist ein **konventionelles digitales Signaturverf.** mit der zusätzlichen Eigenschaft, dass der Unterschreibende **keine Information** über die unterschriebene **Nachricht** und die dazugehörende **Signatur** erhält.

Der allgemeine Ablauf

Kunde kennt Public Key (N, e)	Meldung	Bank kennt Secret Exponent d
1. Wahl der Nachricht m: m ist weltweit eindeutige Seriennummer		
2. Nachricht m „blinden”: Zufällige Wahl von $r \in_R \mathbb{Z}_N^*$ Berechnung von $m' \equiv r^e \cdot \mathrm{m}\ mod\ \mathrm{N}$		
3. geblindete Nachricht m’ schicken:	m‘ ---------->	
		4. Nachricht m’ signieren: $s' \equiv (m')^d\ mod\ \mathrm{N}$
	s‘ <----------	5. Signatur s’ zurückschicken:
6. Signatur s aus s’ extrahieren: $s \equiv s' \cdot r^{-1}\ mod\ \mathrm{N}$		
7. Signatur s prüfen: $\underbrace{m \equiv s^e\ mod\ \mathrm{N}}_{OK?}$		
8. Mit Münze (m, s) bezahlen:		
9. resp. 10 Signatur s prüfen: Shop wie Bank überprüfen die Echtheit der Münze, cf. Schritt 7.		

Beweis der Korrektheit

$$\frac{s'}{r} \equiv \frac{(m')^d}{r} \equiv \frac{(r^e \cdot m)^d}{r} \equiv \frac{(r^e)^d \cdot m^d}{r} \equiv \frac{r \cdot m^d}{r} \equiv m^d \; mod \; N = s$$

Bemerkung:
$\mathbb{Z}_N^*$ ist die Menge der natürlichen Zahlen a mod N, so dass ggt(N, a) = 1 ist. Damit r ein multiplikatives Inverses haben kann, muss $r \in \mathbb{Z}_N^*$ sein.
Es gilt:
$|\mathbb{Z}_N^*| = \varphi(N) = (p-1)(q-1)$, da $N = pq$, mit p, q Primzahlen.
OK: $r = 1$ ist sicher auch nicht gerade eine gute Wahl!

Frage: Fallen viele r weg?
Antwort:
Es fallen alle Vielfachen von p und q weg, das sind p + q Zahlen. Wenn p, q je 1024 Bit sind, dann ist N = 2048 Bit und p + q auch ungefähr 1024 Bit. Bei einer zufälligen Wahl ist die Wsk. sehr klein, nämlich ca. 2^{-1023}. Das ist sehr klein, denn 2^{-256} ist ca. Wsk. ein ganz bestimmtes Atom im Weltall in einem Griff erhaschen. Zudem ist der Test ganz einfach zu kontrollieren, ob man ein Vielfaches von p oder q gewählt hat.

Präsenzbeispiel „Blinde Signaturen“, cf. "ECCS" Skript, Kap. 7.2.1.

Die Parameter des Signatursystems:

(N, e) = (91, 5) und (p, q, d) = (7, 13, 29) seien der öffentliche resp. geheime Schlüssel.

Kontrolle der Parameter des Signatursystems & weitere Daten:

$\varphi(N) = \varphi(7 \cdot 13) = \varphi(7) \cdot \varphi(13) = 6 \cdot 12 = 72 = (2 \cdot 2 \cdot 2 \cdot 3 \cdot 3)$

$e = 5$ ist teilerfremd zu 72.

$d = 5^{-1} mod\ 72 \equiv 29, da\ 5 \cdot 29 \equiv 145 \equiv 2 \cdot 72 + 1 \equiv 1\ mod\ 72$

Sei m = 11, wir wählen zufällig r = 16, es muss gelten:
ggt(N, r) = ggt(91, 16) = 1

Erwartete Signatur:

Mit den Parametern des Signatursystems können wir die Signatur von m = 11 vorausberechnen: $s \equiv m^d\ mod\ N \equiv 11^{29}\ mod\ 91 = 72.$

Präsenzbeispiel „Blinde Signaturen“, Fortsetzung

Kunde kennt Public Key (91, 5)	Meldung	Bank kennt Secret Exponent d = 29
1. Wahl der Nachricht m: m = 11		
2. Nachricht m „blinden”: r = 16 $m' \equiv$ _____		
3. geblindete Nachricht m’ schicken:	m‘ = ____ ---------->	
		4. Nachricht m’ signieren: $s' \equiv$ _____
	s‘ = _____ <----------	5. Signatur s’ zurückschicken:
6. Signatur s aus s’ extrahieren: $s \equiv$_____		
7. Signatur s prüfen: $\underbrace{m \equiv \_\_\_\_\_}_{OK}$		
8. Mit Münze (11, 72) bezahlen:		
9. resp. 10 Signatur s prüfen: Shop wie Bank überprüfen die Echtheit der Münze, cf. Schritt 7.		

