Part I: Crypto

Chapter 2: Crypto Basics

MXDXBVTZWVMXNSPBQXLIMSCCSGXSCJXBOVQXCJZMOJZCVC
TVWJCZAAXZBCSSCJXBQCJZCOJZCNSPOXBXSBTVWJC
JZDXGXXMOZQMSCSCJXBOVQXCJZMOJZCNSPJZHGXXMOSPLH
JZDXZAAXZBXHCSCJXTCSGXSCJXBOVQX

plaintext from Lewis Carroll, *Alice in Wonderland*

The solution is by no means so difficult as you might
be led to imagine from the first hasty inspection of the characters.
These characters, as any one might readily guess,
form a cipher that is to say, they convey a meaning…
Edgar Allan Poe, *The Gold Bug*

Crypto

- **Cryptology** The art and science of making and breaking "secret codes"
- **Cryptography** making "secret codes"
- **Cryptanalysis** breaking "secret codes"
- **Crypto** all of the above (and more)

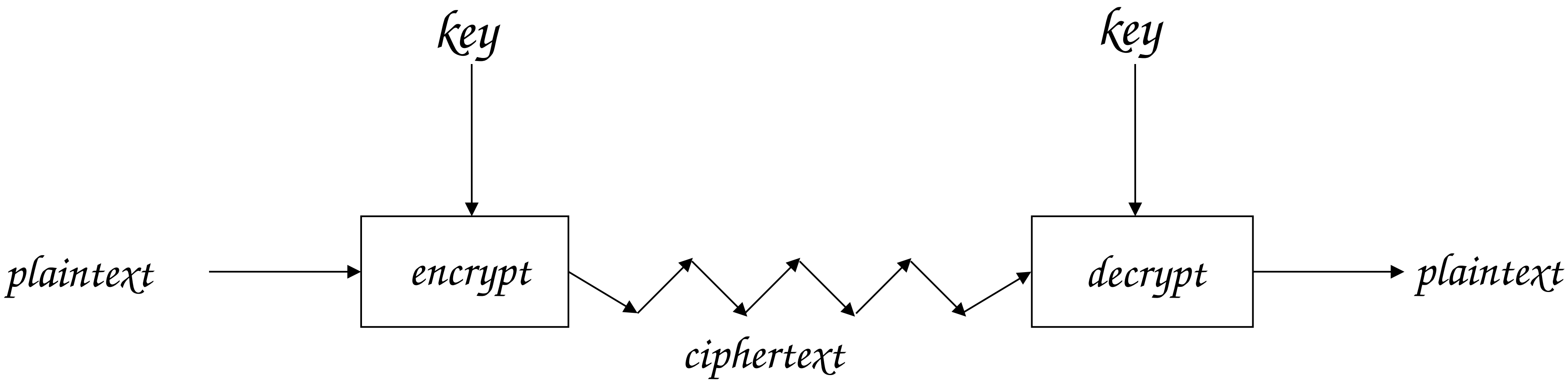
How to Speak Crypto

- A cipher or cryptosystem is used to encrypt the plaintext
- The result of encryption is ciphertext
- We decrypt ciphertext to recover plaintext
- A key is used to configure a cryptosystem
- A symmetric key cryptosystem uses the same key to encrypt as to decrypt
- A public key cryptosystem uses a public key to encrypt and a private key to decrypt

Crypto

- Basic assumptions
  - The system is completely known to the attacker
  - Only the key is secret
  - That is, crypto algorithms are not secret
- This is known as **Kerckhoffs' Principle**
- Why do we make such an assumption?
  - Experience has shown that secret algorithms tend to be weak when exposed
  - Secret algorithms never remain secret
  - Better to find weaknesses beforehand

Crypto as Black Box



A generic view of symmetric key crypto

Simple Substitution

- Plaintext: fourscoreandsevenyearsago
- Key:

Plaintext	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
Ciphertext	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

- Ciphertext: IRXUVFRUHDQGVHYHQBHDUVDJR
- Shift by 3 is "Caesar's cipher"

Ceasar's Cipher Decryption

- Suppose we know a Caesar's cipher is being used:

Plaintext	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
Ciphertext	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

- Given ciphertext:
  VSRQJHEREVTXDUHSDQWV

- Plaintext: spongebobsquarepants

Not-so-Simple Substitution

- Shift by n for some n $\in$ {0,1,2,…,25}
- Then key is n
- Example: key n = 7

Plaintext	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
Ciphertext	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	G

Cryptanalysis I: Try Them All

- A simple substitution (shift by n) is used
  - But the key is unknown
- Given ciphertext: CSYEVIXIVQMREXIH
- How to find the key?
- Only 26 possible keys   try them all!
- **Exhaustive (bruce force) key search**
- Solution: key is n = 4

Simple Substitution: General Case

- In general, simple substitution key can be any **permutation** of letters
  - Not necessarily a shift of the alphabet
- For example

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

- Then $26! > 2^{88}$ possible keys

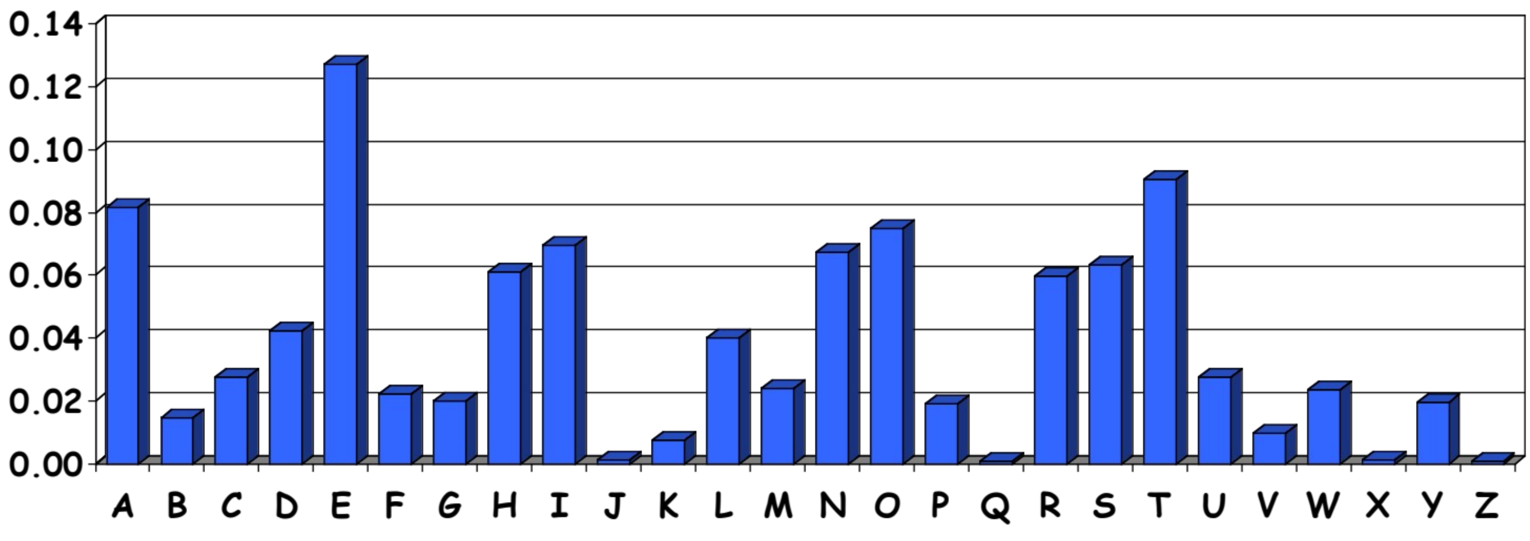
Cryptanalysis II: Be Clever

- We know that a simple substitution is used
- But not necessarily a shift by n
- Find the key given the ciphertext:

