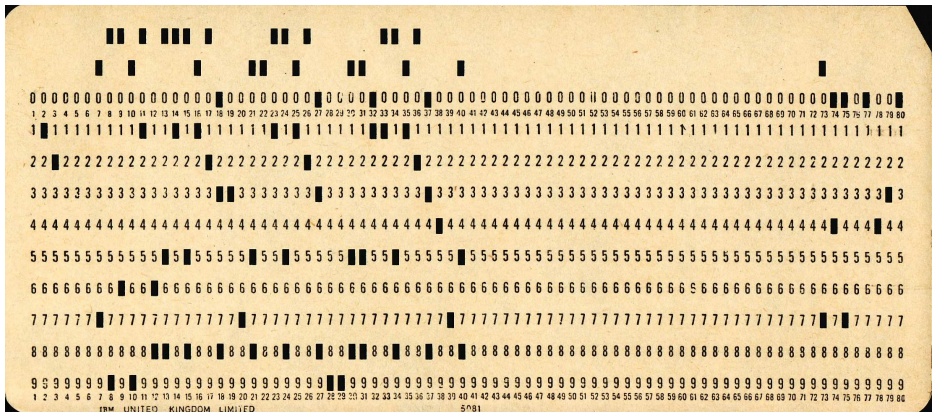
Inrtoduwtion to Erorr Dwtetcion and Erorr Czrrectmon

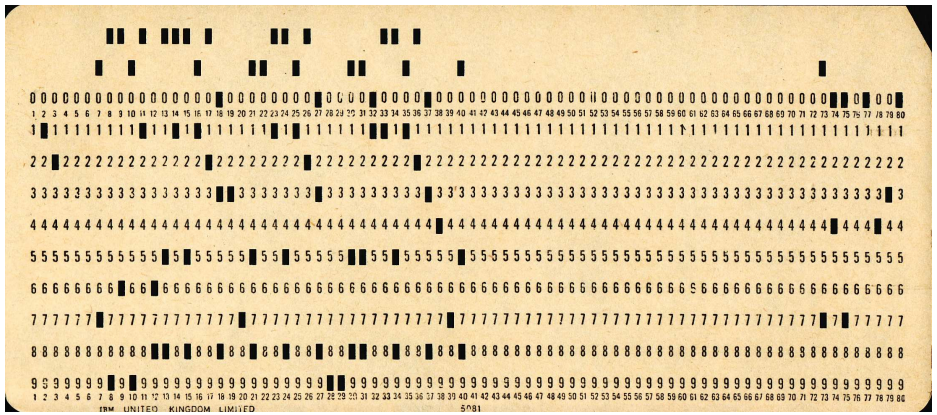
Setevn .J Mzlwer
sjm1@williams.edu
http://www.williams.edu/Mathematics/sjmiller



Bulryngvon, Jnue 14, 2091

Introduction to Error Detection and Error Correction

Steven J. Miller
sjm1@williams.edu
http://www.williams.edu/Mathematics/sjmiller



Burlington, June 19, 2019

Introduction

Cryptography Basics

Enough to send 0's and 1's:

- A = 00000, B = 00001, C = 00010, . . .
  Z = 11010, 0 = 11011, 1 = 11100, ...

Two major issues:

- Transmit message so only desired recipient can read.
- Ensure correct message received.

Cryptography Basics

Enough to send 0's and 1's:

- A = 00000, B = 00001, C = 00010, . . .
  Z = 11010, 0 = 11011, 1 = 11100, ...

Two major issues:

- Transmit message so only desired recipient can read.
- Ensure correct message received.

Bit Error Dangers: RSA

If receive wrong bit in RSA, message completely different.

Bit Error Dangers: RSA

If receive wrong bit in RSA, message completely different.

Secret: $p = 15217$, $q = 17569$, $d = 80998505$.
Public: $N = pq = 267347473$, $e = 3141593$.
Note: $ed = 1 \bmod (p-1)(q-1)$.
Message: $M = 195632041$, send $M^e \bmod N$ or
$X = 121209473$.
Decrypt: $X^d \bmod N$ or 195632041.

Bit Error Dangers: RSA

If receive wrong bit in RSA, message completely different.

Secret: $p = 15217$, $q = 17569$, $d = 80998505$.
Public: $N = pq = 267347473$, $e = 3141593$.
Note: $ed = 1 \bmod (p-1)(q-1)$.
Message: $M = 195632041$, send $M^e \bmod N$ or
$X = 121209473$.
Decrypt: $X^d \bmod N$ or 195632041.

Imagine receive $\widetilde{X} = 121209483$.
Message 195632041
Decrypts 121141028, only two digits are the same!

Outline

Will concentrate on Error Detection and Correction.

- Detection: Check Digit
- Correction: Majority Rules and Generalization

Check Digit

Check Digit

If easy to read again, just need to detect error.

Check Digit

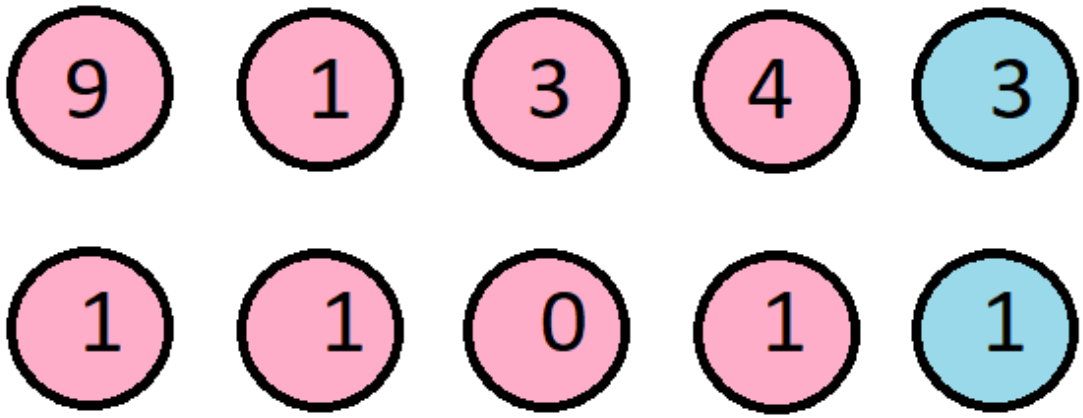
If easy to read again, just need to detect error.

Think scanner at a supermarket....

Check Digit

If easy to read again, just need to detect error.

Think scanner at a supermarket....

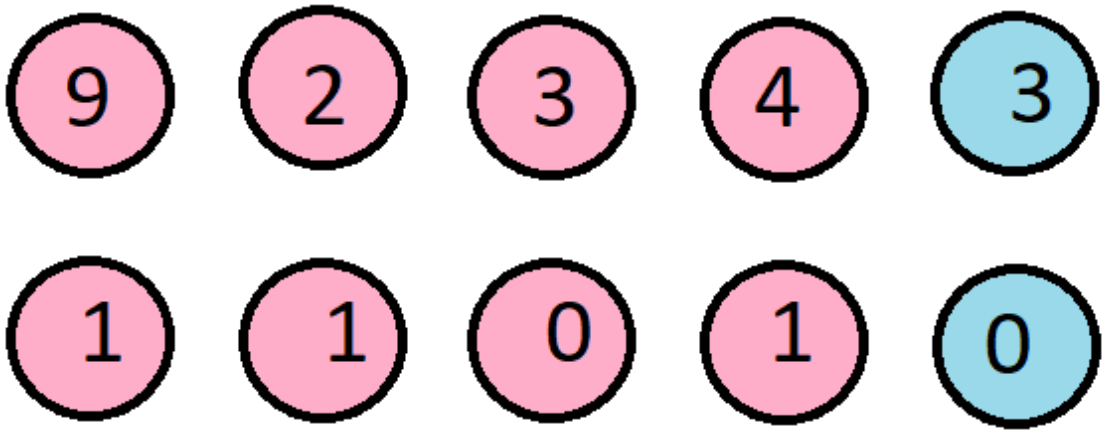


Last digit makes sum 0 mod 10 (or 0 mod 2).

Check Digit

If easy to read again, just need to detect error.

Think scanner at a supermarket....



Last digit makes sum 0 mod 10 (or 0 mod 2).

Next Steps

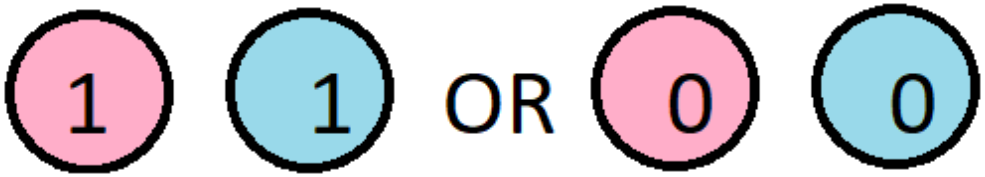
More involved methods detecting more: The Verhoeff algorithm catches single digit errors and flipping adjacent digits: https://en.wikipedia.org/wiki/Verhoeff_algorithm.

