Symmetric Encryption Algorithms (AES)

Blake Childress and Ran Elgedawy

THE UNIVERSITY OF
TENNESSEE
KNOXVILLE

Questions

- What is the minimum key size supported by AES?
- What is the main difference between block cipher and stream cipher?
- How can you avoid key exhaustion?

Ran Elgedawy

- First year PhD student
- Advisor: Dr. Scott Ruoti
- Research interests: User security and privacy, and Applied machine learning

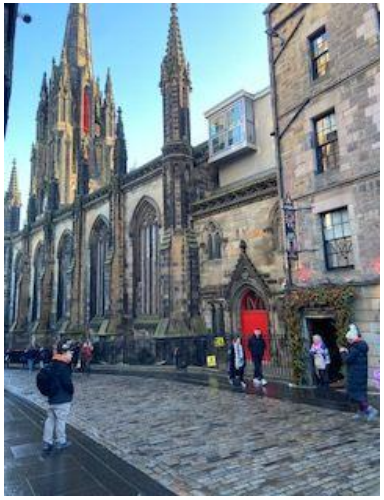
More about me :)

Arab Women Sports Tournament, Sharjah 2018

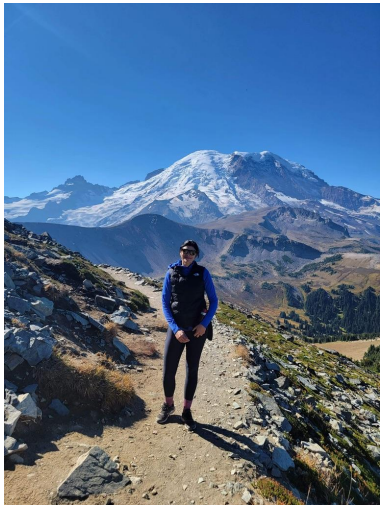
Alexandria Corniche

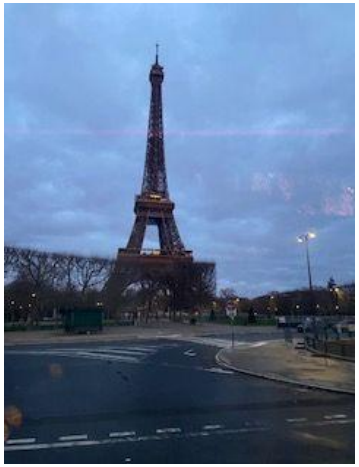
Interests

