How to Keep your Secrets in a Post-Quantum World?

WAM: Uhlenbeck Lectures

Kristin Lauter

Microsoft Research

Course Goals

- Goal: Convey context and status of Post-Quantum Cryptography (PQC)
  - What is PQC?
  - Current Proposals for PQC
  - Familiarity with algorithms and running times
  - Introduce Supersingular Isogeny Graphs (SIG)
  - Introduce Ring-Learning With Errors (RLWE)

Course Outline

- Day 1: Supersingular Isogeny Graphs—definitions and applications
- Day 2: Hard Problems—number theory attacks
- Day 3: RLWE—motivation and definition of schemes
- Day 4: Attacks on Ring-LWE for special rings.

Cryptography:

- The science of keeping secrets!
- But more than that…
  - Confidentiality
  - Authenticity
- Tools:
  - Encryption/Decryption
  - Digital signatures
  - Key exchange

Public Key Cryptography

- Key exchange: two parties agree on a common secret using only publicly exchanged information
- Signature schemes: allows parties to authenticate themselves
- Encryption: preserve confidentiality of data

- Examples of public key cryptosystems:

RSA, Diffie-Hellman, ECDH, DSA, ECDSA

Public Key Cryptography:

- Each party has a *publicly available* key
  - Public key encryption
  - Publicly verifiable signatures
  - Public Key Exchange
- Security of systems in based on some hard math problem:
  - Factoring large integers (RSA)
  - Discrete logarithm problem in (Z/pZ)* (DLP)
  - Elliptic curve groups (ECC):
    - Discrete logarithm problem (ECDLP)
    - Weil pairing on elliptic curves

Applications:

- Secure browser sessions (https: SSL/TLS)
- Signed, encrypted email (S/MIME)
- Virtual private networking (IPSec)
- Authentication (X.509 certificates)

Quantum Computers!

- 1980-82-85: Idea introduced by Benioff, Manin, Feynman, Deutsch
- 1994 Shor's poly time quantum algorithm for factoring
- 2001 factorization of 15 using a 7-qubit NMR computer.
- 2011 researchers factored 143 using 4 qubits
- 2016: Station Q, Microsoft Research, Quantum Compiler,
  LiQuiD

Quantum Arithmetic

- Basic arithmetic is different
- Requires quantum circuits consisting of quantum gates
- Quantum logic gates are represented by unitary matrices
- Dependent on architecture design

Polynomial time Quantum algorithms

- m = # bits
  - Shor's algorithm for factoring $4m^3$ time and $2m$ qbits
  - ECC attack requires $360m^3$ time and $6m$ qbits

  (Proos-Zalka, 2004)

Conclusion:

- RSA: $m = 2048$
- Discrete log $m = 2048$
- Elliptic Curve Cryptography $m = 256$ or 384

Are not resistant to quantum attacks once a quantum computer exists at scale!

Timeline for ECC

- (2006) Suite B set requirements for the use of Elliptic Curve Cryptography
- (2016) CNSA requirements increase the minimum bit-length for ECC from 256 to 384. Advises that adoption of ECC not required.
- (2017) NIST international competition to select post-quantum solutions: PQC Competition

Post-quantum cryptography

- Code-based cryptography (McEliece 1978)
- Multivariate cryptographic systems (Matsumoto-Imai,1988)
- Hash-based cryptosystems (Merkle, 1989)
- Lattice-based cryptography (Hoffstein-Pipher-Silverman, NTRU 1996)
- Supersingular Isogeny Graphs (Charles-Goren-Lauter 2006)

- Challenge! Need to see if these new systems are resistant to *both* classical and quantum algorithms!

Supersingular Isogeny Graphs

New hard problem introduced in 2006: [Charles-Goren-Lauter]

- Finding paths between nodes in a Supersingular Isogeny Graph

Graphs: G = (V, E) = (vertices, edges)

- k-regular, undirected graphs, with optimal expansion
- No known efficient routing algorithm

Hash functions

- A *hash function* maps bit strings of some finite length to bit strings of some fixed finite length

$$h : \{0,1\}^n \rightarrow \{0,1\}^m$$

- easy to compute
- unkeyed (do not require a secret key to compute output)
- Collision resistant
- Uniformly distributed output

Cryptographic Hash functions: Practical applications

- Security of most cryptographic protocols
- Password verification
- Integrity check of received content
- Signed hashes
- Encryption protocols
- Message digest
- Microsoft source code (720 uses of MD5)

Collision-resistance

- A hash function *h* is *collision resistant* if it is computationally infeasible to find two distinct inputs, *x, y*, which hash to the same output $h(x) = h(y)$.
- A hash function h is *preimage resistant* if, given any output of h, it is computationally infeasible to find an input, x, which hashes to that output.

Application: cryptographic hash function [CGL'06]

- k-regular graph G
- Each vertex in the graph has a label

Input: a bit string

- Bit string is divided into blocks
- Each block used to determine which edge to follow for the next step in the graph
- No backtracking allowed!

