

A gentle introduction to elliptic curve cryptography

Craig Costello

Microsoft®
Research



UNIVERSITY OF WATERLOO

2019

Part 1: Motivation

Part 2: Elliptic Curves

Part 3: Elliptic Curve Cryptography

Part 4: Next-generation ECC

Diffie-Hellman key exchange (circa 1976)

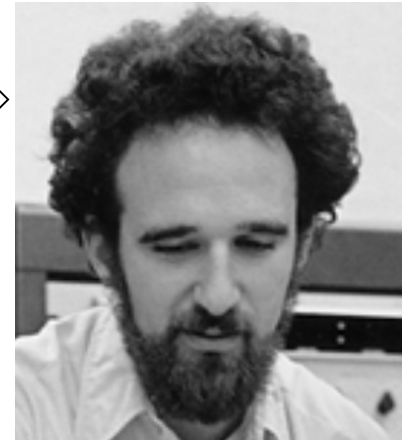
$q = 1606938044258990275541962092341162602522202993782792835301301$

$g = 123456789$



$g^a \bmod q = 78467374529422653579754596319852702575499692980085777948593$

$560048104293218128667441021342483133802626271394299410128798 = g^b \bmod q$



$a =$

685408003627063
761059275919665
781694368639459
527871881531452

$b =$

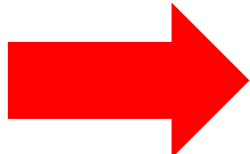
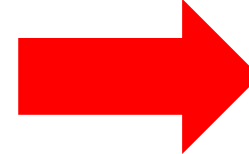
362059131912941
987637880257325
269696682836735
524942246807440

$g^{ab} \bmod q = 437452857085801785219961443000845969831329749878767465041215$

Index calculus

solve $g^x \equiv h \pmod{p}$
 e.g. $3^x \equiv 37 \pmod{1217}$

- factor base $p_i = \{2,3,5,7,11,13,17,19\}$, $\#p_i = 8$
- Find 8 values of k where 3^k splits over p_i , i.e., $3^k \equiv \pm \prod p_i \pmod{p}$

(mod 1217)		(mod 1216)		(mod 1216)
$3^1 \equiv 3$		$1 \equiv L(3)$		$L(2) \equiv 216$
$3^{24} \equiv -2^2 \cdot 7 \cdot 13$		$24 \equiv 608 + 2 \cdot L(2) + L(7) + L(13)$		$L(3) \equiv 1$
$3^{25} \equiv 5^3$		$25 \equiv 3 \cdot L(5)$		$L(5) \equiv 819$
$3^{30} \equiv -2 \cdot 5^2$		$30 \equiv 608 + L(2) + 2 \cdot L(5)$		$L(7) \equiv 113$
$3^{34} \equiv -3 \cdot 7 \cdot 19$		$34 \equiv 608 + L(3) + L(7) + L(19)$		$L(11) \equiv 1059$
$3^{54} \equiv -5 \cdot 11$		$54 \equiv 608 + L(5) + L(11)$		$L(13) \equiv 87$
$3^{71} \equiv -17$		$71 \equiv 608 + L(17)$		$L(17) \equiv 679$
$3^{87} \equiv 13$		$87 \equiv L(13)$		$L(19) \equiv 528$

Index calculus

$$\begin{array}{l} \text{solve} \quad g^x \equiv h \pmod{p} \\ \text{e.g.} \quad 3^x \equiv 37 \pmod{1217} \end{array}$$

$$\begin{array}{l} L(2) \equiv 216 \\ L(3) \equiv 1 \\ L(5) \equiv 819 \\ L(7) \equiv 113 \\ L(11) \equiv 1059 \\ L(13) \equiv 87 \\ L(17) \equiv 679 \\ L(19) \equiv 528 \end{array}$$

Now search for j such that $g^j \cdot h = 3^j \cdot 37$ factors over p_i

$$3^{16} \cdot 37 \equiv 2^3 \cdot 7 \cdot 11 \pmod{1217}$$

$$\begin{aligned} L(37) &\equiv 3 \cdot L(2) + L(7) + L(11) - 16 \pmod{1216} \\ &\equiv 3 \cdot 216 + 113 + 1059 - 1 \\ &\equiv 588 \end{aligned}$$

$$\text{Subexponential complexity } L_p\left[1/3, (64/9)^{1/3}\right] = e^{\left(\left(64/9\right)^{1/3} + o(1)\right)(\ln(p))^{1/3} \cdot (\ln \ln(p))^{2/3}}$$

Diffie-Hellman key exchange (circa 2016)

$$q =$$

58096059953699580628595025333045743706869751763628952366614861522872037309971102257373360445331184072513261577549805174439905295945400471216628856721870324010321116397064404988440498509890516272002447658070418123947296805400241048279765843693815222923216208779044769892743225751738076979568811309579125511333093243519553784816306381580161860200247492568448150242515304449577187604136428738580990172551573934146255830366405915000869643732053218566832545291107903722831634138599586406690325959725187447169059540805012310209639011750748760017095360734234945757416272994856013308616958529958304677637019181594088528345061285863898271763457294883546638879554311615446446330199254382340016292057090751175533888161918987295591531536698701292267685465517437915790823154844634780260102891718032495396075041899485513811126977307478969074857043710716150121315922024556759241239013152919710956468406379442914941614357107914462567329693649

$$g = 123456789$$

$$g^a \pmod{q} =$$

19749664818322719328626201861425055597190979976253376065400814799487577544566705421857810513313821749720689059955492842945066789947685466859559403409349363756245107893829696031348869617884814249135168725305460220296624704610577077157724832168211717424612832119567853763152027864940346479735369199673699357709268717838560229887355895412105643052289961976145372708221782347574622380379001423505139679904944650822466185016814995740147463845671662440190670139447244701505256941774637218509330253573938379198007057238142172902965163930423436126876497170776348430066892397286870912166556869830978657804740157916611563508569886847487726766712073860961529476071145597063402090591037030181826355218987380945462945580355697525966763466146993277420884712557411847558661178122098955149524361601993365326052422101474898256696660124195726100495725510022002932814218768060112310763455404567248761396399633344901857872119208518550803791724

$$g^b \pmod{q} =$$

411604662069593306683228525653441872410777999220572079993574397237156368762038378332742471939666544968793817819321495269833613169937986164811320795616949957400518206385310292475529284550626247132930124027703140131220968771142788394846592816111078275196955258045178705254016469773509936925361994895894163065551105161929613139219782198757542984826465893457768888915561514505048091856159412977576049073563225572809880970058396501719665853110101308432647427865655251213287725871678420337624190143909787938665842005691911997396726455110758448552553744288464337906540312125397571803103278271979007681841394534114315726120595749993896347981789310754194864577435905673172970033596584445206671223874399576560291954856168126236657381519414592942037018351232440467191228145585909458612780918001663308764073238447199488070126873048860279221761629281961046255219584327714817248626243962413613075956770018017385724999495117779149416882188

$$a =$$

7147687166405; 9571879053605547396582692405186145916522354912615715297097100679170037904924330116019497881089087696131592831386326210951294944584400497488929803858493191812844757232102398716043906200617764831887545755623377085391250529236463183321912173214641346558452549172283787727566955898452199622029450892269665074265269127802446416400190259271040043389582611419862375878988193612187945591802864062679186483957813927304368495559776413009721221824915810964579376354556165546298837778595680891578821511273574220422646379170599917677567\30420698422392494816906777896174923072071297603455802621072109220\5466273969774855343758990879608882627763290293452560094576029847\39136138876755438662247926529997805988647241453046219452761811989\9746472529088780604931795419514638292288904557780459294373052654\10485180264002079415193983851143425084273119820368274789460587100\30497747706924427898968991057212096357725203480402449913844583448

$$b =$$

655456209464694; 93360682685816031704969423104727624468251177438749706128879957701\93698826859762790479113062308975863428283798589097017957365590672\8357138638957122466760949930089855480244640303954430074800250796203638661931522988606354100532244846391589798641210273772558373965\486653931285483865070903191974204864923589439190352993032676961005\08840431979272991603892747747094094858192679116146502863521484987\08623286193422239171712154568612530067276018808591500424849476686\706784051068715397706852664532638332403983747338379697022624261377163163204493828299206039808703403575100467337085017748387148822224875309641791879395483731754620034884930540399950519191679471224\0555855709321935074711577569598163700850920394705281936392411084\4360068618352846572496956218643721497262583322544865996160464558\5462993701658947042526445624157899586972652935647856967092689604\42796501209877036845001246792761563917639959736383038665362727158

$$g^{ab} =$$

330166919524192149323761733598426244691224199958894654036331526394350099088627302979833339501183059198113987880066739419999231378970715307039317876258453876701124543849520979430233302775032650107245135512092795731832349343596366965069683257694895110289436988215186894965977582185407675178858364641602894716513645524907139614566085360133016497539758756106596557555674744381803579583602267087423481750455634370758409692308267670340611194376574669939893893482895996003389503722513369326735717434288230260146992320711161713922195996910968467141336433827457093761125005143009836512019611866134642676859265636245898172596372485581049036573719816844170539930826718273452528414333373254200883800592320891749460865366649848360413340316504386926391062876271575757583831289710534010374070317315095828076395094487046179839301350287596589383292751933079161318839043121329118930009948197899907586986108953591420279426874779423560221038468



Diffie-Hellman key exchange (cont.)

- Individual secret keys secure under Discrete Log Problem (DLP): $g, g^x \mapsto x$
- Shared secret secure under Diffie-Hellman Problem (DHP): $g, g^a, g^b \mapsto g^{ab}$
- Fundamental operation in DH is group exponentiation: $g, x \mapsto g^x$
... done via “square-and-multiply”, e.g., $(x)_2 = (1,0,1,1,0,0,0,1 \dots)$
- We are working “**mod** q ”, but only with one operation: multiplication
- Main reason for fields being so big: (sub-exponential) index calculus attacks!

