

Mathematics of Cryptography

A Guided Tour

Joe Hurd

`joe@galois.com`

Galois Tech Talk
Tuesday 28 July 2009

The Tour Starts Here

- This talk will give a guided tour of the mathematics underlying cryptography.
- We'll take apart a related set of **public key** cryptographic algorithms, to see how they work.
- **Disclaimer:** The algorithms are presented in their simplest form—actual systems would implement much more efficient versions.

Talk Plan

- 1 Group Introduction
- 2 Inside RSA
- 3 Case Study
- 4 Elliptic Curves

Diffie-Hellman Key Exchange

The **Diffie-Hellman key exchange** protocol allows two people to use a **public channel** to set up a **shared secret key**:

- 1 Alice and Bob publically agree on a large prime p and an integer x .
- 2 Alice randomly picks an integer a , and sends Bob $x^a \bmod p$.
- 3 Bob randomly picks an integer b , and sends Alice $x^b \bmod p$.
- 4 Alice and Bob both compute $x^{ab} \bmod p$ and use this as a shared secret key.
 - Alice computes $((x^b \bmod p)^a \bmod p) = (x^{ab} \bmod p)$.
 - Bob computes $((x^a \bmod p)^b \bmod p) = (x^{ab} \bmod p)$.

Modular Multiplication Groups

- Multiplication modulo a prime p forms a **group**:
 - There's an **identity** 1 such that $x * 1 = x$.
 - Each element x has an **inverse** x^{-1} such that $x * x^{-1} = 1$.
 - The **operation** $*$ is associative: $x * (y * z) = (x * y) * z$.
- The **order** $|x|$ of x is the smallest n such that $x^n = 1$.
- **Example:** Multiplication modulo 7:

	Operation						Inverse		Order	
*	1	2	3	4	5	6		\cdot^{-1}		$ \cdot $
1	1	2	3	4	5	6	1	1	1	1
2	2	4	6	1	3	5	2	4	2	3
3	3	6	2	5	1	4	3	5	3	6
4	4	1	5	2	6	3	4	2	4	3
5	5	3	1	6	4	2	5	3	5	6
6	6	5	4	3	2	1	6	6	6	2

Group Examples

- **Number groups**
 - Addition of integers $\{\dots, -2, -1, 0, 1, 2, \dots\}$.
 - Multiplication of non-zero real numbers.
- **Permutation groups** (group operation is composition)
 - Substitution ciphers.
 - Card shuffles ($|G| = 52!$, $|\text{riffle}| = 7$).
 - Symmetry groups of regular polygons.
 - Rubik's cube.
- **Product groups** $G \times H$
 - $(x_1, y_1) *_{G \times H} (x_2, y_2) = (x_1 *_G x_2, y_1 *_H y_2)$
 - $1_{G \times H} = (1_G, 1_H)$.
 - $(x, y)^{-1} = (x^{-1}, y^{-1})$.

Group Exponentiation

- Given a group G , we can efficiently compute exponentiation x^n using **repeated squaring**:
 - 1 If $n = 0$ then return the group identity,
 - 2 else if n is even then return $(x * x)^{n/2}$,
 - 3 else return $x * (x^{n-1})$.
- Computing x^n using repeated squaring requires $O(\log n)$ group operations.

The Discrete Logarithm Problem

- Given a group G , the **Discrete Logarithm Problem** tests the difficulty of inverting exponentiation:
 - Given $g, h \in G$, find a k such that $g^k = h$.
- The difficulty of this problem depends on the group G .
 - For addition modulo p , the problem can be solved in $O(\log |G|)$ time.
 - For an ideal black-box group G , solving the discrete logarithm problem requires $O(\sqrt{|G|})$ group operations.
- For multiplication modulo p , the problem is hard.
 - **But:** The best known algorithm can solve it faster than black-box.
 - **And:** Odlyzko (1991) broke the secure identification option of the Sun Network File System which used a prime of 192 bits.