Output: label of the final vertex of the walk

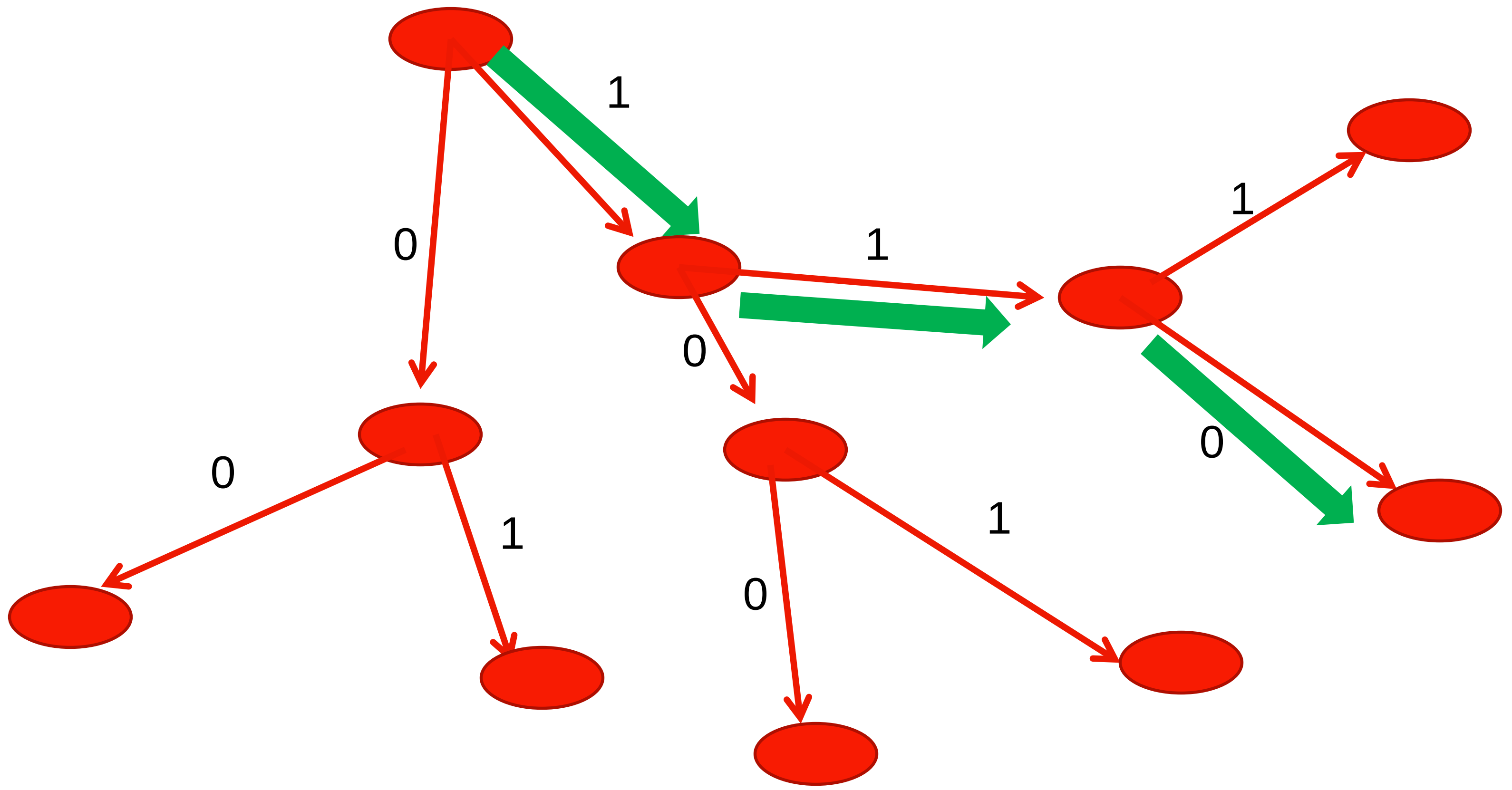
Simple idea

- Random walks on *expander* graphs are a good source of pseudo-randomness
- Are there graphs such that finding collisions is hard? (i.e. finding distinct paths between vertices is hard)
- Bad idea: hypercube (routing is easy, can be read off from the labels)