DH key exchange (Koblitz-Miller style)

If all we need is a group, why not use elliptic curve groups?



MATHEMATICS OF COMPUTATION
VOLUME 46, NUMBER 177
JANUARY 1985, PAGES 203-209

Elliptic Curve Cryptosystems

By Neal Koblitz

This paper is dedicated to Daniel Shanks on the occasion of his seventieth birthday

Abstract. We discuss analogs based on elliptic curves over finite fields of public key cryptosystems which use the multiplicative group of a finite field. These elliptic curve cryptosystems may be more secure, because the analog of the discrete logarithm problem on elliptic curves is likely to be harder than the classical discrete logarithm problem, especially over $GF(2^n)$. We discuss the question of primitive points on an elliptic curve modulo p , and give a theorem on nonsmoothness of the order of the cyclic subgroup generated by a global point.

1. Introduction. The earliest public key cryptosystems using number theory were based on the structure either of the multiplicative group $(\mathbb{Z}/N\mathbb{Z})^*$ or the multiplicative group of a finite field $GF(q)$, $q = p^n$. The subsequent construction of analogous systems based on other finite Abelian groups, together with H. W. Lenstra's success in using elliptic curves for integer factorization, make it natural to study the possibility of public key cryptography based on the structure of the group of points of an elliptic curve over a large finite field. We first briefly recall the facts we need about such elliptic curves (for more details, see [4] or [5]). We then describe elliptic curve analogs of the Massey-Omura and ElGamal systems. We give some concrete examples, discuss the question of primitive points, and conclude with a theorem concerning the probability that the order of a cyclic subgroup is nonsmooth.

I would like to thank A. Odlyzko for valuable discussions and correspondence, and for sending me a preprint by V. S. Miller, who independently arrived at some similar ideas about elliptic curves and cryptography.

2. Elliptic Curves. An elliptic curve E_K defined over a field K of characteristic $\neq 2$ or 3 is the set of solutions $(x, y) \in K^2$ to the equation

$$(1) \quad y^2 = x^3 + ax + b, \quad a, b \in K$$

(where the cubic on the right has no multiple roots). More precisely, it is the set of such solutions together with a "point at infinity" (with homogeneous coordinates $(0, 1, 0)$; if K is the real numbers, this corresponds to the vertical direction which the tangent line to E_K approaches as $x \rightarrow \infty$). One can start out with a more complicated general formula for E_K which can easily be reduced to (1) by a linear change of variables whenever $\text{char} K \neq 2, 3$. If $\text{char} K = 2$ —an important case in

Received October 29, 1985; revised June 5, 1986.
1980 *Mathematics Subject Classification* (1985 Revision). Primary 11T71, 94A60; Secondary 68P25, 11Y11, 11Y40.

©1987 American Mathematical Society
0025-5718/87 \$1.00 + \$.25 per page

203

License or copyright restrictions may apply to redistribution; see <http://www.ams.org/journal-terms-of-use>

Use of Elliptic Curves in Cryptography

Victor S. Miller

Exploratory Computer Science, IBM Research, P.O. Box 218, Yorktown Heights, NY 10598

ABSTRACT

We discuss the use of elliptic curves in cryptography. In particular, we propose an analogue of the Diffie-Hellman key exchange protocol which appears to be immune from attacks of the style of Western, Miller, and Adleman. With the current bounds for infeasible attack, it appears to be about 20% faster than the Diffie-Hellman scheme over $GF(p)$. As computational power grows, this disparity should get rapidly bigger.



H.C. Williams (Ed.): *Advances in Cryptology - CRYPTO '85*, LNCS 218, pp. 417-426, 1986.
© Springer-Verlag Berlin Heidelberg 1986

Rationale: "it is extremely unlikely that an index calculus attack on the elliptic curve method will ever be able to work" [Miller, 85]

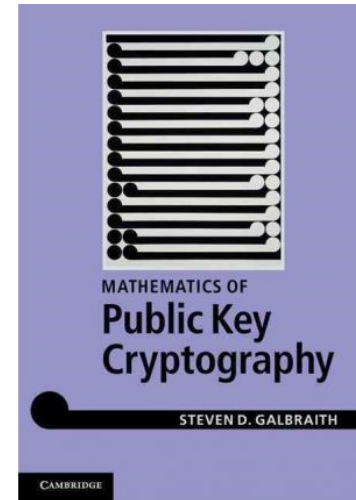
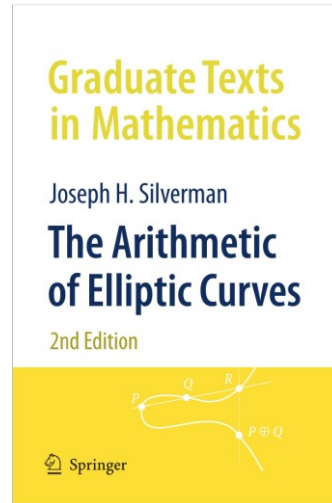
Part 1: Motivation

Part 2: Elliptic Curves

Part 3: Elliptic Curve Cryptography

Part 4: Next-generation ECC

Some good references



Elliptic
curves

Silverman's talk: "An Introduction to the Theory of Elliptic Curves"
<http://www.math.brown.edu/~jhs/Presentations/WyomingEllipticCurve.pdf>

Elliptic
curves

Sutherland's MIT course on elliptic curves:
<https://math.mit.edu/classes/18.783/2015/lectures.html>

ECC

Koblitz-Menezes: ECC: the serpentine course of a paradigm shift
<http://eprint.iacr.org/2008/390.pdf>

group $(G, +)$

can do $+$ $-$

ring $(R, +, \times)$

can do $+$ $-$ \times

field $(F, +, \times)$

can do $+$ $-$ \times \div

If you've never seen an elliptic curve before....

Remember: an elliptic curve is a group defined over a field

elliptic curve group (E, \oplus)

can do $\oplus \ominus$

underlying field $(K, +, \times)$

can do $+ - \times \div$

operations in underlying field are used and combined to
compute the elliptic curve operation \oplus

Boring curves

$$f(x, y) = 0 \quad \text{or} \quad f(X, Y, Z) = 0$$

Degree 1 (lines)

$$ax + by = c$$

$$ab \neq 0$$

Degree 2 (conic sections)

$$ax^2 + bxy + cy^2 + dx + ey + f = 0$$

$$abc \neq 0$$

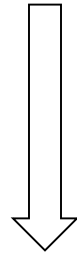
e.g., ellipses, hyperbolas, parabolas

- "Genus" measures geometric complexity, and both are genus 0
- We know how to describe all solutions to these, e.g., over (exts of) \mathbb{Q}
- Not cryptographically interesting

Elliptic curves

- Degree 3 is where all the fun begins...

$$ax^3 + bx^2y + cxy^2 + dy^3 + ex^2 + fxy + gy^2 + hx + iy + j = 0$$

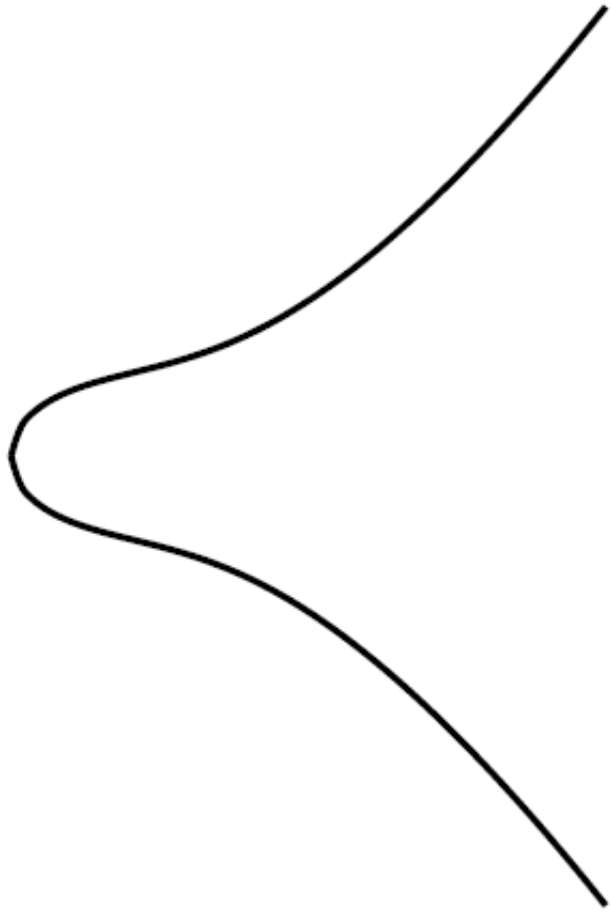


$ch(K) \neq 2,3$

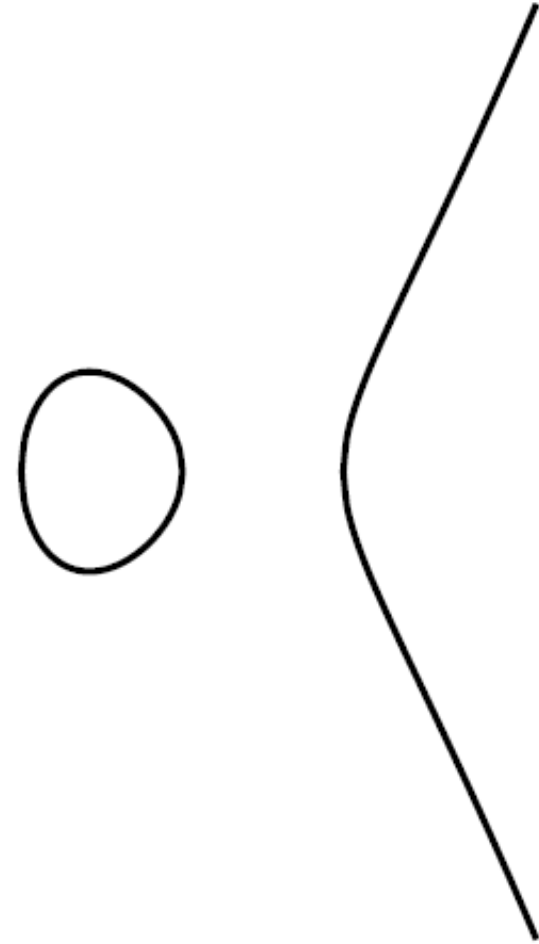
$E/K: y^2 = x^3 + ax + b$ ← E specified by K, a, b

- Elliptic curves \leftrightarrow genus 1 curves
- Set is \approx points $(x, y) \in K \times K$ satisfying above equation
- Geometrically/arithmetically/cryptographically interesting
- Fermat's last theorem/BSD conjecture/ ...

