Innovation and Cryptoventures

Cryptography 101

Campbell R. Harvey

Duke University, NBER and

Investment Strategy Advisor, Man Group, plc

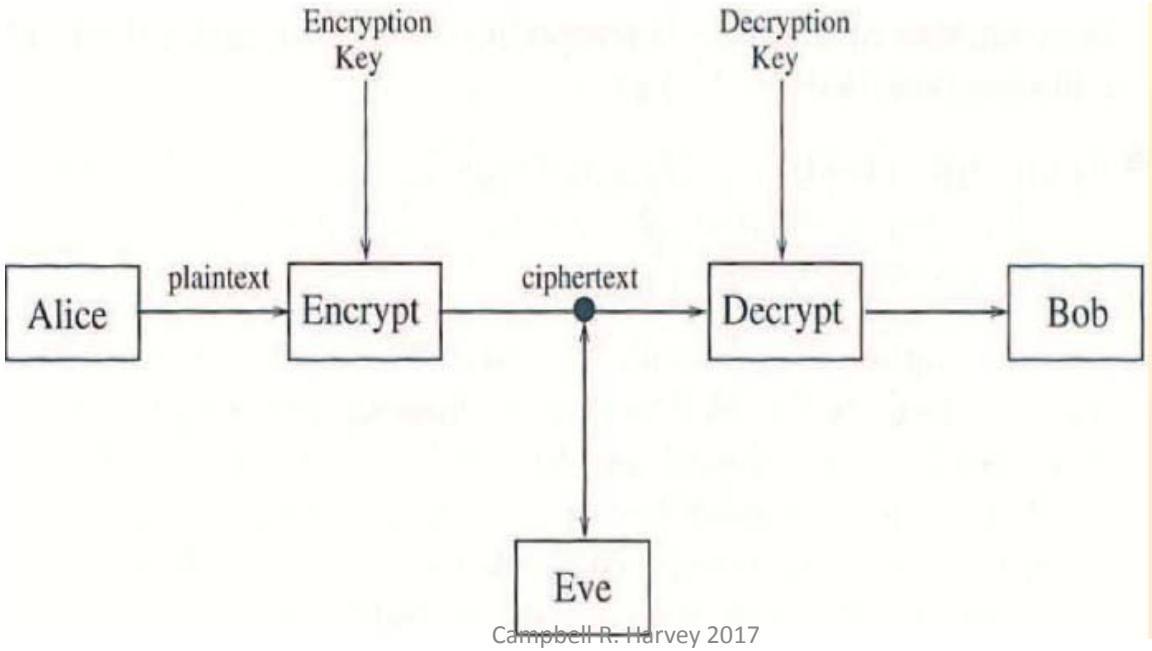
Revised February 6, 2017

Definition

Cryptography is the science of communication in the presence of an adversary. Part of the field of cryptology.

Goals of Adversary

- Alice sends message to Bob
- Eve is the adversary



Goals of Adversary

Eve's goals could be:

1. Eavesdrop
2. Steal secret key so that all future messages can be intercepted
3. Change Alice's message to Bob
4. Masquerade as Alice in communicating to Bob

Symmetric and Public Key

Early algorithms were based on symmetric keys.

- This meant a common key encrypted and decrypted the message
- You needed to share the common key and this proved difficult

Early methods

- Early methods relied on a shared key or code
- A message would be encrypted and sent but the receiver needed to decode with a key or a special machine
- Example: The “Lektor” in James Bond, *From Russia with Love*.

Early methods

- However, you needed to securely share the key or decoder.

Early methods

- However, you needed to securely share the key or decoder.

The “adversary”

Early methods

- Nazi Enigma Machine

https://www.youtube.com/watch?v=G2_Q9FoD-oQ
https://www.youtube.com/watch?v=V4V2bpZlqx8