Want to detect where the error is:

Next Steps

More involved methods detecting more: The Verhoeff algorithm catches single digit errors and flipping adjacent digits: https://en.wikipedia.org/wiki/Verhoeff_algorithm.

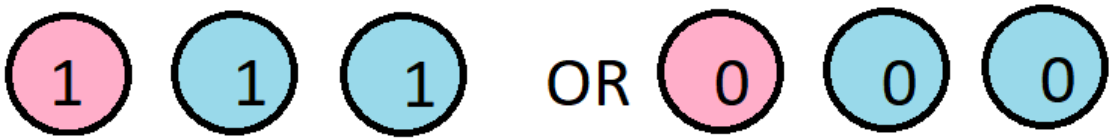
Want to detect where the error is: Tell me twice!



Majority Rules

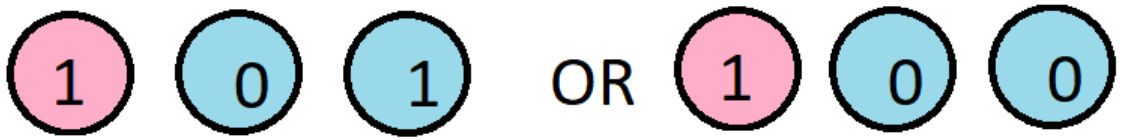
Tell Me Three Times

Tell Me Three Times detects and *probably* corrects (need probability of an error small).



Tell Me Three Times

Tell Me Three Times detects and *probably* corrects (need probability of an error small).



Tell Me Three Times

Crucially uses binary outcome: https://www.youtube.com/watch?v=RerJWv5vwxc and https://www.youtube.com/watch?v=vWCGs27_xPI.

What is the problem with this method?

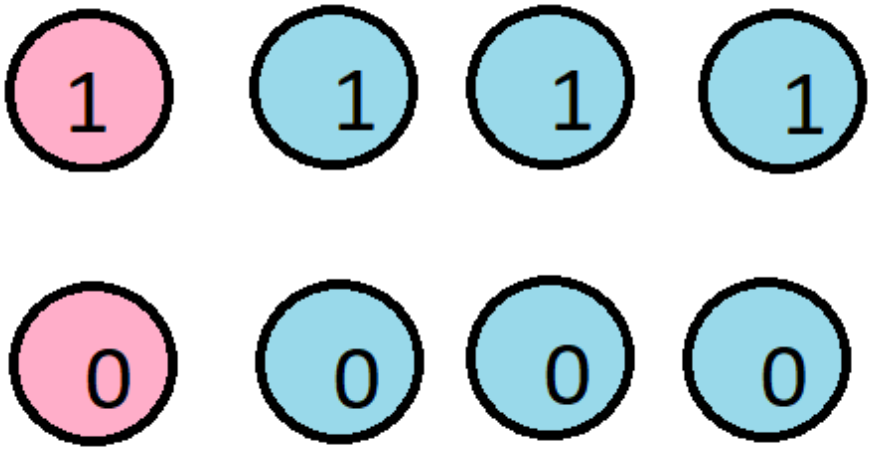
Tell Me Three Times

Crucially uses binary outcome: https://www.youtube.com/watch?v=RerJWv5vwxc and https://www.youtube.com/watch?v=vWCGs27_xPI.

What is the problem with this method? Only one-third is information.

How can we do better?

Tell Me *n* Times

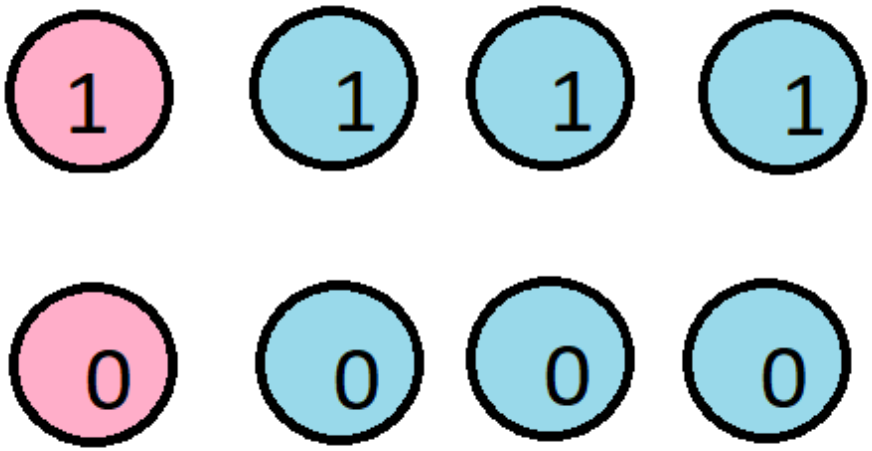


Tell Me Four Times: only 25% of message is data (general case just $1/n$).

Want to correct errors but still send a lot of information.

What's a success?

Tell Me *n* Times



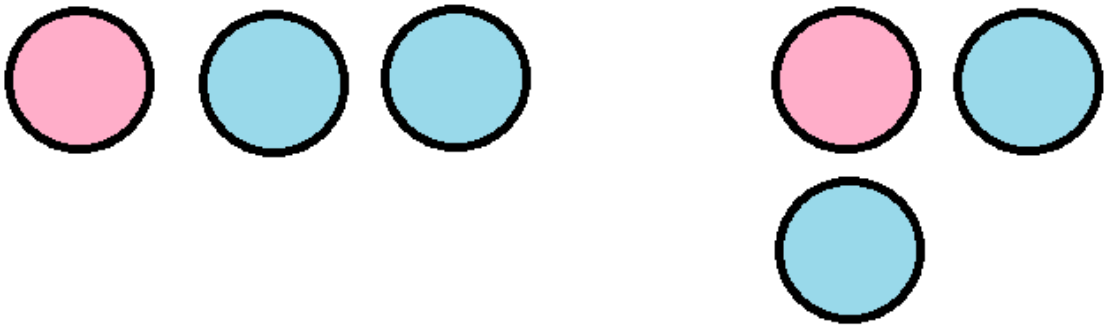
Tell Me Four Times: only 25% of message is data (general case just $1/n$).

Want to correct errors but still send a lot of information.

What's a success? Greater than 50% is data.

Tell Me Three Times (revisited)

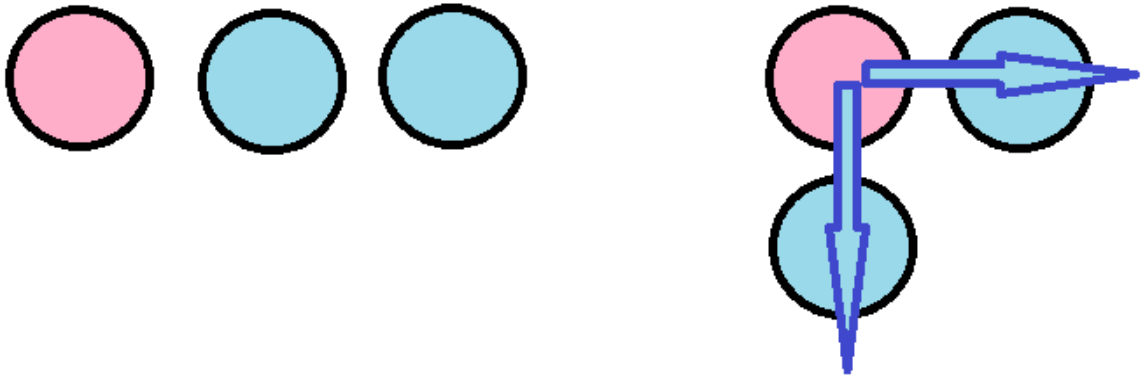
Let's revisit Tell Me Three Times:



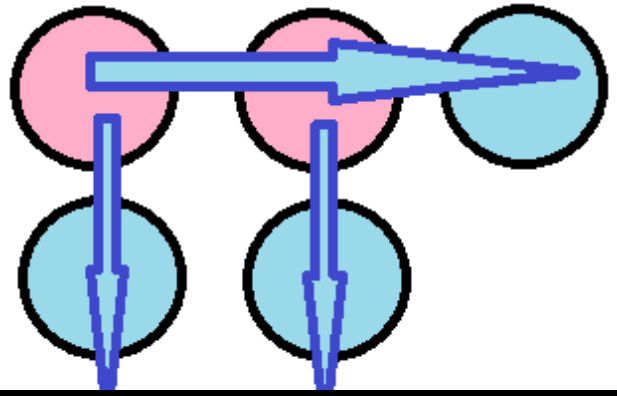
How should we do two data points?
How many check digits do you expect?

