Classical Ciphers

• Do not need to know the history of cryptography.

• Understand the basic terminology: e.g. plaintext, ciphertext, cryptanalysis etc.
Plaintext: Human readable text
Ciphertext: Output of encrypted plaintext
Secret Key: The key that is used to encrypt the plaintext such that it cannot be read unless the other party knows the secret key that was used

• How are the ciphers classified?
– Substitution vs Transposition vs Product
Substitution: Encrypts by replacing elements of the plaintext with elements of the ciphertext
Transposition/Permutation: Shuffling the elements of the plaintext in order to obtain the ciphertext
Product: A cipher that performs both substitution and transposition

– Stream vs Block
Stream: Encrypts plaintext at one element at a time
Block: Encrypts multiple elements at a time

– Symmetric vs Public Key
Symmetric: The same key is used for encryption and decryption
Public Key: One key is used for encryption while a different corresponding key is used for decryption

• What is the difference between a monoalphabetic and a polyalphabetic cipher?
Monoalphabetic: A type of substitution cipher that always replaces the same element of the plaintext with the same element of the ciphertext (Letter “a” is always replaced with Letter “b”)
Polyalphabetic: A type of cipher that doesn’t always guarantee to replace the same element of the plaintext with the same ciphertext element

• Why are attackers more interested in compromising the encryption keys than discovering the content of the specific ciphertext?
If the attacker manages to compromise the encryption key, the attacker can decrypt any plaintext that was encrypted using that key. This is better than only finding out the contents of a single ciphertext.

• What are the basic building blocks of symmetric cryptography?
An encryption algorithm that can take a secret key and the plaintext to output the ciphertext
A decryption algorithm that can take the secret key and the ciphertext to output the plaintext
Trusted third party to distribute the same key to both parties

• What are the requirements for strong symmetric encryption?
A strong symmetric encryption should be computationally difficult for the attacker who does not know the key to be able to decrypt the ciphertext. We need a strong encryption algorithm that is resilient to crypto analysis so that the attacker cannot use plaintext ciphertext pairs to figure out the key. Also the attacker should not be able to figure out the plaintext from the ciphertext if the attacker knows the key.

• What is the difference between unconditional and computational security?
Unconditional security: you cannot break the cipher no matter how much time or computational resources you have available (One-time pad)
Computational security: accepts that the algorithm is breakable but you cannot break the cipher in a reasonable amount of time

• Be able to encrypt and decrypt using all classical ciphers discussed in class e.g. Cesar, Playfair, Vigenère (with and without autokey), Railfence, etc.

- Plaintext is encrypted two letters at a time
  - If a pair is a repeated letter, insert filler x, e.g., bolloon ➔ bo lx lo on
  - If both letters fall in the same row, replace each with letter to right with the first element of the row circularly following the last, e.g. ar ➔ rm
  - If both letters fall in the same column, replace each with the letter below it (again wrapping to top from bottom), e.g. mu ➔ cm

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

– Example: Encrypt plaintext sssssdlfjh using Playfair cipher and key hello.
Block cipher that encrypts 2 letters at a time

h	e	l	o	a
b	c	d	f	g
i/j	k	m	n	p
q	r	s	t	u
v	w	x	y	z

Plaintext: sssssdlfjh = sx sx sx sx sd lf jh
Get the letter below in the corresponding letter if letters are in the same column
Get the letter on the right in the chart if the letters are in the same row
Otherwise get the intersection of the row of the first letter and the column of the second letter
Ciphertext: xl xl xl xl xm od qb

– Example: Decrypt ciphertext dyucumyed using Playfair cipher and key hello.

h	e	l	o	a
b	c	d	f	g
i/j	k	m	n	p
q	r	s	t	u
v	w	x	y	z

Plaintext: dyucumyed = dy uc um ye dx
Get the letter above if the letters are in the same column
Get the letter on the left if the letters are in the same row
Otherwise get the intersection of the row of the first letter and the column of the second letter
Ciphertext: fx rg sp we ls

– Example: Encrypt yuiopvbnm using the Vigenère cipher with autokey and key midterm.

		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
Key	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

(without autokey)
Key: hello
Plaintext: thisisatest

Key	H	E	L	L	O	H	E	L	L	O	H
Plaintext	T	H	I	S	I	S	A	T	E	S	T
Ciphertext	A	L	T	D	W	Z	E	E	P	G	A

Autokey Cipher: Encryption

- Ideally want a key as long as the message
- Vigenère proposed the autokey cipher
- A keyword is concatenated with the plaintext itself providing a running key
- E.g., given key deceptive

```
key:        deceptivewearediscoveredsav
plaintext:  wearediscoveredsaveyourself
ciphertext:ZICVTWQNGKZEIIGASXSTSLVVWLA
```

(with autokey)
Key: midterm
Plaintext: yuiopvbnm
Append the plaintext to the key instead of wrapping the text around

Key	M	I	D	T	E	R	M	Y	U
Plaintext	Y	U	I	O	P	V	B	N	M
Ciphertext	K	C	L	H	T	M	N	L	G

(with autokey)

Autokey Cipher: Decryption

- E.g., given
  - key deceptive
  - Ciphertext: ZICVTWQNGKZEIIGASXSTSLVVWLA

```
key:        deceptivewearediscoveredsav
ciphertext:ZICVTWQNGKZEIIGASXSTSLVVWLA
plaintext:  wearediscoveredsaveyourself
```

Append the output of the plaintext to the key

Railfence Cipher:

Split the message into x amount of rows. Then merge from left to right and top down.

