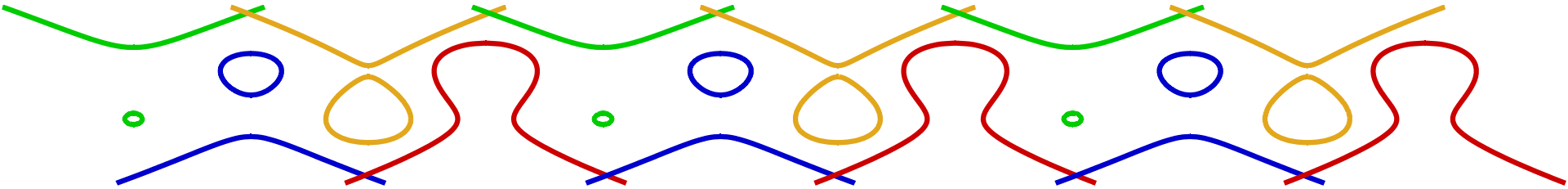
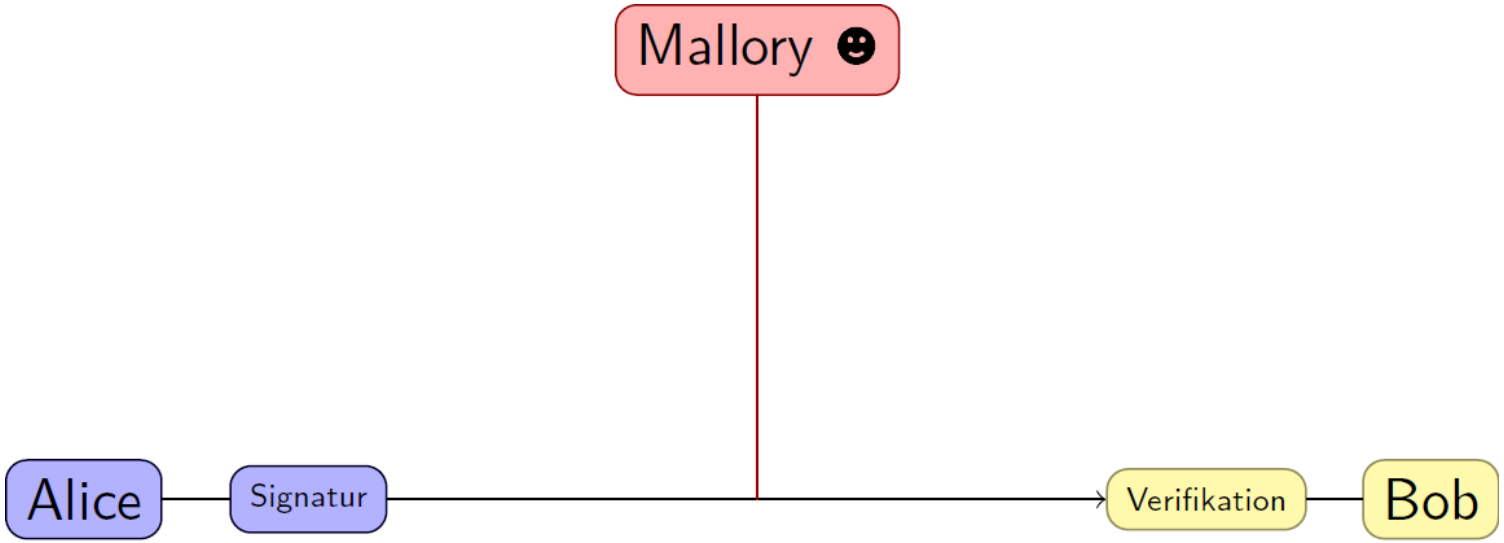
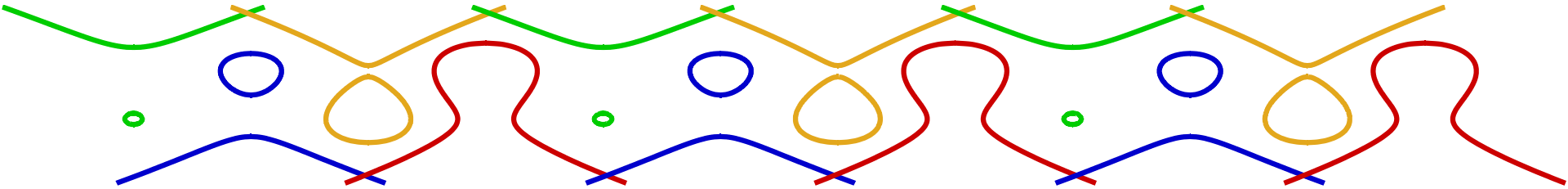


# Digitale Signaturen

Michael Artner, Belinda Brandstetter,
Elisabeth Bräuer, Elisabeth Galyo,
Levi Haunschmid, Rita Höller, Fabian Posch,
Florian Schininger, Stefan Schöberl,
Philipp Sellner, Patrick Weilerscheidt

# Wofür benötigt man eine Signatur?





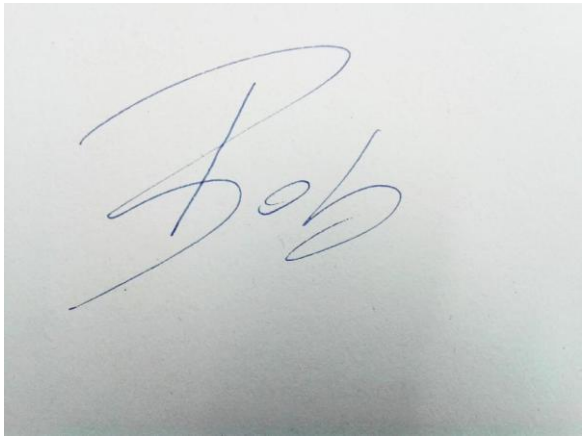
# Was ist eine Signatur?