Elliptic curves, pictorially



$$E/\mathbb{R} : y^2 = x^3 + x + 1$$



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

Elliptic curves are groups

- So E is a set, but to be a group we need an *operation*
- The operation is between points $(x_P, y_P) \oplus (x_Q, y_Q) = (x_R, y_R)$
- Remember: a group (E, \oplus) defined over a field $(K, +, \times)$
- K will be fields we're used to, e.g., $\mathbb{Q}, \mathbb{C}, \mathbb{R}, \mathbb{F}_p$
- Remember: the (boring) operations $+, -, \times, \div$ in K are used to compute the (exotic) operation \oplus on E

Elliptic curve group law is easy

Fun fact: homomorphism between Jacobian of elliptic curve and elliptic curve itself.

Upshot: you don't have to know what a Jacobian is to understand/do elliptic curve cryptography

The elliptic curve group law \oplus

$$\text{We need } (x_P, y_P) \oplus (x_Q, y_Q) = (x_R, y_R)$$

Question: Given two points lying on a cubic curve, how can we use their coordinates to give a third point lying on the curve?

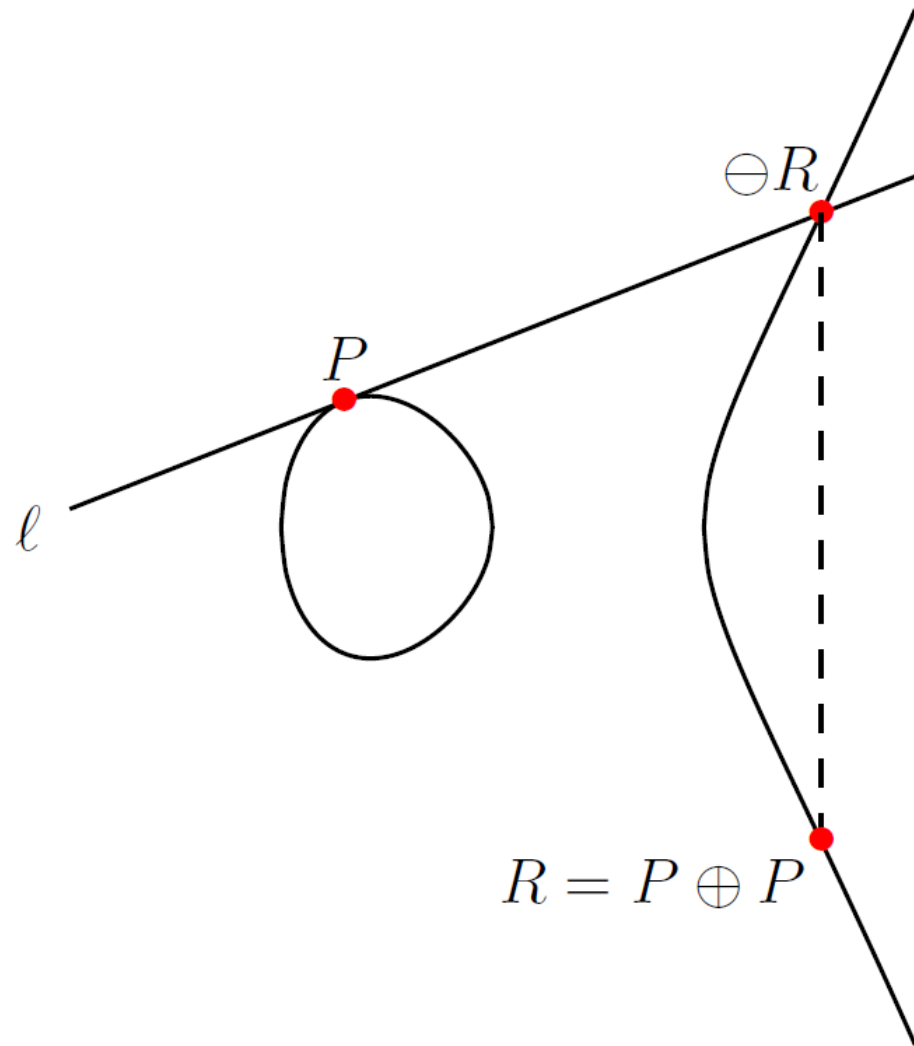
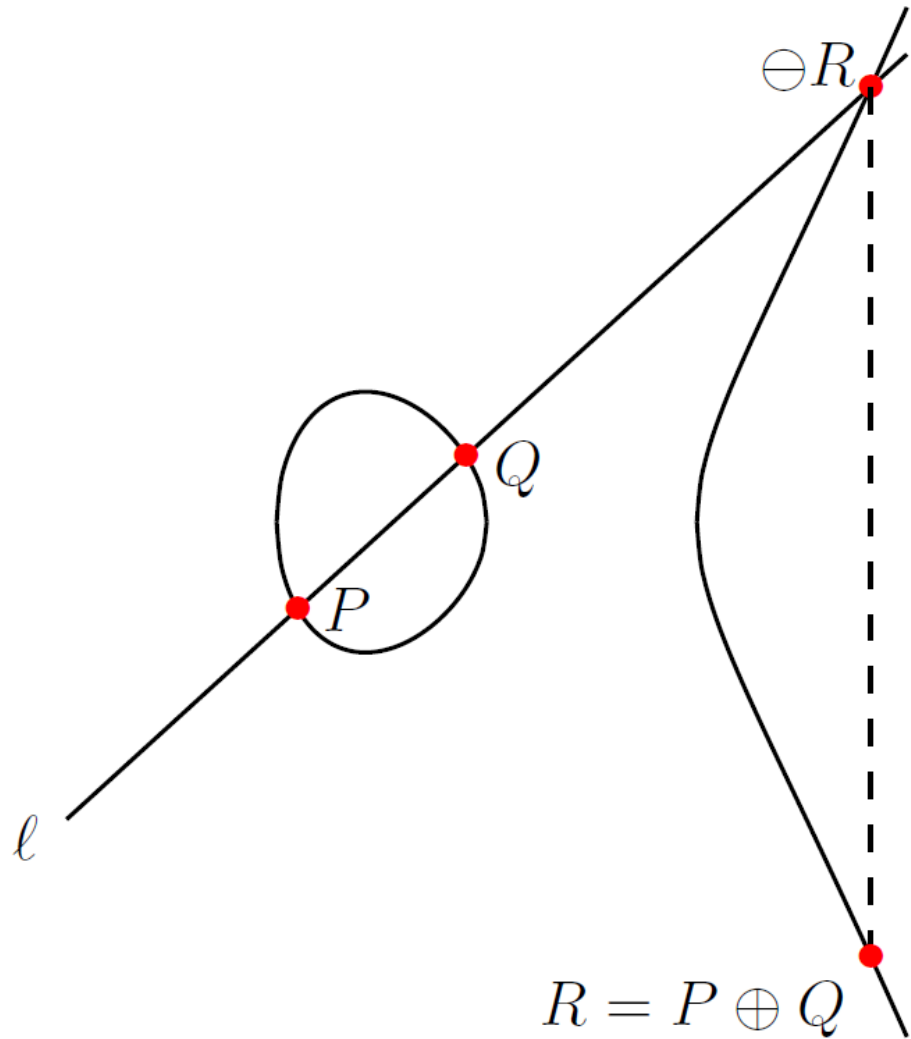
The elliptic curve group law \oplus

$$\text{We need } (x_P, y_P) \oplus (x_Q, y_Q) = (x_R, y_R)$$

Question: Given two points lying on a cubic curve, how can we use their coordinates to give a third point lying on the curve?