Group Encryption: ElGamal

The **ElGamal encryption algorithm** can use **any instance** $g^k = h$ of the Discrete Logarithm Problem.

- 1 Alice obtains a copy of Bob's public key (g, h) .
- 2 Alice generates a randomly chosen natural number $i \in \{1, \dots, |G| - 1\}$ and computes $a = g^i$ and $b = h^i m$.
- 3 Alice sends the encrypted message (a, b) to Bob.
- 4 Bob receives the encrypted message (a, b) . To recover the message m he uses his private key k to compute

$$a^{-k} b = (g^i)^{-k} h^i m = g^{-ik} (g^k)^i m = g^{ki-ik} m = m .$$

Talk Plan

- 1 Group Introduction
- 2 Inside RSA**
- 3 Case Study
- 4 Elliptic Curves

Subgroups

- A group H is a **subgroup** of a group G if $H \subseteq G$ and H has the same operation, inverse and identity.
 - **Example:** Integer addition is a subgroup of real addition.
 - **Example:** Substitution ciphers mapping $A \mapsto A$ are a subgroup of all substitution ciphers.
 - **Non-example:** Substitution ciphers mapping $A \mapsto B$ are not a subgroup of anything (no identity, not a group).
- A group G has two trivial subgroups:
 - the whole group G ; and
 - the subgroup consisting of just the identity.

Lagrange's Theorem

- **Theorem:** If H is a subgroup of a finite group G , then $|H|$ divides $|G|$.
 - **Proof:** Define the equivalence relation $g_1 \sim g_2$ iff there exists $h \in H$ such that $h * g_1 = g_2$.

Lagrange's Theorem

- **Theorem:** If H is a subgroup of a finite group G , then $|H|$ divides $|G|$.
 - **Proof:** Define the equivalence relation $g_1 \sim g_2$ iff there exists $h \in H$ such that $h * g_1 = g_2$.
- **Corollary:** For each element $g \in G$, $|g|$ divides $|G|$.
 - **Proof:** Each group element $g \in G$ generates a subgroup $\{g^n \mid 0 \leq n < |g|\}$.

Lagrange's Theorem

- **Theorem:** If H is a subgroup of a finite group G , then $|H|$ divides $|G|$.
 - **Proof:** Define the equivalence relation $g_1 \sim g_2$ iff there exists $h \in H$ such that $h * g_1 = g_2$.
- **Corollary:** For each element $g \in G$, $|g|$ divides $|G|$.
 - **Proof:** Each group element $g \in G$ generates a subgroup $\{g^n \mid 0 \leq n < |g|\}$.
- **Corollary:** For each element $g \in G$, $g^{|G|}$ is the identity.
 - **Proof:** $g^{|G|} = g^{|g|k} = (g^{|g|})^k = 1^k = 1$.

RSA Encryption

- 1 Bob chooses two large primes p, q and computes $n = pq$.
- 2 Bob chooses an integer e and computes d such that

$$ed \bmod (p - 1)(q - 1) = 1 .$$

- 3 Bob publishes (n, e) as his public key.
- 4 Alice takes her message m and computes $c = m^e \bmod n$.
- 5 Alice sends c to Bob.
- 6 Bob receives c and computes

$$c^d \bmod n = (m^e \bmod n)^d \bmod n = m^{ed} \bmod n = m .$$

“The Magic Words are Squeamish Ossifrage”

- **Chinese Remainder Theorem:** Multiplication modulo n is the **product group** of multiplication modulo p and multiplication modulo q .
- The group of multiplication modulo a prime p consists of elements $\{1, \dots, p-1\}$, and thus has size $p-1$.
- The group G of multiplication modulo n therefore has size $(p-1)(q-1)$, and so

$$\begin{aligned}m^{ed} \bmod n &= m^{k(p-1)(q-1)+1} \bmod n \\ &= m^{k|G|+1} \bmod n \\ &= (m^{|G|} \bmod n)^k m \bmod n \\ &= 1^k m \bmod n \\ &= m \quad \square\end{aligned}$$

