An Introduction to Elliptic Curves and the Computation of Cryptographic Pairings

Craig Costello

Technische Universiteit Eindhoven

October 28, 2012

ECC2012 - Querétaro, Mexico

Why ECC is awesome. . .

- Why ECC (elliptic curve cryptography) is awesome. . .
  - It's faster, more compact and more elegant than other public-key crypto. settings
  - It brings algebraic/arithmetic geometry and number theory to life - these things have real-world importance!
  - **It's more interesting & fun than other crypto. settings**
- Why ECC (this conference) is awesome. . .
  - It brings some of ECC's biggest experts to you!
  - The co-inventors of ECC are both here!
  - **It's more interesting & fun than other crypto. conferences**

This lecture is

1. **. . . for students & newcomers**
2. **. . . slow moving:** I will assume you have not seen ECC before: therefore this talk will be elementary and (intentionally) slow-moving
3. **. . . example driven:** what I lack in formality and completeness, I make up for by referring you to Ben Smith's excellent "Useful stuff" intro from ECC2011: `http://ecc2011.loria.fr/slides/summerschool-smith.pdf`
4. **. . . accompanied by pictures:** what I lack in Spanish, I will make up for in pictures
5. **. . . accompanied by Magma:** I will be working alongside examples in Magma (all examples/code hyperlinked from my thesis)

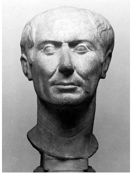
Overview

1. Motivation
2. Elliptic curves are groups
3. Elliptic curves as cryptographic groups
4. Divisors
5. A very brief look at pairings

1. Motivation

Private-key vs. Public-key cryptography

BC - WWII:



Caesar

Mary, Queen of Scots

Enigma Code

must communicate beforehand

1970's:

Diffie-Hellman-Merkle

Rivest-Shamir-Adleman (RSA)

Ellis-Cocks-Williamson

BREAKTHROUGH: no need for prior communication!!!

Diffie-Hellman (Merkle): a toy example

Public values:

$q = 1000000000000061$ (prime), $g = 832022676086941$ (generator of $\mathbb{Z}_q$).

Secret values:

Alice's secret: a=4275315603725493

Alice computes (public key):

$g^a \bmod q = 9213047582249495$

Bob can compute:

Bob's secret: b=1083333300180813

Bob computes (public key):

$g^b \bmod q = 9893308140872135$

Alice can compute:

$$9893308140872135^a = 8817060794020263 = 9213047582249495^b$$

$$= g^{ab}$$

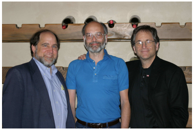
Secret keys safe as long as discrete log problem (DLP) is hard

Joint secret safe as long as Diffie-Hellman problem is hard

Modulus (key) sizes: then and now

1970's:   

$q =$ 1606938044258990275541962092341162602522202993782792835301301.
(200-bit prime)

NOW:   

$q =$
179769313486231590772930519078902473361797697894230657273430081157732675805500963132708477322407536021 1
201138798713933576587897688144166224928474306394741243777678934248654852763022196012460941194530829520 8
500576883815068234246288147391311054082723716335051068458629823994724593847971630483535632962422413711 1.
(1024-bit prime)

Elliptic curves cryptography (ECC)

mid 1980's:

Neal Koblitz

Victor Miller

Use elliptic curve (more abstract) groups instead!

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

Subexponential attacks on standard groups don't apply anymore!!!

$q =$
1797693134862315907729305190789024733617976978942306572734300811577326758055009631327084773224075360211
2011387987139335765878976881441662249284743063947412437776789342486548527630221960124609411945308295208
5005768838150682342462881473913110540827237163350510684586298239947245938479716304835356329624224137111.
(1024-bit prime)

vs.

$q = 1461501637330902918203684832716283019655932542929$
(160-bit prime)

2. Elliptic curves are groups

Abelian groups

Recall the definition of an *abelian group*:

Group (definition)

A group G is a set with an operation $+$ that combines any two elements to form a third element, satisfying four axioms:

- 1. **Closure** - $a, b \in G$ implies $a + b \in G$
- 2. **Associativity** - $(a + b) + c = a + (b + c)$ for $a, b, c \in G$
- 3. **Identity** - unique $e \in G$ such that $a + e = e + a = a$
- 4. **Inverses** - for every $a \in G$, there exists a unique element b such that $a + b = b + a = e$

If, in addition, $c + d = d + c$ (always), then G is said to be **abelian**.

Cubic equations

- Two roots of a cubic polynomial imply the third root
- If α, β are roots of $a_3x^3 + a_2x^2 + a_1x + a_0 = 0$, then the third root is ...

Cubic equations

- Two roots of a cubic polynomial imply the third root
- If α, β are roots of $a_3x^3 + a_2x^2 + a_1x + a_0 = 0$, then the third root is ... $\gamma = a_0/(a_3\alpha\beta)$, since $a_3(x-\alpha)(x-\beta)(x-\gamma) = 0$
- Roughly speaking: elliptic curves are groups that make use of this... more formally...

Bezout's theorem

Two curves with degrees m and n intersect mn times.

Cubic equations

- Two roots of a cubic polynomial imply the third root
- If α, β are roots of $a_3x^3 + a_2x^2 + a_1x + a_0 = 0$, then the third root is ... $\gamma = a_0/(a_3\alpha\beta)$, since $a_3(x-\alpha)(x-\beta)(x-\gamma) = 0$
- Roughly speaking: elliptic curves are groups that make use of this... more formally...

Bezout's theorem (special case - all we need)

Two curves with degrees 3 and 1 intersect 3 times.

Cubic equations

- Two roots of a cubic polynomial imply the third root
- If α, β are roots of $a_3x^3 + a_2x^2 + a_1x + a_0 = 0$, then the third root is ... $\gamma = a_0/(a_3\alpha\beta)$, since $a_3(x-\alpha)(x-\beta)(x-\gamma) = 0$
- Roughly speaking: elliptic curves are groups that make use of this... more formally...

Bezout's theorem (special case - all we need)

Two curves with degrees 3 and 1 intersect 3 times.

Given $P = (x_P, y_P)$ and $Q = (x_Q, y_Q)$ on a cubic curve, the line between them intersects the curve once more
This is what we use!