Edinburgh, Scotland



Mount Rainier



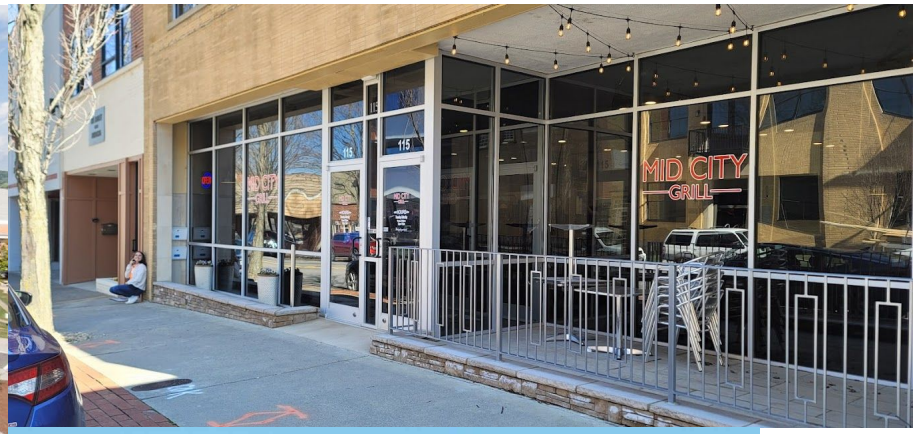


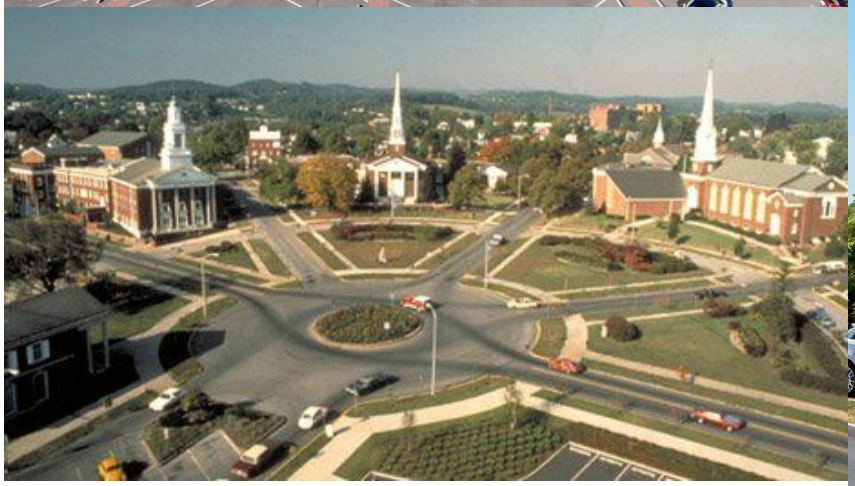
New Year's Eve

Ocoee, TN

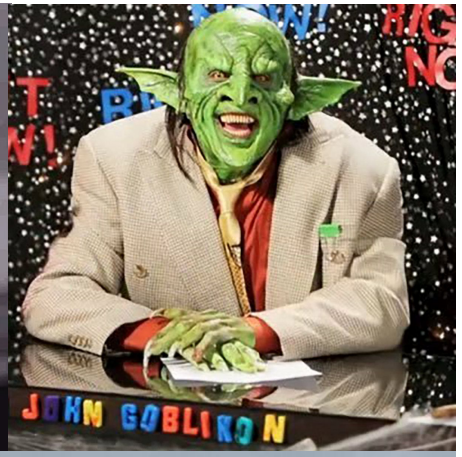
Aspen Snowmass

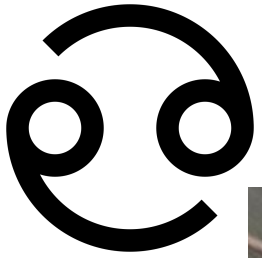
Blake Childress

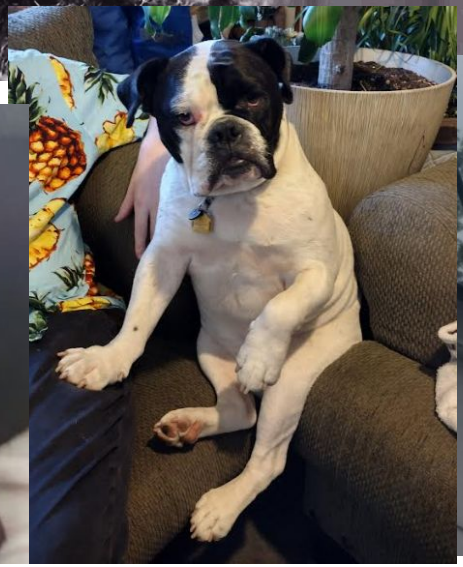


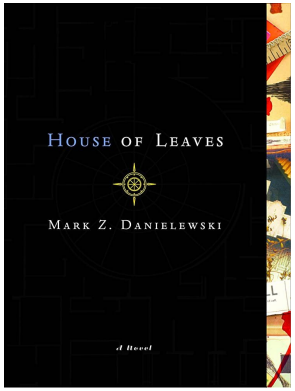


More About Moi







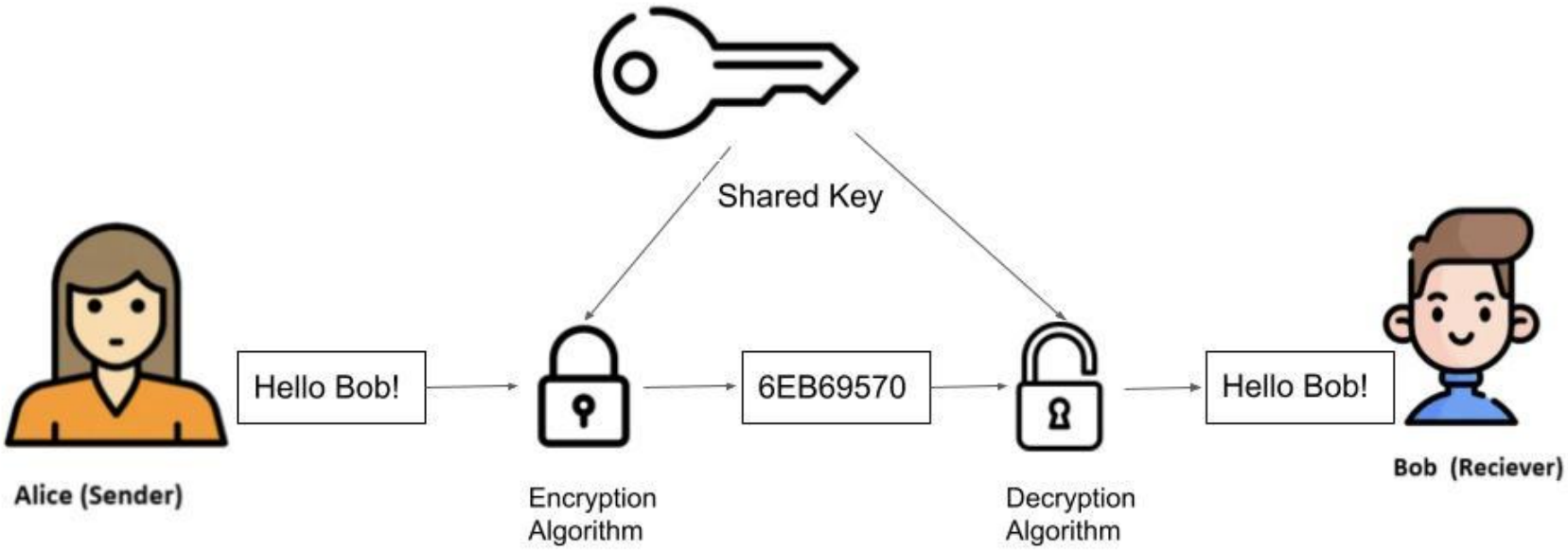


Outline

1. Overview
2. History
3. Background
4. Algorithm details
5. Applications
6. Implementation
7. Open Issues
8. References

Overview

Symmetric Key Encryption



Symmetric Key Encryption

	Data Encryption Standard (DES)	Advanced Encryption Standard (AES)