Tell Me Three Times (revisited)

Let's revisit Tell Me Three Times:

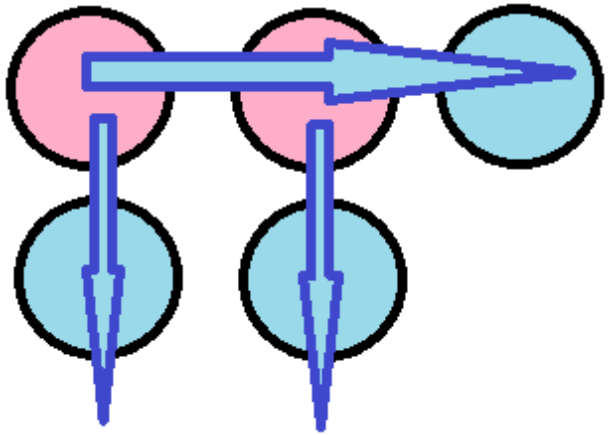


How should we do two data points?
How many check digits do you expect?



Two of Five

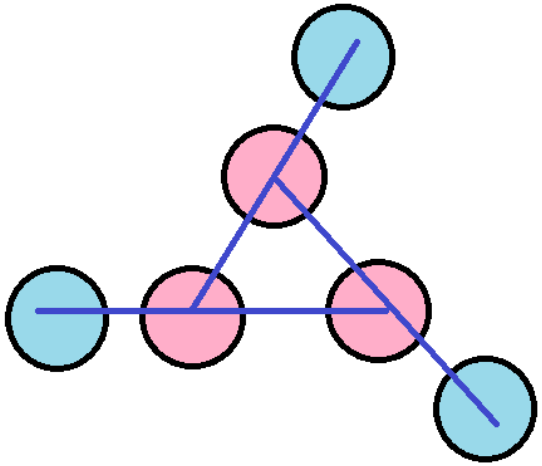
This is better: 2 of 5 or 40% of message is data!

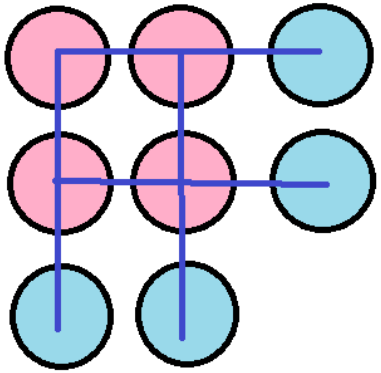


Unfortunately still below 50%.

How many data points should we try next: 3, 4, 5, ...?

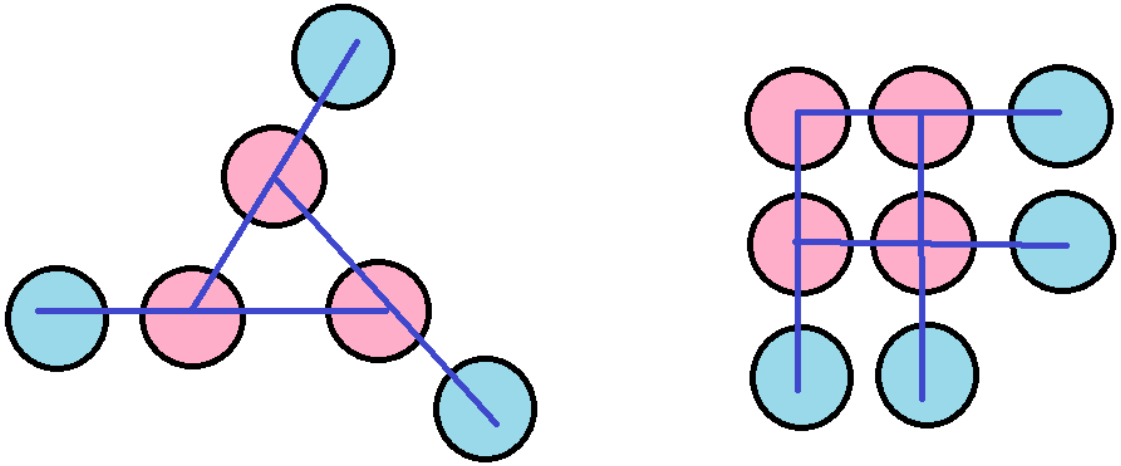
Three and Four Bits of Data





Which is better?

Three and Four Bits of Data

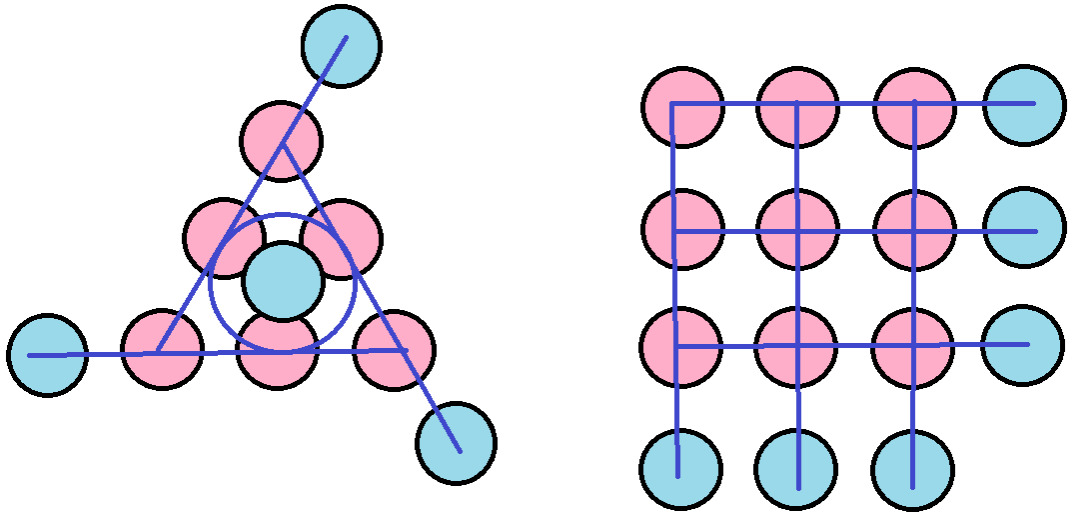


Which is better? Both 50% but fewer needed with triangle.

What should we do next: 5, 6, 7, 8, 9, ...?

Triangle and Square Numbers

$T_n = n(n+1)/2$ and $S_n = n^2$.

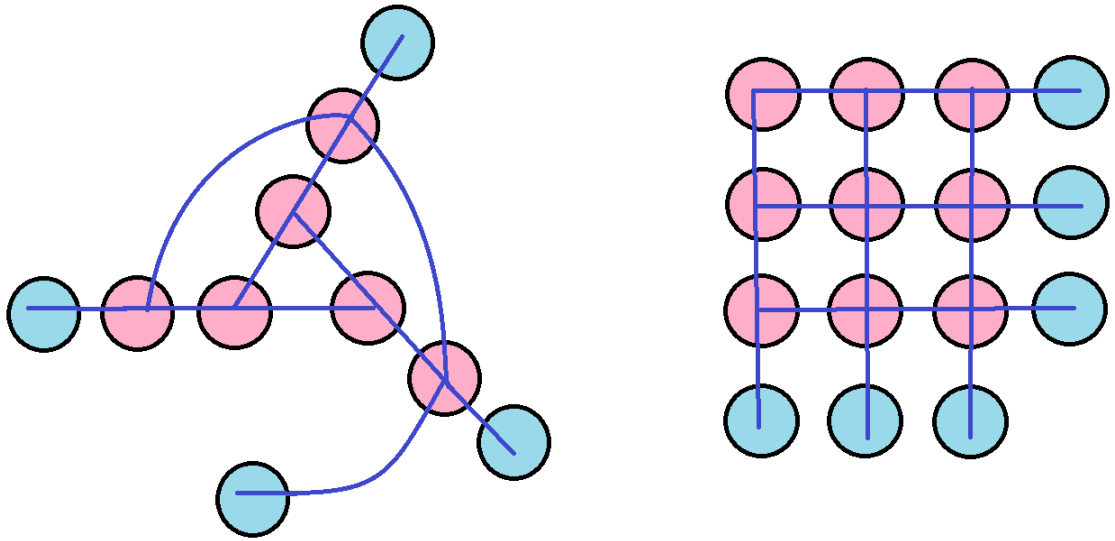


Both give 60% of the message is data. Can we continue?

Data on exactly two lines, check bits on one.

