Encryption and Keywords Using Large Prime Numbers

Larry Wittig
Lexington Computer and Technology Group
October 2018

Overview of this presentation

This presentation is based in part on The Code Book by Simon Singh:

- Simple translation and substitution encryption
- Vigenere (polyalphabetic) encryption up to the Enigma

*** The advent of electronic computers and the Arpanet/Internet ***

- Modern Encryption – DES, AES
- Diffie-Hellman Key Exchange
- RSA (public key – private key stuff)

Besides The Code Book I also consulted Wikipedia (quite often) and other Web pages. I generally tried to give the links.

Discussed in Singh's book but not covered here in any detail:

- Mary Queen of Scotts vs. QE1
- Beale treasure papers
- Zimmerman telegraph (WWI)
- Deciphering lost languages and ancient scripts
- Code breaking
- Navajo code talkers during WW II
- Quantum computers

Extremely Short History of encryption pre Electronic Computers

- Earliest known encryption dates back to Egypt about 2000 BCE, and a simple **translation** encryption scheme that was used by Julius Caesar is well documented.
- **Substitution** (random) ciphers were still used to the late middle ages when they were replaced by **polyalphabetic** ciphers. They were used for both personal and state secretes.
- Before and during World War II, mechanical and **electro-mechanical polyalphabetic** machines were in wide use (e.g. the German Enigma machine).

There are thousands of other “non-computer” or “paper and pencil” ciphers. These are just the most important ones.

Simple Shift/Translational Substitution Encryption

- A SHIFT of 3 letters is know as a Caesar shift cipher.

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

- HELLO becomes KHOOR
- A variation on this is to shift the alphabet by a KEYWORD with the rest of the letters translated but without reusing any of the letters in the KEYWORD.

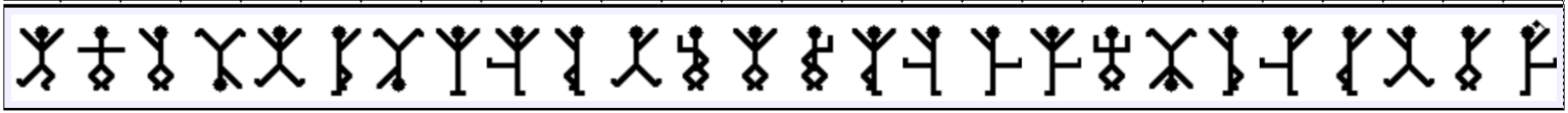
A	B	C	D	E	F	G	H	I	J	K	L	M	N	O	P	Q	R	S	T	U	V	W	X	Y	Z
K	**E**	**Y**	**W**	**O**	**R**	**D**	A	B	C	F	G	H	I	J	L	M	N	P	Q	S	T	U	V	X	Z

- Capitalization, punctuation and the space between words is generally omitted, or the encrypted text may group the letters in sets of N characters.
- Note that on a very elementary level, an eavesdropper (Eve) could know the cipher scheme but would find decoding difficult it without the keyword.

Substitution Encryption

(beyond simple translation)

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



- In this case a random letter or a symbol is substituted for each letter.
- Substitution encryptions can be easily broken with “frequency analysis” especially if the message is long. This was known by Arab scholars as early as 800 (C.E.)
- Homophonic cyphers use more than one code letter/symbol per plaintext letter as a way to minimize the effectiveness of frequency analysis
- Substitution encryptions can be made even more complicated by substituting for pairs of letters or for syllables. (See Singh on the Great Cipher on Louis XIV.)

Letter	Relative frequency in the English language
e	12.702%
t	9.056%
a	8.167%
o	7.507%
i	6.966%
n	6.749%
s	6.327%
h	6.094%
r	5.987%
d	4.253%
l	4.025%
c	2.782%
u	2.758%
m	2.406%
w	2.360%
f	2.228%
g	2.015%
y	1.974%
p	1.929%
b	1.492%
v	0.978%
k	0.772%
j	0.153%
x	0.150%
q	0.095%
z	0.074%

Polyalphabetic Cipher
a.k.a. Vigenere Cipher

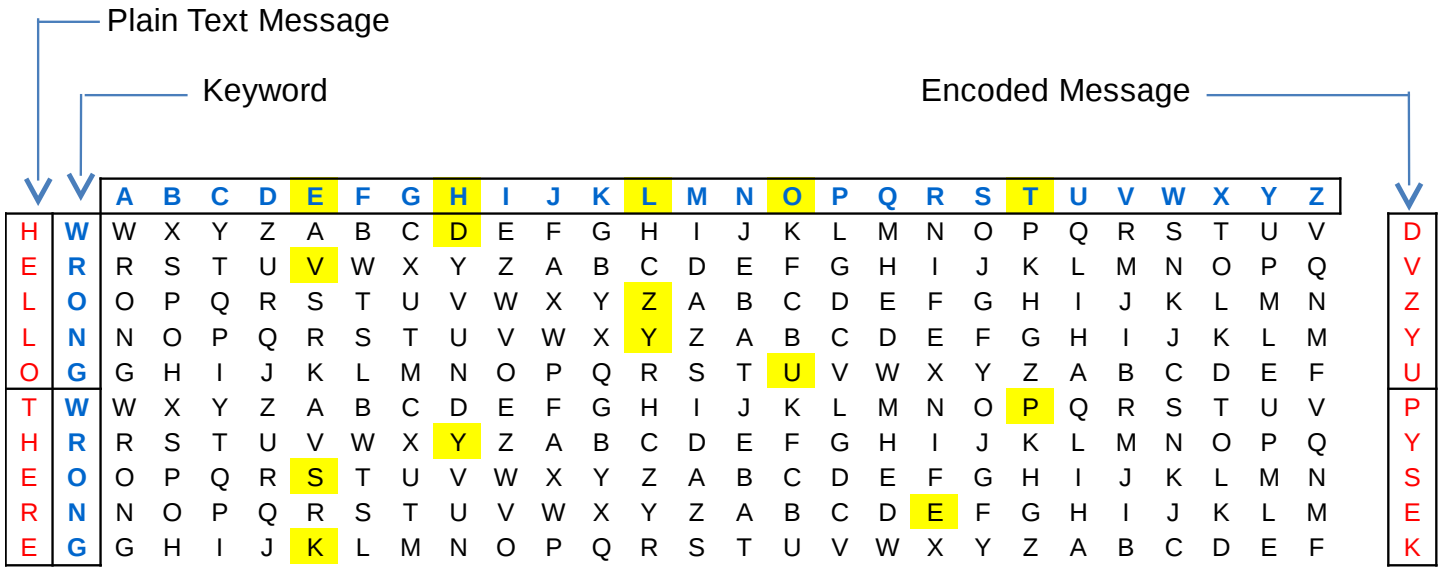
- Singh: the Vigenere cipher [was named] in honor of the man who developed it into its final form in the 16th century.

