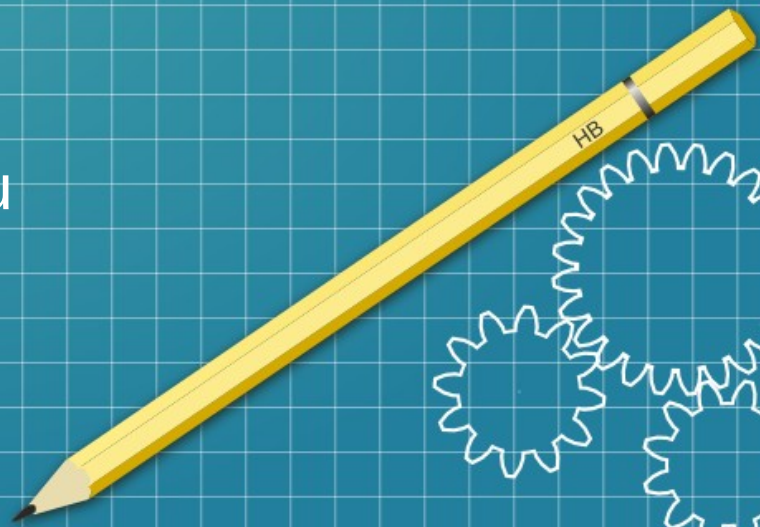
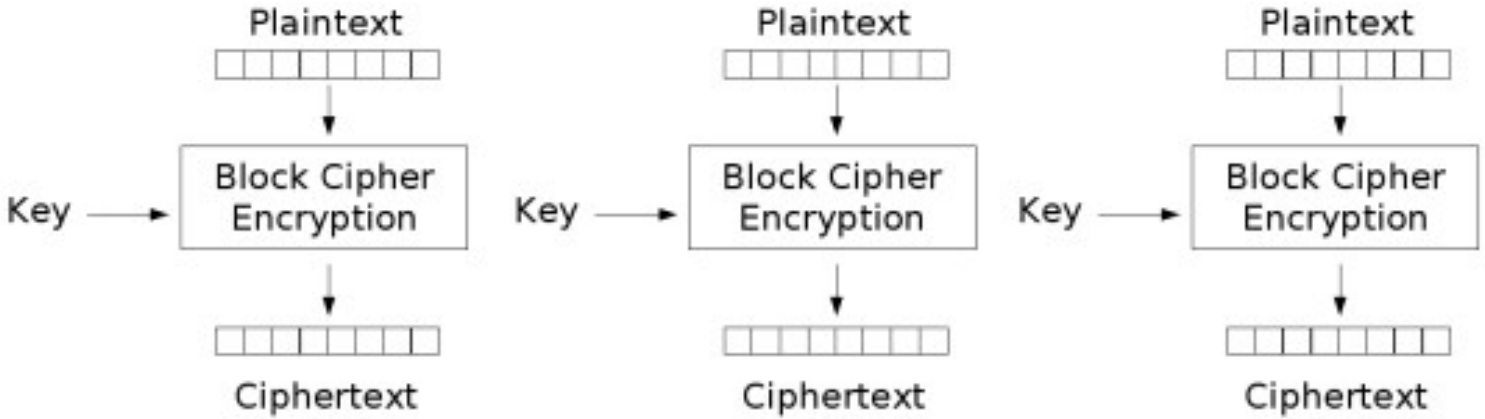
Block cipher modes, stream ciphers, and message authentication

CSE 539
jedimaestro@asu.edu

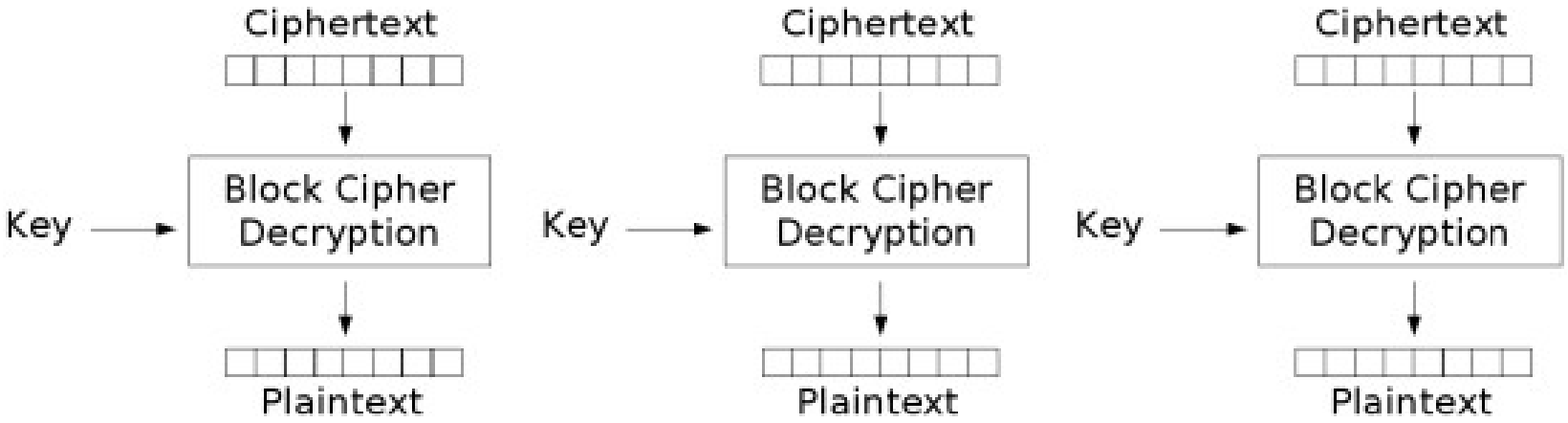


Overview

- Review (ECB and CBC mode)
- Why different block cipher modes?
- Stream ciphers (RC4 and WEP as an example)
- CTR mode
- Message authentication
  - HMAC
  - Galois Counter Mode

