Public-key Cryptography and elliptic curves

Dan Nichols
nichols@math.umass.edu

University of Massachusetts

Oct. 14, 2015

Cryptography is the study of secure communications. Here are some important terms:

- Alice wants to send a message (called the plaintext) to Bob.
- To make sure only the Bob can understand the message, she encrypts it, transforming the plaintext into the ciphertext
- Bob can decrypt the ciphertext and reveal the plaintext, but other people cannot.
- A cipher is an algorithm for performing encryption and/or decryption

Symmetric cryptography

- In a symmetric cipher, the same key is used for both encryption and decryption.
- Alice and Bob must both share the same key, and make sure no one else has access to it.
- This is difficult – how do they exchange keys securely?

- Analogy: a locked safe. Both Alice and Bob have copies of the key to open it. Each can leave messages there for the other to find.

Here's a cipher used by Julius Caesar: to encrypt a message, shift each letter of the alphabet forward by N letters.

- if $N = 3$, replace every letter with the letter three steps after it in the alphabet.
  - `a` $\rightarrow$ `d`, `b` $\rightarrow$ `e`, etc.
  - '`cipher`' $\rightarrow$ '`flskhu`'
- Decrypt by shifting each letter back N steps
- The secret key is N

An easy example

Why is this cipher so easy to break?

- The key space (number of possible keys) is small: only **26** possible keys
  - You could easily break this cipher with a brute force attack: try every key until you find the right one.
- The cipher does not hide all the statistical properties of the message
  - Check the frequency in which each letter appears. Compare to the known frequencies of letters in English text. This is an example of an analytic attack.

Symmetric Cryptography

- Today we have much stronger symmetric ciphers available such as AES (Advanced Encryption Standard)
  - Huge key space – brute force attacks are effectively impossible
  - Carefully designed to prevent analytic attacks
- But all symmetric ciphers, no matter how strong, share some of the same inherent weaknesses:
  - Both parties must first communicate securely to share a secret key, which requires an already secure channel.
  - In a network of people, each pair (e.g. Alice, Bob) needs its own shared key.
    - With N people, that's $N(N-1)/2$ keys in total.

- Public-key cryptography (or asymmetric cryptography) solves these problems
- Basic idea: instead of Alice and Bob sharing a secret key, each person has their own public key and their own private key
- Invented* in 1976 by Whitfeld Diffie, Martin Hellman, and Ralph Merkle
    - Invented years earlier by GCHQ (and probably NSA), but not revealed to the public.

Public-key cryptography

Public-key cryptography outline:

1. Bob generates both a public key and a private key.
   - He makes his public key visible to everyone but keeps his private key secret
2. Alice encrypts a message using Bob's public key, and sends it to Bob
3. Bob can decrypt the message using his private key

So anyone who wants to send Bob an encrypted message can do so using Bob's public key. But decrypting these messages requires Bob's private key, which only Bob has!

- No secure channel necessary. Alice can send Bob a message without them sharing the same secret key.
- In a network, each person just needs their own public key and private key.
    - In a network of N people, this is N public keys and N private keys in total.

- Analogy: each person has their own locked mailbox with a slot to accept incoming messages

How are public-key algorithms used?

- In practice, public-key cryptosystems are much slower and less efficient compared to symmetric ciphers. Not a good way to send large messages quickly.
- These days secure communication usually works like this:
  - Use a public-key protocol to securely exchange symmetric keys for a fast symmetric cipher such as AES.
  - Then we use this symmetric cipher to actually exchange messages.
  - Best of both worlds!

How are public-key algorithms constructed?

- Based on mathematical trapdoor functions: easy computations that are hard to reverse.
  - More technically: a one-to-one function f where it's easy to compute $y = f(x)$ but hard to compute $x = f^{-1}(y)$.
- Example: RSA (Rivest, Shamir, Adelman) is based on the problem of factoring a huge integer into a product of prime numbers
  - If you have two large prime numbers, it's easy to multiply them together
  - But if you have a huge number that you know is the product of two primes, it is very hard to find out what those primes are!
- Another trapdoor is the discrete logarithm problem

Diffie-Hellman key exchange

- One fairly simple public-key scheme is Diffie-Hellman Key Exchange (DH)
  - Allows two people to securely generate a shared key without anyone else knowing. This key can then be used to communicate using a symmetric cipher.
- Before we can study this algorithm, we need a quick number theory primer.

