Part 1: Cryptography

- Cryptography describes how to transfer messages between participants without anyone else being able to read or modify them
- Prerequisite for Computer Security
- Start module with explaining the basics of cryptography (enough to understand how TLS works; for more details see cryptography module)
- Before we start with Cryptography, need to look at how to represent data

Codes versus Ciphers

Codes vs. Ciphers

- A code is any way to represent data.
  Will use bitstrings (sequence of bits) to represent data.
  Examples:
  - – Morse Code, ASCII, Hex, Base64
- A cipher is a code where it is difficult to derive data from code.
  - Almost always uses a key.
  - Data for a cipher usually called *plain text*, encoding called *cipher text*
  - Function from plain text to cipher text called *encryption*
  - Function from cipher text to plain text called *decryption*

Codes vs. Ciphers

What is "27" encoded in binary?

- 0001 1011
- 0010 0111
- 110110 111011
- 0011 0010 0011 0111
- All of the above

Codes vs. Ciphers

What is "27" encoded in binary?

• 0001 1011	27 as decimal
• 0010 0111	27 as hex
• 110110 111011	27 as Base64
• 0011 0010 0011 0111	27 as ASCII
• All of the above	Yes

Hex

- Characters 0 to F encode 4 bits
- Easiest way to write down binary as text

$0 = 0000$	$8 = 1000$
$1 = 0001$	$9 = 1001$
$2 = 0010$	$A = 1010$
$3 = 0011$	$B = 1011$
$4 = 0100$	$C = 1100$
$5 = 0101$	$D = 1101$
$6 = 0110$	$E = 1110$
$7 = 0111$	$F = 1111$

eg 27 in Hex is bitstring 0010 0111

ASCII

Hex	Dec	Char		Hex	Dec	Char	Hex	Dec	Char	Hex	Dec	Char
0x00	0	NULL	null	0x20	32	Space	0x40	64	@	0x60	96	`
0x01	1	SOH	Start of heading	0x21	33	!	0x41	65	A	0x61	97	a
0x02	2	STX	Start of text	0x22	34	"	0x42	66	B	0x62	98	b
0x03	3	ETX	End of text	0x23	35	#	0x43	67	C	0x63	99	c
0x04	4	EOT	End of transmission	0x24	36	$	0x44	68	D	0x64	100	d
0x05	5	ENQ	Enquiry	0x25	37	%	0x45	69	E	0x65	101	e
0x06	6	ACK	Acknowledge	0x26	38	&	0x46	70	F	0x66	102	f
0x07	7	BELL	Bell	0x27	39	'	0x47	71	G	0x67	103	g
0x08	8	BS	Backspace	0x28	40	(	0x48	72	H	0x68	104	h
0x09	9	TAB	Horizontal tab	0x29	41	)	0x49	73	I	0x69	105	i
0x0A	10	LF	New line	0x2A	42	*	0x4A	74	J	0x6A	106	j
0x0B	11	VT	Vertical tab	0x2B	43	+	0x4B	75	K	0x6B	107	k
0x0C	12	FF	Form Feed	0x2C	44	,	0x4C	76	L	0x6C	108	l
0x0D	13	CR	Carriage return	0x2D	45	-	0x4D	77	M	0x6D	109	m
0x0E	14	SO	Shift out	0x2E	46	.	0x4E	78	N	0x6E	110	n
0x0F	15	SI	Shift in	0x2F	47	/	0x4F	79	O	0x6F	111	o
0x10	16	DLE	Data link escape	0x30	48	0	0x50	80	P	0x70	112	p
0x11	17	DC1	Device control 1	0x31	49	1	0x51	81	Q	0x71	113	q
0x12	18	DC2	Device control 2	0x32	50	2	0x52	82	R	0x72	114	r
0x13	19	DC3	Device control 3	0x33	51	3	0x53	83	S	0x73	115	s
0x14	20	DC4	Device control 4	0x34	52	4	0x54	84	T	0x74	116	t
0x15	21	NAK	Negative ack	0x35	53	5	0x55	85	U	0x75	117	u
0x16	22	SYN	Synchronous idle	0x36	54	6	0x56	86	V	0x76	118	v
0x17	23	ETB	End transmission block	0x37	55	7	0x57	87	W	0x77	119	w
0x18	24	CAN	Cancel	0x38	56	8	0x58	88	X	0x78	120	x
0x19	25	EM	End of medium	0x39	57	9	0x59	89	Y	0x79	121	y
0x1A	26	SUB	Substitute	0x3A	58	:	0x5A	90	Z	0x7A	122	z
0x1B	27	FSC	Escape	0x3B	59	;	0x5B	91	[	0x7B	123	{
0x1C	28	FS	File separator	0x3C	60	<	0x5C	92	\	0x7C	124	|
0x1D	29	GS	Group separator	0x3D	61	=	0x5D	93	]	0x7D	125	}
0x1E	30	RS	Record separator	0x3E	62	>	0x5E	94	^	0x7E	126	~
0x1F	31	US	Unit separator	0x3F	63	?	0x5F	95	_	0x7F	127	DEL

