Symmetric-Key Cryptography

CS 161: Computer Security

Prof. Raluca Ada Popa

Feb 5, 2019

Announcements

- Hmw 1 due today, midnight
- Proj 1 due Feb 12
- Hmw 2 (crypto) out today, due Feb 17
- Midterm 1: Feb 21 7-9pm, will cover memory safety and all of crypto

Special guests

- Alice 
- Bob 
- The attacker (Eve - “eavesdropper”, Malice)  
- Sometimes Chris too

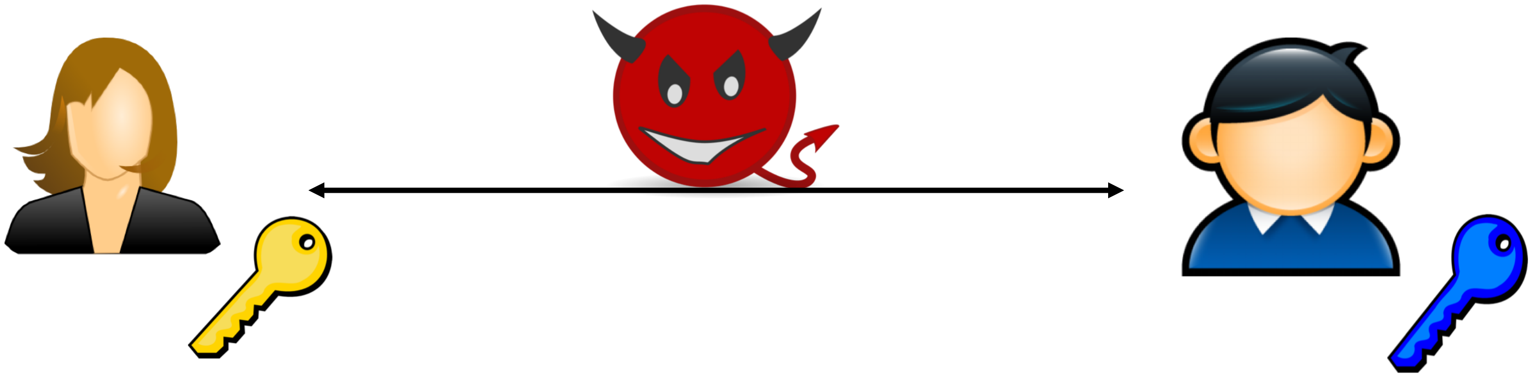
Cryptography

- Too narrow definition: secure communication over insecure communication channels
- Broad definition: a way to provide formal guarantees in the presence of an attacker

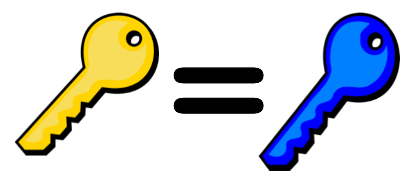
Three main goals

- Confidentiality: preventing adversaries from reading our private data,
- Integrity: preventing attackers from altering some data,
- Authenticity: ensuring that the expected user created some data

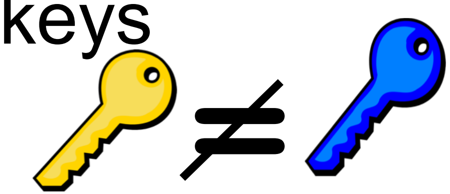
Modern Cryptography



- Symmetric-key cryptography
  - The same secret key is used by both endpoints of a communication



- Public-key (asymmetric-key) cryptography
  - Sender and receiver use different keys



Why are we studying symmetric key and asymmetric key cryptography?

- Very widely used
- Basis of many security mechanisms
  - For example, your online communication are secured using these tools (foundations to TLS)
  - We will learn
    - how to construct these crypto tools,
    - what security they provide,
    - how to use them to construct TLS, and then
    - how TLS is used in your usage of the web.