- Like substitution cipher except the substitution alphabet changes after each of the first N characters, and then the scheme repeats. It makes simple frequency analysis more difficult.

- **REQUIRES A KEYWORD** – this is an important new and enduring aspect, so security of the keyword becomes extremely important – the Vigenere square scheme could be public and secrecy depended solely on the keyword.

- See the excel worksheet in a program called Code Frequencies about 90% of the way down http://www.mathsatwhitehaven.com/mathsclub/codes/codeindex.htm#extools

- Around 1850 Charles Babbage showed that frequency analysis was still able to solve polyalphabetic ciphers if they were of sufficient length.



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

Note that E encryptes to three different letters (V, S & K) minimizing the effectiveness of frequency analys

Cipher Disk

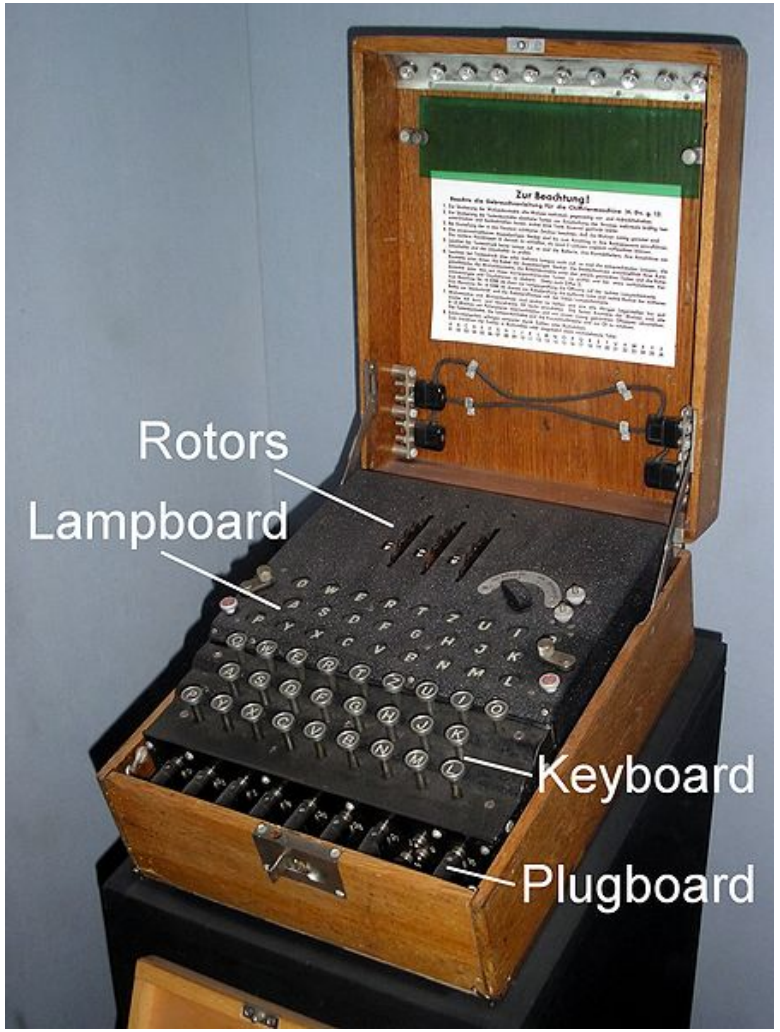
(As opposed to a hard disk drive that's been encrypted)

- Described by Alberti in 1467
- The inner disk and outer ring could rotate w.r.t. each other.
- The letters on the inner disk could be in alphabetical order for a simple translation cipher.
- More generally they could be scrambled for a substitution cipher.
- Alice and Bob would both set their disks up by aligning a given inner disk character with a given outer ring character – this means they have to exchange some information ahead of time.
- If it was rotated after each character by some agreed upon keyword scheme it could be used to handle polyalphabetic encryption.

The cypher disk shown here is dated 1893, and is similar ones were used in the US civil war.

German Enigma Machine

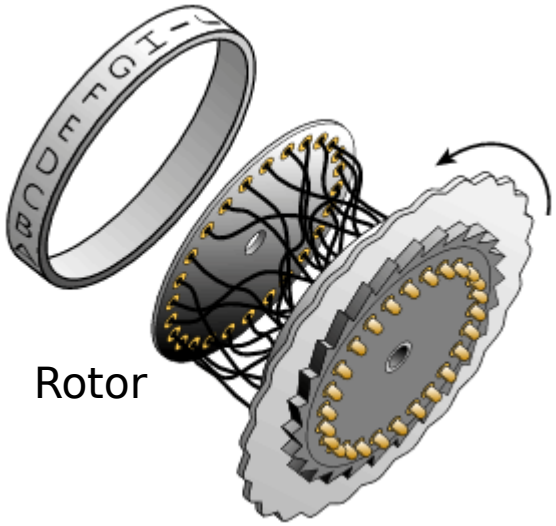
- Developed in 1920's by Arthur Scherbius based patents starting in 1918. There were multiple versions of the Enigma, and other countries had similar machines.
- Based on automating a polyalphabetic cipher plus a patch board and a reflector.
- The machine is set-up on both ends per an agreed upon keyword.
- Requires distribution of keywords – WW ll German would issue a code book for 30 days, longer for U-boats
- Thought to impregnable based on the large number of set-up configurations.



