HKU Mathematics

Rambling in Maths

The Mathematical Key to unlocking the mysteries of Cryptography

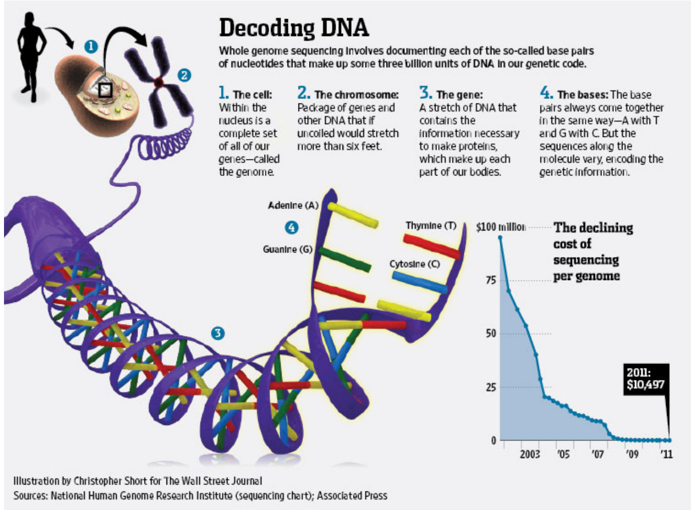
Dr. Ben Kane

24 March, 2018

A Problem

- You want to do a **BIG** calculation, e.g. with **(LOTS of) DNA data** (數據)

- Looking for patterns
  (can **save lives**!)



- Your computer is slow…

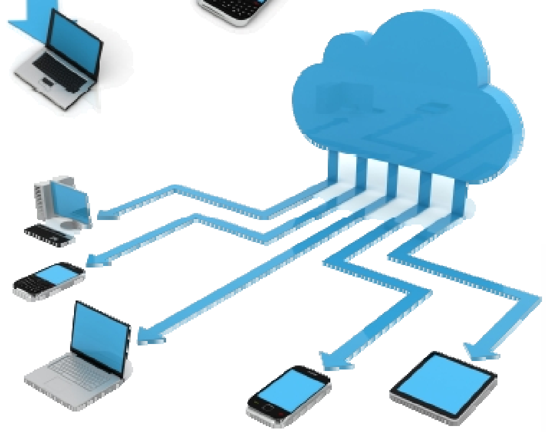
- Share the work?

- Problem: Is it **ethical** to send?

Cloud Computing: Basic idea

- Send the data to the "cloud"

- Cloud does some calculations



- Answers come back; combine answers centrally

A Concern

- Privacy issues (can you trust others?)
- Proposal: Can we find a way to have them do the calculations, but **never see** the data?
- We could “**mess**” with the data
- Can they still do calculations?

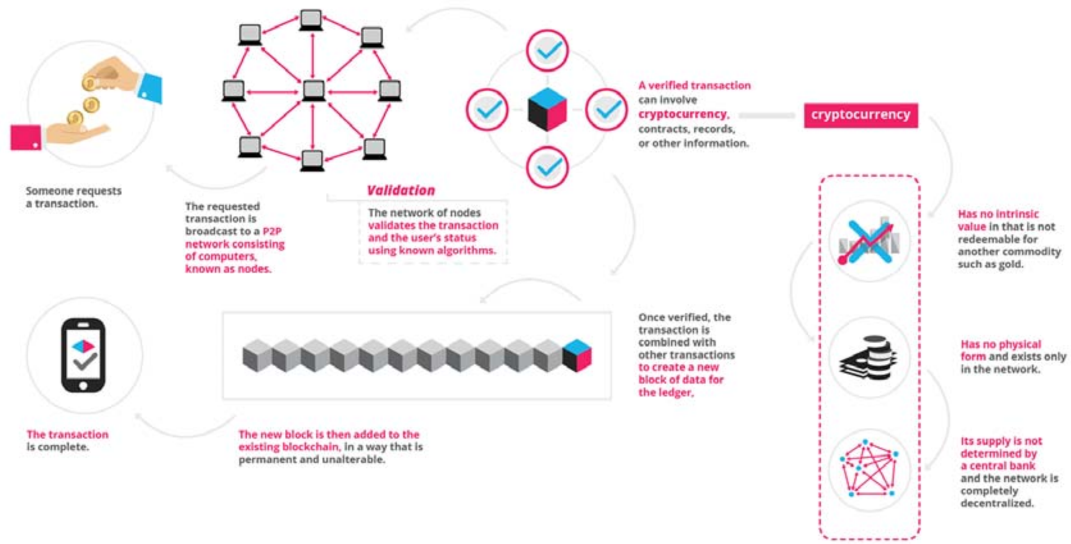
Another Problem

- An authority (bank) keeps track of/protects money.

- Online cryptocurrency (e.g. Bitcoin): Shared protection, decentralized.

Bitcoin and Blockchain

- Transaction made
- Collectively agree
- Added to “chain”
- Cannot be reverted



Credit: Amir Rosic, Blockgeeks.com

- Important that it is secure (cryptography)

Encryption

- Problems:
  1. Can we send data **safely**? (我們可以安全發送數據馬?)
     a) Is someone listening?
     b) Is the receiver trusted?
  2. If the data is safely sent …
     a) Can they do **calculations**? Is it accurate? (計算可以做嗎?)
     b) Can they figure out/guess the original data?
  3. When the data **comes back**… (當數據發送回來…)
     a) Can we get back the answer and/or original info?
     b) Is it the same as if we did it ourselves? Is it really faster?

Fundamental Rule of Encryption

- Can we scramble it?

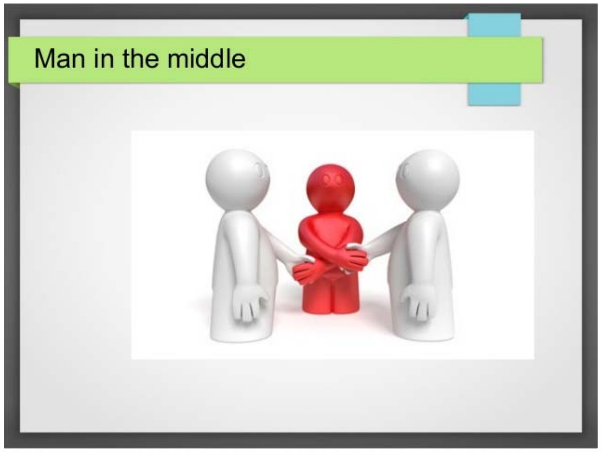
- Idea of (modern) **Encryption** (加密的想法):
- Find problem
  - That is hard to solve,
  - easy to check.

- We’ll see some examples later.