Talk Plan

- 1 Group Introduction
- 2 Inside RSA
- 3 Case Study**
- 4 Elliptic Curves

Blum Integers

- **Fact:** Given a prime p such that $p \bmod 4 = 3$, exactly one of x and $-x$ has square roots. If x has square roots, they can be computed by $\pm(x^{(p+1)/4} \bmod p)$.
- A number n is a **Blum integer** if $n = pq$ with p, q primes equal to 3 modulo 4.
- **Theorem:** If n is a Blum integer and x is a square mod n , then x has four square roots and exactly one of these is itself a square mod n . Call this unique square root the **principal square root**.
- **Theorem:** Computing square roots modulo n is RP-equivalent to factoring n .

Bit Commitment

This protocol allows Alice and Bob to fairly flip a coin over a network.

- 1 Alice randomly picks a large Blum integer $n = pq$ and an integer x .
- 2 Alice computes $y = x^2 \bmod n$, and $z = y^2 \bmod n$.
- 3 Alice sends Bob (n, z) .
- 4 Bob has to guess whether y lies in the range $H = (0, \frac{1}{2}n)$ or the range $T = (\frac{1}{2}n, n)$.
- 5 Bob randomly picks H or T and sends his guess to Alice.
- 6 Alice sends Bob (p, q, x) .

Zero-Knowledge Proof

- Let Alice have a secret: a Hamilton cycle H in a large graph G .
- The bit commitment protocol can be built upon to allow Alice to prove she knows the secret to Bob, but without revealing it:
 - 1 Alice randomly permutes all the vertex labels on G to create a new graph G' .
 - 2 She then makes two commitments: the vertex pairing she used $f : G \rightarrow G'$; and the new Hamilton cycle $H' = f(H)$.
 - 3 She sends G' and these commitments to Bob.
 - 4 Bob randomly chooses either H' or f , and sends his choice to Alice.
 - 5 Alice sends Bob the information he needs to reveal his choice.

Talk Plan

- 1 Group Introduction
- 2 Inside RSA
- 3 Case Study
- 4 Elliptic Curves**

Elliptic Curve Cryptography

- First proposed in 1985 by Koblitz and Miller.
- Part of the 2005 NSA Suite B set of cryptographic algorithms.
- Certicom the most prominent vendor, but there are many implementations.
- Advantages over standard public key cryptography:
 - Known theoretical attacks much less effective,
 - so requires much shorter keys for the same security,
 - leading to **reduced bandwidth** and **greater efficiency**.
- However, there are also disadvantages:
 - The algorithms are **more complex**, so it's harder to implement them correctly.
 - **Patent uncertainty** surrounding many implementation techniques.

Elliptic Curves

- An elliptic curve is the set of points (x, y) satisfying an equation of the form

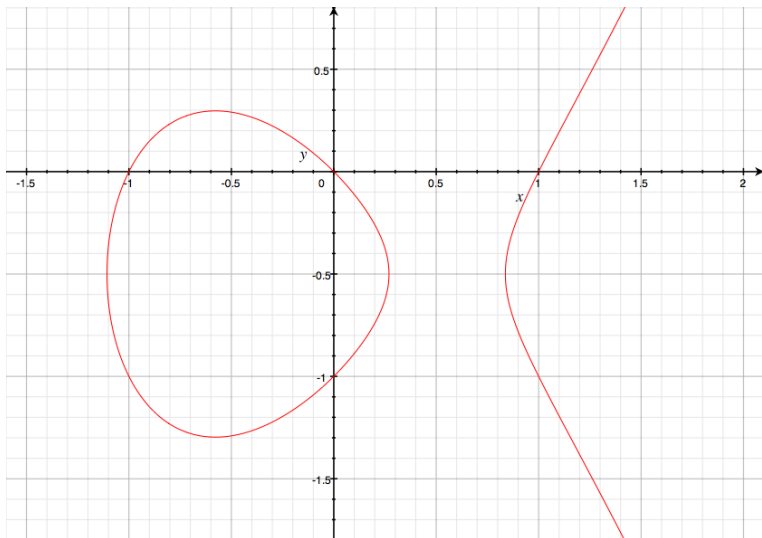
$$y^2 + a_1xy + a_3y = x^3 + a_2x^2 + a_4x + a_6 .$$