Base64

- Shortest way to write binary as printable characters
- Common for keys and crypto
- This module will use Hex

Binary	ASCII	Binary	ASCII	Binary	ASCII	Binary	ASCII
000000	A	010000	Q	100000	g	110000	w
000001	B	010001	R	100001	h	110001	x
000010	C	010010	S	100010	i	110010	y
000011	D	010011	T	100011	j	110011	z
000100	E	010100	U	100100	k	110100	0
000101	F	010101	V	100101	l	110101	1
000110	G	010110	W	100110	m	110110	2
000111	H	010111	X	100111	n	110111	3
001000	I	011000	Y	101000	o	111000	4
001001	J	011001	Z	101001	p	111001	5
001010	K	011010	a	101010	q	111010	6
001011	L	011011	b	101011	r	111011	7
001100	M	011100	c	101100	s	111100	8
001101	N	011101	d	101101	t	111101	9
001110	O	011110	e	101110	u	111110	+
001111	P	011111	f	101111	v	111111	/

Code Demos

See Recording.

Caesar Cipher

- One of the first ciphers was used by Julius Caesar.
- The Caesar Cipher replaces each letter of the alphabet with one three to the right, i.e.
  - a becomes d
  - b becomes e
  - ...
  - z becomes c.

Using a Key

- These ciphers are easy to break because as soon as you know the scheme you can decrypt the message.

Kerckhoffs' principle: A cipher should be secure even if the attacker knows everything about it apart from the key.

- For instance, we can use the Caesar cipher using n rotations.

VIGENERE TABEL

PLAIN TEXT

KEY

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

from Wikipedia

VIGENERE TABEL

PLAIN TEXT

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

from Wikipedia

VIGENERE TABEL

PLAIN TEXT

KEY

Rot 13

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

from Wikipedia

Using a Key

- For instance, we can use the Caesar cipher with n rotations
- But only 26 possible keys so you can just try them all (breaking the cipher is 26 times harder without the key)

Using a Key

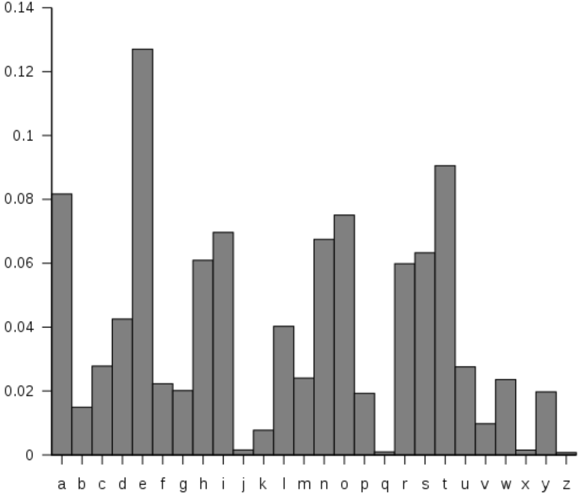
- For instance, we can use the Caesar cipher with n rotations
- But only 26 possible keys so you can just try them all (breaking the cipher is 26 times harder without the key)
- A better scheme replaces each letter with another letter. Here there are $26! \approx 4 \cdot 10^{26}$ possible keys.