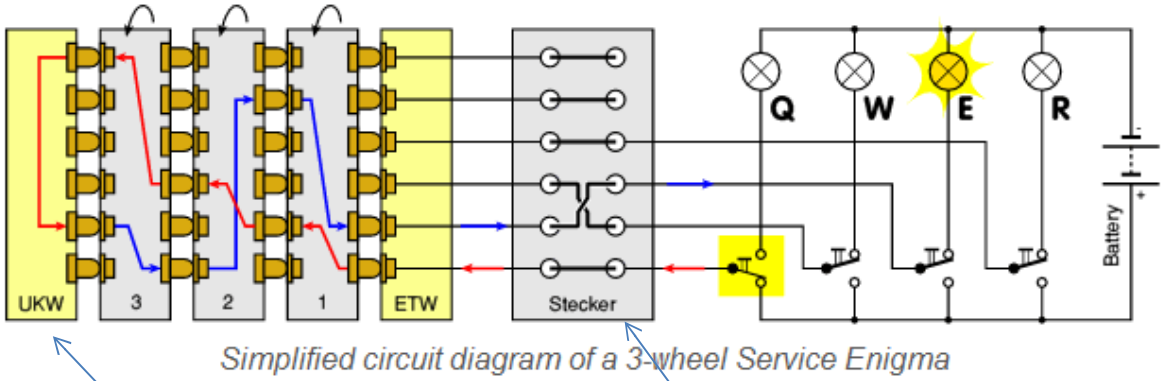
Picture from Wikipedia

German Enigma Machine

How it works: http://www.cryptomuseum.com/crypto/enigma/working.htm



Rotor



Simplified circuit diagram of a 3-wheel Service Enigma

Reflector
Did not Rotate

Plug board

- After a letter was coded the first rotor would index $^{1}/_{26}^{th}$ of a rev, and after 1 rev the second rotor would index, etc. The reflector was stationary.
- **How it works video**: http://www.youtube.com/watch?v=elYw4Ve4F-I
- Computer simulators: http://enigmaco.de/enigma/enigma.html and http://users.telenet.be/d.rijmenants/index.htm

Brief post-WWII overview of encryption

Electronic computers drastically changed what could be done:

- IBM developed **Lucifer** (~1970). Under NBS Lucifer became **DES** (Data Encryption Standard) in 1976 and this was superseded by **AES** (Advanced Encryption Standard) under NIST in 2001.
- With the start of the Arpanet/Internet it became clear that private encryption was important if not necessary. It's no longer just the government that needed encryption.
- In the early 70's this has lead to **Diffie-Hellman Key Exchanges (DHE)** and subsequently what is often referred to **public key** or **RSA** encryption. Both of these are widely used today.
- There has been an ongoing battle between private encryption concerns and **NSA** (National Security Agency, a.k.a. No Such Agency) NSA wants to limit the strength of private encryption software.

Electronic computers also made it easy to crack most pre ~1950 ciphers. One could argue that the machines used by the Brits to crack the Enigma were computers.

Enter ASCII conversion

- If your going to do encryption on a computer you need to first change your text to binary, and the encryption is done on the binary text.
- Be sure to drink your Ovaltine, in 8-bit ASCII becomes:

  01000010 01100101 00100000 01110011 01110101 01110010 01100101 00100000 01110100
  01101111 00100000 01100100 01110010 01101001 01101110 01101011 00100000 01111001
  01101111 01110101 01110010 00100000 01001111 01110110 01100001 01101100 01110100
  01101001 01101110 01100101 0101110

- Note the 8-fold increase in characters. (If 8-bit ASCII is used.)
- This is a short sentence, imagine the number generated by a few paragraphs of plain text. Generally spaces and punctuation is not encryption.

