Elliptische krommen en digitale handtekeningen in Bitcoin

Bas Edixhoven

Universiteit Leiden

KNAW Bitcoin symposium

Elliptische krommen en digitale handtekeningen in Bitcoin

Bas Edixhoven

Universiteit Leiden

KNAW Bitcoin symposium

Deze aantekeningen zal ik op mijn homepage plaatsen.

Een paar woorden over Bitcoin

Bitcoin is een digitale munt, gebaseerd op openbronprogramma's, functionerend zonder centrale autoriteit.

Een paar woorden over Bitcoin

Bitcoin is een digitale munt, gebaseerd op openbronprogramma's, functionerend zonder centrale autoriteit.

Alle details van hoe het werkt zijn openbaar (protocol, software).

Een paar woorden over Bitcoin

Bitcoin is een digitale munt, gebaseerd op openbronprogramma's, functionerend zonder centrale autoriteit.

Alle details van hoe het werkt zijn openbaar (protocol, software).

Doel van deze presentatie.

- Toelichting geven over één van de bouwstenen van Bitcoin: Elliptic Curve Digital Signature Algorithm (ECDSA).

Een paar woorden over Bitcoin

Bitcoin is een digitale munt, gebaseerd op openbronprogramma's, functionerend zonder centrale autoriteit.

Alle details van hoe het werkt zijn openbaar (protocol, software).

Doel van deze presentatie.

- Toelichting geven over één van de bouwstenen van Bitcoin: Elliptic Curve Digital Signature Algorithm (ECDSA).

Met zo'n handtekening bewijs je eigenaar te zijn van de Bitcoins die je uitgeeft.

Een paar woorden over Bitcoin

Bitcoin is een digitale munt, gebaseerd op openbronprogramma's, functionerend zonder centrale autoriteit.

Alle details van hoe het werkt zijn openbaar (protocol, software).

Doel van deze presentatie.

- Toelichting geven over één van de bouwstenen van Bitcoin: Elliptic Curve Digital Signature Algorithm (ECDSA).

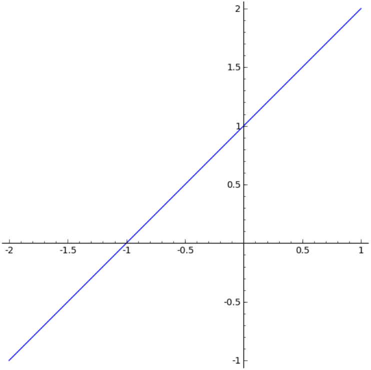
Met zo'n handtekening bewijs je eigenaar te zijn van de Bitcoins die je uitgeeft.

Referenties:

- `https://en.bitcoin.it/wiki/Main_Page`
- `http://en.wikipedia.org/wiki/Elliptic_curve_cryptography`
- `http://en.wikipedia.org/wiki/Elliptic_Curve_DSA`

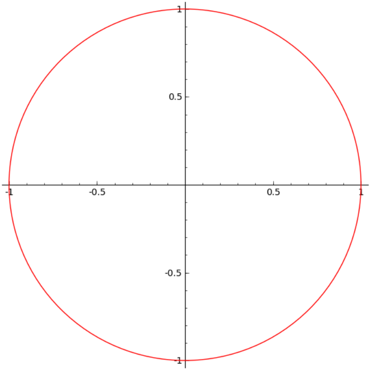
Een vlakke kromme van graad 1

Wiskunde, terug naar René Descartes (1630).



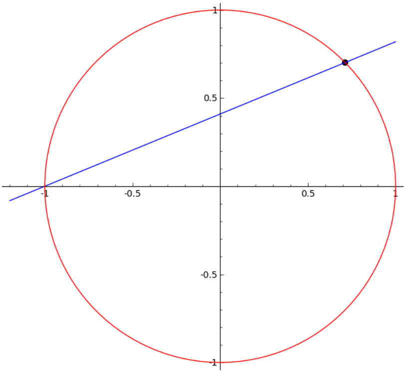
De lijn gegeven door de vergelijking $y = x + 1$.

Een vlakke kromme van graad 2



De cirkel gegeven door de vergelijking $x^2 + y^2 = 1$.

Algebraïsche parametrisatie van de cirkel



Lijnen door $(-1, 0)$:

$$y = a \cdot (x + 1).$$

Tweede snijpunt:

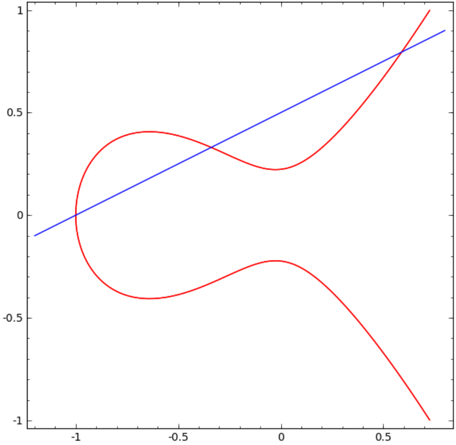
$$\left(\frac{1 - a^2}{1 + a^2}, \frac{2a}{1 + a^2} \right)$$

$$\mathbb{R} \to \text{cirkel}, \quad a \mapsto \left(\frac{1 - a^2}{1 + a^2}, \frac{2a}{1 + a^2} \right).$$

Elliptische krommen zijn van graad 3

Vergelijking:

$$y^2 = (x+1)(x^2+0.05).$$

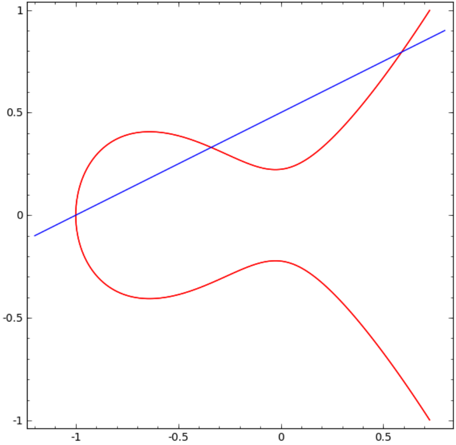


Elliptische krommen zijn van graad 3

Vergelijking:

$$y^2 = (x+1)(x^2+0.05).$$

$E(\mathbb{R})$ is de verzameling oplossingen (x, y) in $\mathbb{R}^2$.



