Arithmétique et Cryptographie Asymétrique

Laurent Imbert

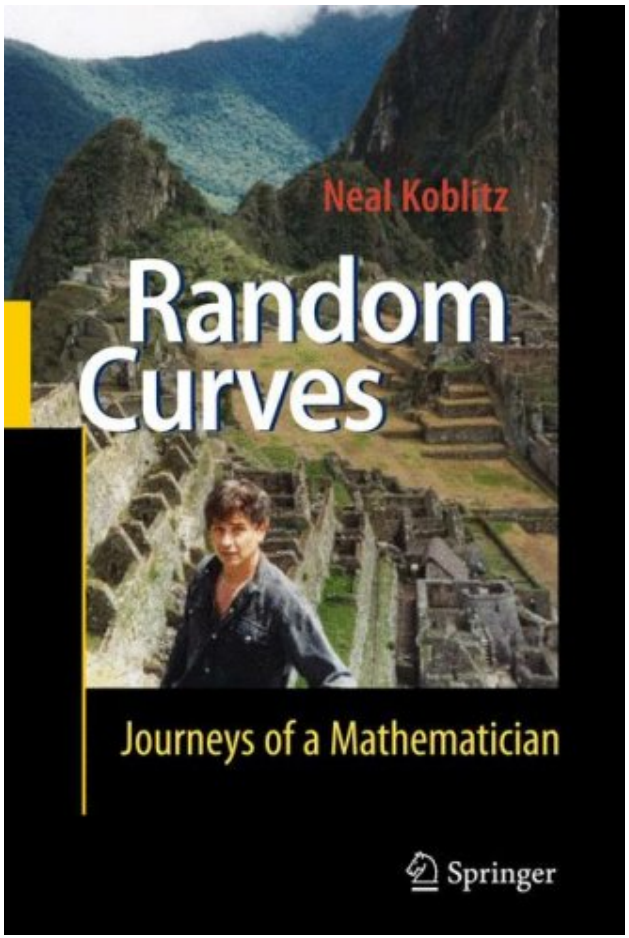
CNRS, LIRMM, Université Montpellier 2

Journée d'inauguration groupe Sécurité

23 mars 2010

This talk is about public-key cryptography

Why did mathematicians end up using complex objects like elliptic curves over finite fields? There must be easier ways to encrypt a message... No?



Public-Key Cryptography (PKC)

Anyone can encrypt a message with a *public* key but only the receiver can decrypt it with its *private* key

Comunications using PKC

1. Bob sends Alice his public key / Alice gets Bob's public key from a database

   ‘

2. Alice encrypts her message using Bob's public key and sends it to Bob

3. Bob then decrypts Alice's message using his private key

PKC solves the key management problem

In the real world, PKC is used to encrypt (symmetric) keys and for digital signature

One-way functions

Easy to compute, but significantly harder to reverse

Easy: can be expressed with simple formula, e.g. polynomials

Hard: It would take millions of years to compute x from $f(x)$, even if all the computers in the world were assigned to the problem. (B. Schneier)

As of today, reversing f should be at least 2^{80} times harder than computing f

Trapdoor one-way functions

One-way functions (if they exist) look like the perfect encryption machinery... if one does not care about decryption!

For PKC, one uses *Trapdoor one-way functions*: one-way functions that become easy to reverse if one knows a secret information

Famous examples:

- Integer factorization: RSA
- Discrete logarithms in finite groups: ElGamal, DH, DSA, ECDSA, etc.

Finite groups

A set of elements G together with a binary operation called the group operation or group law

1. For all a, b in G, then $a \cdot b$ is also in G
2. There exists an identity element e such that $a \cdot e = e \cdot a = a$
3. For all $a \in G$, there exists $a^{-1} \in G$ such that $a \cdot a^{-1} = e$
4. $a \cdot (b \cdot c) = (a \cdot b) \cdot c$

If G has a finite number of elements, then the group is finite.

Examples of Finite Groups

The additive group of integers modulo p

$\mathbb{Z}_p$ or $\mathbb{Z}/p\mathbb{Z} \cong \{0, 1, 2, \ldots, p-1\}$

DLP: given $a \neq 0,\ b \equiv ka \pmod{p}$, find k

Trivial: $a \in \mathbb{F}_p \Rightarrow k \equiv ba^{-1} \pmod{p}$

The multiplicative group of a finite field

If p is prime, then $\mathbb{F}_p^\times \cong \{1, 2, \ldots, p-1\}$

DLP: given $a,\ b \equiv a^k \pmod{p}$, find k

Much more difficult! Good candidate for crypto.