Electronic computer encryption
The turning point

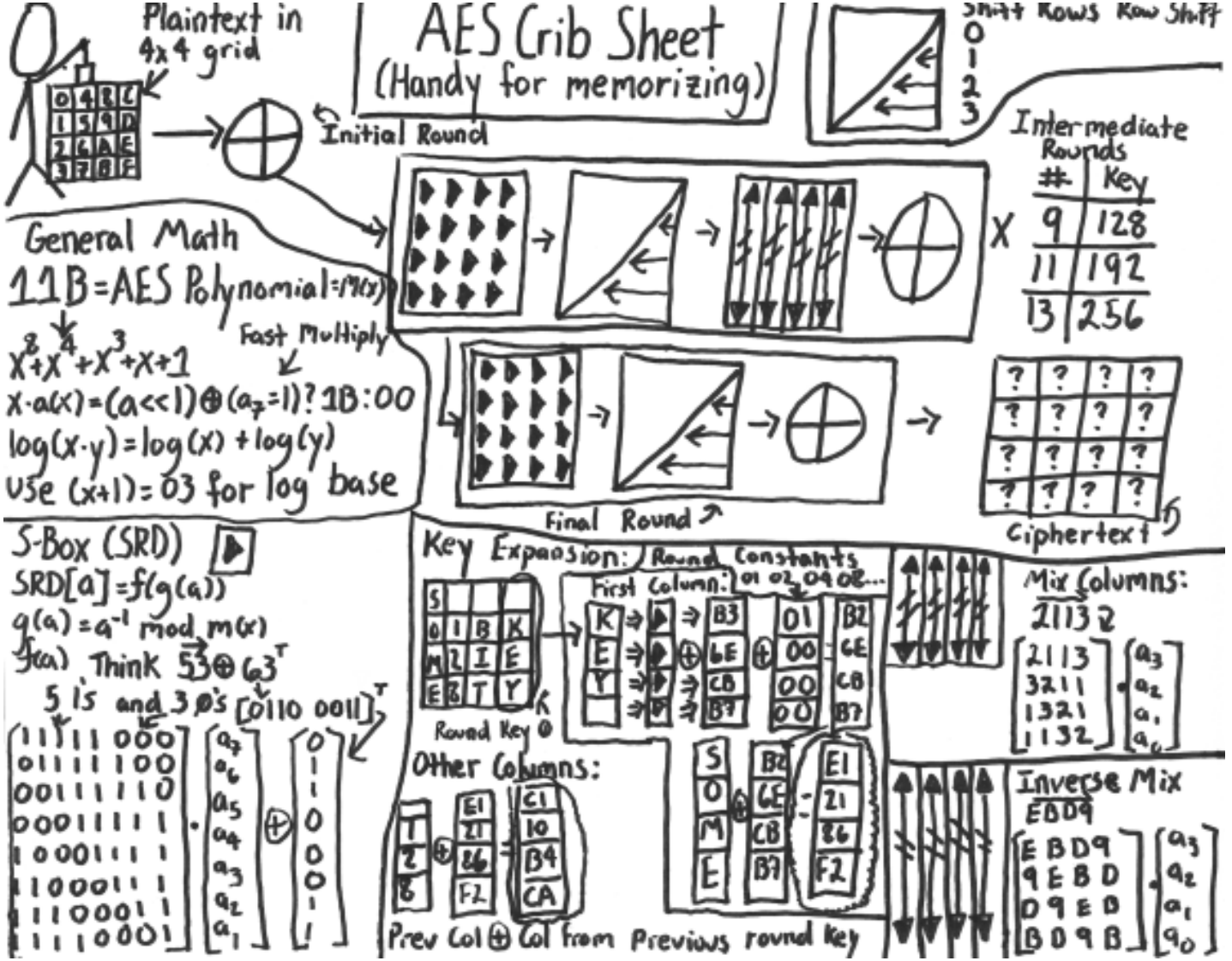
- In the early 70's Horst Feistel et.al at IBM developed a block cypher called **Lucifer** which the NBS turned into **DES** (Data Encryption Standard) in 1975. It evolved into Triple DES. In **2001** it was superseded by the **AES** (Advanced Encryption Standard) by NTIS (formerly NBS). Apple uses a 256-bit AES chip on iPADs, and iPhones.

- There is a sorted history of **NSA** trying to keep public encryption standards from becoming too effective – they wanted to keep them at the point where only they could break them. Mostly this was a fight over keyword lengths, and one of the major changes from DES to AES is the keyword length.

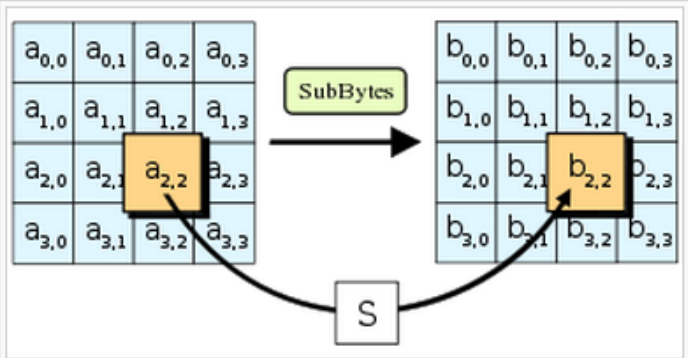
- The Lucifer, DES and AES algorithms are block ciphers, that is they **work on blocks of bits of data by doing cutting, shifting, and substitution**.

- AES was published by NIST as FIPS PUB 197 in November 2001.

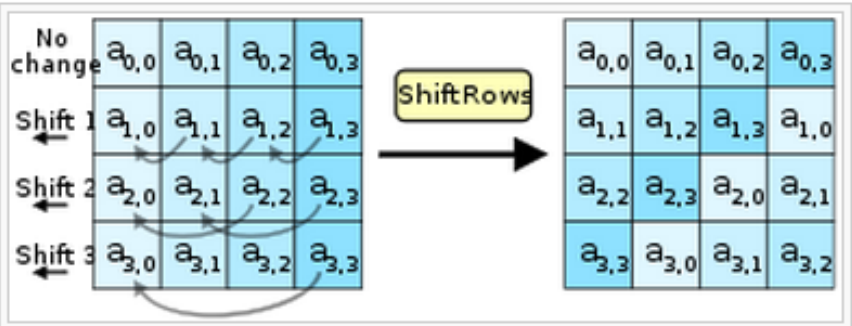
AES (Rijndael) Encryption



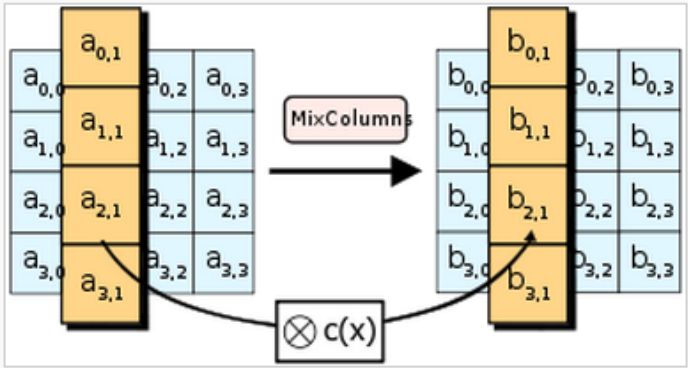
AES (from Wikipedia)