- E.g., Encrypt the message "meet me after the toga party" with a rail fence of depth 2

```
m e m a t r h t g p r y
  e t e f e t e o a a t
```

  - Ciphertext: MEMATRHTGPRYETEFETEOAAT

- E.g., Encrypt the message "meet me after the toga party" with a rail fence of depth 3

```
m   t   a   e   h   o   p   t
  e   m   f   r   e   g   a   y
    e   e   t   t   t   a   r
```

  - Ciphertext: MTAEHOPTEMFREGAYEETTTAR

K = 3
Plaintext: wearediscorvedsaveyourselves

W	R	I	O	R	S	E	U	E	E
E	E	S	V	E	A	Y	R	L	S
A	D	C	E	D	V	O	S	V	

Ciphertext: WRIORSEUEEEESVEAYRLSADCEDVOSV

Decryption

K = 3
Ciphertext: WRIORSEUEEEESVEAYRLSADCEDVOSV

How many letters are placed in each row?
Assume all rows are complete

Ignoring any incomplete columns we can compute the numbers of letters on each row. Take the length of the ciphertext and divide it by the key (Ignore the remainder).

29/3 = 9

Each row will have 9 letters.

1. 9
2. 9
3. 9

However, we cannot ignore the leftover letters. We need to distribute them from the first row going down.

We can figure out the number of letters in the incomplete column as length of ciphertext mod K.

29 % 3 = 2

We have 2 letters in the incomplete column. We will distribute them starting with the first row going down.

1. 9 + 1 = 10
2. 9 + 1 = 10
3. 9

Rows:
WRIORSEUEE
EESVEAYRLS
ADCEDVOSV

WEAREDISCOVEREDSAVEYOURSELVES

Encrypt helloworld using K = 3

H	L	O	D
E	O	R	
L	W	L	

Ciphertext: HLODEORLWL

Decryption:
K = 3
Ciphertext: HLODEORLWL

Number of rows: length(ciphertext) / K = 10 / 3 = 3

1. 3
2. 3
3. 3

Number of letters leftover: length(ciphertext) / K = 10 % 3 = 1

1. 3 + 1 = 4

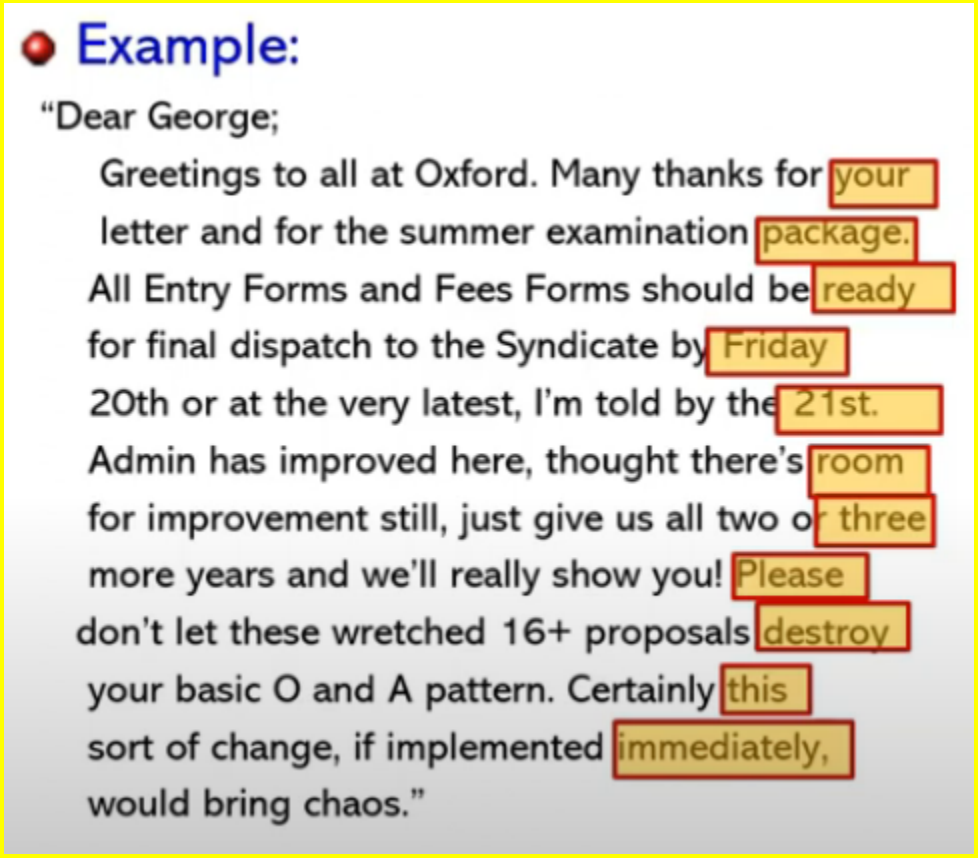
2. 3
3. 3

HLOD
EOR
LWL

HELLOWORLD

Cryptography: Make the message unintelligible

Steganography: Conceal the existence of the message



Example:

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

PERSHING SAILS FROM NY JUNE I

- Modern Examples:
  - Encoding hidden message in the pixels of the image
  - Encoding hidden messages in the bits of sound files
- Advantage: useful to parties who are more concerned with hiding the fact of their secret communication, rather than the contents of messages
- Drawbacks:
  - Significant overhead to hide a small amount of information
  - Once the scheme is known, the system is useless

Block Cipher Principles and DES

• What is the difference between stream and block ciphers?
Stream ciphers encrypt one element at a time (etc. one character, one bit, or one byte). Block ciphers encrypt data multiple elements at a time (etc. 128 bits at a time or 64 bits at a time).