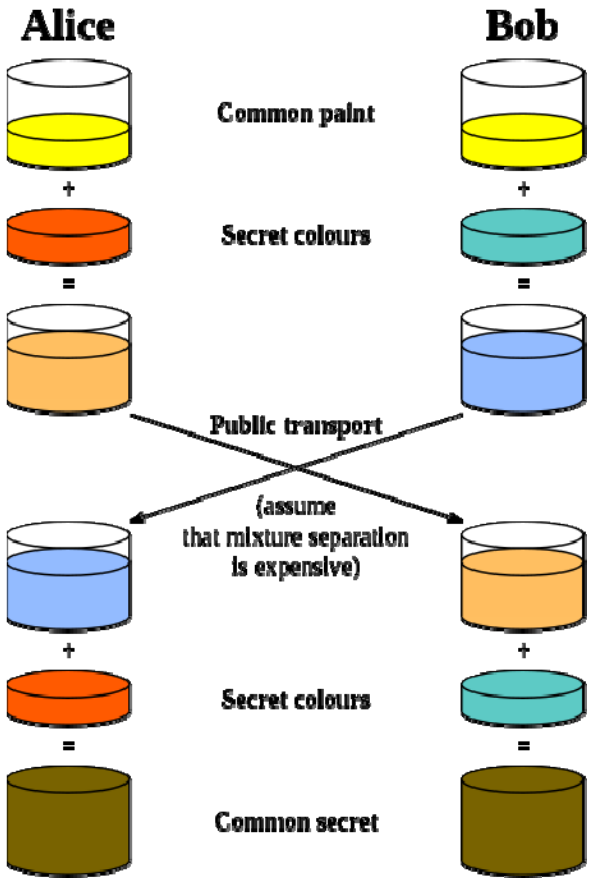
Key Exchange

- Breakthrough in 1976 with Diffie-Hellman-Merkle key exchange
- There is public information that everyone can see. Each person, say Alice and Bob, have secret information.
- The public and secret information is combined in a way to reveal a single secret key that only they know

https://www.youtube.com/watch?v=YEBfamv-_do

Key Exchange

- Paint example:



Key Exchange

- Numerical example
  - Need to know some Modulo arithmetic
  - “5 mod 2” = 1
  - Divide 5 by 2 the maximum number of times(2)
  - 2 is the modulus
  - The remainder is 1
  - Remainders never larger than (mod-1) so for mod 12 (clock) you would never see remainders greater than 11.

Key Exchange

- For our application, we will consider special case. Let p=prime number and let g be the generator (or primitive root)
  - g is a primitive root of p if we consider n=1,…(p-1) and g^n mod p goes through all the numbers, 1 to (p-1) in some order.
  - Going through all the numbers 1-(p-1) means each one is equally probable
  - Better with example!

Key Exchange

- Example: 3 is a primitive root of 5

p:	5	
g:	3	
n	**g^n**	**g^n mod p**
1	3	3
2	9	4
3	27	2
4	81	1

n goes from 1 to (p-1)=4

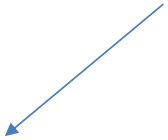
All numbers <p in no particular order

Key Exchange

- Example: 3 is a primitive root of 5

p:	5	
g:	3	
n	g^n	g^n mod p
1	3	3
2	9	4
3	27	2
4	81	1
5	243	3
6	729	4
7	2187	2
8	6561	1
9	19683	3
10	59049	4
11	177147	2
12	531441	1
13	1594323	3
14	4782969	4
15	14348907	2
16	43046721	1

All numbers <p and they repeat

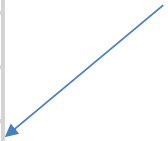


Key Exchange

- Example: 4 is not a primitive root of 5

p:	5	
g:	4	
n	g^n	g^n mod p
1	4	4
2	16	1
3	64	4
4	256	1

2, 3 are missing



Key Exchange

- Aside (can't resist): Fermat's Little Theorem:
  - $g^{p-1} = 1 \bmod p$ if p=prime and g is not zero
  - Our prev. example: $3^{5-1} = 1 \bmod 5$

p:	5	
g:	3	
n	**g^n**	**g^n mod p**
1	3	3
2	9	4
3	27	2
4	**81**	**1**

Fermat

Key Exchange

- Aside (can't resist): Fermat's Little Theorem:
  - $g^{p-1} = 1 \bmod p$ if p=prime and g is not zero
  - Our prev. example: $3^4 = 1 \bmod 5$

p:	5	
g:	3	
n	**g^n**	**g^n mod p**
1	3	3
2	9	4
3	27	2
4	**81**	**1**

Fermat

p:	17	
g:	5	
n	**g^n**	**g^n mod p**
1	5	5
2	25	8
3	125	6
4	**625**	**13**
5	3125	14
6	15625	2
7	78125	10
8	390625	16
9	1953125	12
10	9765625	9
11	48828125	11
12	244140625	4
13	1220703125	3
14	6103515625	15
15	30517578125	7
16	1.52588E+11	1

https://www.youtube.com/watch?v=XPMzosLWGHo&feature=youtu.be

Key Exchange

- Aside 2 (can't resist 2): Fermat's Last Theorem:

  No three positive integers, a, b, c can solve

  $$a^n + b^n = c^n$$

  for any n>2

  – Conjectured in 1637

  – Proven by Andrew Wiles 358 years later in 1994! The proof uses Elliptic Curve math – which we will talk about later.