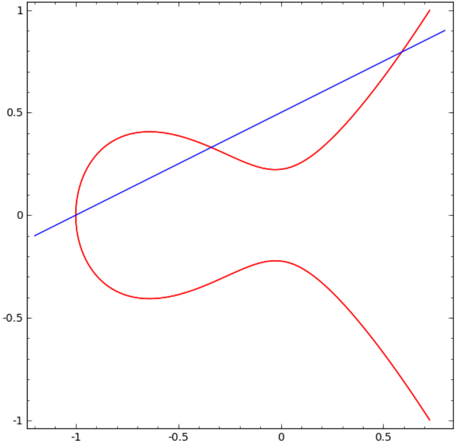
Elliptische krommen zijn van graad 3

Vergelijking:

$$y^2 = (x+1)(x^2+0.05).$$

$E(\mathbb{R})$ is de verzameling oplossingen (x, y) in $\mathbb{R}^2$.

Lijnen snijden $E(\mathbb{R})$ in 1 of 3 punten...



Elliptische krommen zijn van graad 3

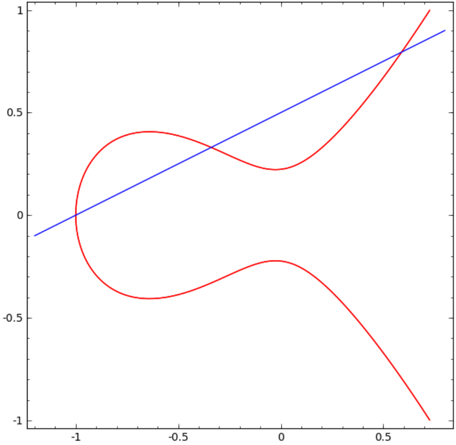
Vergelijking:

$$y^2 = (x+1)(x^2+0.05).$$

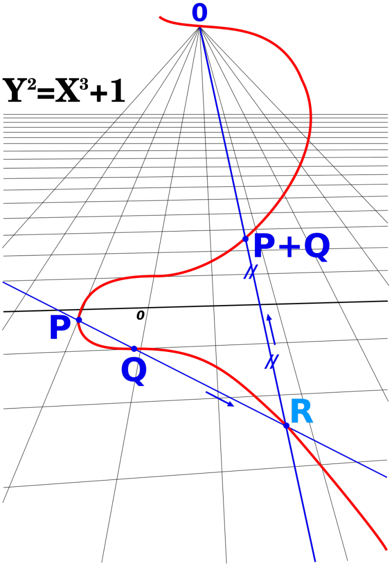
$E(\mathbb{R})$ is de verzameling oplossingen (x, y) in $\mathbb{R}^2$.

Lijnen snijden $E(\mathbb{R})$ in 1 of 3 punten. . .

als we met multipliciteiten tellen *en* een punt "op oneindig" toevoegen.

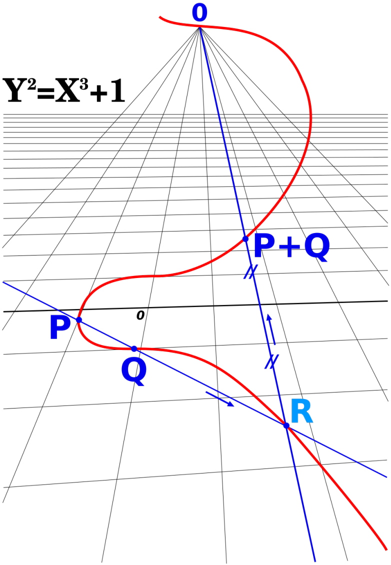


Dit is projectieve meetkunde.



Dit is projectieve meetkunde.

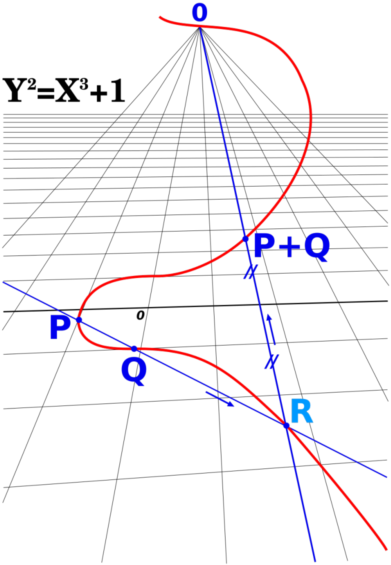
Parallelle lijnen snijden op de horizon.



Dit is projectieve meetkunde.

Parallelle lijnen snijden
op de horizon.

$E(\mathbb{R})$ heeft één punt "O"
op de horizon.

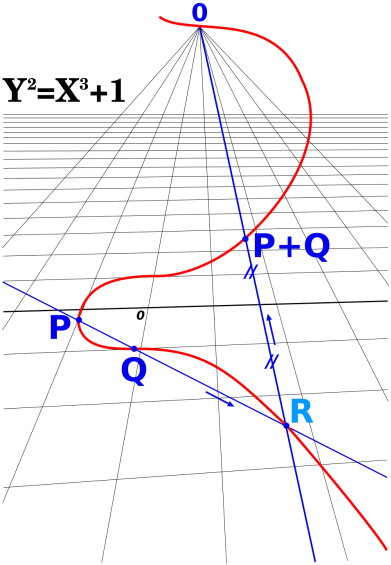


Dit is projectieve meetkunde.

Parallelle lijnen snijden
op de horizon.

$E(\mathbb{R})$ heeft één punt "O"
op de horizon.