Answer: A line that intersects a cubic twice must intersect it again, so we draw a line through the points (x_P, y_P) and (x_Q, y_Q)

The elliptic curve group law \oplus

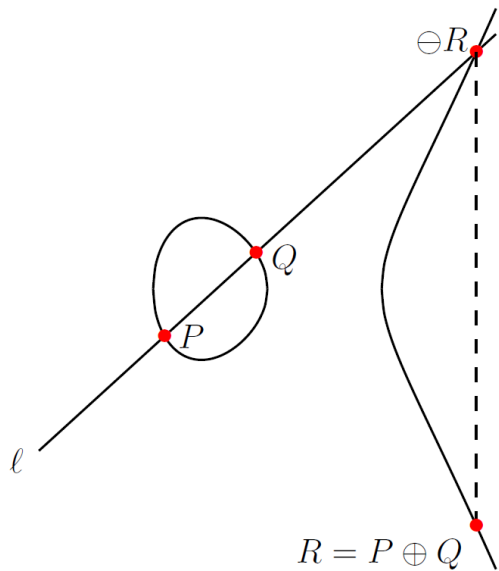


The elliptic curve group law \oplus

$y = \lambda x + v$ intersected with $y^2 = x^3 + ax + b$

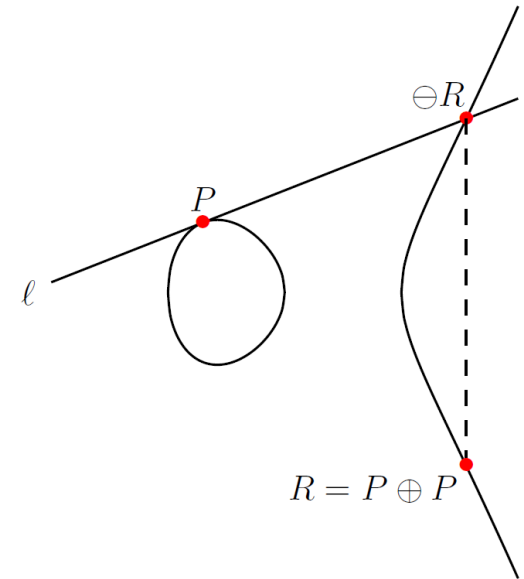
$$x^3 - (\lambda x + v)^2 + ax + b = 0$$

$$x^3 - \lambda^2 x^2 + (a - 2\lambda v)x + (b - v^2) = (x - x_P)(x - x_Q)(x - x_R)$$



$$x_R = \lambda^2 - x_P - x_Q$$

$$y_R = -(\lambda x_R + v)$$

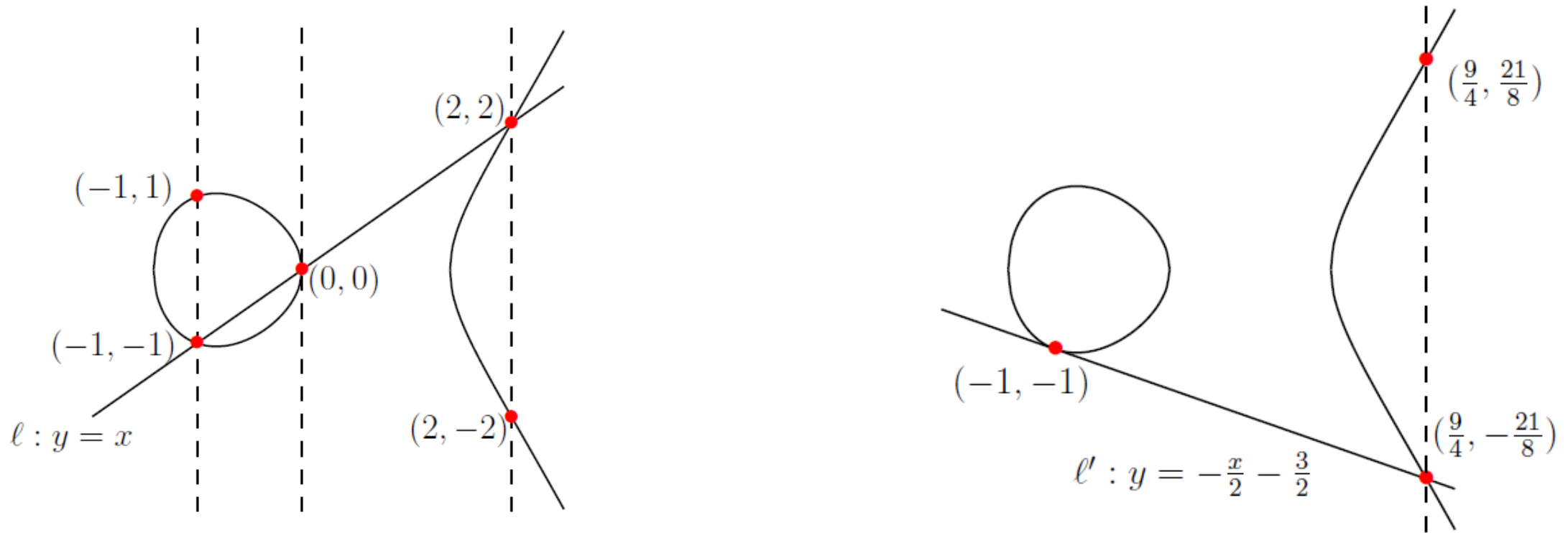


$$\lambda = \frac{dy}{dx} = \frac{3x_P^2 + a}{2y_P}$$

$$\lambda = \frac{y_Q - y_P}{x_Q - x_P}$$

A toy example

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



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

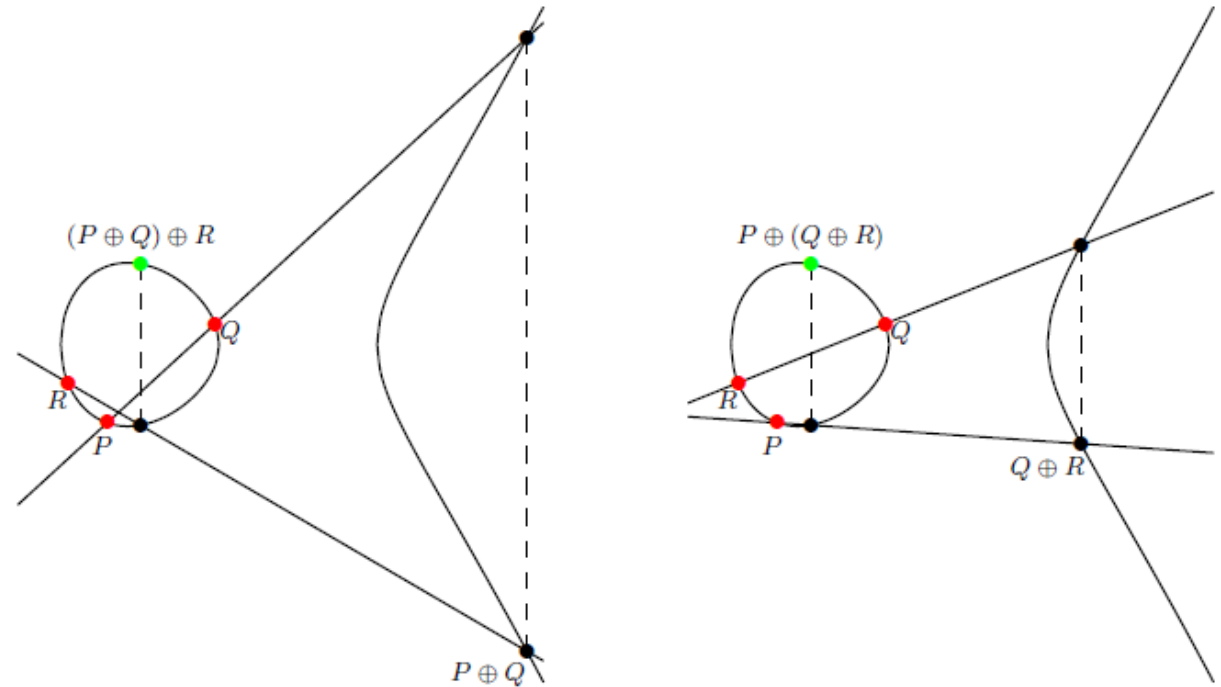
The (abelian) group axioms

- **Closure:** the third point of intersection must be in the field

- **Identity:** $E_{a,b}(K) = \{(x, y) : y^2 = x^3 + ax + b\} \cup \{\infty\}$

- **Inverse:** $\ominus (x, y) = (x, -y)$

- **Associative:** proof by picture

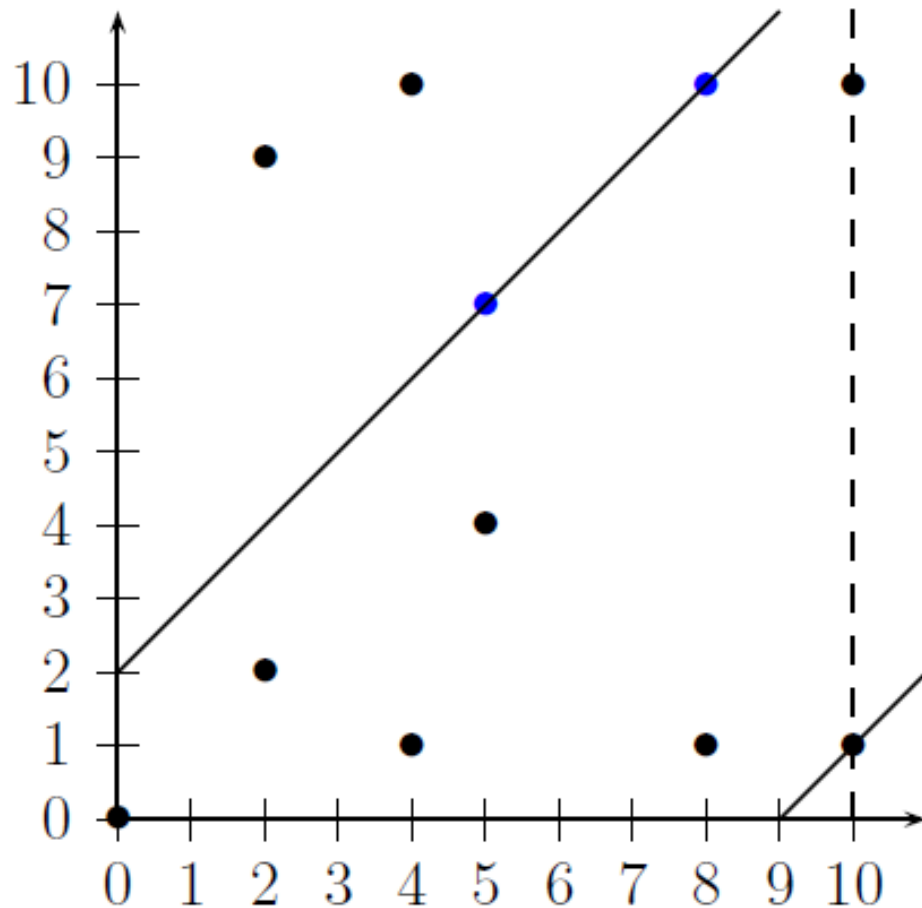


- **Commutative:** line through P and Q same as line through Q and P

A toy example, cont.

