

A decorative graphic on the left side of the slide. It consists of a blue parallelogram and a light green parallelogram, both tilted at an angle. The blue shape is in the foreground, and the green shape is partially behind it. They are set against a dark blue background with faint, lighter blue diagonal stripes.

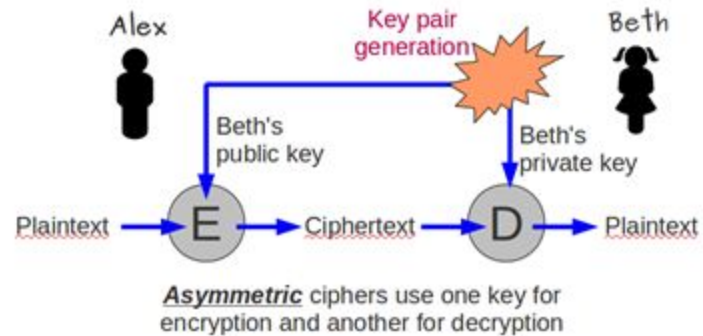
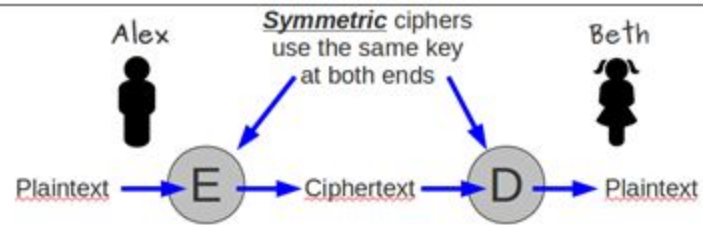
# Encryption



# Symmetric vs Asymmetric Encryptions

Symmetric - For all  $a, b$  in  $X$  if  $a$  is related to  $b$  then  $b$  is related to  $a$ .

Asymmetric - For every  $a, b$  pair in  $X$  if  $a$  is related to  $b$  then  $b$  is not related to  $a$ .





# Block vs Stream Encryption

Block Encryption- When the data being encrypted is placed into blocks of a declared size and padded to fill the room left available.

Stream Encryption- When the data being encrypted is in a one to one relation with with the output encrypted text.



# Modes of Operation

- Electronic Code Book ( ECB )
- Cipher Block Chaining ( CBC )
- Cipher Feedback ( CFB )
- Output Feedback ( OFB )
- Counter ( CTR )

# AES

The Basics





# Advanced Encryption Standard ( AES )

## Background

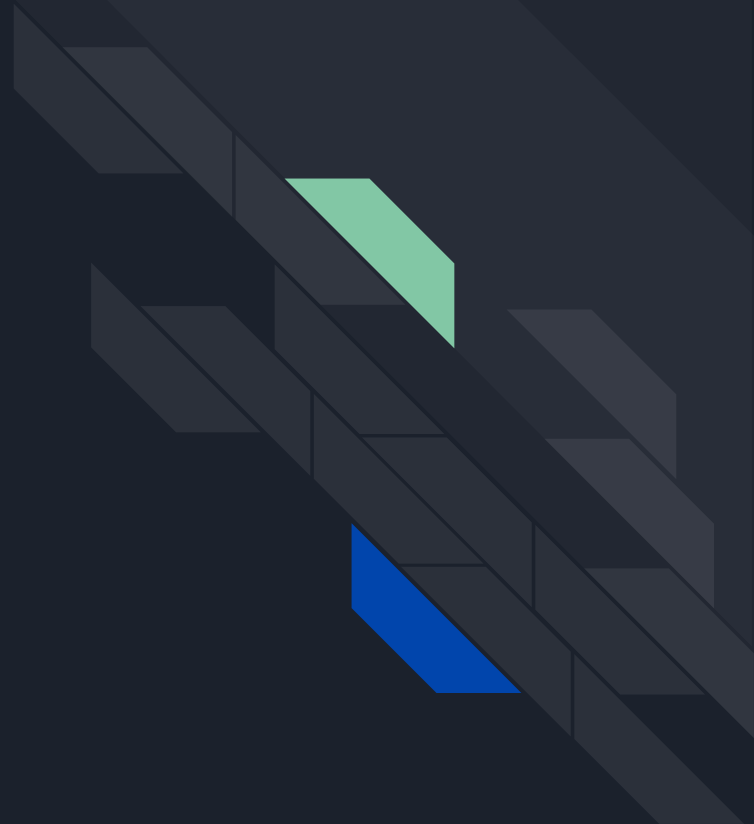
- Encryption Type: Symmetric
- Original Name: Rijndael