Cubic equation $\longrightarrow$ short Weierstrass equation

- General cubic curve (defined over field K)

$$\begin{aligned} C/K: \quad & a_9x^3 + a_8x^2y + a_7xy^2 + a_6y^3 + a_5x^2 \\ & + a_4xy + a_3y^2 + a_2x + a_1y + a_0 = 0 \end{aligned}$$

...after some manipulation (left as an exercise) assuming $\mathrm{Char}(K) \neq 2, 3$ ...

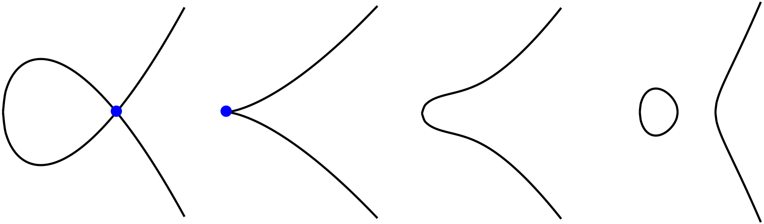
- Short Weierstrass equation (for elliptic curve over K)

$$E/K: \quad y^2 = x^3 + ax + b$$

- Defined over K if $a, b \in K$
- Points on E can be $(x, y) \in \bar{K} \times \bar{K}$

Elliptic curves: singular vs. smooth

- In $E/K : y^2 = x^3 + ax + b$, we need $4a^3 + 27b^2 \neq 0$ in K, or else things don't go "smoothly"



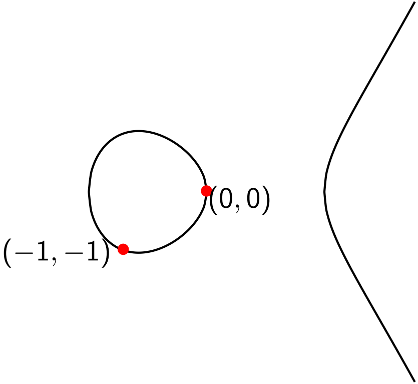
Singular curve $y^2 = x^3 - 3x + 2$ over $\mathbb{R}$.

Singular curve $y^2 = x^3$ over $\mathbb{R}$.

Smooth curve $y^2 = x^3 + x + 1$ over $\mathbb{R}$.

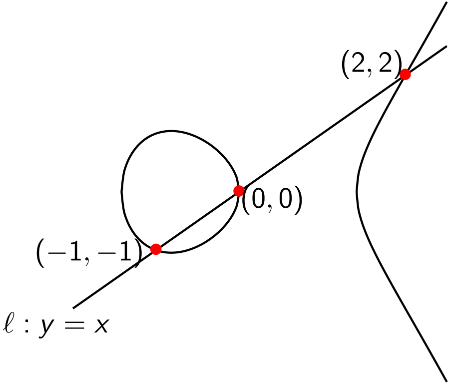
Smooth curve $y^2 = x^3 - x$ over $\mathbb{R}$.

Group law example: addition on $E/\mathbb{R} : y^2 = x^3 - 2x$



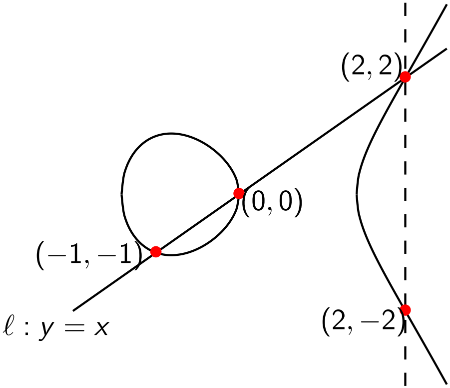
$E/\mathbb{R} : y^2 = x^3 - 2x$:

Group law example: addition on $E/\mathbb{R} : y^2 = x^3 - 2x$



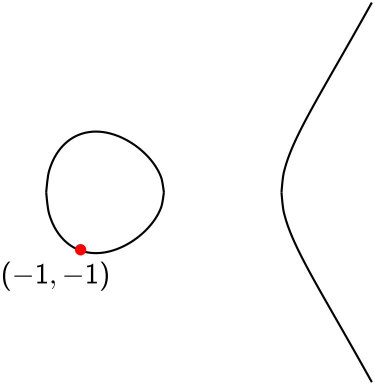
$E/\mathbb{R} : y^2 = x^3 - 2x$: addition.

Group law example: addition on $E/\mathbb{R} : y^2 = x^3 - 2x$



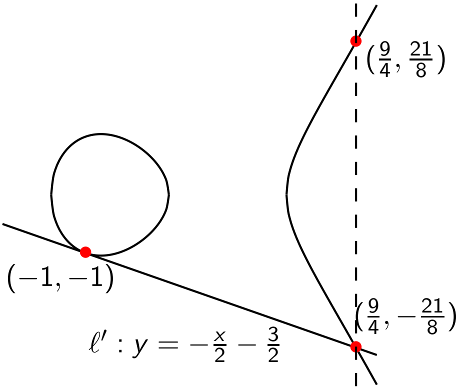
$E/\mathbb{R} : y^2 = x^3 - 2x$: addition.

Group law example: doubling on $E/\mathbb{R} : y^2 = x^3 - 2x$



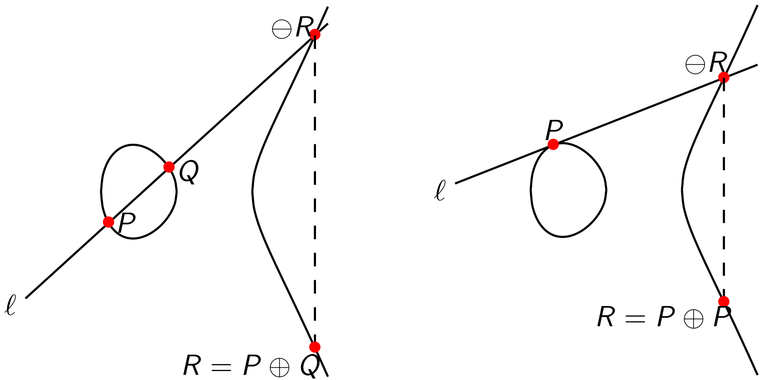
$E/\mathbb{R} : y^2 = x^3 - 2$: doubling.