Modular arithmetic

We want to define a system of arithmetic that is 'closed' on a finite set of numbers. For example, let's use the set $\{0, 1, 2, 3, 4\}$ (first five numbers, starting from zero).

- Problem: when we add (or multiply), the numbers get too big.
  - $3 + 4 = 7$ (outside the set)
  - We want to be able to add and multiply anything and never deal with numbers above 4.
- Solution: for numbers outside the set, we 'wrap around' and consider only the remainder when divided by 5. This works for both $+$ and $\times$.
- Remember that when we divide a number by m, the remainder is always between 0 and $m - 1$.

- We say $a \equiv b \mod m$ (a and b are equivalent modulo m) if m divides $(b - a)$.
  - Or equivalently, if the remainder of $a \div m$ is the same as the remainder of $b \div m$.
  - Examples:
    - $10 \equiv 0 \mod 5$ because 5 divides $10 - 0$
    - $-12 \equiv 3 \mod 5$ because 5 divides $-12 - 3$
- Suppose we want to add $3 + 4$. Normally this would be 7, which is outside our set. But $7 \equiv 2 \mod 5$, so we can say

$$3 + 4 \equiv 2 \mod 5.$$

You can add and multiply the numbers $0, 1, 2, 3, 4$ using this rule, and the answer always stays within this set!

Let's look at the addition table and multiplication table modulo 5:

$+$	**0**	**1**	**2**	**3**	**4**
0	0	1	2	3	4
1	1	2	3	4	0
2	2	3	4	0	1
3	3	4	0	1	2
4	4	0	1	2	3

$\times$	**0**	**1**	**2**	**3**	**4**
0	0	0	0	0	0
1	0	1	2	3	4
2	0	2	4	1	3
3	0	3	1	4	2
4	0	4	3	2	1

The integers modulo m

- We call m the modulus.
- The set of numbers $0, 1, 2, \ldots, m-1$ together with these rules for addition and multiplication is called $\mathbb{Z}/m$, the ring of integers modulo m.
- Modular arithmetic obeys nice algebraic rules, so it's consistent and coherent:
  - If $a \equiv b \mod m$ and $c \equiv d \mod m$, then
    - $ac \equiv bd \mod m$
    - and $a + c \equiv b + d \mod m$

The integers modulo m: examples

- $\mathbb{Z}/14$: (integers modulo 14):
  - $8 + 10 \equiv 4 \mod 14$
    because $8 + 10 = 18$, and $18 \div 14 = 1$ remainder 4.
  - $5 \times 6 \equiv 2 \mod 14$
    because $5 \times 6 = 30$, and $30 \div 14 = 2$ remainder 2.
- $\mathbb{Z}/32$: (integers modulo 32)
  - $11 + 30 \equiv 9 \mod 32$
  - $10 \times 7 \equiv 6 \mod 32$
- $\mathbb{Z}/2147483647$: (integers modulo 2,147,483,647)
  - $235 \times 325284906 \equiv 1280025265$

The integers modulo m: prime modulus

- The number we used in that last example is actually a prime number: 2,147,483,647
  - This very useful number is $2^{31} - 1$, the largest number you can store on a computer using 32 bits.
- If p is a prime number, $\mathbb{Z}/p$ has an important property: every number modulo p has an inverse
  - For every a in $\mathbb{Z}/p$, there's some a^{-1} such that $a \times a^{-1} \equiv 1 \mod p$
    - For example, in $\mathbb{Z}/7$, we have $2 \times 4 \equiv 1 \mod 7$, so the inverse of 2 is $2^{-1} = 4$.
  - This means you can 'divide' by any number: $a \div b = ab^{-1}$
- But if m is not prime, there will be some numbers in $\mathbb{Z}/m$ that don't have a multiplicative inverse.
  - For example, when $m = 6$, there's no inverse of $2 \mod 6$.
- For a prime number p, we call $\mathbb{Z}/p$ the finite field with p elements, or $\mathbb{F}_p$. Extremely important in number theory

Modular exponentiation

Since we can multiply things in $\mathbb{Z}/m$, we can also raise a number to an integer power.

- Computing $3^3 \mod 7$:
  $3^3 = 3 \times 3 \times 3 = 27 \equiv 6 \mod 7$
- Computing $4^5 \mod 100$:
  $4^5 = 4 \times 4 \times 4 \times 4 \times 4 = 1024 \equiv 24 \mod 100$

Suppose we want to calculate $11^{32} \mod 81$.