$$E/\mathbb{F}_{11}: y^2 = x^3 - 2x$$

$$\#E = 12$$



$$(5,7) \oplus (8,10) = (10,10)$$

Part 1: Motivation

Part 2: Elliptic Curves

Part 3: Elliptic Curve Cryptography

Part 4: Next-generation ECC

Diffie-Hellman key exchange (circa 2016)

$$q =$$

58096059953699580628595025333045743706869751763628952366614861522872037309971102257373360445331184072513261577549805174439905295945400471216628856721870324010321116397064404988440498509890516272002447658070418123947296805400241048279765843693815222923216208779044769892743225751738076979568811309579125511333093243519553784816306381580161860200247492568448150242515304449577187604136428738580990172551573934146255830366405915000869643732053218566832545291107903722831634138599586406690325959725187447169059540805012310209639011750748760017095360734234945757416272994856013308616958529958304677637019181594088528345061285863898271763457294883546638879554311615446446330199254382340016292057090751175533888161918987295591531536698701292267685465517437915790823154844634780260102891718032495396075041899485513811126977307478969074857043710716150121315922024556759241239013152919710956468406379442914941614357107914462567329693649

$$g = 123456789$$

$$g^a \pmod{q} =$$

19749664818322719328626201861425055597190979976253376065400814799487577544566705421857810513313821749720689059955492842945066789947685466859559403409349363756245107893829696031348869617884814249135168725305460220296624704610577077157724832168211717424612832119567853763152027864940346479735369199673699357709268717838560229887355895412105643052289961976145372708221782347574622380379001423505139679904944650822466185016814995740147463845671662440190670139447244701505256941774637218509330253573938379198007057238142172902965163930423436126876497170776348430066892397286870912166556866983097865780474015791661156350859886847487726766712073860961529476071145597063402090591037030181826355218987380945462945580355697525966763466146993277420884712557411847558661178122098955149524361601993365326052422101474898256696660124195726100495725510022002932814218768060112310763455404567248761396399633344901857872119208518550803791724

$$g^b \pmod{q} =$$

4116046620695933066832285256534418724107779992205720799935743972371563687620383783327424719396665449687938178193214952698336131699379861648113207956169499574005182063853102924755292845506262471329301240277031401312209687711427883948465928161110782751969552580451787052540164697735099369253619948958941630655511051619296131392197821987575429848264658934577688889155615145050480918561594129775760490735632255728098809700583965017196658531101013084326474277865655251213287725871678420337624190143909787938665842005691911997396726455110758448552553744288464337906540312125397571803103278271979007681841394534114315726120595749993896347981789310754194864577435905673172970033596584445206671223874399576560291954856168126236657381519414592942037018351232440467191228145585909458612780918001663308764073238447199488070126873048860279221761629281961046255219584327714817248626243962413613075956770018017385724999495117779149416882188

$$a =$$

7147687166405; 9571879053605547396582692405186145916522354912615715297097100679170037904924330116019497881089087696131592831386326210951294944584400497488929803858493191812844757232102398716043906200617764831887545755623377085391250529236463183321912173214641346558452549172283787727566955898452199622029450892269665074265269127802446416400190259271040043389582611419862375878988193612187945591802864062679186483957813927304368495559776413009721221824915810964579376354556165546298837778595680891578821511273574220422646379170599917677567\3042069842239249481690677896174923072071297603455802621072109220\5466273969774855343758990879608882627763290293452560094576029847\39136138876755438662247926529997805988647241453046219452761811989\9746472529088780604931795419514638292288904557780459294373052654\10485180264002079415193983851143425084273119820368274789460587100\30497747706924427898968991057212096357725203480402449913844583448

$$b =$$

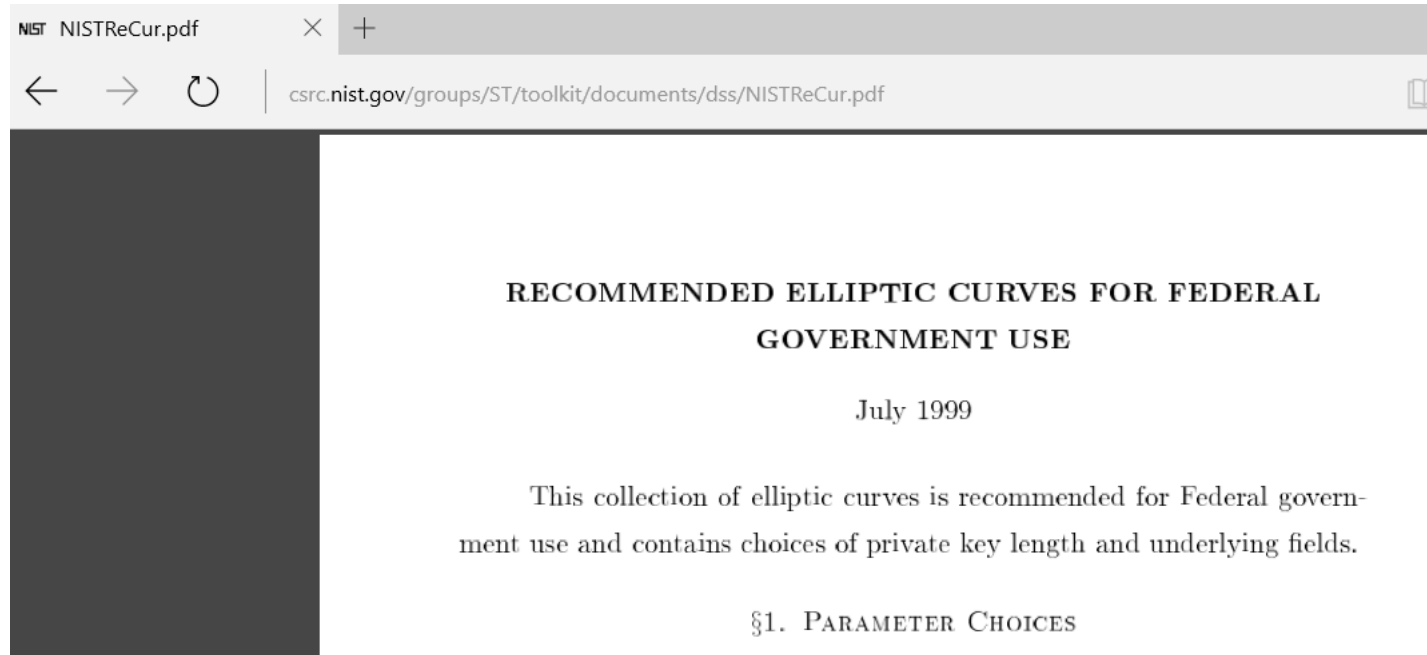
655456209464694; 93360682685816031704969423104727624468251177438749706128879957701\93698826859762790479113062308975863428283798589097017957365590672\835713863895712246676094993008985548024464030395443007480025079620363866193152298860635410053224484639158979864121027372558373965\48653931285483865070903191974204864923589439190352993032676961005\08840431979272991603892747747094094858192679116146502863521484987\08623286193422239171712154568612530067276018808591500424849476686\706784051068715397706852664532638332403983747338379697022624261377163163204493828299206039808703403575100467337085017748387148822224875309641791879395483731754620034884930540399950519191679471224\0555855709321935074711577569598163700850920394705281936392411084\4360068618352846572496956218643721497262583322544865996160464558\5462993701658947042526445624157899586972652935647856967092689604\42796501209877036845001246792761563917639959736383038665362727158

$$g^{ab} =$$

330166919524192149323761733598426244691224199958894654036331526394350099088627302979833339501183059198113987880066739419999231378970715307039317876258453876701124543849520979430233302775032650107245135512092795731832349343596366965069683257694895110289436988215186894965977582185407675178858364641602894716513645524907139614566085360133016497539758756106596557555674744381803579583602267087423481750455634370758409692308267670340611194376574669939893893482895996003389503722513369326735717434288230260146992320711161713922195996910968467141336433827457093761125005143009836512019611866134642676859265636245898172596372485581049036573719816844170539930826718273452528414333373254200883800592320891749460865366649848360413340316504386926391062876271575757583831289710534010374070317315095828076395094487046179839301350287596589383292751933079161318839043121329118930009948197899907586986108953591420279426874779423560221038468



NIST Curve P-256



Curve P-256

$p = 11579208921035624876269744694940757353008614 \backslash$
3415290314195533631308867097853951

$r = 11579208921035624876269744694940757352999695 \backslash$
5224135760342422259061068512044369

$s = c49d3608\ 86e70493\ 6a6678e1\ 139d26b7\ 819f7e90$

$c =$ 7efba166 2985be94 03cb055c
75d4f7e0 ce8d84a9 c5114abc af317768 0104fa0d

$b =$ 5ac635d8 aa3a93e7 b3ebbd55
769886bc 651d06b0 cc53b0f6 3bce3c3e 27d2604b

$G_x =$ 6b17d1f2 e12c4247 f8bce6e5
63a440f2 77037d81 2deb33a0 f4a13945 d898c296

$G_y =$ 4fe342e2 fe1a7f9b 8ee7eb4a
7c0f9e16 2bce3357 6b315ece cbb64068 37bf51f5

§2. CURVES OVER PRIME FIELDS

For each prime p , a pseudo-random curve

$$E : y^2 \equiv x^3 - 3x + b \pmod{p}$$

ECDH key exchange (1999 – nowish)

$$p = 2^{256} - 2^{224} + 2^{192} + 2^{96} - 1$$

$p = 115792089210356248762697446949407573530086143415290314195533631308867097853951$