Beispiel „Blinde Signaturen“

Die Parameter des Signatursystems:

(N, e) = (667, 9) und (p, q, d) = (23, 29, 137) seien der öffentliche resp. geheime Schlüssel.

Kontrolle der Parameter des Signatursystems & weitere Daten:

$\varphi(N) = \varphi(23 \cdot 29) = \varphi(23) \cdot \varphi(29) = 22 \cdot 28 = 616$
$= (2^3 \cdot 7 \cdot 11)$
$e = 9$ ist teilerfremd zu 616.
$d = 9^{-1} mod\ 616 \equiv 137, denn\ 9 \cdot 137 \equiv 1233 \equiv 1232 + 1$
$\equiv 1\ mod\ 72$
$\equiv 2 \cdot 616 + 1 \equiv 1\ mod\ 616$

Sei m = 38, wir wählen zufällig r = 63, es muss gelten:
ggt(N, r) = ggt(667, 57) = 1

Erwartete Signatur:

Mit den Parametern des Signatursystems können wir die Signatur von m = 38 vorausberechnen: $s \equiv m^d\ mod\ N \equiv 38^{137}\ mod\ 667 \equiv 283.$

Beispiel „Blinde Signaturen“, Fortsetzung

Kunde kennt Public Key (667, 9)	Meldung	Bank kennt Secret Exponent d = 137
1. Wahl der Nachricht m: m = 38		
2. Nachricht m „blinden”: r = 63 $m' \equiv 63^9 \cdot 38 \; mod \; 667 \equiv 36$		
3. geblindete Nachricht m’ schicken:	m‘ = 36 ---------->	
		4. Nachricht m’ signieren: $s' \equiv (36)^{137} \; mod \; 667 \equiv 487$
	s‘ = 487 <----------	5. Signatur s’ zurückschicken:
6. Signatur s aus s’ extrahieren: $s \equiv 487 \cdot 63^{-1} \equiv 487 \cdot 180 \equiv 283 \; mod \; 667$		
7. Signatur s prüfen: $\underbrace{m \equiv s^e \equiv 283^9 \equiv 38 \; mod \; 667}_{OK}$		
8. Mit Münze (11, 72) bezahlen:		
9. resp. 10 Signatur s prüfen: Shop wie Bank überprüfen die Echtheit der Münze, cf. Schritt 7.		

Basis-Test Kap. 6 & 7, verteilte und Doppelsig.

Aussage	Richtig oder falsch?	Begründung
Um eine Doppelsignatur mit RSA zu implementieren muss man den öffentlichen Exponenten aufteilen.	☐ richtig ☐ falsch	
Um eine Doppelsignatur mit RSA zu implementieren muss man den geheimen Exponenten aufteilen.	☐ richtig ☐ falsch	
Um eine Doppelsignatur mit RSA zu implementieren kann man den entsprechenden Exponenten additiv oder multiplikativ aufteilen.	☐ richtig ☐ falsch	
Der geheime Exponent kann beliebig additiv aufgeteilt werden.	☐ richtig ☐ falsch	
Der geheime Exponent kann beliebig multiplikativ aufgeteilt werden.	☐ richtig ☐ falsch	
Beim Ablauf zur Erstellung von Doppelunterschriften ist bei einer additiven und multiplikativen Aufteilung identisch.	☐ richtig ☐ falsch	
Blinde Signaturen ist ein mathematisches Konzept, ohne Anwendung.	☐ richtig ☐ falsch	
Neben der Anonymität gibt es noch weitere gewünschte Eigenschaften des digitalen Geldes.	☐ richtig ☐ falsch	

Teil 3 = Kap. 8.4

Kurze Betrachtung von RSA – EC Laufzeiten

In diesen Teil gehen wir kurz auf die unterschiedlichen Laufzeiten beim RSA und Elliptischen Kurven ein. JS Skript
„Elliptische Kurven Kryptosysteme, ECCS und weitere Aspekte zu Signaturen, speziell von RSA“, Kap. 8.4.