See Numberphile video https://www.youtube.com/watch?v=qiNcEguuFSA
Also great video on Riemann Hypothesis: https://www.youtube.com/watch?v=qiNcEguuFSA
Also the Poincare conjecture https://www.youtube.com/watch?v=GItmC9lxeco

Key Exchange

- Mechanics
  - Public information in blue
  - Alice secret information in green
  - Bob secret information in purple
  - Shared secret in red

1. Alice and Bob agree to use prime number p=23 and base (generator) g=5.

Key Exchange

p:	23	
g:	5	
n	**g^n**	**g^n mod p**
1	5	5
2	25	2
3	125	10
4	**625**	**4**
5	3125	20
6	15625	8
7	78125	17
8	390625	16
9	1953125	11
10	9765625	9
11	48828125	22
12	244140625	18
13	1220703125	21
14	6103515625	13

2. Alice chooses a secret integer, $a=6$.
3. Alice sends Bob

$$A=g^a \bmod p \quad \text{or} \quad 5^6 \bmod 23 = 8$$

Note sending Bob the number 8 does not reveal Alice's secret number because 8 will repeat many times

Key Exchange

p:	23	
g:	5	
n	**g^n**	**g^n mod p**
1	5	5
2	25	2
3	125	10
4	**625**	**4**
5	3125	20
6	15625	8
7	78125	17
8	390625	16
9	1953125	11
10	9765625	9
11	48828125	22
12	244140625	18
13	1220703125	21
14	6103515625	13
15	30517578125	19

4. Bob chooses a secret integer, $b=15$.
5. Bob sends Alice

$$B=g^b \bmod p \quad \text{or} \quad 5^{15} \bmod 23 = 19$$

Note sending Alice the number 19 does not reveal Bob's secret number because 19 will repeat many times

Key Exchange

6. Alice calculates

$s=B^a \bmod p$ or $s=19^6 \bmod 23 = 2$

p:	23	
g:	19	
n	**g^n**	**g^n mod p**
1	19	19
2	361	16
3	6859	5
4	**130321**	**3**
5	2476099	11
6	47045881	2

Key Exchange

7. Bob calculates

$s = A^b \bmod 23$ or $s=8^{15} \bmod 23 = 2$

8. Alice and Bob now share a common secret even though neither know each others' individual secret numbers.
   - In practice, the prime number is picked to be 300 digits and the a and b are 100 digits long
   - The generator, g, is usually small

9. Now Alice can encrypt a message for Bob and Bob has the key to decrypt – and vice versa!

Key Exchange

The modulo arithmetic is the key.

- Think of starting 10 years ago and accumulating time in minutes (5,258,880 minutes).
- Suppose you took a picture of a clock at various times during the 10 years.

Key Exchange

The modulo arithmetic is the key.

- The clock says 1:50. However, you don't know AM/PM or which of the days

RSA

Very popular method of encryption called RSA*

- The basic algorithm multiplies two prime numbers
- Multiplication is easy
- However, given the product, it is extremely hard to factor the number
- Important in cryptography is have a large gap between the ease of encryption and the difficulty in decryption (called “trapdoor functions”)

Initials of Ron Rivest, Adi Shamir, and Leonard Adleman. Discovered earlier by the UK Communications-Electronics Security Group (CESG).

RSA

The insight is that encryption done with the public key can only be undone with the private key

- This is different from (but related to) the previous example where there was a common secret number
- Now each person will have a public and private key

https://www.youtube.com/watch?v=wXB-V_Keiu8

RSA

Computers have problems with very large numbers hence we will rely on modulo arithmetic.