On Projector: Symmetric-key Cryptography

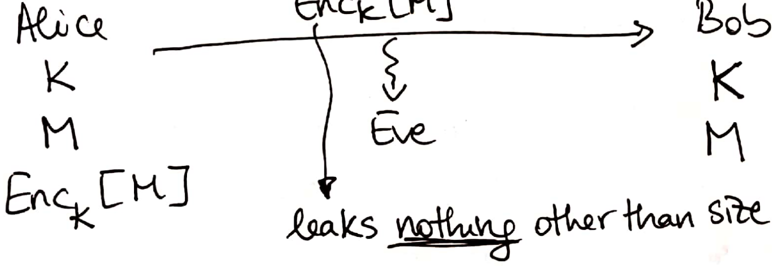
Whiteboard & notes:

- Symmetric encryption definition
- Security definition
- One time pad (OTP)
- Block cipher

Symmetric encryption schemes

Goal: confidentiality

1



Three algorithms:

$Keygen() \to K$

$Enc(K, M) = Enc_K(M) = C$ ciphertext

plaintext

$Dec(K, C) = M$

Correctness: can decrypt to original value

$\forall K, \forall M, \forall C \leftarrow Enc_K(M): Dec(K, C) = M$

Security:

A cryptosystem should be secure even if the attacker knows all its algorithms, <u>except the key</u>

Not enough to say that attacker cannot ~~decrypt~~ original value

Attacker should not learn even <u>partial</u> information.

└ cannot <u>distinguish</u> which of 2 ~~ratio~~ messages M_0, M_1 are encrypted in a ciphertext C.

IND-CPA (Indistinguishability under chosen plaintext attack)

Challenger		**Attacker**
$K \leftarrow \text{Keygen}()$	$\xleftarrow{M}$	can be equal
	$\xrightarrow{Enc_K(M)}$	↻
random bit b	$\longleftarrow$	M_0, M_1 (same length) } challenge
$Enc_K(M_b)$	$\longrightarrow$	$Enc_K(M_b)$
	$\xleftarrow{M}$
	$\xrightarrow{Enc_K(M)}$	↻
		b'

Attacker wins game iff $b' = b$

An enc. scheme is IND-CPA secure iff

$\forall$ Attackers, $Pr[A \text{ wins}] \leq 1/2 + negl \rightarrow \frac{1}{2^{128}}$ ↗ fewer atoms in the universe than 2^{128}

1) $Enc_K(M) = 2 \cdot M$ not secure

2) $Enc_K(M) =$ random bits perfectly secure

3) $Enc_K(m) = (m + K) \bmod p$ public

NO, because deterministic

4) $Enc_K(M) = 4$

Ingredients for sym-key enc. scheme [OTP
block ciphers

OTP (One-time pad)

Alice

Bob

$K = k_1 \dots k_n$

$K = k_1 \dots k_n$

Keygen = generates n random bits

$M = M_1 \dots M_n$

$\xrightarrow{C = Enc_K(M)}$

$Dec_K(C) = C \oplus K$

$Enc_K(M) = M \oplus K$

$M \oplus K$

$11 \oplus 10 = 01$

Security holds if you encrypt only once using same key.

NOT IND-CPA

Claim: Given one ciphertext C, M_0 or M_1 are equally likely

$$C = \underbrace{C \oplus M_0}_{k_0} \oplus M_0$$

$$C = \underbrace{C \oplus M_1}_{k_1} \oplus M_1$$

Assumption: Adv only gets one ciphertext

Ch		Adv
k	$\xleftarrow{M_0, M_1}$	
b	$\xrightarrow{C = M_b \oplus k}$	
		b'

Bayes

$$\Pr[M_0 \mid C] = \frac{\Pr[M_0 \wedge C]}{\Pr[C]}$$

$C = M_0 + $

M_0 was encrypted by challenger (chose $b=0$)

C is ciphertext

$$= \frac{\Pr[M_0 \wedge k = C \oplus M_0]}{\Pr[C]}$$

$$= \frac{1/2 \cdot 1/2^n}{1/2^n} = 1/2$$

Caveat 1) Do not reuse a one-time pad ; 2) can only encrypt n bits

If you encrypt more than one ciphertext,

NO SECURITY.