PBFPVYFBQXZTYFPBFEQJHDXXQVAPTPQJKTOYQWIPBVWLXTOXBTFXQWAX
BVCXQWAXFQJVWLEQNTOZQGGQLFXQWAKVWLXQWAEBIPBFXFQVXGTVJV
WLBTPQWAEBFPBFHCVLXBQUFEVWLXGDPEQVPQGVPPBFTIXPFHXZHVFAG
FOTHFEFBQUFTDHZBQPOTHXTYFTODXQHFTDPTOGHFQPBQWAQJJTODXQHF
OQPWTBDHHIXQVAPBFZQHCFWPFHPBFIPBQWKFABVYYDZBOTHPBQPQJTQ
OTOGHFQAPBFEQJHDXXQVAVXEBQPEFZBVFOJIWFFACFCCFHQWAUVWFLQ
HGFXVAFXQHFUFHILTTAVWAFFAWTEVOITDHFHFQAITIXPFHXAFQHEFZQ
WGFLVWPTOFFA

Cryptanalysis II

- Cannot try all 2^{88} simple substitution keys
- Can we be more clever?
- English letter frequency counts…



Cryptanalysis II

- Ciphertext:

PBFPVYFBQXZTYFPBFEQJHDXXQVAPTPQJKTOYQWIPBVWLXTOXBTFXQWAXBVCXQWA
XFQJVWLEQNTOZQGGQLFXQWAKVWLXQWAEBIPBFXFQVXGTVJVWLBTPQWAEBFPBF
HCVLXBQUFEVWLXGDPEQVPQGVPPBFTIXPFHXZHVFAGFOTHFEFBQUFTDHZBQPOTH
XTYFTODXQHFTDPTOGHFQPBQWAQJJTODXQHFOQPWTBDHHIXQVAPBFZQHCFWPFHP
BFIPBQWKFABVYYDZBOTHPBQPQJTQOTOGHFQAPBFEQJHDXXQVAVXEBQPEFZBVFOJ
IWFFACFCCFHQWAUVWFLQHGFXVAFXQHFUFHILTTAVWAFFAWTEVOITDHFHFQAITI
XPFHXAFQHEFZQWGFLVWPTOFFA

- Analyze this message using statistics below

Ciphertext frequency counts:

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
21	26	6	10	12	51	10	25	10	9	3	10	0	1	15	28	42	0	0	27	4	24	22	28	6	8

Cryptanalysis: Terminology

- Cryptosystem is **secure** if best know attack is to try all keys
  - Exhaustive key search, that is
- Cryptosystem is **insecure** if **any** shortcut attack is known
- But then insecure cipher might be harder to break than a secure cipher!
  - What the … ?

Logic operations

- And: a and b=1 only when a=b=1
- Or: a 0r b = 0 only when a=b=0
- Not: 1->0, 0->1
- Xor : 1 xor 0 = 0 xor 1 = 1 when a#b

  1 xor 1 = 0 xor 0 = 0 when a=b

One-Time Pad: Encryption

e=000 h=001 i=010 k=011 l=100 r=101 s=110 t=111

Encryption: *Plaintext $\oplus$ Key = Ciphertext*

	h	e	i	l	h	i	t	l	e	r
Plaintext:	001	000	010	100	001	010	111	100	000	101
Key:	111	101	110	101	111	100	000	101	110	000
Ciphertext:	110	101	100	001	110	110	111	001	110	101
	s	r	l	h	s	s	t	h	s	r

One-Time Pad: Decryption

e=000 h=001 i=010 k=011 l=100 r=101 s=110 t=111

Decryption: *Ciphertext ⊕ Key = Plaintext*

	s	r	l	h	s	s	t	h	s	r
Ciphertext:	110	101	100	001	110	110	111	001	110	101
Key:	111	101	110	101	111	100	000	101	110	000
Plaintext:	001	000	010	100	001	010	111	100	000	101
	h	e	i	l	h	i	t	l	e	r

One-Time Pad

Double agent claims following "key" was used:

	s	r	l	h	s	s	t	h	s	r
Ciphertext:	110	101	100	001	110	110	111	001	110	101
"key":	101	111	000	101	111	100	000	101	110	000
"Plaintext":	011	010	100	100	001	010	111	100	000	101
	k	i	l	l	h	i	t	l	e	r

e=000 h=001 i=010 k=011 l=100 r=101 s=110 t=111

One-Time Pad

Or claims the key is…

	s	r	l	h	s	s	t	h	s	r
Ciphertext:	110	101	100	001	110	110	111	001	110	101
"key":	111	101	000	011	101	110	001	011	101	101
"Plaintext":	001	000	100	010	011	000	110	010	011	000
	h	e	l	i	k	e	s	i	k	e

e=000 h=001 i=010 k=011 l=100 r=101 s=110 t=111

One-Time Pad Summary

- **Provably** secure
  - Ciphertext gives **no** useful info about plaintext
  - All plaintexts are **equally likely**
- BUT, only when be used correctly
  - Pad must be random, used only once
  - Pad is known only to sender and receiver
- Note: pad (key) is same size as message
- So, why not distribute msg instead of pad?

Real-World One-Time Pad

- Project VENONA
  - Soviet spies encrypted messages from U.S. to Moscow in 30's, 40's, and 50's
  - Nuclear espionage, etc.
  - Thousands of messages
- Spy carried one-time pad into U.S.
- Spy used pad to encrypt secret messages
- Repeats within the "one-time" pads made cryptanalysis possible

VENONA Decrypt (1944)

[C% Ruth] learned that her husband [v] was called up by the army but he was not sent to the front. He is a mechanical engineer and is now working at the ENORMOUS [ENORMOZ] [vi] plant in SANTA FE, New Mexico. [45 groups unrecoverable]

detain VOLOK [vii] who is working in a plant on ENORMOUS. He is a FELLOWCOUNTRYMAN [ZEMLYaK] [viii]. Yesterday he learned that they had dismissed him from his work. His active work in progressive organizations in the past was cause of his dismissal. In the FELLOWCOUNTRYMAN line LIBERAL is in touch with CHESTER [ix]. They meet once a month for the payment of dues. CHESTER is interested in whether we are satisfied with the collaboration and whether there are not any misunderstandings. He does not inquire about specific items of work [KONKRETNAYa RABOTA]. In as much as CHESTER knows about the role of LIBERAL's group we beg consent to ask C. through LIBERAL about leads from among people who are working on ENOURMOUS and in other technical fields.