Frequency Analysis

- While hard to break by brute force, replacing each letter with another is easy to break using *frequency analysis*.
-
- Frequency analysis counts the number of times
  - each symbol occurs
  - each pair of symbols
  - etc.

  and tries to draw conclusions from this.

Frequency Analysis



from Wikipedia

Summary

- Code is any binary representation of data; cipher is a code where it is difficult to derive data from code.
- Looked at various codes, including Hex
- Looked at substitution ciphers, which replace single letters. These are easily breakable.

Symmetric Cryptography

Overview

- Will now look at proper encryption schemes
- Assumption: All participants share common secret key (obviously problematic!)
- Will consider most important encryption schemes and possible attacks against them
- Need some mathematical prerequisites to explain encryption schemes (modular arithmetic)

Modular Arithmetic

- Arithmetic modulo n means that you count up to $n-1$ then loop back to 0
- i.e., $0, 1, 2, \ldots, n-1, 0, 1, 2, \ldots, n-1, 0, 1, 2, \ldots$
- $a \, mod \, b = r$ for largest whole number k such that $a = b * k + r$
- e.g. $9 \, mod \, 4 = 1$ because $9 = 2 * 4 + 1$

xor

- *xor* ($\oplus$) is binary addition modulo 2:

$$\begin{aligned} 0 \oplus 0 &= 0 \\ 1 \oplus 0 &= 1 \\ 0 \oplus 1 &= 1 \\ 1 \oplus 1 &= 0 \end{aligned}$$

- *xor* on bitstrings of same length defined by applying *xor* to corresponding bits
- Important properties
  - *xor* is associative and commutative
  - for all bitstrings M, $M \oplus 0 = M$
  - for all bitstrings M, $M \oplus M = 0$

  where 0 is a bitstring of all 0's of the appropriate length

One Time Pads

- Needs a key as long as the message.

Message:	`HELLOALICE`
Key:	
Cipher text:	

One Time Pads

- Needs a key as long as the message.

Message:	`HELLOALICE`
Key:	`THFLQRZFJK`
Cipher text:	

One Time Pads

- Needs a key as long as the message.
- XOR/add the key and the message:
  (Demonstrated here with strings and addition and subtraction of keys; for bitstrings use xor)

Message:	`HELLOALICE`
Key:	`THFLQRZFJK`
Cipher text:	`ALRWERKNLO`

One Time Pads

- Needs a key as long as the message.

Cipher text : `ALRWERKNLO`
Key:
Plain text:

One Time Pads

- Needs a key as long as the message.

Cipher text :	`ALRWERKNLO`
Key:	`THFLQRZFJK`
Plain text:	

One Time Pads

- Needs a key as long as the message.
- XOR/add the key and the message:
  (Demonstrated here with strings and addition and subtraction of keys; for bitstrings use xor)

Cipher text :	`ALRWERKNLO`
Key:	`THFLQRZFJK`
Plain text:	`HELLOALICE`

One Time Pads

- Needs a key as long as the message.

Cipher text : `ALRWERKNLO`
Key:
Plain text:

One Time Pads

- Needs a key as long as the message.

Cipher text :	`ALRWERKNLO`
Key:	`UXDTDXFHXN`
Plain text:	

One Time Pads

- Needs a key as long as the message.
- XOR/add the key and the message:
  (Demonstrated here with strings and addition and subtraction of keys; for bitstrings use xor)

Cipher text :	`ALRWERKNLO`
Key:	`UXDTDXFHXN`
Plain text:	`GOODBYEBOB`

One Time Pads

Have perfect encryption:
You don't learn anything about the plaintext from the ciphertext

Theorem

Given any ciphertext of a certain length, without knowing the key the probability of the ciphertext being the encryption of a plaintext of the same length is the same for all plaintexts of the same length as the ciphertext.