## Forms of AES include

- AES-128 ( 10 Rounds Keys )
- AES-192 ( 12 Rounds keys )
- AES-256 ( 14 Rounds Keys )

# AES

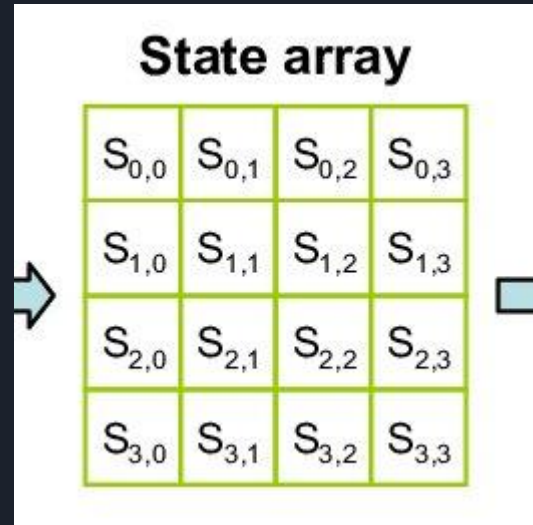
Encryption



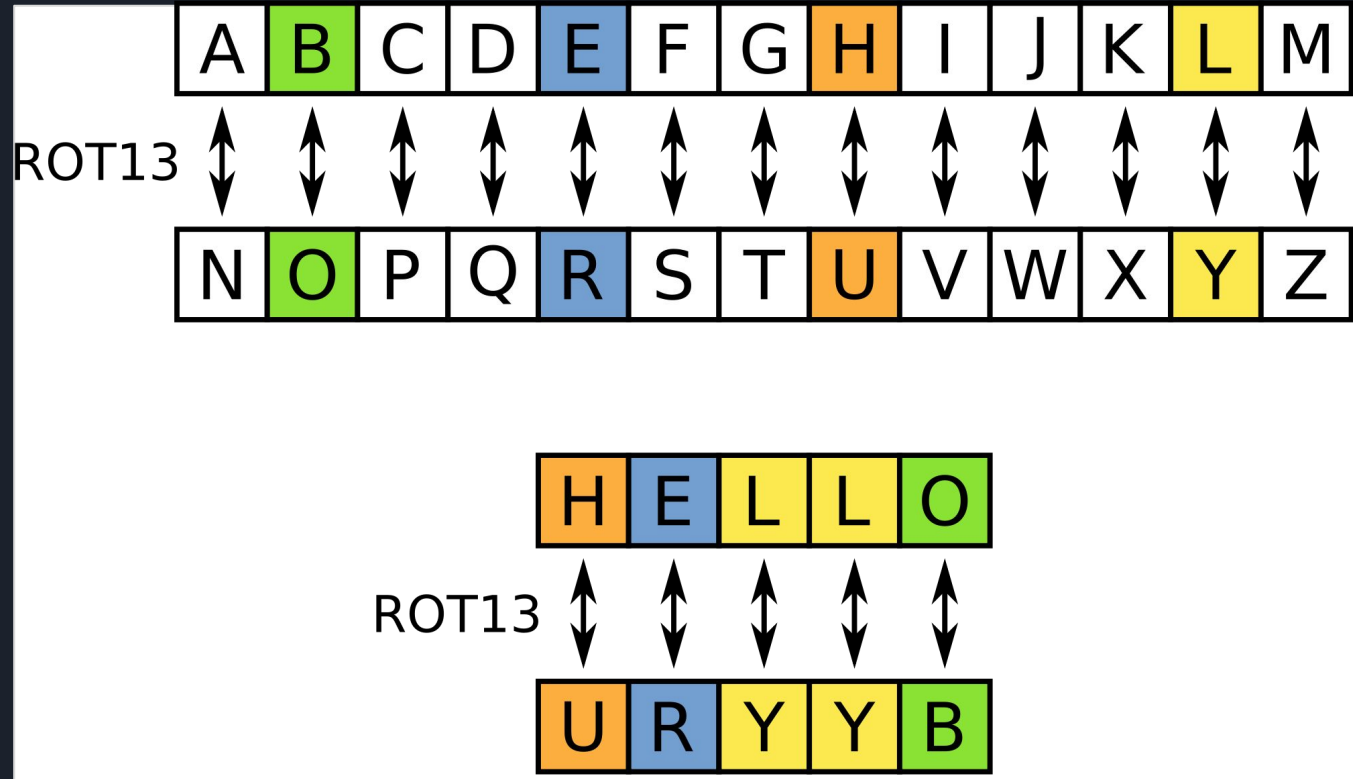


# State Array

- Derived from block data
- Holds state of encryption throughout the process
- Center focus of the encryption