$E(\mathbb{R})$ heeft géén algebraïsche
parametrisering.



Dit is projectieve meetkunde.

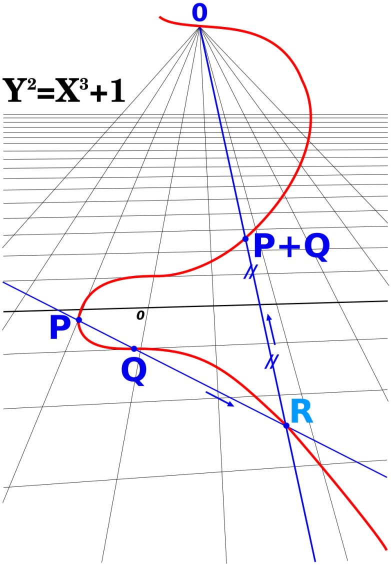
Parallelle lijnen snijden
op de horizon.

$E(\mathbb{R})$ heeft één punt "O"
op de horizon.

$E(\mathbb{R})$ heeft géén algebraïsche
parametrisering.

Maar $E(\mathbb{R})$ (inclusief O) heeft
een *binaire operatie*:

$$E(\mathbb{R}) \times E(\mathbb{R}) \to E(\mathbb{R}),$$
$$(P, Q) \mapsto P + Q.$$



Dit is projectieve meetkunde.

Parallelle lijnen snijden
op de horizon.

$E(\mathbb{R})$ heeft één punt "O"
op de horizon.

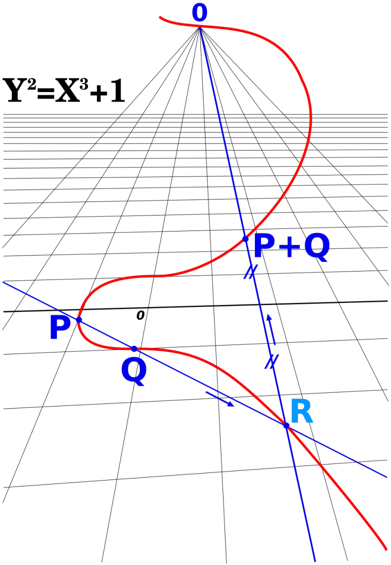
$E(\mathbb{R})$ heeft géén algebraïsche
parametrisering.

Maar $E(\mathbb{R})$ (inclusief O) heeft
een *binaire operatie*:

$$E(\mathbb{R}) \times E(\mathbb{R}) \to E(\mathbb{R}),$$

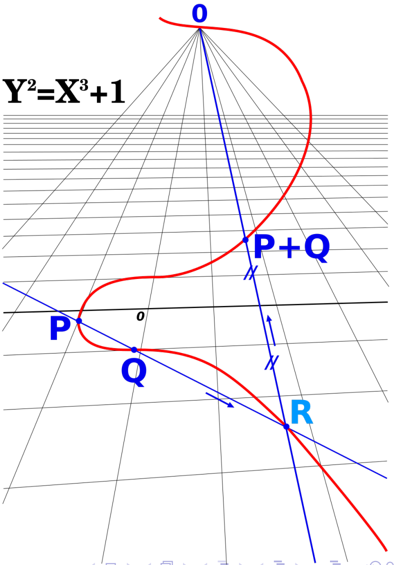
$$(P, Q) \mapsto P + Q.$$

(Maker van het plaatje: Jean Brette.)



Groepsstructuur

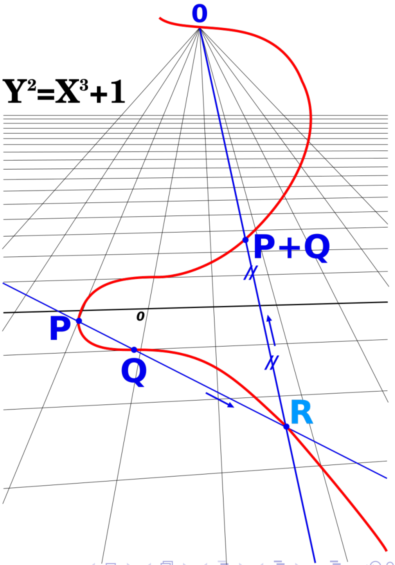
Deze binaire operatie maakt
van $E(\mathbb{R})$ een *commutatieve groep*.



Groepsstructuur

Deze binaire operatie maakt
van $E(\mathbb{R})$ een *commutatieve groep*.
Voor alle P en Q in $E(\mathbb{R})$:

$$Q + P = P + Q.$$



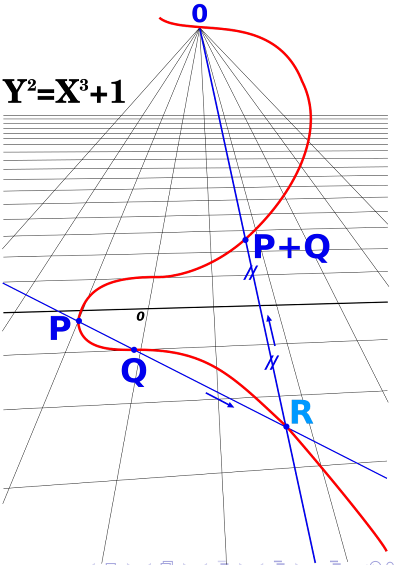
Groepsstructuur

Deze binaire operatie maakt van $E(\mathbb{R})$ een *commutatieve groep*.
Voor alle P en Q in $E(\mathbb{R})$:

$$Q + P = P + Q.$$

Voor alle P in $E(\mathbb{R})$:

$$P + O = P.$$



Groepsstructuur

Deze binaire operatie maakt van $E(\mathbb{R})$ een *commutatieve groep*.
Voor alle P en Q in $E(\mathbb{R})$:

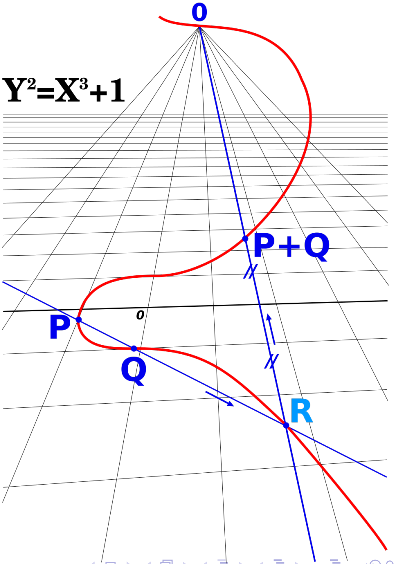
$$Q + P = P + Q.$$

Voor alle P in $E(\mathbb{R})$:

$$P + O = P.$$

Voor alle P is er een Q zodat:

$$P + Q = O.$$



Groepsstructuur

Deze binaire operatie maakt van $E(\mathbb{R})$ een *commutatieve groep*.
Voor alle P en Q in $E(\mathbb{R})$:

$$Q + P = P + Q.$$

Voor alle P in $E(\mathbb{R})$:

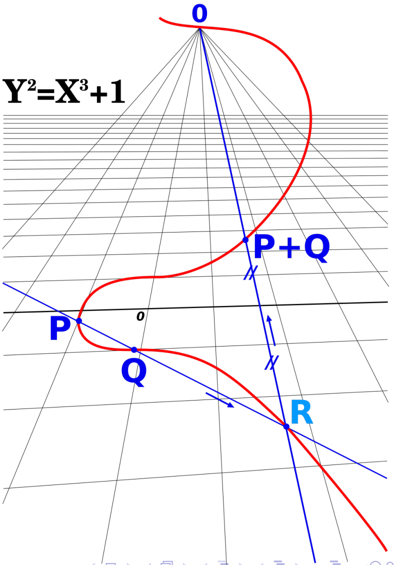
$$P + O = P.$$

Voor alle P is er een Q zodat:

$$P + Q = O.$$

Voor alle P, Q en R in $E(\mathbb{R})$:

$$P + (Q + R) = (P + Q) + R.$$



De groepsoperatie is algebraïsch

De coördinaten van $P + Q$ zijn te krijgen uit die van P en Q d.m.v. de operaties $+$, $-$, $\cdot$, $/$ en de coëfficienten van de vergelijking van E.

De groepsoperatie is algebraïsch

De coördinaten van $P + Q$ zijn te krijgen uit die van P en Q d.m.v. de operaties $+$, $-$, $\cdot$, $/$ en de coëfficienten van de vergelijking van E.

Dus we kunnen $\mathbb{R}$ vervangen door ieder getalsysteem met deze operaties, die aan de gebruikelijke eigenschappen voldoen: lichamen.

De groepsoperatie is algebraïsch

De coördinaten van $P + Q$ zijn te krijgen uit die van P en Q d.m.v. de operaties $+$, $-$, $\cdot$, $/$ en de coëfficienten van de vergelijking van E.

Dus we kunnen $\mathbb{R}$ vervangen door ieder getalsysteem met deze operaties, die aan de gebruikelijke eigenschappen voldoen: lichamen.

Voorbeelden: $\mathbb{Q}$ (rationale getallen), $\mathbb{C}$ (complexe getallen), maar ook *eindige lichamen*.

De groepsoperatie is algebraïsch

De coördinaten van $P + Q$ zijn te krijgen uit die van P en Q d.m.v. de operaties $+$, $-$, $\cdot$, $/$ en de coëfficienten van de vergelijking van E.

Dus we kunnen $\mathbb{R}$ vervangen door ieder getalsysteem met deze operaties, die aan de gebruikelijke eigenschappen voldoen: lichamen.

Voorbeelden: $\mathbb{Q}$ (rationale getallen), $\mathbb{C}$ (complexe getallen), maar ook *eindige lichamen*.

Het lichaam $\mathbb{F}_2 = \{0, 1\}$, met rekenen met resten na deling door 2:

$$1 + 1 = 0, \quad \text{want “oneven plus oneven is even”.}$$

De lichamen $\mathbb{F}_p$

Laat p een priemgetal zijn. Dan noteren we $\mathbb{F}_p = \{0, \ldots, p-1\}$.

De lichamen $\mathbb{F}_p$

Laat p een priemgetal zijn. Dan noteren we $\mathbb{F}_p = \{0, \ldots, p-1\}$.

We definiëren optelling en vermenigvuldiging op $\mathbb{F}_p$ door dezelfde operatie in $\mathbb{Z}$ te doen, en vervolgens de rest te nemen na deling door p.

De lichamen $\mathbb{F}_p$

Laat p een priemgetal zijn. Dan noteren we $\mathbb{F}_p = \{0, \ldots, p-1\}$.

We definiëren optelling en vermenigvuldiging op $\mathbb{F}_p$ door dezelfde operatie in $\mathbb{Z}$ te doen, en vervolgens de rest te nemen na deling door p.

$$
\begin{aligned}
\mathbb{F}_p \times \mathbb{F}_p &\to \mathbb{Z} \to \mathbb{F}_p \\
(a, b) &\mapsto a + b \mapsto (a + b) \bmod p \\
(a, b) &\mapsto a \cdot b \mapsto (a \cdot b) \bmod p.
\end{aligned}
$$

De lichamen $\mathbb{F}_p$

Laat p een priemgetal zijn. Dan noteren we $\mathbb{F}_p = \{0, \ldots, p-1\}$.

We definiëren optelling en vermenigvuldiging op $\mathbb{F}_p$ door dezelfde operatie in $\mathbb{Z}$ te doen, en vervolgens de rest te nemen na deling door p.

$$
\begin{aligned}
&\mathbb{F}_p \times \mathbb{F}_p \to \mathbb{Z} \to \mathbb{F}_p \\
&\quad (a, b) \mapsto a + b \mapsto (a + b) \bmod p \\
&\quad (a, b) \mapsto a{\cdot}b \mapsto (a{\cdot}b) \bmod p.
\end{aligned}
$$

Voorbeeld: in $\mathbb{F}_{13}$ hebben we $10 + 7 = 17 \bmod 13 = 4$, en ook $10{\cdot}7 = 70 \bmod 13 = 5$.