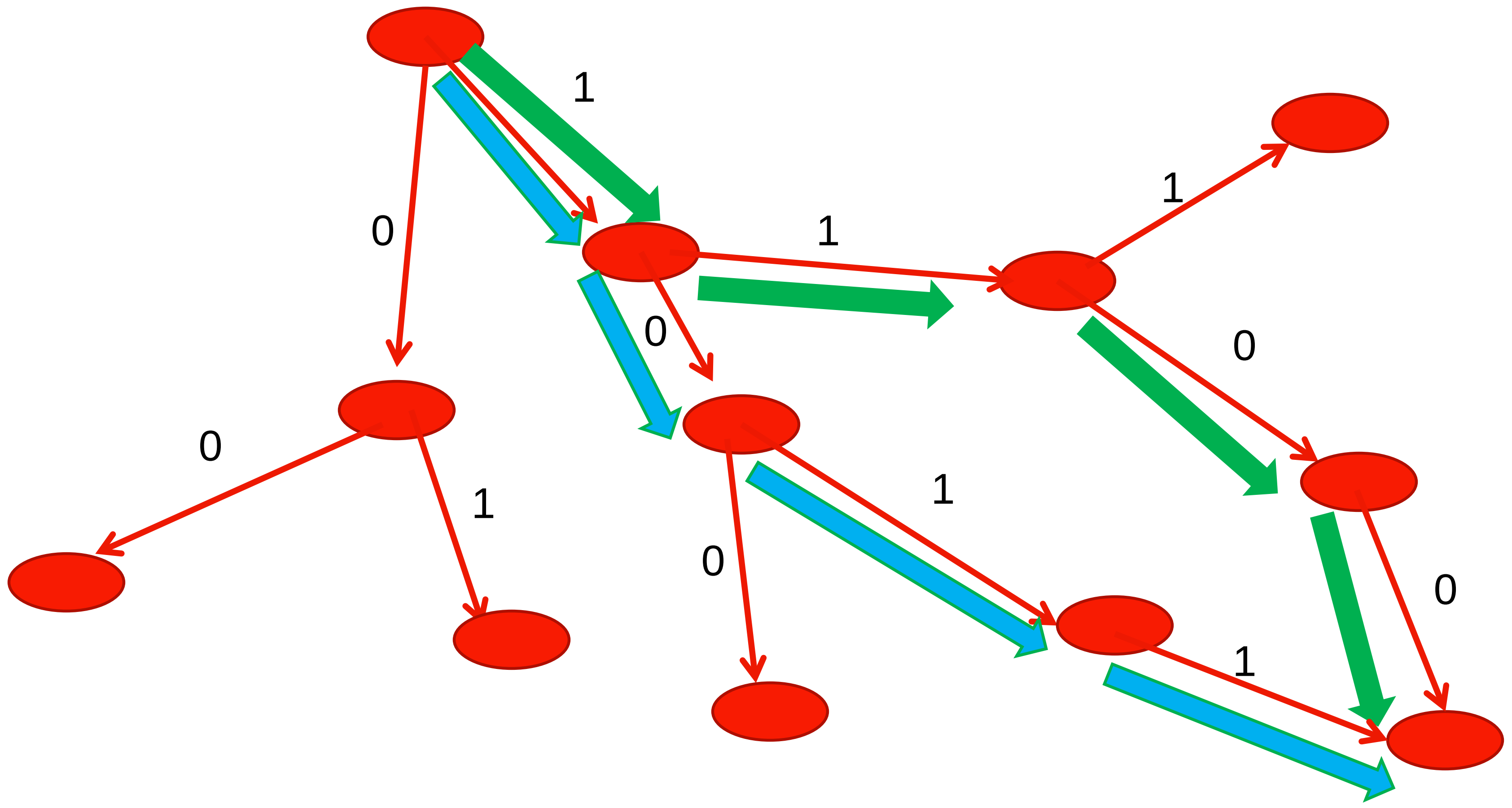
What kind of graph to use?

- Random walks on *expander* graphs mix rapidly: ~log(p) steps to a random vertex, p ~ #vertices
- *Ramanujan* graphs are optimal expanders
- To find a collision: *find two distinct walks of the same length which end at same vertex*

Walk on a graph: 110



Colliding walks: 1100 and 1011



Graph of supersingular elliptic curves modulo p with isogeny edges (Pizer graphs)

- Vertices: supersingular elliptic curves mod p
  - Curves are defined over GF(p^2) (or GF(p))
- Labeled by j-invariants
  - $E_1 : y^2 = x^3 + ax + b$
  - $j(E_1)= 1728*4a^3/(4a^3+27b^2)$
- Edges: Isogenies between elliptic curves

Need to define:

- Elliptic curve
- Supersingular
- Isogeny
- J-invariant

Lots of deep and beautiful theorems in number theory describe the properties of these graphs…

Supersingular is key:

- Graph is Ramanujan (Eichler, Shimura)
- Size, regularity of the graph
- Undirected if we assume p == 1 mod 12

Graph of supersingular elliptic curves modulo p (Pizer)

- Vertices: supersingular elliptic curves mod p
  - # vertices ~ p/12
  - p ~ 2^{256}
  - Curves are defined over GF(p^2)
  - Labeled by j-invariants
- Edges: degree ℓ isogenies between them
  - k = ℓ+1 – regular

Isogenies

- The degree of a separable isogeny is the size of its kernel
- To construct an ℓ -isogeny from an elliptic curve E to another, take a subgroup-scheme C of size ℓ, and take the quotient E/C.
- Formula for the isogeny and equation for E/C were given by Velu.

One step of the walk: (ℓ=2)

