Groups and ECC

Nicolas T. Courtois
- University College of London

Integers

Number Theory

30 years ago mathematicians used to say “Number Theory” will be probably last branch of mathematics that will ever find any practical application….

Basic number theory (like RSA) is easy and fun.

HOWEVER with elliptic curves, the potential number of people that understand the mathematics behind shrinks to a few dozen of elite academics worldwide…

Highly problematic…

Integers

Natural Integers - IN: [0],1,2,3…

Relative Integers - ZZ. -2,-1,0,1,2,3…

Prime number: has no "proper" divisor,

(means except 1 and itself.)

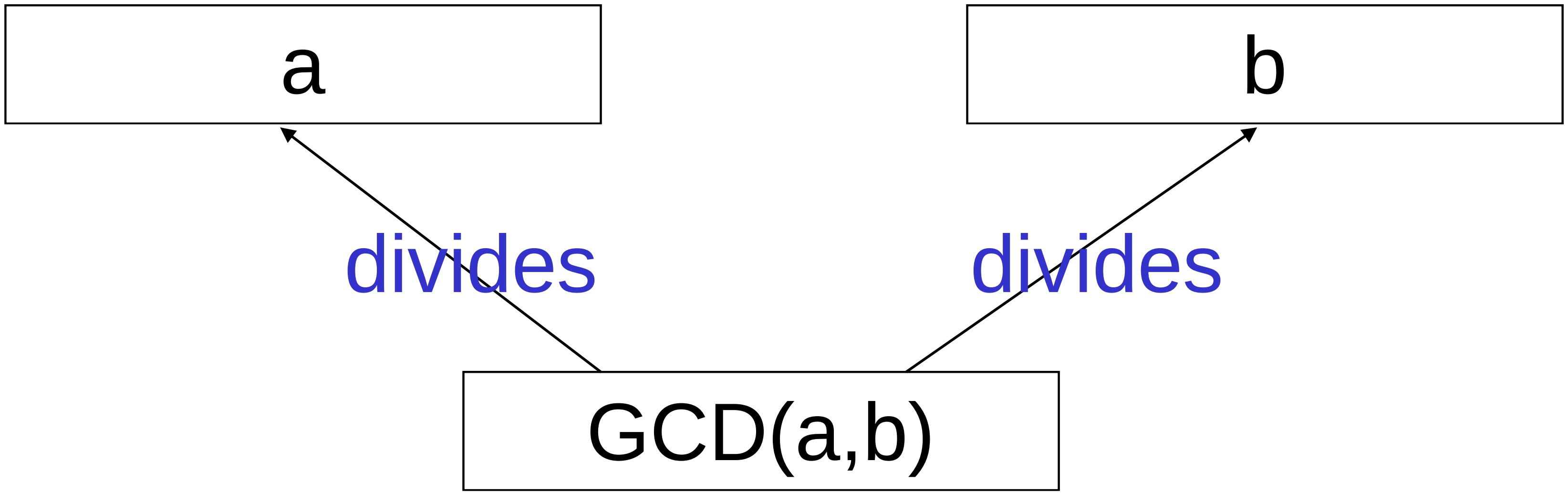
Thm. There is an infinite number of prime numbers.

Easy fact: Each number decomposes in prime factors.

Hard thing: to compute factors of a given number. Currently feasible up to some 800 bits.

Greatest Common Divisor

the biggest such that



Relatively Prime Numbers

$$1=GCD(a,b)$$

"Fundamental Theorem of Arithmetic"

Every integer has a unique decomposition in prime factors.

$$n = p_1^{a\_1} * p_2^{a\_2} * \ldots * p_k^{a\_k}$$

with $p_1 < p_2 < \ldots$.

Nicolas T. Courtois, 2006-2014

Groups

Evariste Galois

Very famous French mathematician.

- At age of 14 started reading very serious books papers about algebra and mathematics.
- For reasons that are not fully explained failed all his exams to enter Ecole Polytechnique and most of his brilliant work was published and recognised only later.
- Did completely solve the problem of solvability of polynomial equations in one variable: “Galois Theory”.
- Was a political activist, against the king of France, frequently arrested and writing math papers while in prison.
- Died at the age of 20 after a fatal duel with an artillery officer, to some in the context of a broken love affair, to some stage-managed by the royalist fractions and the police.

He was the first to use the word Group.

Group - Definition

0. closure

A set M with an operation
$\bullet$: $M \times M \rightarrow M$ such that:

1) Operation $\bullet$ is associative
2) Has an identity element 1.

$1 \bullet a = a \bullet 1 = a$

3) Each element a has an inverse called a^{-1}.

$a^{-1} \bullet a = a \bullet a^{-1} = 1$

$\left.\text{1)}\right\}$ semi-group

$\left.\text{semi-group} + \text{2)}\right\}$ monoid

$\left.\text{monoid} + \text{3)}\right\}$ group

*Pre-Cryptographic Interpretation of Groups and Monoids

One interpretation is as follows:

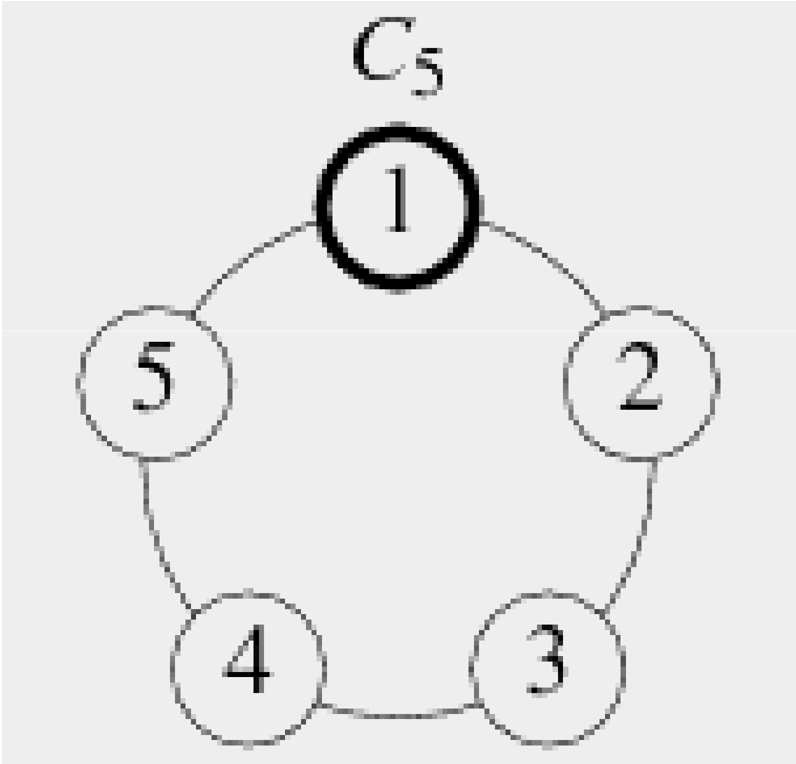
- Each element = transformation on
  the “message space” = a set M.
- Neutral element:
  transformation that does nothing.
- Inverse:
  decrypt a “scrambled” message. Don’t call it encryption (would imply that this is actually somewhat “secure”…
- Group: we always have an inverse:
  every message can be decrypted.
  - (though in crypto we can relax/work around this requirement a lot…)

Abelian == Commutative Groups

a+b = b+a

Cyclic Groups

One element => generates the whole group.



Modular Addition - Congruencies

DEFINITION:

We say that $a \equiv b \bmod n$

if n divides $a-b$.

Congruencies - Properties

Equivalence Relation:

1. Reflexive: $a \equiv a$
2. Symmetric $a \equiv b$ if and only if $b \equiv a$
3. Transitive $a \equiv b$ and $b \equiv c$ implies $a \equiv c$.

Every equivalence relations partitions the set into equivalence classes.

Every congruence mod n partitions ZZ into classes == residues mod n usually represented by numbers {0,1,2,..,n-1}.

Congruencies - Properties

The set of residue classes modulo n is called
Z_n or ZZ / nZZ.

Elements of Z_n of are denoted {0,1,2,..,n-1}.
(This is possible because {0,1,2,..,n-1} is a complete set of representative elements.)

Fact:

Usual integer operations (+,*) and special elements (0,1) translate to the world of residue classes.

Example: if $a \equiv b \bmod n$ and $c \equiv d \bmod n$
THEN $a*c \equiv b*d \bmod n$.

Modular Addition - Congruencies

$a \equiv b \bmod n$

Frequently we simply write

$a=b \bmod n$.

This “equality” is a real equality

in we think in terms of residues mod n and addition modulo n in the set of residues:

$\{0,1,2,...,n-1\}$.

Group – Example 1

Let n>=2.

{0,1,2,…,n-1},+ mod n is a group.

The group of residue classes mod n
represented by {0,1,2,..,n-1}.

Group – Example 1.A.

$\{0,1\},+ \bmod 2$ is a group.

Proof:

-modular addition is always associative.

-identity element: $a+0=0+a=a$.

-we have $-a = a$ here !

Group – Example 1.B.

{0,1,2},+ mod 3 is a group.

BTW: here -a=2a as we have: a+2a=0 mod 3.

Group ?

Fact:

{1,2,..,n-1},* mod n is a group IF AND ONLY IF n is a prime.

We call Z_n* the set {1,2,..,n-1}.

Group – Example 2.A.

{1,2}, * mod 3 is a group.

Proof:

-associative.

-identity element $1*a=a*1=a \bmod 3$.

-inverse: $1^{-1}=1$, $2^{-1}=2$

- check that $2*2=4 \bmod 3 = 1$.

(Actually this group is the same (ISOMORPHIC) to {0,1},+.)

Order of a Group, Subgroups

We call order of G or ord(G) the number of elements in the group (its cardinality).

A sub-group: any subset closed under * that is a group for the same *.

Theorem [Lagrange]: Order of a subgroup $H \subseteq G$ divides the order of the group G.

Order of an Element

Order of an element g: $ord(g)$ =

- Cardinal of the sub-group generated by g: $g^1, g^2, g^3, \ldots g^{ord(g)-1}, 1$.
- It is also the smallest integer such that $g^{ord(g)}=1$.

Theorem: Order of an element
divides the order of the group.

Fermat's Little Theorem

Pierre de Fermat [1601-1655]:

French lawyer and government official,
one of the fathers of number theory
(also involved in breaking enemy ciphers and codes).

Theorem: Let p be a prime.

For any integer $a^p = a \bmod p$.

Corollary: If $a \neq 0 \bmod p$, then $a^{p-1} = 1 \bmod p$.

Example 2.B.

{1,2,3}, * mod 4 is NOT a group.

Proof: 2 has no inverse.