Triangle and Square Numbers

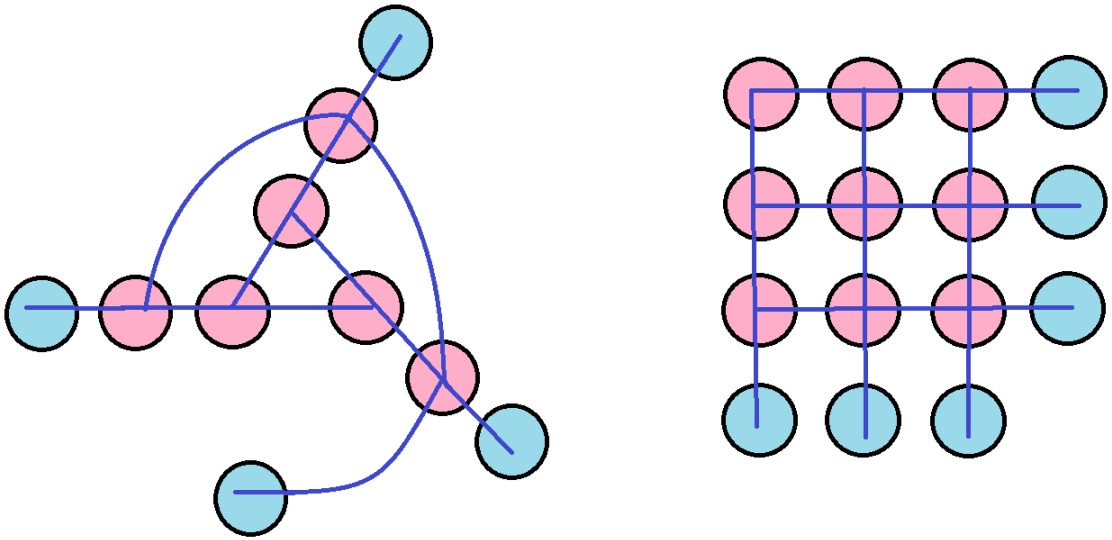
$T_n = n(n+1)/2$ and $S_n = n^2$.



Both give 60% of the message is data. Can we continue?

Data on exactly two lines, check bits on one.

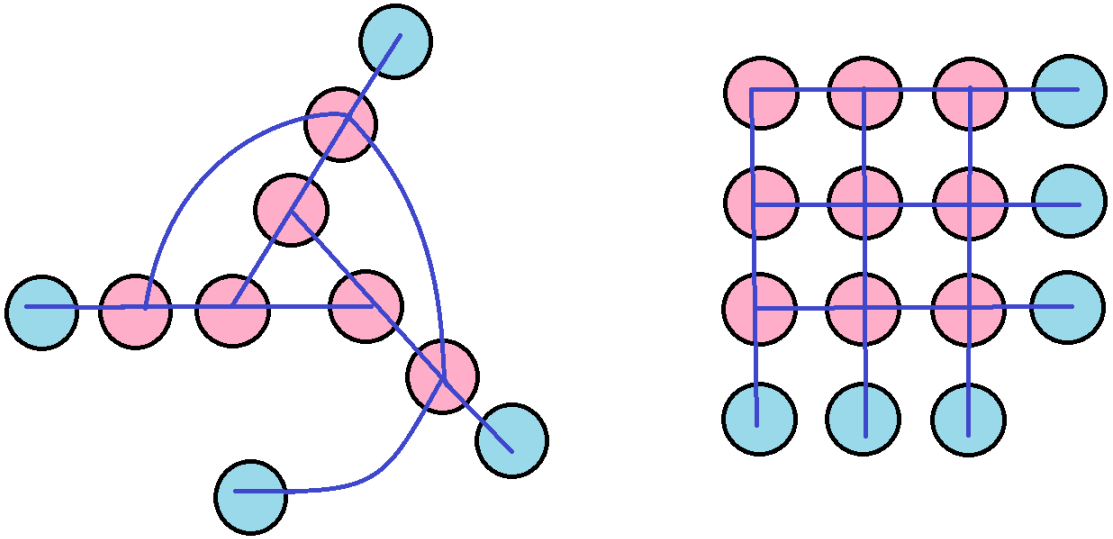
Triangle and Square Systems



Triangle: $T_n = n(n+1)/2$ data, $n+1$ check, so $(n+2)(n+1)/2$ bits total and $n/(n+2)$ information.

Square: $S_n = n^2$ data, $2n$ check, so $n^2 + 2n$ bits total and $n/(n+2)$ information.

Triangle and Square Systems



Can get as high a percentage information as desire, at a cost of longer string (and thus more likely to have two errors).

Generalizations

What is a better geometry to use?

Generalizations

$2 \times 2 \times 2$: 8 data points, 6 check bits (for planes): info is $8/14 \approx 57\%$.

$3 \times 3 \times 3$: 27 data points, 9 check bits (for planes): info is $27/36 = 75\%$.

For 6×6 data square info is $36/48 = 75\%$, for T_7 is $28/36 \approx 77.78\%$.

Generalizations

$4 \times 4 \times 4$: 64 data points, 12 check bits: info is $64/76 \approx 84.21\%$.

For 9×9 data square info is $81/99 \approx 81.82\%$.

For T_{11} triangle: 66 data points, info is $66/79 \approx 83.54\%$.

Generalizations

$n \times n \times n$: n^3 data points, $3n$ check bits: info is $n^2/(n^2+3)$.

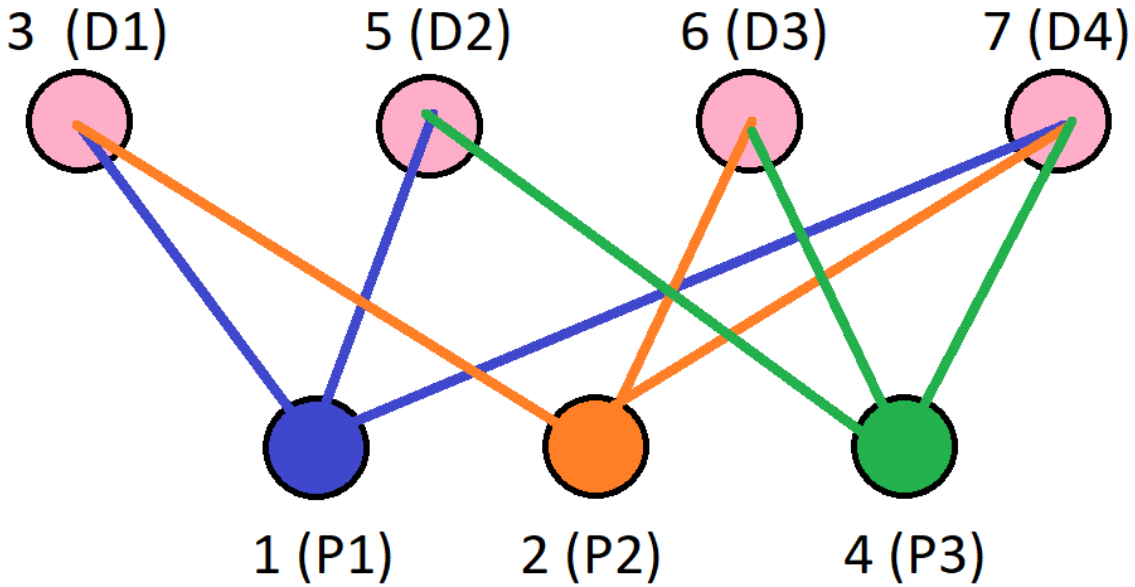
Better percentage is information for large n; how should we generalize?

Other Approaches

Hamming Codes: Can send a message with 7 bits, 4 are data, and can correct one error: https://en.wikipedia.org/wiki/Hamming_code.

Extended binary Golay code: Can send a message with 24 bits, 12 are data, can correct any 3-bit errors and can detect some other errors: https://en.wikipedia.org/wiki/Binary_Golay_code.

Manhamming



- If no errors, all correct.
- If only one color error, is P1, P2 or P3.
- If just blue and orange is D1.
- If just blue and green is D2.
- If just orange and green is D3
- If all wrong is D4.

Comparison

Say want to transmit around $2^{12} = 4096$ bits of data.

Can do a square and cube; the Hamming code will do $2^{12} - 1 - 12$.

- Square: 4096 out of 4224 data: 96.9697%.
- Cube: 4096 out of 4144 data: 98.8417%.
- Hamming: 4083 out of 4095 data: 99.707%.

All converge to 100%, difference narrows as size increases.

Interleaving

Say transmit

01111010010101010101010101010101010101011110...

but a localized burst of noise, receive

01110111010101010101010101010101010101011110...

Interleaving

Transmit every fourth:

- 01000000001 $\mapsto$ 00000000001
- 10111111111 $\mapsto$ 11111111111
- 11000000001 $\mapsto$ 11000000001
- 10111111110 $\mapsto$ 11111111110

Steganography

Can you see the cat in the tree?