Group law example: doubling on $E/\mathbb{R} : y^2 = x^3 - 2x$



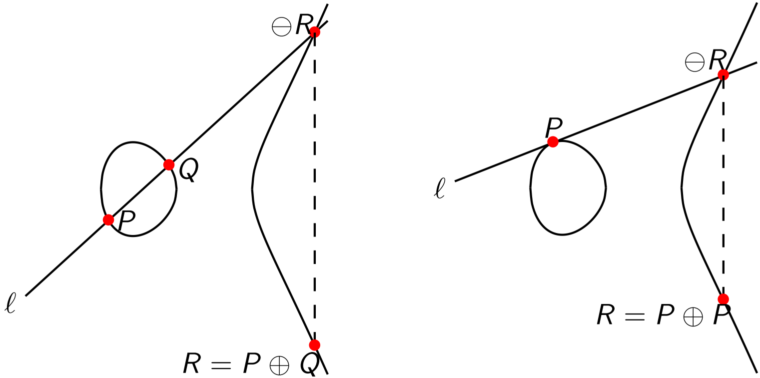
$E/\mathbb{R} : y^2 = x^3 - 2$: doubling.

Elliptic curve group law: addition and doubling



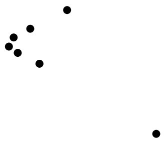
- Note: an elliptic curve is a group that is defined over a field
- Points form a group, but coordinates come from underlying field
- **Computing group operation requires field arithmetic** ...

Elliptic curve group law: addition and doubling

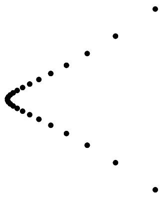


- Addition: $y = \lambda x + \nu, \quad \lambda = \frac{y_Q - y_P}{x_Q - x_P}, \quad \nu = y_P - \lambda x_P,$
  $x_R = \lambda^2 - x_P - x_Q, \quad y_R = -(\lambda x_R + \nu)$
- Doubling: same with $\lambda = \frac{3x_P^2 + a}{2y_P}$ and $x_P = x_Q$

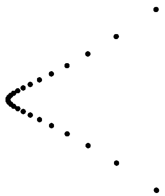
Example $E/\mathbb{Q} : y^2 = x^3 - 2$



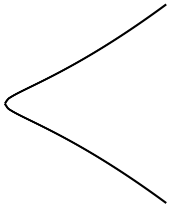
Of the first 10 multiples of $P = (3, 5)$ in $E(\mathbb{Q})$, 7 had $x < 6$.



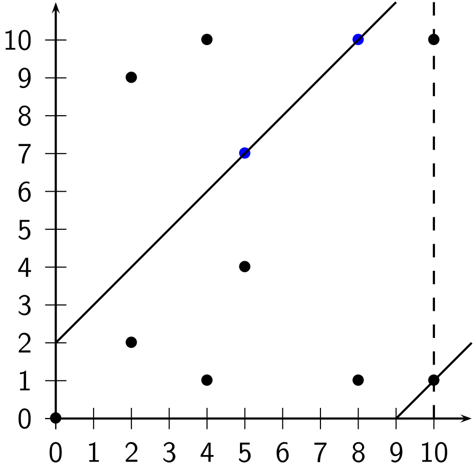
Of the first 100 multiples of $P = (3, 5)$ in $E(\mathbb{Q})$, 64 had $x < 6$.



Of the first 1000 multiples of $P = (3, 5)$ in $E(\mathbb{Q})$, 635 had $x < 6$.



$E : y^2 = x^3 - 2$ over $\mathbb{R}$.

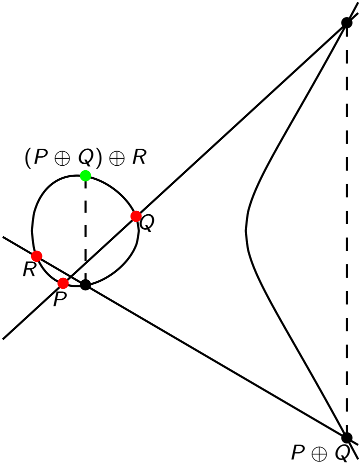


$E/\mathbb{F}_1 1 : y^2 = x^3 - 2x$: the points (excluding $\mathcal{O}$) on $E(\mathbb{F}_{11})$.

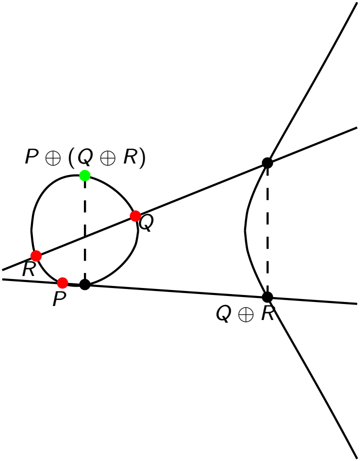
Recall: group law axioms

- **Closure**:
  if $P, Q \in E(K)$
  $\rightarrow$ cubic equation has coefficients in K
  $\rightarrow$ third root in K
  $\rightarrow$ $P + Q \in E(K)$
  $\rightarrow$ closed.
- What about **associativity**?
- What about the **identity**?
- What about **inverses**?
- Is it **abelian**?

Associativity: "proof" by picture



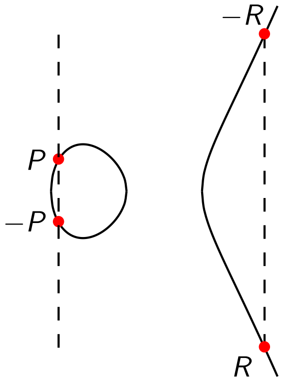
$(P \oplus Q) \oplus R$.



$P \oplus (Q \oplus R)$.

(real proof left as exercise)

Inverse and Identity: $\mathcal{O}$ in affine space?



- Besides all of the rational points $(x, y) \in \mathbb{A}^2(K)$, we need an additional point $\mathcal{O}$, the *point at infinity*
- Helpful in affine drawing to picture it as infinitely high/low, but formal definition requires *projective space* - another coordinate...

Homogeneous projective coordinates for E

- Substitute $x = X/Z$ and $y = Y/Z$ into $E : y^2 = x^3 + ax + b$
- Projective equation is $E_{\mathrm{proj}} : Y^2Z = X^3 + aXZ^2 + bZ^3$, coordinates written as $(X : Y : Z)$
- Notice all $(x, y) \in E$ correspond to $(\lambda X : \lambda Y : \lambda Z) \in E_{\mathrm{proj}}$ for $\lambda \in \bar{K}$
- **But** there is a point on E_{proj} that can't be scaled back to E???