- Simplest way: find 11^{32} (a 33-digit number), take the remainder modulo 81
- Faster way:
    - Start by calculating $11^2 = 121 \equiv 40 \mod 81$
    - Then square that: $11^4 = (11^2)^2 \equiv 40^2 = 1600 \equiv 61 \mod 81$
    - Square it again: $11^8 = (11^4)^2 \equiv 61^2 = 3721 \equiv 76 \mod 81$
    - Eventually we get $11^{32} \equiv 58 \mod 81$
- There's a way to do this for exponents that aren't powers of 2 with only slightly more work. We never have to work with numbers bigger than m^2.
- So it's very fast and easy for a computer to calculate $b^c \mod m$.

It's easy to compute powers in $\mathbb{Z}/m$. What about logarithms?

- In $\mathbb{R}$ (real numbers), it's easy to calculate $\log x$ for any $x > 0$.
  - $y = \log x$ is a continuous function, so it's easy to find approximate solutions using Newton's method (for example)
  - $\log_2 5 \approx 2.32192809\ldots$
- What if we want to find $\log_b a$ where a and b are integers modulo m?
  - That is, find an integer c such that $b^c \equiv a \mod m$.
  - Example: in $\mathbb{Z}/31$, $\log_3 10 = 14$ because $3^{14} \equiv 10 \mod 31$.
- Is there any way to do it faster than just trying every possible exponent until we find one that works?
- This is called the discrete logarithm problem. It's computationally hard, like finding the prime factors of a big number.

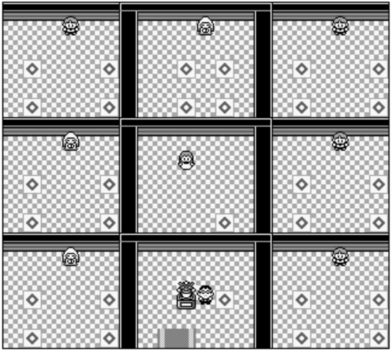
Each number in $\mathbb{Z}/31$ appears as 3^c for some c. But there's no easy way to tell when a particular value will appear.

c	$3^c \mod 31$
0	1
1	3
2	9
3	27
4	19
5	26
6	16
7	17
8	20
9	29
10	25

c	$3^c \mod 31$
11	13
12	8
13	24
14	10
15	30
16	28
17	22
18	4
19	12
20	5
21	15

c	$3^c \mod 31$
22	14
23	11
24	2
25	6
26	18
27	23
28	7
29	21
30	1
31	3

Kind of like a teleporter maze...



If you keep multiplying by b, eventually you'll hit every integer mod m:

$$b, b^2, b^3, b^4, \ldots, b^{m-2}, b^{m-1}$$

But you don't know in what order you'll see these numbers.

Diffie-Hellman key exchange

- Suppose Alice and Bob want to communicate using a symmetric cryptosystem like AES.
- In order to do this, they need to share a symmetric key without letting anyone else know it.
- Ideally they should be able to simultaneously create the key without sharing private information over an unsecured channel. This is called key exchange.
- Diffie-Hellman key exchange uses the difficulty of the discrete logarithm problem to keep the key safe from attackers.

DH key exchange algorithm:

- Alice and Bob choose a large prime number p and a special number g in $\mathbb{Z}/p$. These numbers will be shared publicly.
- Alice chooses a random integer a modulo p to be her private key. She calculates $A = g^a \mod p$, which is her public key.
- Bob chooses a random integer b modulo p to be his private key. He calculates $B = g^b \mod p$, which is his public key.
- Alice and Bob both publish their public keys so everyone can see them. They keep their private keys hidden.

Only Alice knows	Everyone knows	Only Bob knows
a	p, g, A, B	b

Only Alice knows	Everyone knows	Only Bob knows
a	$p,\ g,\ A,\ B$	b

Now it's time to create a shared secret symmetric key.

- Alice calculates $k = B^a \equiv (g^b)^a \equiv g^{ab} \mod m$
- Bob calculates $k = A^b \equiv (g^a)^b \equiv g^{ab} \mod m$
- Now Alice and Bob both know $k = g^{ab}$, which they can use as a shared secret key
- For a third person to compute k, he would have to find either a or b, which are the base-g logarithms of A and B modulo p.