- *"Ruth" == Ruth Greenglass*
- *"Liberal" == Julius Rosenberg*
- *"Enormous" == the atomic bomb*

Codebook Cipher

- Literally, a book filled with "codewords"
- Zimmerman Telegram encrypted via codebook

Februar	13605
fest	13732
finanzielle	13850
folgender	13918
Frieden	17142
Friedenschluss	17149
:	:

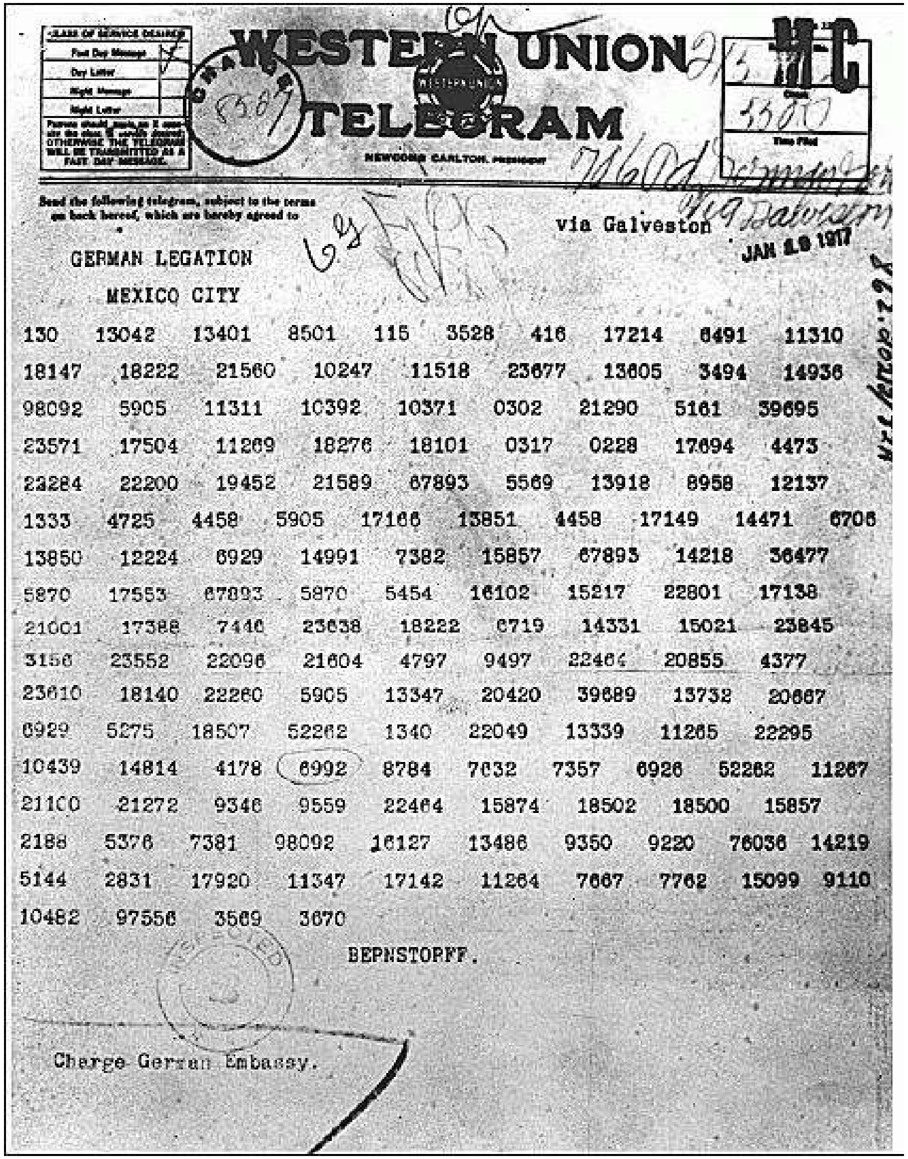
- Modern block ciphers are codebooks!
- More about this later…

Codebook Cipher: Additive

- Codebooks also (usually) use **additive**
- Additive book of "random" numbers
  - Encrypt message with codebook
  - Then choose position in additive book
  - Add in additives to get ciphertext
  - Send ciphertext and additive position (MI)
  - Recipient subtracts additives before decrypting
- Why use an additive sequence?

Zimmerman Telegram

- Perhaps most famous codebook ciphertext ever
- A major factor in U.S. entry into World War I


WESTERN UNION TELEGRAM

NEWCOMB CARLTON, PRESIDENT

Send the following telegram, subject to the terms on back hereof, which are hereby agreed to

via Galveston

JAN 19 1917

GERMAN LEGATION

MEXICO CITY

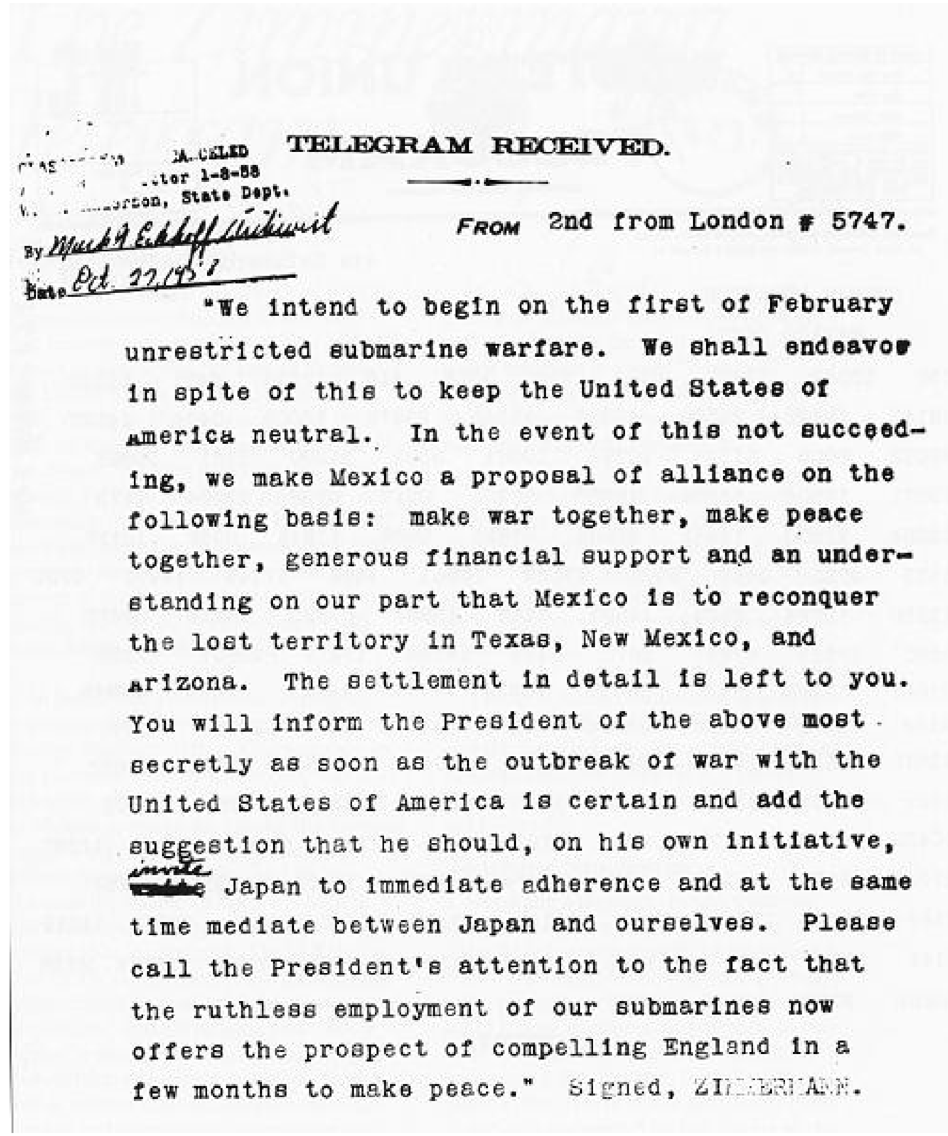
130 13042 13401 8501 115 3528 416 17214 6491 11310
18147 18222 21560 10247 11518 23677 13605 3494 14936
98092 5905 11311 10392 10371 0302 21290 5161 39695
23571 17504 11269 18276 18101 0317 0228 17694 4473
23284 22200 19452 21589 67893 5569 13918 8958 12137
1333 4725 4458 5905 17166 13851 4458 17149 14471 6706
13850 12224 6929 14991 7382 15857 67893 14218 36477
5870 17553 67893 5870 5454 16102 15217 22801 17138
21001 17388 7446 23638 18222 6719 14331 15021 23845
3156 23552 22096 21604 4797 9497 22464 20855 4377
23610 18140 22260 5905 13347 20420 39689 13732 20667
6929 5275 18507 52262 1340 22049 13339 11265 22295
10439 14814 4178 6992 8784 7632 7357 6926 52262 11267
21100 21272 9346 9559 22464 15874 18502 18500 15857
2188 5376 7381 98092 16127 13486 9350 9220 76036 14219
5144 2831 17920 11347 17142 11264 7667 7762 15099 9110
10482 97556 3569 3670