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

$\#E = 115792089210356248762697446949407573529996955224135760342422259061068512044369$

$P = (48439561293906451759052585252797914202762949526041747995844080717082404635286, 36134250956749795798585127919587881956611106672985015071877198253568414405109)$



$a =$

89130644591246033577639
77064146285502314502849
28352556031837219223173
24614395

$[a]P = (84116208261315898167593067868200525612344221886333785331584793435449501658416, 102885655542185598026739250172885300109680266058548048621945393128043427650740)$

$[b]P = (101228882920057626679704131545407930245895491542090988999577542687271695288383, 77887418190304022994116595034556257760807185615679689372138134363978498341594)$

$[ab]P = (101228882920057626679704131545407930245895491542090988999577542687271695288383, 77887418190304022994116595034556257760807185615679689372138134363978498341594)$

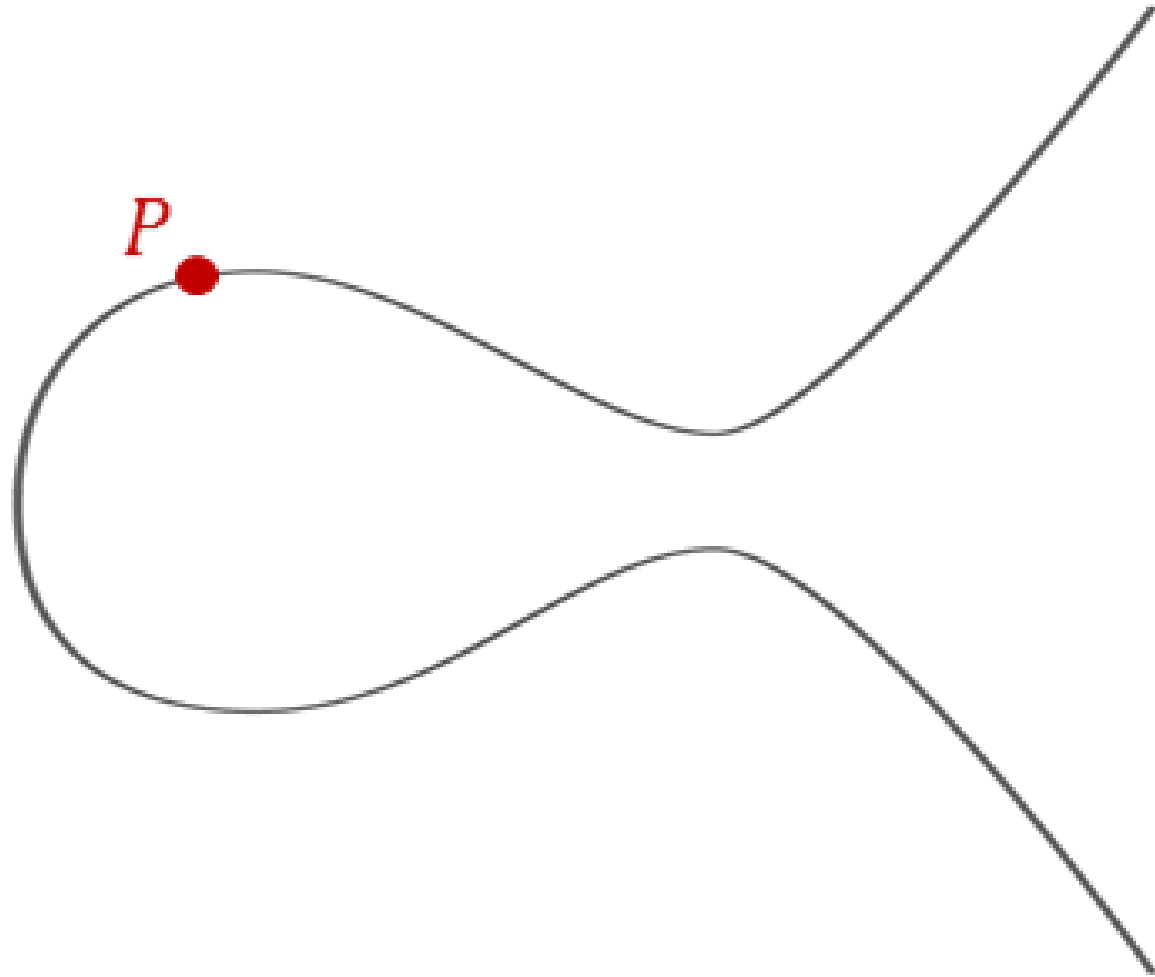


$b =$

10095557463932786418806
93831619070803277191091
90584053916797810821934
05190826

The fundamental ECC operation

$$P, k \mapsto [k]P$$



Scalar multiplications via double-and-add

How to (naively) compute $k, Q \mapsto [k]Q$?

$$P \leftarrow Q$$

$$k = (k_n, k_{n-1}, \dots, k_0)_2$$

for i from $n - 1$ downto 0 do

$$P \leftarrow [2]P$$

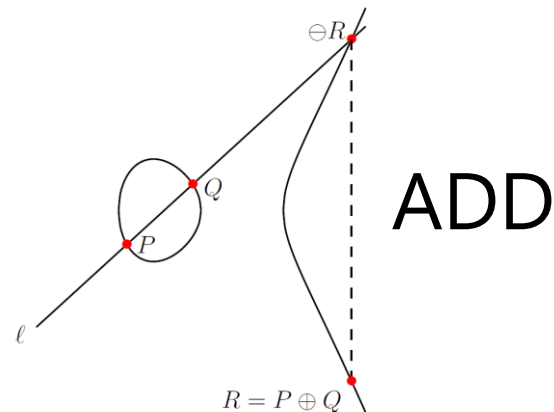
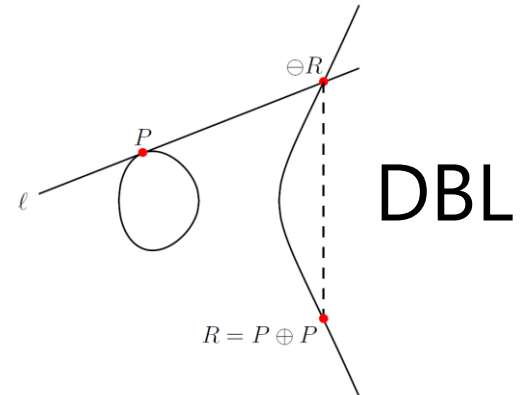
if $k_i = 1$ then

$$P \leftarrow P \oplus Q$$

end if

end for

return $P (= [k]Q)$



Scalar multiplications via double-and-add

How to (naively) compute $k, Q \mapsto [k]Q$?

$$P \leftarrow Q$$

$$k = (k_n, k_{n-1}, \dots, k_0)_2$$

for i from $n - 1$ downto 0 do

$$P \leftarrow [2]P$$

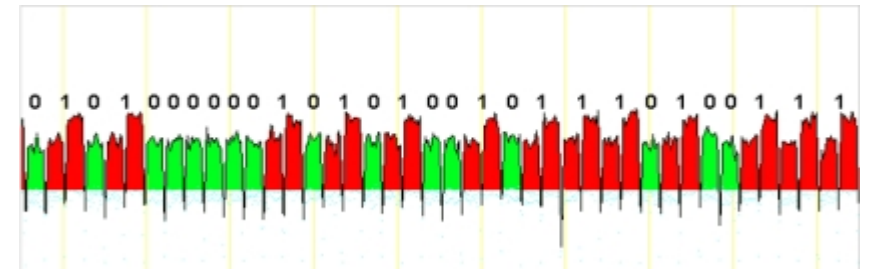
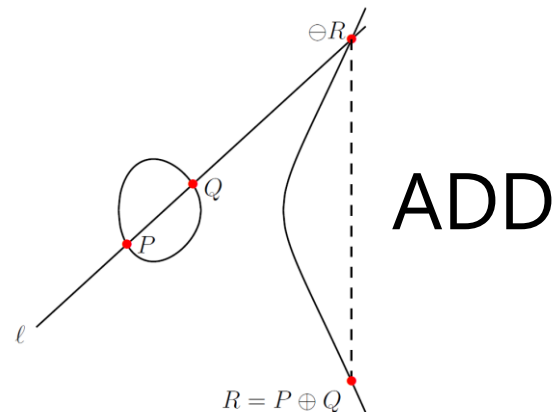
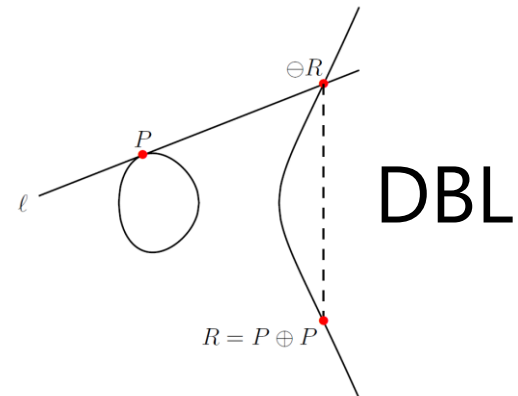
if $k_i = 1$ then

$$P \leftarrow P \oplus Q$$

end if

end for

return $P (= [k]Q)$



Scalar multiplications via double-and-add

How to compute $k, Q \mapsto [k]Q$ on $y^2 = x^3 + ax + b$?