• What are confusion and diffusion? Why are they important when designing a block cipher? Give an example.
Confusion refers to obscuring the relationship between the key and the resulting ciphertext. Confusion is achieved by substitution. Confusion is both used by block and stream ciphers. Mainly used by stream cipher.
Example: a -> b performed with a Caesar Cipher and a key of 1.

Diffusion refers to the obscuring the relationship between the plaintext and the ciphertext. If one bit of the plaintext changes, then the ciphertext will change significantly and in an unpredictable way (pseudo-randomly). Diffusion is achieved through permutation. Only block ciphers use diffusion.
Example: the permutation performed by Row Transposition Cipher

– Example: Consider a block cipher with a property where if one bit of the key changes, many bits of the ciphertext change. Is this an example of confusion or diffusion? Explain.
This is an example of confusion since when the key is changed the ciphertext is changed.

• What primary techniques are used to achieve confusion and diffusion, respectively?
Confusion is achieved by substitution. This is where elements of the plaintext are replaced by elements of the ciphertext.

Diffusion is achieved by shuffling the elements of the plaintext.

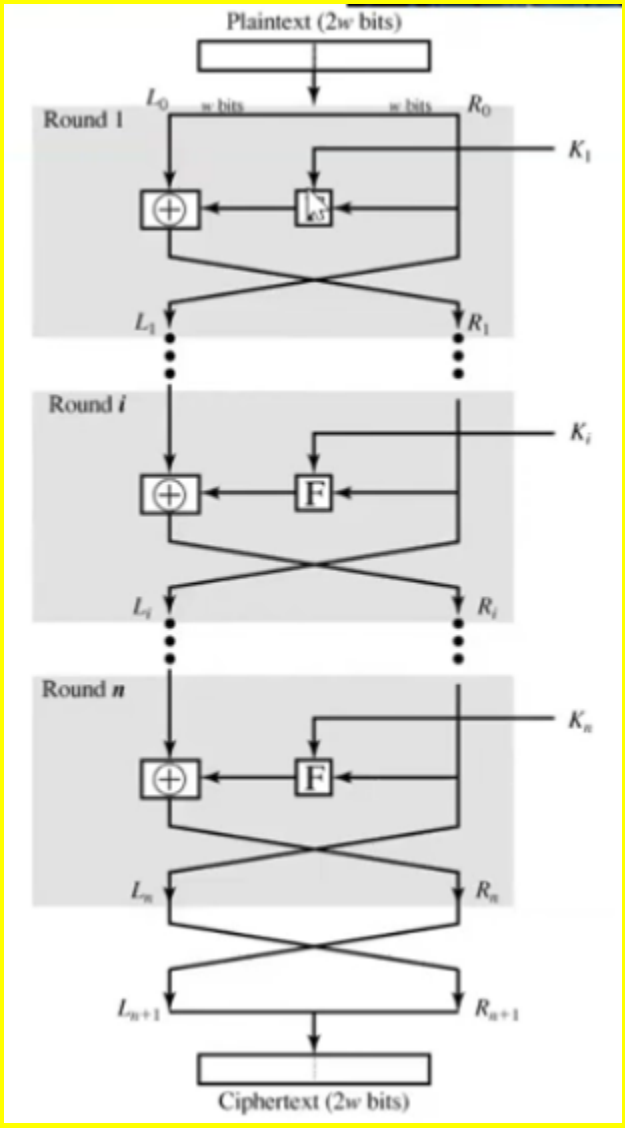
• What makes a block cipher reversible?

For a block cipher to be reversible, there has to be a 1 to 1 mapping between elements of the plaintext and the elements of the ciphertext. Every possible plaintext box has to be mapped to one and exactly one ciphertext block.

• Understand the ideal block cipher concepts.
Different possible keys for a n-bit block cipher: $(2^n)!$
Key size: n * (2^n)
– Example: How many possible keys can I have in an ideal block cipher which encrypts 64-bit text blocks?
$(2^{64})!$

– Example: What is the size of the key in a 64-bit ideal block cipher?
64 * (2^{64})

• How does Feistel cipher perform substitution and permutation?



Takes in an input of a plaintext block that is 2w bits (block is a multiple of two so the block can be divided into two halves) and a key K
The right hand half is fed through a function that will perform some substitution operations (with a key) on the right hand half
The output from this will be xor ed with the original left hand half then the output of the xor function becomes the new right hand half
The original right hand half will become the new left hand half
Each round uses a separate key that is derived from the original key provided by the user

Substitution happens by applying a round function to the right hand half of the data. In addition, substitution is also performed by replacing the original left hand half with an xor of the original left hand half and the output of the round function. Permutation is achieved by swapping the two halves of data after every round.