Homogeneous projective coordinates for E

- Substitute $x = X/Z$ and $y = Y/Z$ into $E : y^2 = x^3 + ax + b$

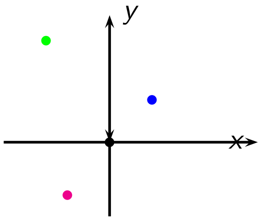
- Projective equation is $E_{\mathrm{proj}} : Y^2Z = X^3 + aXZ^2 + bZ^3$, coordinates written as $(X : Y : Z)$

- Notice all $(x, y) \in E$ correspond to $(\lambda X : \lambda Y : \lambda Z) \in E_{\mathrm{proj}}$ for $\lambda \in \bar{K}$

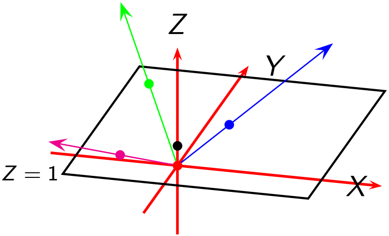
- **But** there is a point on E_{proj} that can't be scaled back to E???

- **This point is $\mathcal{O} = (0 : \lambda : 0)$ - the point at infinity**

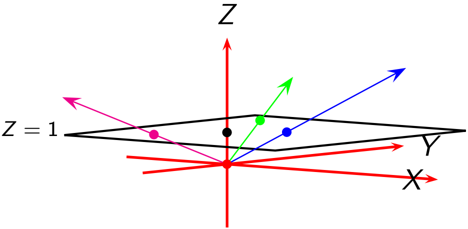
Projective space: points in $\mathbb{A}^2(K)$ are lines in $\mathbb{P}^2(K)$



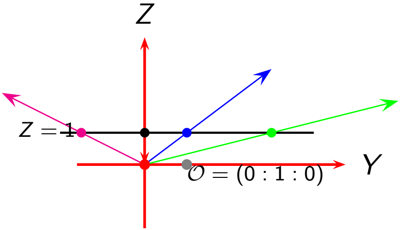
Three points in $\mathbb{A}^2(K)$.



Three lines in $\mathbb{P}^2(K)$.



Three lines in $\mathbb{P}^2(K)$.



Three lines in $\mathbb{P}^2(K)$.

Group law axioms

- **Closure**:
  if $P, Q \in E(K)$
  $\rightarrow$ cubic equation has coefficients in K
  $\rightarrow$ third root in K
  $\rightarrow$ $P + Q \in E(K)$
  $\rightarrow$ closed.

- **Associativity** - yes, "proof by picture", but see textbooks or try for yourself

- **Identity** - the point at infinity $\mathcal{O}$

- **Inverses** - inverse of (x, y) is $(x, -y)$

- **Abelian** - yes, line through P and Q is line through Q and P

Group law axioms

- **Closure**:
  if $P, Q \in E(K)$
  $\longrightarrow$ cubic equation has coefficients in K
  $\longrightarrow$ third root in K
  $\longrightarrow$ $P + Q \in E(K)$
  $\longrightarrow$ closed.
- **Associativity** - yes, "proof by picture", but see textbooks or try for yourself
- **Identity** - the point at infinity $\mathcal{O}$
- **Inverses** - inverse of (x, y) is $(x, -y)$
- **Abelian** - yes, line through P and Q is line through Q and P

Elliptic curves are groups!

3. Elliptic curves as cryptographic groups

Setting up ECDLP instances

- To set up discrete logarithm instances, we need to compute

$$[m]P = P + P + \cdots + P \qquad (m \ \mathit{times})$$

- m will be huge, so we need to *double-and-add* to compute $[m]P$ in $O(\log_2 m)$ steps

e.g. $m = 104143711012733238876513676535587592720823664060901595554869421344539731012577$

(1, 1, 1, 0, 0, 1, 1, 0, 0, 0, 1, 1, 1, 1, 1, 1, 0, 1, 0, 0, 0, 0, 0, 0, 0, 1, 1, 0, 0, 1, 1, 0, 0, 1, 1, 1, 0, 0, 0, 0, 0, 0, 1,
1, 1, 1, 0, 1, 0, 1, 0, 0, 0, 0, 1, 0, 0, 0, 0, 0, 0, 0, 1, 1, 1, 0, 1, 1, 0, 0, 0, 0, 0, 0, 0, 1, 1, 0, 0, 1, 0, 1, 0, 1, 1, 1, 1,
0, 1, 1, 1, 0, 0, 0, 0, 1, 1, 1, 1, 0, 0, 0, 0, 1, 0, 0, 0, 1, 0, 0, 0, 0, 1, 1, 0, 0, 0, 0, 0, 0, 0, 1, 0, 1, 0, 0, 0, 0, 1, 0, 1,
1, 0, 1, 1, 0, 1, 1, 1, 0, 1, 1, 0, 0, 0, 0, 0, 1, 0, 1, 0, 1, 1, 0, 0, 0, 1, 1, 1, 1, 0, 1, 0, 0, 1, 1, 0, 1, 1, 1, 0, 1, 0, 0, 1,
1, 1, 0, 1, 1, 1, 1, 0, 0, 1, 0, 0, 1, 0, 1, 0, 1, 1, 1, 1, 1, 0, 1, 0, 0, 1, 0, 0, 1, 0, 1, 1, 1, 0, 1, 1, 1, 0, 1, 0, 0, 1, 0, 0,
1, 1, 1, 0, 0, 0, 0, 0, 1, 1, 0, 1, 0, 0, 1, 1, 0, 1, 0, 1, 1, 1, 0, 1, 0, 1, 0, 1, 1, 1, 1, 1, 0, 0, 0, 0, 1)

Double-and-add takes 255 doublings and 123 additions

Interlude: Why ECC is awesome (cont.)

- Compare traditional groups to ECC at 128-bit security

Interlude: Why ECC is awesome (cont.)

- Compare traditional groups to ECC at 128-bit security

- ECDLP over 256-bit p

  $= 115792089210356248762697446949407573530086143415290314195533631308867097853951$

Interlude: Why ECC is awesome (cont.)