# Substitution box (S-Box)





# Round constant (Rcon)

1 -> 2

2 -> 4

3 -> 8

4 -> 16

5 -> 32

6 -> 64

7 -> 128

8 -> 27

9 -> 54

10 -> 108



# Round Key

- X keys that are derived from the original key, the first round being the key itself,
- Each Key is derived from the last
- Each key contains 4 32 bits words (W)

Formulas:

- $R(x) =$

x rotated right 8 bits

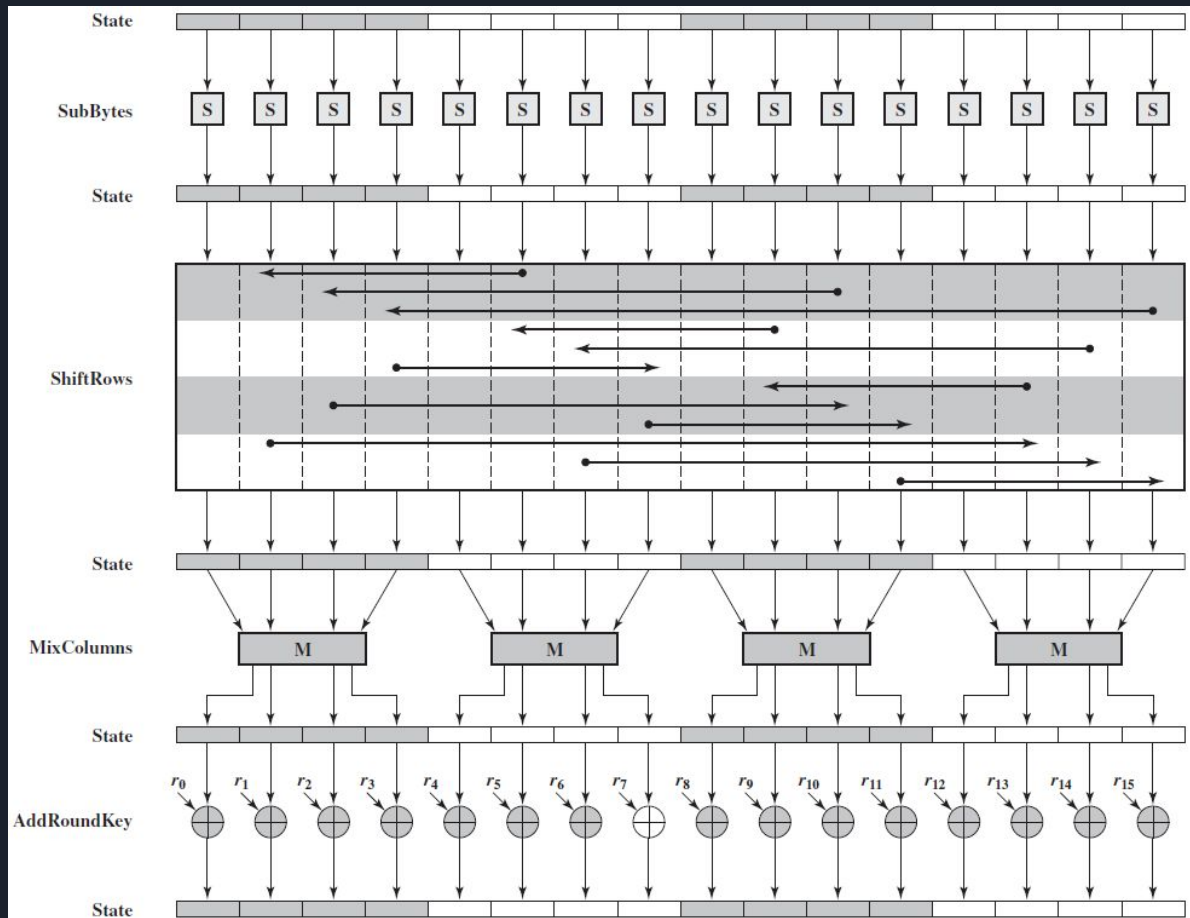
- $\text{Key}(i):W(0) =$

$\text{Key}(i-1):W(0) \text{ XOR } R(\text{Key}(i-1):W(3)) \text{ XOR } \text{Rcon}[i]$