ElGamal Cryptosystem in $\mathbb{F}_p^\times$

Let p be a large prime and a a generator for $\mathbb{F}_p^\times$

Key generation choose a random k and compute $b = a^k \bmod p$

Publish (p, a, b) and keep k secret

Encryption for a secret random integer $0 \leq r < p - 1$

$$\mathcal{E}(x, r) = (y_1, y_2)$$

where

$$y_1 = a^r \bmod p \qquad y_2 = xb^r \bmod p$$

Decryption

$$\mathcal{D}(y_1, y_2) = y_2(y_1^k)^{-1} \bmod p$$

Index Calculus Methods for DLP

Let p be a prime and g a *generator* for the cyclic group $\mathbb{F}_p^\times$. Every $h \not\equiv 0 \pmod{p}$ can be written in the form $h \equiv g^k$ for some integer $0 \leq k < p-1$. Let $k = \ell(h)$ denote the *discrete logarithm* of h

$$g^{\ell(h)} \equiv h \pmod{p}$$

Suppose we have h_1 and h_2. Then

$$g^{\ell(h_1 h_2)} \equiv h_1 h_2 \equiv g^{\ell(h_1)+\ell(h_2)} \pmod{p}$$

which implies

$$\ell(h_1 h_2) \equiv \ell(h_1) + \ell(h_2) \pmod{p-1}$$

The *index calculus* is a method for computing values of ℓ. The idea is to compute $\ell(\pi_i)$ for several small primes π_i, then use this information to compute $\ell(h)$ for arbitrary h.

Example (Index Calculus)

Let $p = 1217$ and $g = 3$. We want to solve $3^k \equiv 37 \pmod{1217}$.
We choose a set of small primes, called a *factor base*

$$B = \{2, 3, 5, 7, 11, 13\}$$

Find relations $3^x \equiv \pm$ product of some primes in $B \pmod{1217}$

$$3^1 \equiv 3 \qquad 3^{25} \equiv 5^3 \qquad 3^{54} \equiv -5 \cdot 11$$
$$3^{24} \equiv -2^2 \cdot 7 \cdot 13 \qquad 3^{30} \equiv -2 \cdot 5^2 \qquad 3^{87} \equiv 13$$

Linear algebra compute $\ell(\pi_i)$, for all $\pi_i \in B$

$$\begin{pmatrix} 1 & 0 & 0 & 0 & 0 & 0 & 0 \\ 0 & 0 & 1 & 0 & 0 & 0 & 0 \\ 1 & 2 & 0 & 0 & 1 & 0 & 1 \\ 0 & 0 & 0 & 3 & 0 & 0 & 0 \\ 1 & 1 & 0 & 2 & 0 & 0 & 0 \\ 1 & 0 & 0 & 1 & 0 & 1 & 0 \\ 0 & 0 & 0 & 0 & 0 & 0 & 1 \end{pmatrix} \begin{pmatrix} \ell(-1) \\ \ell(2) \\ \ell(3) \\ \ell(5) \\ \ell(7) \\ \ell(11) \\ \ell(13) \end{pmatrix} \equiv \begin{pmatrix} 608 \\ 1 \\ 24 \\ 25 \\ 30 \\ 54 \\ 87 \end{pmatrix} \pmod{1216}$$

Example (Index Calculus) cont'd

We now know the discrete logs of all elements of the factor base. Recall that we want to solve $3^k \equiv 37 \pmod{1217}$

Finish the work

Compute $3^j \cdot 37 \pmod{p}$ for random values of j until we get an integer that can be factored into a product of primes in B