Back to the beginning…

- Goal: Send an important message

- Problem: Trust your courier? Man in the middle?



A long time ago…

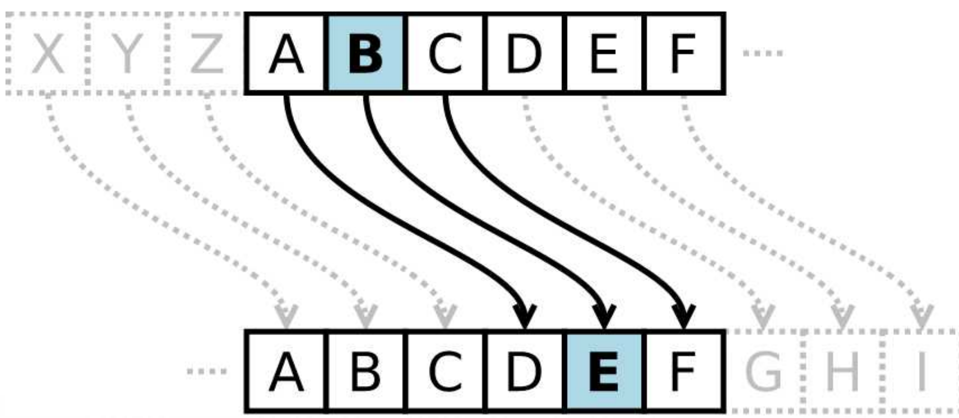
Caesar's solution

- Caesar Cipher: Every letter gets a number

A	B	C	D	E	F	G	H	I	J	K	L	M
1	2	3	4	5	6	7	8	9	10	11	12	13

N	O	P	Q	R	S	T	U	V	W	X	Y	Z
14	15	16	17	18	19	20	21	22	23	24	25	26

- Shift with modular (clock) arithmetic:

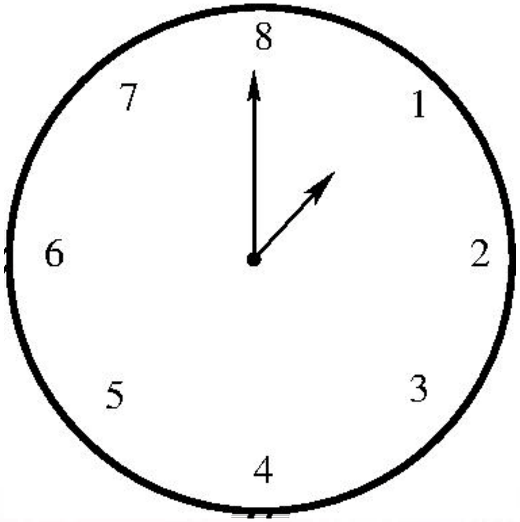


Modular arithmetic

- On a clock, the times 3am and 3pm agree.
- Some call it 03:00 and 15:00.
- So we “pretend” that 3 and 15 are the same.
- Our day is a 24 hour repeating clock.
- Repeating after 26?

Modular arithmetic

- Example:
  - Let's consider a clock with 8 hours.
  - If you start a class at 7 o'clock and it runs for 2 hours, what time will it say on the clock at the end?
    Try yourself…
  - 7+2=9, so …
  - It says **1 o'clock**.
  - We write $7 + 2 \equiv 1\ (mod\ 8)$.



Modular arithmetic

- Another example:
- Every non-leap year has 365 days.
- If your birthday was on Wednesday last year, what day will it be this year (not a leap year)?
- Day of week repeats every 7 days ("7-day clock")
- $364 = 52 \times 7$ ("clock rotates" 52 times)
- Remainder 1, so …
- Thursday

Some exercises

1. Find $0 \le x < 2$ if $12 \equiv x \pmod{2}$.

2. Find $0 \le x < 11$ if $46 \equiv x \pmod{11}$.

3. Find $0 \le x < 7$ if $82 \equiv 2x \pmod{7}$.

Caesar's solution

- Example: Shift by 10 (spaces removed?):
  - Secret message: attack at midnight
  - Number code: 1(20)(20)13(11) 1(20) (13)93(14)978(20)
  - Caesar sends: kddkmu kd wsnxsqrd
- Unscrambled at other end:

Caesar's solution

- Example: Shift by 20
  - Secret message: Hi
  - Number code: 89
  - Add 20: (28)(29)
  - You send (try yourself): … ("Clock" has 26 hours)
  - You send: bc

Exercise

4. Break Caesar's code to find the secret message! The encrypted message is "clxmwtyr ty xlesd".

Luckily, as a loyal attendee, you get the secret decoder information! Thanks for coming!

The shift is 11.

Problem

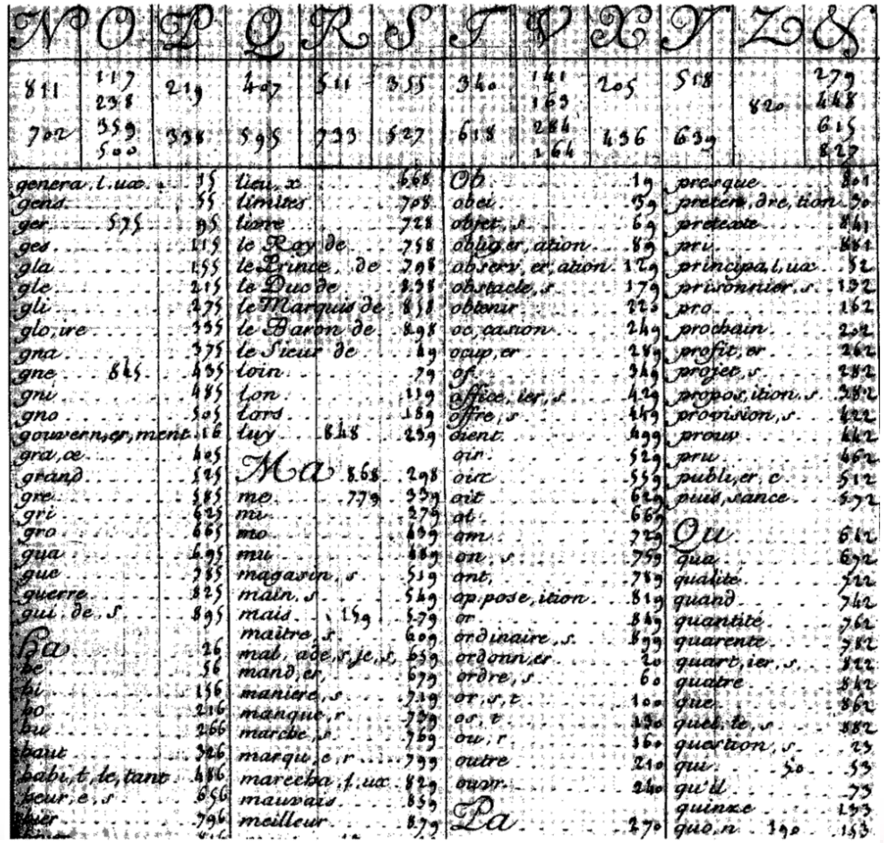
- What if someone figures out the code?
  - Someone steals: lddlnu ld wsnxsqrd
  - Simple to reverse.
  - Look for patterns/ make guesses:
    (many ‘d’ in code, ‘t’ and ‘e’ common in English)
- Other options: Instead of shift, maybe just replace?
  - Still many letters are common. Hmm…
  - Is there a better code?