Developed	1977	2000
Key size	56 bits	128, 192, or 256 bits
Block size	64 bits	128 bits
Security	Proven inadequate	Considered secure

History

History

- In 1997 NIST announced a competition to replace DES for both government and private-sector encryption.
- The algorithm must implement symmetric key cryptography as a block cipher and (at a minimum) support block sizes of 128 bits and key sizes of 128, 192, and 256 bits.

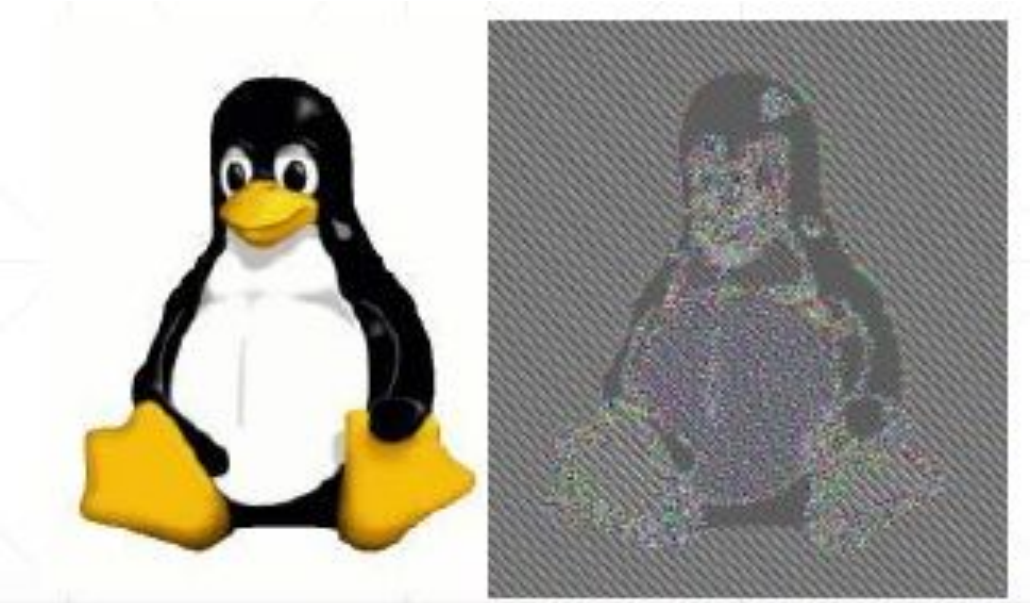
History

- Received 15 proposals from around the world
- On October 2, 2000, NIST selected **Rijndael** (invented by Joan Daemen and Vincent Rijmen) as the AES.