- In RSA, you set a “max” number which is the product of two prime numbers – this is exactly the modulus (what you divide by, or in our clock example, it would be 12x60=720 minutes)
- My clock example is not a good one for encryption because the two numbers are secret that produce the max (and 12 and 60 are not prime)

RSA

The public and private keys are less than the max and greater than zero

- To encrypt, take a number and raise it to the power of the public key and apply mod “max”.
- To decrypt, take the resultant number and raise it to the power of the private key and apply mod “max”
- You get back the original number!
- Example!

RSA

Let's take 7 and 13 as prime numbers and the max will be 91.

- 7 and 13 are secret but 91 is public knowledge
- We will choose a public key of 5
- Given the public key and the 7 and 13, this implies a private key of 29. I will prove this but take my word for now.
- First, let's do a simple encryption.

RSA

Let's choose to encrypt the number "2" given our max is 91.

- Apply the public key to it:
- $2^5=32$
- 32 mod 91 = 32
- To decrypt, we will have to calculate

  32^{29} mod 91 which is the same as

  $(2^5)^{29}=2^{145}$ (remember exponent of exponent you multiply)
- This an enormous number: 10 followed by 41 zeros!

RSA

Need to apply some tools

- First, $xy \bmod 91 = (x \bmod 91)(y \bmod 91) \bmod 91$
- Second, $2^{145}=2^{128}2^{17}$ (multiplying exponents of same base means you add them up)
- Third, we need a way to calculate $2^{128} \bmod 91$

RSA

Note: using mod function in excel

		mod=	91
2^1	2	2	
2^2	4	4	
2^4	16	16	
2^8	256	74	
2^16	65536	16	
2^32	not needed	74	
2^64	not needed	16	
2^128	not needed	74	

Key is

$2^8 \bmod 91 = (2^4 \bmod 91)(2^4 \bmod 91) = 16 \times 16 \bmod 91 = 256 \bmod 91 = 74$

$2^{16} \bmod 91 = (2^8 \bmod 91)(2^8 \bmod 91) = 74 \times 74 \bmod 91 = 16$

So only need 8 calculations not 128!

RSA

Now we are almost done

$2^{145} \bmod 91 =$

$(2^{128} \bmod 91)(2^{16} \bmod 91)(2^{1} \bmod 91) =$

$74 \quad x \quad 16 \quad x \quad 2 \quad =$

$2368 \bmod 91 = 2$

- Hence, we have decrypted our secret number!

RSA

There are some missing links.

- First, it should be clear that there is a mathematical relation between the private key and the public key.
- Second, how do we set the private key (one that will decode) given the public key.
- My goal is to give you a basic understanding. Note that cryptography is a huge field where you could take dozens of courses.

RSA (optional)

For many of you, this will be a review. We will do simplified versions of some methods proposed by Euclid in 300BC.

- While my examples are simple examples, the method is algorithmic and easily coded up to make more complex examples doable.
- We will cover “greatest common denominator” and “The Euclidian Algorithm” as well as “The Extended Euclidian Algorithm”

RSA (optional)

Division Algorithm for Integers

- Let's start simple: 13 divided by 5
- We can decompose the division into two parts
- 13/5 = 2 ("the quotient") + 3/5 ("the remainder").
- We can express this in terms of integers, without reference to the division operation:
- 13 = 2(5) + 3.

RSA (optional)

Division Algorithm for Integers

More formally stated:

- If a and b are positive integers, there exist integers unique non-negative integers q and r so that

$$a = qb + r \text{ , where } 0 \leq r < b.$$

q=quotient

r=remainder

RSA (optional)

Greatest common denominator

- This one I know every one has done before
- If I give you two numbers, say, 5 and 15, and ask what the greatest common denominator is, most of you will say “5”
- 5 is the largest number that divides evenly into both of these numbers
- 1 is another candidate but it is not the “greatest”
- So gcd(5,15)=5

RSA (optional)

Euclidian Algorithm

- Finding the gcd of 81 and 57 by the Euclidean Algorithm:
- 81 = 1(57) + 24 (divide larger by smaller)
  57 = 2(24) + 9 (the "r" in above now "b")
  24 = 2(9) + 6
  9 = 1(6) + **3** **(the gcd is the "r" of second**
  6 = 2(3) + 0. **last line)**
- Hence, gcd(81,57)=3 (also an Excel function!)