$$k = (k_n, k_{n-1}, \dots, k_0)$$

$$(x_P, y_P) \leftarrow Q$$

for i from $n - 1$ downto 0 do

$$\lambda \leftarrow (3x_P^2 + a)/(2y_P); \quad v \leftarrow y_P - \lambda x_P;$$

$$x_P \leftarrow \lambda^2 - 2x_P; \quad y_P \leftarrow -(\lambda x_P + v);$$

if $k_i = 1$ then

$$\lambda \leftarrow (y_P - y_Q)/(x_P - x_Q); \quad v \leftarrow y_P - \lambda x_P;$$

$$x_P \leftarrow \lambda^2 - x_P - x_Q; \quad y_P \leftarrow -(\lambda x_P + v)$$

end for

$$\text{return } (x_P, y_P) = [k](x_Q, y_Q)$$

Projective space

- Recall we defined the group of K -rational points as

$$E_{a,b}(K) = \{(x, y): y^2 = x^3 + ax + b\} \cup \{\infty\}$$

- The *natural habitat* for elliptic curve groups is in $\mathbb{P}^2(K)$, not $\mathbb{A}^2(K)$
- For (easiest) example, rather than $(x, y) \in \mathbb{A}^2$, take $(X:Y:Z) \in \mathbb{P}^2$ modulo the equivalence $(X:Y:Z) \sim (\lambda X : \lambda Y : \lambda Z)$ for $\lambda \in K^*$
- Replace x with X/Z and y with Y/Z , so $E_{a,b}(K)$ is the set of solutions $(X:Y:Z) \in \mathbb{P}^2(K)$ to

$$E : Y^2Z = X^3 + aXZ^2 + bZ^3$$

- So the affine points (x, y) from before become $(x : y : 1) \sim (\lambda x : \lambda y : \lambda)$ and the point at infinity is the unique point with $Z = 0$, i.e., $(0 : 1 : 0) \sim (0 : \lambda : 0)$

Projective space, cont.

- One practical benefit of working over \mathbb{P}^2 is that the explicit formulas for computing \oplus become much faster, by avoiding field inversions
- Thus, the fundamental ECC operation $k, P \mapsto [k]P$ becomes much faster...

$$(x', y') = [2](x, y)$$

$$\lambda \leftarrow (3x^2 + a)/(2y);$$

$$x' \leftarrow \lambda^2 - 2x;$$

$$y' \leftarrow -(\lambda(x' - x) + y);$$

$$1S + 2M + 1I$$

$$(X' : Y' : Z') = [2](X : Y : Z)$$

$$X' = 2XY((3X^2 + aZ^2)^2 - 8Y^2XZ)$$

$$Y' = (3X^2 + aZ^2)(12Y^2XZ - (3X^2 + aZ^2)^2) - 8Y^4Z^2$$

$$Z' = 8Y^3Z^3$$

$$5M + 6S$$

Projective scalar multiplications

How to compute $k, Q \mapsto [k]Q$ on $y^2 = x^3 + ax + b$?

$$k = (k_n, k_{n-1}, \dots, k_0)$$

$$(X_P : Y_P : Z_P) \leftarrow Q$$

for i from $n - 1$ downto 0 do

$$(X_P : Y_P : Z_P) \leftarrow [2](X_P : Y_P : Z_P) \quad 5M + 6S$$

if $k_i = 1$ then

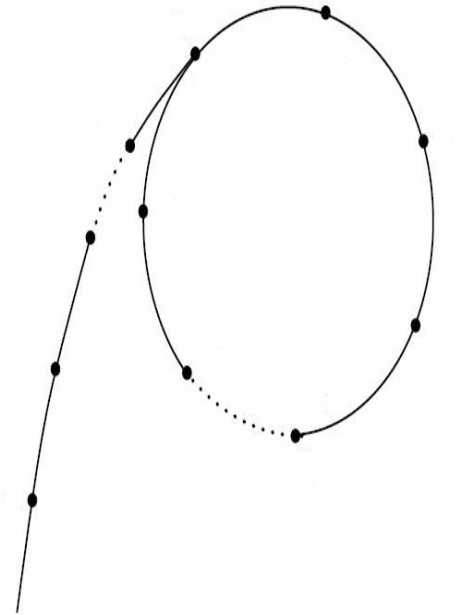
$$(X_P : Y_P : Z_P) \leftarrow (X_P : Y_P : Z_P) \oplus (X_Q : Y_Q : Z_Q) \quad 9M + 2S$$

end for

$$\text{return } (x_P, y_P) \leftarrow (X_P/Z_P, Y_P/Z_P) \quad 1I + 2M$$

ECDLP security and Pollard's rho algorithm

- ECDLP: given $P, Q \in E(\mathbb{F}_p)$ of prime order N , find k such that $Q = [k]P$
- Pollard'78: compute pseudo-random $R_i = [a_i]P + [b_i]Q$ until we find a collision $R_i = R_j$ with $b_i \neq b_j$, then $k = (a_j - a_i)/(b_i - b_j)$
- Birthday paradox says we can expect collision after computing $\sqrt{\pi n/2}$ group elements R_i , i.e., after $\approx \sqrt{N}$ group operations. So 2^{128} security needs $N \approx 2^{256}$
- The best known ECDLP algorithm on (well-chosen) elliptic curves remains generic, i.e., elliptic curves are as strong as is possible



Index calculus on elliptic curves?

[Miller, 85] : "it is extremely unlikely that an index calculus [...] will ever be able to work"

Consider E/\mathbb{F}_{1217} : $y^2 = x^3 - 3x + 139$

$$\#E(\mathbb{F}_{1217}) = 1277$$

$$P = (3,401) \text{ and } Q = (192,847)$$

ECDLP: find k such that $[k]P = Q$

Regardless of factor base, can't efficiently decompose elements!

e.g., factor base $R_i = \{(3,401), (5,395), (7,73), (11,252), (13,104), (19,265)\}$

Writing $S = \sum [k_i]R_i$ involves solving discrete logarithms, compare this to integers **mod** p where we lift and factorise over the integers

Part 1: Motivation

Part 2: Elliptic Curves

Part 3: Elliptic Curve Cryptography

Part 4: Next-generation ECC

What's wrong with old school ECC?

- **Side-channel attacks:** starting with Kocher'99, side-channel attacks and their countermeasures have become extremely sophisticated
- **Decades of new research:** we now know much better/faster/simpler/safer ways to do ECC
- **Suspicion surrounding previous standards:** Snowden leaks, dual EC-DRBG backdoor, etc., lead to conjectured weaknesses in the NIST curves

Next generation elliptic curves

- 2014: CFRG receives formal request from TLS working group for recommendations for new elliptic curves
- 2015: NIST holds workshop on ECC standards
- 2015: CFRG announces two chosen curves, both specified in Montgomery (1987) form

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

- Bernstein's Curve25519 [2006]: $p = 2^{255} - 19$ and $A = 486662$
- Hamburg's Goldilocks [2015]: $p = 2^{448} - 2^{224} - 1$ and $A = 156326$
- Both primes offer fast software implementations!
- Their group orders are divisible by 8 and 4, but this form offers several advantages.

Montgomery's fast differential arithmetic

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

- drop the y -coordinate, and work with x -only.
- projectively, work with $(X : Z) \in \mathbb{P}^1$ instead of $(X : Y : Z) \in \mathbb{P}^2$
- But (pseudo-)addition of $\mathbf{x}(P)$ and $\mathbf{x}(Q)$ requires $\mathbf{x}(Q \ominus P)$

Extremely fast pseudo-doubling: **xDBL**

$$X_{[2]P} = (X_P + Z_P)^2 (X_P - Z_P)^2$$

$2M + 2S$

$$Z_{[2]P} = 4X_P Z_P ((X_P - Z_P)^2 + (A + 2)X_P Z_P)$$

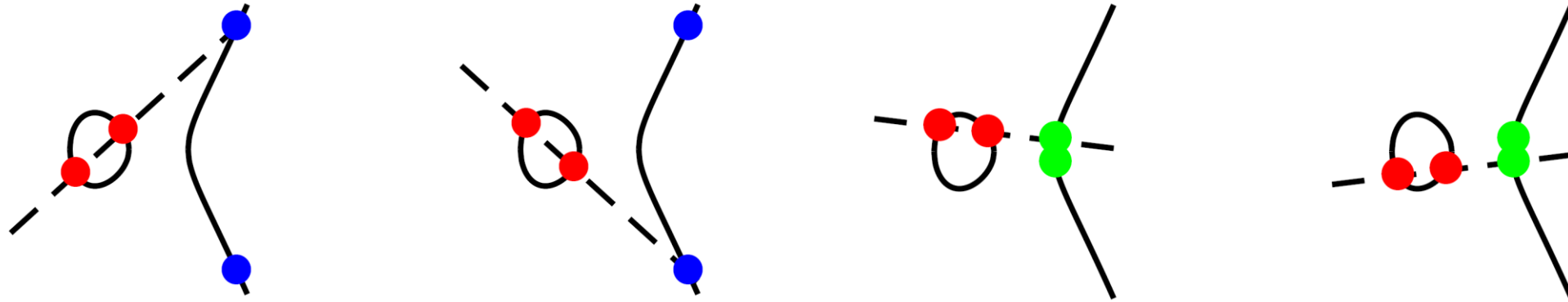
Extremely fast pseudo-addition: **xADD**

$$X_{P+Q} = Z_{P-Q} \left[(X_P - Z_P)(X_Q + Z_Q) + (X_P + Z_P)(X_Q - Z_Q) \right]^2$$

$4M + 2S$

$$Z_{P+Q} = X_{P-Q} \left[(X_P - Z_P)(X_Q + Z_Q) - (X_P + Z_P)(X_Q - Z_Q) \right]^2$$

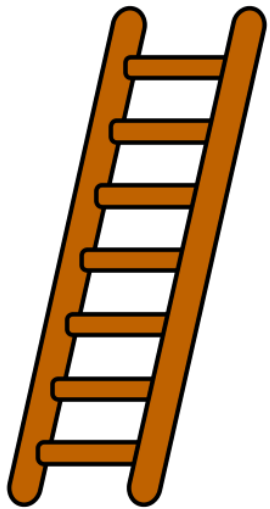
Differential additions and the Montgomery ladder