BERNSTORFF.

Charge German Embassy.

Zimmerman Telegram Decrypted

- British had recovered partial codebook
- Then able to fill in missing parts



TELEGRAM RECEIVED.

FROM 2nd from London # 5747.

"We intend to begin on the first of February unrestricted submarine warfare. We shall endeavor in spite of this to keep the United States of America neutral. In the event of this not succeeding, we make Mexico a proposal of alliance on the following basis: make war together, make peace together, generous financial support and an understanding on our part that Mexico is to reconquer the lost territory in Texas, New Mexico, and Arizona. The settlement in detail is left to you. You will inform the President of the above most secretly as soon as the outbreak of war with the United States of America is certain and add the suggestion that he should, on his own initiative, invite Japan to immediate adherence and at the same time mediate between Japan and ourselves. Please call the President's attention to the fact that the ruthless employment of our submarines now offers the prospect of compelling England in a few months to make peace." Signed, ZIMMERMANN.

Random Historical Items

- Crypto timeline
- Spartan Scytale   transposition cipher
- Caesar's cipher
- Poe's short story: The Gold Bug
- Election of 1876

Election of 1876

- "Rutherfraud" Hayes vs "Swindling" Tilden
  - Popular vote was virtual tie
- Electoral college delegations for 4 states (including Florida) in dispute
- Commission gave all 4 states to Hayes
  - Voted on straight party lines
- Tilden accused Hayes of bribery
  - Was it true?

Election of 1876

- Encrypted messages by Tilden supporters later emerged
- Cipher: Partial codebook, plus transposition
- Codebook substitution for important words

ciphertext	plaintext
Copenhagen	Greenbacks
Greece	Hayes
Rochester	votes
Russia	Tilden
Warsaw	telegram
:	:

Election of 1876

- Apply codebook to original message
- Pad message to multiple of 5 words (total length, 10,15,20,25 or 30 words)
- For each length, a fixed permutation applied to resulting message
- Permutations found by comparing several messages of same length
- Note that the **same key** is applied to all messages of a given length

Election of 1876

- Ciphertext: ***Warsaw they read all unchanged last are idiots can't situation***
- Codebook: Warsaw → telegram
- Transposition: 9,3,6,1,10,5,2,7,4,8

can't read last Warsaw situation unchanged they are all idiots.

- Plaintext: ***Can't read last telegram. Situation unchanged. They are all idiots.***
- A weak cipher made worse by reuse of key
- Lesson? Don't overuse keys!

Early 20th Century

- WWI Zimmerman Telegram
- "Gentlemen do not read each other's mail"
  - Henry L. Stimson, Secretary of State, 1929
- WWII golden age of cryptanalysis
  - Midway/Coral Sea
  - Japanese **Purple** (codename **MAGIC**)
  - German **Enigma** (codename **ULTRA**)

Post-WWII History

- Claude Shannon — father of the science of information theory
- Computer revolution — lots of data to protect
- Data Encryption Standard (DES), 70's
- Public Key cryptography, 70's
- CRYPTO conferences, 80's
- Advanced Encryption Standard (AES), 90's
- The crypto genie is out of the bottle…

Claude Shannon

- The founder of Information Theory
- 1949 paper: [Comm. Thy. of Secrecy Systems]
- Fundamental concepts
  - **Confusion** obscure relationship between plaintext and ciphertext
  - **Diffusion** spread plaintext statistics through the ciphertext
- Proved one-time pad is secure
- One-time pad is confusion-only, while double transposition is diffusion-only

Taxonomy of Cryptography

- **Symmetric Key**
  - Same key for encryption and decryption
  - Modern types: Stream ciphers, Block ciphers
- **Public Key** (or "asymmetric" crypto)
  - Two keys, one for encryption (public), and one for decryption (private)
  - And digital signatures   nothing comparable in symmetric key crypto
- **Hash algorithms**
  - Can be viewed as "one way" crypto

Taxonomy of Cryptanalysis

- From perspective of info available to Trudy…
  - Ciphertext only Trudy's worst case scenario
  - Known plaintext
  - Chosen plaintext
    - "Lunchtime attack"
    - Some protocols will encrypt chosen data
  - Adaptively chosen plaintext
  - Related key
  - Forward search (public key crypto)
  - And others…

Chapter 3: Symmetric Key Crypto

The chief forms of beauty are order and symmetry…
Aristotle

"You boil it in sawdust: you salt it in glue:
You condense it with locusts and tape:
Still keeping one principal object in view
To preserve its symmetrical shape."
Lewis Carroll, *The Hunting of the Snark*

Symmetric Key Crypto

- Stream cipher generalize one-time pad
  - Except that key is relatively short
  - Key is stretched into a long **keystream**
  - Keystream is used just like a one-time pad
- Block cipher generalized codebook
  - Block cipher key determines a codebook
  - Each key yields a different codebook
  - Employs both "confusion" and "diffusion"

Stream Ciphers

Stream Ciphers

- Once upon a time, not so very long ago… stream ciphers were the king of crypto
- Today, not as popular as block ciphers
- We'll discuss two stream ciphers:
- A5/1
  - Based on shift registers
  - Used in GSM mobile phone system
- RC4
  - Based on a changing lookup table
  - Used many places