Transmitting Images

How to transmit an image?

- Have an $L \times W$ grid with LW pixels.
- Each pixel a triple, maybe (Red, Green, Blue).
- Often each value in $\{0, 1, 2, 3, \ldots, 2^n - 1\}$.
- $n = 8$ gives 256 choices for each, or 16,777,216 possibilities.

Steganography

Steganography: Concealing a message in another message: https://en.wikipedia.org/wiki/Steganography.

Steganography

Steganography: Concealing a message in another message: https://en.wikipedia.org/wiki/Steganography.

Take one of the colors, say red, a number from 0 to 255.

Write in binary: $r_7 2^7 + r_6 2^6 + \cdots + r_1 2 + r_0$.

If change just the last or last two digits, very minor change to image.

Can hide an image in another.

If just do last, can hide a black and white image easily....

Can you see the cat in the tree?

Can you see the cat in the tree?

Introduction to Cryptography: Alphabet Codes

Introduction to Cryptography: Alphabet Codes:

Steven J. Miller

http://www.williams.edu/Mathematics/sjmiller/public_html

VCTAL: Burlington, June 19, 2019

Caesar and Affine Ciphers

Encryption Functions

Function from alphabet to alphabet such that:

- Different letters go to different letters.
- All letters hit.

Do we need both conditions?

Encryption Functions

Function from alphabet to alphabet such that:

- Different letters go to different letters.
- All letters hit.

Do we need both conditions?
Setup: x input letter, $f(x)$ is the encrypted letter.

Caesar Cipher

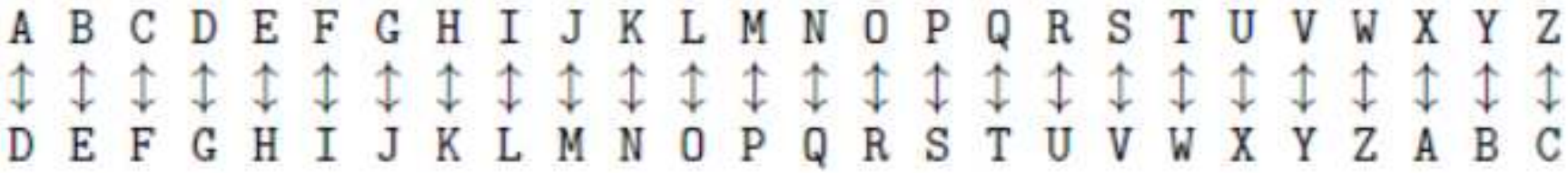
Cannot use $f(x) = c$ for a constant c: Why?
Next simplest function is $f(x) = x + c$ for some constant.

Caesar Cipher

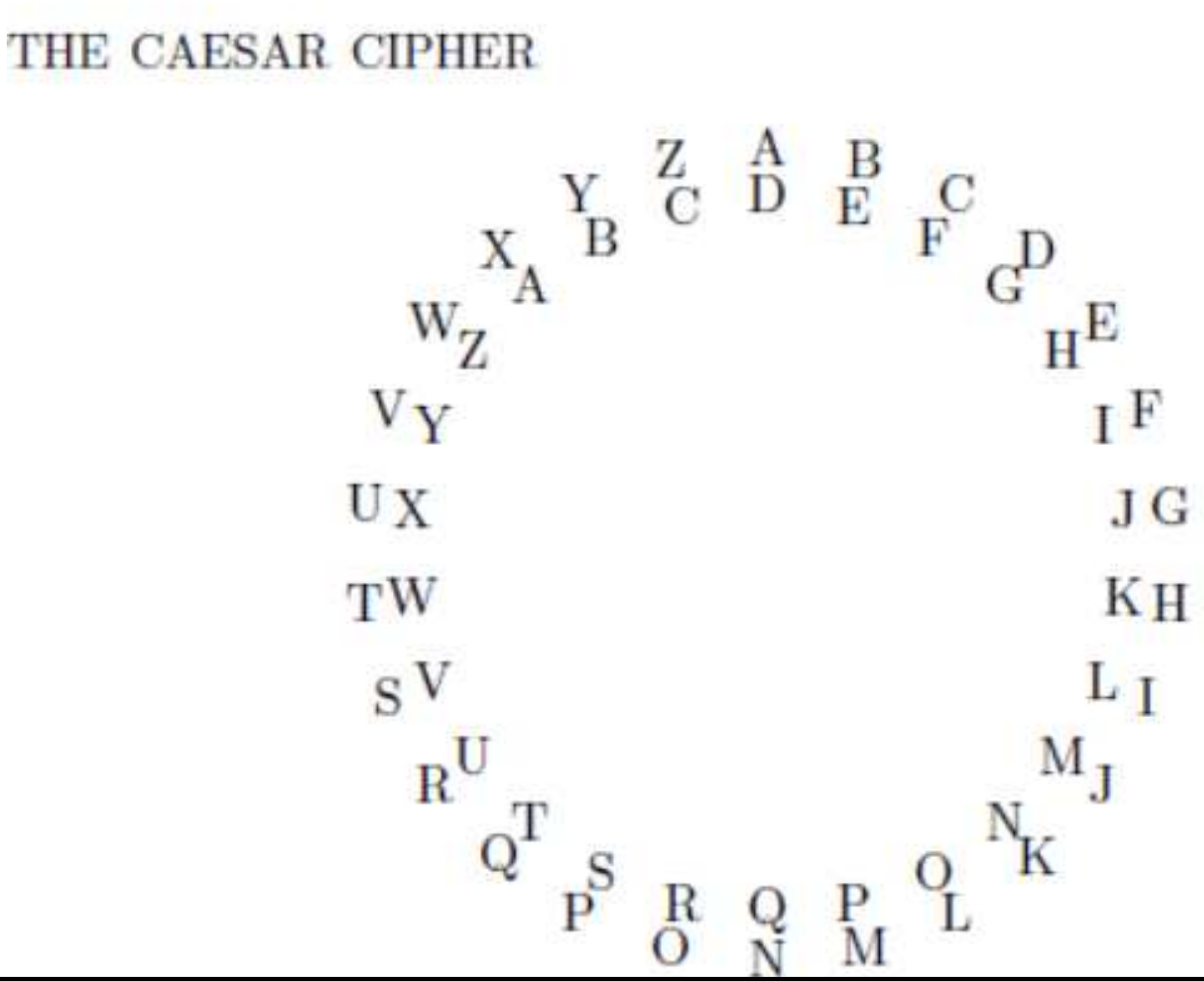
Cannot use $f(x) = c$ for a constant c: Why?
Next simplest function is $f(x) = x + c$ for some constant.

- $c = 3$ is the standard Caesar cipher: how many choices?
- $A \to D, B \to E, \ldots, W \to Z, X \to A, Y \to B, Z \to C$ (clock arithmetic).
- How do we decrypt?

Caesar Cipher: Illustration



Caesar Cipher: Illustration



Caesar Cipher: Example

	M	E	E	T	A	T	T	E	N
	12	4	4	19	0	19	19	4	13
add 3:	15	7	7	22	3	22	22	7	16
	P	H	H	W	D	W	W	H	Q

	P	H	H	W	D	W	W	H	Q
	15	7	7	22	3	22	22	7	16
subtract 3:	12	4	4	19	0	19	19	4	13
	M	E	E	T	A	T	T	E	N

Affine Cipher: $f(x) = ax + b$

Try more complicated function: $f(x) = ax + b$.

What b are possible?

Often best to start simple: take 6 letter alphabet: $\{A, B, C, D, E, F\}$.

To do calculations let $A = 1$, $B = 2$, $\ldots$, $F = 6$.

Do all b work? Do all a work?

Affine Cipher: II: $f(x) = ax + b$

Have $A = 1, B = 2, \ldots, F = 6$, look modulo 6.

Enough to study a and take $b = 0$: $f(x) = ax$.

Affine Cipher: II: $f(x) = ax + b$

Have $A = 1, B = 2, \ldots, F = 6$, look modulo 6.

Enough to study a and take $b = 0$: $f(x) = ax$.

$$\begin{array}{ccccccccccc} A & +1 & \Rightarrow & B & +1 & \Rightarrow & C & +1 & \Rightarrow & D & +1 & \Rightarrow & E & +1 & \Rightarrow & F \\ \updownarrow & & & \updownarrow & & & \updownarrow & & & \updownarrow & & & \updownarrow & & & \updownarrow \\ C & +1 & \Rightarrow & D & +1 & \Rightarrow & E & +1 & \Rightarrow & F & +1 & \Rightarrow & A & +1 & \Rightarrow & B \end{array}$$

$$\begin{array}{ccccccccccc} A & +1 & \Rightarrow & B & +1 & \Rightarrow & C & +1 & \Rightarrow & D & +1 & \Rightarrow & E & +1 & \Rightarrow & F \\ \updownarrow & & & \updownarrow & & & \updownarrow & & & \updownarrow & & & \updownarrow & & & \updownarrow \\ C & +2 & \Rightarrow & E & +2 & \Rightarrow & A & +2 & \Rightarrow & C & +2 & \Rightarrow & E & +2 & \Rightarrow & A \end{array}$$

Affine Cipher: II: $f(x) = ax + b$

Have $A = 1, B = 2, \ldots, F = 6$, look modulo 6.