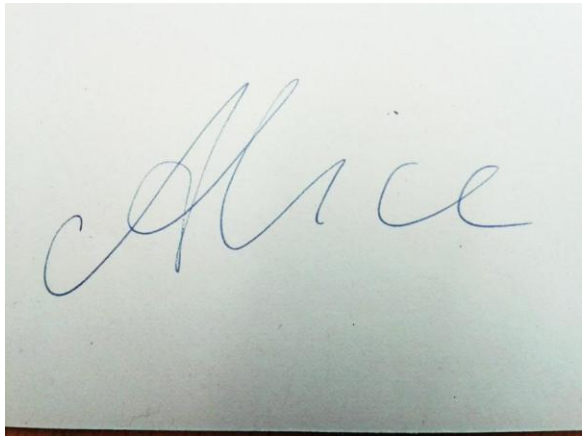
- Eindeutig zuordenbares Zeichen
- Kann von jedem erstellt werden
- Auf Echtheit überprüfbar → Verifizieren
- Anwendungen im Alltag:

  E-card, Zeugnisse, E-Mails, ...

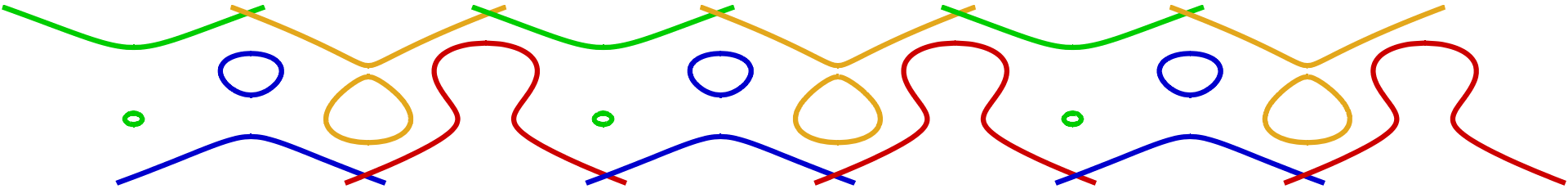
# Was ist eine Signatur?