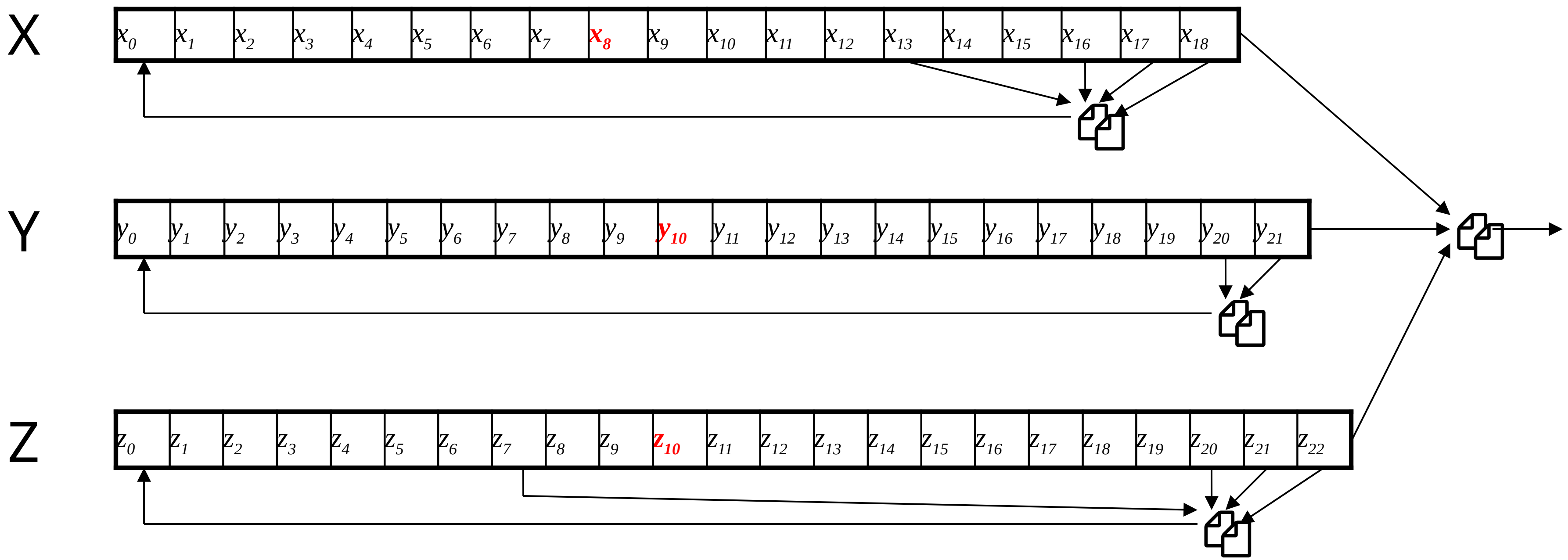
A5/1: Shift Registers

- A5/1 uses 3 shift registers
  - X: 19 bits $(x_0,x_1,x_2,\ldots,x_{18})$
  - Y: 22 bits $(y_0,y_1,y_2,\ldots,y_{21})$
  - Z: 23 bits $(z_0,z_1,z_2,\ldots,z_{22})$

A5/1: Keystream

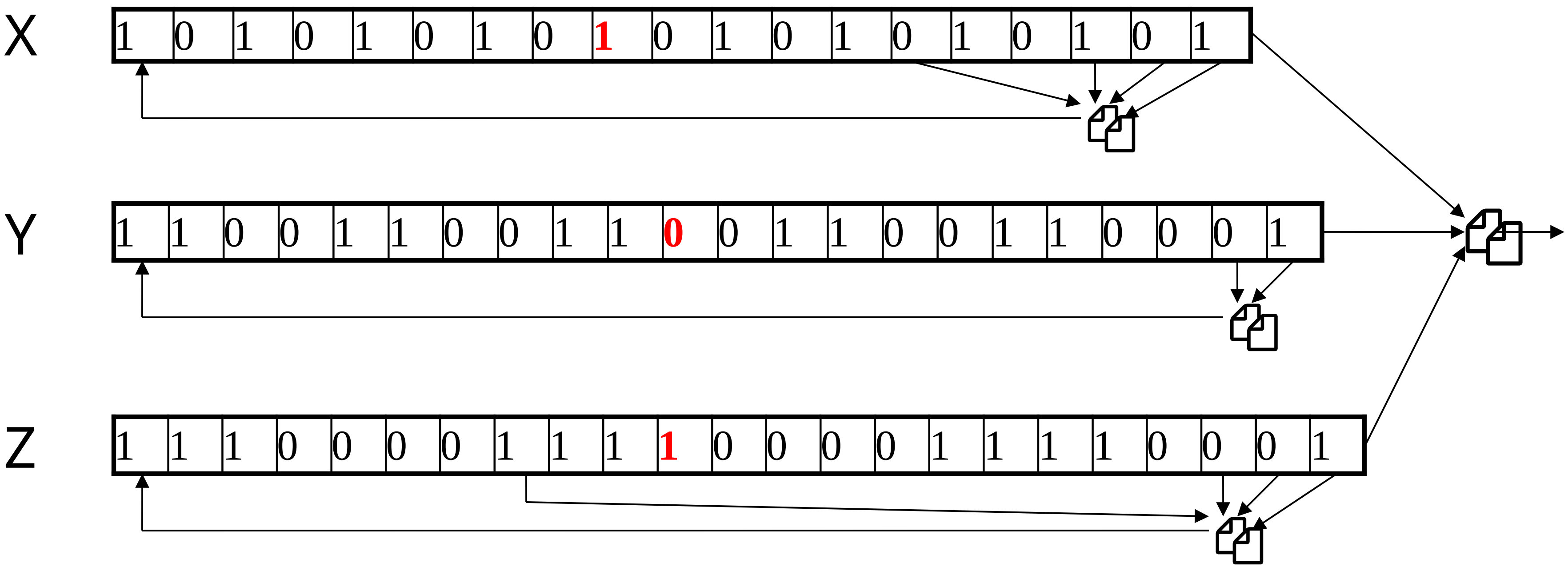
- *At each iteration:* $m = \text{maj}(x_8, y_{10}, z_{10})$
  - *Examples:* maj(0,1,0) = 0 *and* maj(1,1,0) = 1
- *If* $x_8 = m$ *then* X *steps*
  - $t = x_{13} \oplus x_{16} \oplus x_{17} \oplus x_{18}$
  - $x_i = x_{i-1}$ for i = 18,17,…,1 and $x_0 = t$
- *If* $y_{10} = m$ *then* Y *steps*
  - $t = y_{20} \oplus y_{21}$
  - $y_i = y_{i-1}$ for i = 21,20,…,1 and $y_0 = t$
- *If* $z_{10} = m$ *then* Z *steps*
  - $t = z_7 \oplus z_{20} \oplus z_{21} \oplus z_{22}$
  - $z_i = z_{i-1}$ for i = 22,21,…,1 and $z_0 = t$
- *Keystream* ***bit*** *is* $x_{18} \oplus y_{21} \oplus z_{22}$

A5/1



- *Each variable here is a single bit*
- *Key is used as **initial fill** of registers*
- *Each register steps (or not) based on* $\mathrm{maj}(x_8, y_{10}, z_{10})$
- *Keystream bit is XOR of rightmost bits of registers*