Zusammenfassung der Laufzeiten RSA-EC

Die Laufzeiten hängen von vielen Faktoren ab:

- Prozessortyp und mit/ohne Coprozessoren.
- Implementation der Exponentiation und Modulus beim RSA.
- Implementation der Punkteberechnung und Modulus bei ECC.
- Es kommt auch sehr darauf an, ob kleine öffentliche Exponenten (bei RSA) genommen werden oder nicht.
- Usw.
- Zusammenfassung:
  - Der ungefähre 7-fache Aufwand beim Verdoppeln des RSA Schlüssels bestätigt sich.
  - Ein ECC-224 ist immer noch schneller als ein RSA-1024.
  - Der Aufwand von ECC-160 zu ECC-224 erhöht sich um ca. 40%.
- Bei Interesse, beachten Sie das Kap. 8.4 im JS Skript „ECCS“.

Lösungen

Aufgabe 1

$$y^2 = x^3 - 4x \Rightarrow 4a^3 + 27b^2 \underbrace{=}_{a=-4\ \&\ b=0} 4 \cdot (-4)^3 + 27 \cdot 0^2 = -256 \neq 0$$

Aufgabe 2

a) $y^2 = x^3 + 17\ mod\ 541 \Rightarrow 4a^3 + 27b^2 \underbrace{=}_{a=0\ \&\ b=17} 4 \cdot 0^3 + 27 \cdot 17^2 \equiv 7803 \equiv 229 \not\equiv mod\ 541$

b) Theorem von Hasse mit p = 541

$$p + 1 - 2\sqrt{p} \leq |E| \leq p + 1 + 2\sqrt{p}$$

$541 + 1 - 2\sqrt{541} \leq |E| \leq 541 + 1 + 2\sqrt{541}$, also $495{,}5 \leq |E| \leq 588{,}5$
Die Kurve hat zwischen 496 und 588 Punkte.

Aufgabe 3

a) Es muss den gleichen Punkt (9; 5) geben.
b) Individuelle Lösungen, die anhand der Tabelle selber kontrolliert werden können.

Aufgabe 4

Individuelle Lösungen, die anhand der Tabelle selber kontrolliert werden können.

Aufgabe 5

Arbeitneh. kennt Secret Exp. d_2 & (N, e)	**Vertrag(*)**	**Arbeitgeber kennt Secret Exp. d_1**
		Arbeitgeber berechnet die Signatur s_1 von m. $s_1 = m^{d_1} mod\ N$
	m, s_1 **<----------**	
Arbeitnehmer gegenzeichnet das Dokument, d.h. berechnet die Signatur s von s_1 und damit von m. $s = (s_1)^{d_2} mod\ N$ Und prüft dann, ob $s^e mod\ N = m$ ist.		
	m, s **--------->**	Prüft dann, ob $s^e mod\ N = m$ ist.

(*) Resp. Hash des Vertrages.

Basis-Test Kap. 2, Theorie der EC

Aussage	Richtig oder falsch?	Begründung
Mit EC kann man – ähnlich zu RSA – signieren, Schlüsselaustauschen & (hybrid) verschlüsseln.	☑ richtig ☐ falsch	
Die EC werden den RSA kaum je ablösen.	☐ richtig ☑ falsch	Wegen der langen Schlüssel, wird der RSA abgelöst werden müssen. Ob es dann EC oder ggf. andere, sog. Post Quanten Algorithmen sind, sei einmal dahingestellt.
Bei EC wird – analog zu RSA – immer mit mod N ($N = p \cdot q$, p, q grosse Primzahlen – gerechnet.	☐ richtig ☑ falsch	Es wird mod p, p eine Primzahl gerechnet.
Bei EC wird – im Gegensatz zu RSA – addiert und verdoppelt, anstatt multipliziert und quadriert.	☑ richtig ☐ falsch	
Es können beliebige EC genommen werden.	☐ richtig ☑ falsch	Es müsse gewisse Bedingungen erfüllt sein, z.B. die Nichtsingularitätsbedingung.
Bei zwei gegebenen Punkten, z.B. P(2; 4), Q(3; 5) können für P + Q einfach die jeweiligen Koordinaten zusammengezählt werden, also: P(2; 4) + Q(3; 5) = S(5; 9).	☐ richtig ☑ falsch	Es muss die Formel zur Punktaddition verwendet werden.
Die Anzahl der Punkte können mit einer Formel abgeschätzt werden.	☑ richtig ☐ falsch	Mit dem Theorem von Hasse wird die Anzahl der Punkte abgeschätzt.
Die Anzahl der Punkte einer EC können die Anzahl der Atome im Weltall weit übertreffen.	☑ richtig ☐ falsch	Ja, bei weitem.
Die Sicherheit von EC basiert auf einer Einwegfunktion mit Trapdoor.	☐ richtig ☑ falsch	Es ist eine Einwegfunktion ohne Trapdoor.
Die Sicherheit von EC liegt im diskreten Logarithmen Problem für EC.	☑ richtig ☐ falsch	