Enough to study a and take $b = 0$: $f(x) = ax$.

- $a = 1$: works: get $\{1, 2, 3, 4, 5, 6\}$.
- $a = 2$: fails: get $\{2, 4, 6, 2, 4, 6\}$ (thus 4 and 6 will also fail).
- $a = 3$: fails: get $\{3, 6, 3, 6, 3, 6\}$ (and also get again 6 fails).
- $a = 5$: works: get $\{5, 4, 3, 2, 1, 6\}$.

Affine Cipher: II: $f(x) = ax + b$

Have $A = 1, B = 2, \ldots, F = 6$, look modulo 6.

Enough to study a and take $b = 0$: $f(x) = ax$.

- $a = 1$: works: get $\{1, 2, 3, 4, 5, 6\}$.
- $a = 2$: fails: get $\{2, 4, 6, 2, 4, 6\}$ (thus 4 and 6 will also fail).
- $a = 3$: fails: get $\{3, 6, 3, 6, 3, 6\}$ (and also get again 6 fails).
- $a = 5$: works: get $\{5, 4, 3, 2, 1, 6\}$.

1, 5 work and 2, 3, 4, 6 fail: What's the pattern?

Affine Cipher: III: $f(x) = ax + b$

Alphabet is $A = 1, \ldots, F = 6$.

The a that work are relatively prime to 6....

What do you think works for 26 letter alphabet?

Affine Cipher: III: $f(x) = ax + b$

Alphabet is $A = 1, \ldots, F = 6$.

The a that work are relatively prime to 6....

What do you think works for 26 letter alphabet?

Answer: 1, 3, 5, 7, 9, 11, 15, 17, 19, 21, 23, 25: 12 choices.

Caesar cipher: 26 choices; Affine cipher: $12 \cdot 26 = 312$.

Can we go to higher degree polynomials?

Totient Function

Will use the Euler totient function $\phi(n)$ later.
$\phi(n)$ is the number of numbers from 1 to n that are relatively prime to n.

If p prime, $\phi(p) =$

Totient Function

Will use the Euler totient function $\phi(n)$ later.
$\phi(n)$ is the number of numbers from 1 to n that are relatively prime to n.

If p prime, $\phi(p) = p - 1$.

If $n = p^2$, $\phi(p^2) =$

Totient Function

Will use the Euler totient function $\phi(n)$ later.
$\phi(n)$ is the number of numbers from 1 to n that are relatively prime to n.

If p prime, $\phi(p) = p - 1$.

If $n = p^2$, $\phi(p^2) = p^2 - p$ (lose $p, 2p, 3p, \ldots, p^2$.

If $n = p^3$, $\phi(p^3) =$

Totient Function

Will use the Euler totient function $\phi(n)$ later.
$\phi(n)$ is the number of numbers from 1 to n that are relatively prime to n.

If p prime, $\phi(p) = p - 1$.

If $n = p^2$, $\phi(p^2) = p^2 - p$ (lose $p, 2p, 3p, \ldots, p^2$.

If $n = p^3$, $\phi(p^3) = p^3 - p^2$.

If $n = pq$ are distinct primes then $\phi(pq) =$

Totient Function

Will use the Euler totient function $\phi(n)$ later.
$\phi(n)$ is the number of numbers from 1 to n that are relatively prime to n.

If p prime, $\phi(p) = p - 1$.

If $n = p^2$, $\phi(p^2) = p^2 - p$ (lose $p, 2p, 3p, \ldots, p^2$.

If $n = p^3$, $\phi(p^3) = p^3 - p^2$.

If $n = pq$ are distinct primes then $\phi(pq) = (p-1)(q-1)$:

- Lose $p, 2p, 3p, \ldots, qp$.
- Lose $q, 2q, 3q, \ldots, pq$.
- Double counted pq so add back:
  $pq - q - p + 1 = (p-1)(q-1)$.

Vigenére, and Permutation Ciphers

Vigenére

Issue is always send a letter to same new letter....

Take a keyphrase, write under and use for shifts:

D A D C A N A D D A B E D

A B C A B C A B C A B C A

E C G D C Q B F G B D H E

So D shifted to E, F, G; A shifted to B, C; ....
How secure is this?

Vigenére

How secure is the Vigenére cipher?

- https://www.telegraph.co.uk/news/worldnews/northamerica/usa/8225871/CIA-decodes-Civil-War-message-in-a-bottle-after-147-years.html
- https://owlcation.com/humanities/1863-Siege-of-Vicksburg-Secret-Message-Decoded
- http://archive.boston.com/news/nation/articles/2010/12/26/civil_war_note_finally_deciphered/
- http://cryptiana.web.fc2.com/code/civilwar4.htm
- https://en.wikipedia.org/wiki/Kasiski_examination

Permutation Ciphers

26 choices for A, 25 for B, ...

$26! \approx$

Permutation Ciphers

26 choices for A, 25 for B, ...

$26! \approx 4 \cdot 10^{26}$.

A lot more than affine case!

How secure are these?

Frequency Attacks

1	e	12.58%	14	m	2.56%
2	t	9.09%	15	f	2.35%
3	a	8.00%	16	w	2.22%
4	o	7.59%	17	g	1.98%
5	i	6.92%	18	y	1.90%
6	n	6.90%	19	p	1.80%
7	s	6.34%	20	b	1.54%
8	h	6.24%	21	v	0.98%
9	r	5.96%	22	k	0.74%
10	d	4.32%	23	x	0.18%
11	l	4.06%	24	j	0.15%
12	u	2.84%	25	q	0.12%
13	c	2.58%	26	z	0.08%

TABLE 1. Frequencies of letters in English text, from 9,481 English works from Project Gutenberg; see http://www.cryptograms.org/letter-frequencies.php.

Frequency Attacks

most common bigrams	
1	th
2	he
3	in
4	en
5	nt
6	re
7	er
8	an
9	ti
10	es
11	on
12	at
13	se
14	nd
15	or
16	ar
17	al

common trigrams	
1	the
2	and
3	tha
4	ent
5	ing
6	ion
7	tio
8	for
9	nde
10	has
11	nce
12	edt
13	tis
14	oft
15	sth

Why Primes?

Two Systems

p, q are 200 digit primes, $N = pq$ public, password p **or** q.

X is a 5000 digit random number, password is X.

The more secure system is

Two Systems

p, q are 200 digit primes, $N = pq$ public, password p **or** q.

X is a 5000 digit random number, password is X.

The more secure system is the first (know it when here it, versus have to know).

Two Systems

p, q are 200 digit primes, $N = pq$ public, password p **or** q.

X is a 5000 digit random number, password is X.

The more secure system is the first (know it when here it, versus have to know).

Say every atom in the universe (about 10^{80} such) is a universe, and each atom there is a supercomputer checking 10^{15} items a second. About $3.2 \cdot 10^7$ seconds in a year, check about $3.2 \cdot 10^{182}$ per year.

Universe about 15 billion years old, so in the life of the universe would check about $5 \cdot 10^{192}$. Less than the number of primes to check!

RSA Description
(Rivest, Shamir, and Adleman)

Set-up: Example

Alice always sends to Bob, Charlie or Eve tries to intercept.

Bob does the following (could have b subscripts):

- Secret: $p = 15217$, $q = 17569$, $d = 80998505$.

Set-up: Example

Alice always sends to Bob, Charlie or Eve tries to intercept.

Bob does the following (could have b subscripts):

- Secret: $p = 15217$, $q = 17569$, $d = 80998505$.
- Public: $N = pq = 267347473$, $e = 3141593$.

Set-up: Example

Alice always sends to Bob, Charlie or Eve tries to intercept.

Bob does the following (could have b subscripts):

- Secret: $p = 15217$, $q = 17569$, $d = 80998505$.
- Public: $N = pq = 267347473$, $e = 3141593$.
- Note: $ed = 1 \bmod (p-1)(q-1)$.

Set-up: Example

Alice always sends to Bob, Charlie or Eve tries to intercept.