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

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

• What are the advantages of the Feistel cipher network?
Encryption and decryption are the same process. The only difference is that you change the order of the decryption keys. The last key used for encryption will be the first key used for decryption.

• Know and understand how to use all DES-related tables.
– Example: Consider binary key string:
0011000110110111000000111001111011101111000010100111100101110100. What will be the first five bits after running this key through the PC-1 table?

– Example: What will be the output of S-Box 8 given input 111011?

• What important function do S-Boxes perform in DES encryption?

• Do not need to know all of the S-Box design criteria but need to know at least a few and describe them.

• What is the important property of block cipher security achieved by swapping two halves of the text block after every round?

• What is the avalanche effect? Why is it important?
The avalanche effect is when the plaintext or key is changed a little, which leads to the resulting ciphertext changing dramatically.

• Why not double DES? What is the method of breaking double DES?

• Understand why when using 3DES, it's better to use EDE instead of EEE.

• What is the difference between differential and linear cryptoanalysis?

Twofish

• What key sizes and block sizes does Twofish support?
Supports keys of up to 256 bits. Uses 128-bit blocks.
• Know that Twofish is based upon a Feistel cipher structure (as is DES).
Uses different techniques to derive round keys, different round functions.

• What is the distinguishing feature of Twofish S-Boxes? How do they compare with, say, DES S-Boxes.
In 2DES, S-Boxes are static. In TwoFish, S-Boxes are dynamic. Half of the key is used to randomly generate another set of S-boxes. Different keys yield different S-Boxes. Change in 1 bit of key results can change aggressively.

• What is the distinguishing feature of the Twofish key scheduling algorithm?
Half of the key is used to derive the S-boxes. Other half is for encryption.

Advanced Encryption Standard (AES)

• Compare and contrast AES and DES in terms of security, key size, block size, structure, and the number of rounds.
DES:
Vulnerable to brute force attacks (hence the need for 3DES (aka 3 rounds of DES encryption))
Supports 64 bit key (Removes the last bit of every byte. This results to a 56 bit key)
Block size of 64 bit blocks
Based on the Feistel cipher
Does 16 rounds of substitution and permutations
Focused on bits
DES was originally designed for implementation in hardware

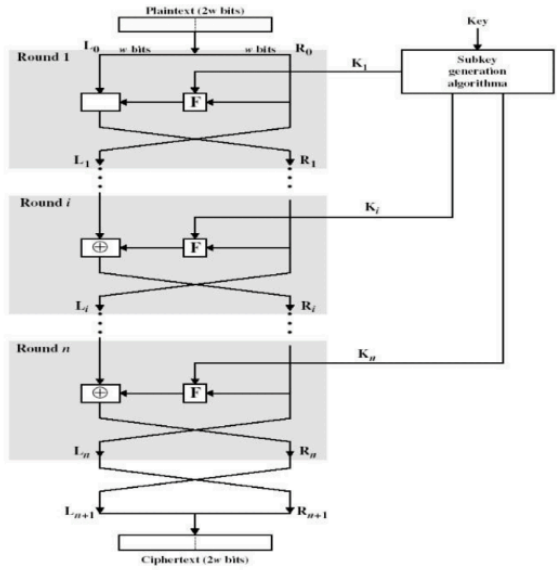
AES:
To date, AES has not been broken
Unclassified
Publicly Disclosed
Available royalty-free for use worldwide
Support key sizes of 128, 192, and 256 bits

Block size of 128-bit blocks
Based on the Substitution-Permutation network
10 rounds if 128 bit key is used
12 rounds if 192 bit key is used
14 rounds if 256 bit key is used
Focused on bytes
AES is designed for implementation in software

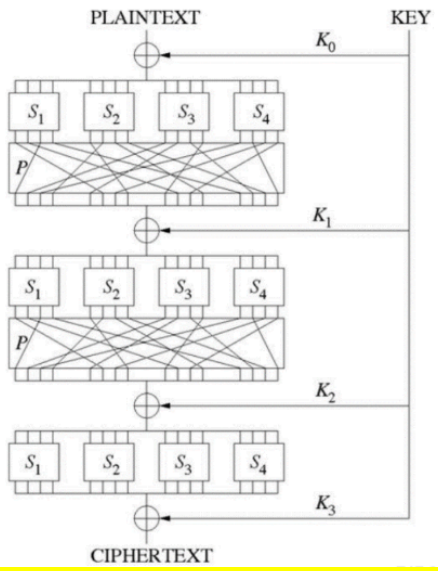
• Is AES based on the Feistel cipher network?
No, AES is based on the Substitution Permutation network.

• Be able to compare and contrast the Feistel cipher framework to the Substitution Permutation framework.