- Signatur = Unterschrift
- Handschriftliche Unterschrift



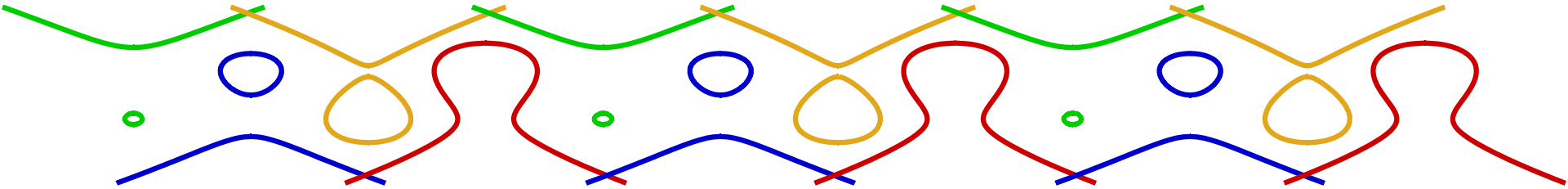


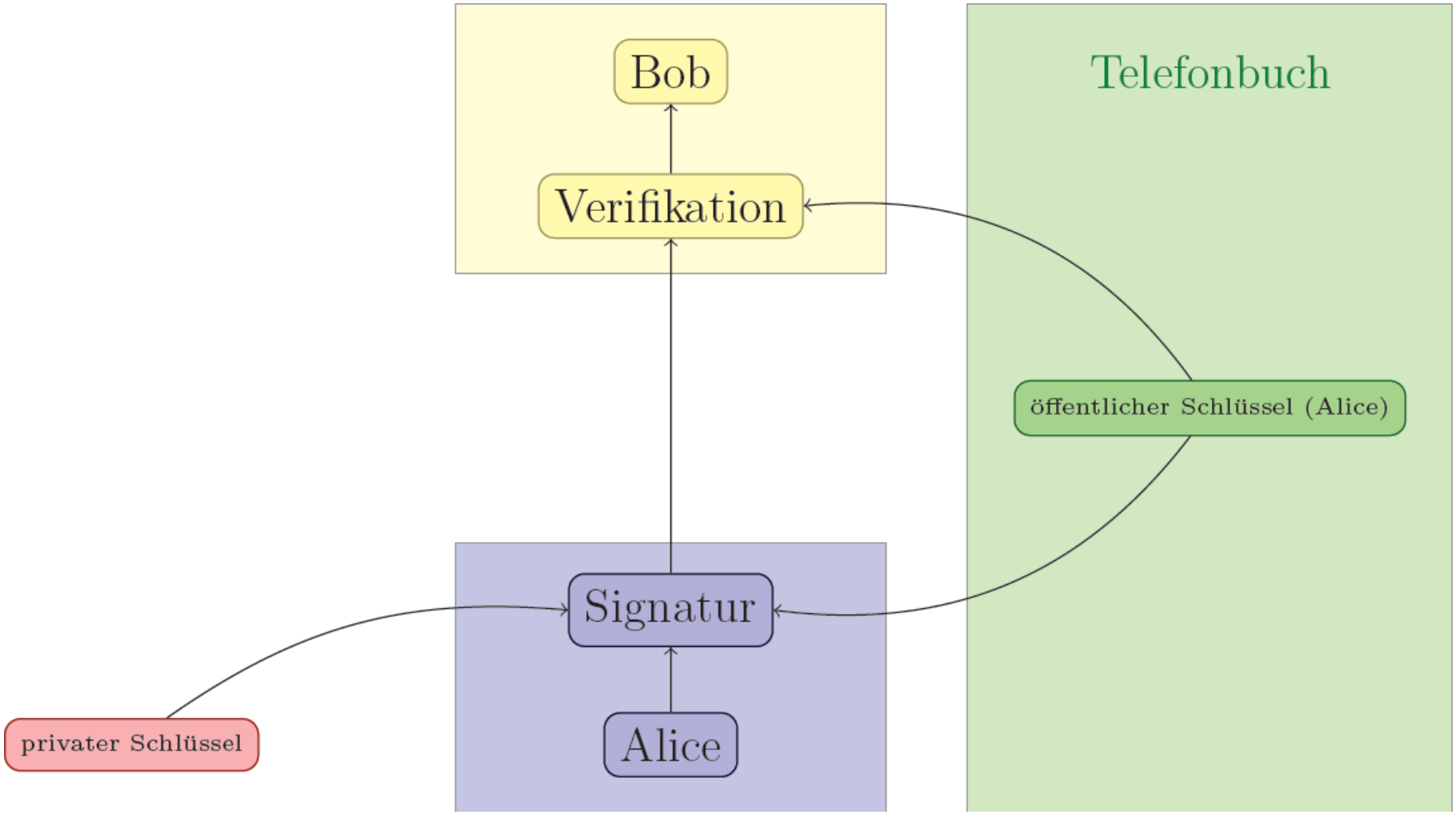
- Digitale Signatur

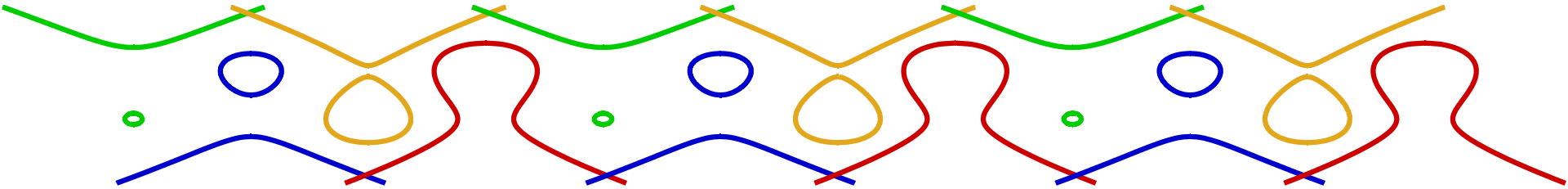


# Öffentlicher / Privater Schlüssel

- Öffentlicher Schlüssel: allen über Trustcenter („Telefonbuch“) zugänglich
- Privater Schlüssel: nur signierender Person bekannt
- Signieren: beide Schlüssel benötigt
- Verifizieren: nur öffentlicher Schlüssel benötigt

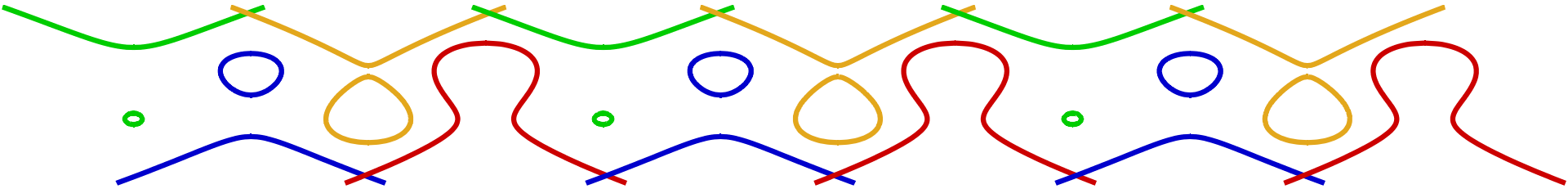



Bob
Verifikation
Telefonbuch
öffentlicher Schlüssel (Alice)
Signatur
privater Schlüssel
Alice