One Time Pads

- Problem
  - The key needs to be as long as the message
  - Must use key only once
- Russia during and after WW2
  - Reused the key material (ie encrypted several messages with the same key)
  - Broken by the Venona project

Block Ciphers

- Modern ciphers work on blocks of plain text, not just a single symbol.
- They are made up of a series of permutations and substitutions repeated on each block.
- The key controls the exact nature of the permutations and subsitutions

Advanced Encryption Standard (AES)

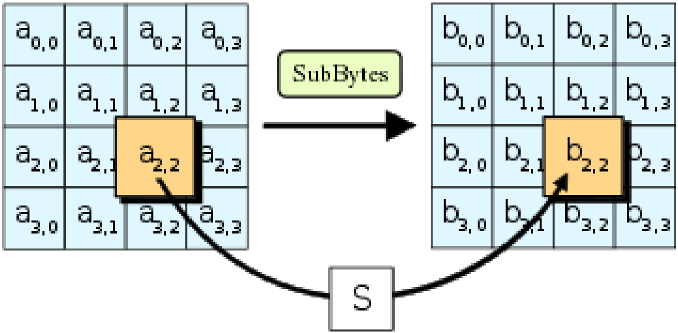
- AES is a state-of-the-art block cipher.
- It works on blocks of 128-bits.
- It generates 10 round keys from a single 128-bit key.
- It uses one permutation: *ShiftRows* and three substitutions *SubBytes*, *MixColumns*, *AddRoundKey*.

Advanced Encryption Standard (AES)

A block of 128 bits is represented by a 4×4-matrix where each matrix element is a byte (8 bits), written as

$$\begin{pmatrix} a_{0,0} & a_{0,1} & a_{0,2} & a_{0,3} \\ a_{1,0} & a_{1,1} & a_{1,2} & a_{1,3} \\ a_{2,0} & a_{2,1} & a_{2,2} & a_{2,3} \\ a_{3,0} & a_{3,1} & a_{3,2} & a_{3,3} \end{pmatrix}$$

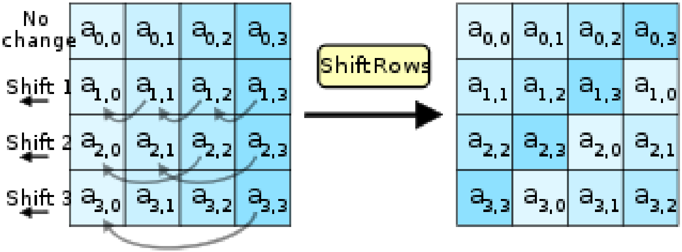
SubBytes: S-box



from Wikipedia

SubByte is an operation on bytes using finite field arithmetic

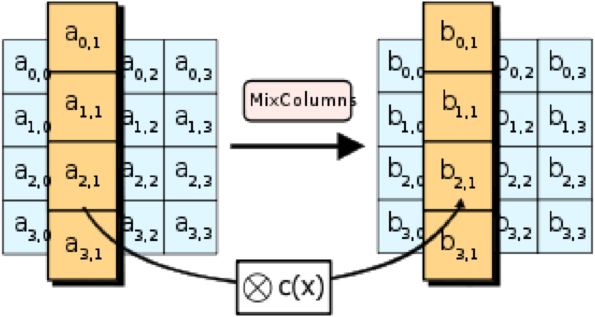
ShiftRows



from Wikipedia

- *ShiftRows* moves the
  - 2nd row one byte to the left,
  - the 3rd row two bytes
  - and the 4th row 3 bytes.

MixColumn

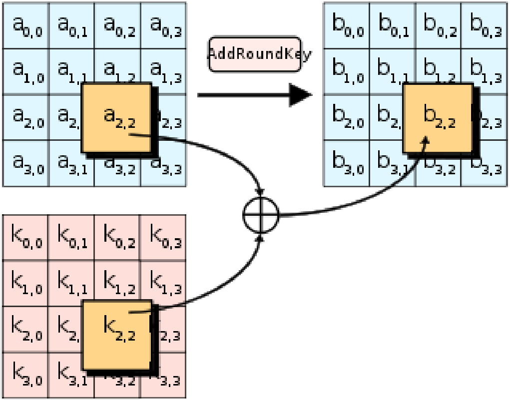


from Wikipedia

MixColumn is a substitution of each column such that:

$$(a_0x^3 + a_1x^2 + a_2x + a_3) \times (3x^3 + x^2 + x + 2)\, mod(x^4 + 1) =$$
$$(b_0x^3 + b_1x^2 + b_2x + b_3)$$

AddRoundKey



from Wikipedia

AddRoundKey applies $\oplus$ to the block and the 128-bit round key (which was generated from the main key).