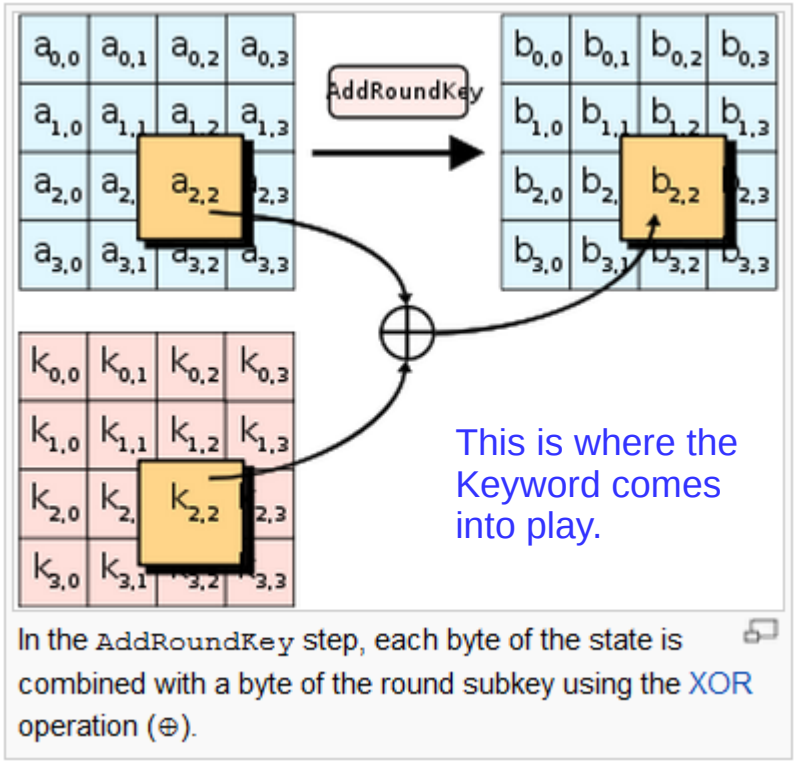
In the `SubBytes` step, each byte in the state is replaced with its entry in a fixed 8-bit lookup table, S; $b_{ij} = S(a_{ij})$.



In the `ShiftRows` step, bytes in each row of the state are shifted cyclically to the left. The number of places each byte is shifted differs for each row.



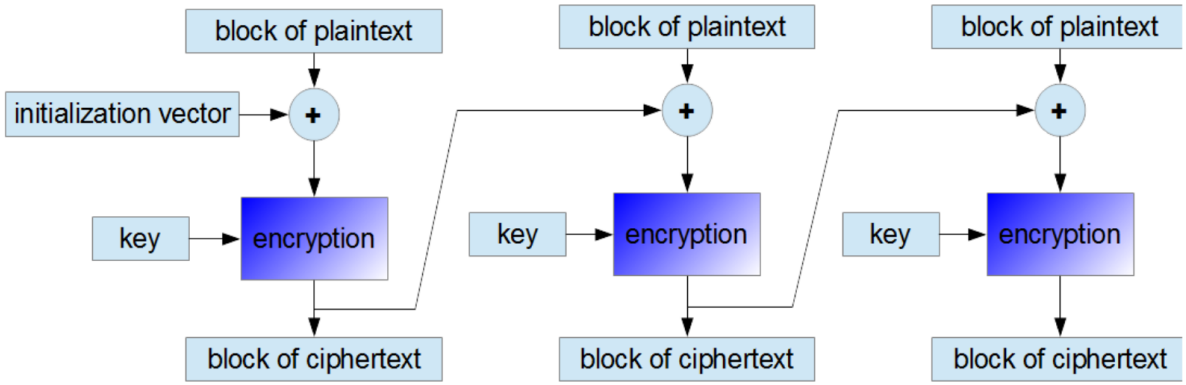
In the `MixColumns` step, each column of the state is multiplied with a fixed polynomial $c(x)$.



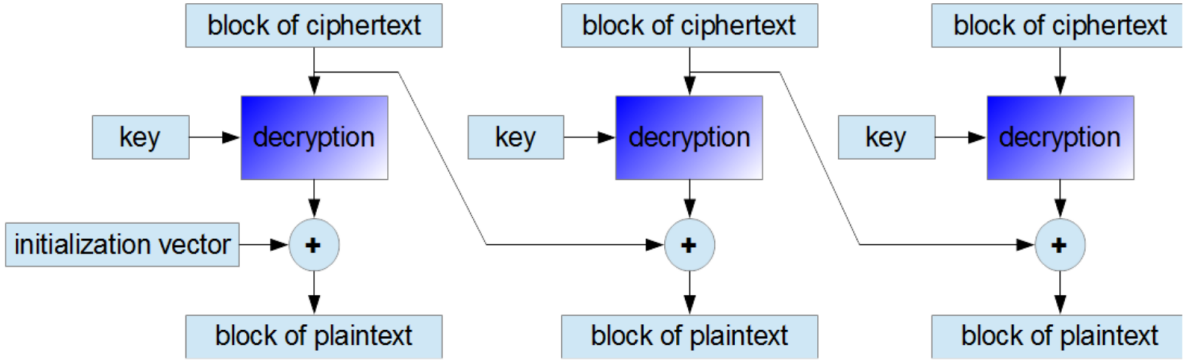
In the `AddRoundKey` step, each byte of the state is combined with a byte of the round subkey using the XOR operation (⊕).

AES is a subset of the Rijndael block cipher developed by two Belgian cryptographers, Vincent Rijmen and Joan Daemen. The AES block size has a maximum of 128 bits, and a key size of 128, 192 or 256 bits. A “256 bit key”, that allows 2^{256} possible keys.

AES Modes of Operation: A mode of operation describes how to repeatedly apply a cipher's single-block operation to securely transform amounts of data larger than a block. Cipher Block Chaining shown below.



Encryption in the CBC mode



Decryption in the CBC mode

Other encryption methods and keyword transmittal

- A number of other encryption that have emerged since AES (~2001).
- These include: DES, Double-DES, Triple-DES, Blowfish (Two fish, Three Fish),Pretty good privacy, Skipjack, and elliptical curves (my favorite name is is *twisted Edwards curves*).
- Most of these secure are very secure and probably more efficient, **but AES is the de facto standard.** This may be because CPU manufactures modified their chips to facilitate AES calculations. iPhones do AES(256) encryption.
- Encryption appears to be set for a while – at least until quantum computers take hold. NIST presently has a competition running to find a replacement for AES that is quantum computer resistant.

Next we will consider:

- Public/Private Key transmission, and
- RSA which can be used for both key transmission and encryption.