Fiestal Cipher



Substituion Permutation Framework

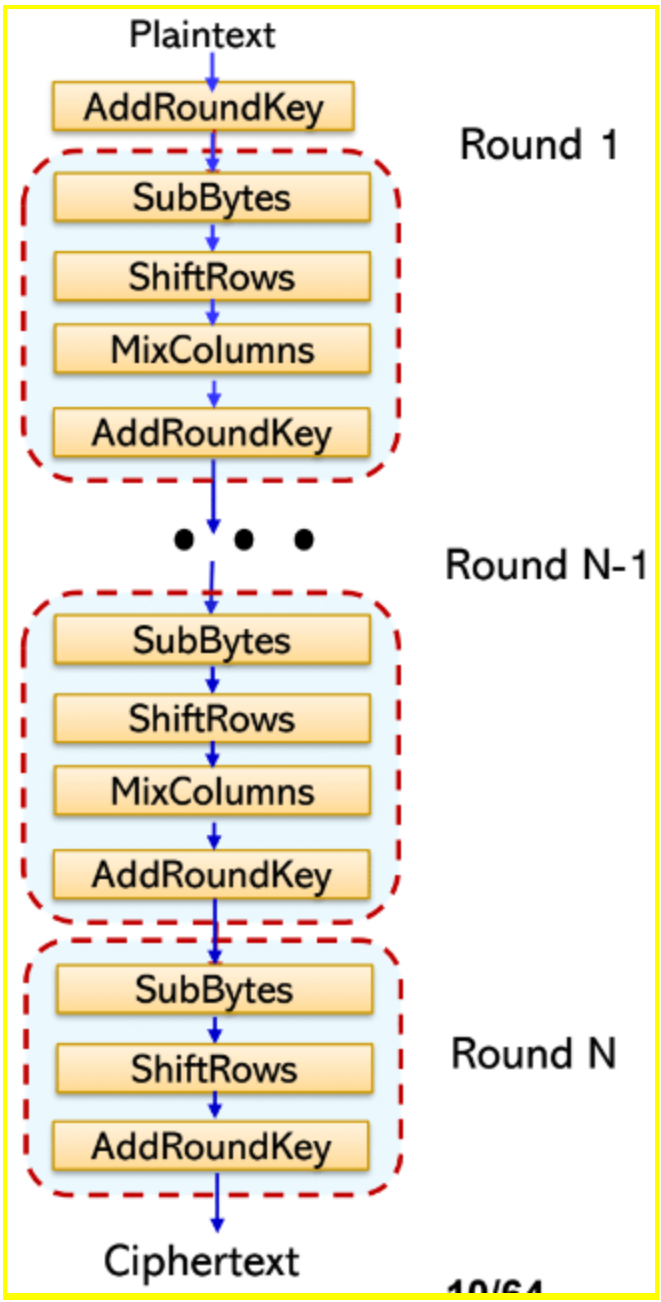


S-Boxes perform substitution
P-Boxes perform permutation

Substitution Permutation network:
Takes plaintext and a key
From the user given key, a subway is derived for each round
We take the plaintext block and xor it with the original key
The output is ran through S-boxes, and then that output is ran through P-Boxes
The output of the P-Boxes will be xor with the round key for the next round

Unlike the Feistel cipher, the decryption process is a series of inverse S-Boxes and P-Boxes (which is different from encryption).

• Know the basic structure of the single AES round.



AddRoundKey: xor the plaintext with the original key
128 bit (16 bytes) plaintext block is arranged as a 4x4 matrix of bytes
SubBytes: perform substitutions
ShiftRows: rotate the rows of the matrix
MixColumns: shuffle the columns of the matrix
AddRoundKey: derive the key for the round and xor the output with the newly derived key

The last round will always only have SubBytes, ShiftRows, and AddRoundKey.

– Example: What sequence of operations does the first AES round perform?

The sequence of operations the first AES round performs is SubBytes, ShiftRows, MixColumns, and AddRoundKey.

– Example: How does the last round of AES differ from all other rounds?
The last round of AES does not have MixColumns.

– Example: In DES, the number of rounds is fixed. Is the same true of AES?
No, the number of rounds is determined by the key size.
10 rounds if 128 bit key is used
12 rounds if 192 bit key is used
14 rounds if 256 bit key is used

• Know how to convert a given plaintext into an AES state.

- Consider a 128-bit (i.e. 16 byte) plaintext block.
  - Each byte of the plaintext block is labeled as follows:

$(a_{0,0}, a_{1,0}, a_{2,0}, a_{3,0}, a_{0,1}, a_{1,1}, a_{2,1}, a_{3,1}, a_{0,2}, a_{1,2}, a_{2,2}, a_{3,2}, a_{0,3}, a_{1,3}, a_{2,3}, a_{3,3})$

- The bytes are then arranged in a 4 x 4 matrix i.e., the "state", as follows:

$$\begin{bmatrix} a_{0,0} & a_{0,1} & a_{0,2} & a_{0,3} \\ a_{1,0} & a_{1,1} & a_{1,2} & a1_{,3} \\ a_{2,0} & a_{2,1} & a_{2,2} & a2_{,3} \\ a_{3,0} & a_{3,1} & a_{3,2} & a_{3,3} \end{bmatrix}$$

Such arrangement is called a column major form

- Example: A 128-bit plaintext block:

EA 83 5C F0 04 45 33 2D 65 5D 98 AD 85 96 B0 C5 is converted into the following matrix.

