How to Share a Secret

Matthew Roughan
<matthew.roughan@adelaide.edu.au>

Discipline of Applied Mathematics
School of Mathematical Sciences
University of Adelaide

Cryptography

THE UNIVERSITY OF ADELAIDE AUSTRALIA

- Cryptography is a critical part of modern life
  - not just for 007
  - banks use it all the time
  - secure web sites (look for `https` in the URL)
- Take some data and encrypt it using a key
  - if we know the key its easy to decrypt
  - if we don't know the key, it is impossible
  - actually, we usually only require that it would be very (very, very) unlikely that someone could translate it back.

Cryptography

THE UNIVERSITY OF ADELAIDE AUSTRALIA

Classical example far predates Da Vinci

- e.g. Caesar cipher (attributed to Julius Caesar)

text	A	B	C	D	E	F	G	H	I	J	K	L	M	N	O	P	Q	R	S	T	U	V	W	Y	X	Z
cipher	Y	X	Z	A	B	C	D	E	F	G	H	I	J	K	L	M	N	O	P	Q	R	S	T	U	V	W

- For example: shift letters by 3 for
  Friends, Romans, countrymen, lend me your ears
  Cofbkap, Oljykp, zlrkqoujbk, ibka jb ulro byop
- The key is how far you shift the letters.
  - not a very good cipher
  - could just try all 26 possible shifts
  - there are much better codes!!!

We can do better

- convert letters into numbers (and back)
- perform arbitrary computations
- the "key" is also a number, but much bigger
- we still have a problem with sharing keys?

Public Key Cryptography

THE UNIVERSITY OF ADELAIDE AUSTRALIA

- really clever
  - one key to encrypt, another to decrypt
  - avoids some problems of sharing keys
  - can use for authentication signatures as well as encryption
- based around mathematical problems
  - RSA (named after Rivest, Shamir and Adleman)
  - based on factoring large numbers into primes
- no one knows whether this is secure!!!
  - it depends on factoring into primes being hard
  - if this is true, then you can't get the key, even if you know the ciphertext, and plaintext!

Steganography

The German Embassy in Washington, DC, sent these messages in telegrams to their headquarters in Berlin during World War I (Kahn, 1996).

PRESIDENT'S EMBARGO RULING SHOULD HAVE IMMEDIATE NOTICE. GRAVE SITUATION AFFECTING INTERNATIONAL LAW. STATEMENT FORESHADOWS RUIN OF MANY NEUTRALS. YELLOW JOURNALS UNIFYING NATIONAL EXCITEMENT IMMENSELY.

APPARENTLY NEUTRAL'S PROTEST IS THOROUGHLY DISCOUNTED AND IGNORED. ISMAN HARD HIT. BLOCKADE ISSUE AFFECTS PRETEXT FOR EMBARGO ON BYPRODUCTS, EJECTING SUETS AND VEGETABLE OILS.

Steganography

Reading the first character of every word in the first message or the second character of every word in the second message.

PRESIDENT'S EMBARGO RULING SHOULD HAVE IMMEDIATE NOTICE. GRAVE SITUATION AFFECTING INTERNATIONAL LAW. STATEMENT FORESHADOWS RUIN OF MANY NEUTRALS. YELLOW JOURNALS UNIFYING NATIONAL EXCITEMENT IMMENSELY.

APPARENTLY NEUTRAL'S PROTEST IS THOROUGHLY DISCOUNTED AND IGNORED. ISMAN HARD HIT. BLOCKADE ISSUE AFFECTS PRETEXT FOR EMBARGO ON BYPRODUCTS, EJECTING SUETS AND VEGETABLE OILS.

Steganography

Reading the first character of every word in the first message or the second character of every word in the second message will yield the following hidden text:

PERSHING SAILS FROM N.Y. JUNE 1

Steganography

THE UNIVERSITY OF ADELAIDE AUSTRALIA

Steganography: (covered writing) The art and science of hiding information by embedding messages within other, seemingly harmless messages.

- has often been used to code information in text
- more recently, used to encode info. in other forms of data, as digital watermarks
  - images
  - audio
- invisible, but allows traceback of the source of data

Secrets

THE UNIVERSITY OF ADELAIDE AUSTRALIA

There are secrets, and then there are secrets

- cryptography/steganography aimed aimed keeping secrets from our enemies
- what if we want to keep them from our friends?
  - Imagine the secret was the key to arming your nuclear devices
    - You don't give the code to just one general
      - what if he goes mad and launches the rockets for a laugh.
    - You could break the code into three bits, and give one each to three generals
      - What if one of them dies of heart attack when his code was needed?