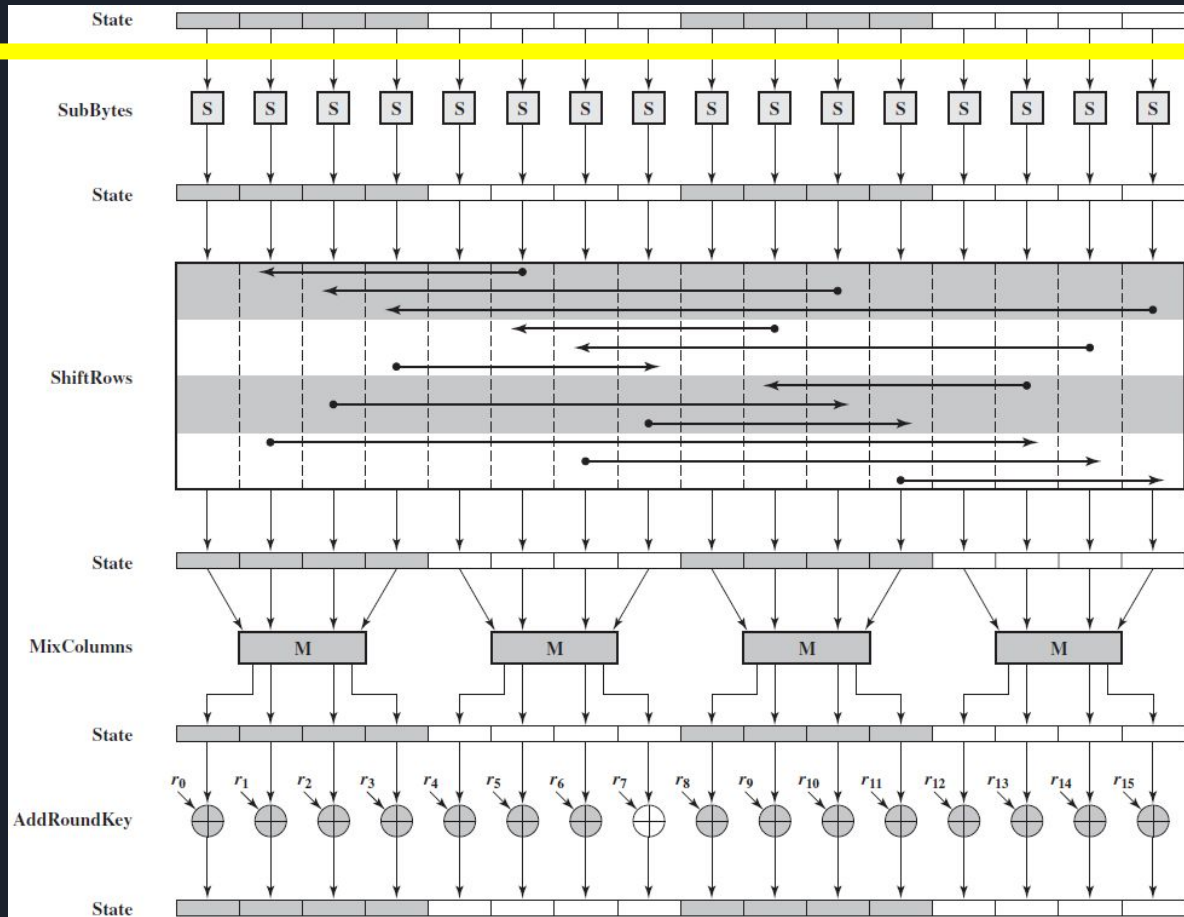
- $\text{Key}(i):W(i) =$

$\text{Key}(i-1):W(i) \text{ XOR } \text{Key}(i):W(i-1)$

# The Process



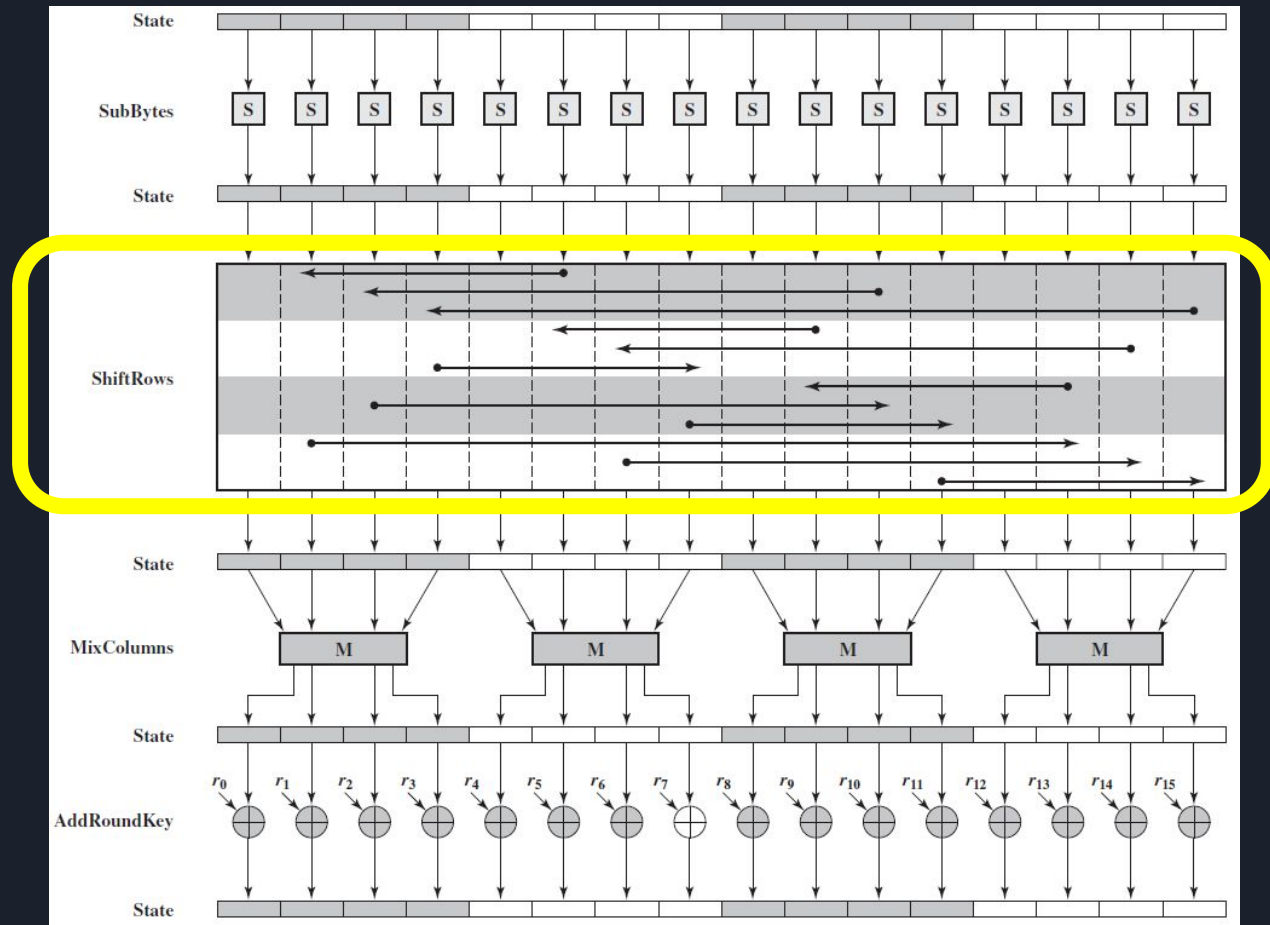
# The Process



# The Process

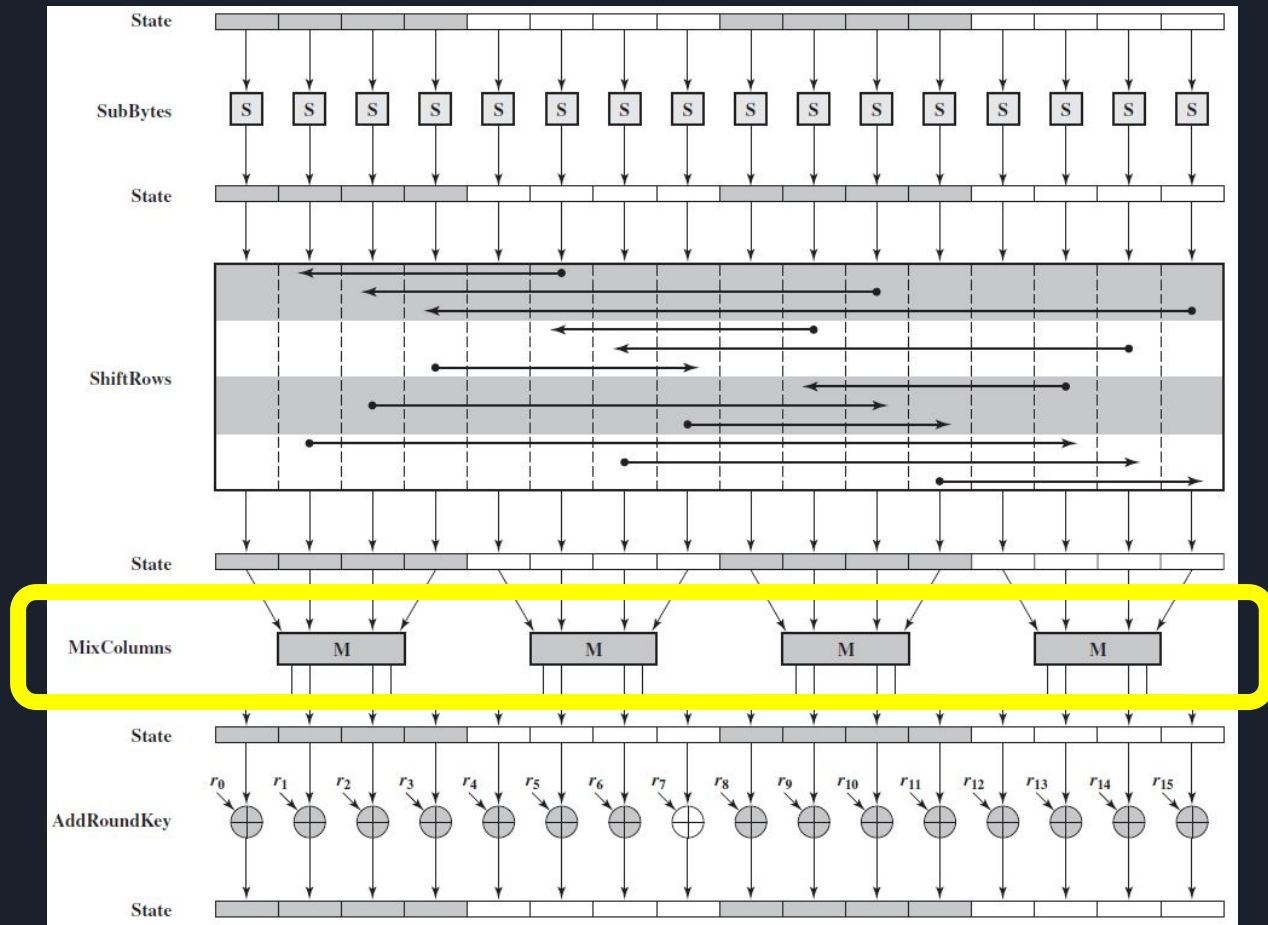


# The Process

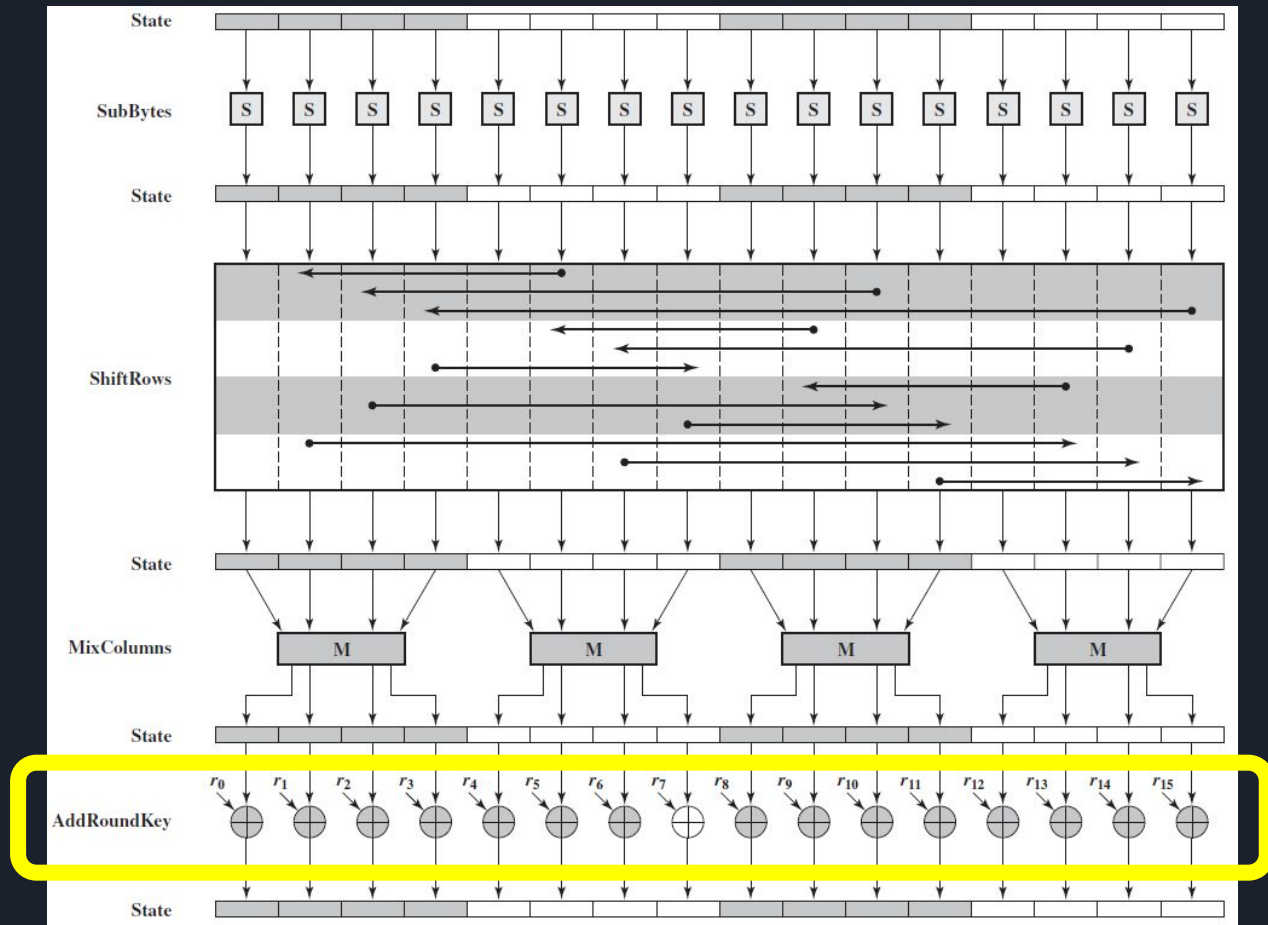




# The Process

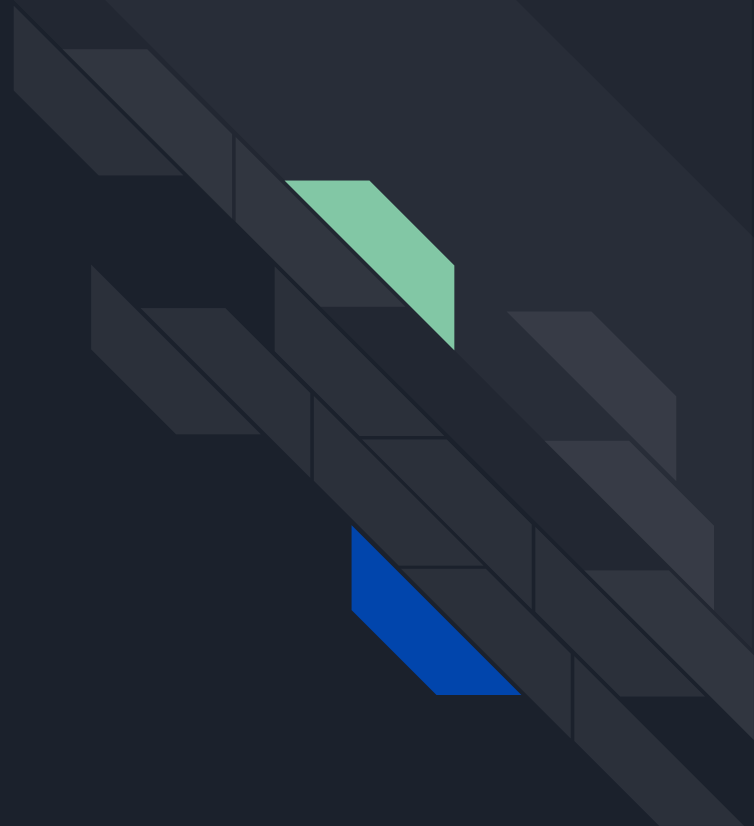


# The Process



# AES

Decryption





# Advanced Encryption Standard Decryption

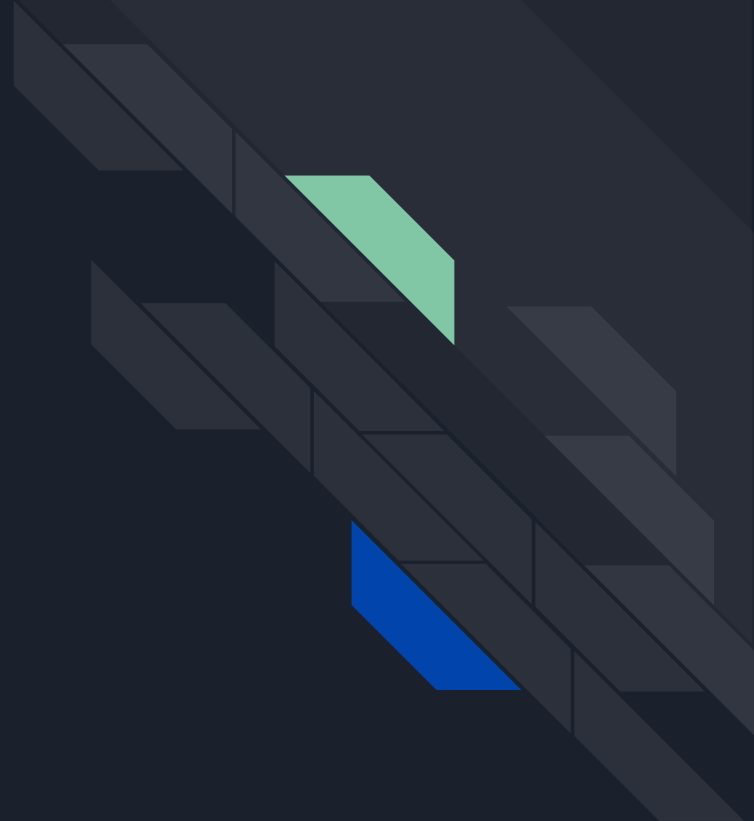
## Methods/ Functions Required

- Inverse Substitute Bytes
- Inverse Shift Rows
- Inverse Mix Columns

Apply all the rules in reverse, since everything is XORed we can use XOR to return to the previous state

# RSA

The Basics





# Rivest, Shamir, Adleman ( RSA)

## Background

- Named after its founders;