Background

Block Cipher

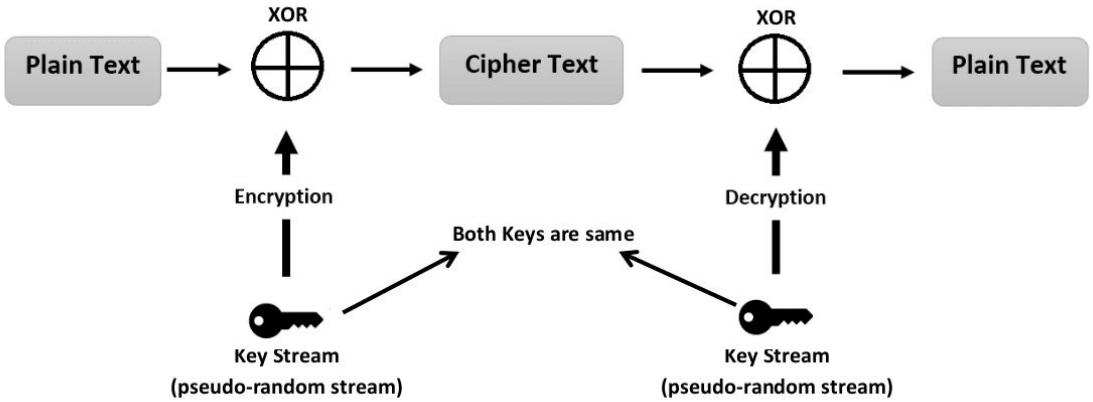
- Operate on fixed number of bits
- Fixed key
- Varying modes of operation



https://lwn.net/Articles/770750/

Stream Cipher

- Combine plaintext and keystream
- Operate on a single digit (bit) at a time
- Useful whenever data comes in unspecified length/quantity (e.g., WiFi)



https://www.javainterviewpoint.com/chacha20-encryption-and-decryption/

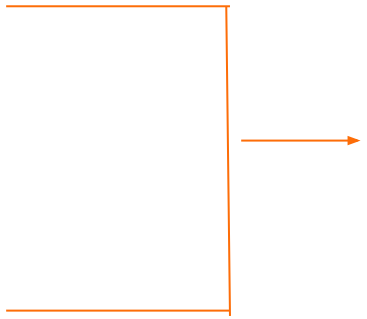
Algorithm details

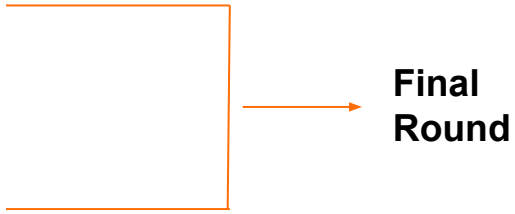
Terms & Definitions

- Substitution permutation network
  - A network takes a block of the plaintext and the key as inputs, and applies several rounds of substitution boxes (S-boxes) and permutation boxes (P-boxes) to produce the ciphertext block
- S-box
  - Non-linear substitution table
- P-box
  - Bit shuffling to permute bits across S-box inputs

Algorithm

State = X
    AddRoundKey(State, Key0)
    for r = 1 to Nr - 1
        SubBytes(State, S-box)
        ShiftRows(State)
        MixColumns(State)
        AddRoundKey(State, Key_r)
    endfor
    SubBytes(State, S-box)
    ShiftRows(State)
    AddRoundKey(State, Key_{Nr})
    Y = State





Algorithm - Add Round key

- Block data (stored in the state array) is passed through an XOR function with the first key generated.
- Resulting state array is input to the next step.