How to share a secret

THE UNIVERSITY OF ADELAIDE AUSTRALIA

Shamir's secret sharing: [1]

- generate a random degree $M-1$ polynomial

$$y = a_0 + a_1 x + a_2 x^2 + \cdots a_{M-1} x^{M-1}$$

  - make the y-intercept the "secret"
    - we can always write our data as a number
- each "share" is one point on the polynomial (x_i, y_i) for $x_i \neq 0$
- with M shares we can exactly reconstruct the polynomial (and hence the secret)
- with $M-1$ or fewer shares we don't learn anything useful

Experiment

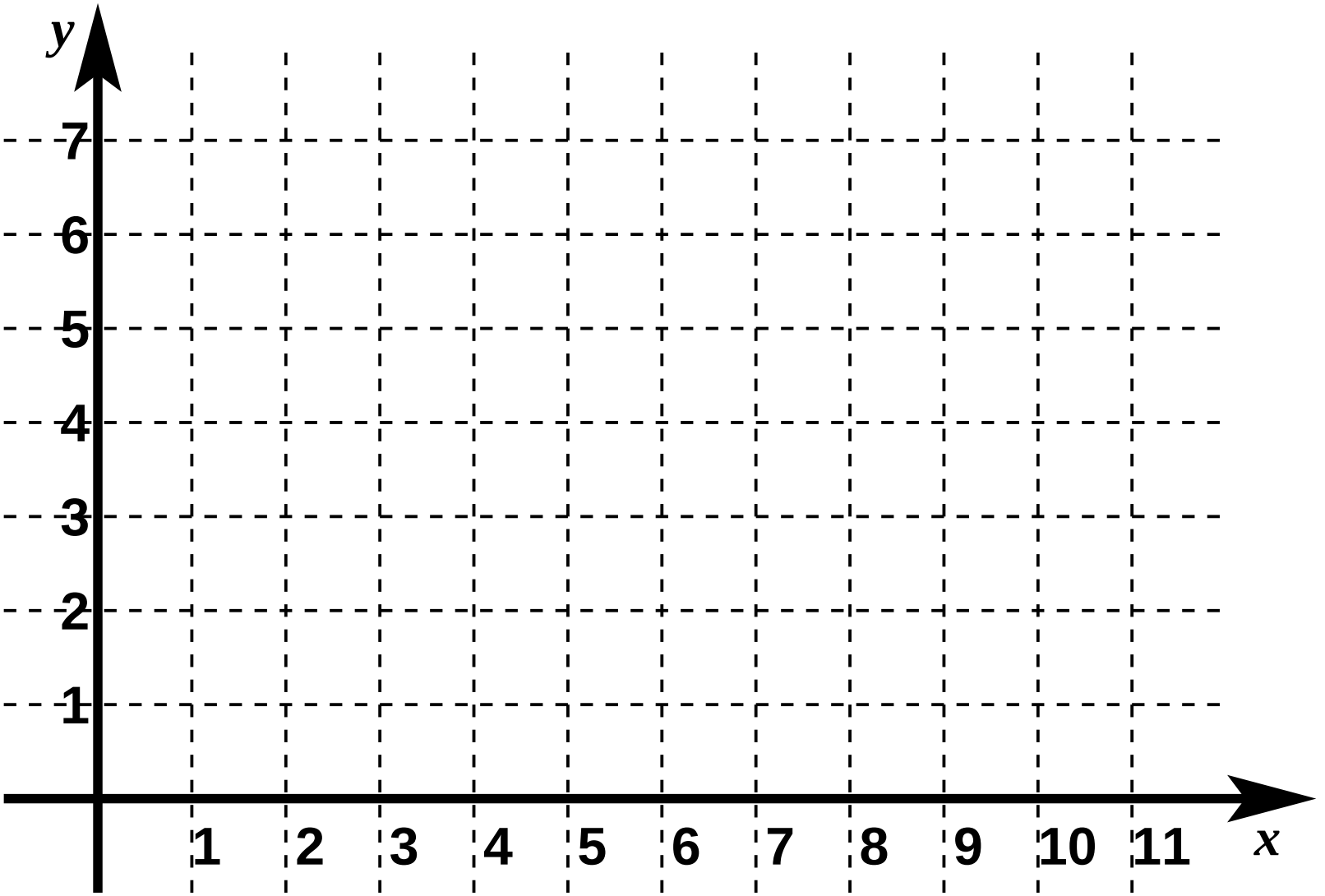
Lets try for $M = 2$

- you each have a single point (x_i, y_i)
- get into groups of 2
- any pair of you have enough information to work out the polynomial
  - in the case $M = 2$, we look for straight lines