De lichamen $\mathbb{F}_p$

Laat p een priemgetal zijn. Dan noteren we $\mathbb{F}_p = \{0, \ldots, p-1\}$.

We definiëren optelling en vermenigvuldiging op $\mathbb{F}_p$ door dezelfde operatie in $\mathbb{Z}$ te doen, en vervolgens de rest te nemen na deling door p.

$$\begin{aligned} \mathbb{F}_p \times \mathbb{F}_p &\to \mathbb{Z} \to \mathbb{F}_p \\ (a, b) &\mapsto a + b \mapsto (a + b) \bmod p \\ (a, b) &\mapsto a \cdot b \mapsto (a \cdot b) \bmod p. \end{aligned}$$

Voorbeeld: in $\mathbb{F}_{13}$ hebben we $10 + 7 = 17 \bmod 13 = 4$, en ook $10 \cdot 7 = 70 \bmod 13 = 5$.

Deze optelling en vermenigvuldiging hebben de gebruikelijke eigenschappen, en omdat p een priemgetal is, kunnen we ook delen door elke $a \neq 0$ in $\mathbb{F}_p$.

De lichamen $\mathbb{F}_p$

Laat p een priemgetal zijn. Dan noteren we $\mathbb{F}_p = \{0, \ldots, p{-}1\}$.

We definiëren optelling en vermenigvuldiging op $\mathbb{F}_p$ door dezelfde operatie in $\mathbb{Z}$ te doen, en vervolgens de rest te nemen na deling door p.

$$\begin{aligned} \mathbb{F}_p \times \mathbb{F}_p &\to \mathbb{Z} \to \mathbb{F}_p \\ (a, b) &\mapsto a + b \mapsto (a + b) \bmod p \\ (a, b) &\mapsto a{\cdot}b \mapsto (a{\cdot}b) \bmod p. \end{aligned}$$

Voorbeeld: in $\mathbb{F}_{13}$ hebben we $10 + 7 = 17 \bmod 13 = 4$, en ook $10{\cdot}7 = 70 \bmod 13 = 5$.

Deze optelling en vermenigvuldiging hebben de gebruikelijke eigenschappen, en omdat p een priemgetal is, kunnen we ook delen door elke $a \neq 0$ in $\mathbb{F}_p$.

In $\mathbb{F}_{13}$ hebben we $1/2 = 7$ (want $2{\cdot}7 = 14 = 1 + 13$), $1/3 = 9$ (want $3{\cdot}9 = 27 = 1 + 2{\cdot}13$), etc.

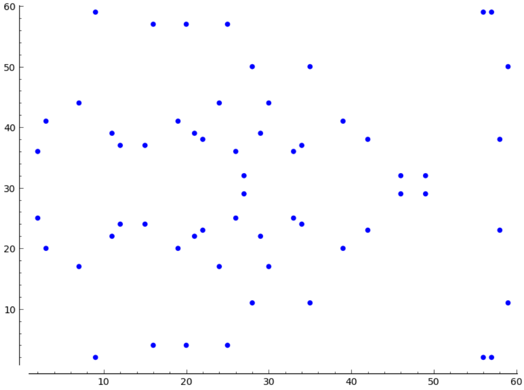
Elliptische krommen over $\mathbb{F}_p$

Voor p priem en E een elliptische kromme $y^2 = x^3 + ax + b$ met a en b in $\mathbb{F}_p$ is $E(\mathbb{F}_p)$ een eindige commutatieve groep, waarvan het aantal elementen $\#E(\mathbb{F}_p)$ efficiënt is uit te rekenen (René Schoof, 1983).

Elliptische krommen over $\mathbb{F}_p$

Voor p priem en E een elliptische kromme $y^2 = x^3 + ax + b$ met a en b in $\mathbb{F}_p$ is $E(\mathbb{F}_p)$ een eindige commutatieve groep, waarvan het aantal elementen $\#E(\mathbb{F}_p)$ efficiënt is uit te rekenen (René Schoof, 1983).
Voorbeeld: de elliptische kromme $y^2 = x^3 + 7$ over $\mathbb{F}_{61}$:

$E(\mathbb{F}_{61}) =$



$\cup \{O\}$

Bitcoins elliptische kromme

Bitcoin gebruikt $p = 2^{256} - 2^{32} - 2^9 - 2^8 - 2^7 - 2^6 - 2^4 - 1$, $a = 0$ en $b = 7$.

Bitcoins elliptische kromme

Bitcoin gebruikt $p = 2^{256} - 2^{32} - 2^9 - 2^8 - 2^7 - 2^6 - 2^4 - 1$, $a = 0$ en $b = 7$.

$$
\begin{aligned}
p = & 11579208923731619542357098500868790785 \\
& 3269984665640564039457584007908834671663.
\end{aligned}
$$

Bitcoins elliptische kromme

Bitcoin gebruikt $p = 2^{256} - 2^{32} - 2^9 - 2^8 - 2^7 - 2^6 - 2^4 - 1$, $a = 0$ en $b = 7$.

$$
\begin{aligned}
p = &115792089237316195423570985008687907853 \\
&269984665640564039457584007908834671663.
\end{aligned}
$$

Dit is "secp256k1" van de Standards for Efficient Cryptography Group, een internationaal consortium opgericht in 1998 om commerciële standaarden voor efficiënte ECC te ontwikkelen.