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

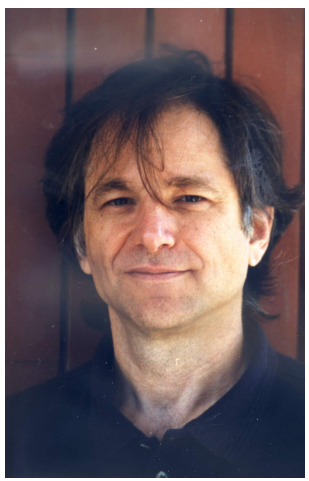
Therefore

$$\ell(37) \equiv 3\ell(2) + \ell(7) + \ell(11) - 16 \equiv 588 \pmod{1216}$$

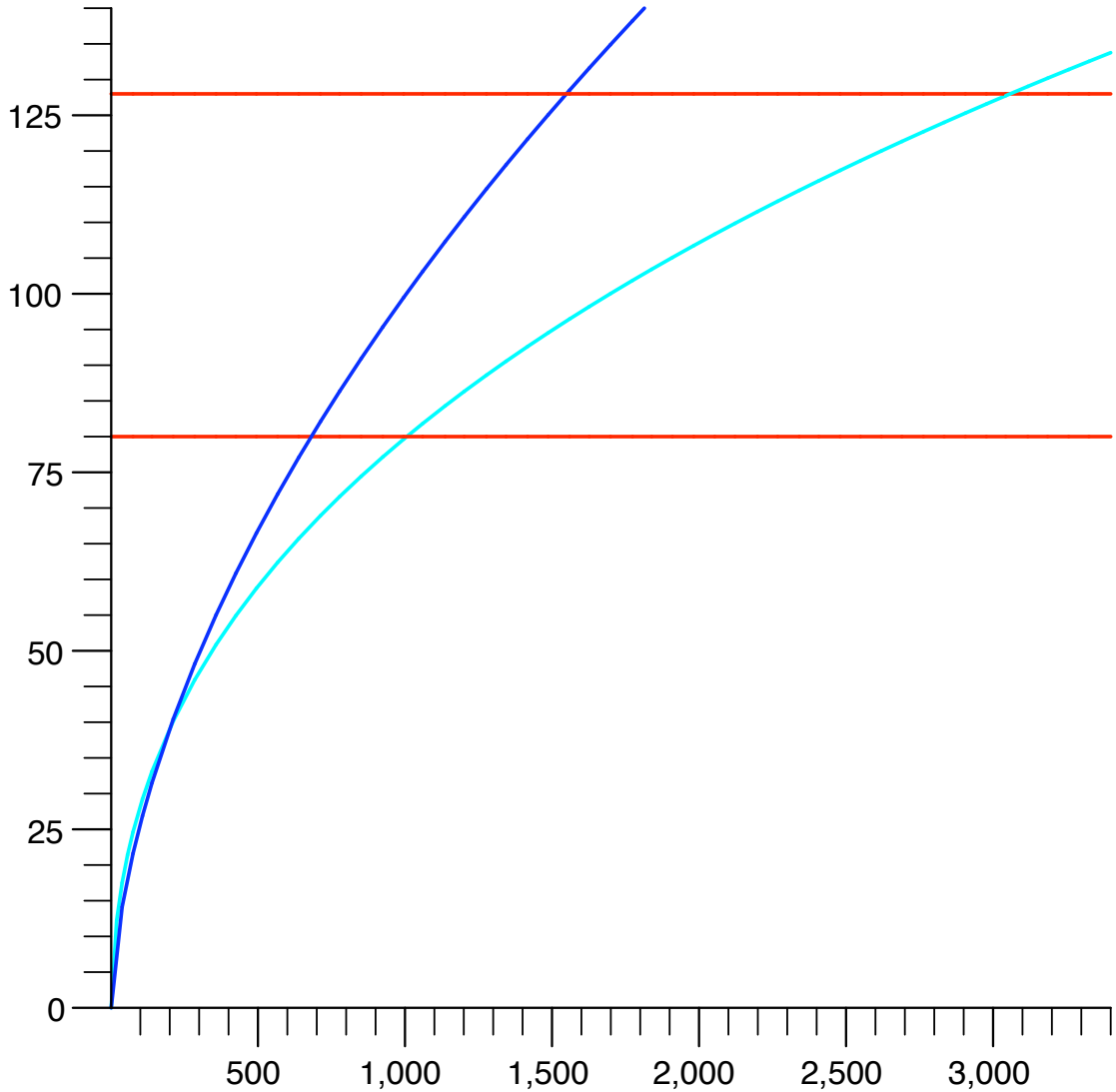
$$3^{588} \equiv 37 \pmod{1217}$$

The Index Calculus is subexponential

$$L_n(\alpha, c) = O\left(e^{(c+o(1))(\ln n)^{\alpha}(\ln \ln n)^{1-\alpha}}\right)$$




So, how large should p be?



Computing with big integers

Integer multiplication
Karatsuba, Toom-Cook, Schönhage-Strassen (FFT)

Modular multiplication $a, b, p \longrightarrow ab \bmod p$
Montgomery, Barrett

Fast exponentiation $a, r, p \longrightarrow a^r \bmod p$
square-multiply, high-radix/window methods, double-base

Inversion $a, p \longrightarrow a^{-1} \bmod p$
extended gcd, Fermat

GMP, NTL, Pari/GP, Sage, Magma, LinBox, etc.

So why do we need something else?

So far, we need very large keys because discrete logarithms in $\mathbb{F}_p^\times$ can be computed in subexponential time.

Is it possible to use smaller keys without compromising security?