A5/1



- *In this example,* $m = \mathrm{maj}(x_8, y_{10}, z_{10}) = \mathrm{maj}(1,0,1) = 1$
- *Register* X *steps,* Y *does not step, and* Z *steps*
- *Keystream bit is XOR of right bits of registers*
- *Here, keystream bit will be* $0 \oplus 1 \oplus 0 = 1$

Shift Register Crypto

- Shift register crypto efficient in hardware
- Often, slow if implemented in software
- In the past, very, very popular
- Today, more is done in software due to fast processors
- Shift register crypto still used some
  - Especially in resource-constrained devices

RC4

- A self-modifying lookup table
- Table always contains a permutation of the byte values 0,1,…,255
- Initialize the permutation using key
- At each step, RC4 does the following
  - Swaps elements in current lookup table
  - Selects a keystream byte from table
- Each step of RC4 produces a **byte**
  - Efficient in software
- Each step of A5/1 produces only a bit
  - Efficient in hardware

RC4 Initialization

- S[] is permutation of 0,1,...,255
- key[] contains N bytes of key

```
for i = 0 to 255
        S[i] = i
        K[i] = key[i (mod N)]
next i
j = 0
for i = 0 to 255
        j = (j + S[i] + K[i]) mod 256
        swap(S[i], S[j])
next i
i = j = 0
```

RC4 Keystream

- At each step, swap elements in table and select keystream byte

```
i = (i + 1) mod 256
j = (j + S[i]) mod 256
swap(S[i], S[j])
t = (S[i] + S[j]) mod 256
keystreamByte = S[t]
```

- Use keystream bytes like a one-time pad
- **Note:** first 256 bytes should be discarded
  - Otherwise, related key attack exists

Stream Ciphers

- Stream ciphers were popular in the past
  - Efficient in hardware
  - Speed was needed to keep up with voice, etc.
  - Today, processors are fast, so software-based crypto is usually more than fast enough
- Future of stream ciphers?
  - Shamir declared "the death of stream ciphers"
  - May be greatly exaggerated…

Block Ciphers

(Iterated) Block Cipher

- Plaintext and ciphertext consist of fixed-sized blocks
- Ciphertext obtained from plaintext by iterating a **round function**
- Input to round function consists of **key** and **output** of previous round
- Usually implemented in software

Feistel Cipher: Encryption

- ***Feistel cipher*** *is a type of block cipher*
  - ***Not*** *a specific block cipher*
- *Split plaintext block into left and right halves:* $P = (L_0, R_0)$
- *For each round* $i = 1, 2, \ldots, n$*, compute*

  $$L_i = R_{i-1}$$

  $$R_i = L_{i-1} \oplus F(R_{i-1}, K_i)$$

  where F *is* ***round function*** *and* K_i *is* ***subkey***
- *Ciphertext:* $C = (L_n, R_n)$

Feistel Cipher: Decryption

- *Start with ciphertext* $C = (L_n, R_n)$
- *For each round* $i = n, n-1, \ldots, 1$, *compute*

  $R_{i-1} = L_i$

  $L_{i-1} = R_i \oplus F(R_{i-1}, K_i)$

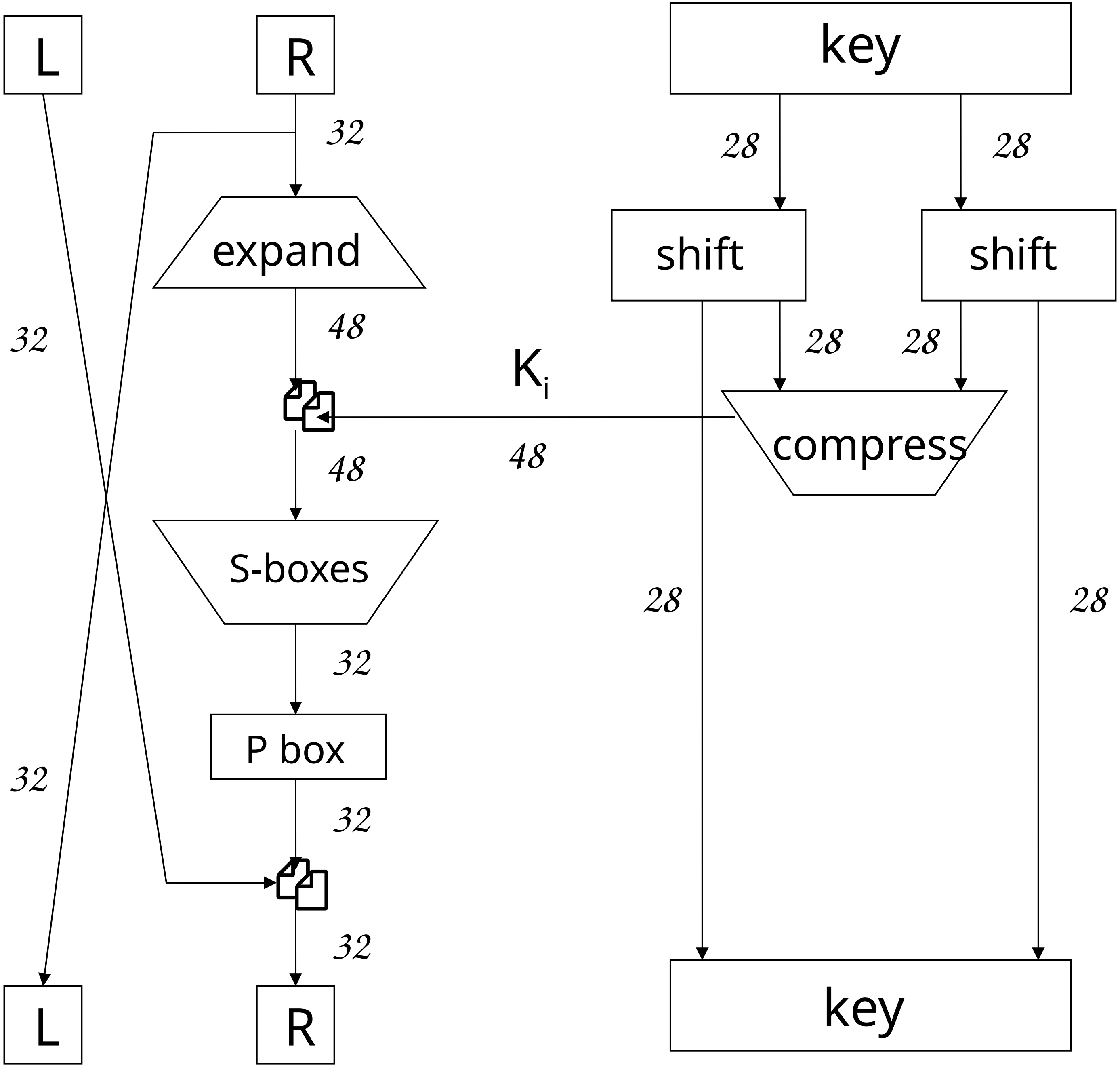
  where F *is round function and* K_i *is subkey*
- *Plaintext:* $P = (L_0, R_0)$
- *Decryption works for any function* F
  - *But only secure for certain functions* F