Another try

- Maybe try randomly sending {a,…,z}to{a,…,z}?
- Spartan army:

- Common letters still a problem.

A whole new world / alphabet



N	O	P	Q	R	S	T	V	X	Y	Z	&
811	117 238	219	407	511	355	340	141 163	205	518	820	279 448
702	359 500	338	595	733	527	618	284 164	436	639		615 827

genera,l,ux	35	lieu,x	668	Ob	19	presque	801
gens	55	limites	708	obei	39	preten,dre,tion	30
ger 575	95	livre	728	objet,s	69	pretexte	841
ges	115	le Roy de	758	oblig,er,ation	89	pri	881
gla	155	le Prince, de	798	observ,er,ation	129	principa,l,ux	52
gle	215	le Duc de	838	obstacle,s	179	prisonnier,s	132
gli	275	le Marquis de	858	obtenir	220	pro	162
glo,ire	335	le Baron de	898	oc,casion	249	prochain	202
gna	375	le Sieur de	49	ocup,er	289	profit,er	262
gne 845	435	loin	79	of	349	projet,s	282
gni	485	lon	119	office,ier,s	429	propos,ition,s	382
gno	505	lors	189	offre,s	449	provision,s	422
gouvern,er,ment	16	luy 848	239	oient	499	prouv	442
gra,ce	405			oir	529	pru	462
grand	525	Ma 868	298	oire	559	publi,er,c	512
gre	585	me 779	339	oit	629	puis,sance	572
gri	625	mi	279	ot	669		
gro	665	mo	439	om	729	Qu	612
gua	695	mu	489	on,s	759	qua	692
gue	785	magasin,s	519	ont,	789	qualité	722
guerre	825	main,s	549	op,pose,ition	819	quand	742
gui,de,s	895	mais 159	579	or	849	quantité	762
		maitre,s	609	ordinaire,s	899	quarente	782
Ha	26	mal,ade,s,ie,s	659	ordonn,er	20	quart,ier,s	822
he	56	mand,er	679	ordre,s	60	quatre	842
hi	156	maniere,s	719	or,s,t	100	que	862
ho	216	manque,r	739	os,t	130	quel,le,s	882
hu	266	marche,s	769	ou,r	160	question,s	23
haut	326	marqu,e,r	799	outre	210	qui 50	53
habi,t,te,tant	486	marecha,l,ux	829	ouvr	240	qu'il	73
heur,e,s	656	mauvais	859			quinze	133
hier	796	meilleur	879	Pa	270	quo,n 190	153

Non-unique replacement

- Does ‘a’ **always** need to go to ‘t’?
- Maybe try randomly sending {a,…,z} to {a,…,z}?
- Replace ‘a’ with multiple choices
  - ‘a’ → ‘b’, ‘c’, ‘d’
  - ‘b’ → ‘b’, ‘e’, ‘f’
- Example: Great Cipher/Grand Chiffre
  - Replace syllables with similar-sounding choices
  - Unbroken for long time