Public & Private Keys are based on Prime Numbers, Exponential Powers, and Modular Math

- A prime number (or a prime) is a natural number greater than 1 that has no positive divisors other than 1 and itself. Two is the only even prime. Numbers that are not primes can be factored into primes starting with small primes and working your way up: $315 = 3*3*5*7 = 3^2*5*7$.
- Primes: 2, 3, 5, 7, 11, 13, 17, 19, 23, 29, 31, 37, 41, 43, 47, 53, 59, 61, 67, 71, 73, 79, 83, 89, 97, 101, 103, 107, 109, 113, 127, 131, ...
- The are theorems that make it relative easy to determine if a number is prime – so to generate a large prime the approach is to generate a large random number and then to check to see if it is prime.
- The largest known prime number (as of July 2018) is $2^{77,232,917} - 1$, a number with 23,249,425 digits. It was found by the Great Internet Mersenne Prime Search (GIMPS) in 2017.

Key Exchange also used exponential powers of large numbers:

- $\mathbf{a^b}$ means you multiply **a** by itself **b** times. That is $5^6 = 5*5*5*5*5*5$ and $10{,}237^{13{,}379}$ (which are small number compare to what is generally used) would contain ~54,000 characters.
- $\mathbf{a^{bc}} = \mathbf{(a^b)^c}$ means you multiply **a** by itself **b*c** times, or $\mathbf{a^b}$ by itself **c** times.

Modular Arithmetic

- Asymmetric functions are ones where it is very easy to go one way, but very hard to reverse (not impossible but say not in the life of the universe)
- **a(mod b)** means you divide a by b and the answer is the remainder
- For example 2130(mod 1200) is 930 as for a 24 hr. clock, or 7(mod5)=2
- Modular arithmetic was studied by many great mathematicians including **Euler** and **Fermat** who developed theorems that help make computations easier.
- Modular arithmetic :
  - Gives a very erratic answers for small changes in the input – this makes it extremely hard to go reverse the process.
  - Makes it possible to handle very large numbers – w/o modular math the numbers generated by public/private encryption would be too big to even be stored by any computer.

x	0	1	2	3	4	5	6	7	8
21^X	1	21	441	9261	194481	4084101	85766121	1801088541	37822859361
$21^X \pmod{43}$	1	21	11	16	35	4	41	1	21

Diffie-Hellman Key Exchange

- Whitfield Diffie (MIT ’65 and MITRE) became interested in security and foresaw the need for it on the emerging ARPANet and what was to come.
- He moved to Sanford to work on encryption with Martin Hellman and subsequently Ralph Merkle.
- Up to this time encryption was mostly done with two-way (symmetric*) algorithms and they focused on **one-way (asymmetric^) algorithms.**
- To do this they used **modular arithmetic.**
- This procedure allows Alice and Bob the establish a common secret key even when Eve sees the messages they exchange – DHE is susceptible to “man-in-middle attacks” but that can be overcome with a digital signatures.
- Internet TLS protocol uses DHE – this what makes http https.

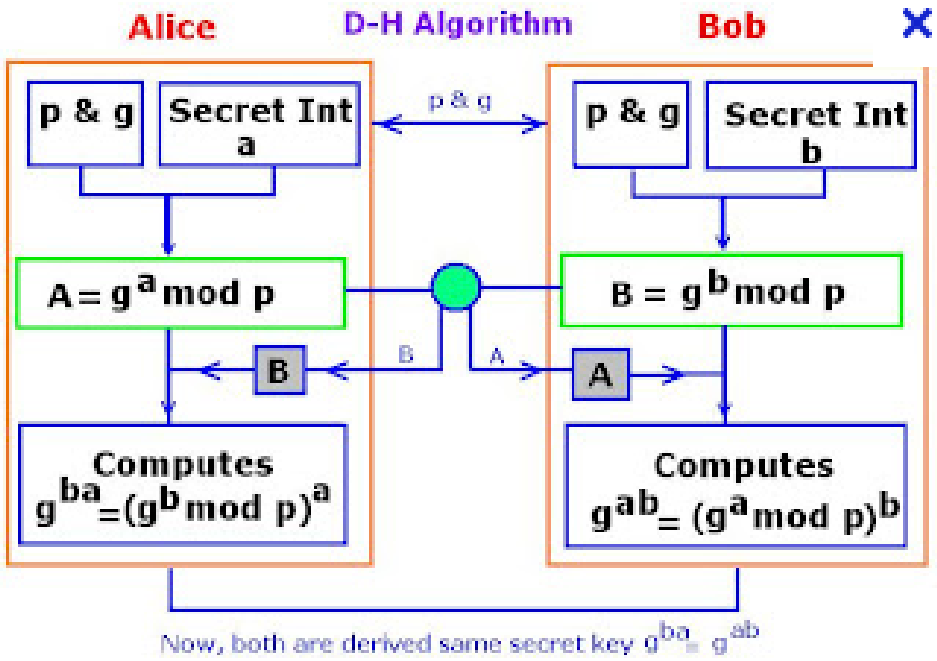
* A symmetric encryption can be decrypted by reversing the steps – order is important

^ An asymmetric encryption is considerably harder to decrypt and can not be decrypted by reversing the encryption steps – this is were large prime numbers are involved

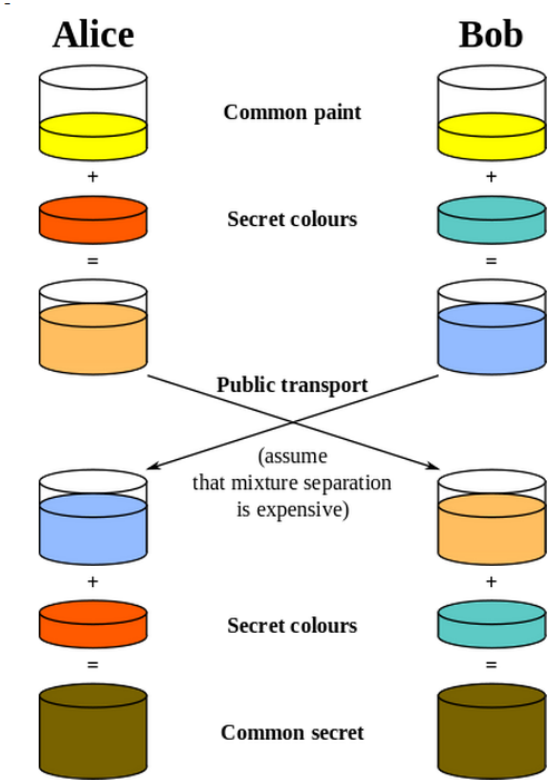
Diffie-Hellman Key Exchange

(Better name: Diffie-Hellman-Merkle key agreement protocol)