$$\begin{bmatrix} EA & 04 & 65 & 85 \\ 83 & 45 & 5D & 96 \\ 5C & 33 & 98 & B0 \\ F0 & 2D & AD & C5 \end{bmatrix}$$

Take the matrix and xor it with the original key

- Example:
  - Key: 47 37 94 ED 40 D4 E4 A5 A3 70 3A A6 4C 9F 42 BC
  - Plaintext: AC 77 66 F3 19 FA DC 21 28 D1 29 41 57 5C 00 6A
  - The matrices are then XORed together:

$$\begin{bmatrix} 47 & 40 & A3 & 4C \\ 37 & D4 & 70 & 9F \\ 94 & E4 & 3A & 42 \\ ED & A5 & A6 & BC \end{bmatrix} \oplus \begin{bmatrix} AC & 19 & 28 & 57 \\ 77 & FA & D1 & 5C \\ 66 & DC & 29 & 00 \\ F3 & 21 & 41 & 6A \end{bmatrix} = \begin{bmatrix} EB & 59 & 8B & 1B \\ 40 & 2E & A1 & C3 \\ F2 & 38 & 13 & 42 \\ 1E & 84 & E7 & D6 \end{bmatrix}$$

E.g. 47 ⊕ AC = EB

40 ⊕ 19 = 59

BC ⊕ 6A = D6

0 XOR 0 = 0
1 XOR 1 = 0
1 XOR 0 = 1
0 XOR 1 = 1

47
0100 0111

AC
1010 1100

XOR
1110 1011

1110 = 14 = E
1011 = 11 = B

• In the SubBytes step, know how to use the Rijnadel S-Box in order to perform substitutions.

		y															
		0	1	2	3	4	5	6	7	8	9	A	B	C	D	E	F
x	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

– Example: What is the result of substituting byte AE into the Rijnadel S-Box?
The first four bits of AE are A (Value of x).
The second four bits of AE are E (Value of y).
The result is E4.

– Example: What is the result of substituting byte 2F into the Rijnadel S-Box?
The first four bits of AE are 2 (Value of x).
The second four bits of AE are F (Value of y).
The result is 15.

• Understand how the ShiftRows step works.

Row Number	LCS Shift Amount
1	0
2	1
3	2
4	3

Left Circular Shifted in bytes (LCS)
Move the leftmost byte to the right

– Example: Suppose 12 34 56 78 is the third row of the state. How will the row be transformed in the ShiftRows step?

69	39	A4	7E
0F	A5	D1	98
12	34	56	78
95	67	FD	AB

69	39	A4	7E
A5	D1	98	0F
56	78	12	34
AB	95	67	FD

• Understand how the MixColumns step works.
Matrix multiplication is not cumulative

- Review of matrix multiplication:
- Example: multiply the following matrices

1. $\begin{bmatrix}1 & 2 & 3\\ 4 & 5 & 6\end{bmatrix} \times \begin{bmatrix}7 & 8\\ 9 & 10\\ 11 & 12\end{bmatrix} = \begin{bmatrix}1\times 7+2\times 9+3\times 11 & \end{bmatrix}$

2. $\begin{bmatrix}1 & 2 & 3\\ 4 & 5 & 6\end{bmatrix} \times \begin{bmatrix}7 & 8\\ 9 & 10\\ 11 & 12\end{bmatrix} = \begin{bmatrix}1\times 7+2\times 9+3\times 11 & 1\times 8+2\times 10+3\times 12\end{bmatrix}$

3. $\begin{bmatrix}1 & 2 & 3\\ 4 & 5 & 6\end{bmatrix} \times \begin{bmatrix}7 & 8\\ 9 & 10\\ 11 & 12\end{bmatrix} = \begin{bmatrix}1\times 7+2\times 9+3\times 11 & 1\times 8+2\times 10+3\times 12\\ 4\times 7+5\times 9+6\times 11 & \end{bmatrix}$

4. $\begin{bmatrix}1 & 2 & 3\\ 4 & 5 & 6\end{bmatrix} \times \begin{bmatrix}7 & 8\\ 9 & 10\\ 11 & 12\end{bmatrix} = \begin{bmatrix}1\times 7+2\times 9+3\times 11 & 1\times 8+2\times 10+3\times 12\\ 4\times 7+5\times 9+6\times 11 & 4\times 8+5\times 10+6\times 12\end{bmatrix}$

- How to multiply two matrices using finite field GF(2^8) arithmetic?
  - Same algorithm conventional matrix multiplication, BUT all intermediate additions and multiplications are done as follows:
    - Addition is replaced with XOR operation.
    - Multiplication is done by cross-referencing two tables:
      - E table and
      - L table
      - Cross-referencing is done to avoid directly dealing with complexities of field theory mathematics and to speed up implementation.