A toy example

- $p = 29$, $g = 10$
- Alice chooses $a = 6$ for her private key. She calculates $A = 10^6 \equiv 22 \mod 29$ for her public key
- Bob chooses $b = 21$ for his private key. He calculates $B = 10^{21} \equiv 12 \mod 29$ for his public key
- Alice computes $k = B^a \equiv 12^6 \equiv 28 \mod 29$
- Bob computes $k = A^b \equiv 22^{21} \equiv 28 \mod 29$
- The shared secret key is $k = 28$.

Solving the discrete logarithm problem

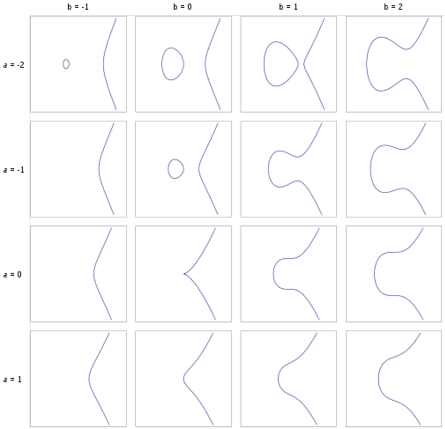
- In the real world, p is probably a 1024-bit number (about 308 digits!)
- Brute force attack: try all 2^{1024} possibilities.
  - would take many, many years even for a supercomputer
- But there are some clever algorithms which speed things up...
  - Pollard rho
  - Baby-step giant-step
  - Pollard Kangaroo
- These algorithms have running time $O(\sqrt{p})$. (Birthday paradox)
  - Better than brute force; equivalent to trying 2^{512} numbers instead of 2^{1024}.
  - Still slow

Discrete log algorithms: index calculus

- There's a much better algorithm to find logarithms in $\mathbb{Z}/m$ called index calculus.
  - Uses the fact that many integers modulo m are products of small primes. (smooth numbers)
  - Factor some of these integers, create a system of linear equations based on this
  - Solve the system, use the solution to find the logarithm
  - Much more complicated than Pollard rho, but faster
- The better the algorithms that attack a cryptosystem, the larger the key we need to remain secure.
  - Index calculus is such a strong attack that it would force us to use very big keys (large key space)
  - As an alternative, we can define a cryptosystem using a different type of discrete log that's not vulnerable to index calculus...

What is an elliptic curve?

An **elliptic curve** is a curve in $\mathbb{R}^2$ defined by an equation of the form $y^2 = x^3 + ax + b$ for some constants a and b.



A **group** is a set of things (like numbers), together with an operation (like addition), such that:

1. If you add any two elements of the group, the sum is an element of the group
2. The operation is associative, meaning $a + (b + c) = (a + b) + c$
   - Sometimes (but not always) the operation is commutative also: $a + b = b + a$
3. There is an **identity element** e. Any element plus the identity is itself
4. Every element has an inverse. If you add an element and its inverse, you get the identity: $a + -a = e$.

The integers as a group

One example of a group: the set of all integers $\mathbb{Z} = \{\dots, -2, -1, 0, 1, 2, \dots\}$ with addition as the operation.

- for any two integers a and b, $a + b$ is also an integer
- the identity is 0. For any integer a, $a + 0 = a$
- The inverse of a is $-a$. Clearly $a + (-a) = 0$.
- This is an infinite group
- But it's not very interesting

Some other examples:

- The integers modulo m under addition form a group
- The integers modulo m under multiplication form a group, if you take out “bad elements” (like 0) that don’t have inverses.
  - When m is prime, we only have to take out 0 because every other number has an inverse.
- Permutation groups: a set of all permutations (rearrangements) of a set of things, with function composition as the group operation
- Symmetry groups of geometric objects

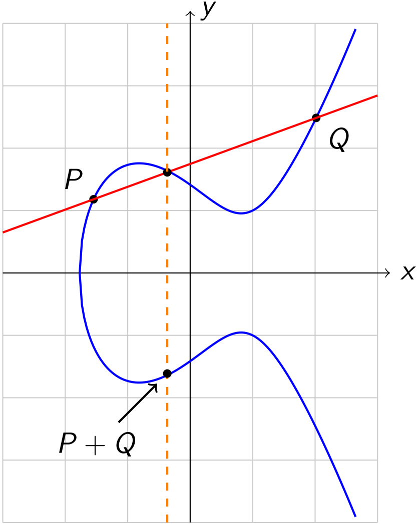
We can define a version of the discrete log problem in any finite cyclic group.

Group structure of an elliptic curve

We can create a group using the set of rational points on an elliptic curve, if we choose the appropriate group operation.