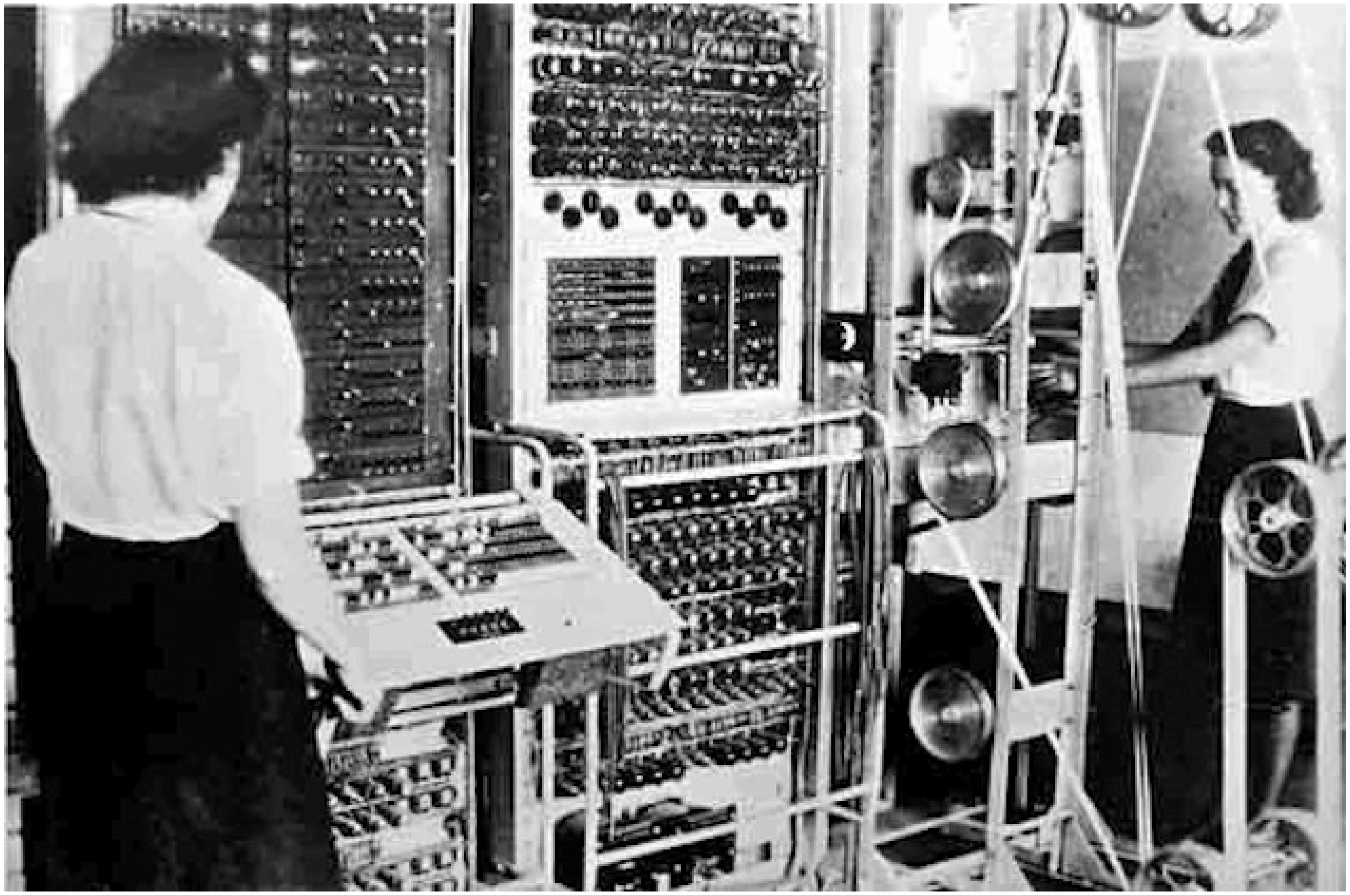
Non-unique replacement

- Why is it better?
  - A lot more choices
  - Frequency counting harder (‘e’ common in French)
- Problems?
  - Need to figure out how/when to switch
  - Indicator of switch might make it less secure
    - Example: Capital letters for language switch
    - Regular switching also can be detected

Polycipher

- Example: Take different shifts.
  - Shift 3, then 11, then 5, then 3, then 11, then 5, then …
- Input: “Here is a message”
  - Take out spaces/capitals: “hereisamessage”
  - Shift: ‘h’+3 = ‘k’, ‘e’+11=‘p’, ‘r’+5 =‘w’… :
- Output: “kpwhtxdxjvdfjp”
- Can still find patterns of repeated strings:
  - “the” appears a lot if you have many phrases.

A new challenge…



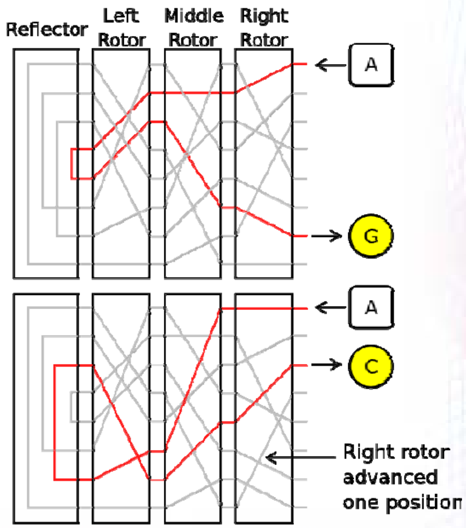
Non-unique replacement, large numbers

- Try to increase **number** of choices? (增加選擇)
- Example: Enigma machine

- **Rotates**, so ‘aaaa’ can become ‘rfgw’
  (轉子旋轉了→不同的字母出來了)

Enigma

- Plugboard: Fixed matching (‘a’ to ‘t’)
- First rotor (‘a’ to ‘x’), second rotor (‘x’ to ‘r’), third rotor (‘r’ to ‘d’).
- Reflector (‘d’ to ‘l’),
- Rotors (‘l’ to ‘z’ to ‘b’ to ‘m’)
- Output is ‘m’.
- Right rotator turns (add with carry)



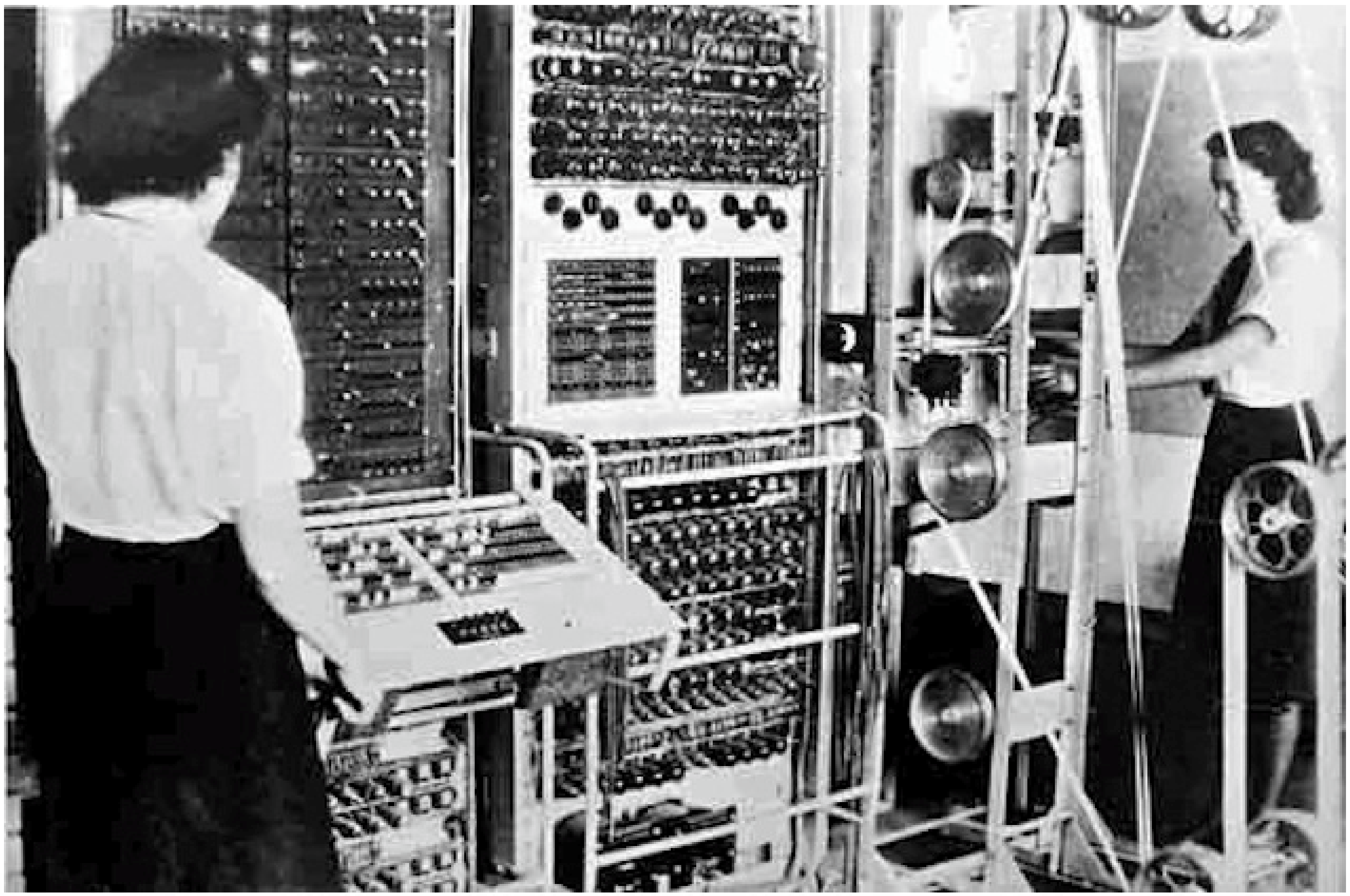
Enigma Decryption

- Process reverses itself:
  - If rotor sends ‘d’ to ‘x’, then it sends ‘x’ to ‘d’.
  - Gives extra symmetry (like a mirror)
- Key observation: If reflector sends ‘d’ to ‘d’, then message comes back unchanged.
  - Designed never to send ‘a’ to ‘a’
  - Reduces total number of possibilities (a lot)