- Example:
  - MixColumns multiples matrix by the state using GF(2^8) arithmetic:

$$\begin{bmatrix} 2 & 3 & 1 & 1 \\ 1 & 2 & 3 & 1 \\ 1 & 1 & 2 & 3 \\ 3 & 1 & 1 & 2 \end{bmatrix}$$

Consider state:

$$\begin{bmatrix} 87 & F2 & 4D & 97 \\ 6E & 4C & 90 & EC \\ 46 & E7 & 4A & C3 \\ A6 & 8C & D8 & 95 \end{bmatrix}$$

What is

$$\begin{bmatrix} 2 & 3 & 1 & 1 \\ 1 & 2 & 3 & 1 \\ 1 & 1 & 2 & 3 \\ 3 & 1 & 1 & 2 \end{bmatrix} \times \begin{bmatrix} 87 & F2 & 4D & 97 \\ 6E & 4C & 90 & EC \\ 46 & E7 & 4A & C3 \\ A6 & 8C & D8 & 95 \end{bmatrix} \quad ?$$

(2 * 87) + (3 * 6E) + (1 * 46) + (1 * A6)

Now in GF(2^8)

Replace all + with XOR

(2 * 87) XOR (3 * 6E) XOR (1 * 46) XOR (1 * A6)

```
Multiplication:
1. Lookup both multiplicands in the L-Table
2. Add both output of the L-Table
   if(result > FF) result = result - FF
3. Look up the result in the E-Table
```

– Example: What is the rationale behind the MixColumns step?
Multiple the matrix with the output of ShiftRow

– Given the MixColumns matrix and the state, know how to multiply the matrix by the state.

– Know how to E-table and L-table for performing GF (2^8) multiplication.
∗ Example: What is AB × 34 in GF (2^8)?
L(AB) * L(34) = 2D + 21 = 4E

```
>>> hex(0x2d + 0x21)
'0x4e'
```

Since 4E <= FF, no need to subtract FF from the result

E(4E) = 78

Compute 02 * 87 in GF(2^8)
Use L-table

L(02) = 19
L(87) = 74

Add the two values together
L(02) + L(87) = 19 + 74 = 8D

```
>>> hex(0x19 + 0x74)
'0x8d'
```

Since 8D <= FF, no need to subtract FF from the result

Run the result through the E-table
E(8D) = 15

02 * 87 = 15 (in GF(2^8))

Compute (3 * 6E) in GF(2^8)

L(03) * L(6E) = 01 + 3D = 3E

```
>>> hex(0x01 + 0x3D)
'0x3e'
```

Since 3E <= FF, no need to subtract FF from the result

E(3E) = B2

(3 * 6E) = B2 (in GF(2^8))

F6 * DD
L(F6) + L(DD) = FE + F2 = 1F0

Since 1F0 > FF, we need to subtract FF from 1F0

1F0 - FF = F1

```
>>> hex(0x1f0 - 0xff)
'0xf1'
```

E(F1) = 4B

15 + B2 + 46 + A6
15 XOR B2 XOR 46 XOR A6 = 47

```
>>> hex(0x15 ^ 0xb2 ^ 0x46 ^ 0xa6)
'0x47'
```

• Understand how the AddRoundKey step works.

- The 16-byte (i.e. 128-bit) key is converted into a vector of 4 sixteen byte elements.

$$\begin{bmatrix} K_0 & K_4 & K_8 & K_{12} \\ K_1 & K_5 & K_9 & K_{13} \\ K_2 & K_6 & K_{10} & K_{14} \\ K_3 & K_7 & K_{11} & K_{15} \end{bmatrix} \Rightarrow [W_0 \;\; W_1 \;\; W_2 \;\; W_3]$$

Where $W_0 = K_0K_1K_2K_3$, $W_1 = K_4K_5K_6K_7$, etc..

– Example: Given the initial round key 93 1E D7 36 D6 C2 1B D9 D2 01 86 98 EA 07 D2 F5 , write the key in the matrix form, convert the key into four 4-byte vectors W 0 , W 1 , and W 2 , and W 3 , and derive vectors W 4 , W 5 , and W 6 , and W 7 for the next round.

Original Key: 0f 15 71 c9 47 d9 e8 59 0c b7 ad d6 af 7f 67 98

W0 = 0F 15 71 C9
W1 = 47 D9 E8 59
W2 = 0C B7 AD D6
W3 = AF 7F 67 98

g(W3) =

SubBytes(W3) = SubBytes ([AF 7F 67 98]) = = X1
Rotate one byte to the left
RotWord(X1) = RotWord([79 D2 85 46]) = [D2 85 46 79] = Y1
Y1 XOR RoundConstant(1) = [D2 85 46 79] XOR [01 00 00 00] = [D3 85 46 79] = Z1

W4 = Z1 XOR W0 = [D3 85 46 79] XOR [0F 15 71 C9] = [DC 90 37 B0]
W5 = W4 XOR W1 = [DC 90 37 B0] XOR [47 D9 E8 59] = [9B 49 DF E9]
W6 = W5 XOR W2 = [9B 49 DF E9] XOR [0C B7 AD D6] = [97 FE 72 3F]
W7 = W6 XOR W3 = [97 FE 72 3F] XOR

Block Cipher Modes

• What are advantages and disadvantages of ECB, CBC, CFB, and CTR modes? Be able to analyze their security, efficiency, etc.

• Be able to prove the correctness of the given mode (similar to what we did in class). – Example: Show that CBC cipher mode is correct (e.g. Slide 14 of Block Cipher Modes of Operation Slideset).

• What is the importance of padding? When is it done?