- Let G be the set of pairs of rational numbers (x, y) which satisfy $y^2 = x^3 + ax + b$.
- We want to define a group on this set. We need:
  1. an operation $+$ such that for any two elements P, Q in G, $P + Q$ is also in G.
     - must be associative, i.e. $P + (Q + R) = (P + Q) + R$
  2. an identity element I such that $P + I = P$ for all P in G
  3. an inverse $-P$ for each element P, such that $P + -P = I$.
- How do we do this?

- To add $P + Q$:
  - Draw the line $\overline{PQ}$
  - $\overline{PQ}$ intersects the curve at exactly 3 points*
  - Define $P + Q$ to be the reflection across the x-axis of the third intersection point (besides P and Q).
- Easy to prove the following:
  - $P + Q$ is always rational, so $P + Q$ is in G
  - $+$ is associative (and commutative)
- To add $P + P$, draw the tangent to the curve at P



Wait a minute...

Two questions:

1. What happens if you add two points with the same x coordinate?
   - $\overline{PQ}$ is a vertical line
   - Only intersects the curve at P and Q – there's no third point!
2. What is the identity element?
   - We need some point I such that for every point P on the curve, $P + I = P$ and $P + -P = I$.

To answer these questions, we need to add an extra imaginary point to the curve.

Let's add one more point to this group: ∞, the point at infinity.

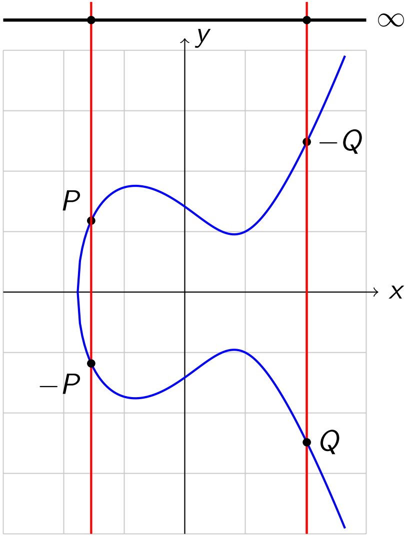
- Think of this as a magical point that exists infinitely far above (and/or below) the curve
- Think of a vertical line $\overline{PQ}$ as passing through three points: P, Q on the curve and ∞.
- ∞ is the identity element
  - To add $P + \infty$, draw a vertical line through P. The line $\overline{P\infty}$ intersects the curve directly above or below P, so $P + \infty = -(-P) = P$.
- The inverse of P is its reflection across the x-axis, $-P$.
  - The line $\overline{P(-P)}$ intersects the curve at $P, -P, and \infty$, so $P + -P = \infty$.

Now we have an actual group operation on the elliptic curve! The group of rational points on the curve E is called $E(\mathbb{Q})$.

Elliptic curve group operation: identity and inverse

The identity element is ∞

- $P + \infty = P$
- $P + -P = \infty$
- $Q + \infty = Q$
- $Q + -Q = \infty$



Elliptic curve group operation: formula

Let E be an elliptic curve with equation $y^2 = x^3 + ax + b$ and let $P(x_1, y_1)$ and $Q(x_2, y_2)$ be points on E with $x_1 \neq x_2$.

- If $P \neq Q$, let $s = \frac{y_2 - y_1}{x_2 - x_1}$
- If $P = Q$, let $s = \frac{3x_1^2 + a}{2y_1}$
- Let $x_3 = s^2 - x_1 - x_2$ and let $y_3 = y_1 - s(x_1 - x_3)$
- Then (x_3, y_3) is the third intersection point of E and $\overline{PQ}$
- Therefore $P + Q = (x_3, -y_3)$.

So you don't have to actually draw lines on a graph to add points. You can just use this formula.

For cryptography, we need to work in a finite set, not all the rational numbers

- Consider the integer pairs (x, y) with $0 \leq x, y < p$ which satisfy the equation $y^2 \equiv x^3 + ax + b \mod p$
- Example: $y^2 \equiv x^3 + x + 6 \mod 7$
  - $(4, 2)$ is a solution because $2^2 \equiv 4^3 + 4 + 6 \equiv 4 \mod 7$
- The group operation still works. (Use the formula from the previous slide)
- The group of points on E modulo p is called $E(\mathbb{F}_p)$.
  - This is a finite cyclic group
  - Hasse's theorem: there are (roughly) p points on the curve modulo p.