$E_1 : y^2 = x^3 + ax + b$

- $j(E_1) = 1728 * 4a^3/(a^3 + 27b^2)$
- 2-torsion point $Q = (r, 0)$

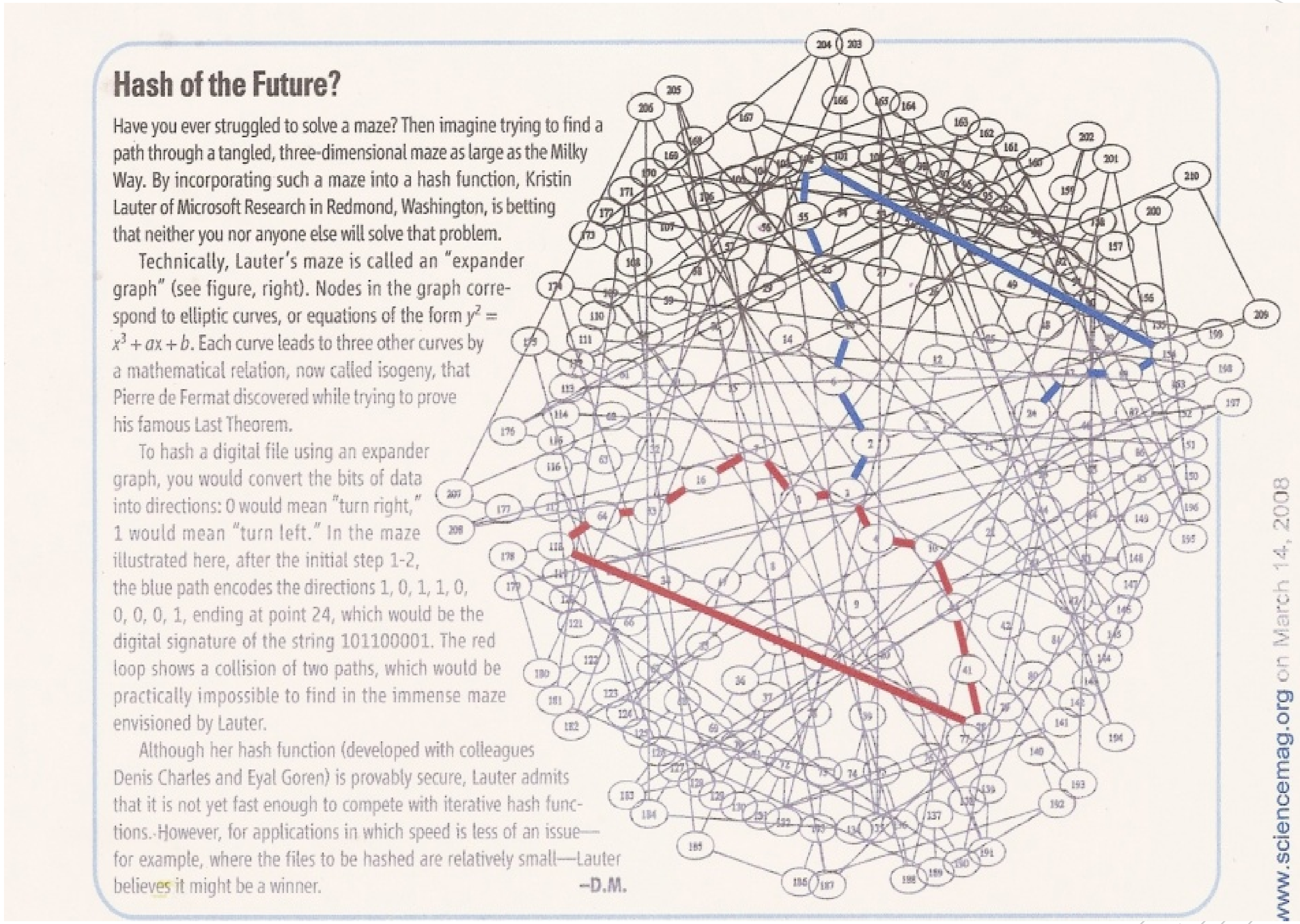
$E_2 = E_1 / Q$ (quotient of groups)

- $E_2 : y^2 = x^3 - (4a + 15r^2)x + (8b - 14r^3)$.

$E_1 \rightarrow E_2$

$(x, y) \rightarrow (x + (3r^2 + a)/(x-r), y - (3r^2 + a)y/(x-r)^2)$

Science magazine 2008



Hash of the Future?