RSA (optional)

Euclidian Algorithm

- It is well known that if the

$$\gcd(a, b) = r$$

then there exist integers p and s so that:

$$p(a) + s(b) = r$$

Relevance check

- We will be using above equation to link the private and public keys
- In our case, the gcd of two prime numbers = 1 so r=1

RSA (optional)

Extended Euclidian Algorithm

- Let's go back to 81 and 57

$$p(a) + s(b) = r$$

$$p(57) + s(81) = 3$$

Let's solve for the p and the s

Again, it is a bit awkward but note the algorithmic nature of the solution

RSA (optional)

81 = 1(57) + 24
57 = 2(24) + 9
24 = 2(9) + 6
9 = 1(6) + **3**

or rearranging

24 = 81 - 1(57)
9 = 57 - 2(24)
6 = 24 - 2(9)
3 = 9 - 1(6)

3 = 9 -1(6) (start with last line)

3 = 9 - 1(24 -2(9)) (sub for 6 2nd last line)

3 = 3(9) -1(24) (rearrange above)

3 = 3(57 - 2(24)) -1(24) (sub for 9 3rd last)

3 = 3(57) - 7(24) (rearranged)

3 = 3(57) - 7(81 - 1(57)) (sub for 24 4th last)

3 = 10(57) - 7(81) (rearranged)

We are done: p=10; s=-7

RSA (optional)

Deriving the private key

- We know pq=N; p=7; q=13; N=91; public key=5. How do we get the private key 29?

RSA (optional)

RSA:

- Select prime numbers p and q where p≠q: 7, 13
- N="max" = pq; N will be modulus: 91
- Introduce a simple function:

$$\phi(n)=(p-1)(q-1)$$

$$72=6x12$$

- Choose public encryption exponent, e, such that

$$\gcd(e, (p-1)(q-1))=1 \quad \text{[in our case 5]}$$

RSA (optional)

RSA:

- Verify gcd(e, (p-1)(q-1))=1 with Extended Euclidian Algorithm – and also determine the decryption key “d”

RSA (optional)

RSA:

- Use Euclidian Algorithm to show gcd(72,5)=1

72 = 14(5) + 2

5 = 2(2) + 1 ←gcd

2 = 2(1) + 0

RSA (optional)

72 = 14(5) + 2

5 = 2(2) + 1

or rearranging

2 = 72 - 14(5)

1 = 5 - 2(2)

1 = 5 -2(2) (start with last line)

1 = 5 - 2(72 -14(5)) (sub for 2)

1 = 29(5) -2(72) (rearrange above)

We are done: d=29.

RSA (optional)

RSA Review

1. Bob chooses secret numbers p, q and sets “max” or N=pq (p≠q). N is public.
2. Bob sets a public key, e, such that

   $$\gcd(e, (p-1)(q-1))=1$$

3. Along the way, Bob also calculates his private key, d, using Extended Euclidian Algorithm. Note public: (e, N); Private (p,q,d)

RSA (optional)

RSA Review

4. Alice wants to send a secret message m to Bob.
5. Alice encrypts with Bob's public key and sends

$$c = m^e \bmod N$$

6. Bob receives the message c (known as a ciphertext) and decrypts with his private key, d

$$m = c^d \bmod N$$

RSA

Example

- Max=91; public key 5, private key 19
- Encrypt the message “CAM”
- Start with UTF-8 (**U**niversal Character Set **T**ransformation **F**ormat – **8** bit)

A	B	C	D	E	F	G	H	I	J	K	L	M
65	66	67	68	69	70	71	72	73	74	75	76	77
N	O	P	Q	R	S	T	U	V	W	X	Y	Z
78	79	80	81	82	83	84	85	86	87	88	89	90

Hence, CAM is 67, 65, 77.

RSA

Example

- We need to encrypt each number, 67, 65, 77
- Raise each number to power of public key (5) and apply mod 91.