Smaller keys $\Longrightarrow$ faster arithmetic, less memory, less bandwidth

Are there any finite groups for which index calculus methods do not work?

Here Comes Elliptic Curves

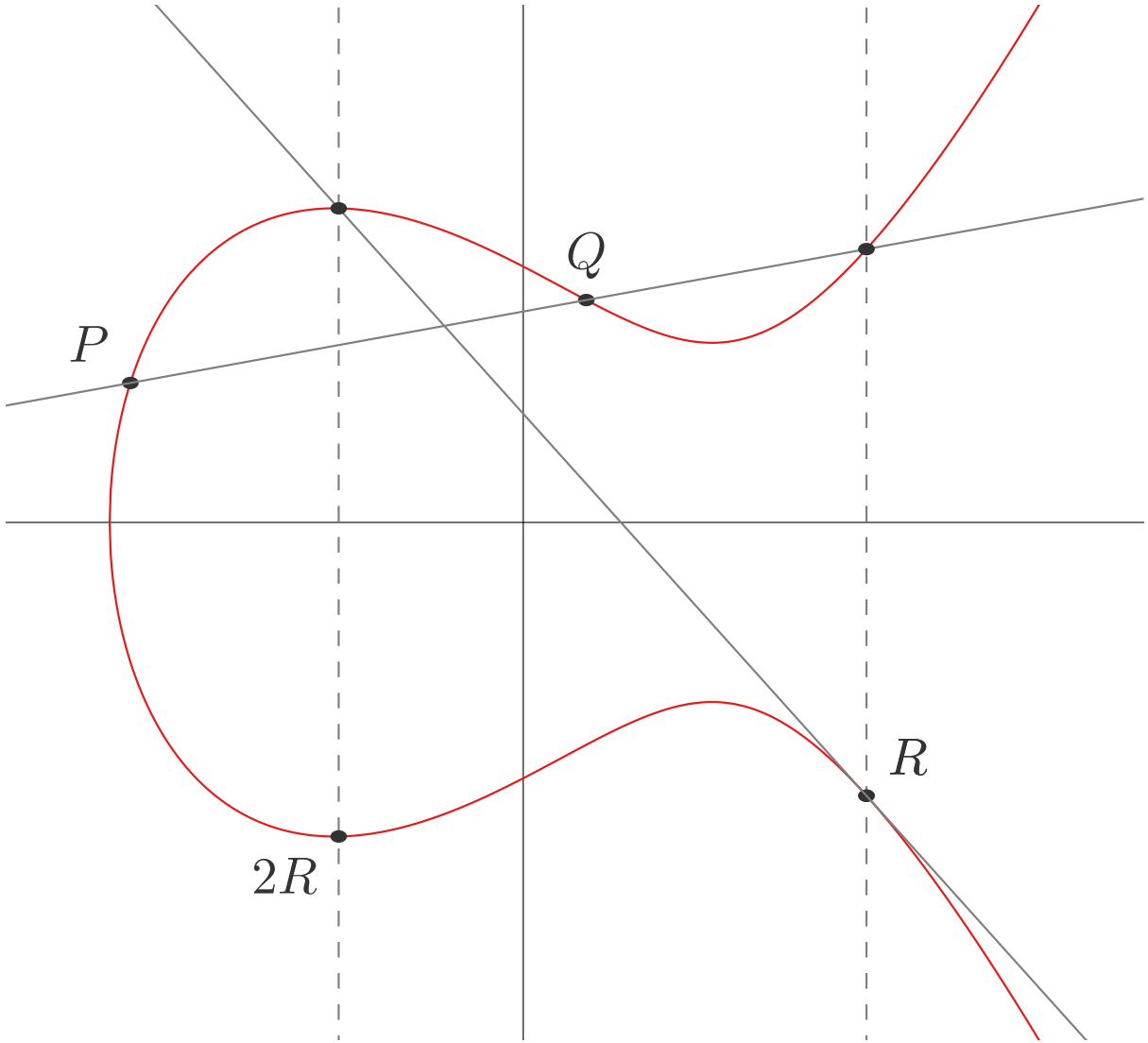
An elliptic curve is

- a geometrical object: a nonsingular curve given by an equation

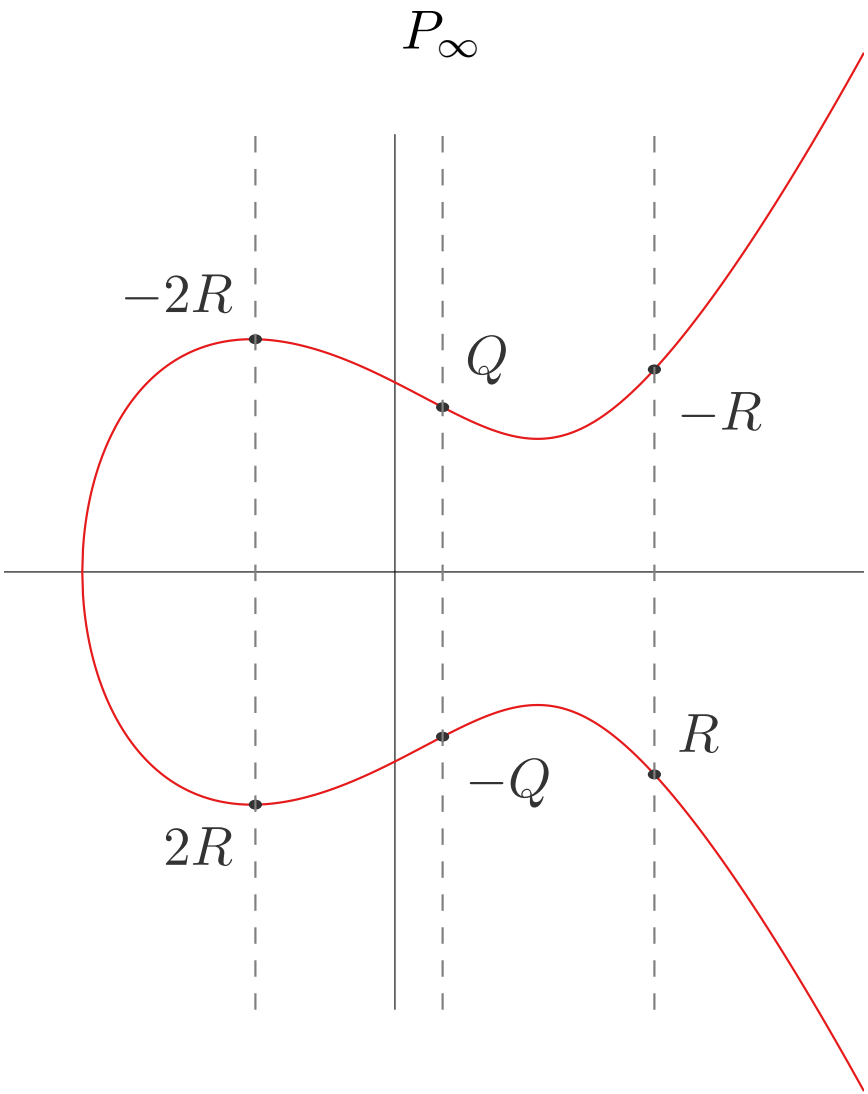
$$y^2 = f(x), \quad \text{with } \deg f = 3, 4$$

- an algebraic object: the set has a natural geometric law, which also respects field of definition (i.e. works over finite fields)

Adding points on an elliptic curve



Point negation and the point at infinity



Algebraic description of the addition operation

Let $P_1 = (x_1, y_1)$ and $P_2 = (x_2, y_2)$ be two points on the curve, i.e. which satisfy the equation