  - Ron Rivest,
  - Adi Shamir
  - Leonard Adlema

## Important Details:

- Encryption Type: Asymmetric
- Stream Encryption
- Main focus of RSA is in the modulus and the exponent
- Math oriented
- Limited to 245 bytes at a time



# Key Generation - How does it work

## Private Key:

1. Two prime numbers are picked at random,  $p$  and  $q$
2. The modulus is then found
3. Length is found
4. List all integers from 2 to length
5. Choose one that is a coprime with both the length and the modulus
6. That number is now the exponent

## Equations:

$$\text{modulus} = p * q$$

$$\text{Length(Euler's Totient)} = (q - 1) * (p - 1)$$



# Key Generation - How does it work

## Public Key:

1. Retrieve the private key's modulus and exponent
2. Using the exponent from the private key we can solve for  $d$ , there will be multiple answers for this where the range is infinit. Any of the possibilities may be chosen.
3. Choose one at random, at that becomes the exponent
4. The modulus will match that of the private key's

## The Rule:

$$(D * \text{exponent}) \% \text{length} = 1$$





# RSA Encryption

1. The public key is obtained
2. Each byte( $b$ ) is then put into the following equation where  $e$  is its encrypted value

$$b^{(\text{exponent})} \% \text{modulus} = e$$

3. All the bytes are then placed together again and form a new byte array



# RSA Decryption

1. The private key is obtained
2. The bytes( $b$ ) given in the encrypted text are put through the following equation  
just as before, where  $d$  is the decrypted byte

$$b^{(\text{exponent})} \% \text{modulus} = d$$