The Elliptic Curve Discrete Log Problem

- If you have a number n and a point P on the curve, it's easy to add P to itself n times and find the point nP
- But, if you have P and an arbitrary point Q, how do you find a number n such that P added to itself n times is Q?
  - If you keep adding P you'll eventually hit every point on the curve, but in an unpredictable order.
- This is the same thing as the discrete log problem, but in a different group: $E(\mathbb{F}_p)$ instead of $(\mathbb{Z}/p)^\times$.
  - Takes a long time to solve, even with computers
- ECDHE is a version of Diffie-Helman that uses the elliptic curve version of the discrete logarithm problem.

Elliptic Curve Diffie-Hellman

Alice and Bob want to securely generate a shared secret key

- They agree on an elliptic curve E, a prime p, and a point P on E. These things are all shared publicly.
- Alice chooses a random positive integer a to be her private key. She adds P to itself a times to get a point $A = aP$ on E. This is her public key.
- Bob chooses a random positive integer b to be his private key. He adds P to itself b times to get a point $B = bP$ on E. This is his public key.
- Alice and Bob publish their public keys, but keep their private keys secret.

Elliptic Curve Diffie-Hellman

- Alice adds B to itself a times, getting $k = a(bP) = (ab)P$.
- Bob adds A to itself b times, getting $k = b(aP) = (ab)P$.
- Now Alice and Bob both know $k = (ab)P$, which they can use as a shared secret key.
- For a third person to find k, they would have to compute a or b, i.e. the discrete log of A or B in $E(\mathbb{F}_p)$.

Why use ECC?

- Some discrete log algorithms work in any group, including $E(\mathbb{F}_p)$: Pollard rho, Kangaroo, etc
- But index calculus does NOT work!
  - the elliptic curve group is too 'strange'. Index calculus relies on using information about $\mathbb{Z}$ (prime factorization)
- So the best known attacks are of the Rho/Kangaroo/BSGS type, which are much slower
- Same security level with much smaller keys!

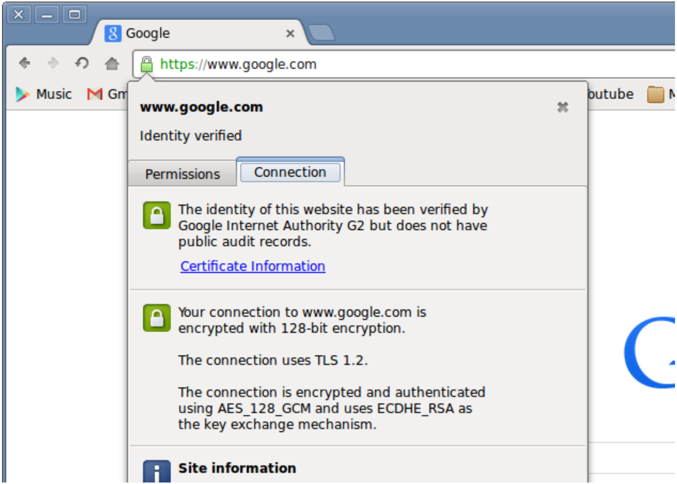
However. . .

- Like any cryptosystem, someone may discover a new, much better method of attack
- Particular elliptic curves may have hidden weaknesses
  - Some people don't trust NIST standards for this reason (NSA backdoor?)
  - Both number theorists and terrifyingly Orwellian government agencies are VERY interested in studying elliptic curves.
- Quantum computing / Shor's algorithm

Nevertheless, ECC has become an extremely popular public-key paradigm in the last 2 decades.

ECDHE in online communication protocols

TLS, HTTPS



ECDSA

ECDSA (digital signature algorithm) is used in:

Bitcoin (blockchain)

iMessage

PS3 (oops)

Android

Summary

- Public-key cryptography allows people to communicate securely without sharing the same key
- Public-key algorithms are based on hard (usually number-theoretic) mathematical problems
- The discrete logarithm problem is used in cryptosystems such as Diffie-Hellman
- Elliptic curve cryptography uses discrete logarithms in an elliptic curve group $E(\mathbb{F}_p)$ to provide even better security

Thanks!

Further reading:

- *The Code Book* by Simon Singh
  - Non-technical history of cryptography from antiquity to the present day
- *Understanding Cryptography* by C. Paar, J. Pelzl
  - Excellent textbook on modern crypto algorithms. Written for engineers, explains all the required math.
- Bruce Schneier's blog: `schneier.com`