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

Then

$$P + Q = (x_3, y_3)$$

where

$$x_3 = \lambda^2 - x_1 - x_2, \qquad y_3 = \lambda(x_1 - x_3) - y_1$$

and

$$\lambda = \begin{cases} \dfrac{y_2 - y_1}{x_2 - x_1} & \text{if } P_1 \neq \pm P2 \\ \dfrac{3x_1^2 + a}{2y_1} & \text{if } P_1 = P_2 \end{cases}$$

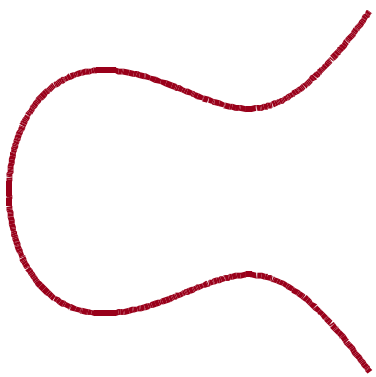
Elliptic Curves are Abelian Groups

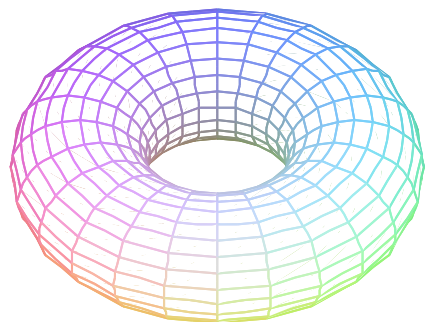
Let $\mathbb{K}$ be a field. The set

$$E(\mathbb{K}) = \{(x, y) \in \mathbb{K}^2 \; ; \; y^2 = x^3 + ax + b\} \cup \{P_\infty\}$$

together with the addition operation form an abelian group

What can we choose for $\mathbb{K}$?