S_0	S_1	S_2	S_3
S_4	S_5	S_6	S_7
S_8	S_9	S_{10}	S_{11}
S_{12}	S_{13}	S_{14}	S_{15}

State

$\oplus$

K_0	K_1	K_2	K_3
K_4	K_5	K_6	K_7
K_8	K_9	K_{10}	K_{11}
K_{12}	K_{13}	K_{14}	K_{15}

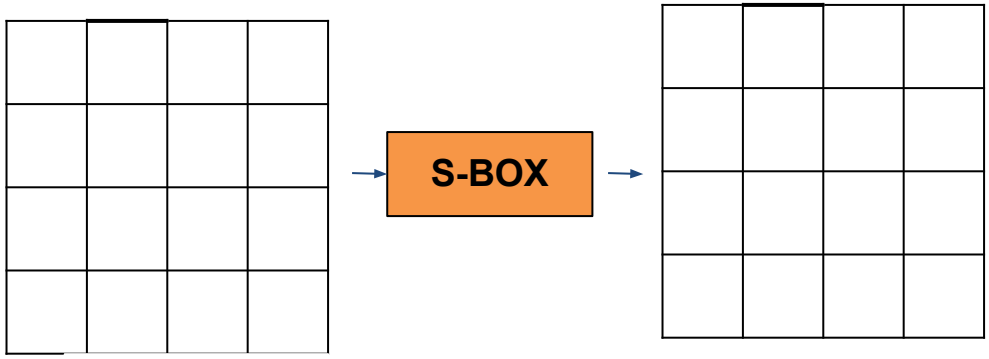
key

Algorithm - Sub Bytes

- Byte substitution using the S-Box
- S-box is represented as a 16x16 array, rows and columns indexed by hexadecimal bits

1001 0101

Row index Column index



Initial State Final State

Algorithm - Shift Rows

S_0	S_4	S_8	S_{12}
S_1	S_5	S_9	S_{13}
S_2	S_6	S_{10}	S_{14}
S_3	S_7	S_{11}	S_{15}

← circular left shift with 0 step

← circular left shift with 1 steps

← circular left shift with 2 steps

← circular left shift with 3 steps

S_0	S_4	S_8	S_{12}
S_1	S_5	S_9	S_{13}
S_2	S_6	S_{10}	S_{14}
S_3	S_7	S_{11}	S_{15}

→

S_0	S_4	S_8	S_{12}
S_5	S_9	S_{13}	S_1
S_{10}	S_{14}	S_2	S_6
S	S_3	S_7	S_{11}

Image taken from: https://zerofruit.medium.com/what-is-aes-step-by-step-fcb2ba41bb20

Algorithm - Mix Columns

- Interpret columns as a vectors of length 4.
- Each column is replaced by another column obtained by multiplying that column with a predefined matrix