$M_0 \oplus K = C_0$

$M_1 \oplus k = C_1$

$C_0 \oplus C_1 = M_0 \oplus M_1$

Scanned by CamScanner

Ingredient #2: Block ciphers

$E: \{0,1\}^k \times \{0,1\}^n \to \{0,1\}^n$

E_K : permutation : one-to-one | bijective
encipher

E_K : $\Rightarrow$ deterministic $\Rightarrow$ not IND-CPA

$M \longrightarrow C$

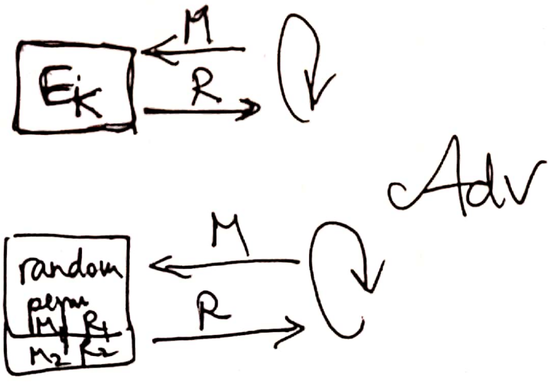
$M' \not\longrightarrow$

D_K : inverse E_K
decipher

Correctness: $D_K \left(E_K(M) \right) = M$

Security: behaves like a random permutation

Ch
random K



Block cipher is secure if $\forall$ Adv
$\Pr[\text{Adv guesses box}] \leq 1/2 + \text{negl}$

Random fact about … Nick

Grew up in Huntington Beach, CA

“surf city”

Took him 8 years to finish grad school because he didn’t want to work for a living.

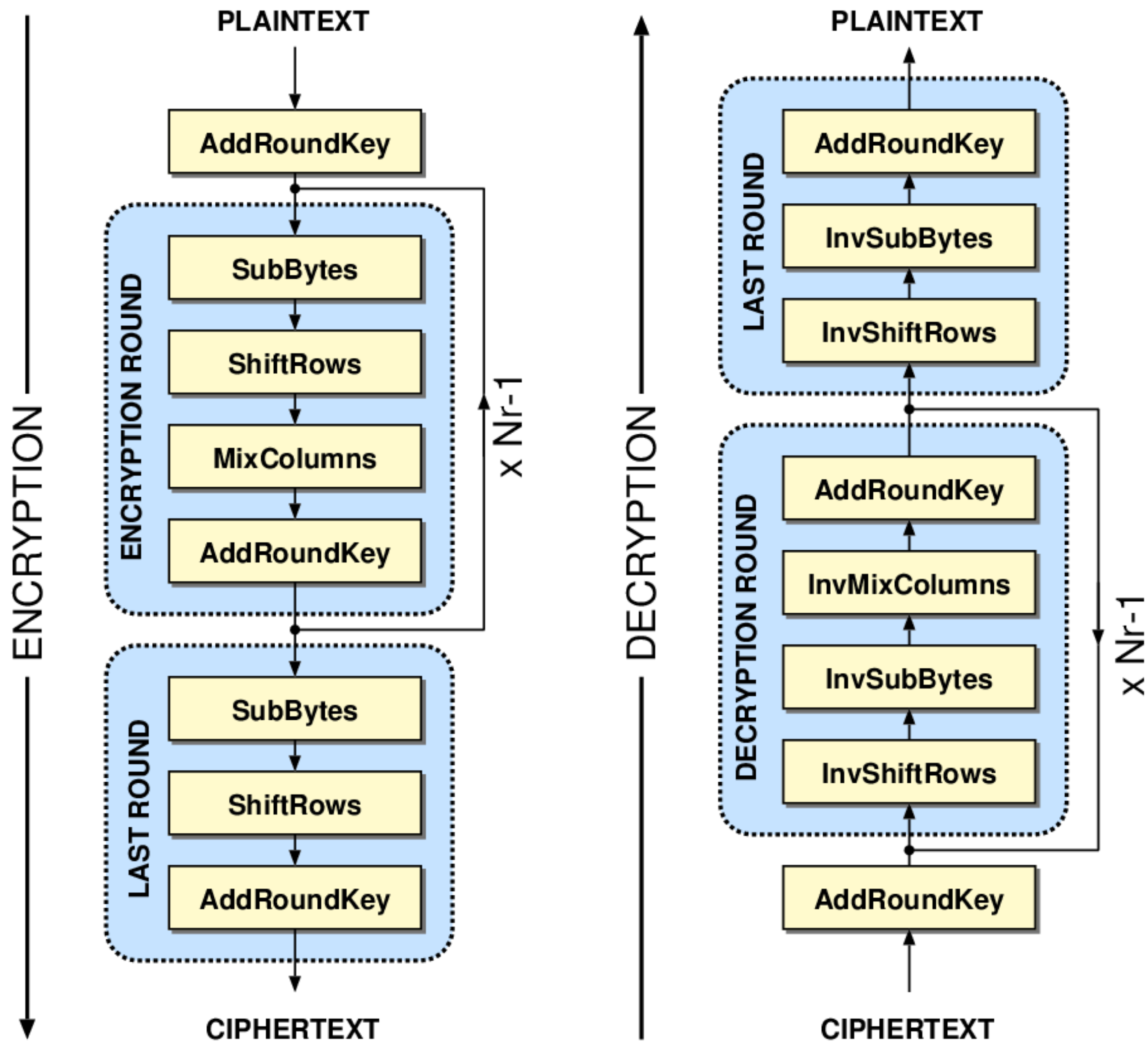
So he teaches you because he really enjoys it.