Basis-Test Kap. 2, Theorie der EC, Fortsetzung

Aussage	Richtig oder falsch?	Begründung
Die Sicherheit von EC liegt, dass bei gegebenen Punkten P, Q mit $Q = i \cdot P$ das „i“ nicht berechnet werden kann.	☑ richtig ☐ falsch	Das ist das diskrete Logarithmen Problem für EC
$Q = i \cdot P$ wird – analog zum RSA mit dem SaM – mit „double and add“ gerechnet.	☑ richtig ☐ falsch	„daa“ ist der Standardalgorithmus. Es gibt aber – wie beim SaM auch verbesserte und schnellere Algorithmen. Diese werden in diesem Modul aber nicht besprochen.
Der Schnittpunkt des Koordinatensystems ist der Punkt (0; 0)	☑ richtig ☐ falsch	
Der Punkt (0; 0) kann nie Punkt einer Elliptischen Kurve sein.	☐ richtig ☑ falsch	Siehe z.B. die zwei Kurven in Folie 13.
Der Punkt (0; 0) ist das Nullelement der abelschen Gruppe mit der Punkteaddition.	☐ richtig ☑ falsch	Der unendlich ferne Punkt $\mathcal{O}$ ist das Nullelement der Gruppe.
Alle Punkte der EC bilden eine abelsche Gruppe, sie hat aber kein Nullelement.	☐ richtig ☑ falsch	Eine additive Gruppe muss ein Nullelement (= Neutralelement) haben. Bei einer multiplikativen Gruppe ist das Einselement das Neutralelement.
Alle Punkte der EC bilden mit der Punktaddition eine abelsche Gruppe, das Nullelement ist der unendlich ferne Punkt $\mathcal{O}$.	☑ richtig ☐ falsch	
Alle Punkte der EC bilden mit der Punktaddition eine abelsche Gruppe, die Gruppenordnung ist immer p, weil mod p gerechnet werden muss.	☐ richtig ☑ falsch	Die Anzahl der Elemente (= Anzahl der Punkte der Kurve) der Gruppe ist die Gruppenordnung.
Es gibt immer $\varphi(p) = p - 1$ primitive Elemente.	☐ richtig ☑ falsch	
Die Anzahl der Punkte der Kurve ist auch gleichzeitig die Ordnung der Gruppe.	☑ richtig ☐ falsch	

Basis-Test Kap. 2, Theorie der EC, Fortsetzung

Aussage	Richtig oder falsch?	Begründung
Die Gruppe kann auch Untergruppen enthalten.	☑ richtig ☐ falsch	Eine Gruppe G enthält immer {e} und G als sog. triviale Untergruppen. Wenn die Gruppenordnung prim ist, dann sind das auch die zwei einzigen Untergruppen. So oder so muss die Ordnung der Untergruppe, die Gruppenordnung teilen.
Für EC werden nur Gruppen genommen, die zyklisch sind.	☐ richtig ☑ falsch	Zyklische Gruppen sind natürlich von Vorteil. Oft enthalten die Gruppen sehr grosse Untergruppen, siehe z.B. bei der Beschreibung der Verschlüsselung nach Volker-Müller.
Für EC werden nur Gruppen genommen, deren Ordnung prim ist.	☐ richtig ☑ falsch	Siehe oben
Im Rahmen der klassischen Kryptoanalyse braucht es für EC eine deutlich kleinere Schlüsselgrösse als bei RSA, um die gleiche Sicherheit zu erreichen.	☑ richtig ☐ falsch	Ein 3072 Bit RSA entspricht etwa einer 256 Bit EC.
Beim Einsatz von Quantencomputer QC braucht es viel weniger Qubits um einen RSA zu knacken, als beim EC.	☐ richtig ☑ falsch	Aber mit QC müssen auch für EC grössere Schlüssel genommen werden. Mit QC entspricht ein 3072 Bit RSA etwa einer 1164 Bit EC. Daher werden nun Post Quanten Algorithmen entwickelt.