2	3	1	1
1	2	3	1
1	1	2	3
3	1	1	2

X

s_0
s_1
s_2
s_3

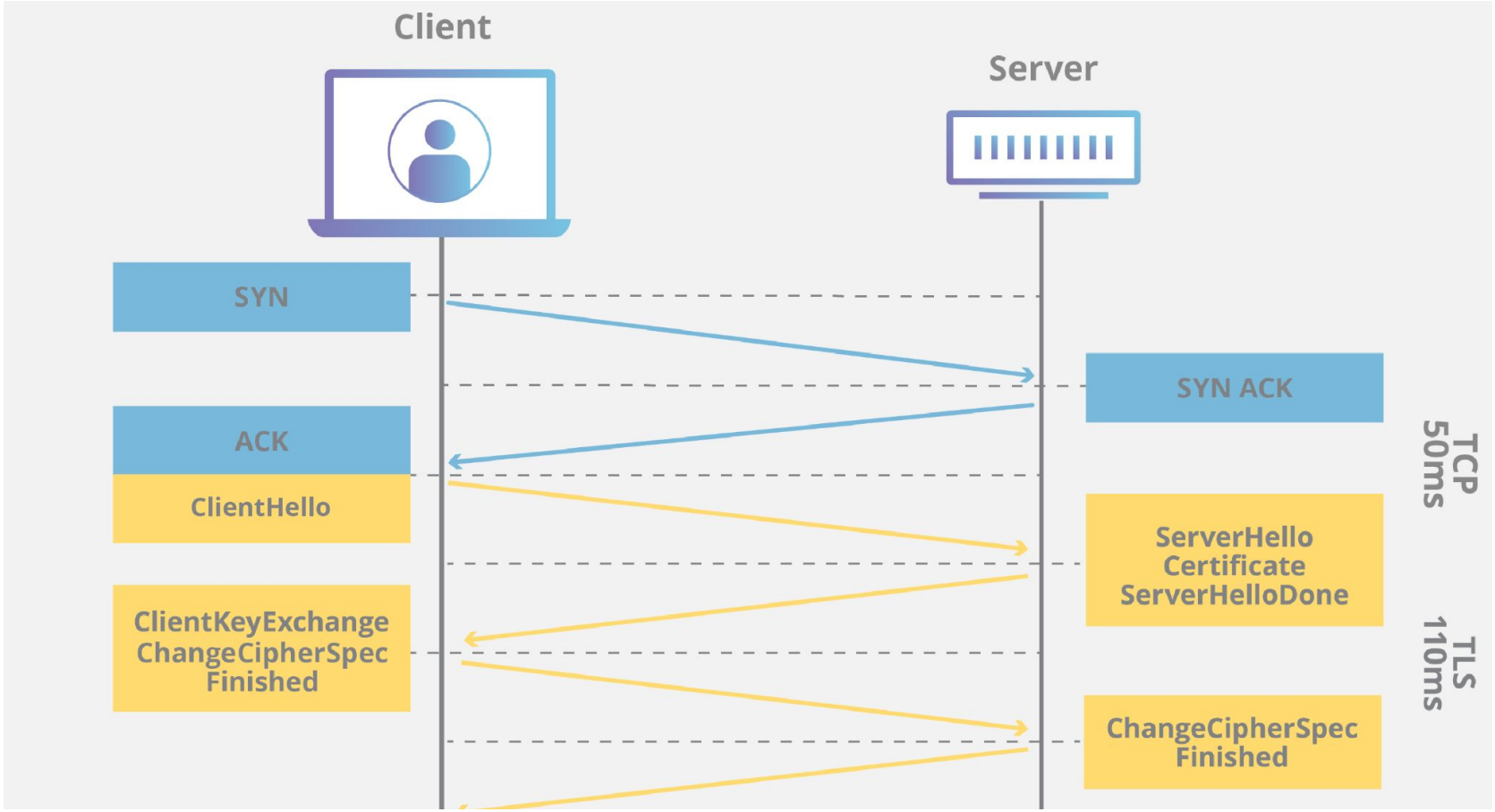
=

s'_0
s'_1
s'_2
s'_3

Image taken from: https://zerofruit.medium.com/what-is-aes-step-by-step-fcb2ba41bb20