Bob does the following (could have b subscripts):

- Secret: $p = 15217$, $q = 17569$, $d = 80998505$.
- Public: $N = pq = 267347473$, $e = 3141593$.
- Note: $ed = 1 \bmod (p-1)(q-1)$.
- **Message:** $M = 195632041$**, send** $M^e \bmod N$ **or** $X = 121209473$**.**

Set-up: Example

Alice always sends to Bob, Charlie or Eve tries to intercept.

Bob does the following (could have b subscripts):

- Secret: $p = 15217$, $q = 17569$, $d = 80998505$.
- Public: $N = pq = 267347473$, $e = 3141593$.
- Note: $ed = 1 \bmod (p-1)(q-1)$.
- **Message:** $M = 195632041$**, send** $M^e \bmod N$ **or** $X = 121209473$**.**
- Decrypt: $X^d \bmod N$ or 195632041.

Set-up: Example

Alice always sends to Bob, Charlie or Eve tries to intercept.

Bob does the following (could have b subscripts):

- Secret: $p = 15217$, $q = 17569$, $d = 80998505$.
- Public: $N = pq = 267347473$, $e = 3141593$.
- Note: $ed = 1 \bmod (p-1)(q-1)$.
- **Message:** $M = 195632041$**, send** $M^e \bmod N$ **or** $X = 121209473$**.**
- Decrypt: $X^d \bmod N$ or 195632041.

Imagine receive $\widetilde{X} = 12120948\,3$.
Message 195632041
Decrypts 12114102 8, only two digits are the same!

Implementation Questions

A lot of implementation issues.

- How do we find large primes? How large is large?
- How do we find e and d so that $ed = 1 \bmod (p-1)(q-1)$?
- How do we compute $M^e \bmod N$ efficiently?
- Can Eve determine d from e and N?

Introduction to Cryptography: RSA

Introduction to Cryptography: RSA: Steven J. Miller

http://www.williams.edu/Mathematics/sjmiller/public_html

VCTAL, Burlington, June 20

RSA Description
(Rivest, Shamir, and Adleman)

Set-up: Example

Alice always sends to Bob, Charlie or Eve tries to intercept.

Bob does the following (could have b subscripts):

- Secret: $p = 15217$, $q = 17569$, $d = 80998505$.

Set-up: Example

Alice always sends to Bob, Charlie or Eve tries to intercept.

Bob does the following (could have b subscripts):

- Secret: $p = 15217$, $q = 17569$, $d = 80998505$.
- Public: $N = pq = 267347473$, $e = 3141593$.

Set-up: Example

Alice always sends to Bob, Charlie or Eve tries to intercept.

Bob does the following (could have b subscripts):

- Secret: $p = 15217$, $q = 17569$, $d = 80998505$.
- Public: $N = pq = 267347473$, $e = 3141593$.
- Note: $ed = 1 \bmod (p-1)(q-1)$.

Set-up: Example

Alice always sends to Bob, Charlie or Eve tries to intercept.

Bob does the following (could have b subscripts):

- Secret: $p = 15217$, $q = 17569$, $d = 80998505$.
- Public: $N = pq = 267347473$, $e = 3141593$.
- Note: $ed = 1 \bmod (p-1)(q-1)$.
- **Message:** $M = 195632041$**, send** $M^e \bmod N$ **or** $X = 121209473$**.**

Set-up: Example

Alice always sends to Bob, Charlie or Eve tries to intercept.

Bob does the following (could have b subscripts):

- Secret: $p = 15217$, $q = 17569$, $d = 80998505$.
- Public: $N = pq = 267347473$, $e = 3141593$.
- Note: $ed = 1 \bmod (p-1)(q-1)$.
- **Message:** $M = 195632041$**, send** $M^e \bmod N$ **or** $X = 121209473$**.**
- Decrypt: $X^d \bmod N$ or 195632041.

Set-up: Example

Alice always sends to Bob, Charlie or Eve tries to intercept.

Bob does the following (could have b subscripts):

- Secret: $p = 15217$, $q = 17569$, $d = 80998505$.
- Public: $N = pq = 267347473$, $e = 3141593$.
- Note: $ed = 1 \bmod (p-1)(q-1)$.
- **Message:** $M = 195632041$**, send** $M^e \bmod N$ **or** $X = 121209473$**.**
- Decrypt: $X^d \bmod N$ or 195632041.

Imagine receive $\widetilde{X} = 12120948\,3$.
Message 195632041
Decrypts 12114102 8, only two digits are the same!

Implementation Questions

A lot of implementation issues.

- How do we find large primes? How large is large?
- How do we find e and d so that $ed = 1 \bmod (p-1)(q-1)$?
- How do we compute $M^e \bmod N$ efficiently?
- Can Eve determine d from e and N?

Fermat's little Theorem

Euler totient function

$\phi(n)$ is the number of integers from 1 to n relatively prime to n.

$\phi(p) = p - 1$ and $\phi(pq) = (p-1)(q-1)$ if p, q distinct primes.

Do not need, but $\phi(mn) = \phi(m)\phi(n)$ if $\gcd(m, n) = 1$, and $\phi(p^k) = p^k - p^{k-1}$.

A lot of group theory lurking in the background, only doing what absolutely need.

Fermat's little Theorem

Fermat's little Theorem (FlT)

Let a be relatively prime to n. Then $a^{\phi(n)} = 1 \bmod n$.

Special cases: $a^{p-1} = 1 \bmod p$, $a^{(p-1)(q-1)} = 1 \bmod pq$.

Will only prove these two cases....

Proof of Fermat's little Theorem: $n = p$

Proof: Let $n = p$, let $\gcd(a, p) = 1$.

Consider $1, 2, \ldots, p-1$ and $a, 2a, \ldots, (p-1)a$.

Claim both sets are all residues modulo p.

Proof of Fermat's little Theorem: $n = p$

Proof: Let $n = p$, let $\gcd(a, p) = 1$.

Consider $1, 2, \ldots, p-1$ and $a, 2a, \ldots, (p-1)a$.

Claim both sets are all residues modulo p.

If $ia = ja \bmod p$ then $(i-j)a = 0 \bmod p$ so $i = j \bmod p$.

Proof of Fermat's little Theorem: $n = p$

Proof: Let $n = p$, let $\gcd(a, p) = 1$.

Consider $1, 2, \ldots, p-1$ and $a, 2a, \ldots, (p-1)a$.

Claim both sets are all residues modulo p.

If $ia = ja \bmod p$ then $(i-j)a = 0 \bmod p$ so $i = j \bmod p$.
Thus $(p-1)! = (p-1)!a^{p-1} \bmod p$, so $a^{p-1} = 1 \bmod p$. $\square$

Proof of Fermat's little Theorem: $n = p$

Proof: Let $n = p$, let $\gcd(a, p) = 1$.

Consider $1, 2, \ldots, p-1$ and $a, 2a, \ldots, (p-1)a$.

Claim both sets are all residues modulo p.

If $ia = ja \bmod p$ then $(i-j)a = 0 \bmod p$ so $i = j \bmod p$.
Thus $(p-1)! = (p-1)!a^{p-1} \bmod p$, so $a^{p-1} = 1 \bmod p$. □

Note: General case: $x_1, \ldots, x_{\phi(n)}$ and $ax_1, \ldots, ax_{\phi(n)}$.

Proof of Fermat's little Theorem: $n = pq$

Proof: Let $n = pq$, let $\gcd(a, pq) = 1$.

Proof of Fermat's little Theorem: $n = pq$

Proof: Let $n = pq$, let $\gcd(a, pq) = 1$.

Apply FlT with a^{q-1} and p: $(a^{q-1})^{p-1} = 1 \bmod p$.

Apply FlT with a^{p-1} and q: $(a^{p-1})^{q-1} = 1 \bmod q$.

Proof of Fermat's little Theorem: $n = pq$

Proof: Let $n = pq$, let $\gcd(a, pq) = 1$.

Apply FlT with a^{q-1} and p: $(a^{q-1})^{p-1} = 1 \bmod p$.

Apply FlT with a^{p-1} and q: $(a^{p-1})^{q-1} = 1 \bmod q$.

Thus $a^{(p-1)(q-1)}$ is $1 \bmod p$ and is $1 \bmod q$.

$a^{(p-1)(q-1)} = 1 + \alpha p = 1 + \beta q$.

Proof of Fermat's little Theorem: $n = pq$

Proof: Let $n = pq$, let $\gcd(a, pq) = 1$.

Apply FlT with a^{q-1} and p: $(a^{q-1})^{p-1} = 1 \bmod p$.

Apply FlT with a^{p-1} and q: $(a^{p-1})^{q-1} = 1 \bmod q$.

Thus $a^{(p-1)(q-1)}$ is $1 \bmod p$ and is $1 \bmod q$.

$a^{(p-1)(q-1)} = 1 + \alpha p = 1 + \beta q$.