# Rechnen mit Kongruenzen

$$a \equiv b \bmod m \Leftrightarrow m \mid (b - a)$$



# ElGamal – Schlüssel erzeugen

$$p \in P$$

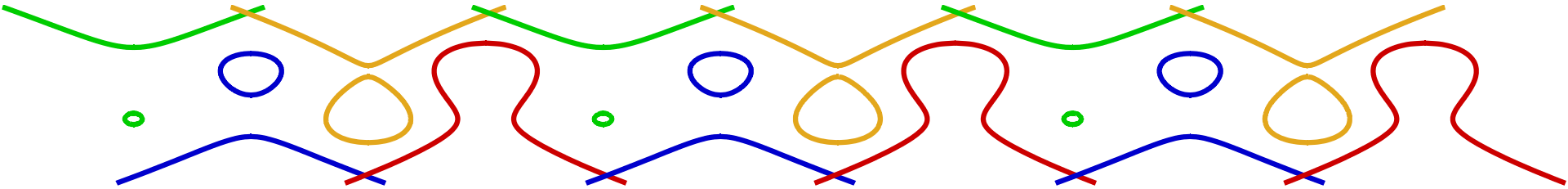
$$g \in \mathbb{Z}_p$$

$$x \in \mathbb{Z}_p$$

$$y \equiv g^x \bmod p$$

Öffentlicher Schlüssel: (p,g,y)

Privater Schlüssel: x



# ElGamal – Signieren

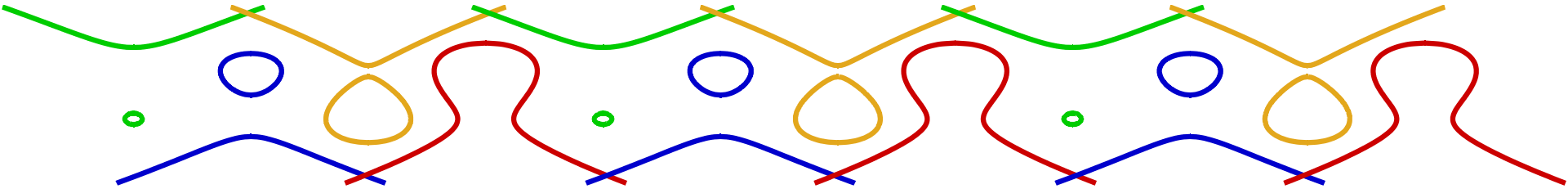
$$Nachricht: m \in \mathbb{Z}_p$$

$$k \in \mathbb{Z}_p mit\ ggt(k, p-1) = 1$$

$$r \equiv g^k \bmod p$$

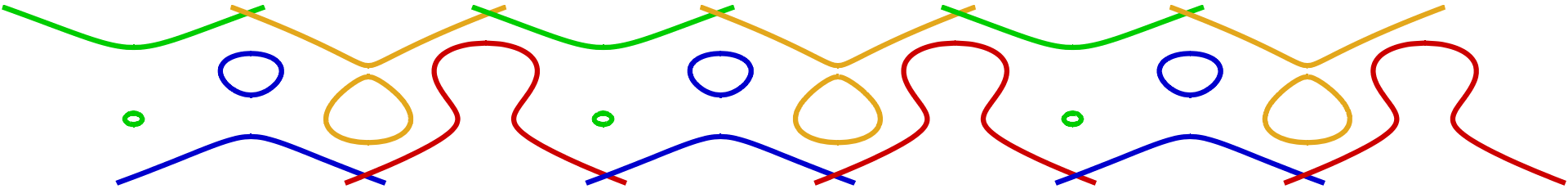
$$s \equiv (m - xr)k^{-1} \bmod p - 1$$

Signatur: (r,s)