Security of AES

- No formal proof of security (P = NP?) but best known cryptographic attack requires 2^{126} key guesses - an (irrelevant) improvement of factor 4 compared to 2^{128} key guesses via brute force attack
- There are side channel attacks (eg via measuring power consumption, execution time) — see later in the course.
- Key aspects of security:
    - Shuffling of rows and columns to ensure small change in input causes very big change in output
    - Require at least one non-linear operation (in the sense of linear algebra) on the data - provided by the *SubByte*-operation

DES

- The Data Encryption Standard (DES), was the previous standard.
- Designed by IBM in early 1970's
- Before it was accepted as a standard the NSA stepped in and added S-boxes and fixed the key length at 56 bits

DES

- S-boxes are a type of substitution.
- It was unclear at the time why the NSA added S-boxes to the design.
- Many believed these were a back door for the NSA.

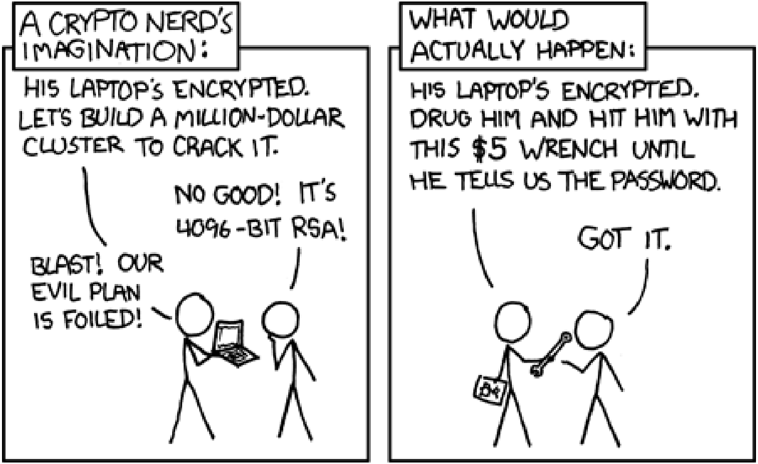
DES

- In 1990, Biham and Shamir discovered differential cryptanalysis.
- The S-boxes had made DES resistant to differential cryptanalysis.
- It seems that the NSA knew about differential cryptanalysis, at the start of the 1970s and had step into to protect DES.

Cost to Break DES

- 1977, Diffie and Hellman, theoretically: $20 million, break in 1 day.
- 1993, theoretically $1 million, in 7 hours.
- 1997, RSA Security offer $10,000 for a real break, won by a distributed computing project, at "no cost"
- EFF (Electronic rights group) break in 56 hours for $250,000
- 2006, COPACOBANA, general purpose brute force, break DES for $10,000
- 2016, `hashcat` on Nvidia GeForce GTX 1080 Ti GPU costing $1000 USD recovers a key in an average of 15 days

A word about key length



Source: `https://xkcd.com/538/`

3-DES

- Triple DES, was a stop gap until AES
- 3-DES takes 3 keys, k_1, k_2 and k_3.

$$E_{k_1,k_2,k_3}(M) = E_{k_3}(D_{k_2}(E_{k_1}(M)))$$

- Setting $k_1 = k_2 = k_3$ gives you DES
- Expected to be good until 2030
- Used in bank cards and RFID chips

Padding

- Block ciphers only work on fixed size blocks.
- If the message isn't of the right block size we need to pad the message.
- But receiver needs to tell the difference between the padding and message.

Padding

- Add random bytes to the end of the block?

Padding

- Add random bytes to the end of the block? No.

Padding

- Add random bytes to the end of the block? No.
- Add zeros to the end of the block?

Padding

- Add random bytes to the end of the block? No.
- Add zeros to the end of the block? No.