Downfall of Enigma

- Number of permutations (reorderings) of 26 letters:
  $$26! \approx 4.0 \times 10^{26}$$
- Number of permutations of 26 letters, none repeated:
  - 25 choices for ‘a’, 24 choices for ‘b’, …, overall:
    $$25! \approx 1.5 \times 10^{25}$$
- Other symmetries help, too. (a matched to b is b matched to a, etc.)
- Common signals helped.

A big jump forward…



Computer age

- Computer search not expected (沒預料到電腦)
- Computers search differently
- Naïve search still often not enough
- Algorithm designs, search for weakenesses
- Beginnings/foundations of computer science
- Eventually split off of math departments
- Something new needed for cryptography

One-Way Functions

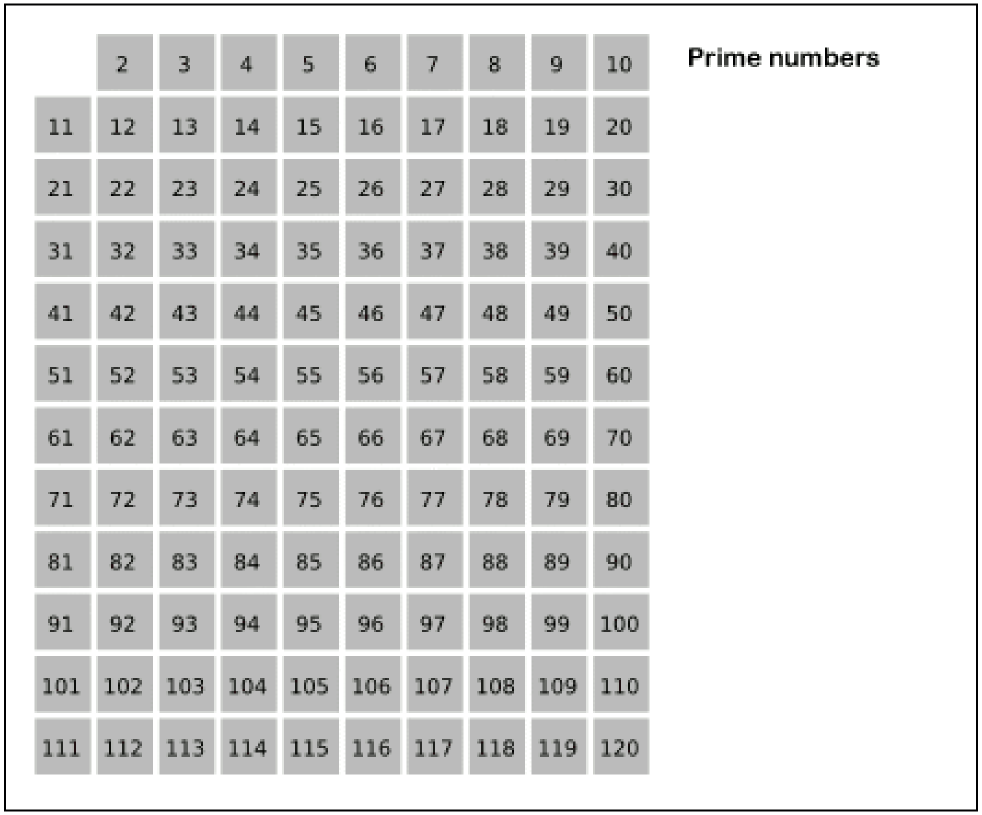
- Idea: What if there is an operation which is **easy to do**, but **hard** to **reverse**?做簡單, 顛倒計算很難
- Doing the operation is encryption, and reversing it is decryption.
- Authorized people know a secret that allows reversal.
- Example:
  - Multiply 3571 and 6997 to get 24986287.
  - Given 24986287, can you find 3571 and 6997?

Prime Factorization and RSA

- Question: Given an integer, can you find all of its **factors** (those integers which divide it equally)?
- **Prime numbers** are those whose only factors are 1 and themselves. (素數)
- Break a number up into primes (try to divide by 2, then 3, then 5, …): Prime factorization.
- RSA core idea: Multiplying primes is fast/easy, prime factoring is slow/hard.
- Can you even find all possible primes?

Prime numbers

- Sieve of Eratosthenes (Greece, c. 276BC – 195BC):



Prime Factorization

- Example: Consider 144
- Even: Divide by 2: now 2×72.
- Even: Divide by 2: now $2^2 \times 36$.
- Even: Divide by 2: …
- …
- Eventually get:

$$2^4 \times 9 = 2^4 \times 3^2.$$

Prime Factorization

- Example: Consider 481
- Not Even: … Try 3
- $481 \div 3$: Remainder 1. No. Try 5
- $481 \div 5$: Remainder 1. No. Try 7
- $481 \div 7$: Remainder 5. No. Try 11
- $481 \div 11$: Remainder 8. No. Try 13
- $481 \div 13 = 37$:

$$481 = 13 \times 37$$

Prime Factorization

- What about bigger numbers?
- Try 132523411?
- Not 2, not 3, not 5, not 7, …,
- …, 1039 divides!

$$132523411 = 1039 \times 127549$$

- Q: Is <u>127549</u> prime?

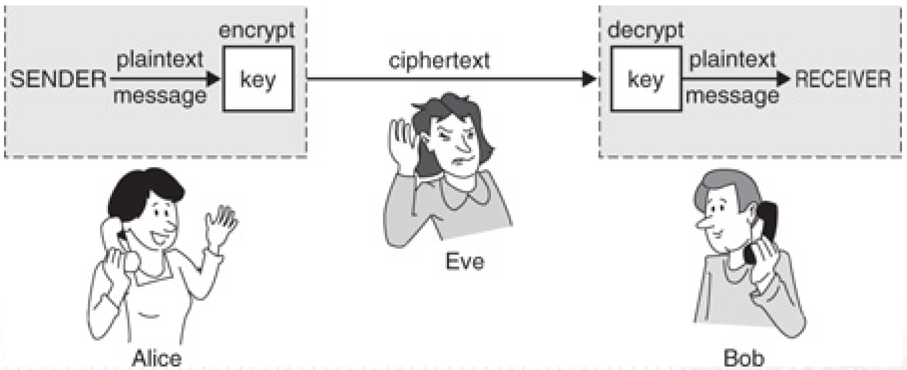
RSA

- Problem: Relatively slow if primes are big.