- Compare traditional groups to ECC at 128-bit security

- ECDLP over 256-bit p

  $= 115792089210356248762697446949407573530086143415290314195533631308867097853951$

- Comparable to standard DLP over $q = p^{12}$
  $= 580960597913810644809636622989416492519102906096473608804383010287470126087077\allowbreak6$
  6482173033611664754318259437411636363721864991002021782371074490715059944181840
  0342787085568400070101868479077691777913820328875711553963993872359271410692118
  2842047937244197120686781367972159048261418604511611344078747035121694997713540
  2837492929213422600429025184602648562538880748083950512261873985381670986780770
  3289556673190854870391629285162566732999470768100097360667042569028375009057813
  2879917770402824470487308666594740986238656937509695173630104358360328020157275
  7031519995321613484296864529039945826777471856903687985607353073750418292349157
  7048399629016421118853172422732137921315256777383769924837799393651892520114015
  5310447529182432257109321985968734700568576388269448211861140030742321384794381
  4986670503424178211639598575042459804527837974825281240063036698943378230288199
  846224571658305361842332438508243475140384917368602624 01

Secure vs. insecure curves: the importance of $\#E$

e.g. NIST-p256 curve (128-bit security)

Let $E/\mathbb{F}_p : y^2 = x^3 - 3x + b$

$$\begin{aligned} p &= 115792089210356248762697446949407573530086143415290314195533631308867097853951 \\ b &= 41058363725152142129326129780047268409114441015993725554835256314039467401291 \\ \#E &= 115792089210356248762697446949407573529996955224135760342422259061068512044369 \end{aligned}$$

$\#E = 256-$ bit prime ($\approx$ 128-bit security)

e.g. $b = 4$ instead

Let $E/\mathbb{F}_q : y^2 = x^3 - 3x + b$

$$\begin{aligned} q &= 115792089210356248762697446949407573530086143415290314195533631308867097853951 \\ b &= 4 \\ \#E &= 115792089210356248762697446949407573530301458765764575276748425375978192226668 \\ \#E &= 2^2 \cdot 13 \cdot 19 \cdot 179 \cdot 13003 \cdot 1307093479 \cdot 218034068577407083 \\ &\quad \cdot 16884307952548170257 \cdot 10464321644447000442097 \end{aligned}$$

$\#E$'s biggest prime factor is 74-bits (37-bit ~~security~~)

How many points on $E(\mathbb{F}_q)$?

- Hasse's bound for $\#E$, namely

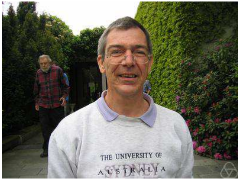
$$q + 1 - 2\sqrt{q} \leq \#E \leq q + 1 + 2\sqrt{q}$$

- e.g. take q from NISTp256

$$\begin{aligned}
q + 1 - \lfloor 2\sqrt{q} \rfloor &= 115792089210356248762697446949407573529405578681527665431107311373540212604928 \\
&\vdots \\
\#E(\text{good}) &= 115792089210356248762697446949407573529996955224135760342422259061068512044369 \\
&\vdots \\
q &= 115792089210356248762697446949407573530086143415290314195533631308867097853951 \\
&\vdots \\
\#E(\text{bad}) &= 115792089210356248762697446949407573530301458765764575276748425375978192226668 \\
&\vdots \\
q + 1 + \lfloor 2\sqrt{q} \rfloor &= 115792089210356248762697446949407573530766708149052962959959951244193983102976
\end{aligned}$$

- This offset of $\#E$ from $q + 1$ is called t - the trace of Frobenius, i.e. $\#E = q + 1 - t$, $|t| \leq 2\sqrt{q}$

Schoof's algorithm to find $\#E = q + 1 - t$



- Computing $\#E$ means computing the trace of Frobenius t
- Schoof's alg. computes t mod 3, t mod 5, t mod 7, ..., t mod ℓ such that $3 \cdot 5 \cdot 7 \cdot \cdots \cdot \ell > 4\sqrt{q}$
- ... *(skipping details for now)* ...
- Computes $\#E$ in $O((\log q)^8)$ (polynomial time)
- Makes ECC practical (also timely, invented in '85)

Torsion

- A point P is said to be in the r-torsion $E[r]$ of E, if it is killed by r, i.e. if $[r]P = \mathcal{O}$

e.g. Let $E/\mathbb{F}_{101} : y^2 = x^3 + x + 1$, $\#E = 105 = |\langle P \rangle|$, $P = (47, 12)$

- Lagrange's theorem: points in $\langle P \rangle$ will have order in $\{1, 3, 5, 7, 15, 21, 35, 105\}$.
- $[3]P = (27, 7) \in E[35]$
- $[7]P = (83, 3) \in E[15]$
- $[21]P = (46, 76) \in E[5]$, also $(46, 76) \in E[15]$ and $(46, 76) \in E[35]$

- For $P \in E[r]$, division by 0 occurs in addition of P and $[r-1]P = -P$ (same x coordinate)
- Can we know r-torsion in advance. . . ?

Division polynomials of $E : y^2 = x^3 + ax + b$

- Can we guess points of order r in advance (i.e. without testing multiplication by r)?
- Compute $[r](x, y)$ (leave indeterminate) and look at which (x, y) values make denominators vanish
- More formally, *division polynomials* (defined recursively depending on E) do this ...

Division polynomials on E

The roots of the r-th division polynomial $\psi_r(x, y)$ correspond to r-torsion points of E

- $\psi_{2m+1} \in \mathbb{Z}[x, a, b]$ and $\psi_{2m} \in 2y\mathbb{Z}[x, a, b]$

e.g. recall $E/\mathbb{F}_{101} : y^2 = x^3 + x + 1$ with $\#E = 105 = 3 \cdot 5 \cdot 7$

- $\psi_2(x) = 4x^3 + 4x + 4$ - irreducible in $\mathbb{F}_p[x]$, so no 2-torsion over $\mathbb{F}_p$
- $\psi_3(x) = 3x^4 + 6x^2 + 12x + 100 = (x + 73)(x + 84)(x^2 + 45x + 36)$, $x = 17$ and $x = 28$ will give 3-torsion points (over $\mathbb{F}_p$ or $\mathbb{F}_p^2$)

Endomorphisms on E/K

- Endomorphisms ϕ are homomorphisms from E to itself, i.e. $\phi : E \to E$.
- We have already seen them several times: e.g. the doubling map is an endomorphism on E, i.e. $[2] : E \to E$
- In fact, the *multiplication-by-m* map $[m] : E \to E$ is an endomorphism for all $m \in \mathbb{Z}$

$$[m] : (x, y) \mapsto \left(x - \frac{\psi_{m-1}\psi_{m+1}}{\psi_n^2}, \frac{\psi_{2m}}{2\psi_m^4} \right)$$