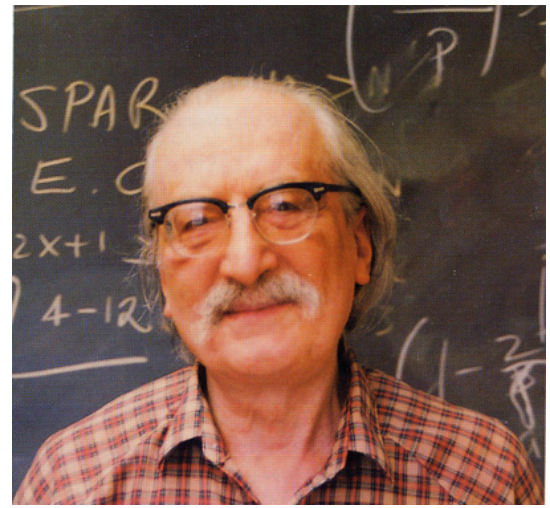
Why elliptic curves over finite fields?

What about $\mathbb{Q}$?
Bad idea: points are either of infinite order or have order less than 12 over the rationals

Good candidates: elliptic curves over $\mathbb{F}_q$. Special cases of interest are when q is a prime or $q = 2^m$

Solving ECDLP

Let E be an elliptic curve over $\mathbb{K}$. Let N be the order of E
ECDLP: given $P, Q \in E$, solve $kP = Q$. (We assume that P generates E)



Shanks' baby-step giant-step algorithm

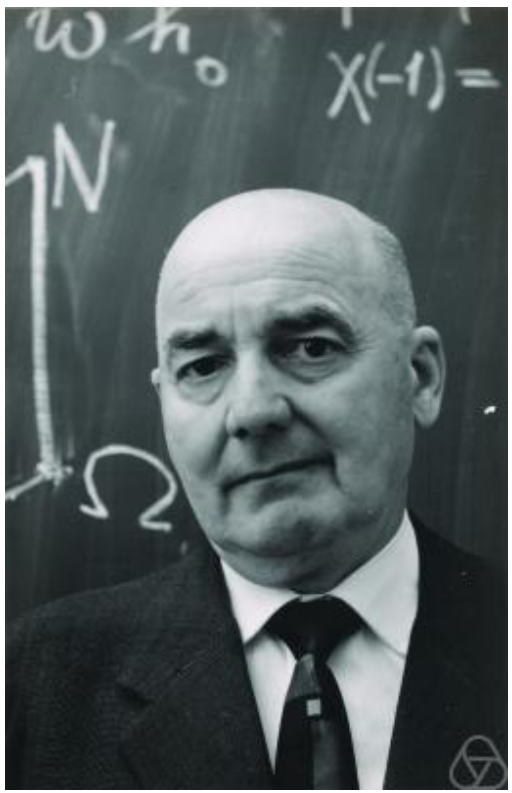
- Fix $m \geq \sqrt{N}$
- Store a list of iP for $0 \leq i < m$ (BS)
- Compute $Q - jmP$ for $j = 0, 1, \ldots, m-1$ until one finds an element from the precomputed list (GS)
- $iP = Q - jmP$, then $Q = kP$ with $k \equiv i + jm \pmod{N}$