- Despite the name, they don't look like ellipses!
- Elliptic curves are used in number theory: Wiles proved Fermat's Last Theorem by showing that the elliptic curve

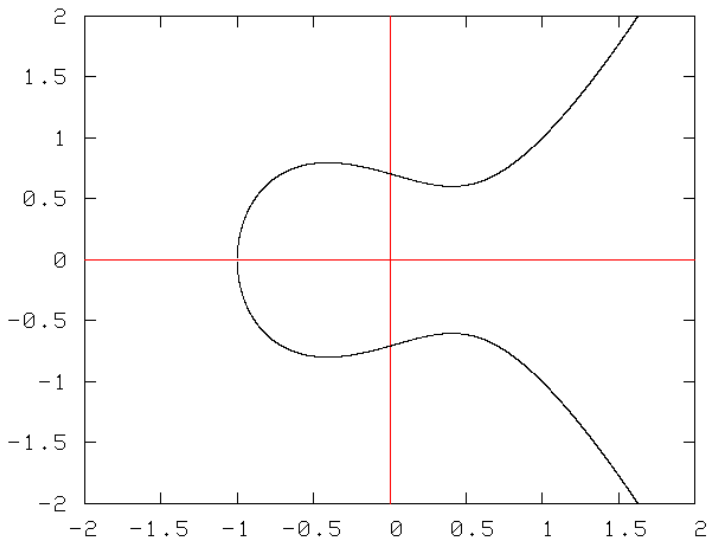
$$y^2 = x(x - a^n)(x + b^n)$$

generated by a counter-example $a^n + b^n = c^n$ cannot exist.

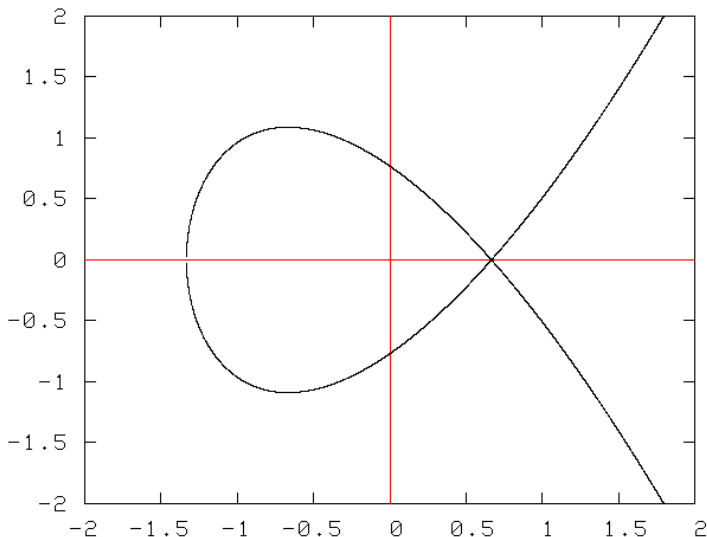
Example Elliptic Curve $y^2 + y = x^3 - x$