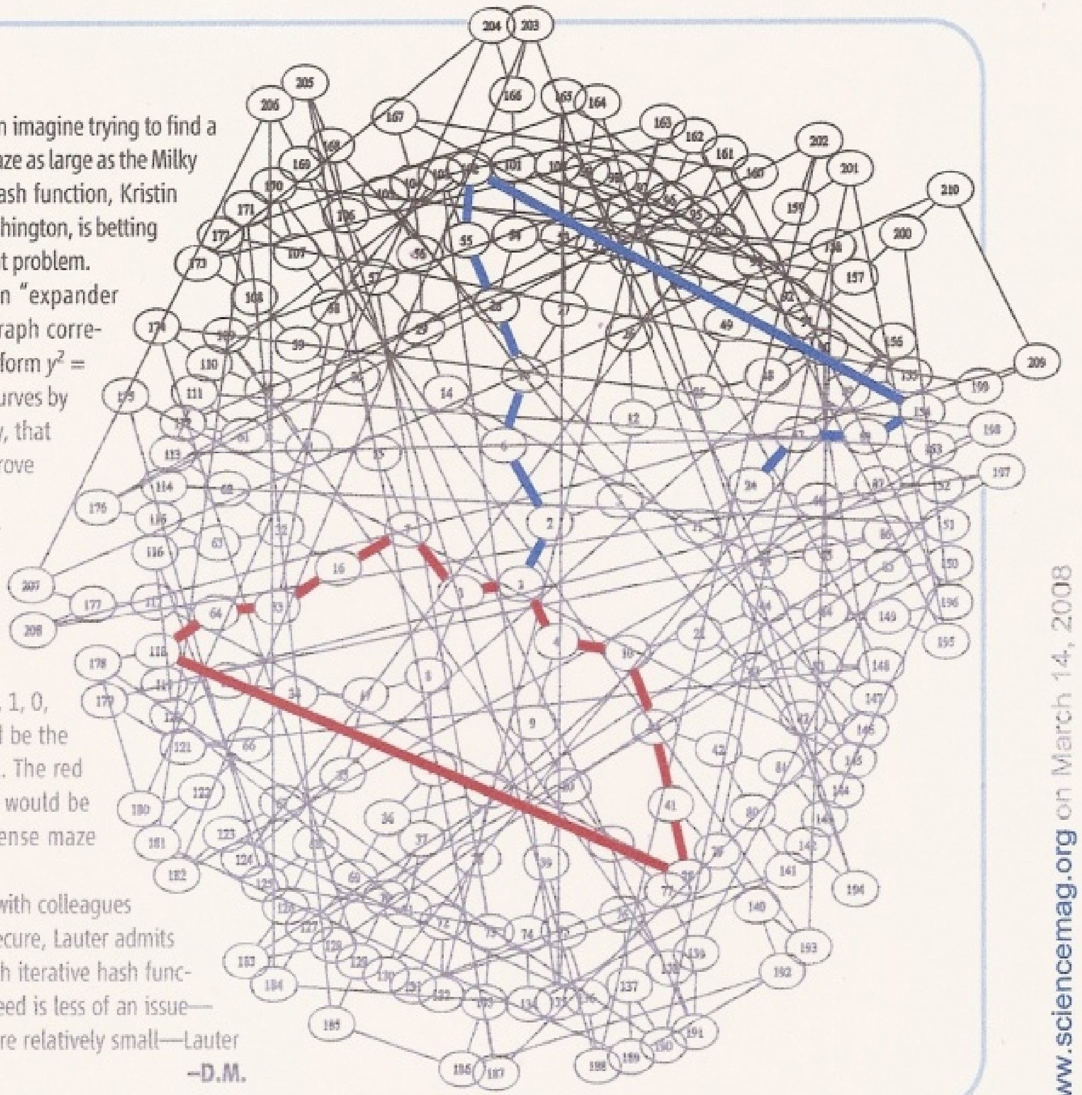
Have you ever struggled to solve a maze? Then imagine trying to find a path through a tangled, three-dimensional maze as large as the Milky Way. By incorporating such a maze into a hash function, Kristin Lauter of Microsoft Research in Redmond, Washington, is betting that neither you nor anyone else will solve that problem.

Technically, Lauter's maze is called an "expander graph" (see figure, right). Nodes in the graph correspond to elliptic curves, or equations of the form $y^2 = x^3 + ax + b$. Each curve leads to three other curves by a mathematical relation, now called isogeny, that Pierre de Fermat discovered while trying to prove his famous Last Theorem.

To hash a digital file using an expander graph, you would convert the bits of data into directions: 0 would mean "turn right," 1 would mean "turn left." In the maze illustrated here, after the initial step 1-2, the blue path encodes the directions 1, 0, 1, 1, 0, 0, 0, 0, 1, ending at point 24, which would be the digital signature of the string 101100001. The red loop shows a collision of two paths, which would be practically impossible to find in the immense maze envisioned by Lauter.

Although her hash function (developed with colleagues Denis Charles and Eyal Goren) is provably secure, Lauter admits that it is not yet fast enough to compete with iterative hash functions. However, for applications in which speed is less of an issue—for example, where the files to be hashed are relatively small—Lauter believes it might be a winner.

–D.M.



Hash of the Future?