2min break

Advanced Encryption Standard (AES)

- Block cipher developed in 1998 by Joan Daemen and Vincent Rijmen
- Recommended by US National Institute for Standard and Technology (NIST)
- Block length n = 128bits, key length k = 256bits

AES ALGORITHM

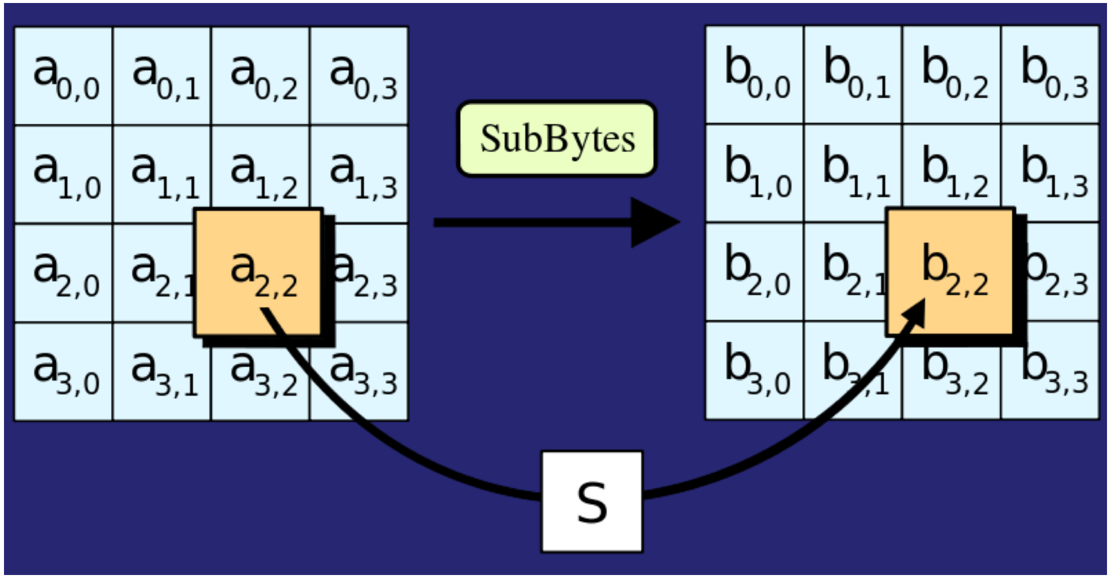


- 14 cycles of repetition for 256-bit keys.