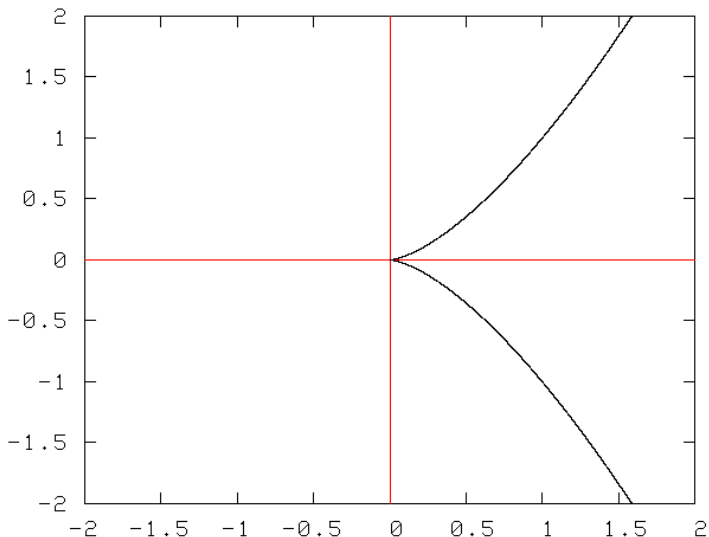
Example Elliptic Curve $y^2 = x^3 - \frac{1}{2}x + \frac{1}{2}$



Example Elliptic Curve $y^2 = x^3 - \frac{4}{3}x + \frac{16}{27}$



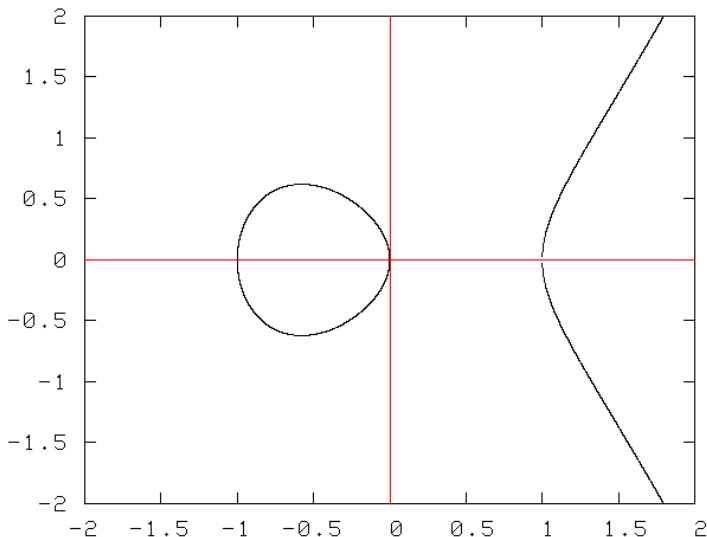
Example Elliptic Curve $y^2 = x^3$



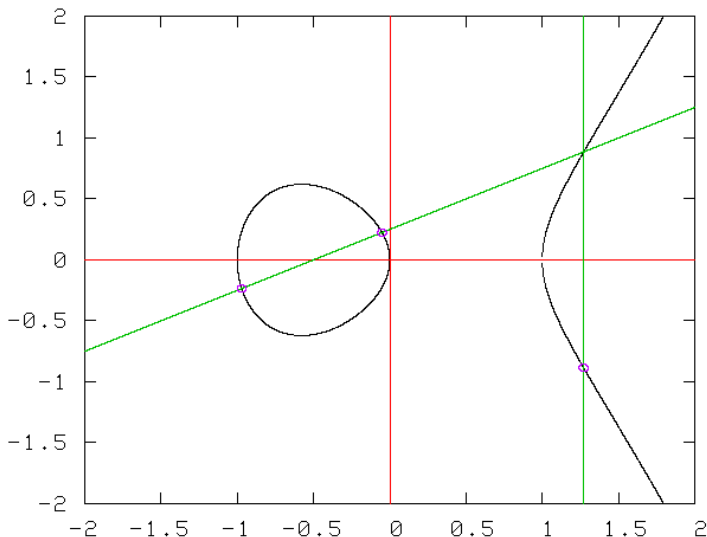
Elliptic Curve Group

- **Fact:** The points (x, y) satisfying the elliptic curve equation form a group.
- It's possible to 'add' two points on an elliptic curve to get a third point on the curve.
- The identity is a special zero point \mathcal{O} *infinitely far up the y-axis*.