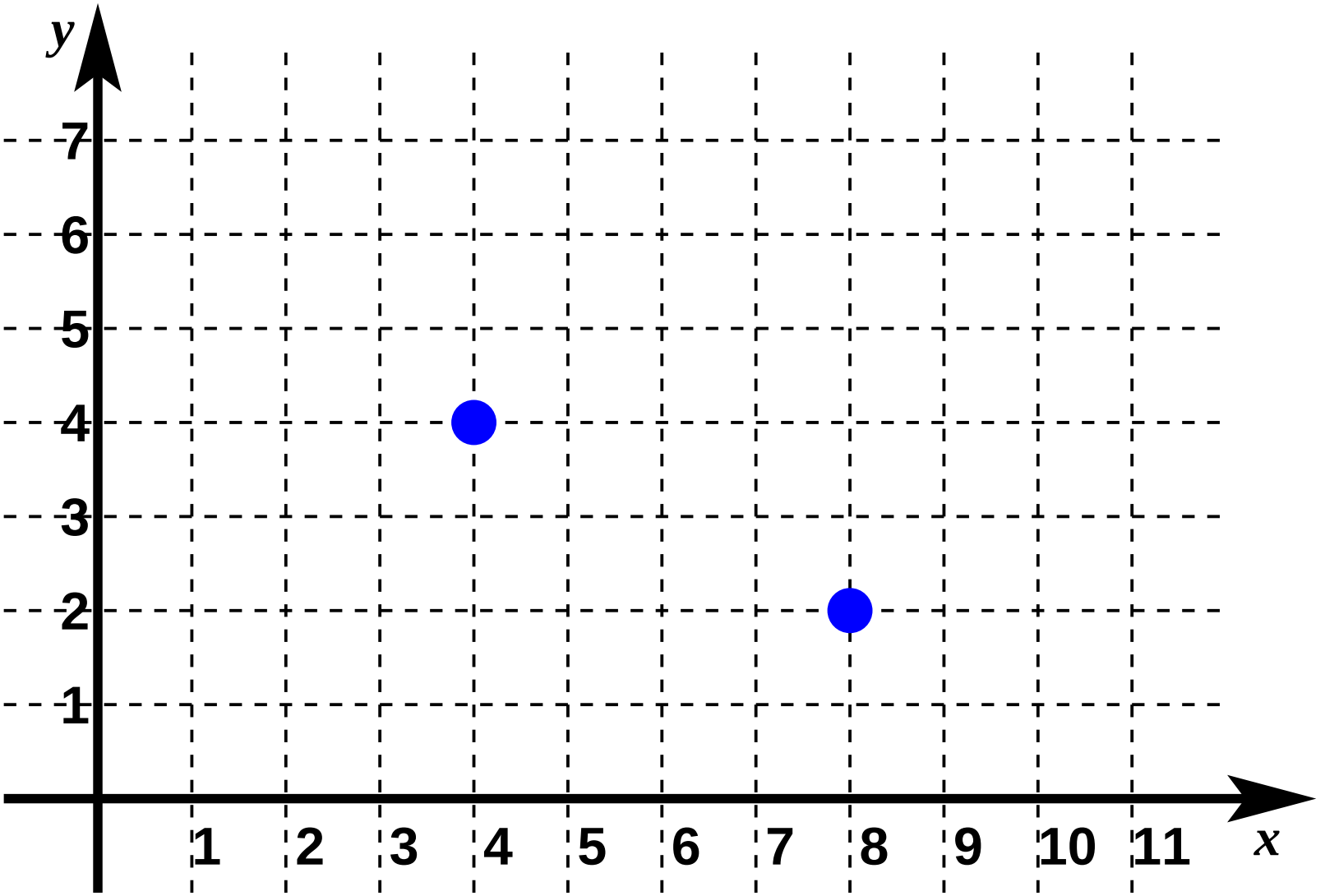
$$y = a_0 + a_1 x$$

- the secret is the y-intercept a_0

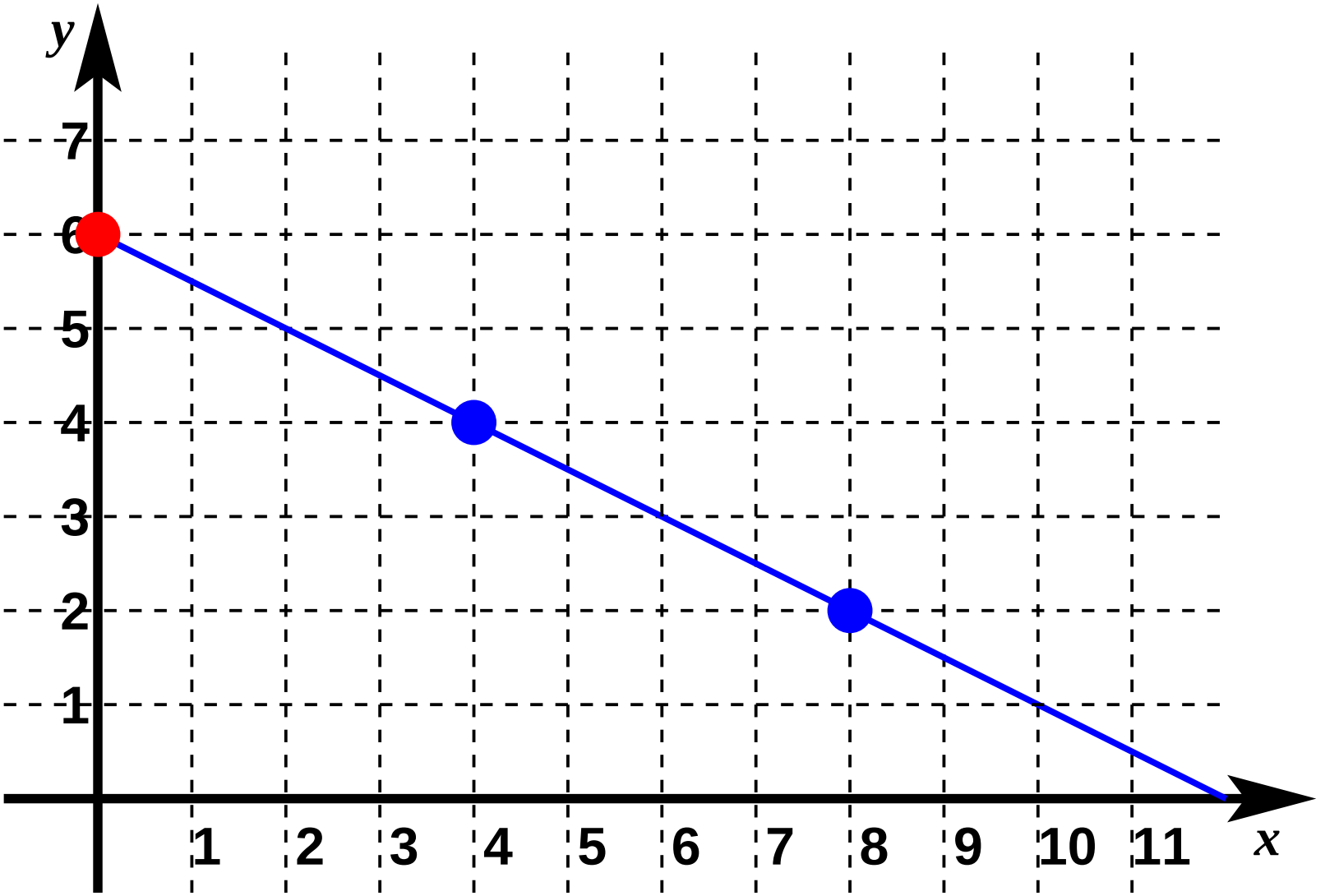
Experiment



Experiment



Experiment



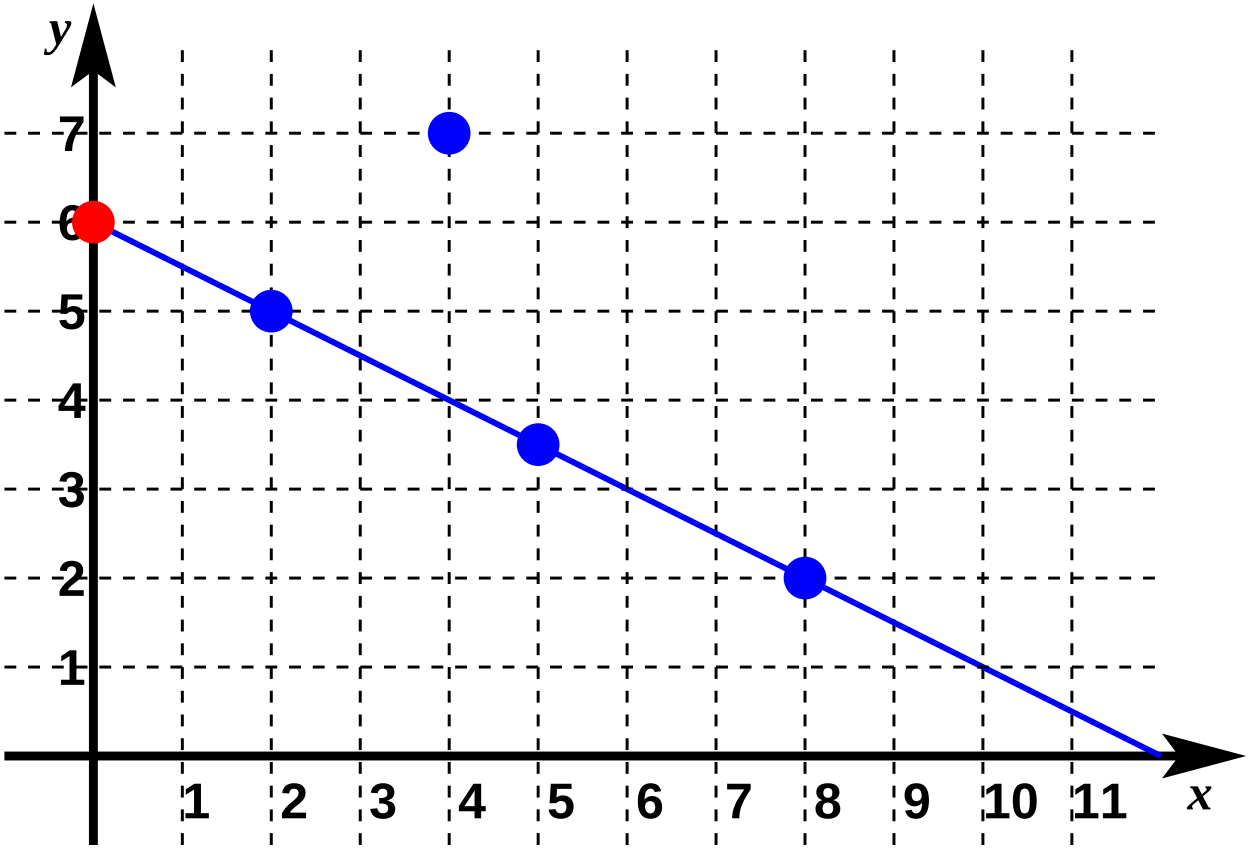
Experiment with cheating

THE UNIVERSITY OF ADELAIDE AUSTRALIA

- Some of you got the wrong answer
- That's because I told some people to lie!
- now lets try it in groups of 4

Experiment with cheating

- Some of you got the wrong answer
- That's because I told some people to lie!
- now lets try it in groups of 4



Properties

- The honest majority wins the day!
  - secret sharing can be made more robust
  - can avoid problems of deliberate cheats
  - and accidental errors
- It has other nice properties
  - its easy to generate new secrets if one is lost
  - easy to add people (or remove them)
  - different people can have different levels of trust (give them more shares)

So why do I care?

THE UNIVERSITY OF ADELAIDE AUSTRALIA

- How much traffic is there on the Internet?
  - the argument is made [2] that lack of such data contributed to the tech-wreck
  - regulators need such information
    - e.g. anti-trust cases
- Detecting distributed attacks
  - DDoS (Distributed Denial of Service), Worms/viruses,
  - e.g. Worms are easy to detect once they are well under way, but if you want to detect it early, the more data points you have the better.
- but data is hard to get

Why is data hard to get?