Applications

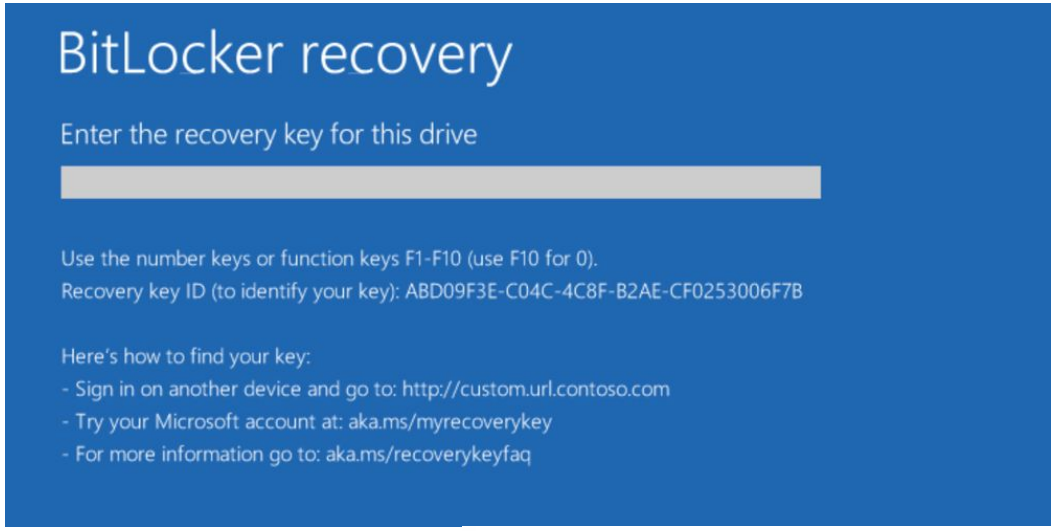
SSL/TLS Handshake



https://www.cloudflare.com/learning/ssl/what-happens-in-a-tls-handshake/

Disk Encryption

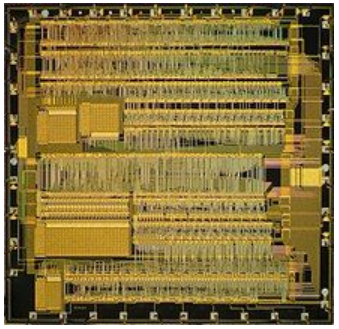
DS8800 storage cabinet



Windows BitLocker and Apple FileVault

Hardware-based Encryption

IBM 4758 Cryptographic Module

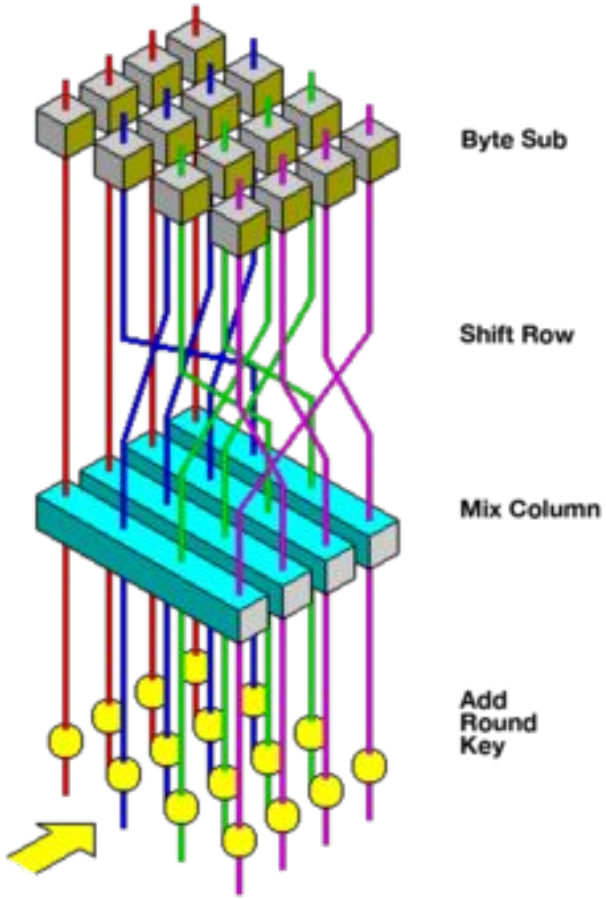


Example cryptoprocessor (top) and crypto Accelerator (bottom)

Implementation

AES

- Variant of Rijndael block cipher
- Fixed block size of 128 bits
- Key length may be 128, 192, or 256 bits



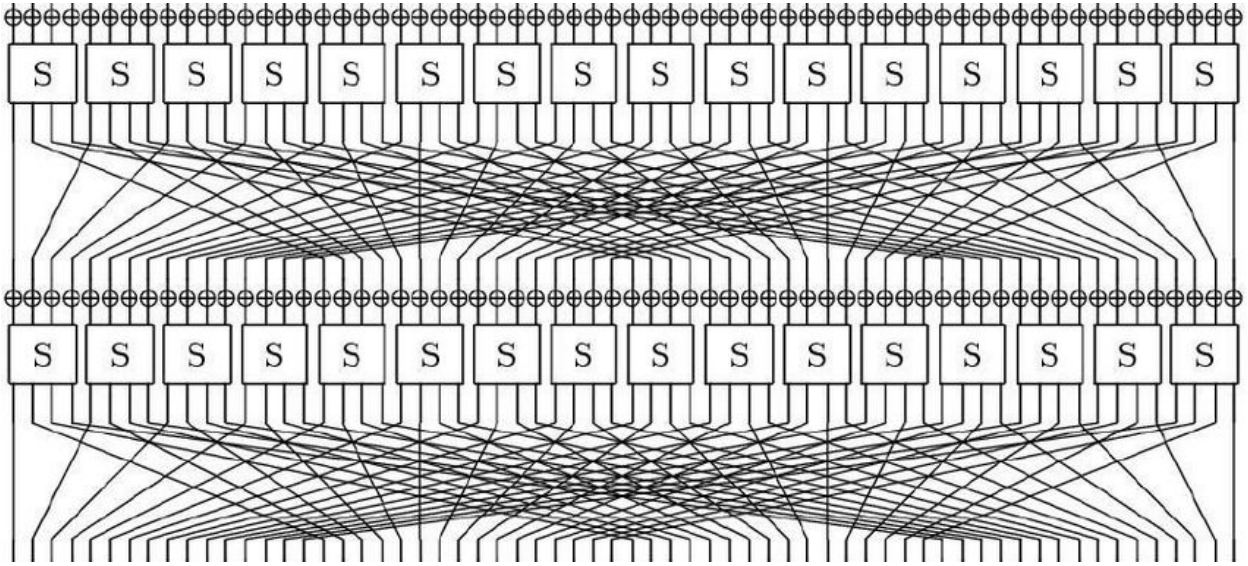
https://en.wikipedia.org/wiki/Advanced_Encryption_Standard