Authentication Protocols

• What are the main issues in authentication protocols (slide 3).
Confidentiality: To prevent masquerade and to prevent compromise of session keys, essential identification and session key information must be communicated in encrypted form
Timeliness: To prevent replay attacks

• Know and understand all types of replay attacks (slide 4).
Simple replay: the attacker simply copies a message and replays it later
Repetition that can be logged: an opponent can replay a time stamped message within the valid time window

Repetition that cannot be detected: may arise because the original message could have been suppressed and thus did not arrive at its destination; only the replay message arrives
Backward replay without modification: a reply back to the sender (when using symmetric encryption, the sender cannot easily recognize the difference between messages sent and messages received

• What are the main countermeasures against replay attacks (i.e. nonces and time stamps)?
Sequence number: attach a sequence number to each message in an authentication exchange
Requires each party to keep track of the last sequence number for each claimant has dealt with
Timestamps: party A accepts a message fresh only if the message contains a timestamp that, in A's judgment, is close enough to A's knowledge of current time
Needs synchronized clocks
Nonce: a random number that illustrates the freshness of a session
Party A sends B a nonce and requires that the subsequent response received from B contains the correct nonce value

• Explain the flow of logic in the Needham-Schroeder and Needham-Schroeder-Lowe protocols (slides 8 - 18).

- Original third-party key distribution protocol
- For session between A B mediated by KDC
- Protocol:

1. A->KDC: $ID_A \| ID_B \| N_1$
2. KDC -> A: $E_{Ka}[Ks \| ID_B \| N_1 \| E_{Kb}[Ks \| ID_A]]$
3. A -> B: $E_{Kb}[Ks \| ID_A]$
4. B -> A: $E_{Ks}[N_2]$
5. A -> B: $E_{Ks}[f(N_2)]$

• What is the vulnerability in the Needham-Schroeder protocol? Explain (slides 9-12).
The attack is vulnerable to a replay attack in step 3. If the attacker has compromised an old leaked session key and copy of the message in step 3, the attacker can replay the old message to Bob. Bob will decrypt this message using his private key and think that Alice wants to communicate with him. Bob will send the session key as well as the nonce to the attacker. The attacker will then perform a transformation on the nonce and send it to Bob.

• Why is it challenging to counter replay attacks with time stamps? (slide 12).
A timestamp will assure that the session key has just been generated.

• Explain the flow logic in the Needham-Schroeder Public Key Protocol (15-18).
– Is there a vulnerability?
There is a vulnerability. Trudy can use Alice's trust to impersonate Alice to Bob.

– How do we fix the vulnerability?
Attach the user's identity to the message to verify the message comes from the right user.

• How does one-way authentication work in order to protect email? Does it provide confidentiality? Does it protect against replay attacks?

- Can refine use of KDC but cannot have exchange of nonces:

1. A->KDC: $ID_A \| ID_B \| N_1$
2. KDC -> A: $E_{Ka}[Ks \| ID_B \| N_1 \| E_{Kb}[Ks \| ID_A]]$
3. A -> B: $E_{Kb}[Ks \| ID_A] \| E_{Ks}[M]$

One-way authentication works in order to protect email since only one party needs to participate in the challenge response protocol. This does provide confidentiality since the message is encrypted with the session key. It does not protect against replay attacks because Bob is not available to participate to challenge response protocol.

• Explain why the information transferred through the Web is vulnerable (slides 24-25)? Be able to give examples of attacks. How can we protect against such vulnerabilities?

	Threats	Consequences	Countermeasures
Integrity	• Modification of user data • Trojan horse browser • Modification of memory • Modification of message traffic in transit	• Loss of information • Compromise of machine • Vulnerabilty to all other threats	Cryptographic checksums
Confidentiality	• Eavesdropping on the net • Theft of info from server • Theft of data from client • Info about network configuration • Info about which client talks to server	• Loss of information • Loss of privacy	Encryption, Web proxies
Denial of Service	• Killing of user threads • Flooding machine with bogus requests • Filling up disk or memory • Isolating machine by DNS attacks	• Disruptive • Annoying • Prevent user from getting work done	Difficult to prevent
Authentication	• Impersonation of legitimate users • Data forgery	• Misrepresentation of user • Belief that false information is valid	Cryptographic techniques

All information for HTTP is sent in plaintext. If the data is unencrypted, all your data is sent in plaintext.

• At what layer does SSL run?
Somewhere between the application layer and the transport layer. It was take the data sent by the application layer protocol and it will encrypt it, authenticate it, and pass it onto the transport layer.
SSL/TLS (Secure Socket Layer/Transport Layer Security)

• Know the protocols involved in SSL/TLS and their functions. Do not need to remember the details of the SSL/TLS handshake protocol.
Level 1:
SSL Record Protocol: Provides basic security services to various higher-layer protocols
Level 2:
HTTP: Provides transfer service for Web client/server interaction, can operate on top of SSL.
Three higher-layer protocols: Used in the management of SSL exchanges.

• What services does the SSL/TLS protocol provide?

• Know that SSL fragments messages. Each fragment can be compressed (optionally).

• Does the Heartbleed vulnerability comprise a security flaw in the design of SSL? Explain.

• What are two fundamental ways of protecting passwords (slide 42)?

– Access control.

– One-way functions.

• What are hashing and salting?

• How is the password file protected in UNIX (as discussed in class)?