- Basic algorithm:
  - Pick 2 large primes.
  - Multiply them together and give answer to others.
  - Others use this as “public key” to encrypt information
  - You know secret primes (“private key”).

Passing of messages

- Alice wants to send Bob a message.
- Alice knows Bob's public key (everyone does)
- Alice encrypts/locks information with public key
- Bob uses the private key to decrypt/unlock
- People in middle can see message, but it is locked



RSA details

- Encryption:
  - Turn message M into number m (e.g., ‘a’=1, ‘b’=2,…)
  - Public key is e and some (special) number N
  - Compute
    $$c = m^e (mod\ N)$$
  - Here “$mod\ N$” means r equals $N + r$ (clock arithmetic)
  - Answer c is sent
- “Special” N and secret d satisfy:
  $$m = m^{de} (mod\ N)$$

RSA details

- Further details:
  - N is product of two primes
  - e is (basically) random
  - Given e and primes, can compute d
  - Uses Fermat's little Theorem: If p is prime, then

$$m^p = m(mod\ p)$$

- Decrypt: Take $c^d = m^{ed} = m(mod\ N)$
- Security: Not easy to find d from e and N.

RSA Example

- Primes 17 and 13.
- Product is $N = 17 \times 13 = 221$
- Fermat's little Theorem:
  - $m^{16} = 1(mod\ 17)$
  - $m^{12} = 1(mod\ 13)$
  - 1 times 1 is 1 (also works with modular arith.), so
    - $m^{48} = (m^{16})^3 = 1\ (mod\ 17)$
    - $m^{48} = (m^{12})^4 = 1\ (mod\ 13)$
    - $m^{48} = 1\ (mod\ 17 \times 13)$

RSA Example

- Choose say $e = 11$
- Want $de = 1\ (mod\ 48)$
- Solve:
  - $d = 3\ (mod\ 16)$ (so $d = 3$, $d = 19$, or $d = 35$),
  - $d = 2\ (mod\ 3)$ (so $d = 35$)
- Message $m = 12 < 221$
  - $12^{11} = 743008370688 = 142\ (mod\ 221)$
  - $142^{35} = \cdots = 12\ (mod\ 221)$

Other one-way functions

- Another choice: ECC (elliptic curve cryptography)
- Elliptic curve is solutions to ($f(x)$ poly. degree 3)

$$C:\ \ y^2 = f(x)$$

- Example:

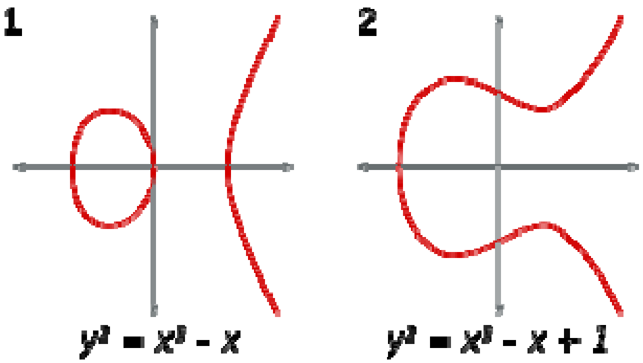
$$C:\ \ y^2 = x^3 - x$$

- Points (x, y) on curve:

  (0,0), (1,0), (-1,0), $\left(2, \sqrt{6}\right)$, $\left(2, -\sqrt{6}\right)$, …

Elliptic curves

- Can graph the solutions:

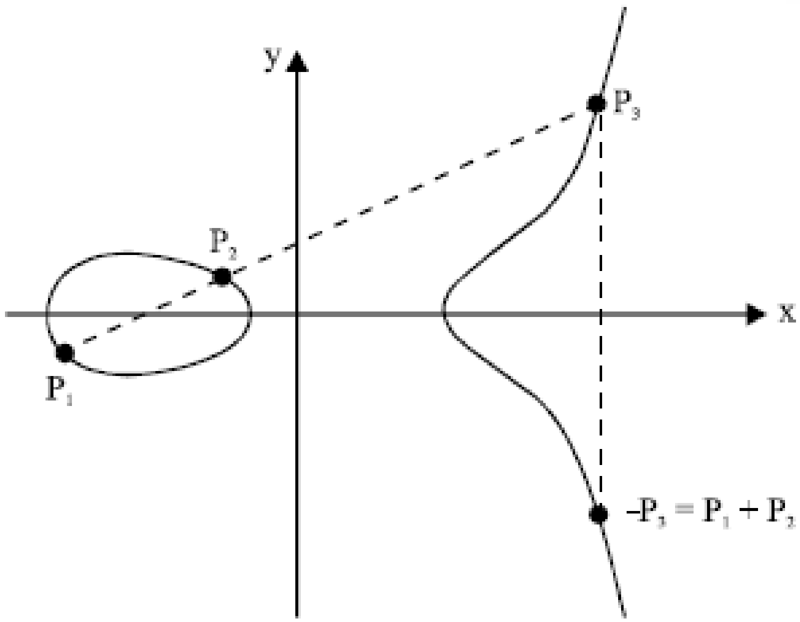


By GYassineMrabetTalk CC BY-SA 3.0,

- Between two points, there is a unique line
- Connects to one other point on the curve

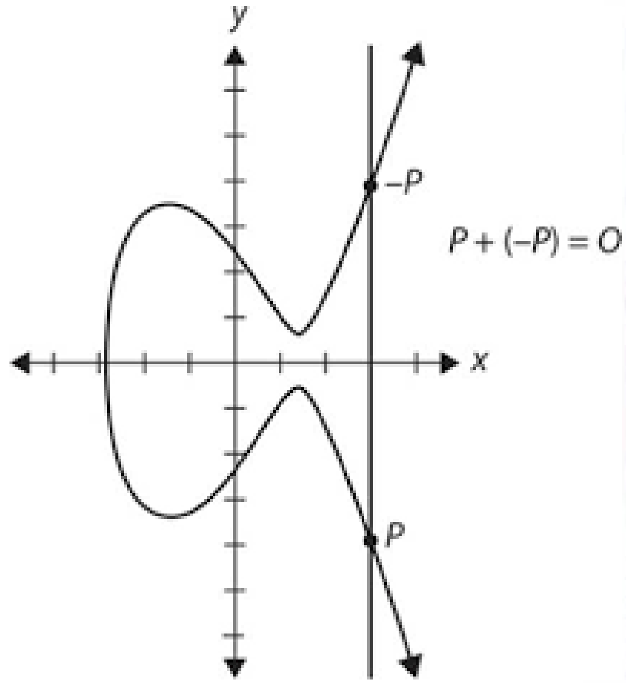
Elliptic curves: Addition (加法)

- Addition on points:
  - Between two points, there is a unique line
  - Connects to one other point on the line
  - Rotate around x-axis:
  - Extra point “at infinity”
    - Vertical lines add to infinity
    - Infinity like zero (0_c)
    - Why?