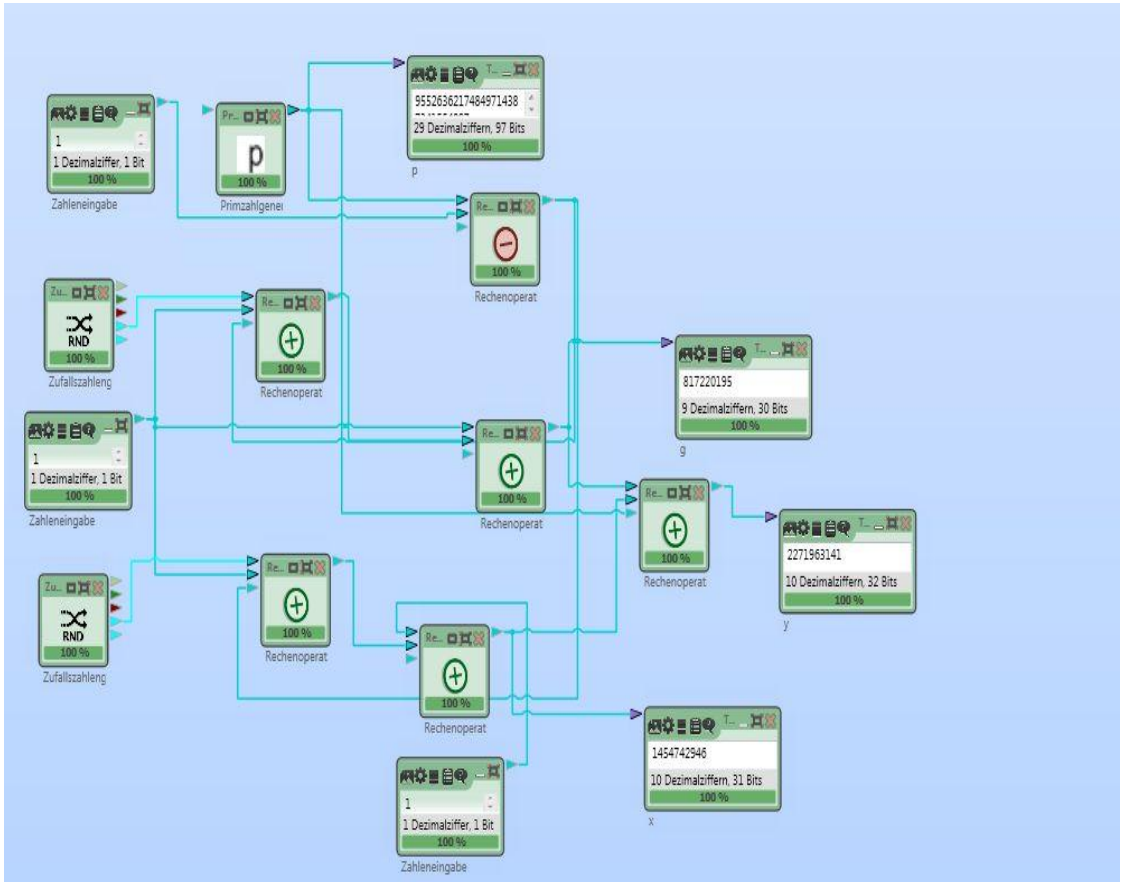
# ElGamal – Verifizieren

$$g^m \equiv y^r r^s \bmod p$$

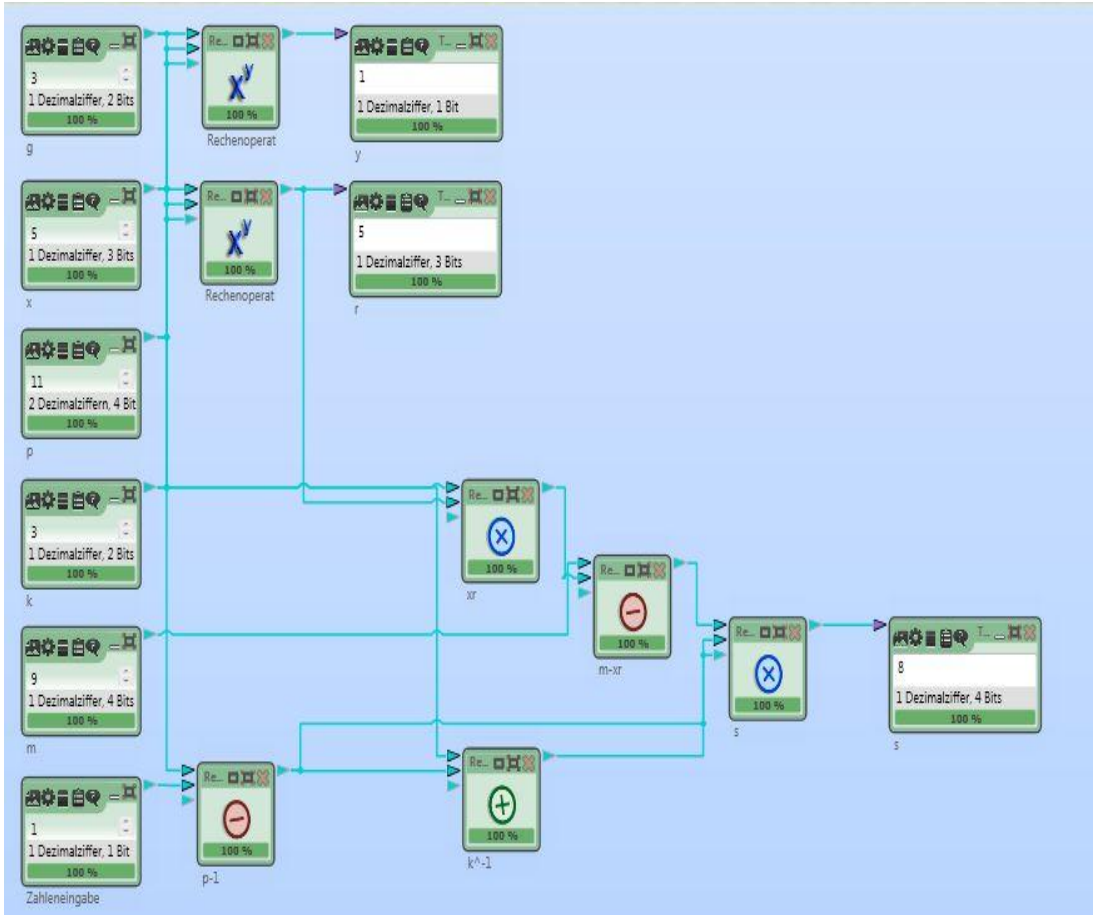



- Keine Programmierkenntnisse benötigt
- Arbeiten mit Funktionsblöcken
- Erleichterung der Arbeit
- www.cryptool.org