Group ?

{1,2,..,n-1}, * mod n is a group IF AND ONLY IF n is a prime.

Rings

Rings

"When two operations work together nicely" like + and *.

(R,+,*,0,1) is a Ring if:

- $0 \neq 1$ (not serious, avoids one "trivial" ring {0},+,*)
- R,+ is an Abelian group
- R\{0}, * is a monoid with identity element 1.
- * distributes over +:

$$a(b+c)=ab+ac$$

$$(b+c)a=ba+ca$$

Fields

in modern usage always commutative

Fields [Abel, Galois]

<u>2 added requirements:</u>

- commutative
- each element $a \neq 0$ has an inverse

<u>Corollary:</u> When p is prime, Z_p is a field.

Example 2.B.

({0,1,2,3}, + mod 4, * mod 4) is a ring.

It is NOT a field.

Why ?

Example 2.B.

({0,1,2,3}, + mod 4, * mod 4) is a ring.

It is NOT a field.

Why ?

Proof: 2 has no inverse.

Fields vs. Rings

- In a field we have all the 4 arithmetic operations +,-,*,/.
- In a ring we do not have /.

Z_n^*

We call Z_n^* the set of the invertible elements mod n.

Theorem: it is a group under *.

When n is a prime, $Z_n^* = Z_n \backslash \{0\}$.

Otherwise it is even smaller set,
other elements are excluded.

How many?

Leonhard Euler

Swiss mathematician [1707-1783],
10 Swiss franks bills,

Have published some 600 very clever papers.
No scientist has done as much…

Fig. 298. — Euler.

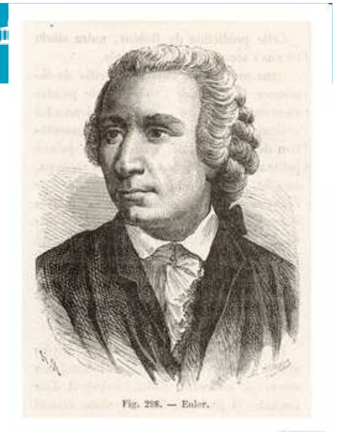
Euler Totient φ Function

Question 1: How many elements of Z_n are invertible ?

Question 2: How many integers between 0 and n-1 are relatively prime with n ?

Definition: this number is called $\varphi(n)$.

Euler Totient φ Function



How many elements of Z_n are invertible ?

$\varphi(n)$.

- $\varphi(p) = p-1$. For ANY prime (even if $p=2$).
- Prime powers:

$\varphi(p^a) = p^a - p^{a-1} = p^{a-1}(p-1)$.

Euler-Fermat Theorem

Theorem: GCD(a,n)=1
$a^{\varphi(n)} \equiv 1 \bmod n$.

Re-formulation in Z_n:
for each $a \in Z_n^*$ we have $a^{\varphi(n)} = 1$.

Finite Fields

Question:

$K=GF(p)=Z_p$, p prime.

For certain polynomials, $Z_p[X] / P(X)$ is a field.

Theorem: If and only if $P(X)$ is an irreducible polynomial.

Irreducible == has no proper divisor of lower degree.

Note: p is called the characteristic of this field.

$x+x+\ldots$ p times $= 0$.

Theorem:

ALL FINITE FIELDS are of the form $Z_p[X] / P(X)$, with p prime.

Corollary: the number of elements of a finite field is always $q=p^n$:

They are represented by all polynomials

$$a_0 + a_1 X^1 + \ldots + a_{n-1} X^{n-1}.$$

corresponds to all possible n-tuples

$$(a_0, a_1, \ldots, a_{n-1}).$$

Moreover

There is only “one” field that has $q=p^n$ elements: means that all finite fields that have q elements are isomorphic (and therefore have exactly the same properties).

Cycling

In Z_p we had $a^p = a$ [Fermat's Little Thm.]

In any finite field F that has q elements

$a^q = a$.

Theorem:

The multiplicative group of a finite field F is cyclic.

(in most cases false for Z_n* in general)

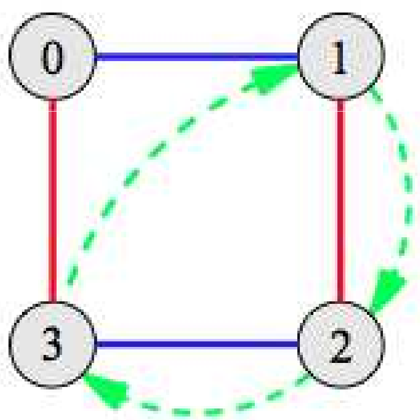
There is a generator element g, called primitive element, such that every element of the field F\{0} is a power of g.

(in fact there are MANY such elements).

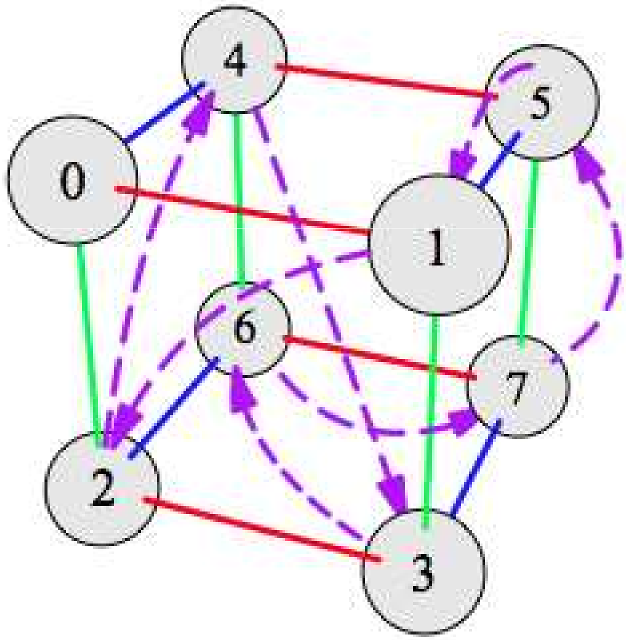
Theorem:

The multiplicative group of a finite field F is cyclic.

NOT OBVIOUS!



GF(4)



GF(8)

DL Problems

Various DL Problems

Discrete Logarithms

Given p, q, g and h find x such that

$$h = g^x \pmod{p}$$

where g is an element of order q in $\mathbb{F}_p$.

Elliptic Curve Discrete Logarithms

Given a curve E of order q over a field $\mathbb{F}_p$, and two points P and Q on the curve, find x such that

$$Q = [x]P.$$

*Slides from Jan 2013

Can take any (finite) group.

Bad Choices:

- Additive group $\mathbb{Z}$ or $\mathbb{F}_q$.
- Multiplicative group of or $\mathbb{C}$.

Apparently Good Choices:

- Finite fields $\mathbb{F}_q^*$. → **is it OK?**
- Elliptic curves over finite fields.
- Ideal class groups of number fields.
- Jacobian varieties of curves over finite fields.

Why NOT the cyclic group GF(p^m)?

find x such that

$$h = g^x$$ in GF(p^m)?

kind of broken…

DL in GF(p^m) is Broken!

A quasi-polynomial algorithm for discrete logarithm in finite fields of small characteristic

eprint/2013/400

Razvan Barbulescu[1], Pierrick Gaudry[1], Antoine Joux[2,3], and Emmanuel Thomé[1]

[1] Inria, CNRS, University of Lorraine, France
[2] Cryptology Chair, Foundation UPMC – LIP 6, CNRS UMR 7606, Paris, France
[3] CryptoExperts, Paris, France

quasi-polynomial…

Abstract The difficulty of computing discrete logarithms in fields $\mathbb{F}_{q^k}$ depends on the relative sizes of k and q. Until recently all the cases had a sub-exponential complexity of type $L(1/3)$, similar to the factorization problem. In 2013, Joux designed a new algorithm with a complexity of $L(1/4+\epsilon)$ in small characteristic. In the same spirit, we propose in this article another heuristic algorithm that provides a quasi-polynomial complexity when q is of size at most comparable with k. By quasi-polynomial, we mean a runtime of $n^{O(\log n)}$ where n is the bit-size of the input. For larger values of q that stay below the limit $L_{q^k}(1/3)$, our algorithm loses its quasi-polynomial nature, but still surpasses the Function Field Sieve.

Elliptic Curves

Curves?

In analysis curves can be defined by continuous parameterisations…

This continuous “close neighbourhood” approach does NOT work in finite fields (discrete points).

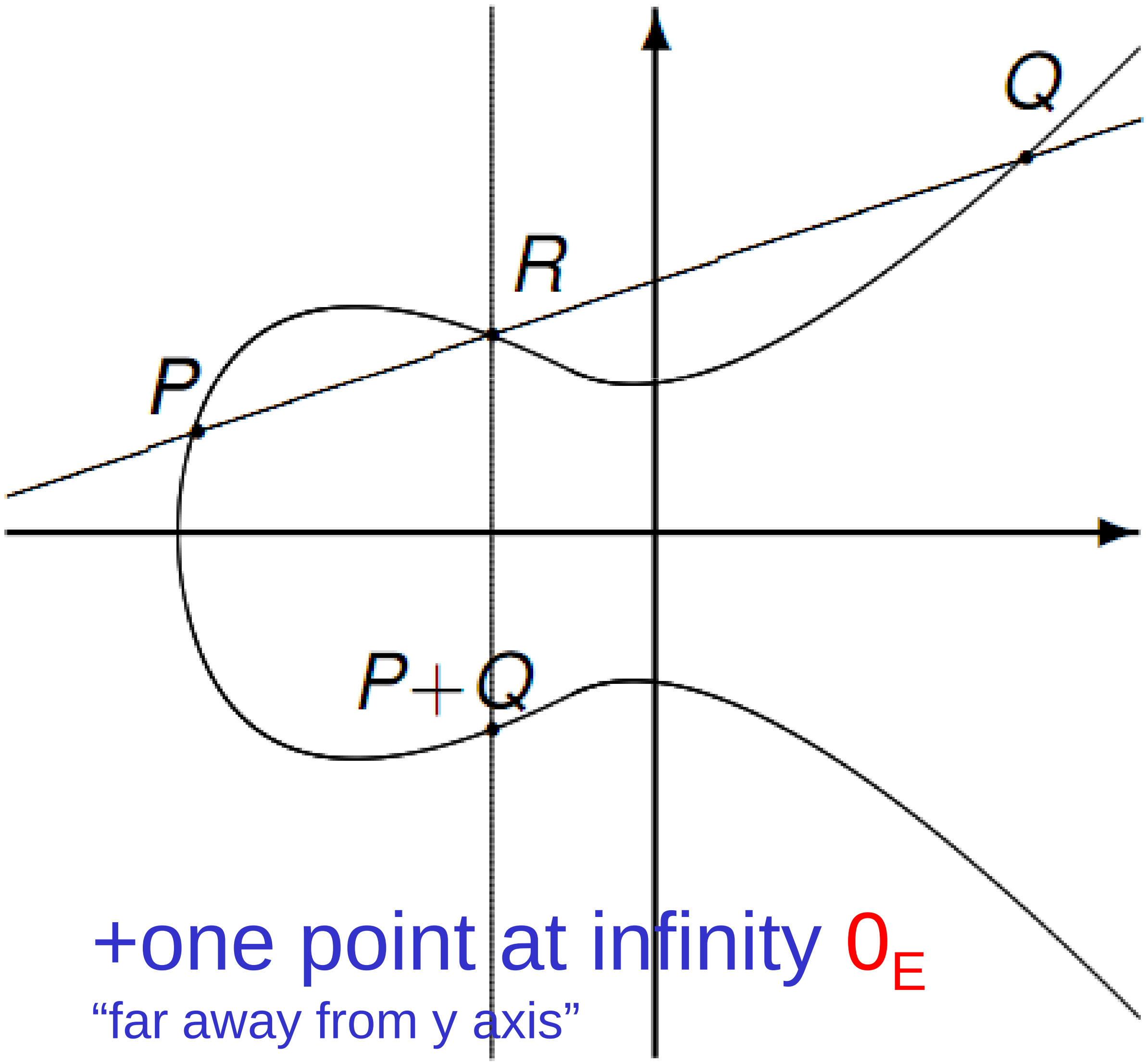
Curves are more generally defined by sets of points which satisfy a certain systems of equations for example $f(X,Y)=0$…