RSA

Example

Max=	**91**	
Number=	**67**	
		mod
67^1	67	67
67^2	4489	30
67^4=67^2 x 67^2	900	81
67^5=67^4 x 67^1	5427	58

Max=	**91**	
Number=	**65**	
		mod
65^1	65	65
65^2	4225	39
65^4=65^2 x 65^2	1521	65
65^5=65^4 x 65^1	4225	39

Max=	**91**	
Number=	**77**	
		mod
77^1	77	77
77^2	5929	14
77^4=77^2 x 77^2	196	14
77^5=77^4 x 77^1	1078	77

Hence, encryption is 58, 39, 77

RSA

Example

- To decrypt, we need to take each of the numbers, 58, 37, 77 and raise to the power of the private key (29) mod 91

RSA

Example

Max=	**91**	
Number=	**58**	
		mod
58^1	58	58
58^2	3364	88
58^4=58^2 x 58^2	7744	9
58^8=58^4 x 58^4	81	81
58^16=58^8 x 58^8	6561	9
58^29=58^16 x 58^8 x 58^4 x 58^1	380538	67

Original letter “C”

Works with other numbers: 37 is decrypted to 75 and 77 decrypted to 77

RSA

Issues with RSA

- RSA relies on factoring
- N is public
- If you can guess the factors, p,q, then you can discover the private key

RSA

Issues with RSA

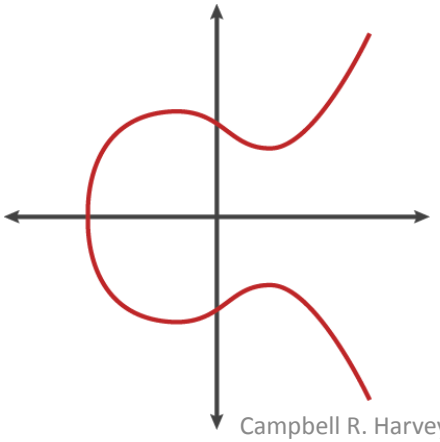
- Factoring algorithms have become very efficient
- To make things worse, the algorithms become more efficient as the size of the N increases
- Hence, larger and larger numbers are needed for N
- This creates issues for mobile and low power devices that lack the computational power

Elliptic Curve Encryption

Mathematics of elliptic curves

- Do not rely on factoring
- Curve takes the form of

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

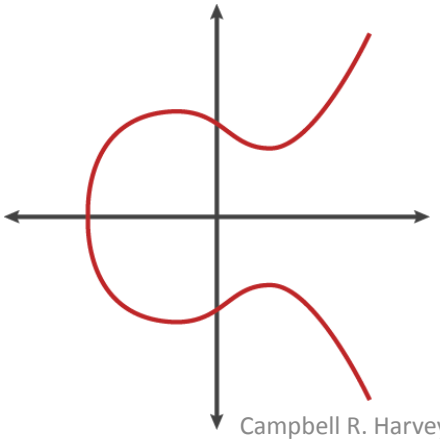


Note: $4a^3 + 27b^2 \neq 0$

Elliptic Curve Encryption

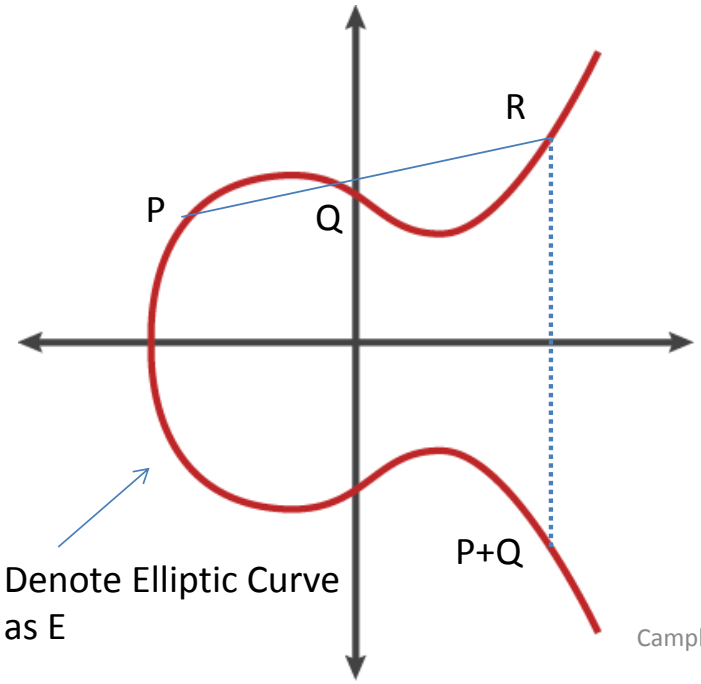
Properties

- Symmetric in x-axis
- Any non-vertical line intersects in three points
- Algebraic representation