THE UNIVERSITY OF ADELAIDE AUSTRALIA

- No particular company sees all the Internet
  - the Internet is (by its nature) distributed
  - each company can add a perspective
- But companies don't share
  - companies don't want to reveal data
    - afraid of misuse of data
    - afraid it will reveal business secrets
    - afraid it will reveal incompetence
  - sometimes they are not allowed to
    - e.g. privacy legislation [3]

Dining cryptographers

THE UNIVERSITY OF ADELAIDE AUSTRALIA

- N cryptographers are having dinner
- When it is time to pay the bill, the waiter tells them that someone has already paid
- the cryptographers are suspicious by nature (particularly Alice and Bob).
  - they suspect the NSA has paid
- not wanting to be compromised by such an association, they need to find out if someone at the table paid, or an external party such as the NSA
- how can they do so, without anyone revealing whether they paid or not?
  - of course, the waiter is sworn to secrecy

Secure Distributed Summation

Problem: N parties each have one value v_i and they want to compute the sum

$$V = \sum_{i=1}^{N} v_i$$

but they don't want any other party to learn their value.

SDS algorithm [4]

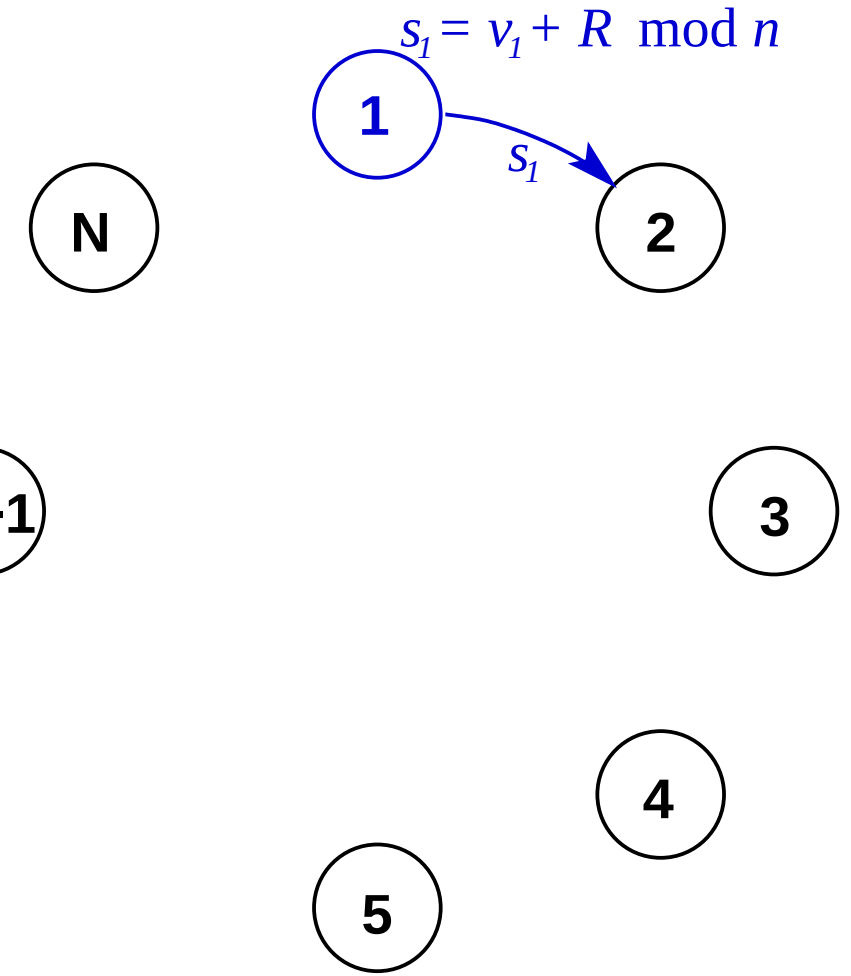
Assume the value $V \in [0,n]$ (for large n)

party 1: randomly generate $R \sim U(0,n)$
party 1: compute $s_1 = v_1 + R \mod n$
party 1: pass s_1 to party 2
for i=2 to N
  party i: compute $s_i = s_{i-1} + v_i \mod n$
  party i: pass s_i to party $i+1$
endfor
party 1: compute $v_N = s_N - R \mod n$

Finally, party 1 has to share the result with the others.

s_i will be uniformly randomly distributed over $[0,n]$ and so we learns nothing about any other parties values.