You don't need to understand why AES is this way, just get a sense of its inner workings

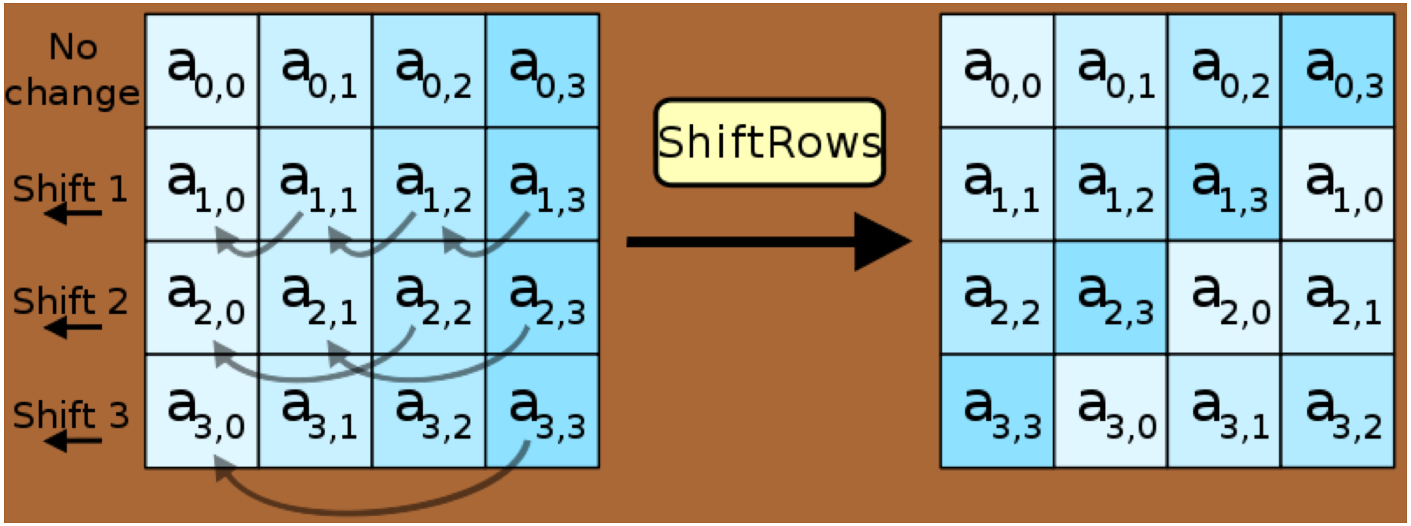
Algorithm Steps - Sub bytes

- each byte in the *state* matrix is replaced with a SubByte using an 8-bit substitution box
- $b_{ij} = S(a_{ij})$