Shanks' BSGS complexity: $\tilde{O}(\sqrt{N})$

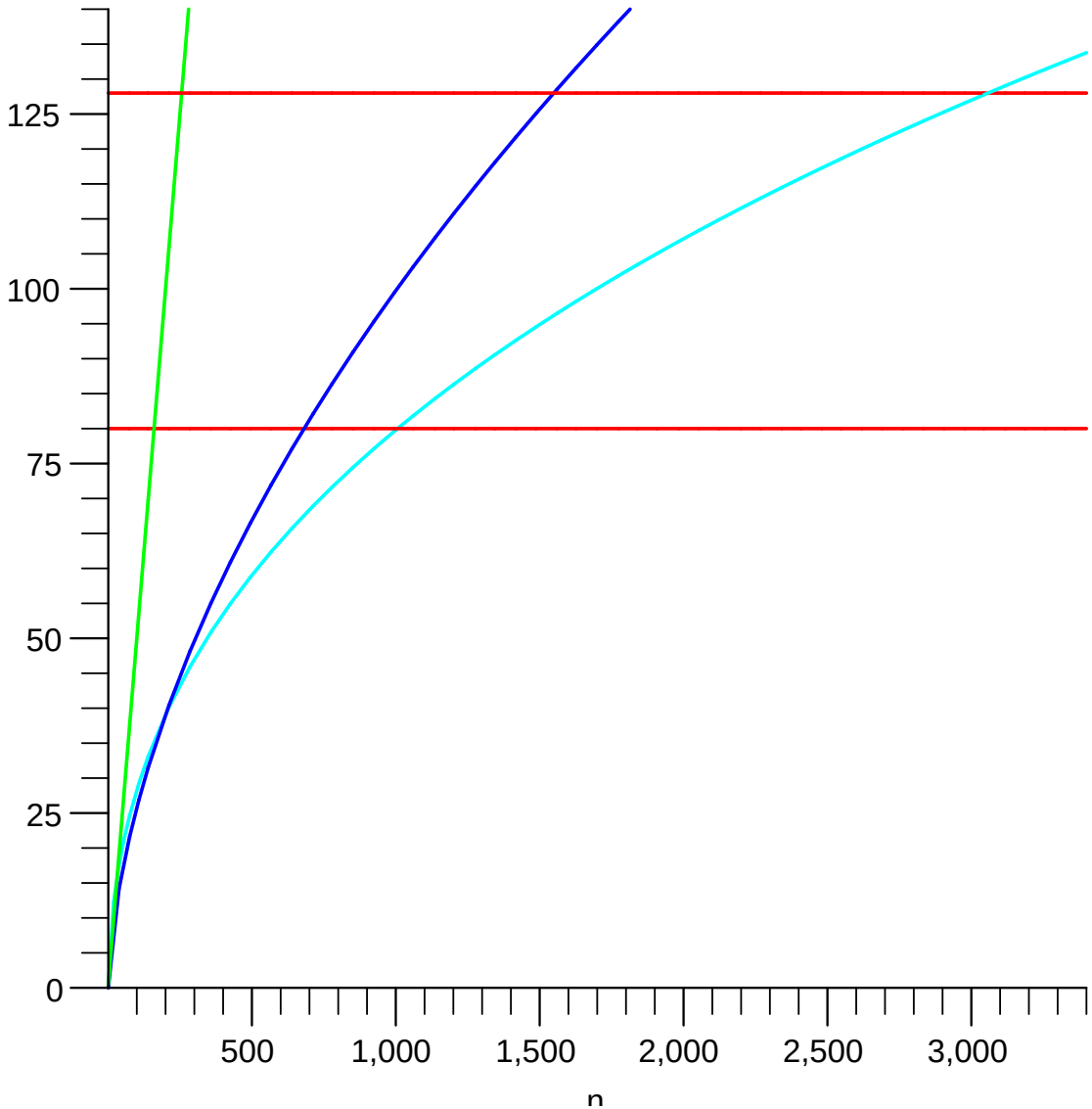
Hasse Theorem

Let E be an elliptic curve over $\mathbb{F}_q$. Then the order of $E(\mathbb{F}_q)$ satisfies

$$|q + 1 - \#E(\mathbb{F}_q)| \leq 2\sqrt{q}$$



So, how large should q be?