Bitcoins elliptische kromme

Bitcoin gebruikt $p = 2^{256} - 2^{32} - 2^9 - 2^8 - 2^7 - 2^6 - 2^4 - 1$, $a = 0$ en $b = 7$.

$$\begin{aligned} p =& 115792089237316195423570985008687907853 \\ & 269984665640564039457584007908834671663. \end{aligned}$$

Dit is "secp256k1" van de Standards for Efficient Cryptography Group, een internationaal consortium opgericht in 1998 om commerciële standaarden voor efficiënte ECC te ontwikkelen.

$$\begin{aligned} n := \#E(\mathbb{F}_p) =& 115792089237316195423570985008687907852 \\ & 837564279074904382605163141518161494337. \end{aligned}$$

Bitcoins elliptische kromme

Bitcoin gebruikt $p = 2^{256} - 2^{32} - 2^9 - 2^8 - 2^7 - 2^6 - 2^4 - 1$, $a = 0$ en $b = 7$.

$$
\begin{aligned}
p = & 115792089237316195423570985008687907853 \\
& 269984665640564039457584007908834671663.
\end{aligned}
$$

Dit is “secp256k1” van de Standards for Efficient Cryptography Group, een internationaal consortium opgericht in 1998 om commerciële standaarden voor efficiënte ECC te ontwikkelen.

$$
\begin{aligned}
n := \#E(\mathbb{F}_p) = & 115792089237316195423570985008687907852 \\
& 837564279074904382605163141518161494337.
\end{aligned}
$$

Dit aantal n is ook een priemgetal, en dat is belangrijk. We hebben dus ook het eindige lichaam $\mathbb{F}_n$. In secp256k1 is ook een $G \neq O$ in $E(\mathbb{F}_p)$ gespecificeerd.

ECDSA: Sleutelparen

Laat p, E, en G als boven. Dan is de afbeelding

$$\mathbb{F}_n \to E(\mathbb{F}_p), \quad d \mapsto d{\cdot}G = G + \cdots + G \quad (d \text{ termen})$$

bijectief: voor iedere Q in $E(\mathbb{F}_p)$ is er precies één d in $\mathbb{F}_n$ met $d{\cdot}G = Q$.

ECDSA: Sleutelparen

Laat p, E, en G als boven. Dan is de afbeelding

$$\mathbb{F}_n \to E(\mathbb{F}_p), \quad d \mapsto d{\cdot}G = G + \cdots + G \quad (d \text{ termen})$$

bijectief: voor iedere Q in $E(\mathbb{F}_p)$ is er precies één d in $\mathbb{F}_n$ met $d{\cdot}G = Q$.

Alice maakt een geheime sleutel door een random element d_A in $\mathbb{F}_n$ te kiezen.

ECDSA: Sleutelparen

Laat p, E, en G als boven. Dan is de afbeelding

$$\mathbb{F}_n \to E(\mathbb{F}_p), \quad d \mapsto d{\cdot}G = G + \cdots + G \quad (d \text{ termen})$$

bijectief: voor iedere Q in $E(\mathbb{F}_p)$ is er precies één d in $\mathbb{F}_n$ met $d{\cdot}G = Q$.

Alice maakt een geheime sleutel door een random element d_A in $\mathbb{F}_n$ te kiezen.

Haar publieke sleutel is dan het punt $Q_A := d_A{\cdot}G$ in $E(\mathbb{F}_p)$. Dit kan snel worden uitgerekend: ongeveer 500 rekenstappen.

ECDSA: Sleutelparen

Laat p, E, en G als boven. Dan is de afbeelding

$$\mathbb{F}_n \to E(\mathbb{F}_p), \quad d \mapsto d{\cdot}G = G + \cdots + G \quad (d \text{ termen})$$

bijectief: voor iedere Q in $E(\mathbb{F}_p)$ is er precies één d in $\mathbb{F}_n$ met $d{\cdot}G = Q$.

Alice maakt een geheime sleutel door een random element d_A in $\mathbb{F}_n$ te kiezen.

Haar publieke sleutel is dan het punt $Q_A := d_A{\cdot}G$ in $E(\mathbb{F}_p)$. Dit kan snel worden uitgerekend: ongeveer 500 rekenstappen.

Het snelst nu bekende algoritme om uit Q_A weer d_A te bepalen kost ongeveer $2^{128} \cong \sqrt{n}$ rekenstappen. Dat is al 30 jaar zo. Dit kost 1079028307080 jaar met een botnet van 10^9 computers van elk 10 GHz.

ECDSA: Sleutelparen

Laat p, E, en G als boven. Dan is de afbeelding

$$\mathbb{F}_n \to E(\mathbb{F}_p), \quad d \mapsto d{\cdot}G = G + \cdots + G \quad (d \text{ termen})$$

bijectief: voor iedere Q in $E(\mathbb{F}_p)$ is er precies één d in $\mathbb{F}_n$ met $d{\cdot}G = Q$.

Alice maakt een geheime sleutel door een random element d_A in $\mathbb{F}_n$ te kiezen.

Haar publieke sleutel is dan het punt $Q_A := d_A{\cdot}G$ in $E(\mathbb{F}_p)$. Dit kan snel worden uitgerekend: ongeveer 500 rekenstappen.

Het snelst nu bekende algoritme om uit Q_A weer d_A te bepalen kost ongeveer $2^{128} \cong \sqrt{n}$ rekenstappen. Dat is al 30 jaar zo. Dit kost 1079028307080 jaar met een botnet van 10^9 computers van elk 10 GHz.

Die aanval is gebaseerd op de “birthday paradox”. Het is een “generiek” algoritme: maakt geen gebruik van specifieke eigenschappen van elliptische krommen.

ECDSA: ondertekenen

Alice heeft nu een boodschap m, zeg een string karakters, die ze wil ondertekenen.

ECDSA: ondertekenen

Alice heeft nu een boodschap m, zeg een string karakters, die ze wil ondertekenen.