Basis-Test Kap. 3, Anwendungen der EC

Aussage	Richtig oder falsch?	Begründung
Um eine analoge Sicherheit von EC wie bei Blockchiffren haben, muss die Schlüsselgrösse etwa doppelt so gross wie bei Blockchiffren sein.	☑ richtig ☐ falsch	
Von einer EC sind die Parameter a, b und p sowie die Punkte P und Q gegeben. Daher kann die Gleichung $Q = k \cdot P$ recht schnell auf „k“ aufgelöst werden, denn $k = \frac{P}{Q}$.	☐ richtig ☑ falsch	Genau das geht nicht. Das ist das disk. Log.Problem für EC, die Punktedivision gibt es (zumindest bis jetzt) nicht.
Es gibt einen Diffie-Hellman mit EC, genannt DHEC.	☑ richtig ☐ falsch	
Nach einem DHEC haben beide Partner zwei gleiche Schlüssel, da man vom ausgetauschten Punkt sowohl die x- wie die y-Koordinate als symmetrischen Schlüssel verwenden kann.	☐ richtig ☑ falsch	Es darf nur die x-Koordinate des ausgetauschten Punktes als gemeinsamer sym. Schlüssel verwendet werden.
Es gibt weitere Varianten und Untervarianten zu einem Schlüsselaustausch mit EC.	☑ richtig ☐ falsch	
Es gibt – analog zu RSA – nur eine Variante, wie man mit EC verschlüsseln kann.	☐ richtig ☑ falsch	Es gibt mehrere Varianten, wir haben nur diejenige nach Volker-Müller in der Präsenz besprochen.
Die Verschlüsselung mit EC ist – analog zu RSA – deterministisch.	☐ richtig ☑ falsch	Die Verschlüsselung ist automatisch probabilistisch.
Wird der geheime Schlüssel bei EC bekannt, so müssen – analog wie beim RSA – sämtliche Parameter neu berechnet werden.	☐ richtig ☑ falsch	Es muss nur ein neues „k“ und damit ein neuer Punkt $Q = k \cdot P$ berechnet werden.
Bei EC darf ich standardisierte Kurven verwenden. M.a.W. ev. verwenden Millionen von Teilnehmern die gleiche Kurve.	☑ richtig ☐ falsch	Das ist – im Gegensatz zu RSA – absolut kein Problem. Der geheime Schlüssel k muss individuell gewählt werden.
Es gibt keine wesentlichen Unterschiede zw. RSA und EC, denn mit beiden kann man verschlüsseln, signieren und Schlüsselaustauschen.	☐ richtig ☑ falsch	Der zweite Teil (die Aufzählung der Anwendungen) ist zwar korrekt. Es gibt jedoch schon Unterschiede → cf. die entsprechende Folie zu Kap. 3.8.
EC sind – im Gegensatz zu RSA – immun gegen Side Channel Attacken.	☐ richtig ☑ falsch	Siehe entsprechende Folie in Kap. 3.9.

Basis-Test Kap. 6 & 7, verteilte und Doppelsig.

Aussage	Richtig oder falsch?	Begründung
Um eine Doppelsignatur mit RSA zu implementieren muss man den öffentlichen Exponenten aufteilen.	☐ richtig ☑ falsch	Der geheime Exponent muss aufgeteilt werden.
Um eine Doppelsignatur mit RSA zu implementieren muss man den geheimen Exponenten aufteilen.	☑ richtig ☐ falsch	
Um eine Doppelsignatur mit RSA zu implementieren kann man den entsprechenden Exponenten additiv oder multiplikativ aufteilen.	☑ richtig ☐ falsch	
Der geheime Exponent kann beliebig additiv aufgeteilt werden.	☑ richtig ☐ falsch	
Der geheime Exponent kann beliebig multiplikativ aufgeteilt werden.	☐ richtig ☑ falsch	Der erste Teil der Aufteilung muss teilerfremd zu $\varphi(N)$ sein, also $ggt(d_1;\ \varphi(N)) = 1$.
Beim Ablauf zur Erstellung von Doppelunterschriften ist bei einer additiven und multiplikativen Aufteilung identisch.	☐ richtig ☑ falsch	Z.B. muss bei der multiplikativen Aufteilung die Reihenfolge eingehalten werden.
Blinde Signaturen ist ein mathematisches Konzept, ohne Anwendung.	☐ richtig ☑ falsch	Sie wird für anonymes, digitales Geld verwendet.
Neben der Anonymität gibt es noch weitere gewünschte Eigenschaften des digitalen Geldes.	☑ richtig ☐ falsch	

Herzlichen Dank an

- Prof. Dr. A. Steffen, Hochschule Rapperswil. Einige Folien entstammen aus der Vorlesung „Sichere Netzwerkkommunikation“.
- Prof. Dr. U. Maurer, ETHZ, für seine Vorlesungen in Kryptologie, wo ich diese Themen habe kennenlernen dürfen.
- Ein ganz besonderes Danke schön geht an Adrian Mischler – ein ehemaliger Student – für die Bereitstellung seiner ECC Software. Es erleichtert mir in vieler Hinsicht das Arbeiten mit EC.