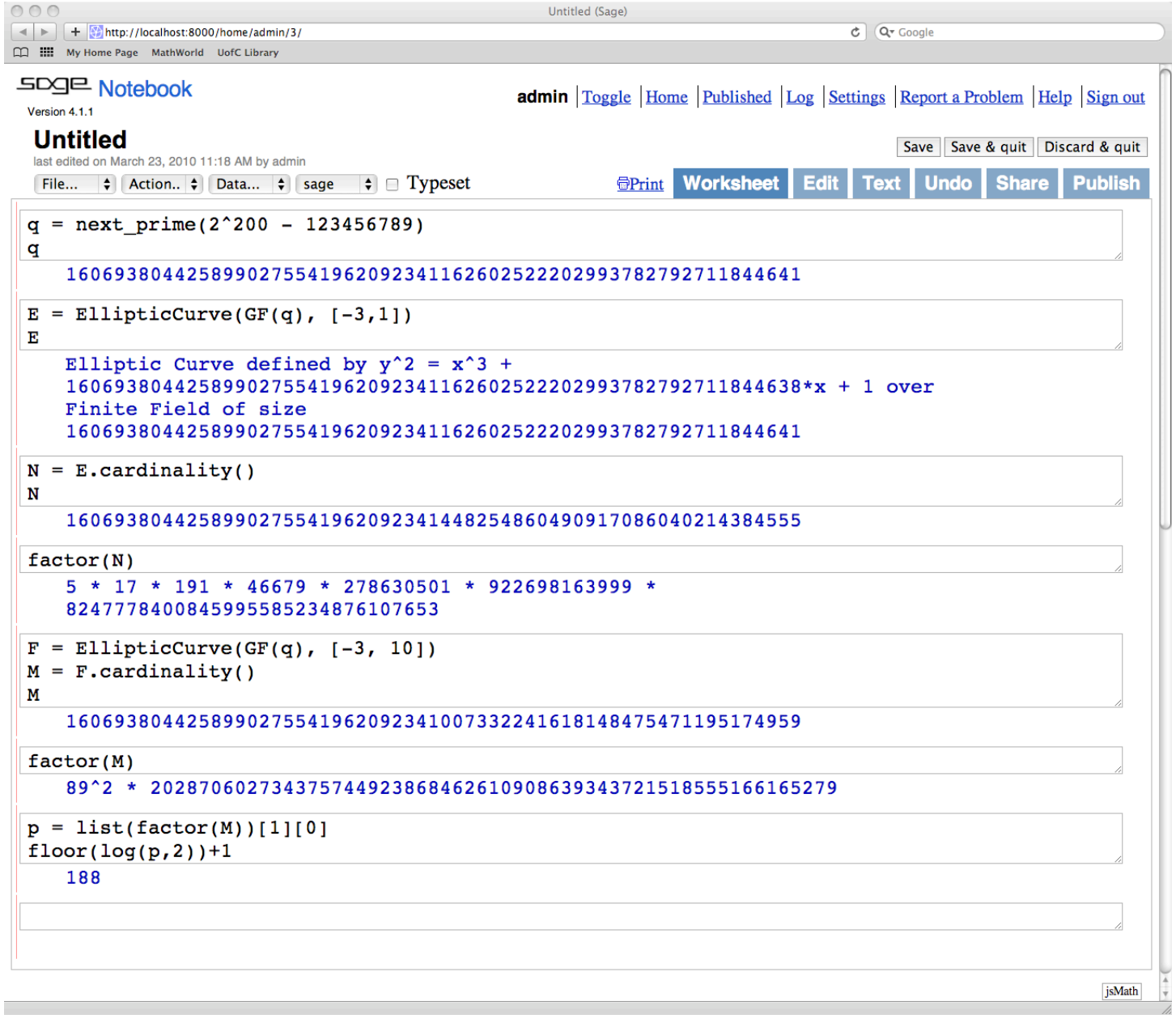


Are all curves good for crypto?

The group order must contain a large prime factor

$$\#E(\mathbb{F}_q) = h \cdot p, \qquad \text{with } p > 2^{160}$$

Shoof's algorithm can be used to count the number of points on elliptic curves over $\mathbb{F}_q$ in polynomial time


Untitled (Sage)
http://localhost:8000/home/admin/3/
My Home Page MathWorld UofC Library
Notebook
Version 4.1.1
admin | Toggle | Home | Published | Log | Settings | Report a Problem | Help | Sign out
Untitled
last edited on March 23, 2010 11:18 AM by admin
Save | Save & quit | Discard & quit
File... Action.. Data... sage Typeset
Print | Worksheet | Edit | Text | Undo | Share | Publish
q = next_prime(2^200 - 123456789)
q
1606938044258990275541962092341162602522202993782792711844641
E = EllipticCurve(GF(q), [-3,1])
E
Elliptic Curve defined by y^2 = x^3 +
1606938044258990275541962092341162602522202993782792711844638*x + 1 over
Finite Field of size
1606938044258990275541962092341162602522202993782792711844641
N = E.cardinality()
N
1606938044258990275541962092341448254860490917086040214384555
factor(N)
5 * 17 * 191 * 46679 * 278630501 * 922698163999 *
824777840084599558523487610765 3
F = EllipticCurve(GF(q), [-3, 10])
M = F.cardinality()
M
1606938044258990275541962092341007332241618148475471195174959
factor(M)
89^2 * 2028706027343757449238684626109086393437215185551661652 79
p = list(factor(M))[1][0]
floor(log(p,2))+1
188
jsMath

ECC In the real world

- Elliptic curve cryptography is now used in many standards (IEEE, NIST, etc.)
- Elliptic curve cryptography is used by the NSA
- The ANSSI (French NSA Information Assurance Dept.) also recommends elliptic curve cryptography (ECDSA)
- Used in Blackberry, Windows Media Player, standards for biometric data on passport, etc.

Merci !

http://www.lirmm.fr/~imbert

Laurent.Imbert@lirmm.fr