Data Encryption Standard

- *DES* developed in 1970's
- Based on IBM's Lucifer cipher
- DES was U.S. government standard
- Development of DES was controversial
  - NSA secretly involved
  - Design process was secret
  - Key length reduced from 128 to 56 bits
  - Subtle changes to Lucifer algorithm

DES Numerology

- DES is a Feistel cipher with…
  - 64 bit block length
  - 56 bit key length
  - 16 rounds
  - 48 bits of key used each round (subkey)
- Round function is simple (for block cipher)
- Security depends heavily on "S-boxes"
  - Each S-box maps 6 bits to 4 bits



One Round of DES

DES Expansion Permutation

- Input 32 bits

0	1	2	3	4	5	6	7	8	9	10	11	12	13	14	15
16	17	18	19	20	21	22	23	24	25	26	27	28	29	30	31

- Output 48 bits

31	0	1	2	3	4	3	4	5	6	7	8
7	8	9	10	11	12	11	12	13	14	15	16
15	16	17	18	19	20	19	20	21	22	23	24
23	24	25	26	27	28	27	28	29	30	31	0

DES S-box

- 8 “substitution boxes” or S-boxes
- Each S-box maps 6 bits to 4 bits
- Here is S-box number 1

input bits (0,5)

input bits (1,2,3,4)

	0000	0001	0010	0011	0100	0101	0110	0111	1000	1001	1010	1011	1100	1101	1110	1111
00	1110	0100	1101	0001	0010	1111	1011	1000	0011	1010	0110	1100	0101	1001	0000	0111
01	0000	1111	0111	0100	1110	0010	1101	0001	1010	0110	1100	1011	1001	0101	0011	1000
10	0100	0001	1110	1000	1101	0110	0010	1011	1111	1100	1001	0111	0011	1010	0101	0000
11	1111	1100	1000	0010	0100	1001	0001	0111	0101	1011	0011	1110	1010	0000	0110	1101

DES P-box

- Input 32 bits

0	1	2	3	4	5	6	7	8	9	10	11	12	13	14	15
16	17	18	19	20	21	22	23	24	25	26	27	28	29	30	31

- Output 32 bits

15	6	19	20	28	11	27	16	0	14	22	25	4	17	30	9
1	7	23	13	31	26	2	8	18	12	29	5	21	10	3	24

DES Subkey

- *56 bit DES key, numbered 0,1,2,…,55*
- *Left half key bits,* LK

49	42	35	28	21	14	7
0	50	43	36	29	22	15
8	1	51	44	37	30	23
16	9	2	52	45	38	31

- *Right half key bits,* RK

55	48	41	34	27	20	13
6	54	47	40	33	26	19
12	5	53	46	39	32	25
18	11	4	24	17	10	3

DES Subkey

- For rounds `i=1,2,...,16`
  - Let LK = (LK circular shift left by r_i)
  - Let RK = (RK circular shift left by r_i)
  - Left half of subkey K_i is of LK bits

```
13 16 10 23  0  4  2 27 14  5 20  9
22 18 11  3 25  7 15  6 26 19 12  1
```

  - Right half of subkey K_i is RK bits

```
12 23  2  8 18 26  1 11 22 16  4 19
15 20 10 27  5 24 17 13 21  7  0  3
```

DES Subkey

- For rounds 1, 2, 9 and 16 the shift r_i is 1, and in all other rounds r_i is 2
- Bits 8,17,21,24 of LK omitted each round
- Bits 6,9,14,25 of RK omitted each round
- **Compression permutation** yields 48 bit subkey K_i from 56 bits of LK and RK
- **Key schedule** generates subkey

DES Last Word (Almost)

- An initial permutation before round 1
- Halves are swapped after last round
- A final permutation (inverse of initial perm) applied to (R_{16}, L_{16})
- None of this serves any security purpose

Security of DES

- Security depends heavily on S-boxes
  - Everything else in DES is linear
- 35+ years of intense analysis has revealed no back door
- Attacks, essentially exhaustive key search
- **Inescapable conclusions**
  - Designers of DES knew what they were doing
  - Designers of DES were way ahead of their time (at least wrt certain cryptanalytic techniques)

Block Cipher Notation

- P = *plaintext block*
- C = *ciphertext block*
- *Encrypt* P *with key* K *to get ciphertext* C
  - C = E(P, K)
- *Decrypt* C *with key* K *to get plaintext* P
  - P = D(C, K)
- *Note:* P = D(E(P, K), K) *and* C = E(D(C, K), K)
  - *But* P   D(E(P, K_1), K_2) *and* C   E(D(C, K_1), K_2) *when* K_1 K_2

Triple DES

- Today, 56 bit DES key is too small
  - Exhaustive key search is feasible
- But DES is everywhere, so what to do?
- **Triple DES** or **3DES** (112 bit key)
  - $C = E(D(E(P,K_1),K_2),K_1)$
  - $P = D(E(D(C,K_1),K_2),K_1)$
- Why Encrypt-Decrypt-Encrypt with 2 keys?
  - Backward compatible: $E(D(E(P,K),K),K) = E(P,K)$
  - And 112 is a lot of bits

3DES

- Why not $C = E(E(P,K),K)$ instead?
  - Trick question — still just 56 bit key
- Why not $C = E(E(P,K_1),K_2)$ instead?
- A (semi-practical) **known plaintext** attack
  - Pre-compute table of $E(P,K_1)$ for every possible key K_1 (resulting table has 2^{56} entries)
  - Then for each possible K_2 compute $D(C,K_2)$ until a match in table is found
  - When match is found, have $E(P,K_1) = D(C,K_2)$
  - Result gives us keys: $C = E(E(P,K_1),K_2)$

Advanced Encryption Standard

- Replacement for DES
- AES competition (late 90's)
  - NSA openly involved
  - Transparent selection process
  - Many strong algorithms proposed
  - Rijndael Algorithm ultimately selected (pronounced like "Rain Doll" or "Rhine Doll")
- Iterated block cipher (like DES)
- Not a Feistel cipher (unlike DES)

AES: Executive Summary

- **Block size:** 128 bits (others in Rijndael)
- **Key length:** 128, 192 or 256 bits (independent of block size in Rijndael)
- 10 to 14 rounds (depends on key length)
- Each round uses 4 functions (3 "layers")
  - ByteSub (nonlinear layer)
  - ShiftRow (linear mixing layer)
  - MixColumn (nonlinear layer)
  - AddRoundKey (key addition layer)

AES ByteSub

- Treat 128 bit block as 4x4 byte array

$$\begin{bmatrix} a_{00} & a_{01} & a_{02} & a_{03} \\ a_{10} & a_{11} & a_{12} & a_{13} \\ a_{20} & a_{21} & a_{22} & a_{23} \\ a_{30} & a_{31} & a_{32} & a_{33} \end{bmatrix} \longrightarrow \texttt{ByteSub} \longrightarrow \begin{bmatrix} b_{00} & b_{01} & b_{02} & b_{03} \\ b_{10} & b_{11} & b_{12} & b_{13} \\ b_{20} & b_{21} & b_{22} & b_{23} \\ b_{30} & b_{31} & b_{32} & b_{33} \end{bmatrix}.$$

- ByteSub is AES's "S-box"
- Can be viewed as nonlinear (but invertible) composition of two math operations