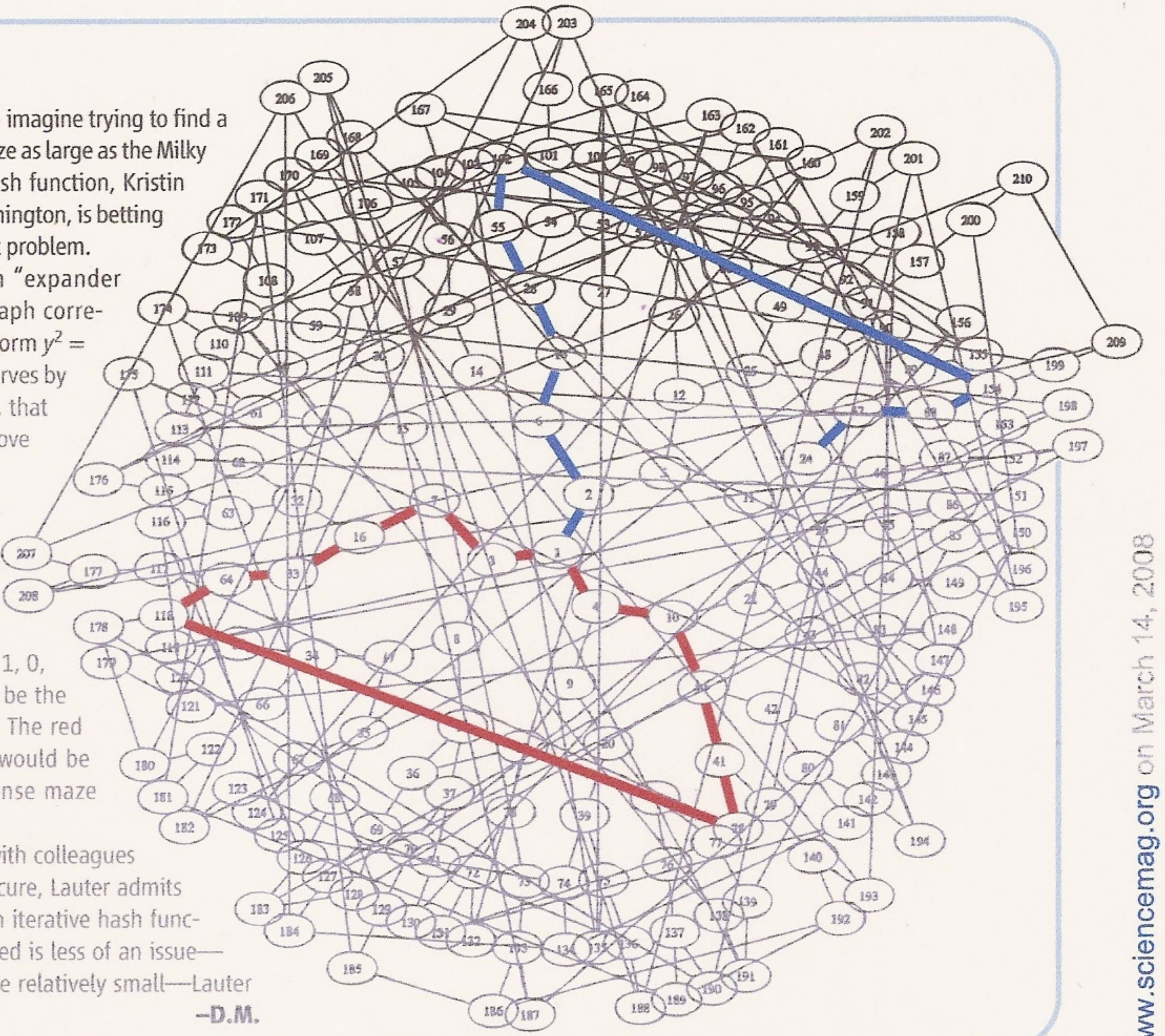
Have you ever struggled to solve a maze? Then imagine trying to find a path through a tangled, three-dimensional maze as large as the Milky Way. By incorporating such a maze into a hash function, Kristin Lauter of Microsoft Research in Redmond, Washington, is betting that neither you nor anyone else will solve that problem.

Technically, Lauter's maze is called an "expander graph" (see figure, right). Nodes in the graph correspond to elliptic curves, or equations of the form $y^2 = x^3 + ax + b$. Each curve leads to three other curves by a mathematical relation, now called isogeny, that Pierre de Fermat discovered while trying to prove his famous Last Theorem.

To hash a digital file using an expander graph, you would convert the bits of data into directions: 0 would mean "turn right," 1 would mean "turn left." In the maze illustrated here, after the initial step 1-2, the blue path encodes the directions 1, 0, 1, 1, 0, 0, 0, 0, 1, ending at point 24, which would be the digital signature of the string 101100001. The red loop shows a collision of two paths, which would be practically impossible to find in the immense maze envisioned by Lauter.

Although her hash function (developed with colleagues Denis Charles and Eyal Goren) is provably secure, Lauter admits that it is not yet fast enough to compete with iterative hash functions. However, for applications in which speed is less of an issue—for example, where the files to be hashed are relatively small—Lauter believes it might be a winner.

–D.M.



History

- Charles-Goren-Lauter presented at NIST 2005 competition, IACR eprint 2006, published J Crypto 2009
- Later in 2006, two papers on eprint, never published:
  - Couveignes, ordinary case (Hard Homogeneous Spaces)
  - Rostovtsev-Stolbunov, ordinary case (Encryption)
- Ordinary case is very different for many reasons:
  - Volcanoe structure of graph
  - Action of an abelian class group
  - Known subexponential classical algorithms to attack

Back-up slides

RSA cryptosystems (~1975)

Security based on hardness of factoring n=p*q

(*n*) = (*p*) (*q*) = (*p* 1)(*q* 1) = *n* - (*p* + *q* -1)

Choose an integer *e* such that gcd(*e*, (*n*)) = 1

Determine *d* as *d* *e* ^{1} (mod (*n*));

Public key (n, *e)*

Private key (n,d)

p, *q*, and (*n*) secret
(because they can be used to calculate *d)*

Encryption

$$c \equiv m^e \pmod{n}$$

Decryption

$$m \equiv c^d \pmod{n}$$

Diffie-Hellman Key Exchange

Given a cyclic group G generated by g

Alice picks random a

Bob picks random b

Alice sends g^a $\longrightarrow$

$\longleftarrow$ Bob sends g^b

Secret :