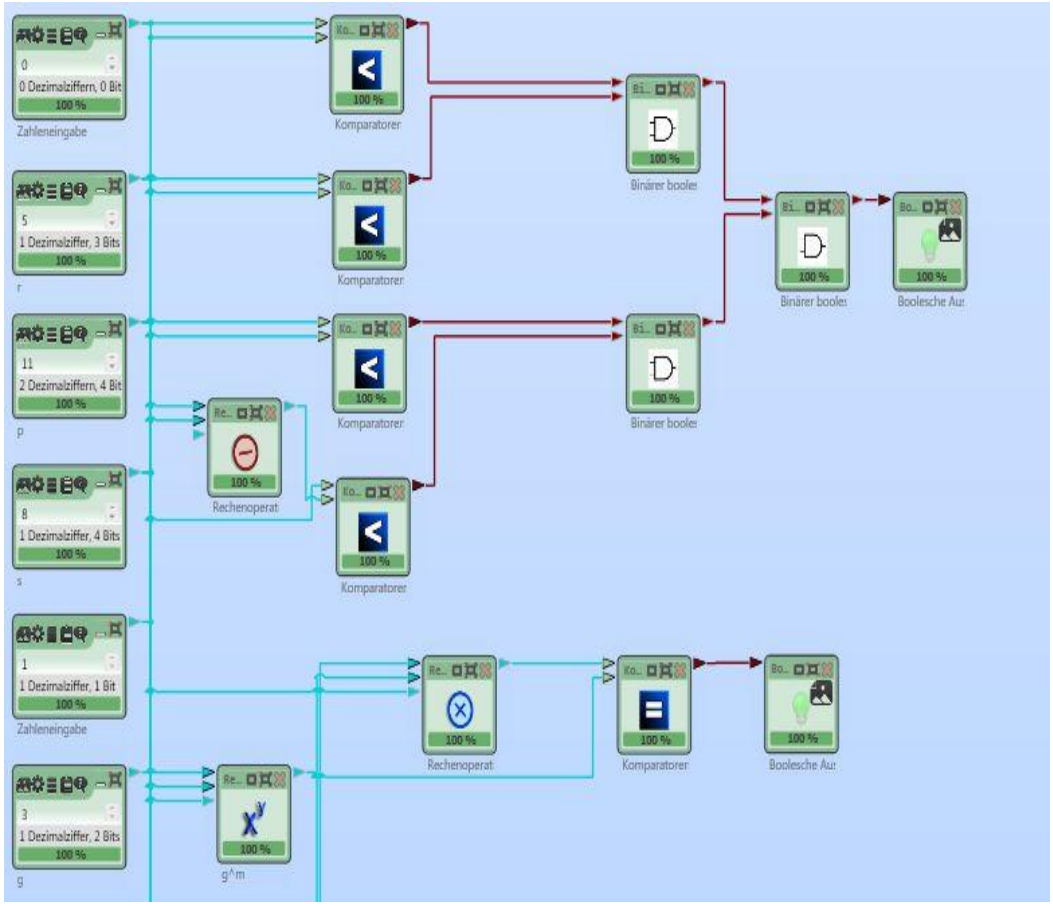
# Schlüsselgenerator



# Signieren


3
1 Dezimalziffer, 2 Bits
100 %
g
Rechenoperat
1
1 Dezimalziffer, 1 Bit
100 %
y
5
1 Dezimalziffer, 3 Bits
100 %
x
Rechenoperat
5
1 Dezimalziffer, 3 Bits
100 %
r
11
2 Dezimalziffern, 4 Bit
100 %
p
3
1 Dezimalziffer, 2 Bits
100 %
k
xr
m-xr
9
1 Dezimalziffer, 4 Bits
100 %
m
s
8
1 Dezimalziffer, 4 Bits
100 %
s
1
1 Dezimalziffer, 1 Bit
100 %
Zahleneingabe
p-1
k^-1