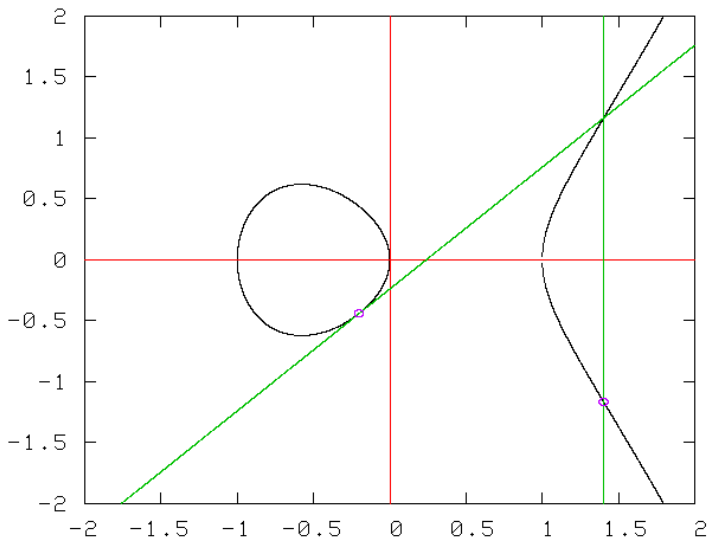
Example Elliptic Curve $y^2 = x^3 - x$



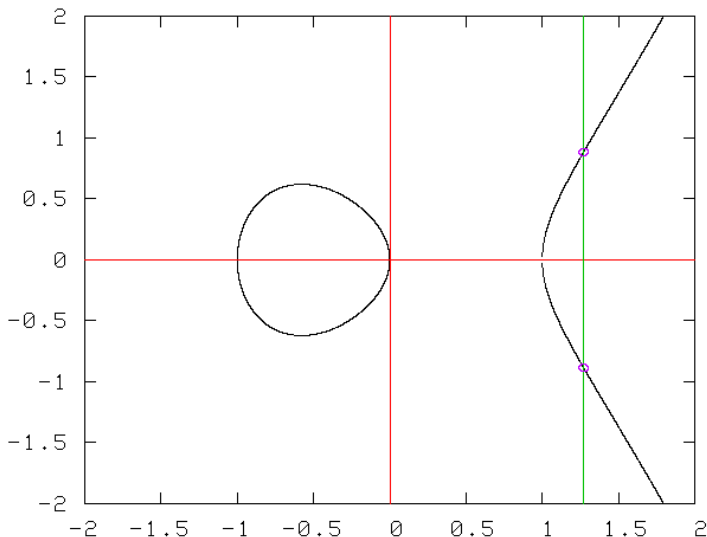
Example Elliptic Curve $y^2 = x^3 - x$: Addition



Example Elliptic Curve $y^2 = x^3 - x$: Doubling



Example Elliptic Curve $y^2 = x^3 - x$: Negation



Elliptic Curve Cryptography

- The graphs showed elliptic curves points (x, y) where x and y were real numbers.
- But the elliptic curve operations can be defined for any underlying field.
- Instead of the geometric definition, use algebra:

$$-(x, y) = (x, -y - a_1x - a_3) .$$

- Elliptic curve cryptography uses **finite fields** $\text{GF}(p^n)$.
 - $\text{GF}(p)$ is the field $\{0, \dots, p - 1\}$ where all arithmetic is performed modulo the prime p .
 - $\text{GF}(2^n)$ is the field of polynomials where all the coefficients are either 0 or 1.