Modular Math Exchange



Paint analogy



- Asymmetric math is what allows Alice & Bob to freely exchange values even if Eve is listening
- Internet TLS protocol uses DHE – this what makes http http**s**

Diffie -- Hellman Example

Non-secret values in blue and secret values in red.

1. Alice and Bob agree to use a prime number modulus of $p=23$ and a base of $g=5$ (a primitive root modulo p). The larger the numbers the harder it will be to crack.
2. Alice chooses a secret integer $a=4$, computes $A=g^a \bmod p$, and sends it to Bob. ($A=5^4 \bmod 23=4$)
3. Bob chooses a secret integer $b=3$, computes $B=g^b \bmod p$, and sends it to Alice. ($B=5^3 \bmod 23=10$)
4. Alice computes $s = B^a \bmod p$ ($s=10^4 \bmod 23 = 18$)
5. Bob computes $s = A^b \bmod p$ ($s=4^3 \bmod 23 = 18$)
6. Alice and Bob have now established a shared secret value (the number 18).

D-H Exchange protocol has been expanded to handle multiple parties.

See Kahn video

Public/Private Key Encryption

The Diffie, Hellman and Merkle key agreement protocol had/has some drawbacks:

- Its open to man-in-the-middle attacks – Eve could pretend to be Alice/Bob during the initial contact.
- Inefficient – have to go back & forth – its not as if Alice could simply deciding to send an encrypted message to Bob on the spur of the minute.

Therefore, in addition to developing D-E key exchange, Diffie and Hellman developed the framework for an asymmetric cypher in 1975 (but not the final algorithm). They presented the concept in a paper, but it was **RS&A** who devised with the actual implementation.

DHE is still used extensively, but in most cases RSA is better.

RSA Public Key – Private Key Encryption

- Meanwhile back at MIT three young number theorists decide to take a crack at the D-H suggestion.
- **RSA** – **R**ivest, **S**hamir and **A**dleman – RSA could refer to the people who developed the algorithm, the algorithm itself, or the company they founded (purchased by EMC for $2.1 B in 2006)
- The concept is shown on the next slide per the PGP paper cited earlier (ftp://ftp.pgpi.org/pub/pgp/6.5/docs/english/IntroToCrypto.pdf).

First conventional symmetric encryption:

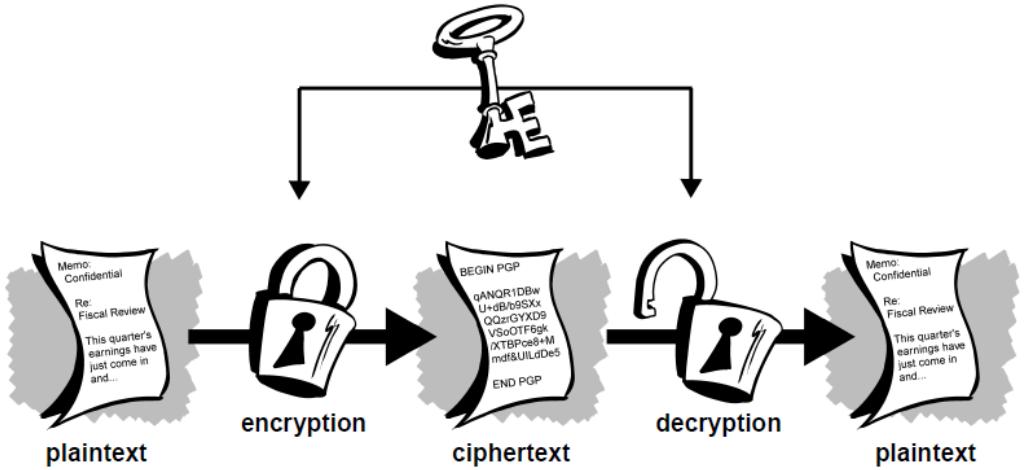


Figure 1-2. Conventional encryption

Public Key Encryption Concept

Bob makes his public key(s) available, generally through a trusted third party, Alice would get one and use it to encrypt a message and send it to Bob. Only Bob could open it using his private key.