Padding

- Add random bytes to the end of the block? No.
- Add zeros to the end of the block? No.
- Write "this is padding"?

Padding

- Add random bytes to the end of the block? No.
- Add zeros to the end of the block? No.
- Write "this is padding"? No.

Padding: PKCS 5/7

- If there is 1 byte of space write 01
- If there are 2 byte of space write 0202
- If there are 3 byte of space write 030303
- ...
- If the message goes to the end of the block add a new block of 16161616..

PKCS 7: 16 byte block, PKCS 5: 8 byte block

Block Cipher Modes

- Block Ciphers can be used in a number of modes:
  1. Electronic codebook mode (ECB)
     - each block is encrypted individually,
     - encrypted blocks are assembled in the same order as the plain text blocks.
     - if blocks are repeated in the plain text, this is revealed by the cipher text.

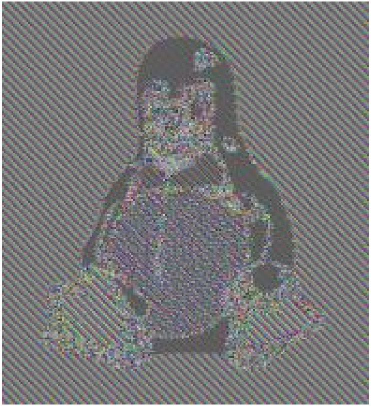
Demo Block Problems

Block Cipher Modes

Original

Block Cipher Modes

Original



ECB

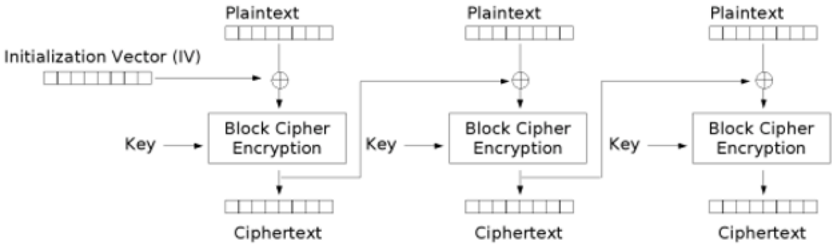
Source: Wikipedia

Block Cipher Modes

2. Cipher Block Chaining Mode (CBC)
   -
     - each block XOR'd with previous block
     - start with a random Initialization Vector (IV)
     - helps overcome replay attack.
   - Suppose the plain text is $B_1, B_2, \ldots, B_n$.

$$\begin{aligned} IV &= \text{random number (sent in the clear)} \\ C_1 &= encrypt(B_1 \oplus IV) \\ C_2 &= encrypt(B_2 \oplus C_1) \\ &\cdots \\ C_n &= encrypt(B_n \oplus C_{n-1}) \end{aligned}$$

Block Cipher Modes



Cipher Block Chaining (CBC) mode encryption

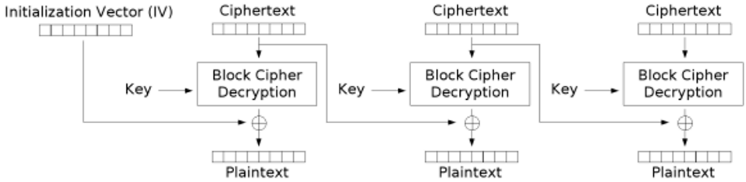
Source: Wikipedia

CBC decrypt

- Receive IV
- Receive cipher text $C_1, C_2, \ldots, C_n$
- Plain text is $B_1, B_2, \ldots, B_n$, where

$$
\begin{aligned}
B_1 &= decrypt(C_1) \oplus IV \\
B_2 &= decrypt(C_2) \oplus C_1 \\
&\cdots \\
B_n &= decrypt(C_n) \oplus C_{n-1}
\end{aligned}
$$