Over Real Numbers

1 dim curve
in 2 dim space

some pairs (x,y)
belong
to the curve

In practice the do
NOT LOOK
like this!



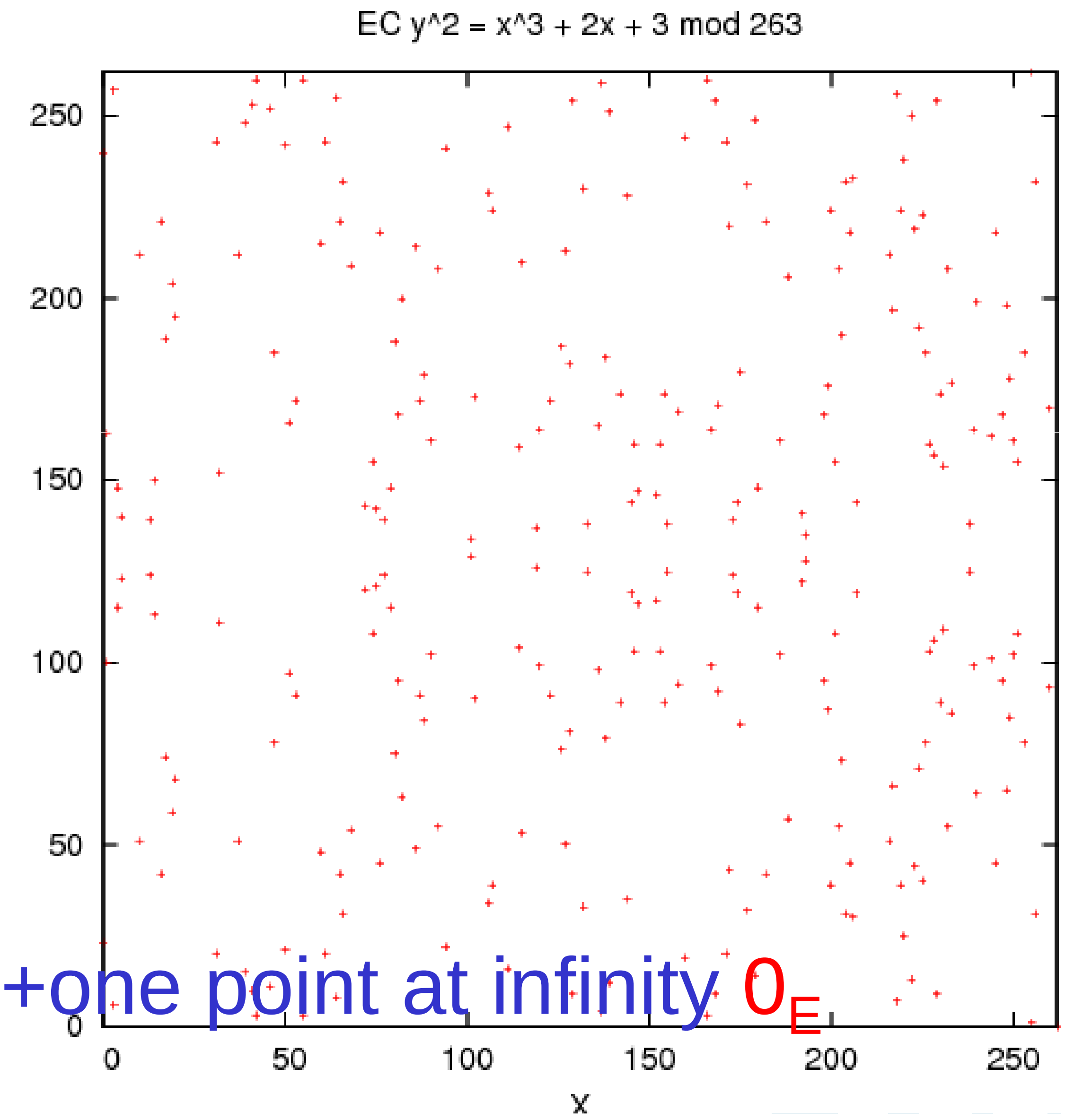
+one point at infinity 0_E
"far away from y axis"

Over Finite Fields – Say Prime Fields

1 dim curve
in 2 dim space

some pairs (x,y)
belong
to the curve

THEY RATHER
LOOK like this!

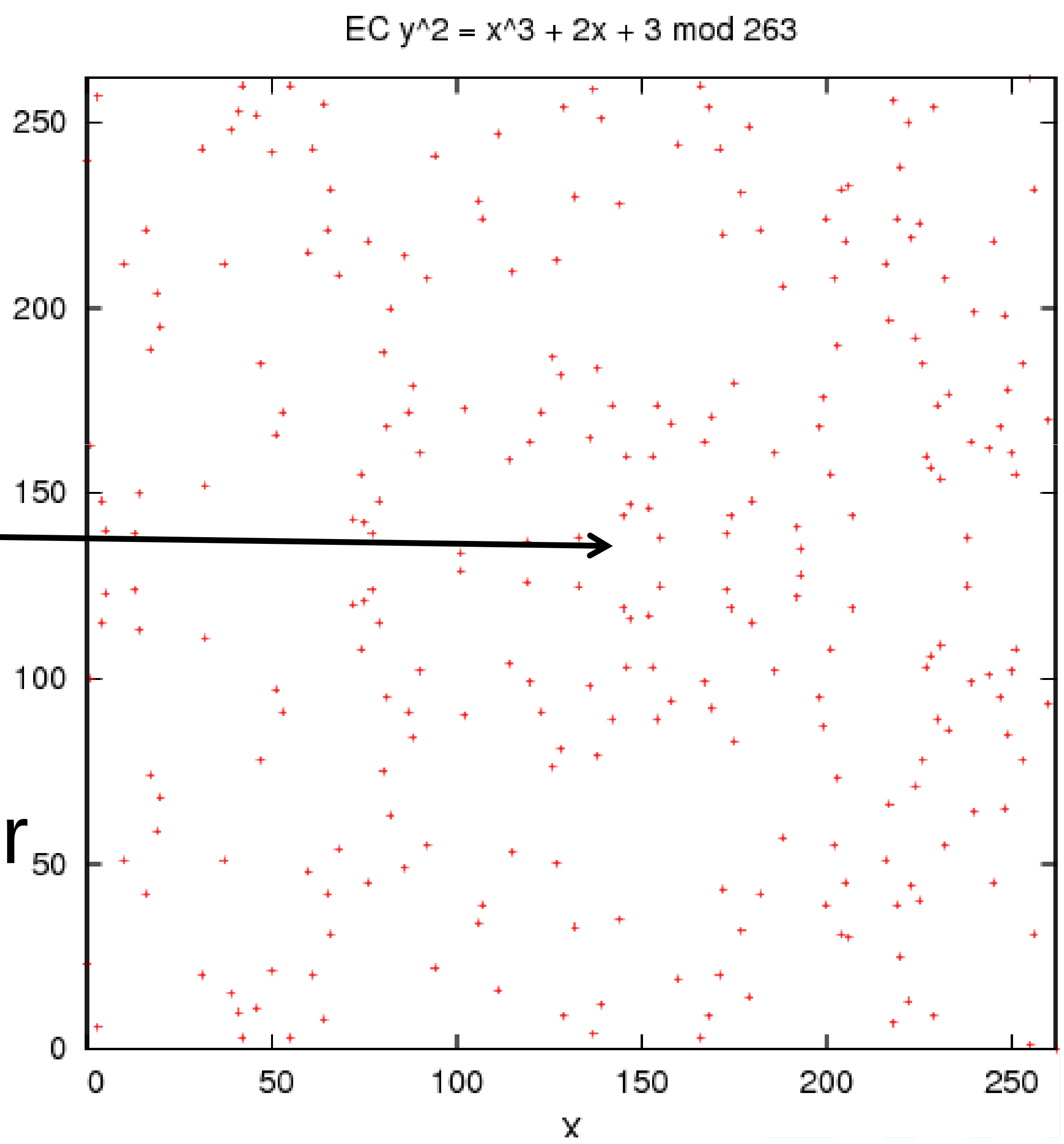


+one point at infinity 0_E

OWF Problem

We need a cyclic group
which is “complicated”
yet we can do
computations in it.