My AES Implementation

- C++
- Followed FIPS 197 AES standard
- ~16 kB source file

PRESENT

- Lightweight block cipher
- Published 2007
- Block size of 64 bits and key size of 80 or 128 bit



My PRESENT-80 Implementation

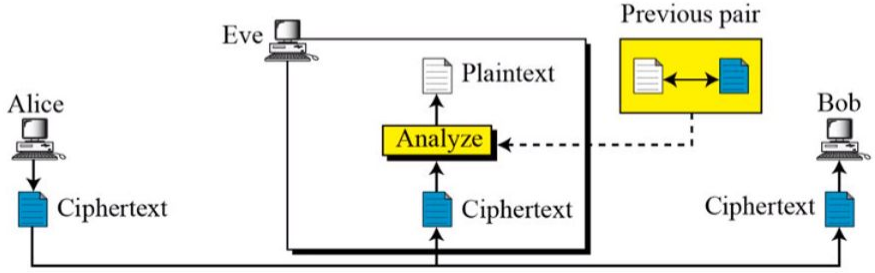
- Verilog (HDL)
- https://github.com/saiedhk/PresentCryptoEngine
- ~3.4 kB

Open Issues

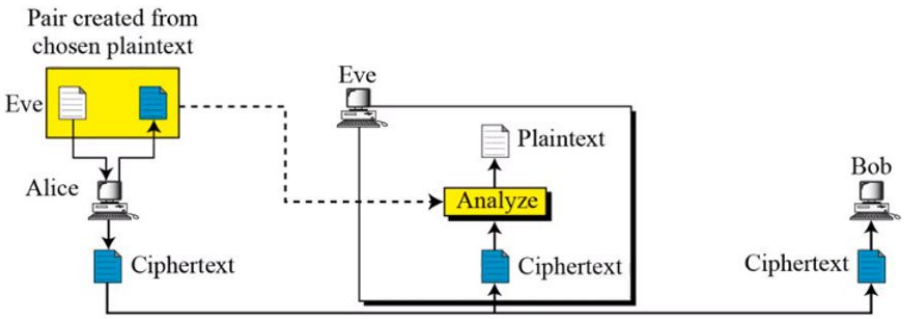
Attacks

- Known-plaintext
- Chosen-plaintext
- Differential cryptanalysis
- Linear cryptanalysis

Known-Plaintext Attack



Chosen-Plaintext Attack

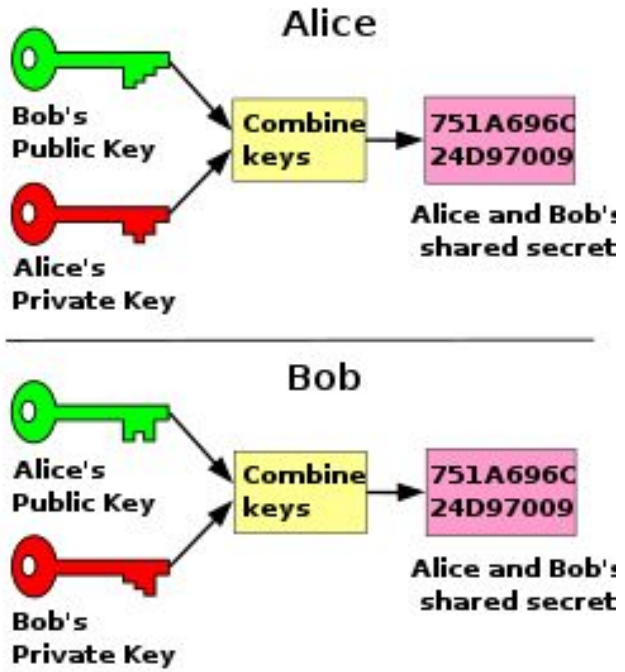


Key Exhaustion

- If we use the same key to encrypt data, it may be possible to derive it after so much information is processed
- Attacker requires access “enough” encrypted data
- Rotate/make new keys!

Key Management

- Must keep symmetric-key secure
- Mitigated with Diffie-Hellman or similar asymmetric protocol



Symmetric and Asymmetric encryption

References

- Dr. Scott Ruoti's AES notes
- https://en.wikipedia.org/wiki/Advanced_Encryption_Standard
- https://en.wikipedia.org/wiki/PRESENT
- https://nvlpubs.nist.gov/nistpubs/fips/nist.fips.197.pdf
- https://soatok.blog/2020/12/24/cryptographic-wear-out-for-symmetric-encryption/

Thank you.
Any questions?