CBC decrypt

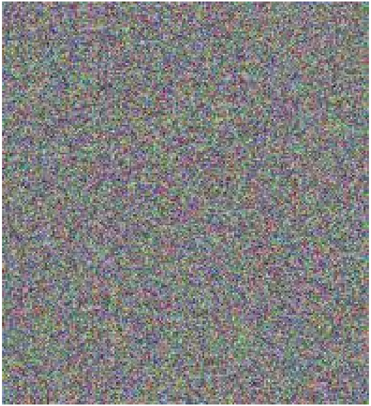


Cipher Block Chaining (CBC) mode decryption

Source: Wikipedia

Block Cipher Modes

Original



CBC

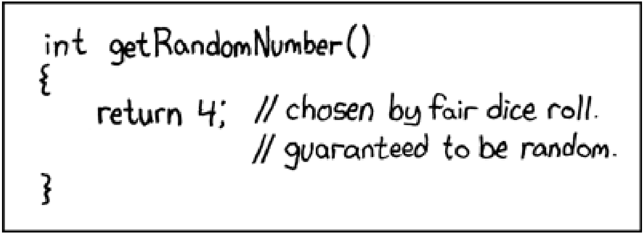
Source: Wikipedia

Probabilistic Encryption

- Probabilistic encryption schemes use random elements to make every encryption different.
- CBC with a random IV is a good way to make encryption probabilistic.
- Using CBC and random IVs lets me encrypt the same message, and with the same key, without an attacker realising.

Misuse of IV

IV must be random, different for each encrypted block



https://xkcd.com/221

Choosing fixed *IV* can have devastating effect
(Zerologon vulnerability for Microsoft Windows
(https://www.secura.com/pathtoimg.php?id=2055))

Zerologon

- Windows has RPC for updating password on the domain controller
- Uses cryptographic protocols for authenticating these requests
- In particular, use AES with block cipher mode called CFB8, which uses IV exactly as CBC mode.
- This is secure with randomly chosen IV
- Implementation chooses IV to be always 0

Zerologon

- Consequence: AES-CFB8 encryption on all-zero plaintext will produce all-zero ciphertext
- This can be used to bypass authentication completely and set domain controller password
- Attack only requires network access to domain controller, which is available from any machine in the domain
- CVSS score of 10.0
- Department of Homeland Security forced all US government institutions to patch Windows Servers within three days

Sony PlayStation

- Sony needs to stop games being copied.
- CD and full disk encryption
- User can read and write areas of the hard disk, for own files, notes, etc
- Why won't CBC work?

Sony PlayStation

- With CBC, you need to encrypt, or decrypt, the whole file to get to the end.
- The Sony PlayStation uses ECB full disk encryption, to stop people copying games.
- User can access files they made themselves
- Hardware controls user access to data.

Sony PlayStation Disk Encryption Attack

1. Remove disk and make copy
2. Replace disk in Playstation.

Sony PlayStation Disk Encryption Attack

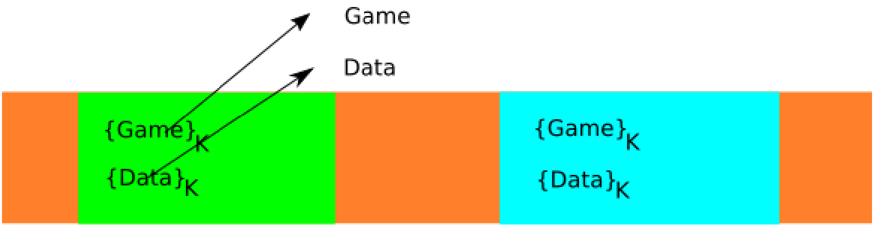
1. Remove disk and make copy
2. Replace disk in Playstation.
3. Copy a file to the disk
4. Remove disk and find the bit of disk that changed (this is the encrypted file)

Sony Play Station Disk Encryption Attack

5. Copy target data to user area

Sony PlayStation Disk Encryption Attack

5. Copy target data to user area
6. Restart the PlayStation and ask for your file back
7. PlayStation decrypts the file and gives you back the plain text