$g^{ab} = (g^{b})^{a} = (g^{a})^{b}$

Elliptic Curve Cryptography

- Elliptic Curve Cryptography (ECC) is an alternative to RSA and Diffie-Hellman, primarily signatures and key exchange
- Proposed in 1985 (vs. 1975 for RSA) by Koblitz and Miller
- Security is based on a hard mathematical problem different than factoring ECDLP
- ECC 25th anniversary conference October 2010 hosted at MSR Redmond
- *Pairing-based cryptography* currently entirely on pairings on elliptic curves

Elliptic CURVE Groups

- Group of points (x, y) on an elliptic curve,

$$y^2 = x^3 + a x + b,$$

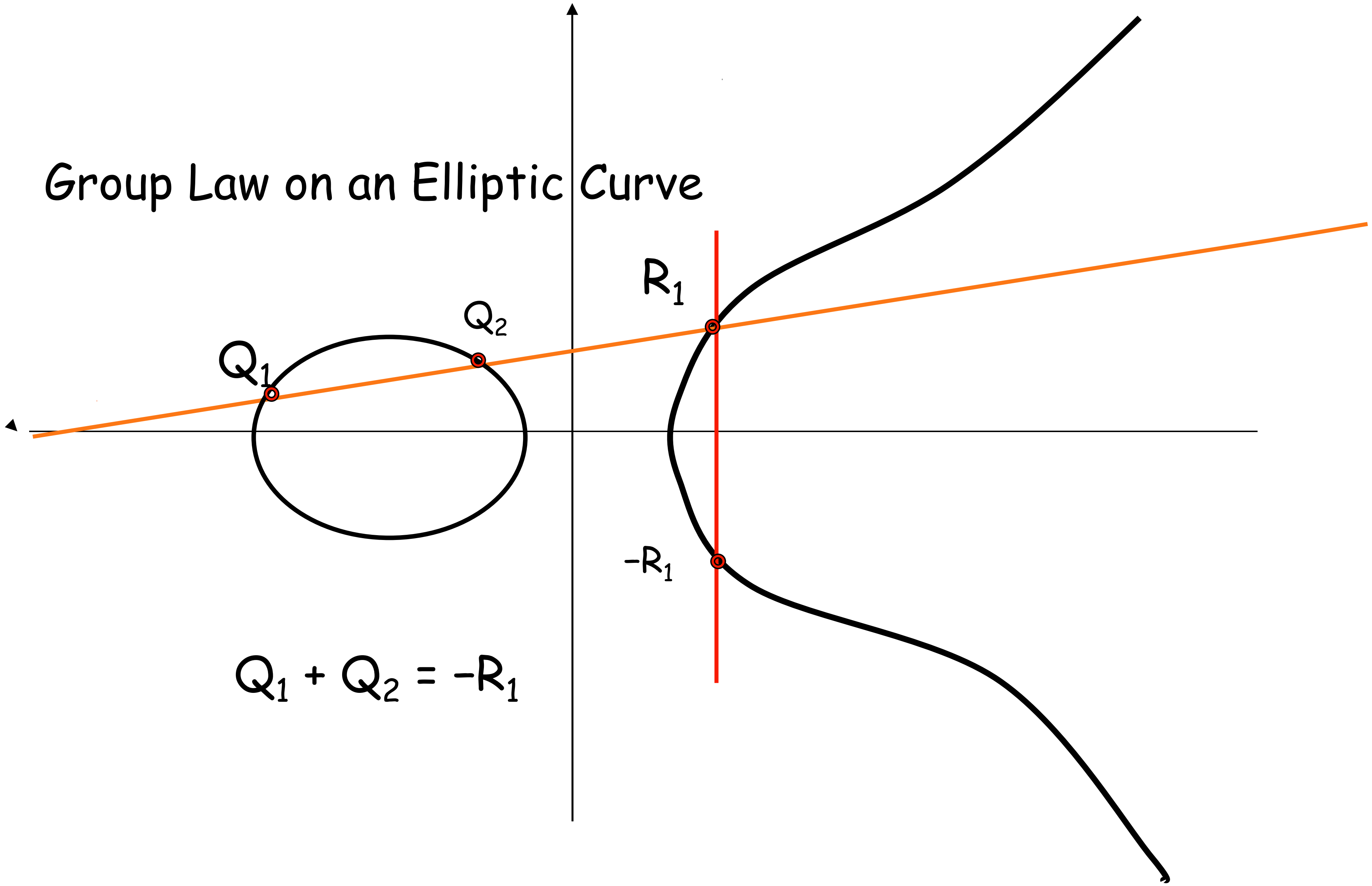
Over a field of minimum size: 256-bits

(short Weierstrass form, characteristic not 2 or 3)

Identity in the group is the “point at infinity”

Group law computed via “chord and tangent method”