Elliptic Curve Encryption

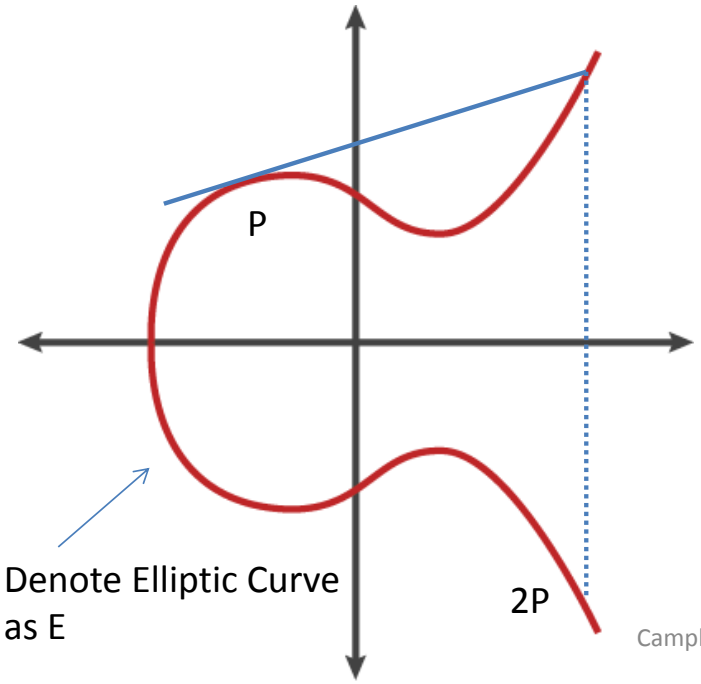
Properties



Define a system of "addition". To add "P" and "Q" pass a line through and intersect at third point "R". Drop a vertical line down to symmetric part. This defines P+Q (usually denoted $P \oplus Q$)

Elliptic Curve Encryption

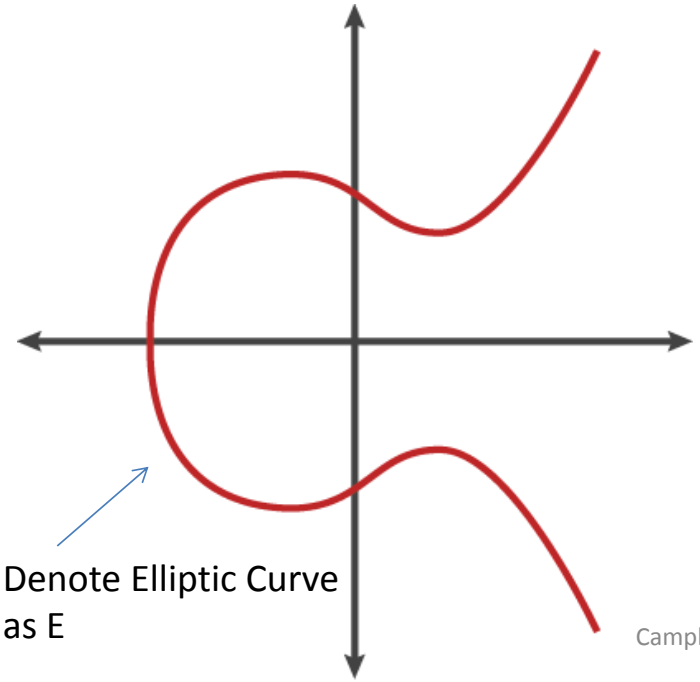
Properties



Define a system of "addition". To add "P" and "P" use a tangent line and intersect at third point. Drop a vertical line down to symmetric part. This definite 2P (usually denoted $P \oplus P$)

Elliptic Curve Encryption

Properties



Denote Elliptic Curve as E

(a) $P + O = O + P = P$ for all $P \in E$.
(existence of identity)
(b) $P + (-P) = O$ for all $P \in E$.
(existence of inverse)
(c) $P + (Q + R) = (P + Q) + R$ for all $P, Q, R \in E$.
(associative)
(d) $P + Q = Q + P$ for all $P, Q \in E$
(communativity)

Elliptic Curve Encryption

Why use in cryptography?