- Given only the x -coordinates of two points, the x -coordinate of their sum can be two possibilities
- Inputting the x -coordinate of the *difference* resolves ambiguity
- The (ingenious!) Montgomery ladder fixes all *differences* as the input point: in $k, x(P) \mapsto x([k]P)$, every **xADD** is of the form
$$\mathbf{xADD}(x([n+1]P), x([n]P), x(P))$$
- We carry two multiples of P "up the ladder": $x(Q)$ and $x(Q \oplus P)$
- At i^{th} step: compute $x([2]Q \oplus P) = \mathbf{xADD}(x(Q \oplus P), x(Q), x(P))$
- At i^{th} step: pseudo-double (**xDBL**) one of them depending on k_i

Fast, compact, simple, safer Diffie-Hellman

- Write $k = \sum_{i=0}^{\ell-1} k_i 2^i$ with $k_{\ell-1} = 1$ and $P = (x_P, y_P)$ in E (e.g., on Curve25519 or Goldilocks)



```
(x0, x1) ← (xDBL(xP), xP)
for i = ℓ - 2 downto 0 do
  (x0, x1) ← cSWAP((ki+1 ⊗ ki), (x0, x1))
  (x0, x1) ← (xDBL(x0), xADD(x0, x1, xP))
end for
(x0, x1) ← cSWAP(k0, (x0, x1))
return x0 (= x[k]P)
```

Inherently uniform, much easier to implement in constant-time

- x -only Diffie-Hellman (Miller'85): $x([ab]P) = x([a]([b]P)) = x([b]([a]P))$

see <https://tools.ietf.org/html/rfc7748>

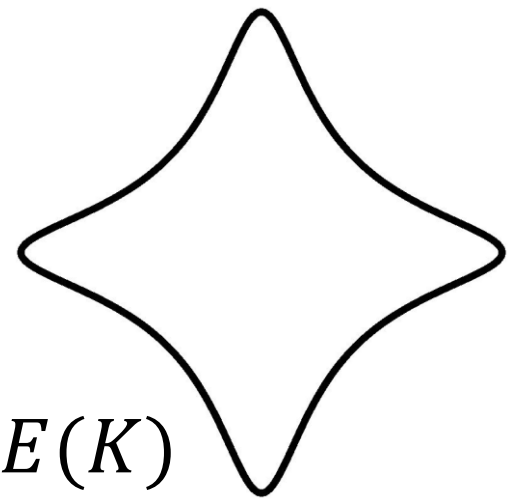
(Elliptic curves for security)

Curve25519 and Goldilocks in the real world

- See “Elliptic curves for security” <https://tools.ietf.org/html/rfc7748>
- Both curves integrated into TLS ciphersuites
- In 2014, OpenSSH defaults to Curve25519
- Curve25519 is used in Signal Protocol (Facebook Messenger, Google Allo, WhatsApp), iOS, GnuPG, etc (<https://en.wikipedia.org/wiki/Curve25519>)

(Twisted) Edwards curves

$$E : ax^2 + y^2 = 1 + dx^2y^2$$

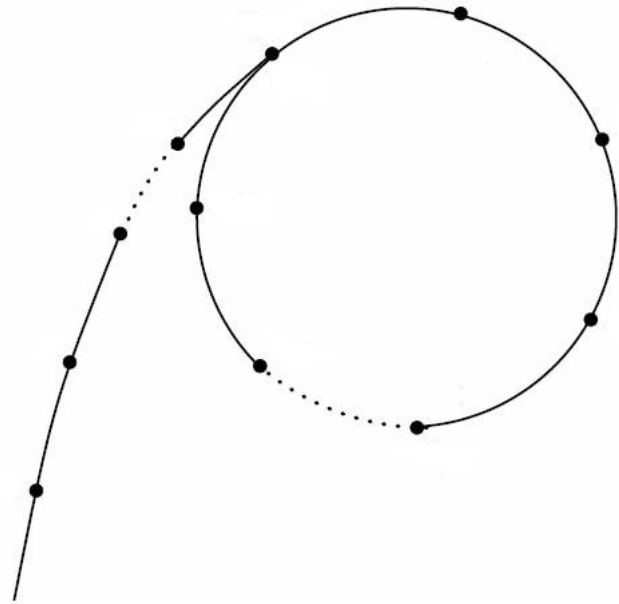


- Neutral element is $(0,1)$ - no projective space needed for $E(K)$
- Addition law is *complete* (for well-chosen E)

$$(x_1, y_1) + (x_2, y_2) = \left(\frac{x_1y_1 + x_2y_2}{y_1y_2 - x_1x_2}, \frac{x_1y_1 - x_2y_2}{x_1y_2 - y_1x_2} \right)$$

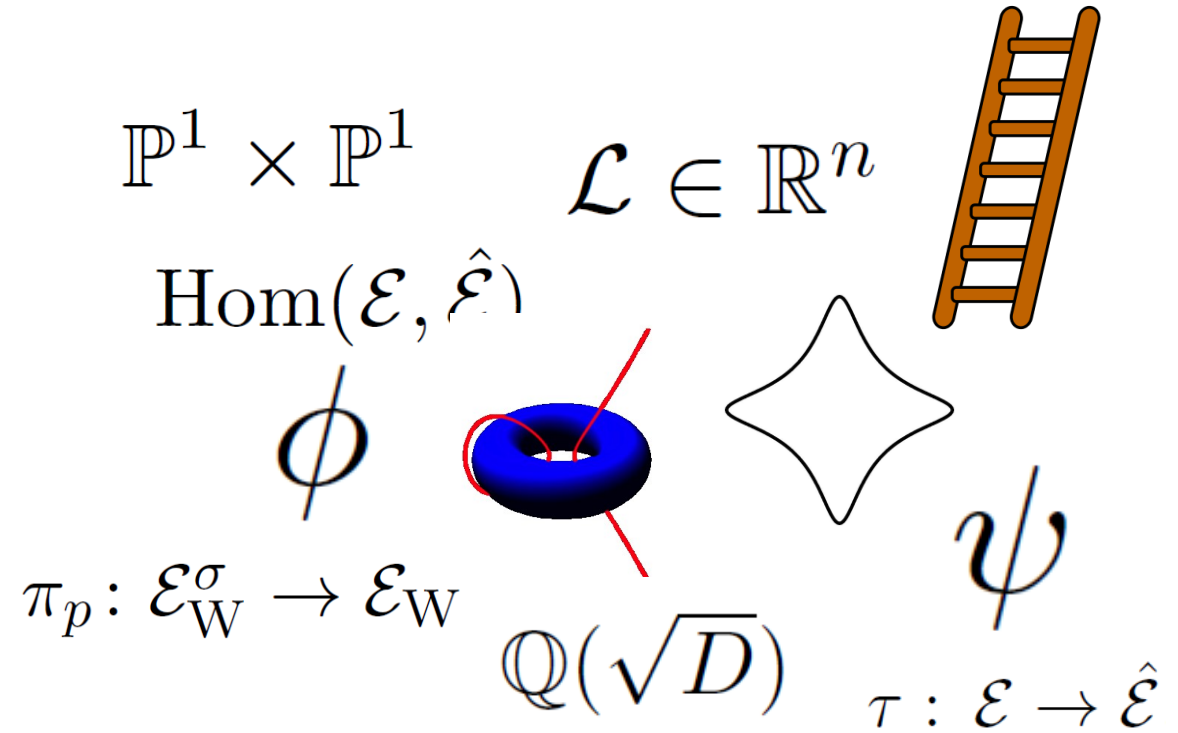
- Extremely fast: **8M!** Also works for doubling, inverses, everything
- Fast, simple, exception-free implementations that always compute correctly
- Also birationally equivalent to Montgomery curves!

Elliptic curves: the best of both worlds



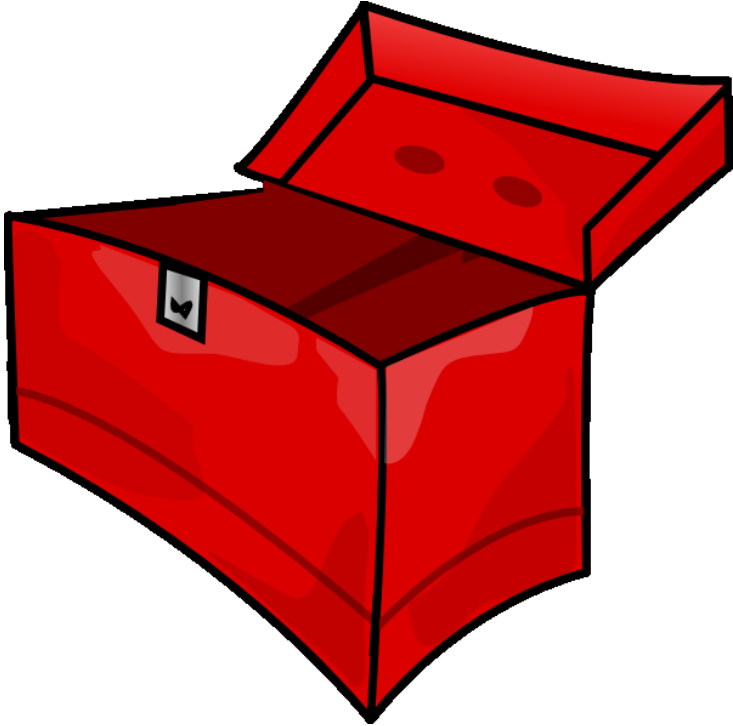
attacker: generic

vs.



us: not generic

ECC is the best of both worlds



attacker's toolbox



vs.



our toolbox

Questions?