AES "S-box"

Last 4 bits of input

First 4 bits of input

	0	1	2	3	4	5	6	7	8	9	a	b	c	d	e	f
0	63	7c	77	7b	f2	6b	6f	c5	30	01	67	2b	fe	d7	ab	76
1	ca	82	c9	7d	fa	59	47	f0	ad	d4	a2	af	9c	a4	72	c0
2	b7	fd	93	26	36	3f	f7	cc	34	a5	e5	f1	71	d8	31	15
3	04	c7	23	c3	18	96	05	9a	07	12	80	e2	eb	27	b2	75
4	09	83	2c	1a	1b	6e	5a	a0	52	3b	d6	b3	29	e3	2f	84
5	53	d1	00	ed	20	fc	b1	5b	6a	cb	be	39	4a	4c	58	cf
6	d0	ef	aa	fb	43	4d	33	85	45	f9	02	7f	50	3c	9f	a8
7	51	a3	40	8f	92	9d	38	f5	bc	b6	da	21	10	ff	f3	d2
8	cd	0c	13	ec	5f	97	44	17	c4	a7	7e	3d	64	5d	19	73
9	60	81	4f	dc	22	2a	90	88	46	ee	b8	14	de	5e	0b	db
a	e0	32	3a	0a	49	06	24	5c	c2	d3	ac	62	91	95	e4	79
b	e7	c8	37	6d	8d	d5	4e	a9	6c	56	f4	ea	65	7a	ae	08
c	ba	78	25	2e	1c	a6	b4	c6	e8	dd	74	1f	4b	bd	8b	8a
d	70	3e	b5	66	48	03	f6	0e	61	35	57	b9	86	c1	1d	9e
e	e1	f8	98	11	69	d9	8e	94	9b	1e	87	e9	ce	55	28	df
f	8c	a1	89	0d	bf	e6	42	68	41	99	2d	0f	b0	54	bb	16

AES ShiftRow

- Cyclic shift rows

$$\begin{bmatrix} a_{00} & a_{01} & a_{02} & a_{03} \\ a_{10} & a_{11} & a_{12} & a_{13} \\ a_{20} & a_{21} & a_{22} & a_{23} \\ a_{30} & a_{31} & a_{32} & a_{33} \end{bmatrix} \longrightarrow \texttt{ShiftRow} \longrightarrow \begin{bmatrix} a_{00} & a_{01} & a_{02} & a_{03} \\ a_{11} & a_{12} & a_{13} & a_{10} \\ a_{22} & a_{23} & a_{20} & a_{21} \\ a_{33} & a_{30} & a_{31} & a_{32} \end{bmatrix}$$

AES MixColumn

- Invertible, linear operation applied to each column

$$\begin{bmatrix} a_{0i} \\ a_{1i} \\ a_{2i} \\ a_{3i} \end{bmatrix} \longrightarrow \texttt{MixColumn} \longrightarrow \begin{bmatrix} b_{0i} \\ b_{1i} \\ b_{2i} \\ b_{3i} \end{bmatrix} \quad \text{for } i = 0, 1, 2, 3$$

- Implemented as a (big) lookup table

AES AddRoundKey

- XOR subkey with block

$$\begin{bmatrix} a_{00} & a_{01} & a_{02} & a_{03} \\ a_{10} & a_{11} & a_{12} & a_{13} \\ a_{20} & a_{21} & a_{22} & a_{23} \\ a_{30} & a_{31} & a_{32} & a_{33} \end{bmatrix} \oplus \begin{bmatrix} k_{00} & k_{01} & k_{02} & k_{03} \\ k_{10} & k_{11} & k_{12} & k_{13} \\ k_{20} & k_{21} & k_{22} & k_{23} \\ k_{30} & k_{31} & k_{32} & k_{33} \end{bmatrix} = \begin{bmatrix} b_{00} & b_{01} & b_{02} & b_{03} \\ b_{10} & b_{11} & b_{12} & b_{13} \\ b_{20} & b_{21} & b_{22} & b_{23} \\ b_{30} & b_{31} & b_{32} & b_{33} \end{bmatrix}$$

Block Subkey

- RoundKey (subkey) determined by **key schedule** algorithm

AES Decryption

- To decrypt, process must be invertible
- Inverse of MixAddRoundKey is easy, since ‘⊕’ is its own inverse
- MixColumn is invertible (inverse is also implemented as a lookup table)
- Inverse of ShiftRow is easy (cyclic shift the other direction)
- ByteSub is invertible (inverse is also implemented as a lookup table)

A Few Other Block Ciphers

- Briefly…
  - IDEA
  - Blowfish
  - RC6
- More detailed…
  - TEA

IDEA

- Invented by James Massey
  - One of the giants of modern crypto
- IDEA has 64-bit block, 128-bit key
- IDEA uses **mixed-mode arithmetic**
- Combine different math operations
  - IDEA the first to use this approach
  - Frequently used today

Blowfish

- Blowfish encrypts 64-bit blocks
- Key is variable length, up to 448 bits
- Invented by Bruce Schneier
- Almost a Feistel cipher

$$R_i = L_{i-1} \oplus K_i$$

$$L_i = R_{i-1} \oplus F(L_{i-1} \oplus K_i)$$

- The round function F uses 4 S-boxes
  - Each S-box maps 8 bits to 32 bits
- **Key-dependent S-boxes**
  - S-boxes determined by the key

RC6

- Invented by Ron Rivest
- Variables
  - Block size
  - Key size
  - Number of rounds
- An AES finalist
- Uses **data dependent rotations**
  - Unusual for algorithm to depend on plaintext

Time for TEA…

- Tiny Encryption Algorithm (TEA)
- 64 bit block, 128 bit key
- Assumes 32-bit arithmetic
- Number of rounds is variable (32 is considered secure)
- Uses "weak" round function, so large number of rounds required

TEA Encryption

Assuming 32 rounds:

```
(K[0], K[1], K[2], K[3]) = 128 bit key
(L,R) = plaintext (64-bit block)
delta = 0x9e3779b9
sum = 0
for i = 1 to 32
    sum += delta
    L += ((R<<4)+K[0])^(R+sum)^((R>>5)+K[1])
    R += ((L<<4)+K[2])^(L+sum)^((L>>5)+K[3])
next i
ciphertext = (L,R)
```

TEA Decryption

Assuming 32 rounds:

```
(K[0], K[1], K[2], K[3]) = 128 bit key
(L,R) = ciphertext (64-bit block)
delta = 0x9e3779b9
sum = delta << 5
for i = 1 to 32
    R −= ((L<<4)+K[2])^(L+sum)^((L>>5)+K[3])
    L −= ((R<<4)+K[0])^(R+sum)^((R>>5)+K[1])
    sum −= delta
next i
plaintext = (L,R)
```

TEA Comments

- ***"Almost"*** *a Feistel cipher*
  - *Uses + and - instead of ⊕ (XOR)*
- *Simple, easy to implement, fast, low memory requirement, etc.*
- *Possibly a "related key" attack*
- *eXtended TEA (XTEA) eliminates related key attack (slightly more complex)*
- *Simplified TEA (STEA) insecure version used as an example for cryptanalysis*