Elliptic curves: Addition and zero

- Addition with “infinity” :
  - Line between $P = (x, y)$ and infinity vertical
  - Other point on line is $(x, -y)$
  - Rotate around x-axis:
  - Gives $(x, y) = P$!
  - So adding infinity does nothing
  - Just like zero in addition!



Elliptic Curves: Addition

- Example:
  - Curve: C: $y^2 = x^3 + x - 1$
  - Points $(1,1), (2,3)$
  - Line between points:
    - slope = $\frac{3-1}{2-1} = 2,$
    - $y = 2x + b \rightarrow 1 = 2(1) + b \rightarrow y = 2x - 1$
  - Both equations at same time:
    $(2x - 1)^2 = x^3 + x - 1 \rightarrow x^3 - 4x^2 + 5x - 2 = 0$
    $(x - 1)^2(x - 2) = 0$

Elliptic Curves: Addition

- Overall:

$$(1,1) +_C (2,3) = (1,-1)$$

- Minus: flip over x-axis: $(1,1) +_C (1,-1) = 0_C$.
- So define $-_C(1,1) \coloneqq (1,-1)$
- Given a point P, can we compute

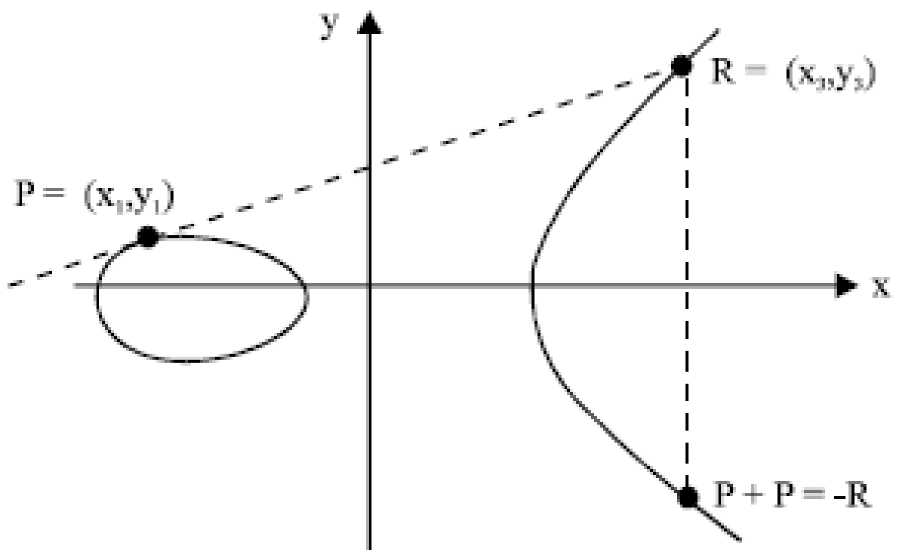
$$2_C P = P +_C P, 3_C P = P +_C P +_C P, \ldots ?$$

- Remember example: $(1,1) +_C (2,3) = -_C(1,1)$,

$$2_C(1,1) = -_C(2,3) = (2,-3)$$

Elliptic Curves: Addition

- Geometric interpretation of $2_C P$?
- Line with point and itself?
- **Tangent line**:



Kefa Rabah, *Theory and Implementation of Elliptic Curve Cryptography*, Journal of Applied Sciences **5** (2005), 604-633.

Elliptic Curves: Addition

- Example: Find $3_c(1,1)$
  - Curve: C: $y^2 = x^3 + x - 1$
  - Points $(1,1), (2,-3) = 2_c(1,1)$
  - Line between points:
    - slope = $\frac{-3-1}{2-1} = -4,$
    - $y = -4x + b \rightarrow 1 = -4(1) + b \rightarrow y = -4x + 5$
  - Both equations at same time:
    $(-4x+5)^2 = x^3 + x - 1 \rightarrow$
    $$x^3 - 16x^2 + 41x - 26 = 0$$
    $$\rightarrow (x-1)(x-2)(x-13) = 0$$

Elliptic Curves: Addition

- Overall:
  $$3_C(1,1) = (1,1) +_C (2,-3) = (13,-47)$$
- Can continue like this …
- Find $4_C(1,1), 5_C(1,1), \ldots$ Elliptic curve 的乘法.
- Easy to **teach a computer** to repeat the process!
- Question: How **fast** is calculation of $100_C(1,1)$?

Elliptic Curves: Fast Addition

- Add 100 times (**100 sums**, kind of **slow**)

- But … Can easily **double**:

$$2_C(1,1) +_C 2_C(1,1) = 4_C(1,1)$$
$$4_C(1,1) +_C 4_C(1,1) = 8_C(1,1)$$
$$\vdots$$

- Compute
$$100_C(1,1) = 64_C(1,1) + 32_C(1,1) + 4_C(1,1)$$

- Only double 6 times to get 64… **8 sums**!

Binary numbers to the rescue

- Can write number in binary:
  $abcd_2 = a \times 2^3 + b \times 2^2 + c \times 2^1 + d \times 2^0$
- So $1011_2 = 2^3 + 2 + 1 = 11$
- For $(2^n)_C(1,1)$, only need to double n times
- Note that 10 is **a lot** less than $2^{10} = 1024$
- **Many** less calculations, so faster