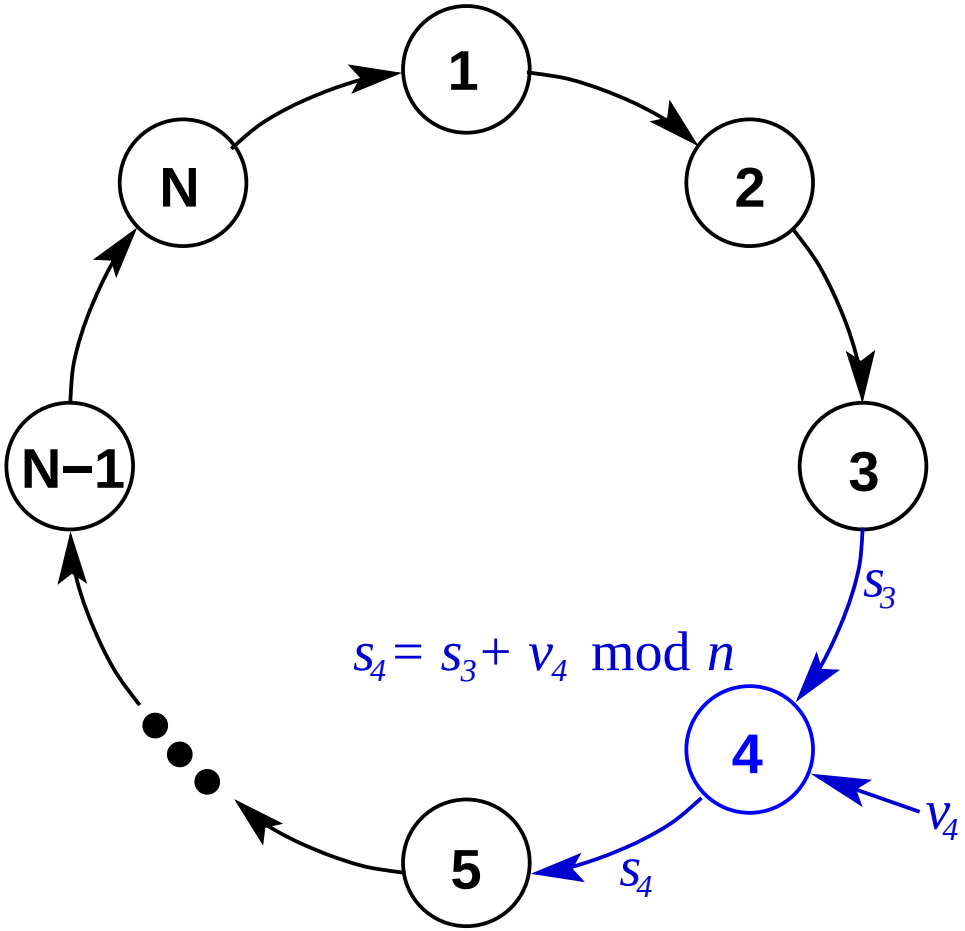
SDS algorithm



party 1: randomly generate $R \sim U(0,n)$
party 1: compute $s_1 = v_1 + R \mod n$
party 1: pass s_1 to party 2
for i=2 to N
  party i: compute $s_i = s_{i-1} + v_i \mod n$
  party i: pass s_i to party $i+1$
endfor
party 1: compute $v_N = s_N - R \mod n$