- Suggested by Koblitz and Miller in 1985
- Implemented in 2005
- Key insight:
  - Adding on the Elliptic curve is easy but undoing the adding is very difficult

Elliptic Curve Encryption

Discrete Logarithm Problem

- Let G represent an additive group and x,y are part of G and x=ny
- Discrete Logarithm Problem is to solve for n
  - Example: suppose y=12 and x=4185072, it is easy to solve for n=348756, i.e. by division n=y/x= 4185072/12

Elliptic Curve Encryption

Discrete Logarithm Problem

- Why call it logarithm problem?
- Another representation is $x=y^n$
- To solve for n, we could do it the hard way (so called square and multiply), i.e. keep on squaring y until it exceeds x, then multiply by y until you hit x -- similar to when we had to figure out 2^{128}

Elliptic Curve Encryption

Discrete Logarithm Problem

- Square and multiply might seem trivial but it is important
- Suppose we have 131072 = 2^n

Elliptic Curve Encryption

Discrete Logarithm Problem

- Suppose we have $131072 = 2^n$

2^0	1
2^1	2
2^2	4
2^4 = 2^2 x 2^2	16
2^8 = 2^4 x 2^4	256
2^16 = 2^8 x 2^8	65536
2^32 = 2^16 x 2^16	4294967296
2^16 x 2^1	131072

Went too far, go back 1 and multiply by 2

Note we don't have 17 operations to get to the solution.

Elliptic Curve Encryption

Discrete Logarithm Problem

- Why call it logarithm problem?
- Another representation is $x=y^n$
- To solve for n, we could do it the hard way (so called square and multiply), i.e. keep on squaring y until it exceeds x, then multiply by y until you hit x
- Or we could log both sides and solve directly:

  $\log x = \log y^n = n \log y$

- We solve for n by <u>division</u>, i.e. $n=\log x/\log y$

Elliptic Curve Encryption

Discrete Logarithm Problem

- Why am I underlining division?
- On the elliptic curve, it is easy to add – but there is no division!
- The best we can do is add y over and over.
- Hence, elliptic curve provides an ideal setting for encryption.

References

- Don Johnson and Alfred Menezes, The Ellipical Curve Digital Signature Algorithm (ECDSA), http://cs.ucsb.edu/~koc/ccs130h/notes/ecdsa-cert.pdf
- Diffie-Hellman key exchange, http://en.wikipedia.org/wiki/Diffie%E2%80%93Hellman_key_exchange
- Primitive Roots mod p, http://homepages.math.uic.edu/~leon/mcs425-s08/handouts/PrimitiveElements.pdf
- Privitive roots and Diffie-Hellman key exchange, http://homepage.smc.edu/morgan_david/vpn/assignments/assgt-primitive-roots.htm

References

- Diffie Hellman, Successive Square, Modulo Arithmetic Explained, http://www.eecs.harvard.edu/cscie2a/Modular%20Arithmetic%20Explained.pdf
- Nick Sullivan, A (relatively easy to understand) primer on elliptic curve cryptography, http://arstechnica.com/security/2013/10/a-relatively-easy-to-understand-primer-on-elliptic-curve-cryptography/
- The Extended Euclidean Algorithm, http://www-math.ucdenver.edu/~wcherowi/courses/m5410/exeucalg.html
- Calculating RSA private exponent when given public exponent and the modulus factors using extended euclid, http://crypto.stackexchange.com/questions/5889/calculating-rsa-private-exponent-when-given-public-exponent-and-the-modulus-fact

References

- Kalyan Chakraborty, An Introduction to Basic Cryptography, http://www.hri.res.in/~kalyan/lecture1.pdf
- Joseph Silverman, An Introduction to the Theory of Elliptic Curves, http://www.math.brown.edu/~jhs/Presentations/WyomingEllipticCurve.pdf
- Jeremy Kun, http://jeremykun.com/2014/02/10/elliptic-curves-as-elementary-equations/
- http://jeremykun.com/2014/02/16/elliptic-curves-as-algebraic-structures/
- Goldwasser, Shaffi and Mihir Bellare, 2008, Lecture Notes on Cryptography
- Dan Boneh, Stanford University, Introduction to Cryptography.

  Dan Boneh, Stanford University, Cryptography II
- ECCHacks – A gentle introduction to elliptic-curve cryptography https://www.youtube.com/watch?v=l6jTFxQaUJA