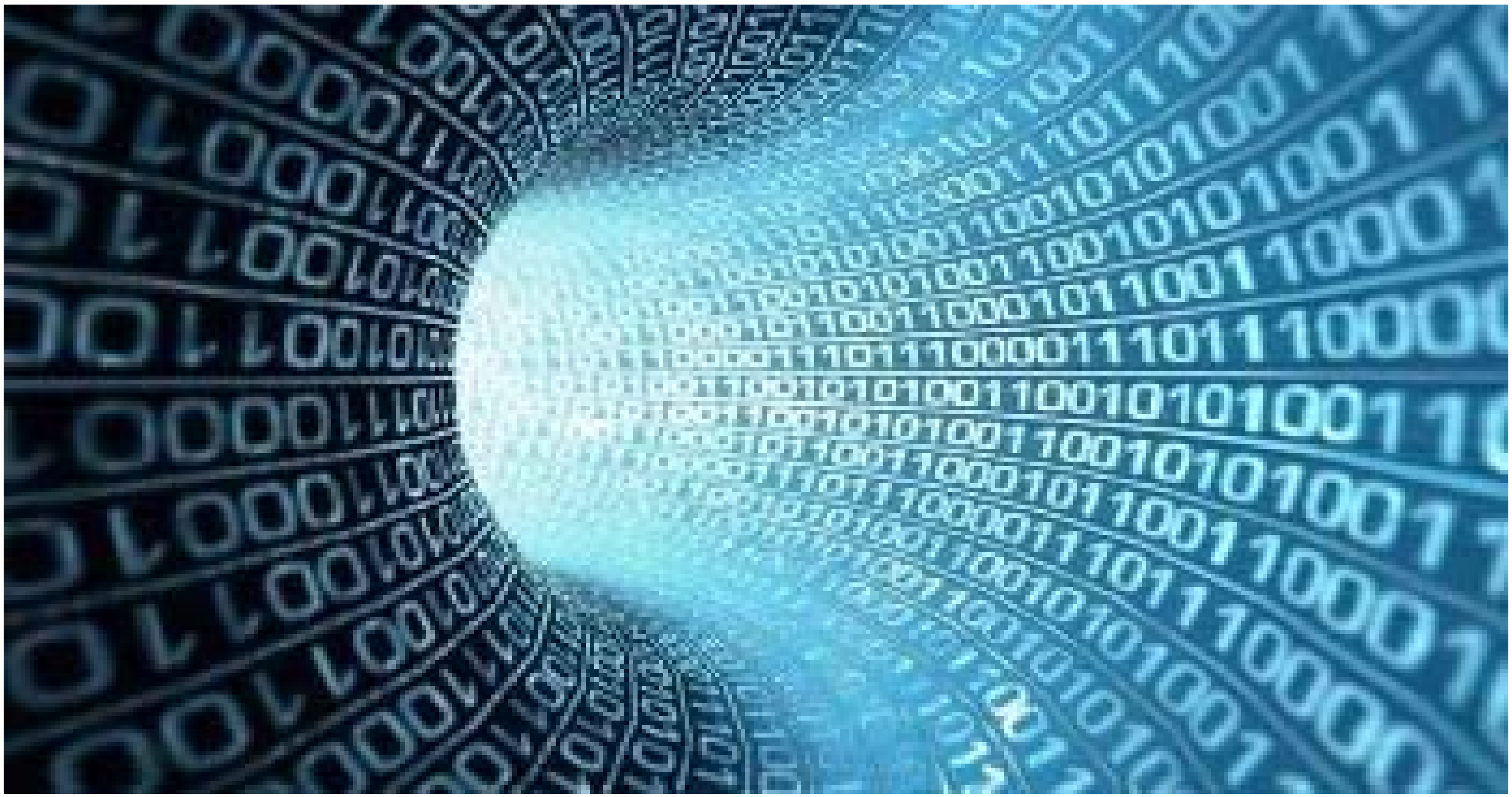
Elliptic curve cryptography

- One way function?
- Given $1000_C P$, can you find P?
- Easy in one way, seemingly hard in the other way
- Advantages/disadvantages of ECC:
  - Adv.: Smaller keys generally, fast to create
  - Dis.: Complicated to implement
- Used in Blockchain.

Key Sharing?

- Can we share a key?
- Session keys:
  - If I send you my key:
    - others can see it
    - Someone steals your message, replaces with own
  - Send you my key, encrypted with your public key
    - You decrypt with private key
    - Maybe someone in the middle pretended to be me?
    - You send back confirmation encrypted with my public key

Back to the future…



Homomorphic Encryption

- Suppose you have everyone's biological data.
- You want to compute some statistics/information.
- Your computer takes too long.
- You need help, but who can you **trust**?
- Idea: What if others could do your calculations, but get no data?
- How would you do that? Is it even possible?

Homomorphic Encryption

- You give the data m in encrypted form (say $E(m)$)
- You want others to be able to add and multiply, but never see m.
- What if
$$E(m_1 + m_2) = E(m_1) + E(m_2),$$
$$E(m_1 \times m_2) = E(m_1) \times E(m_2)?$$
- This is ***homomorphic Encryption***.

Partial Homomorphic Encryption

- Is homomorphic cryptography possible? Do we know some examples?
- Caesar (shift by 1):
  $E(m_1 + m_2) = m_1 + m_2 + 1,$
  $E(m_1 + m_2) = m_1 + 1 + m_2 + 1,$
  $E(m_1 \times m_2) = m_1 \times m_2 + 1$
  $E(m_1) \times E(m_2) = (m_1 + 1) \times (m_2 + 1)$
- Some methods partially work:
  - RSA: $E(m_1 \times m_2) = E(m_1) \times E(m_2)$

Homomorphic Encryption

- There are some rules …
- Needs to be safe …
  - Can‘t guess based on saving lots of encrypted messages
  - In particular, everything is zeros/ones
  - So can’t guess what is zero.
- A problem:

$$E(m) = E(m + 0) = E(m) + E(0)$$
$$\rightarrow E(0) = 0$$

Homomorphic Encryption

- Seems impossible to “hide” zero.
- Can it be “mostly true”?
- If “mostly true”, is the answer accurate?
- Should we give up?

Somewhat homomorphic encryption

- Gentry (2009): Added some “noise” so that encryption is almost homomorphic.
- Is it accurate?
  - Noise is “small” compared to main answer
- Allows **many** additions/multiplications
- Builds off of this to get homomorphic encryption:
  - Noise cancellation method
  - Takes a lot of operations, though

Homomorphic Encryption

- A trick:
  - Is zero really zero?
  - 12 o’clock is midnight, but also noon.
  - What if someone doesn’t know # hours on clock?
  - We pick a number of hours, but don’t tell anyone!
  - They read 26 o’clock, but don’t know the “real time”
- Pass info one way …
- Interpret the info differently yourself …

Homomorphic Encryption

- Now we have “a lot of zeros”
  - 0 is zero
  - 12 is zero
  - 24 is zero
  - 36 is zero
  - …
- Problem: Pattern too simple
  - Can be guessed
  - Not safe

Homomorphic Encryption

- Need to combine with other ideas
  - Pass some data…
  - Do something …
  - Get a value (secret)
  - Now apply modular arithmetic (secret)
- Need to make sure others can't compute value
  - Otherwise they can guess pattern in modular arith.
  - Starts to fall apart piece-by-piece
  - Some new ideas out there …


Back
to
Work!