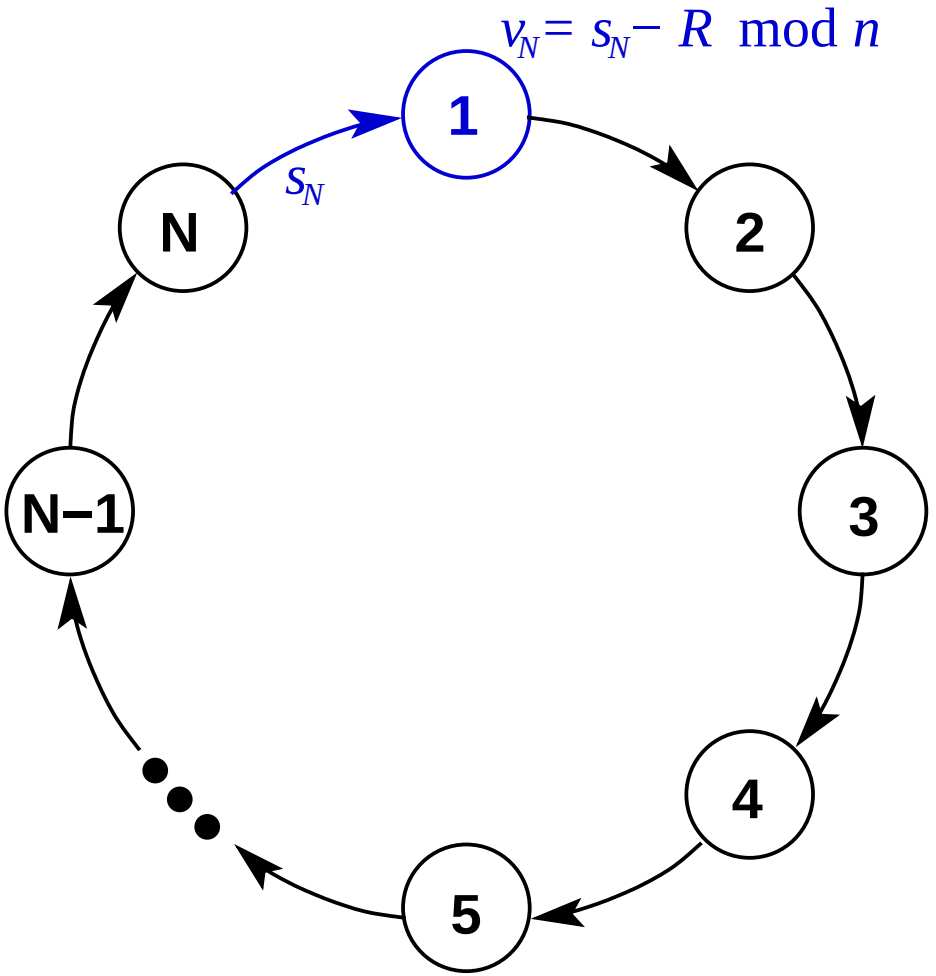
SDS algorithm

party 1: randomly generate $R \sim U(0,n)$
party 1: compute $s_1 = v_1 + R \mod n$
party 1: pass s_1 to party 2
for i=2 to N
  party i: compute $s_i = s_{i-1} + v_i \mod n$
  party i: pass s_i to party $i+1$
endfor
party 1: compute $v_N = s_N - R \mod n$



SDS algorithm

party 1: randomly generate $R \sim U(0,n)$
party 1: compute $s_1 = v_1 + R \mod n$
party 1: pass s_1 to party 2
for i=2 to N
  party i: compute $s_i = s_{i-1} + v_i \mod n$
  party i: pass s_i to party $i+1$
endfor
party 1: compute $v_N = s_N - R \mod n$



The Problem

THE UNIVERSITY OF ADELAIDE AUSTRALIA

What if $j-1$ and $j+1$ collude

- They would know s_j and s_{j-1}
- $s_j - s_{j-1} = v_j \bmod n$
- collusion breaks the secrecy

Can we fix this?

- Yes!
- we need Shamir's secret sharing

The Trick

We can add polynomials, i.e. $p(x) = p_1(x) + \cdots + p_N(x)$

- So each party generates a secret polynomial, such that $p_i(0) = v_i$
- All parties agree on M values x_j
- Each party creates M secret shares

$$y_{ij} = p_i(x_j)$$

- We perform a SDS on each of these sets of shares*

$$Y_j = \sum_i y_{ij}$$

- Given the (x_j, Y_j) we can determine $p(x)$, and hence

$$p(0) = \sum_i v_i$$

Conclusion

THE UNIVERSITY OF ADELAIDE AUSTRALIA

- There are lots of ways of sharing secrets
- They are based on some rather elegant maths.
  - there are interesting unsolved problems
- My interest is in applications of secure distributed computing
  - how can we compute values without sharing input
  - privacy preserving data mining
- Finally: The purpose of cryptography is to make the message unintelligible except to one person
  - hopefully I have been unsuccessful

References

[1] A. Shamir, "How to share a secret," Communications of the ACM, vol. 22, no. 11, pp. 612–613, 1979.

[2] A. M. Odlyzko, "Internet traffic growth: Sources and implications," in Optical Transmission Systems and Equipment for WDM Networking II (B. B. Dingel, W. Weiershausen, A. K. Dutta, and K.-I. Sato, eds.), vol. 5247, pp. 1–15, Proc. SPIE, 2003.

[3] "Data-mining moratorium act of 2003." Introduced in Senate of the United States in January 2003. `http://thomas.loc.gov/cgi-bin/query/z?c108:S.188:`.

[4] C. Clifton, M. Kantarcioglu, J. Vaidya, X. Lin, and M. Zhu, "Tools for privacy preserving distributed data mining," SIGKDD Explorations, vol. 4, December 2002.