Dan kiest ze een random k in $\mathbb{F}_n$, en berekent r en s in $\mathbb{F}_n$ gedefinieerd door:

$$r = f(x(k \cdot G)), \quad \text{waar} \quad f\colon \mathbb{F}_p \to \mathbb{F}_n, \quad a \mapsto a \bmod n,$$

$$s = k^{-1} \cdot (h(m) + d_A \cdot r), \quad \text{waar} \quad h(m) = H(m) \bmod n,$$

hierbij is H de hash-functie SHA-256, twee keer toegepast. (Meer over hash-functies in de volgende twee presentaties.)

ECDSA: ondertekenen

Alice heeft nu een boodschap m, zeg een string karakters, die ze wil ondertekenen.

Dan kiest ze een random k in $\mathbb{F}_n$, en berekent r en s in $\mathbb{F}_n$ gedefinieerd door:

$$r = f(x(k \cdot G)), \quad \text{waar} \quad f \colon \mathbb{F}_p \to \mathbb{F}_n, \quad a \mapsto a \bmod n,$$

$$s = k^{-1} \cdot (h(m) + d_A \cdot r), \quad \text{waar} \quad h(m) = H(m) \bmod n,$$

hierbij is H de hash-functie SHA-256, twee keer toegepast. (Meer over hash-functies in de volgende twee presentaties.)

De handtekening is dan (r, s).

ECDSA: ondertekenen

Alice heeft nu een boodschap m, zeg een string karakters, die ze wil ondertekenen.

Dan kiest ze een random k in $\mathbb{F}_n$, en berekent r en s in $\mathbb{F}_n$ gedefinieerd door:

$$r = f(x(k \cdot G)), \quad \text{waar} \quad f\colon \mathbb{F}_p \to \mathbb{F}_n, \quad a \mapsto a \bmod n,$$

$$s = k^{-1} \cdot (h(m) + d_A \cdot r), \quad \text{waar} \quad h(m) = H(m) \bmod n,$$

hierbij is H de hash-functie SHA-256, twee keer toegepast. (Meer over hash-functies in de volgende twee presentaties.)

De handtekening is dan (r, s).

Zwakte: $-k$ geeft $(r, -s)$ ("existential forgery").

ECDSA: ondertekenen

Alice heeft nu een boodschap m, zeg een string karakters, die ze wil ondertekenen.

Dan kiest ze een random k in $\mathbb{F}_n$, en berekent r en s in $\mathbb{F}_n$ gedefinieerd door:

$$r = f(x(k\cdot G)), \quad \text{waar} \quad f\colon \mathbb{F}_p \to \mathbb{F}_n, \quad a \mapsto a \bmod n,$$

$$s = k^{-1}\cdot(h(m) + d_A\cdot r), \quad \text{waar} \quad h(m) = H(m) \bmod n,$$

hierbij is H de hash-functie SHA-256, twee keer toegepast. (Meer over hash-functies in de volgende twee presentaties.)

De handtekening is dan (r, s).

Zwakte: $-k$ geeft $(r, -s)$ (“existential forgery”).

Oplossing: eis dat $s < n/2$.

ECDSA: ondertekenen

Alice heeft nu een boodschap m, zeg een string karakters, die ze wil ondertekenen.

Dan kiest ze een random k in $\mathbb{F}_n$, en berekent r en s in $\mathbb{F}_n$ gedefinieerd door:

$$r = f(x(k{\cdot}G)), \quad \text{waar} \quad f\colon \mathbb{F}_p \to \mathbb{F}_n, \quad a \mapsto a \bmod n,$$

$$s = k^{-1}{\cdot}(h(m) + d_A{\cdot}r), \quad \text{waar} \quad h(m) = H(m) \bmod n,$$

hierbij is H de hash-functie SHA-256, twee keer toegepast. (Meer over hash-functies in de volgende twee presentaties.)

De handtekening is dan (r, s).

Zwakte: $-k$ geeft $(r, -s)$ ("existential forgery").

Oplossing: eis dat $s < n/2$.

Over veiligheid en betekenis van zo'n handtekening straks meer.

ECDSA: controle van de handtekening

Voor willekeurige r, s, en k in $\mathbb{F}_n - \{0\}$ zijn equivalent:

- $s = k^{-1} \cdot (h(m) + d_A \cdot r)$,
- $k = s^{-1} \cdot (h(m) + d_A \cdot r)$,
- $k \cdot G = s^{-1} \cdot (h(m) + d_A \cdot r) \cdot G$,
- $k \cdot G = s^{-1} \cdot (h(m) \cdot G + r \cdot Q_A)$,

en deze laatste gelijkheid impliceert:

$$f(x(k \cdot G)) = f(x(s^{-1} \cdot (h(m) \cdot G + r \cdot Q_A))).$$

De vraag is dus of voor het paar (r, s) van Alice r (gedefinieerd als $f(x(k \cdot G)))$ en $f(x(s^{-1} \cdot (h(m) \cdot G + r \cdot Q_A)))$ gelijk zijn. De input voor deze berekening bestaat uit publieke data.

ECDSA: veiligheid

Het best nu bekende algoritme om handtekeningen als boven (256 bit ECDSA) te vervalsen kost ongeveer 2^{128} rekenstappen.

ECDSA: veiligheid

Het best nu bekende algoritme om handtekeningen als boven (256 bit ECDSA) te vervalsen kost ongeveer 2^{128} rekenstappen.

Er is geen wiskundig bewijs dat er geen snellere aanvallen zijn.

ECDSA: veiligheid

Het best nu bekende algoritme om handtekeningen als boven (256 bit ECDSA) te vervalsen kost ongeveer 2^{128} rekenstappen.

Er is geen wiskundig bewijs dat er geen snellere aanvallen zijn.

Cryptologen hebben praktische digitale handtekeningen ontwikkeld waarvan de veiligheid wel bewezen is (onder onbewezen “standaard cryptografische aannames”), zelfs tegen quantumcomputers.

ECDSA: veiligheid

Het best nu bekende algoritme om handtekeningen als boven (256 bit ECDSA) te vervalsen kost ongeveer 2^{128} rekenstappen.