- There can be others depending on E, e.g. $E : y^2 = x^3 + b$ then $\phi : (x, y) \mapsto (\xi_3 x, y)$ is a map

$\mathrm{End}(E)$ - the endomorphism ring of E

- Endomorphisms on E form a ring $\mathrm{End}(E)$,
  - addition in $\mathrm{End}(E)$ is as usual - $(\phi_1 + \phi_2)(P) = \phi_1(P) + \phi_2(P)$
  - multiplication is composition - $(\phi_1\phi_2)(P) = \phi_1(\phi_2(P))$
- Since each $m \in \mathbb{Z}$ induces an endomorphism $[m]$ on E, $\mathrm{End}(E)$ is as least as big as $\mathbb{Z}$
- If there is any additional, e.g. $\phi : (x, y) \mapsto (\xi_3 x, y)$ on $E : y^2 = x^3 + b$, then we saw E has *complex multiplication* (CM)
- Over finite fields $\mathbb{F}_q$, we always have an additional endomorphism regardless of E ...

The Frobenius endomorphism π

The q-power Frobenius endomorphism

For an elliptic curve $E/\mathbb{F}_q$, the q-power Frobenius endomorphism $\pi : E \to E$ is defined by $\pi : (x, y) \mapsto (x^q, y^q)$

e.g. $q = 67$, $E/\mathbb{F}_q : y^2 = x^3 + 4x + 3$, $\mathbb{F}_q^2 = \mathbb{F}_q(i)$ where $i^2 = -1$

- $P_1 = (15, 50) \in E(\mathbb{F}_q)$, so
  $\pi(P_1) = (15^q, 50^q) = (15, 50) = P_1$
- $P_2 = (16 + 2i, 39 + 30i) \in E(\mathbb{F}_q^2)$, so
  $\pi(P_2) = ((16 + 2i)^q, (39 + 30i)^q) = (16 + 65i, 39 + 37i)$
  ("complex conjugation")

- π maps any point in $E(\bar{F}_q)$ to another point in $E(\bar{F}_q)$...
- the set of points fixed by π is exactly $E(\mathbb{F}_q)$

Schoof using π's characteristic poly.

- In $\mathrm{End}(E)$, π satisfies
$$\pi^2 - [t] \circ \pi + [q] = 0,$$
meaning that for any point on $E(\overline{K})$, we have
$$(x^{q^2}, y^{q^2}) - [t](x^q, y^q) + [q](x, y) = \mathcal{O}$$
- Recall Schoof wanted t mod ℓ for many primes $\ell \rightarrow$ work on above equation modulo ℓ to find it!
- We don't know where/what ℓ-torsion points are (since we don't know $\#E$), so we treat them as $(x, y) \in \mathbb{F}_q[x, y]$
- **How to work "modulo ℓ" on E = work modulo division polynomials $\psi_\ell(x, y)$**
- This is what keeps computations feasible, allows us to compute $\#E$ in polynomial time

Summary so far. . .

- **What we have seen**
  - How to compute the group law (double and add) on E, so we can compute $[m]P$ efficiently (and therefore do (EC)DLP-based protocols)
  - How to count points efficiently, so we can also make sure the curves we work on are secure (large prime subgroup)

- **What we haven't seen**
  - There have been many advances to making ECC even more efficient
  - e.g. different curve models (not $y^2 = x^3 + ax + b$) that allow faster arithmetic (Edwards, Hessian, Jacobi-Quartic)
  - e.g. using endomorphisms to speed up $[m]P$ computation (GLV/GLS scalar decomposition)
  - e.g. extensions of *double-and-add*, i.e. windowing, double-base, NAF etc
  - Hyperelliptic curves. . .
  - **Attacks and cryptanalysis!!!**
  - Much more . . .

4. Divisors

The language of divisors

- The language of divisors is very natural and convenient
- A divisor D on E is a nice way to write a multi-set of points on E, written as the formal sum

$$D = \sum_{P \in E(K)} n_P(P)$$

  where all but finitely many n_P are zero.
- (Defn:) The *support* $\mathrm{supp}(D)$ of D is the set of P where $n_P \neq 0$
- (Defn:) The *degree* $\deg(D)$ of D is the sum of all the n_P
- Divisors form a group $\mathrm{Div}(E)$, where addition is natural

An example

$D_1 = 2(P) - 3(Q)$ and $D_2 = 3(Q) - (R) - (S)$ for $P, Q, R, S \in E$

- $D_1 \in \mathrm{Div}(E)$ and $D_2 \in \mathrm{Div}(E)$
- $\mathrm{supp}(D_1) = \{P, Q\}$ and $\mathrm{supp}(D_2) = \{Q, R, S\}$
- $\deg(D_1) = -1$ and $\deg(D_2) = 1$
- $D_1 + D_2 = 2(P) - (R) - (S)$
- $\deg(D_1 + D_2) = 0$

Divisors of functions

- Divisors are most useful because the simplify everything we need to know about a function f on the curve
- When studying $f \in \mathbb{F}_q(E)$, we only care about where f intersects/coincides with E
- The *divisor of a function* f, written as (f), writes down the zeros and poles (with multiplicities) of f on E

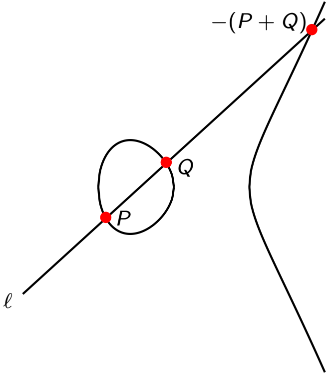
$$(f) = \sum_{P \in E(\bar{F}_q)} \mathrm{ord}_P(f)(P),$$

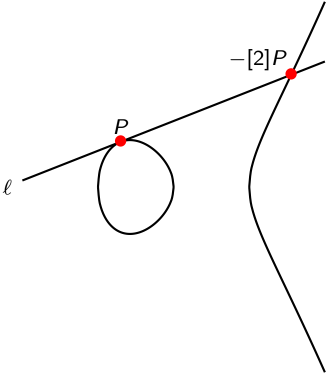
- $(fg) = (f) + (g)$ and $(f/g) = (f) - (g)$, etc
- **if** $D = (f)$**, then** D **determines** f **up to constant**

Thm: Divisors of functions have degree 0.