Thus $\alpha p = \beta q$ so $q|\alpha$ and $p|\beta$, so $a^{(p-1)(q-1)} = 1 \bmod pq$. $\square$

Primality Tests from FlT

If $\gcd(a, n) = 1$ and $a^{n-1} \neq 1 \bmod n$ then n cannot be prime.

If equalled 1 then n ***might be*** prime.

Primality Tests from FlT

If $\gcd(a, n) = 1$ and $a^{n-1} \neq 1 \bmod n$ then n cannot be prime.

If equalled 1 then n ***might be*** prime.

- If can take high powers, very fast!
- Can suggest candidate primes, and then use better, slower test for certainty.
- Carmichael numbers: Composites that are never rejected: 561, 1105, 1729, 2465, 2821, 6601, 8911, 10585, 15841, 29341, ... (OEIS A002997).

Fast Multiplication

Cost of Standard Polynomial Evaluation

Multiplication far more expensive than addition....

$f(x) = 3x^5 - 8x^4 + 7x^3 + 6x^2 - 9x + 2$: Cost is
$5 + 4 + 3 + 2 + 1 + 0 = 15$ multiplications.

These are triangle numbers: degree d have $d(d+1)/2$.

Cost of Standard Polynomial Evaluation

Multiplication far more expensive than addition....

$f(x) = 3x^5 - 8x^4 + 7x^3 + 6x^2 - 9x + 2$: Cost is
$5 + 4 + 3 + 2 + 1 + 0 = 15$ multiplications.

These are triangle numbers: degree d have $d(d+1)/2$.

$$\begin{aligned} S(d) &= 1 + 2 + \cdots + d \\ S(d) &= d + (d-1) + \cdots 1 \end{aligned}$$

Cost of Standard Polynomial Evaluation

Multiplication far more expensive than addition....

$f(x) = 3x^5 - 8x^4 + 7x^3 + 6x^2 - 9x + 2$: Cost is
$5 + 4 + 3 + 2 + 1 + 0 = 15$ multiplications.

These are triangle numbers: degree d have $d(d+1)/2$.

$$\begin{aligned} S(d) &= 1 + 2 + \cdots + d \\ S(d) &= d + (d-1) + \cdots 1 \end{aligned}$$

Thus $2S(d) = d \cdot (d+1)$ and claim follows.

Horner's Algorithm

$f(x) = 3x^5 - 8x^4 + 7x^3 + 6x^2 - 9x + 2$: Cost is $5 + 4 + 3 + 2 + 1 + 0 = 15$ multiplications.

Horner's algorithm:

$$\Big(\big(\big((3x - 8)x + 7\big)x + 6\big)x - 9\Big)x + 2.$$

Horner's Algorithm

$f(x) = 3x^5 - 8x^4 + 7x^3 + 6x^2 - 9x + 2$: Cost is $5 + 4 + 3 + 2 + 1 + 0 = 15$ multiplications.

Horner's algorithm:

$$\Big(\Big(\big((3x - 8)x + 7\big)x + 6\Big)x - 9\Big)x + 2.$$

Cost is degree d multiplications!

Useful also in fractal plotting.... Shows can often do common tasks faster.

Fast Multiplication

Horner is best in general, but maybe for special polynomials can do better?

Try polynomials of the form $f(x) =$

Fast Multiplication

Horner is best in general, but maybe for special polynomials can do better?

Try polynomials of the form $f(x) = x^n$.

Write n in binary: Say $n = 100 = 64 + 32 + 4 = 1100100_2$.

Fast Multiplication

Horner is best in general, but maybe for special polynomials can do better?

Try polynomials of the form $f(x) = x^n$.

Write n in binary: Say $n = 100 = 64 + 32 + 4 = 1100100_2$.

$$\begin{aligned} x \cdot x &= x^2 \\ x^2 \cdot x^2 &= x^4 \\ x^4 \cdot x^4 &= x^8 \\ x^8 \cdot x^8 &= x^{16} \\ x^{16} \cdot x^{16} &= x^{32} \\ x^{32} \cdot x^{32} &= x^{64} \end{aligned}$$

Fast Multiplication

Horner is best in general, but maybe for special polynomials can do better?

Try polynomials of the form $f(x) = x^n$.

Write n in binary: Say $n = 100 = 64 + 32 + 4 = 1100100_2$.

$$\begin{aligned} x \cdot x &= x^2 \\ x^2 \cdot x^2 &= x^4 \\ x^4 \cdot x^4 &= x^8 \\ x^8 \cdot x^8 &= x^{16} \\ x^{16} \cdot x^{16} &= x^{32} \\ x^{32} \cdot x^{32} &= x^{64} \end{aligned}$$

Fast Multiplication

Horner is best in general, but maybe for special polynomials can do better?

Try polynomials of the form $f(x) = x^n$.

Write n in binary: Say $n = 100 = 64 + 32 + 4 = 1100100_2$.

$$\begin{aligned} x \cdot x &= x^2 \\ x^2 \cdot x^2 &= x^4 \\ x^4 \cdot x^4 &= x^8 \\ x^8 \cdot x^8 &= x^{16} \\ x^{16} \cdot x^{16} &= x^{32} \\ x^{32} \cdot x^{32} &= x^{64} \end{aligned}$$

Fast Multiplication

Horner is best in general, but maybe for special polynomials can do better?

Try polynomials of the form $f(x) = x^n$.

Write n in binary: Say $n = 100 = 64 + 32 + 4 = 1100100_2$.

$$\begin{aligned} x \cdot x &= x^2 \\ x^2 \cdot x^2 &= x^4 \\ x^4 \cdot x^4 &= x^8 \\ x^8 \cdot x^8 &= x^{16} \\ x^{16} \cdot x^{16} &= x^{32} \\ x^{32} \cdot x^{32} &= x^{64} \end{aligned}$$

Recap

Horner takes us from order d^2 to order d.

Fast multiplication takes us to order $\log_2 d$, but only for special polynomials; these though are the ones used in RSA!

Euclidean Algorithm

Preliminaries

Input x, y with $y > x$.

Goals: find $\gcd(x, y)$, find a, b so that $ax + by = \gcd(x, y)$.

Lot of ways to go: non-constructive proofs of a, b but need values; Euclidean algorithm is *very* fast.

Euclidean Algorithm

Let $r_0 = y, r_1 = x$.

$$r_0 = q_1 r_1 + r_2, \quad 0 \leq r_2 < r_1.$$

Euclidean Algorithm

Let $r_0 = y, r_1 = x$.

$$r_0 = q_1 r_1 + r_2, \quad 0 \leq r_2 < r_1.$$

$$r_1 = q_2 r_2 + r_3, \quad 0 \leq r_3 < r_2.$$

Euclidean Algorithm

Let $r_0 = y, r_1 = x$.

$$r_0 = q_1 r_1 + r_2, \quad 0 \leq r_2 < r_1.$$

$$r_1 = q_2 r_2 + r_3, \quad 0 \leq r_3 < r_2.$$

Continue until....

$$r_n = q_{n+1} r_{n+1} + r_{n+2}, \quad r_{n+2} \in \{0, 1\}.$$

Euclidean Algorithm

Let $r_0 = y, r_1 = x$.

$$r_0 = q_1 r_1 + r_2, \quad 0 \leq r_2 < r_1.$$

$$r_1 = q_2 r_2 + r_3, \quad 0 \leq r_3 < r_2.$$

Continue until....

$$r_n = q_{n+1} r_{n+1} + r_{n+2}, \quad r_{n+2} \in \{0, 1\}.$$

Note $\gcd(r_0, r_1) = \gcd(r_1, r_2) = \gcd(r_2, r_3), \ldots$.

Euclidean Algorithm

Let $r_0 = y, r_1 = x$.

$$r_0 = q_1 r_1 + r_2, \quad 0 \leq r_2 < r_1.$$

$$r_1 = q_2 r_2 + r_3, \quad 0 \leq r_3 < r_2.$$

Continue until....

$$r_n = q_{n+1} r_{n+1} + r_{n+2}, \quad r_{n+2} \in \{0, 1\}.$$

Note $\gcd(r_0, r_1) = \gcd(r_1, r_2) = \gcd(r_2, r_3), \ldots$.

Can 'climb upwards' to get a, b such that $ax + by = \gcd(x, y)$.

Implementing RSA

Implementing RSA

- Choose large primes p, q: Use FlT to get candidates.... If random choice is composite implement by 2 and try again.
- Use Euclidean algorithm to find e, d such that $ed = 1 \bmod \phi(pq)$; choose a candidate e randomly and apply Euclidean algorithm to $x = e$ and $y = (p-1)(q-1)$. If gcd equals 1 win, else increase e by 2 and try again.
- Use fast multiplication to compute $M^e \bmod pq$ efficiently, and also for that to the d^{th} power.