$x=0\ldots p-1 \longrightarrow [x]G$



p = prime

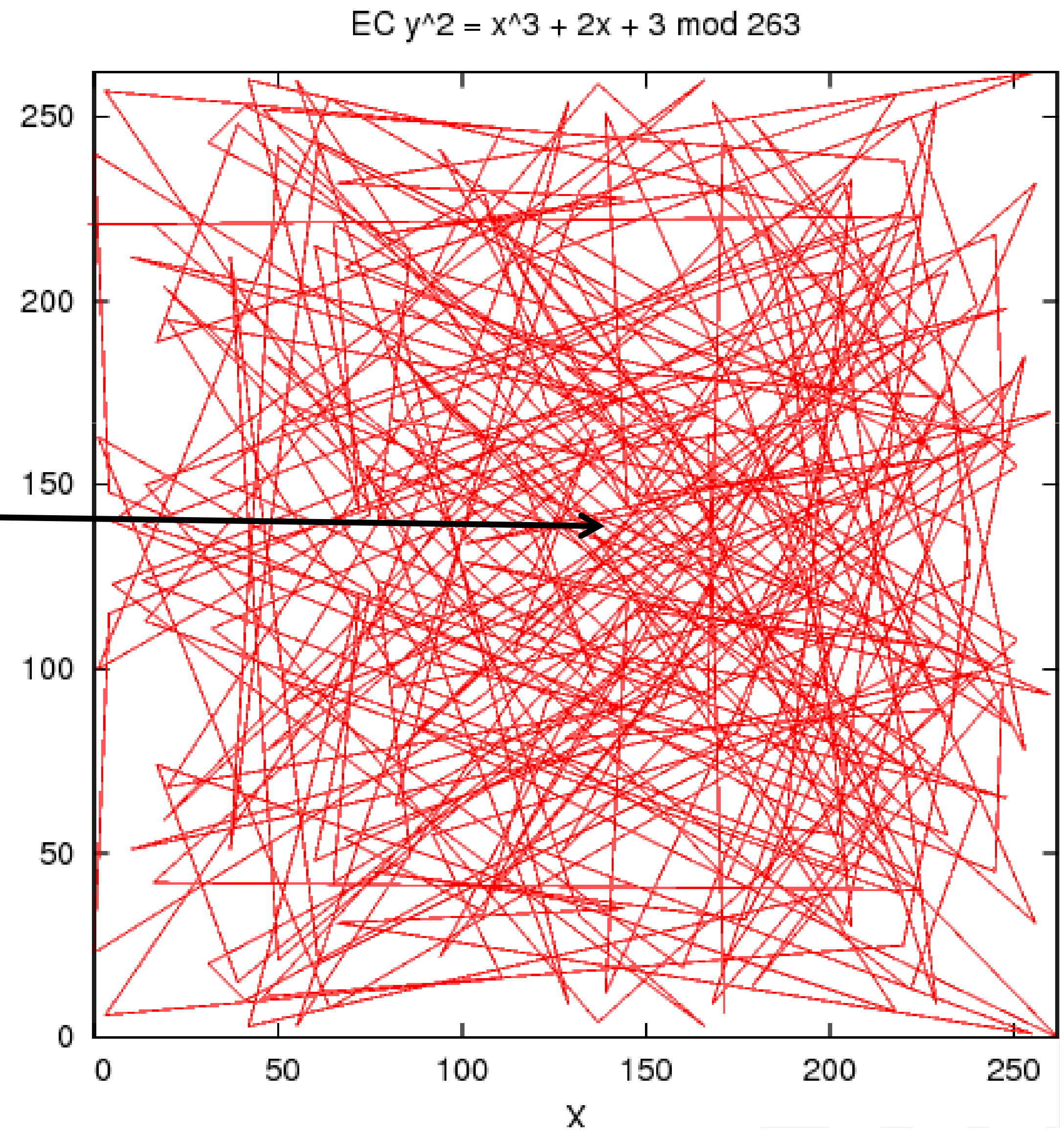
G = generator point

OWF Problem

Straight Order (a cycle)
=>
"Complicated" Order

[x]G

x=0…p-1



Elliptic Curves vs. RSA
Key Sizes

ECC Inventors

Neal Koblitz and Victor Miller,

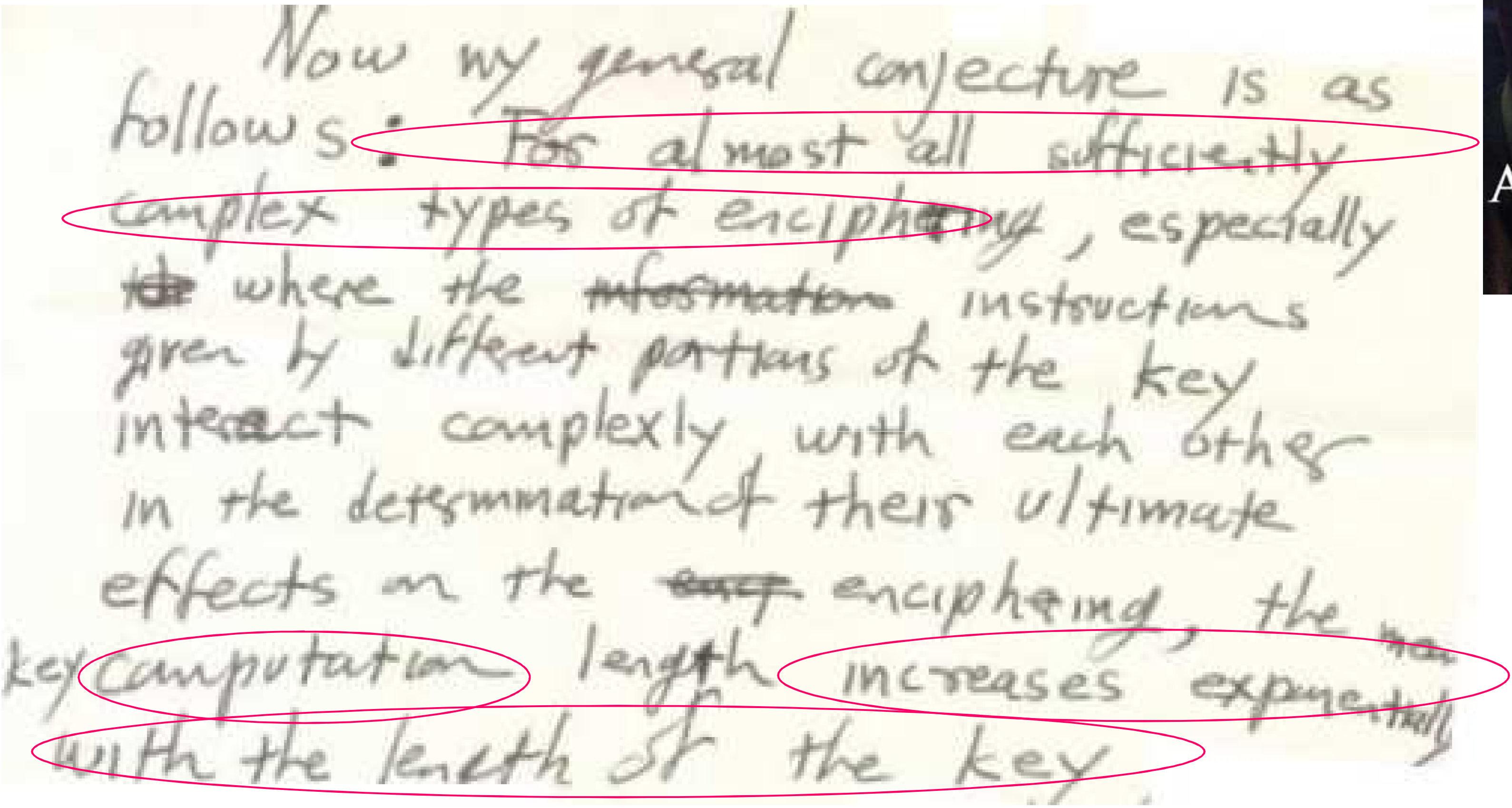
independently, in 1985,

(proposed to do a DH in EC groups

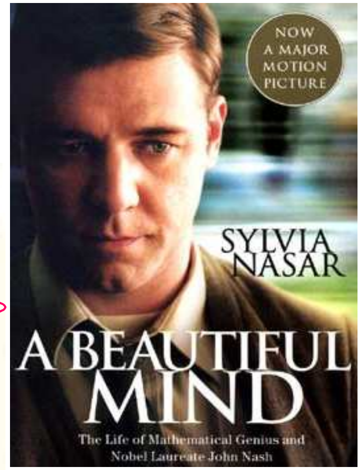
=> whole of modern cryptography)

John Nash - 1955

In 2012 the NSA declassified his hand-written letter:



Now my general conjecture is as follows: For almost all sufficiently complex types of enciphering, especially ~~the~~ where the ~~information~~ instructions given by different portions of the key interact complexly with each other in the determination of their ultimate effects on the ~~enc~~ enciphering, the ~~ma~~ key computation length increases exponentially with the length of the key



[…] "It means that it is quite feasible to design ciphers that are effectively unbreakable.

As ciphers become more sophisticated the game of cipher breaking by skilled teams, etc., should become a thing of the past." […] "The nature of this conjecture is such that I cannot prove it, even for a special type of ciphers. Nor do I expect it to be proven."

ECC vs RSA

RSA: sub-exponential algos,

1024 bit keys provide only 80 bits of security with NFS attack (a form of index calculus).

1986 Victor Miller claimed that:

“It is extremely unlikely that an ‘index calculus’ attack on the elliptic curve method will ever be able to work.“

In V. Miller: “Use of elliptic curves in cryptography“ 1986.

Best known attacks on general elliptic curves: $2^{n/2}$.

160 bit ECC key = 1024 bit RSA key? Not exactly but close:

192-bit ECC over prime field = 80-bit security = RSA 1024 = "breakable today for the NSA"

ECC/RSA Key Size Comparisons

(FIPS 186-2, Lenstra/Verheul, NESSIE)

Security level in bits	Block cipher	$\mathbb{F}_p$ $\|\|p\|\|$	$\mathbb{F}_{2^m}$ m	RSA $\|\|n\|\|$
80	SKIPJACK	192	163	1024
112	Triple-DES	224	233	2048
128	AES Small	256	283	3072
192	AES Medium	384	409	7680
256	AES Large	521	571	15360

256-bit ECC over prime field = 128-bit security

= secure for 50 years in absence of Quantum Computers

ECC/RSA Key Size Comparisons

(FIPS 186-2, Lenstra/Verheul, NESSIE)

Security level in bits	Block cipher	$\mathbb{F}_p$ $\|\|p\|\|$	$\mathbb{F}_{2^m}$ m	RSA $\|\|n\|\|$
80	SKIPJACK	192	163	1024
112	Triple-DES	224	233	2048
128	AES Small	256	283	3072
192	AES Medium	384	409	7680
256	AES Large	521	571	15360

ECC - Certicom Challenges [1997, revised 2009]

ECC2K-95	97	18322	$ 5,000
ECC2-97	97	180448	$ 5,000

Challenge	Field size (in bits)	Estimated number of machine days	Prize (US$)
ECC2K-108	109	1.3×10^6	$10,000
ECC2-109	109	2.1×10^7	$10,000
ECC2K-130	131	2.7×10^9	$20,000
ECC2-131	131	6.6×10^{10}	$20,000

Challenge	Field size (in bits)	Estimated number of machine days	Prize (US$)
ECC2K-163	163	2.48×10^{15}	$30,000
ECC2-163	163	2.48×10^{15}	$30,000
ECC2-191	191	4.07×10^{19}	$40,000
ECC2K-238	239	6.83×10^{26}	$50,000
ECC2-238	239	6.83×10^{26}	$50,000
ECC2K-358	359	7.88×10^{44}	$100,000
ECC2-353	359	7.88×10^{44}	$100,000

ECCp-97	97	71982	$ 5,000

Challenge	Field size (in bits)	Estimated number of machine days	Prize (US$)
ECCp-109	109	9.0×10^6	$10,000
ECCp-131	131	2.3×10^{10}	$20,000

Challenge	Field size (in bits)	Estimated number of machine days	Prize (US$)
ECCp-163	163	2.3×10^{15}	$30,000
ECCp-191	192	4.8×10^{19}	$40,000
ECCp-239	239	1.4×10^{27}	$50,000
ECCp-359	359	3.7×10^{45}	$100,000

TOTAL = 725,000 USD

Bitcoin Elliptic Curve

secp256k1

Bitcoin EC

We define a finite field F_p with 256-bit prime $p=$

FFFFFFFF FFFFFFFF FFFFFFFF FFFFFFFF FFFFFFFF FFFFFFFF FFFFFFFE FFFFFC2F =

115792089237316195423570985008687907853269984665640564039457584007908834671663

Arithmetic modulo p is quite efficient. $p= 2^{256}-2^{32}-977$ $(977=2^9+2^8+2^7+2^6+2^4+1)$.