Er is geen wiskundig bewijs dat er geen snellere aanvallen zijn.

Cryptologen hebben praktische digitale handtekeningen ontwikkeld waarvan de veiligheid wel bewezen is (onder onbewezen “standaard cryptografische aannames”), zelfs tegen quantumcomputers.

Onder dat soort aannamen kunnen we dus vertrouwen hebben in:

ECDSA: veiligheid

Het best nu bekende algoritme om handtekeningen als boven (256 bit ECDSA) te vervalsen kost ongeveer 2^{128} rekenstappen.

Er is geen wiskundig bewijs dat er geen snellere aanvallen zijn.

Cryptologen hebben praktische digitale handtekeningen ontwikkeld waarvan de veiligheid wel bewezen is (onder onbewezen “standaard cryptografische aannames”), zelfs tegen quantumcomputers.

Onder dat soort aannamen kunnen we dus vertrouwen hebben in:

authenticiteit, Alice heeft ondertekend, of haar geheime sleutel is uitgelekt,

ECDSA: veiligheid

Het best nu bekende algoritme om handtekeningen als boven (256 bit ECDSA) te vervalsen kost ongeveer 2^{128} rekenstappen.

Er is geen wiskundig bewijs dat er geen snellere aanvallen zijn.

Cryptologen hebben praktische digitale handtekeningen ontwikkeld waarvan de veiligheid wel bewezen is (onder onbewezen “standaard cryptografische aannames”), zelfs tegen quantumcomputers.

Onder dat soort aannamen kunnen we dus vertrouwen hebben in:

authenticiteit, Alice heeft ondertekend, of haar geheime sleutel is uitgelekt,

integriteit, wat ze ondertekende is precies *m*, de boodschap is niet veranderd,

ECDSA: veiligheid

Het best nu bekende algoritme om handtekeningen als boven (256 bit ECDSA) te vervalsen kost ongeveer 2^{128} rekenstappen.

Er is geen wiskundig bewijs dat er geen snellere aanvallen zijn.

Cryptologen hebben praktische digitale handtekeningen ontwikkeld waarvan de veiligheid wel bewezen is (onder onbewezen “standaard cryptografische aannames”), zelfs tegen quantumcomputers.

Onder dat soort aannamen kunnen we dus vertrouwen hebben in:

authenticiteit, Alice heeft ondertekend, of haar geheime sleutel is uitgelekt,

integriteit, wat ze ondertekende is precies m, de boodschap is niet veranderd,

onweerlegbaarheid, Alice kan haar handtekening niet ontkennen, tenzij ze haar geheime sleutel herroepen had vóór de tijd van ondertekening.

Praktijk, conclusies

- De gebruiker van Bitcoin heeft programma's (apps) die de nodige berekeningen doen.

Praktijk, conclusies

- De gebruiker van Bitcoin heeft programma's (apps) die de nodige berekeningen doen.
- Omdat het openbronprogramma's zijn kunnen we de werking ervan controleren.

Praktijk, conclusies

- De gebruiker van Bitcoin heeft programma's (apps) die de nodige berekeningen doen.
- Omdat het openbronprogramma's zijn kunnen we de werking ervan controleren.
- Bitcoin's digitale handtekeningen berusten op eeuwenoude wiskundige concepten (elliptische krommen, eindige lichamen).

Praktijk, conclusies

- De gebruiker van Bitcoin heeft programma's (apps) die de nodige berekeningen doen.
- Omdat het openbronprogramma's zijn kunnen we de werking ervan controleren.
- Bitcoin's digitale handtekeningen berusten op eeuwenoude wiskundige concepten (elliptische krommen, eindige lichamen).
- Al 30 jaar is er geen betere aanval dan de generieke.

Praktijk, conclusies

- De gebruiker van Bitcoin heeft programma's (apps) die de nodige berekeningen doen.
- Omdat het openbronprogramma's zijn kunnen we de werking ervan controleren.
- Bitcoin's digitale handtekeningen berusten op eeuwenoude wiskundige concepten (elliptische krommen, eindige lichamen).
- Al 30 jaar is er geen betere aanval dan de generieke.
- De correctheid van ECDSA is gegarandeerd (wiskundige stellingen, op niveau van een masteropleiding in algebra, meetkunde en getaltheorie).

Praktijk, conclusies

- De gebruiker van Bitcoin heeft programma's (apps) die de nodige berekeningen doen.
- Omdat het openbronprogramma's zijn kunnen we de werking ervan controleren.
- Bitcoin's digitale handtekeningen berusten op eeuwenoude wiskundige concepten (elliptische krommen, eindige lichamen).
- Al 30 jaar is er geen betere aanval dan de generieke.
- De correctheid van ECDSA is gegarandeerd (wiskundige stellingen, op niveau van een masteropleiding in algebra, meetkunde en getaltheorie).
- Er zijn nu digitale handtekeningen waarvan veiligheid onder duidelijke aannamen is bewezen (wiskundige stellingen van cryptologen, op onderzoeksniveau).

Praktijk, conclusies

- De gebruiker van Bitcoin heeft programma's (apps) die de nodige berekeningen doen.
- Omdat het openbronprogramma's zijn kunnen we de werking ervan controleren.
- Bitcoin's digitale handtekeningen berusten op eeuwenoude wiskundige concepten (elliptische krommen, eindige lichamen).
- Al 30 jaar is er geen betere aanval dan de generieke.
- De correctheid van ECDSA is gegarandeerd (wiskundige stellingen, op niveau van een masteropleiding in algebra, meetkunde en getaltheorie).
- Er zijn nu digitale handtekeningen waarvan veiligheid onder duidelijke aannamen is bewezen (wiskundige stellingen van cryptologen, op onderzoeksniveau).
- Expertise in Nederland. CWI (Cramer), RUN (Bosma, Verheul), TUe (Bernstein, Lange, Schoenmakers, de Weger), UL (Edixhoven, Lenstra, Stevenhagen).