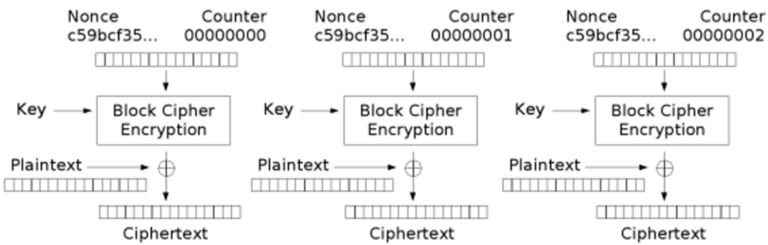


Counter Mode (CTR)

- Plain text: $B_1, B_2, \ldots, B_n$
- IV: random number (sent in clear)
- Cipher text: $C_1, C_2, \ldots, C_n$ where

$$
\begin{array}{rcl}
C_1 & = & B_1 \oplus encrypt(IV) \\
C_2 & = & B_2 \oplus encrypt(IV + 1) \\
 & \ldots & \\
C_n & = & B_n \oplus encrypt(IV + n - 1)
\end{array}
$$

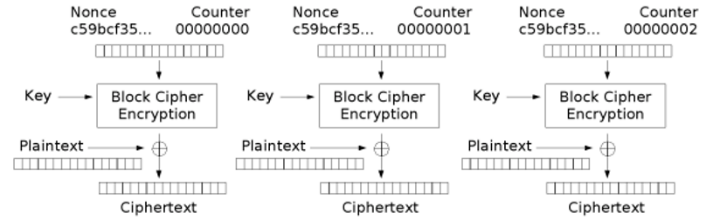
Counter Mode (CTR)



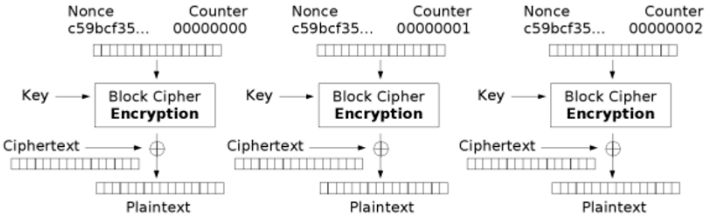
Counter (CTR) mode encryption

Source: Wikipedia

Counter Mode (CTR)



Counter (CTR) mode encryption



Counter (CTR) mode decryption

Cipher Texts Can Be Altered

- AES encryption with a particular key maps any 128-bit block to a 128-bit block (or 256)
- AES decrypt also maps any 128-bit block to a 128-bit block.
- Decrypt can be run on any block (not just encryptions).

Known Plain Text Attacks

- If I know the plaintext I can change CTR encrypted messages
- eg If I know $Enc_{CTR}(M_1)$ and I know M_1, I can make a ciphertext that decrypts to any message I want, eg M_2
- New ciphertext is

$$Enc_{CTR}(M_1) \oplus (M_1 \oplus M_2)$$

Known Plain Text Attacks

Decrypt it:

$$Dec_{CTR}(Enc_{CTR}(M_1) \oplus (M_1 \oplus M_2)) =$$

Known Plain Text Attacks

Decrypt it:

$$Dec_{CTR}(Enc_{CTR}(M_1) \oplus (M_1 \oplus M_2)) =$$
$$Dec_{CTR}(Enc(N||Ctr) \oplus M_1) \oplus (M_1 \oplus M_2) =$$

Known Plain Text Attacks

Decrypt it:

$$
\begin{aligned}
&Dec_{CTR}(Enc_{CTR}(M_1) \oplus (M_1 \oplus M_2)) = \\
&Dec_{CTR}(Enc(N||Ctr) \oplus M_1) \oplus (M_1 \oplus M_2) = \\
&Enc(N||Ctr) \oplus (Enc(N||Ctr) \oplus M_1) \oplus (M_1 \oplus M_2) = \\
&M_2
\end{aligned}
$$

Have to stop this
Subject of future lecture

Summary

- Introduced symmetric encryption. Assumption: All participants share common secret key.
- One-time pad provides perfect encryption (no attack possible), but requires key as long as message.
- Discussed DES and AES algorithms for symmetric encryption of one block
- Discussed block cipher modes (ECB, CTR, CCB) for encryption of several blocks
- Cryptographic schemes are brittle: small errors give rise to attacks. Some examples were given.