Group Law on an Elliptic Curve

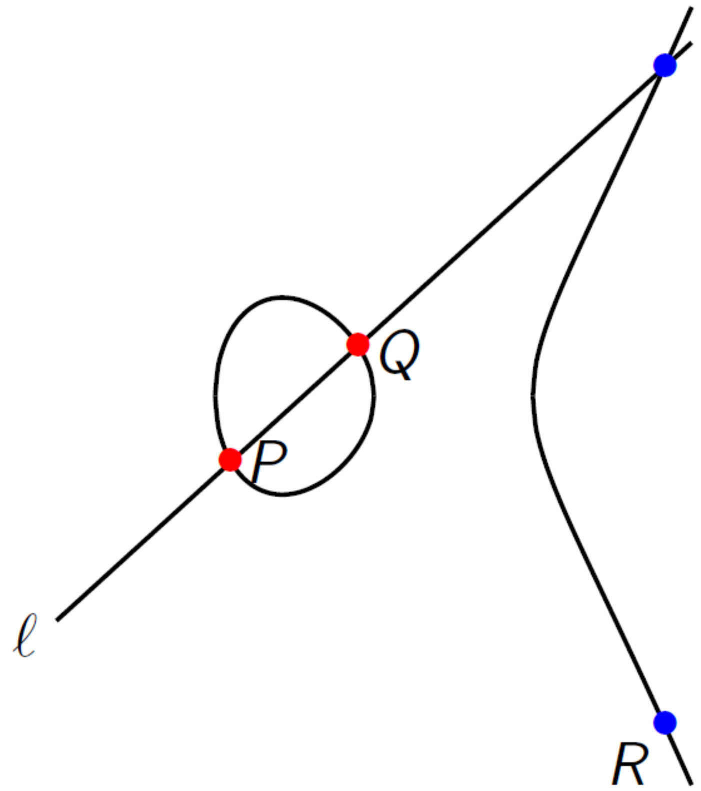


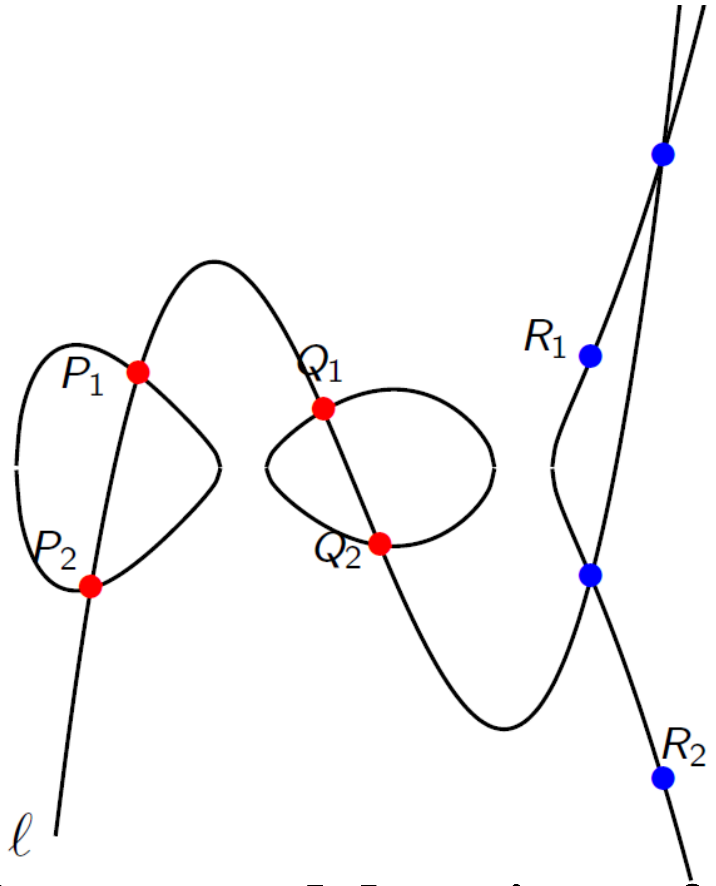
$$Q_1 + Q_2 = -R_1$$

Genus 2 Jacobians

$y^2 = x^3 + a_2x^2 + a_1x + a_0$

$y^2 = x^5 + a_4x^4 + a_3x^3 + a_2x^2 + a_1x + a_0$





How to add pairs of points?

$\#E(\mathbf{F}_p) \approx p$

$\#\mathrm{Jac}_C(\mathbf{F}_p) \approx p^2$

RSA Security

- Security based on hardness of factoring n=p*q
  - p and q have equal size
- Otherwise: Elliptic curve factoring method finds factors in time proportional to the size of the factor (H. Lenstra, `85)
- Quadratic Sieve (Fermat, Kraitchik, Lehmer-Powers, Pomerance)
- Number field sieve (NFS) runs in subexponential time

$$O(e^{c\ (\log n)^{1/3}\ (\log\log n)^{2/3}})$$

c=1.526… Special NFS;

c=1.92… General NFS

Pollard `88, Lenstra-Lenstra-Manasse `90, Coppersmith `93,

Discrete logarithm problem in (Z/pZ)*

- Square-root algorithms:
  - Baby-Step-Giant-Step (Shanks `71)
  - Pollard rho (Pollard, `78)
  - Pohlig-Hellman, `78
- Subexponential:
  - Index calculus (Adleman, `79)
- Recent significant breakthroughs, improving the exponent in subexponential algorithms for DLP to ¼ for small characteristic:
  - Function Field Sieve (Joux 2013)

Elliptic Curve Cryptography

- Menezes–Okamoto–Vanstone (MOV) attack `93:
  - supersingular elliptic curves
- Semaev, Satoh, Smart `98-`99 (Trace 1)
- Generic square-root algorithms:
  - Baby-Step Giant-Step, Pollard's rho
- No generic, classical sub-exponential algorithm known