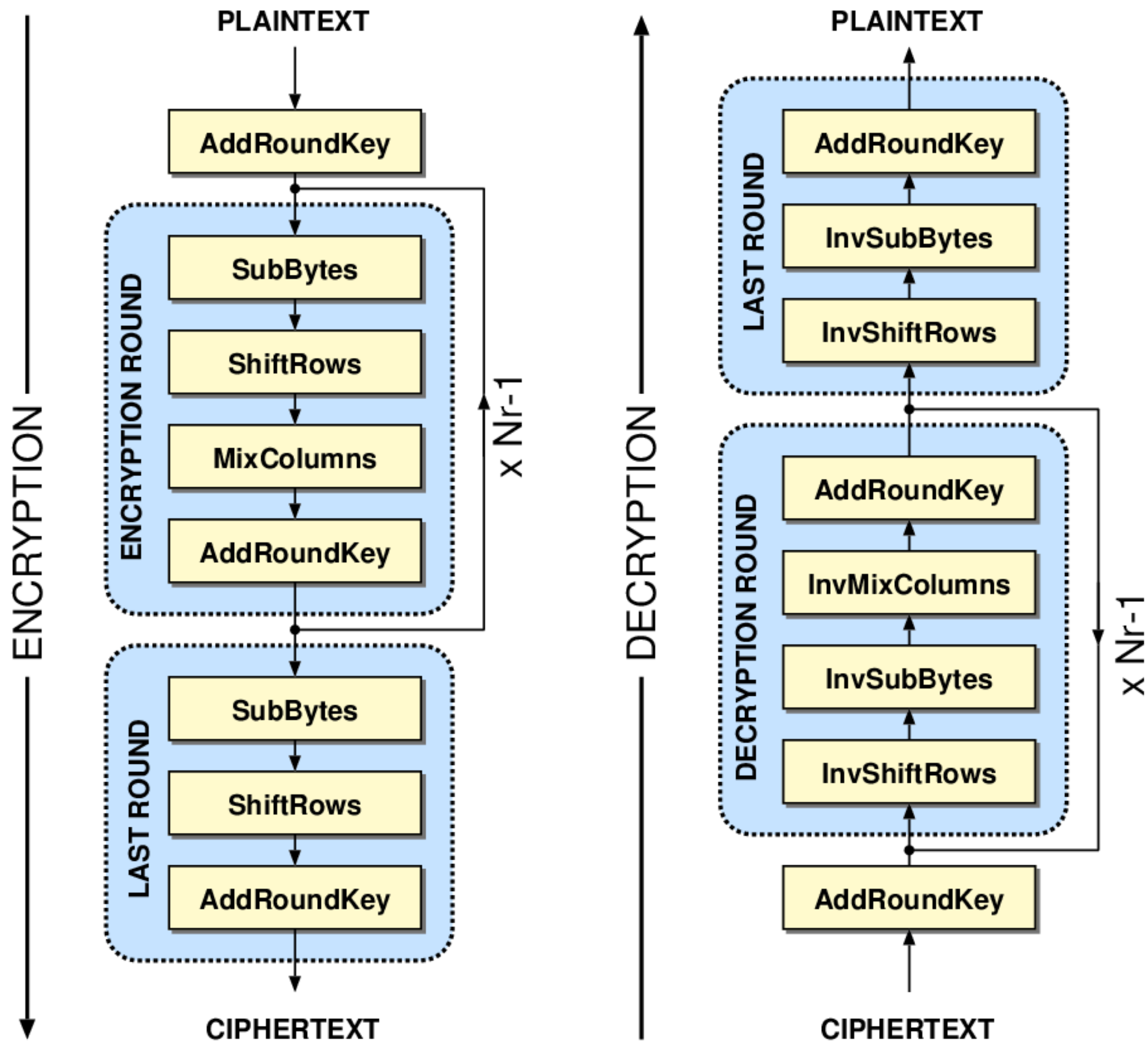


Shift Rows

- Cyclically shifts the bytes in each row by a certain offset
- The number of places each byte is shifted differs for each row



AES ALGORITHM



- The key gets converted into round keys via a different procedure
- 14 cycles of repetition for 256-bit keys.

You don't need to understand why AES is this way, just get a sense of its inner workings

AES slides, credit Kevin Orr

Why secure?

- Not provably secure but we assume it is
- By “educated” belief/assumption: it stood the test of time and of much cryptanalysis (field studying attacks on encryption schemes)
- Various techniques to boost confidence in its security
- If we were to even have something probably secure, P is not NP

Uses

- Government Standard
  - AES is standardized as Federal Information Processing Standard 197 (FIPS 197) by NIST
  - To protect classified information
- Industry
  - SSL / TLS
  - SSH
  - WinZip
  - BitLocker
  - Mozilla Thunderbird
  - Skype

Used as part of symmetric-key encryption or other crypto tools