The curve equation is $y^2 = x^3+7 \bmod p$.

(very special, not like "normal" elliptic curves, small integers only)

Bitcoin EC

We define a finite field F_p with 256-bit prime $p=$
115792089237316195423570985008687907853269984665640564039457584007908834671663

The curve equation is $y^2 = x^3+7 \bmod p$.

The base point G (generator) is: (could be any element)

in compressed form is:

G = 02 79BE667E F9DCBBAC 55A06295 CE870B07 029BFCDB 2DCE28D9 59F2815B 16F81798

In uncompressed form it is: G = 04

79BE667E F9DCBBAC 55A06295 CE870B07 029BFCDB 2DCE28D9 59F2815B 16F81798

483ADA77 26A3C465 5DA4FBFC 0E1108A8 FD17B448 A6855419 9C47D08F FB10D4B8

or simply x,y are

x=55066263022277343669578718895168534326250603453777594175500187360389116729240

y=32670510020758816978083085130507043184471273380659243275938904335757337482424

Bitcoin EC

We define a finite field F_p with 256-bit prime $p=$

115792089237316195423570985008687907853269984665640564039457584007908834671663

The curve equation is $y^2 = x^3+7 \bmod p$.

The base point G (generator) is:

G = 02 79BE667E F9DCBBAC 55A06295 CE870B07 029BFCDB 2DCE28D9 59F2815B 16F81798

The order of G is $n=$

115792089237316195423570985008687907852837564279074904382605163141518161494337

another 256-bit prime such that $n.G=0$.

BTW. All the points on the curve lie in <G>. $\#E(F_p)=n$

What's Wrong With Bitcoin Elliptic Curve?

ECC - Certicom Challenges [1997, revised 2009]

ECC2K-95	97	18322	\$ 5,000
ECC2-97	97	180448	\$ 5,000

ECCp-97	97	71982	\$ 5,000

Challenge	Field size (in bits)	Estimated number of machine days	Prize (US\$)
ECC2K-108	109	1.3×10^{6}	\$10,000
ECC2-109	109	2.1×10^{7}	\$10,000
ECC2K-130	131	2.7×10^{9}	\$20,000
ECC2-131	131	6.6×10^{10}	\$20,000

Challenge	Field size (in bits)	Estimated number of machine days	Prize (US\$)
ECCp-109	109	9.0×10^{6}	\$10,000
ECCp-131	131	2.3×10^{10}	\$20,000

Challenge	Field size (in bits)	Estimated number of machine days	Prize (US\$)
ECC2K-163	163	2.48×10^{15}	\$30,000
ECC2-163	163	2.48×10^{15}	\$30,000
ECC2-191	191	4.07×10^{19}	\$40,000
ECC2K-238	239	6.83×10^{26}	\$50,000
ECC2-238	239	6.83×10^{26}	\$50,000
ECC2K-358	359	7.88×10^{44}	\$100,000
ECC2-353	359	7.88×10^{44}	\$100,000

Challenge	Field size (in bits)	Estimated number of machine days	Prize (US\$)
ECCp-163	163	2.3×10^{15}	\$30,000
ECCp-191	192	4.8×10^{19}	\$40,000
ECCp-239	239	1.4×10^{27}	\$50,000
ECCp-359	359	3.7×10^{45}	\$100,000

secp256k1

NOT INCLUDED

no price if you

break it ☹

Koblitz citation:

"Once I heard a speaker from NSA complain about university researchers who are cavalier about proposing untested cryptosystems. He pointed out that in the real world if your cryptography fails, you lose a million dollars or your secret agent gets killed.

In academia, if you write about a cryptosystem and then a few months later find a way to break it, you've got two new papers to add to your résumé!”

Neal Koblitz,

Notices of the American Mathematical Society,

September 2007.

k1 Promoters: Late Denial?

SECG = Standards for Efficient Cryptography group,

Industry consortium offspring of Canadian Certicom: the people who proposed/promoted secp256k1 in the first place.

Their document claims that both offer the same level of security=RSA-3072

=> not even strictly true in current literature, k1 has a slightly faster attack, cf. Cryptrec report by Smart and Galbraith

Timely denial:

Dan Brown, the SECG chair has written on 18 September 2013:

-"I did not know that BitCoin is using secp256k1.
I am surprised to see anybody use secp256k1 instead of secp256r1",

https://bitcointalk.org/index.php?topic=289795.80

Comparison:

Used/recommended by:	secp256k1	secp256r1
Bitcoin, anonymous founder, no one to blame…	Y	
SEC Certicom Research	now surprised	Y
TLS, OpenSSL	Y, ever used???	Y **98.3%** of EC
U.S. ANSI X9.63 for Financial Services	Y	Y
NSA suite B, NATO military crypto		Y
U.S. NIST		Y
IPSec		Y
OpenPGP		Y
Kerberos extension		Y
Microsoft implemented it in Vista and Longhorn		Y
EMV bank cards XDA [2013]		Y
German BSI federal gov. infosec agency, y=2015		Y
French national ANSSI agency beyond 2020		Y

TLS Adoption

7 % of TLS and 10% of SSH connections use ECCs [December 2013].

98.3 % of these use secp256r1.

(no data reported on k1 counterpart, negligible or zero).

Source: Bos-Halderman-Heninger-Moore-Naehrig-Wustrow, Paper= eprint/2013/734, slides=ask me.

Remark: Many SSL libraries such GnuTLS.org support only a subset of what OpenSSL does, and they don't support k1.

Is k1 Weak?

Weakness?

Bitcoin curve is characterized by the so called ``small class number'' which some researchers suspect to be less secure than general curves, see Sections 5.1 and 5.3 in http://www.ipa.go.jp/security/enc/CRYPTREC/fy15/doc/1029_report.pdf

All Koblitz curves have small class number.
In fact, the Koblitz curves in the SEC standard all have class number 1.

A STRONGER (more paranoid) requirement of having a large so called CM Field Discriminant D to be $|D|>2^{100}$, cf. http://safecurves.cr.yp.to/disc.html

The bitcoin elliptic curve has the LOWEST |D| of all known standardized elliptic curves, and therefore it is potentially the least secure.

Such curves allow “slight speedups” for discrete log attacks however "the literature does not indicate any mechanism that could allow further speedups".

Wanna Bet?

BetMoose BETA

Bitcoin Cryptography Broken in 2015

Category: Bitcoin

By NCourtois ★★★★★

Description

The digital signature scheme of bitcoin with SHA256+secp256k1 ECDSA will be broken before 1 September 2015 by cryptography researchers.
The attack should allow to forge digital signatures for at least a proportion of 1/1 million bitcoin users and steal money from them.
It should be done faster than 2^100 point additions total including the time to examine the data.

Decision Logic

bitcoin, cryptography, SHA256, ECDSA, ECDL, secp256k1

betmoose.com - Totally Anonymous Bets In BTC!

FEATURED

Bitcoin Cryptography Broken in 2015

Category: Bitcoin By NCourtois ★★★★★

Description

The digital signature scheme of bitcoin with SHA256+secp256k1 ECDSA will be broken before 1 September 2015 by cryptography researchers.
The attack should allow to forge digital signatures for at least a proportion of 1/1 million bitcoin users and steal money from them.
It should be done faster than 2^100 point additions total including the time to examine the data.

YES

Volume:	฿ 0.140
# of Bets:	3

฿

PAYOUT	ROI
฿ 0.00	0%

*assumes current weight and volumes

Place Anonymously

NO

Volume:	฿ 0.189
# of Bets:	6

฿ 0.1

PAYOUT	ROI
฿ 0.14327	43.27%

*assumes current weight and volumes

Place Anonymously

Decision Logic

SHA256, ECDSA, ECDL, secp256k1

Computations on Elliptic Curves

Follow The Picture

Formulas for
point
addition

char p>2

$$\begin{aligned} x_3 &= \lambda^2 - x_1 - x_2, \\ y_3 &= (x_1 - x_3)\lambda - y_1 \end{aligned}$$