Proof: Galbraith's new book (Th 7.7.1)

Examples (we've already seen)





$(\ell) = (P) + (Q) + (-(P+Q)) - 3(\mathcal{O}).$

$(\ell) = 2(P) + (-[2]P) - 3(\mathcal{O}).$

- If you have a function and you know all the zeros on E, just subtract the appropriate multiple of $\mathcal{O}$

The divisor class group

- Divisors of functions are called *principal divisors*, denoted $\mathrm{Prin}(E)$
- Divisors of functions have degree 0, but the converse is not always true, i.e.

$$\mathrm{Prin}(E) \subset \mathrm{Div}^0(E) \subset \mathrm{Div}(E)$$

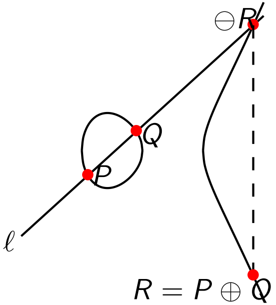
The divisor class group

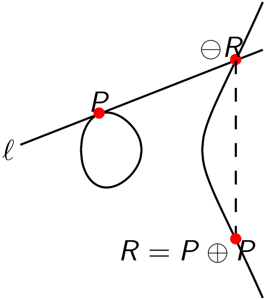
The *divisor class group*, or *Picard group*, of E is the quotient group

$$\mathrm{Pic}^0(E) = \mathrm{Div}^0(E)/\mathrm{Prin}(E).$$

- So, we work only with degree zero divisors, and all divisors which are (f) for any f's are zero

The group law in terms of divisors





- $(\ell) = (P) + (Q) + (-R) - 3(\mathcal{O})$ and $(v) = (R) + (-R) - 2(\mathcal{O})$
- So $(\ell/v) = (P) + (Q) - (R) - (\mathcal{O})$, but $(\ell/v) \sim 0$ in $\mathrm{Pic}^0(E)$...

$$(P) - (\mathcal{O}) + (Q) - (\mathcal{O}) = (R) - (\mathcal{O})$$

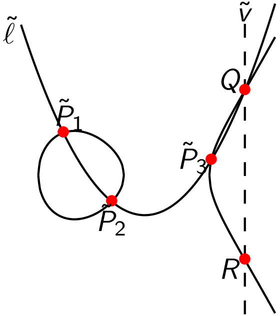
- $T \in E$ to $(T) - (\mathcal{O}) \in \mathrm{Pic}^0(E)$ is a group homomorphism

Reduced divisors

- A divisor $\sum_{P \in E(\bar{K})} n_P(P)$ is called effective if $n_P \geq 0$
- Take the "effective part" to be the part with all $n_P \geq 0$

A consequence of the Riemann-Roch theorem

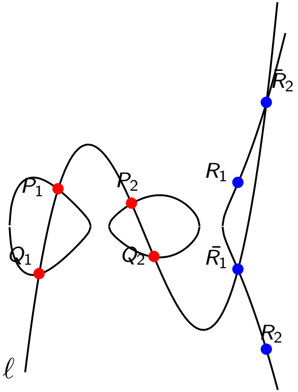
On a curve of genus g, every divisor class has a representative divisor with effective part of degree at most g



Reduce $(\tilde{P}_1) + (\tilde{P}_2) + (\tilde{P}_3) - 3(\mathcal{O})$ to $(R) - (\mathcal{O})$ in $\mathrm{Pic}^0(\mathrm{E})$ (genus 1).

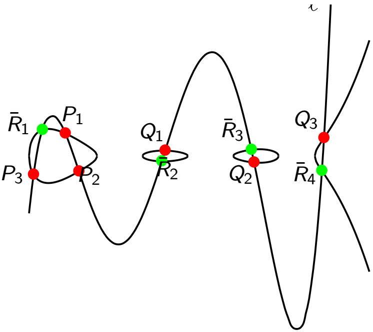
The genus 2 group law

- Elliptic curves are genus 1 - their higher genus analogues are called *hyperelliptic curves*
- e.g. addition on a genus 2 curve: $y^2 = x^5 + a_4x^4 + \cdots + a_0$

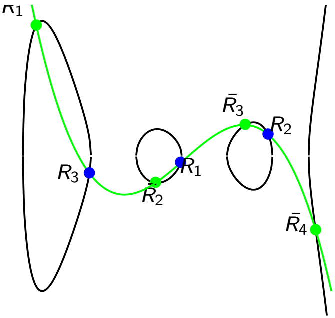


The genus 3 group law

- A genus 3 hyperelliptic curve $y^2 = x^7 + \cdots + a_0$



The first stage of reduction.



The second stage of reduction.

Functions of divisors

- Let f be a function on E, and $D = \sum_{P \in E(\bar{K})} n_P(P)$, then

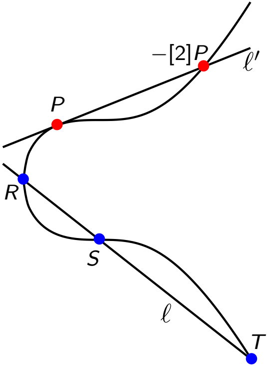
$$f(D) = \prod_{P \in E(\bar{K})} f(P)^{n_P}$$

e.g. $E/\mathbb{F}_{163} : y^2 = x^3 - x - 2$ $P = (43, 154)$, $Q = (46, 38)$, $R = (12, 35)$, $S = (5, 66)$

- functions $\ell_{P,Q} = y + 93x + 85$, $\ell_{P,P} = y + 127x + 90$ and $\ell_{Q,Q} = y + 13x + 16$
- divisors $D_1 = 2(R) + (S)$, $D_2 = 3(R) - 3(S)$ and $D_3 = (R) + (S) - 2(\mathcal{O})$
- e.g. $\ell_{P,Q}(D_1) = (y_R + 93x_R + 85)^2(y_S + 93x_S + 85) = 122$
- e.g. $\ell_{P,P}(D_2) = (y_R + 127x_R + 90)^3/(y_S + 127x_S + 90)^3 = 53$
- e.g. can't evaluate any functions at D_3, since $\mathcal{O} \in \mathrm{supp}(D_3)$ and $\mathcal{O}$ also in supports of $(\ell_{P,Q})$, $(\ell_{P,P})$ and $(\ell_{Q,Q})$