3. All the bytes are then placed together again and form a new byte array that can then be turned into a decrypted string



# Data Encryption Standard

- Outdated
- Key with the size of 64 bits
- 8 of the bits from the key are reserved and unused during encryption, due to this  
DES is considered to have 56 bit keys
- Considered easy to break using a brute force attack



# Triple Data Encryption Standard

- The same algorithm that Single Data Encryption Standard uses
- 3 unique keys adding up to a total of 168-bits for a key
- Saved DES from being removed from the eyes of the world



# Blowfish Encryption

- Said to be the fastest algorithm, however it all depends on what mode is used at the time of encryption
- Discovered in 1993 in an attempt to replace DES
- Rated inferior to AES

The image features a dark navy blue background. On the left side, there are two overlapping parallelogram shapes: a blue one in the foreground and a light green one behind it. Both shapes are oriented diagonally, with their longer sides running from the top-left towards the bottom-right. The word "Hashing" is written in a white, sans-serif font to the right of these shapes.

# Hashing



# Message Digest

- Output: 128 bits
- Breaks words down into blocks
- Designed by Rivest, who also played a role in creating RSA



# Secure Hashing Algorithms

- Works in the form of blocks
- Sha Forms are broken into groups called SHA-0, SHA-1, SHA-2, SHA-3
  - SHA-0 contains only SHA-0
  - SHA-1 contains only SHA-1
  - SHA-2 contains SHA-224, SHA-256, SHA-384, SHA-512, SHA-512/224,  
SHA-512/256
  - SHA-3 contains SHA-224, SHA-256, SHA-384, SHA-512, SHAKE128,  
SHAKE256