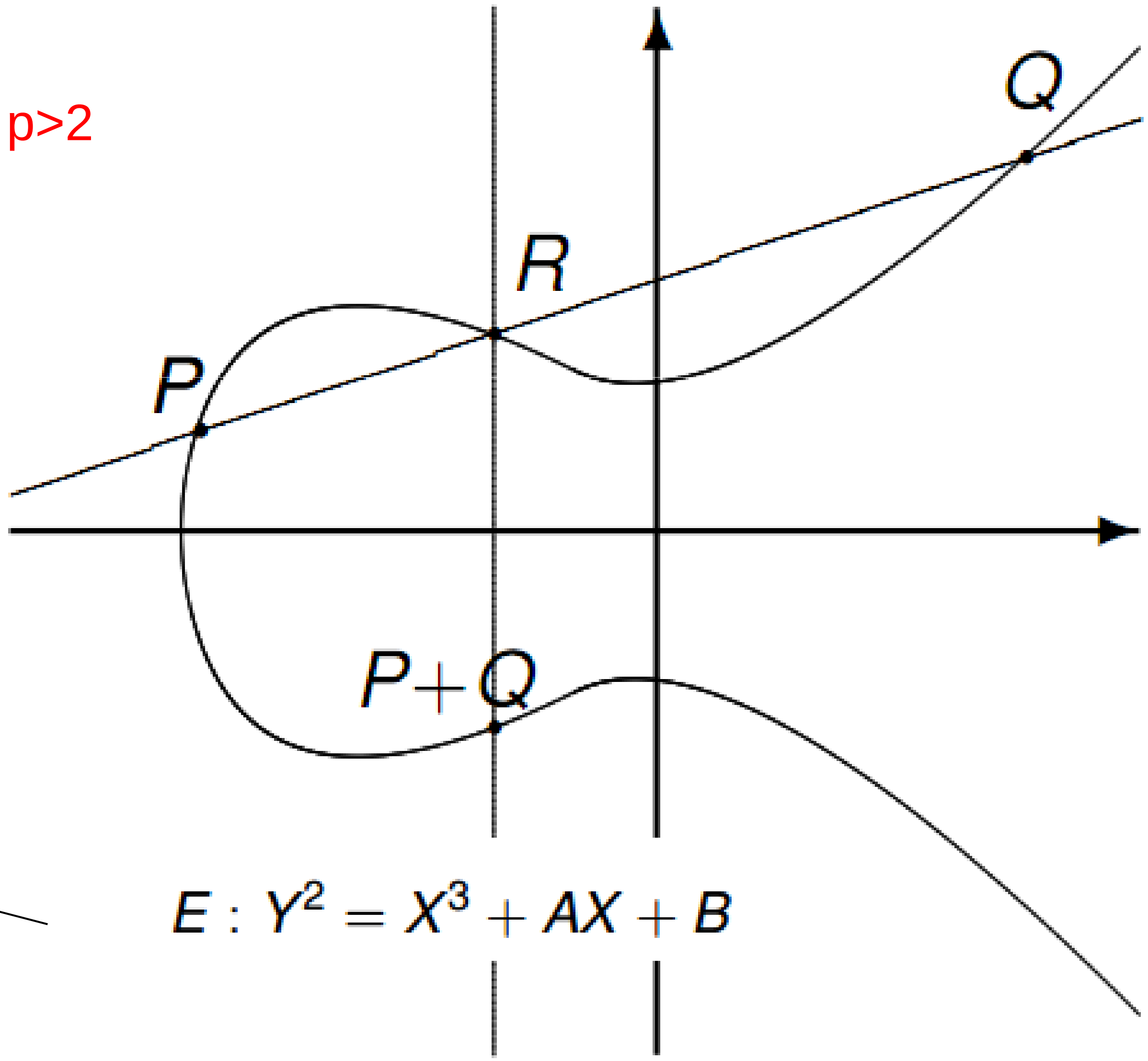
when $x_1 \neq x_2$ we set

$$\lambda = \frac{y_2 - y_1}{x_2 - x_1},$$

~~when $x_1 = x_2$ and $y_1 \neq 0$ we set~~

see next slide

$$\cancel{\lambda = \frac{3x_1^2 + A}{2y_1}}$$

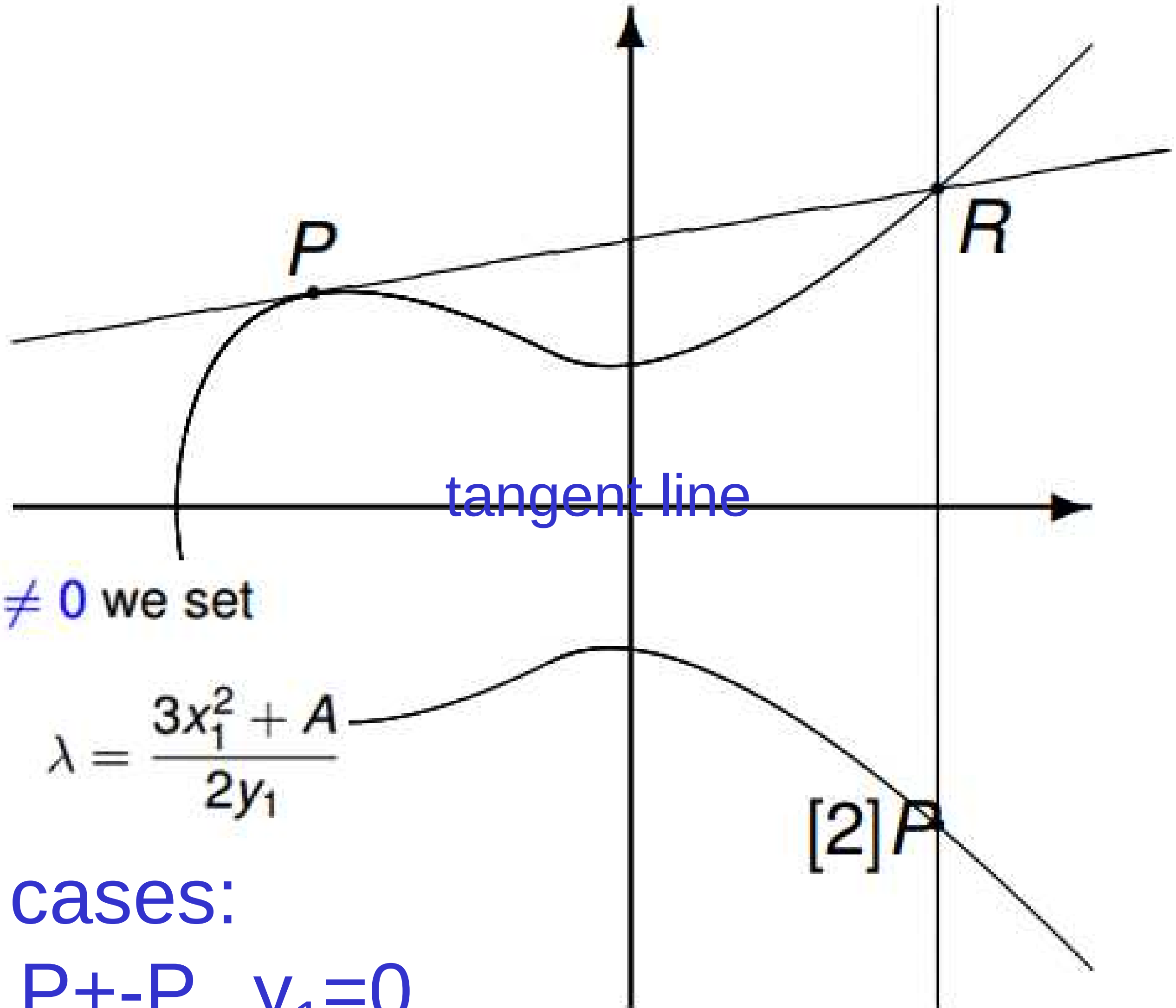


The Case of Doubling

Different formula for point doubling

when $x_1 = x_2$ and $y_1 \neq 0$ we set

$$\lambda = \frac{3x_1^2 + A}{2y_1}$$

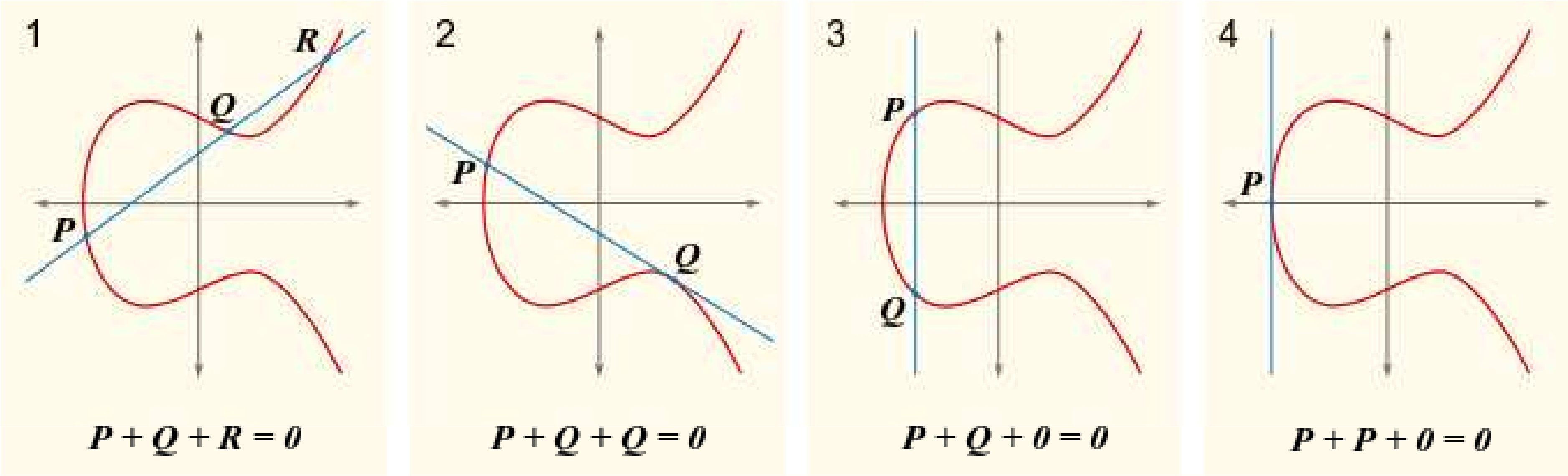


NOT a complete law!

many special cases:

0_E+0_E, $P+0_E$, $P+-P$, $y_1=0$

Special Cases => Not a “Complete Law”



tangent line

tangent line

RSA: exponentiation, modulo n

- e has $k=\log_2 n$ bits.

SQUARE AND MULTIPLY method:

Let $e = \Sigma_{i=1} 2^k e_i$ be the binary expansion of e.

- Compute $x, x^2, x^4, x^8, \ldots, x^{2^\wedge(k-1)}$,
- Multiply all those for which $e_i=1$.

Cost: about k S $+k/2$ M on average.

ECC: exponentiation, 2 moduli p,n

- e has $k=\log_2 n$ bits.

DOUBLE AND ADD method:

Let $e = \Sigma_{i=1} 2^k e_i$ be the binary expansion of e.

- Compute G, 2.G, 4.G,8.G,…, (2^(k-1)).G
- Add all those for which $e_i=1$.

Cost: about k*cost(double) +k/2*cost(add) on average.