Weil reciprocity on elliptic curves (but general)

Let f, g on E have **disjoint support**, then $f((g)) = g((f))$



$$\ell(\ell') = \ell'(\ell).$$

5. Pairings on elliptic curves

Pairings are **bilinear** maps

- The most general definition of an elliptic curve pairing e

$$\begin{array}{llclcl} e: & \mathbb{G}_1 & \times & \mathbb{G}_2 & \to & \mathbb{G}_T \\ e: & E/\mathbb{F}_q[r] & \times & E/\mathbb{F}_q[r] & \to & \mu_r \in \mathbb{F}_{q^k}^{*} \\ e: & P & \times & Q & \mapsto & e(P,Q) \end{array}$$

- **Bilinear** means

$$e(P+P',Q) = e(P,Q)\cdot e(P',Q),$$
$$e(P,Q+Q') = e(P,Q)\cdot e(P,Q'),$$

from which it follows that, for scalars $a, b \in \mathbb{Z}$, we have

$$e([a]P,[b]Q) = e(P,[b]Q)^a = e([a]P,Q)^b = e(P,Q)^{ab} = e([b]P,[a]Q).$$

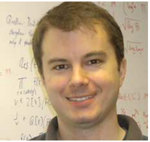
The power of bilinearity (some famous examples)

Joux

One-ride tripartite DH

Boneh-Franklin

Identity-based encryption (IBE)



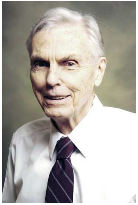

Gentry-Silverberg

Heirarchical ID-based encryption (HIBE)


Sahai- Waters

Attribute-based encryption (ABE)

André Weil



John Tate

Let $f_{r,P}$ be the (unique up to constant) function with divisor $(f_{r,P}) = r(P) - r(\mathcal{O})$

Weil pairing (in crypto): $e(P,Q) = \frac{f_{r,P}(Q)}{f_{r,Q}(P)}$;

Tate pairing (in crypto): $e(P,Q) = f_{r,P}(Q)^{(q^k-1)/r}$,

The function $f_{r,P}(Q)$ is huuuuuge!

The size of $f_{r,P}(Q)$: 128-bit security

- The pairing function $f_{r,P}(Q)$ is of degree r, where
  $r = 16798108731015832284940804142231733909759579603404752749028378864165570215949$
- The coefficients in $f_{r,P}(Q)$ depend on P's coordinates, so are all of the size
  $P_x = 15283023184232661393336451140837190640382743162584629974443682653991135323854$
- This huge function is impossible to store with all the computing power in the world. Somehow we need to evaluate it at Q, whose x coordinate is

$$
\begin{aligned}
Q_x = {} & ((15550921060303536733405227206218421303411153835059642979852113370177068459559 \cdot u + \\
& 36006906447969872904421351370312852062497895145888276790029208075555440045456) \cdot v^2 + \\
& (5475264847170057761513968927972623766794030526092071182289628553939256498415 \cdot u + \\
& 16045231392378269041781500461472571507692250280489500368315808811462293278705) \cdot v + \\
& (13578969743206791049626159973437892548805434308942546900125761281664803554809 \cdot u + \\
& 8414705805435201691796063348962631501393112240468038251361145485591996962517)) \cdot w + \\
& (2095760324718272519234982374519336043146898698412090865684809945855004557738 \cdot u + \\
& 10991749562144480578133596744105999544930359103290000221828602811069330922292) \cdot v^2 + \\
& (563526440913857199739302175501170867491400605855901007410492904987821568516 \cdot u + \\
& 12175465566401923735806619064706225201231722038674162959277121785143969709483) \cdot v + \\
& 5977392629488041467394421854470109162392545860735885669496575455742917555185 \cdot u + \\
& 16414735455238441715243107544357668247548687753217062857281803216595664241398
\end{aligned}
$$

The size of $f_{r,P}(Q)$: 128-bit security

- The pairing function $f_{r,P}(Q)$ is of degree r, where
  $r = 16798108731015832284940804142231733909759579603404752749028378864165570215949$
- The coefficients in $f_{r,P}(Q)$ depend on P's coordinates, so are all of the size
  $P_x = 15283023184232661393336451140837190640382743162584629974443682653991135323854$
- This huge function is impossible to store with all the computing power in the world. Somehow we need to evaluate it at Q, whose x coordinate is

$$
\begin{aligned}
Q_x = \; & ((15550921060303536733405227206218421303411153835059642979852113370177068459559 \cdot u+ \\
& 36006906447969872904421351370312852062497895145888276790029208075555440045456) \cdot v^2+ \\
& (5475264847170057761513968927972623766794030526092071182289628553939256498415 \cdot u+ \\
& 16045231392378269041781500461472571507692250280489500368315808811462293278705) \cdot v+ \\
& (13578969743206791049626159973437892548805434308942546900125761281664803554809 \cdot u+ \\
& 8414705805435201691796063348962631501393112240468038251361145485591996962517)) \cdot w+ \\
& (2095760324718272519234982374519336043146898698412090865684809945855004557738 \cdot u+ \\
& 10991749562144480578133596744105999544930359103290000221828602811069330922292) \cdot v^2+ \\
& (563526440913857199739302175501170867491400605855901007410492904987821568516 \cdot u+ \\
& 12175465566401923735806619064706225201231722038674162959277121785143969709483) \cdot v+ \\
& 5977392629488041467394421854470109162392545860735885669496575455742917555185 \cdot u+ \\
& 16414735455238441715243107544357668247548687753217062857281803216595664241398
\end{aligned}
$$

Remarkably, this can actually be done in less than a millisecond on your PC!!! - find out how on Tuesday!

Summary

1. Motivation
   - ECDLP much harder to solve than DLP $\rightarrow$ ECC has shorter keys and is faster than standard groups
2. Elliptic curves are groups
   - The group operation: chord-and-tangent rule
   - Projective space and the point at infinity
   - Group axioms
3. Elliptic curves as cryptographic groups
   - Setting up ECDLP instances
   - Best (secure) curves have close to prime order
   - Point counting, division polynomials, the endomorphism ring
4. Divisors
   - Divisors of functions and functions of divisors
   - Divisor class group
   - Higher genus examples
   - Weil reciprocity
5. A very brief look at pairings
   - A bilinear map that's very useful, but requires huge function to be computed . . . much more on Tuesday . . .

Questions?

Thanks for your attention