# Verifizieren

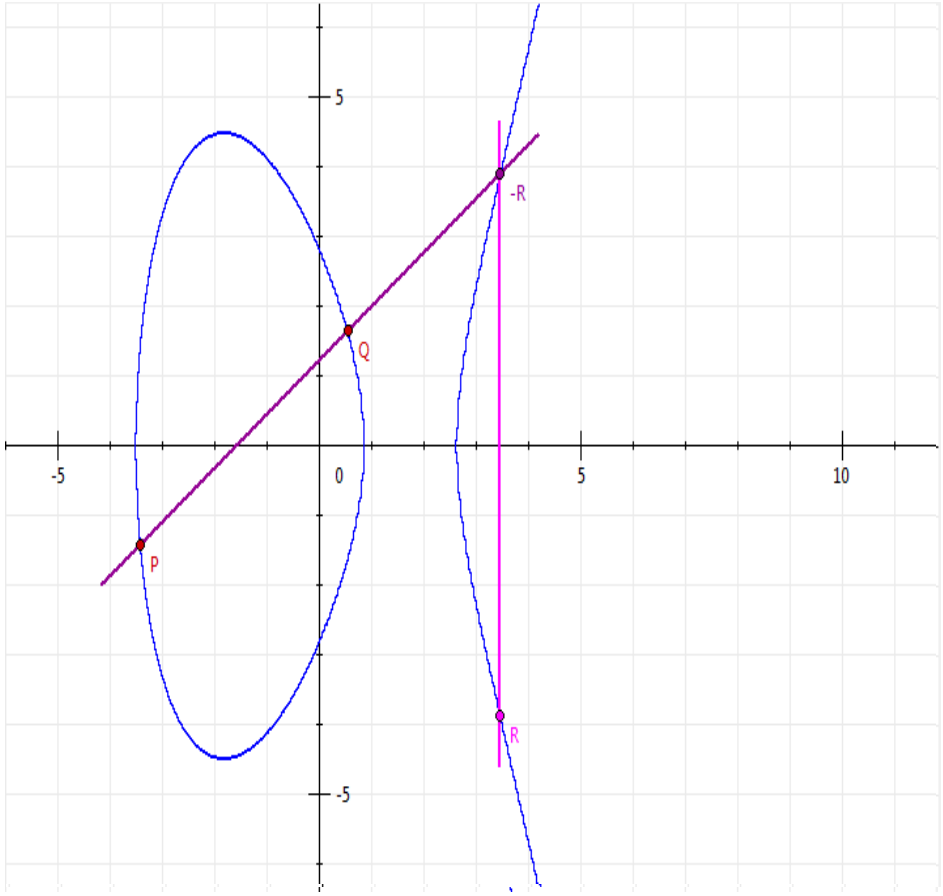


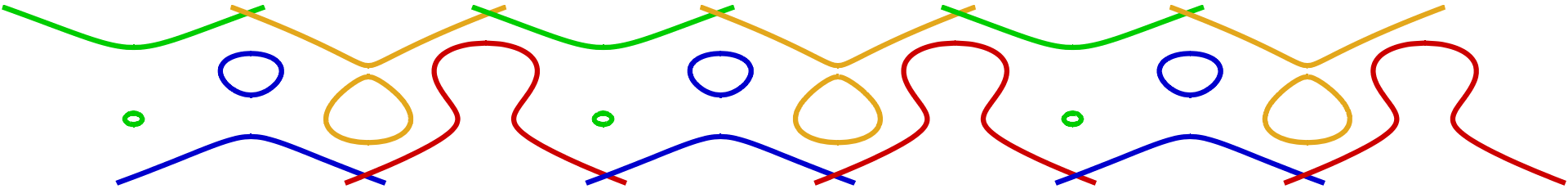
# Elliptische Kurven

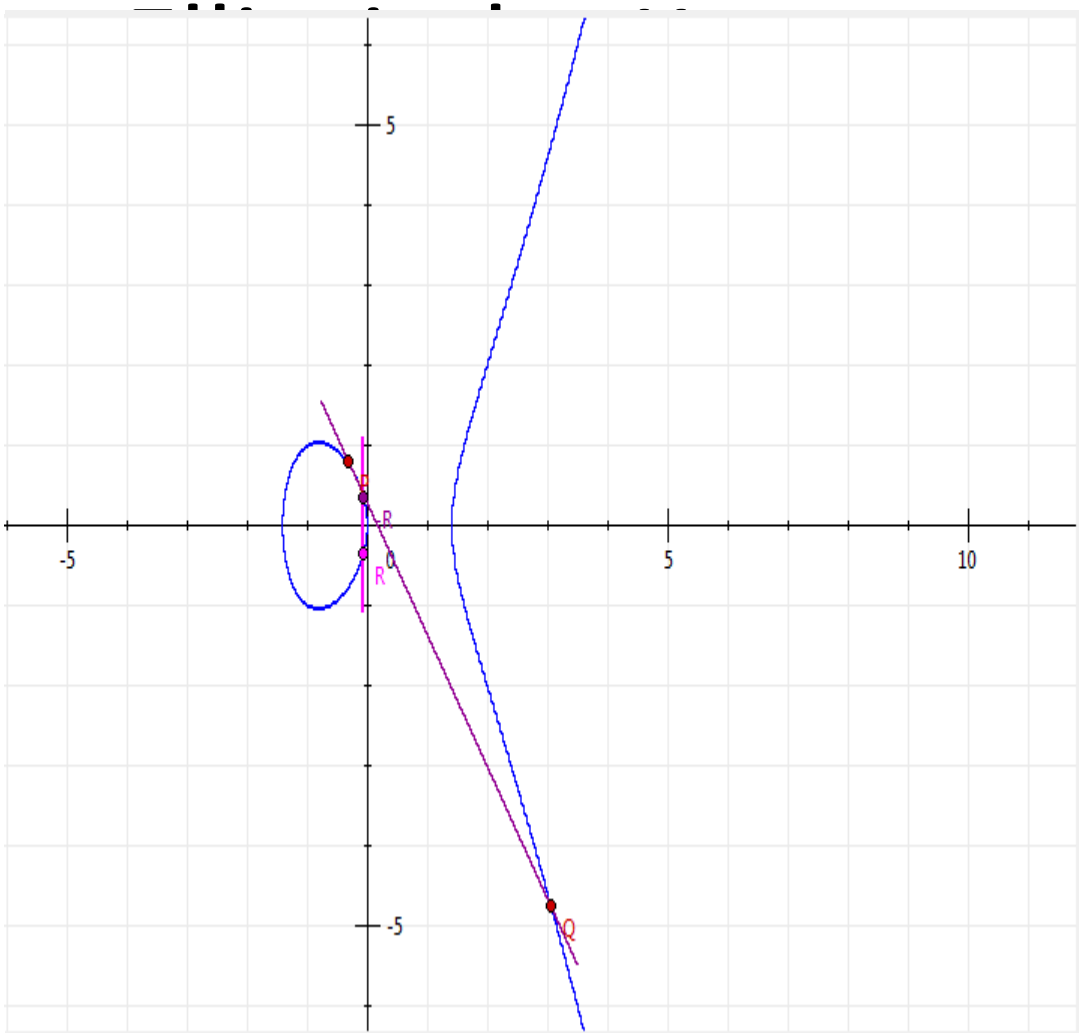
$$y^2 = x^3 + ax + b$$

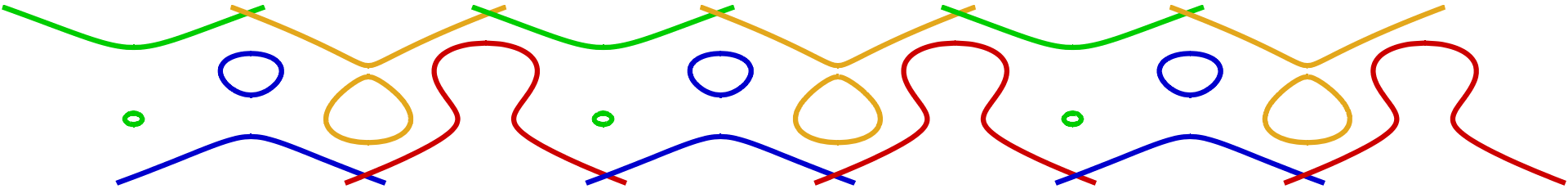
- Formgebung durch a und b

- P $\oplus$ Q = R

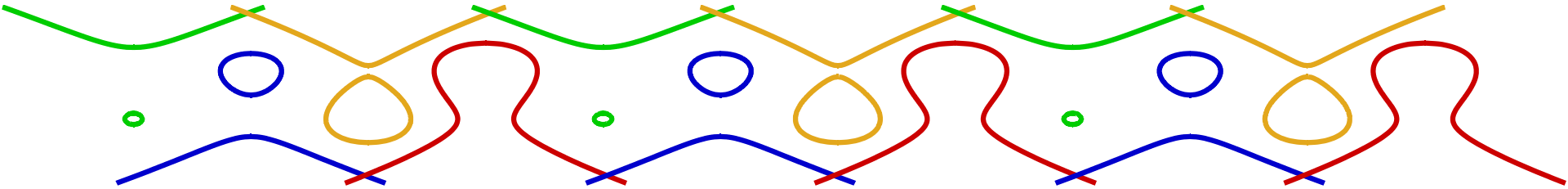





5
-5
0
5
10
-5
P
R
R'
Q



# Implementierung in Java

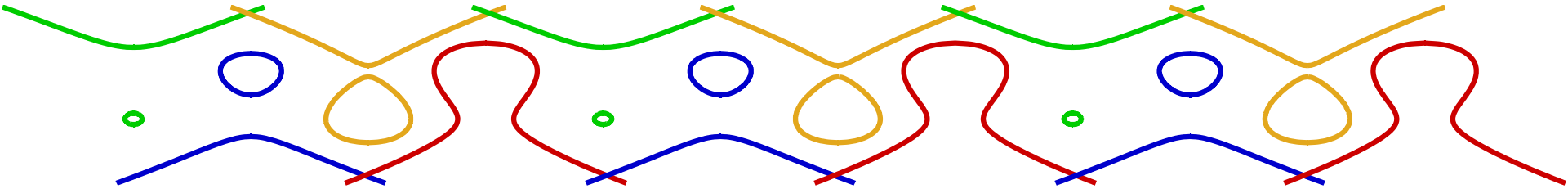


Hallo Bob,

Ich habe gerade in so einer sehr interessanten Präsentation von jungen und gebildeten Forschern erfahren, dass es eine noch sicherere Methode gibt Nachrichten zu signieren, als wir es jetzt tun. Die Schülerinnen und Schüler haben etwas von einem Verfahren mit elliptischen Kurven erzählt. Diese soll angeblich erheblich schwerer zu fälschen sein als das ElGamal Verfahren, welches wir derzeit verwenden. Darum würde ich vorschlagen das wir uns mal treffen, um uns mal diese Methode mit den elliptischen Kurven anzuschauen und eventuell zu verwenden.

Deine Alice

Signatur: (48304758851598623094070256454;
63815415811261394083760787354O )



Wir hoffen, Sie wissen nun, wie Sie Ihre E-Mails, Dokumente, Briefe, … signieren können.

Signatur: (74280264462318477115933050213; 171924217970664450544169350860 )