=>Can further save on additions, with a bit more of memory (store more multiples, user larger ‘digits’ than 1 bit)

Costs for ECC in “odd char” p

Point Addition:

F_p

- 6 Field Additions (Trivial)
- 3 General Field Multiplications 3M+1I
- 1 Field Inversion

$\boxed{1I=10\text{-}100M}$

Point Doubling:

- 5 Field Additions (Trivial) 4M+1I
- 2 Scalar/Field Multiplications (Trivial)
- 4 General Field Multiplications
- 1 Field Inversion

Projective Coordinates

x,y => x,y,z

avoids field inversions

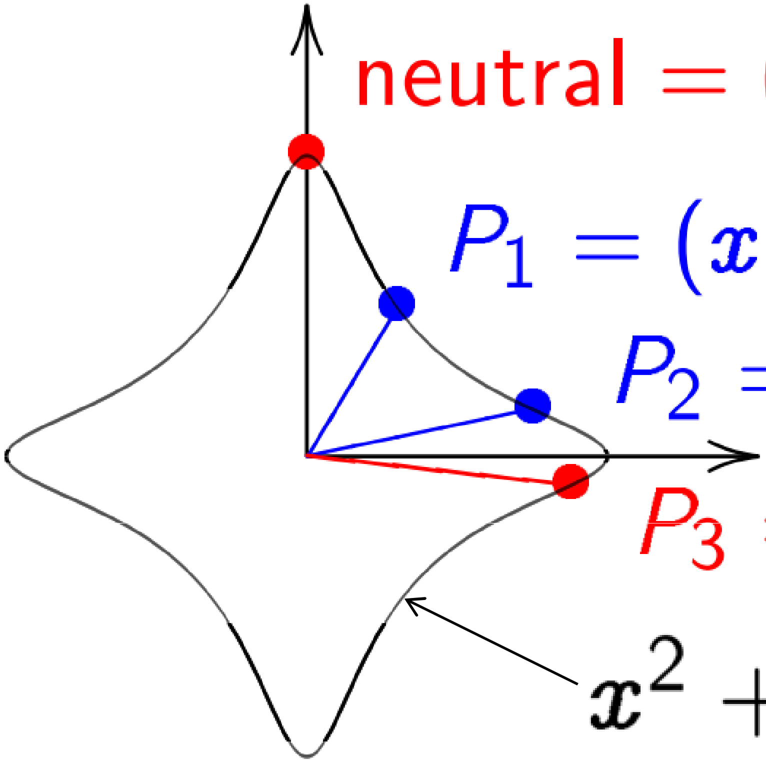
Operation	Affine	Projective
Addition	3M + 1I	16M
Doubling	4M + 1I	10M

Edwards Coordinates [2007]

revisited by Bernstein and Lange

	Weierstrass		
Operation	Affine	Projective	Edwards
Addition	3M + 1I	16M	10M
Doubling	4M + 1I	10M	3M

Edwards Coordinates



no special points!
complete law!

$$x^2 + y^2 = 1 - 30x^2y^2$$

Sum of (x_1, y_1) and (x_2, y_2) is

$$((x_1y_2+y_1x_2)/(1-30x_1x_2y_1y_2),$$
$$(y_1y_2-x_1x_2)/(1+30x_1x_2y_1y_2))$$

Edwards mod p [Bernstein-Lange version]

Choose an odd prime p.

Choose a *non-square* $d \in \mathbf{F}_p$.

$$\{(x, y) \in \mathbf{F}_p \times \mathbf{F}_p : x^2 + y^2 = 1 + dx^2y^2\}$$

is a "complete Edwards curve"

no special points!
complete law!

=> secure against side-channel attacks

$$(x_1, y_1) + (x_2, y_2) = (x_3, y_3)$$

where

$$x_3 = \frac{x_1y_2 + y_1x_2}{1 + dx_1x_2y_1y_2},$$

$$y_3 = \frac{y_1y_2 - x_1x_2}{1 - dx_1x_2y_1y_2}.$$

never 0

PROBLEM:
not every curve can be converted to Edwards curve, must have a point of order 4…
not for bitcoin curve k1

Edwards Coordinates [2007]

actually invented by Bernstein and Lange

	Weiestrass		
Operation	Affine	Projective	Edwards
Addition	3M + 1I	16M	10M
Doubling	4M + 1I	10M	3M

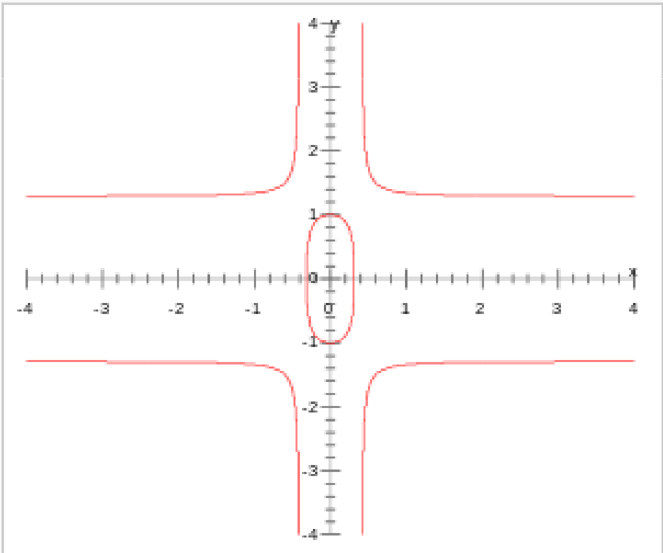
Twisted Edwards – More Curves

Definition 6.1 (Twisted Edwards curve). Let $\mathbb{F}_q$ be a finite field with $char(\mathbb{F}_q) \notin \{2,3\}$ and let $a, d \in \mathbb{F}_q^*$ with $a \neq d$. Then, the *twisted Edwards curve with coefficients* a, d is defined as:

$$\boxed{E^{E_T,a,d} : aX^2 + Y^2 = 1 + dX^2Y^2} \tag{6.4}$$

with j-invariant $j(E^{E_T,a,d}) = 16(a^2 + 14ad + d^2)^3(ad(a-d)^4)^{-1}$.

For $a = 1$ this coincides with the notion of ordinary non-binary Edwards curves.



A Twisted Edwards curve of equation
$10x^2 + y^2 = 1 + 6x^2y^2$

$$x_3 = \frac{x_1y_2 + y_1x_2}{1 + dx_1x_2y_1y_2}$$

$$y_3 = \frac{y_1y_2 - ax_1x_2}{1 - dx_1x_2y_1y_2}$$

no special points!
complete law!

=> secure against side-channel attacks

More Curves!

More curves can be converted to twisted Edwards
PROBLEM: still not for bitcoin curve k1

a Montgomery curve over $\mathbb{F}_q$ with coefficients $e, f \in \mathbb{F}_q$ with $e \neq \pm 2$, $f \neq 0$ is defined by the equation:

$$E^{M,e,f} : fY^2 = X^3 + eX^2 + X.$$

Thm. Cf. Christian Hanser thesis: New Trends in Elliptic Curve Cryptography,

- *Each twisted Edwards curve over $\mathbb{F}_q$ with coefficients $a, d \in \mathbb{F}_q, a \neq d$ is birationally equivalent over $\mathbb{F}_q$ to some Montgomery curve. The birational equivalence is defined by the map:*

$$\begin{array}{rccc} \phi: & E^{E_T,a,d} & \to & E^{M,e,f} \\ & (x_0, y_0) & \mapsto & (\alpha(y_0), x_0 \cdot \alpha(y_0)) \end{array}$$

  with $\alpha(y) = (1+y)(1-y)^{-1}$ and coefficients a, d mapped to $(e, f) = (2(a+d)(a-d)^{-1}, 4(a-d)^{-1})$.

- *On the flipside, each Montgomery curve over $\mathbb{F}_q$ with coefficients $e \in \mathbb{F}_q \setminus \{\pm 2\}$ and $f \in \mathbb{F}_q^*$ is birationally equivalent over $\mathbb{F}_q$ to some twisted Edwards curve. The birational equivalence is defined by the inverse map ϕ^{-1}:*

$$\begin{array}{rccc} \phi^{-1}: & E^{M,e,f} & \to & E^{E_T,a,d} \\ & (x'_0, y'_0) & \mapsto & (x'_0 y'^{-1}_0, (x'_0 - 1)(x'_0 + 1)^{-1}) \end{array}$$

  and the coefficients e, f are mapped to $(a, d) = ((e+2)f^{-1}, (e-2)f^{-1})$.

Twisted Edwards – More Curves

$$aX^2 + Y^2 = 1 + dX^2Y^2$$

Good news:
More curves can be converted to twisted Edwards
PROBLEM: still not for bitcoin curve k1

Vincent Verneuil paper: problem solved, but curves over GF(p^3)!

MOST elliptic curves need to be reformed,
NOT GOOD anymore. New curves to be used:
safecurves.cr.yp.to

Edwards Curve Ed25519

"superfast, super secure ECDH"

$\#E(G)=8(2^{252}+27742317777372353535851937790883648493)$

D. J. Bernstein: Curve25519: new Diffie-Hellman speed records. In PKC 2006.

Ed25519 is used in Tor project, secure DNS, SSH and much more…

ECDSA

ECDSA

The Elliptic Curve Digital Signature Algorithm

- the elliptic curve analogue of DSA/DSS=outdated, 80-bit security.
- invented in 1992 by Scott Vanstone [died March 2014] in response to a NIST request for public comments on their first proposal for DSS

Adoption:

- accepted in 1998 in ISO 14888-3
- accepted in 1999 by ANSI X9.62 Financial Institutions standard.
- accepted in 2000 to become IEEE 1363-2000 standard
- accepted in 2000 as a NIST FIPS 186-2 standard.
- since 2002 also in ISO 15946-2
- since 2005 crucial part of NSA suite B (not RSA)

ECDSA Signature Generation

To sign a message m, A does the following:

1. Select a random integer k, $1 \leq k \leq n-1$.
2. Compute $R = kP$ and $r = x(R) \bmod n$.
   If $r = 0$ then go to step 1.
3. Compute $k^{-1} \bmod n$.
4. Compute $e = H(m)$, where H is a hash function.
5. Compute $s = k^{-1}(e + dr) \bmod n$. d=private key
   If $s = 0$ then go to step 1.
6. A's signature for the message m is (r, s).

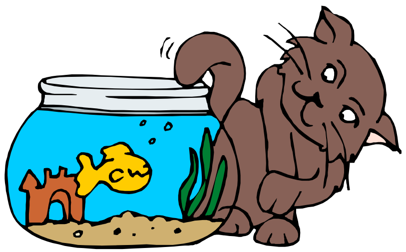
ECDSA Signature Verif.