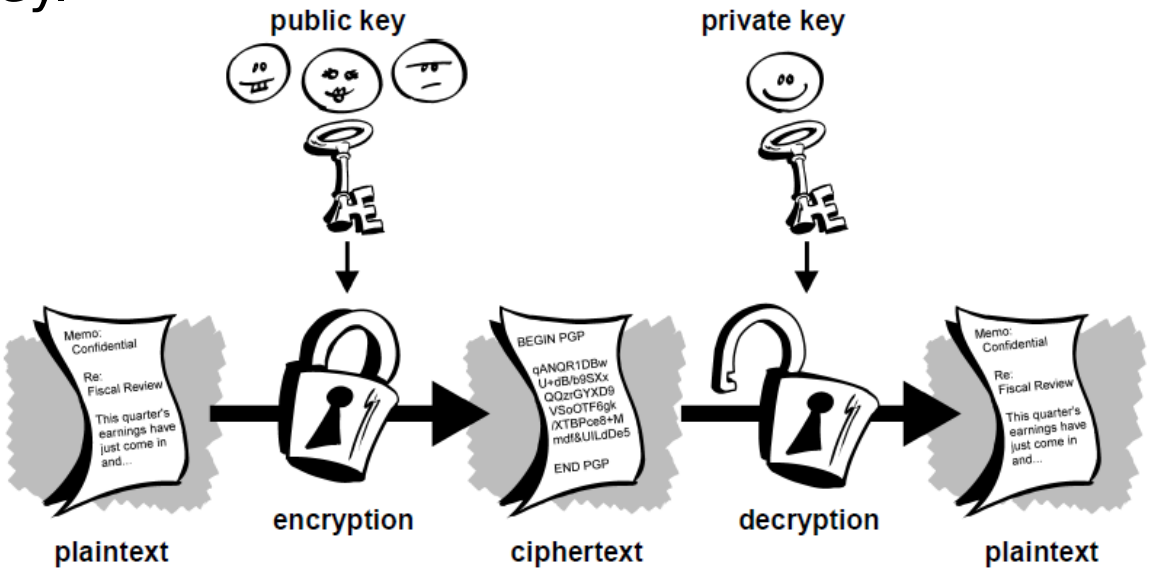


Figure 1-3. Public key encryption

It's a bit like Bob distributing a bunch of open padlocks for others to use to lock up information but that only he could open.

Public & Private Keys using Prime Numbers

- **If two large primes are multiplied together to form a number N, it can be very difficult for someone else to determine the two primes**.

- The other things that PKE use are:
  - Raising the message, expressed as a number, to a large power, for example $3980565198558505351 37^{840613460135916861}$ although there are some shortcut ways to do this with modular math
  - Modular math – this is almost a requirement – without MM the encrypted messages (expressed as a number) gets so big they approach the number of atoms in the known universe

Factoring Products of Large Prime Numbers

- Martin Gardner (Scientific American) published a co-prime value N with 129 digits in 1977 and it took 17 years before the answer was found using a network of 1600 workstations. Now days RSA recommends a value of N with approximately 800 bits.
- The most difficult integers to factor in practice(using existing algorithms) are those that are products of two large primes of similar size, and for this reason these are the integers used in cryptographic applications. The largest such semi prime yet factored was RSA-768**,** a 768-bit number with 232 decimal digits, on December 12, 2009. This factorization was a collaboration of several research institutions, spanning two years and taking the equivalent of almost **2000 years of computing** on a single core 2.2 GHz AMD Opteron. (See PDF document)

Source: http://en.wikipedia.org/wiki/Integer_factorization

Public Key Encryption -- the Math

Red numbers are secret, and blue numbers public.

1. Randomly choose two prime numbers: ***p*** and ***q***.
   - *p* = 127, *q* = 211
2. Compute ***N*** = *p*q, N* is made public
   - *N* = 127 * 211 = 26,797
3. Compute ***N'*** = (*p* – 1)(*q* – 1) a.k.a Euler's totient function often called Φ(N)
   - *N'* = (127 – 1) * (211 – 1) = 26,460
4. Choose ***e*** another prime < *N'* (there are some other less stringent restrictions)
   - *e* = 13,379 e is used for encryption, and d is used for decryption. If e is of the form $2^k + 1$ the computation is faster; a large value of e is more secure.
5. Compute ***d*** as the multiplicative inverse of (*e* mod *N'*) ,*that is [ed(modN') = 1].*

This is important because d depends on N' which depends on p & q the factors of N. That is, to compute d you need to factor N back into p & q.

   - *d* = 11,099 since *e* (13,379) * *d* (11,099) mod *N'* (26,460) = 1
   - This is not too complicated, and there's a procedure for doing it known as the *extended Euclidean algorithm* that dates back to c.300 BCE, and is concerned with finding the Greatest Common Denominator of two numbers. *See the Kahn video*.

Public Key Encryption -- Example

6. Destroy **p, q** & **N'** , keep **e, N** & **d**

7. Use **e** and **N** to encrypt a message using Public key information –that is, send these out as your public key.

8. Use **d** and **N** to decrypt.

Encryption Example – Encrypt a message M which is expressed as a number

- encrypt (M) = $M^e \pmod{N}$, where in this example M is 10,237
- Encrypt(10,237) = $10{,}237^{13.379} \pmod{26797} = 8422$
- 8422 is the encrypted message (R)

Decryption Example

- decrypt(R) = $R^d \pmod{N}$ R is the received encrypted message
- decrypt(8422) = $8422^{11099} \pmod{26797} = 10{,}237$

Public Key Encryption -- Example

6. Destroy **p, q, N'** keep **e, N** & **d**

7. Use **e** and **N** to encrypt a message using Public key information – that is, you send these out as your public key(s).

8. Use **d** to decrypt.

Encryption Example

- encrypt (***M***) = $M^e \pmod N$, where *M* is the message to be sent
- Encrypt(10,237) = $10{,}237^{13{,}379} \pmod{26797} = 8422$
- **R** = 8422 is the encrypted *M*

Note that for this example M^e would have ~54,000 digits, and this example uses small prime numbers.*

Decryption Example

To determine d one needs to know the factors of N.

- decrypt(***R***) = $R^d \pmod N$, where *R* is the received encrypted message
- decrypt(8422) = $8422^{11099} \pmod{26797} = 10{,}237$*

* In modular math there are shortcut ways to handle large powers -- you can not do this on your pocket calculator.

An Alternative History of Public Key Encryption

- The Brits did it first – about three years before RSA, but kept it secret until 1997. Also they didn't know it could be patented
- Bletchley Park's wartime operations were absorbed into the newly formed CESG (Communications Electronic Security Group) and they kept working after the war. CESG is part of the GCHQ (the UK's Government Communications Headquarters). GCHQ ≡ NSA
- Ellis, Cocks and Williamson essentially discovered all the fundamentals of RSA before Diffie, Hellman and Merkle discovered DHE.
- Only several years after RSA was well known was the CESG work acknowledged.
- What seems to be amazing is that not only did come up with the asymmetric public key – private key concept, but also that it was almost identical to RSA
- There is some hint that the U.S. NSA also did work on PKE even earlier in the 60's. But what they did is still classified.
- Show Kahn Video on PKE/RSA