To verify A's signature (r, s) on m, B should do the following:

1. Verify that r and s are integers in the interval $[1, n-1]$.
2. Compute $e = H(m)$.
3. Compute $u_1 = es^{-1} \bmod n$ and $u_2 = rs^{-1} \bmod n$.
4. Compute $R = u_1P + u_2Q$ and $v = x(R) \bmod n$.
5. Accept the signature if and only if $v = r$.

r,-s also valid

Dangers of ECDSA



Recall Sign. Generation

To sign a message m, A does the following:

1. Select a random integer k, $1 \leq k \leq n-1$.
2. Compute $R = kP$ and $r = x(R) \bmod n$.
   If $r = 0$ then go to step 1.
3. Compute $k^{-1} \bmod n$.
4. Compute $e = H(m)$, where H is a hash function.
5. Compute $s = k^{-1}(e + dr) \bmod n$. d=private key
   If $s = 0$ then go to step 1.
6. A's signature for the message m is (r, s).

Attack Vectors (0)

random k: must be kept secret!

RNG

↓

random a

↓

R=a.P

r ↓

s= (H(m)+dr) / a mod n → (r,s)

side channel attacks/SPA/DPA on the private key d

Bad/Good RNG Attacks

1.

Bad RNG => the attacker CAN guess/brute force k. => recover private key

2.

Bad RNG but attacker cannot guess it [e.g. obscurity]
=> there are still attacks (see next slides)!

3.

Good RNG but with side channels…

Key Problem

random k: should NEVER be revealed.

RNG

↓

random a

↓

$R=a.P$

↓ r

$s = (H(m)+dr) / a \bmod n$ → (r,s)

if a is revealed, the private key can be computed!

$d=(sa-H(m))/r \bmod n$

Attack Vectors (1)

random k: must be kept secret!

RNG

↓

random a

↓

R=a.P

r ↓

s= (H(m)+dr) / a mod n → (r,s)

used /dev/random
or /dev/urandom
or MsWin32
CryptGenRandom
<u>but</u> OS/CPU has
logged the seed!

Attack Vectors (2)

random k: must be kept secret!

RNG

random a

R=a.P

r

s= (H(m)+dr) / a mod n

(r,s)

side channel attack/SPA: a USB drive powered by the PC has recorded the power consumption when generating/using a

Attack Vectors (3)

random k: must be kept secret!

RNG

random a

R=a.P

r

s= (H(m)+dr) / a mod n

(r,s)

=>has happened 100s of times!

same a used twice => detected in public blockchain =>

$(s_1 a - H(m_1))/d_1 = r = (s_2 a - H(m_2))/d_2 \bmod n$

=>

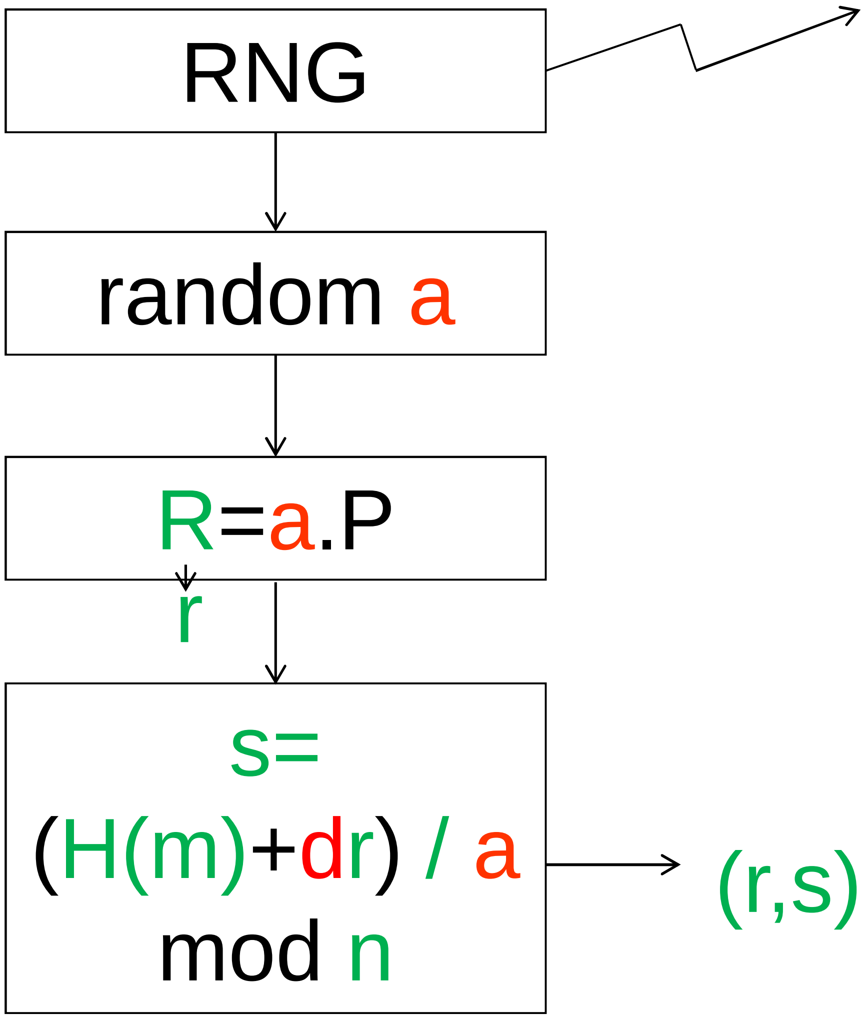
$r(d_1 - d_2) + a(s_1 - s_2) = H(m_2) - H(m_1) \bmod n$

each person can steal the other person's bitcoins!

=>any of them CAN recompute k used

Attack Vectors (3)

random k: must be kept secret!



=>has happened a few times…

same a used twice by the same user ($d_1=d_2$). In this case we have:

$(s_1a-H(m_1)) = rd = (s_2a-H(m_2))$ mod n

=> $a=(H(m_1)-H(m_2))/(s_1-s_2)$ mod n

AND now

$d=(sa-H(m))/r$ mod n

anybody can steal the bitcoins!

Deterministic ECDSA!

Barwood-Wigley-Naccache-M’Raihi-Levy-dit-Vehel-Naccache-Pointcheval-Vaudenay-Katz-Wang etc…

Deterministic ECDSA

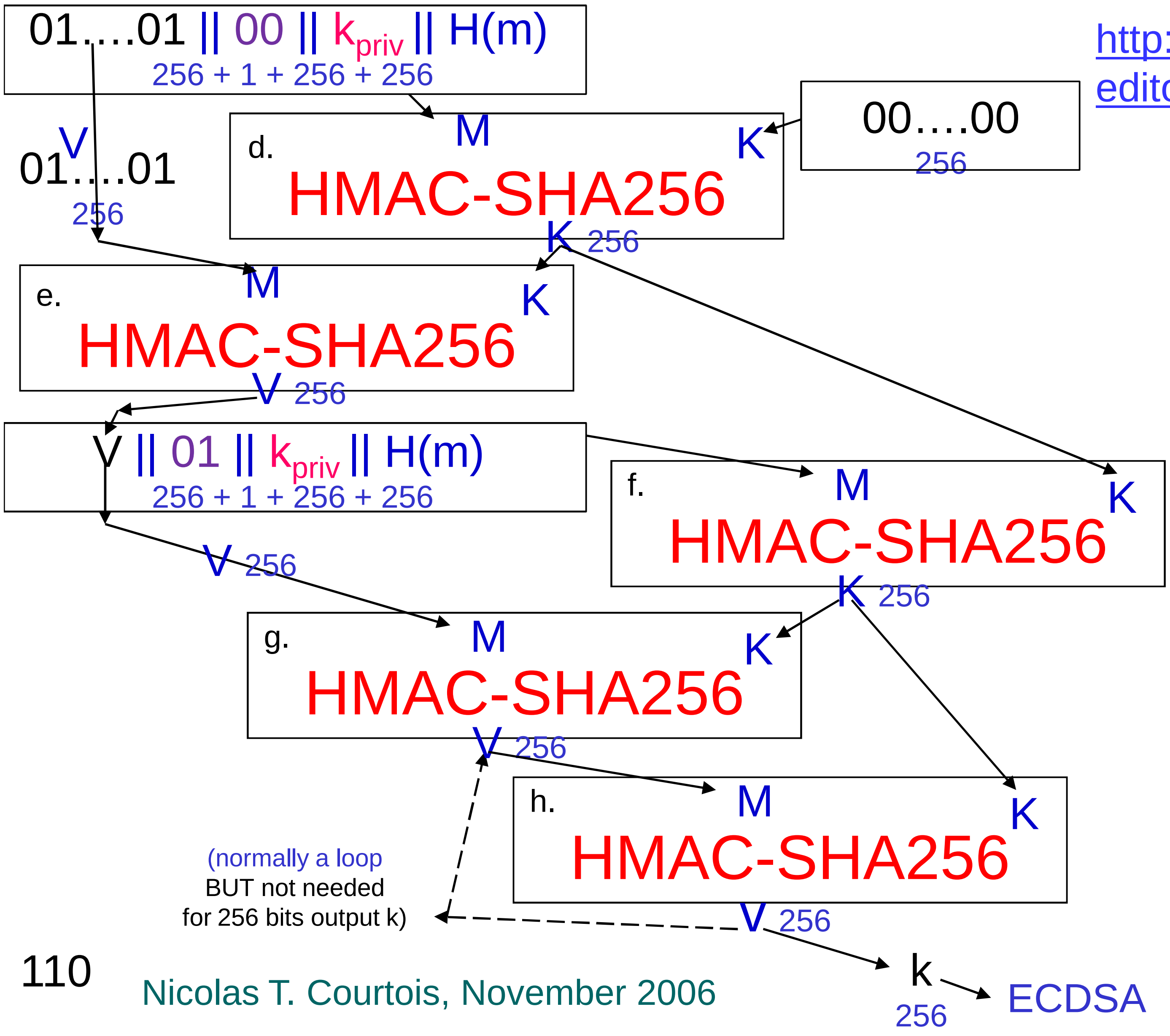
- Avoids attacks with bad RNG.
- Very strong protection against NSA backdoors such as hacking the RNG on the fly etc.
  - Deterministic => do it twice with different implementations, compare result.

Solution:

- RFC6979
- In pycoin library tx.py program:
  - uses a deterministic algorithm to create the ECSDA signatures, example to imitate.

RFC6979 [Pornin] = 5+ applications of HMAC

http://www.rfc-editor.org/rfc/rfc6979.txt



**HMAC-SHA256

Hashes twice with a key.

Definition (from RFC 2104)

$HMAC(K,m) = H\left((K \oplus opad)|I\right.$

where

H is a cryptographic hash function,

K is a secret key padded to the right with

the hash of the original key if it's longer th

m is the message to be authenticated,

| denotes concatenation,

$\oplus$ denotes exclusive or (XOR),

opad is the outer padding (0x5c5c5c…5c

and *ipad* is the inner padding (0x363636…