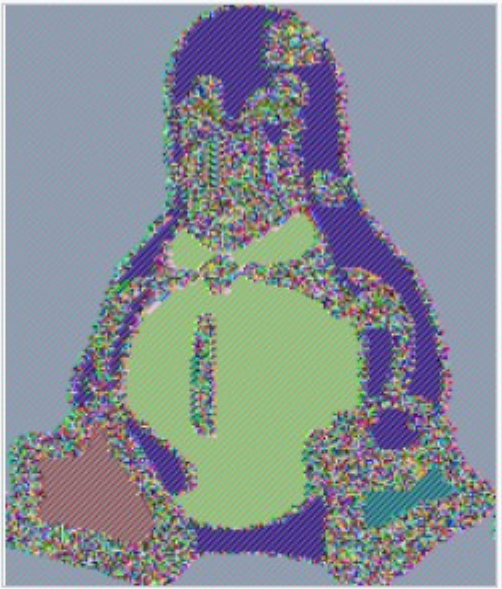


Electronic Codebook (ECB) mode encryption

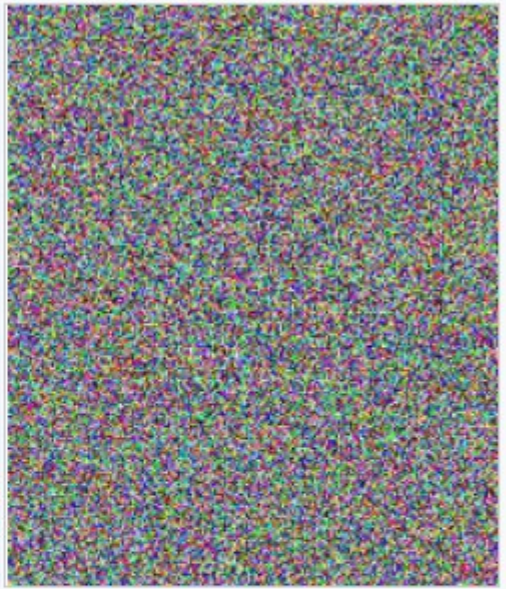


Electronic Codebook (ECB) mode decryption

Original image

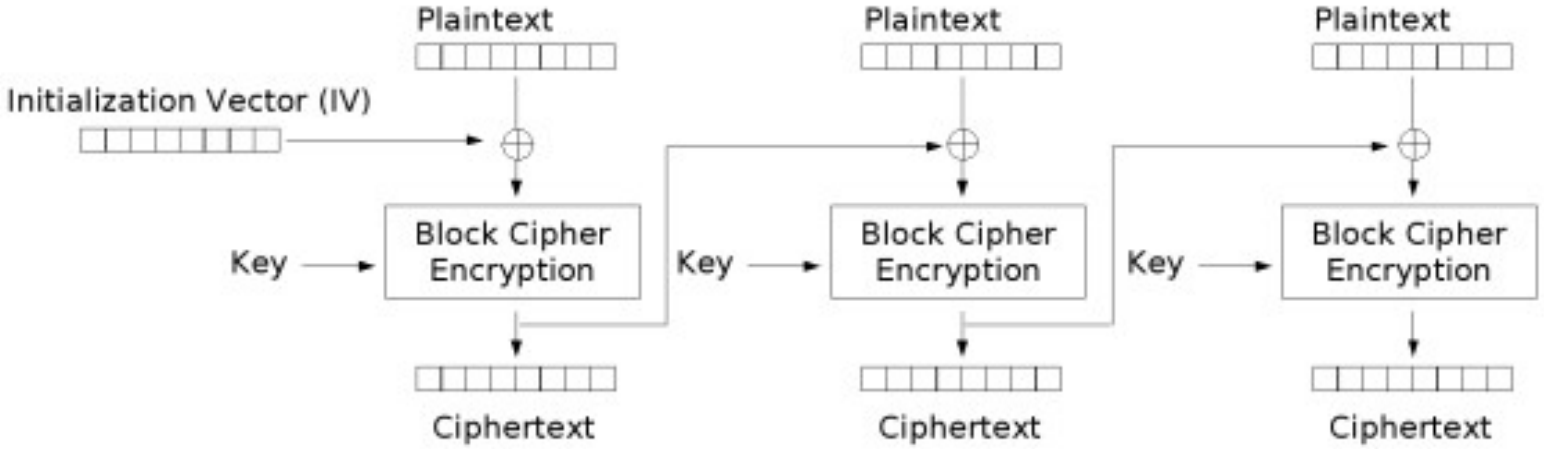


Encrypted using ECB mode

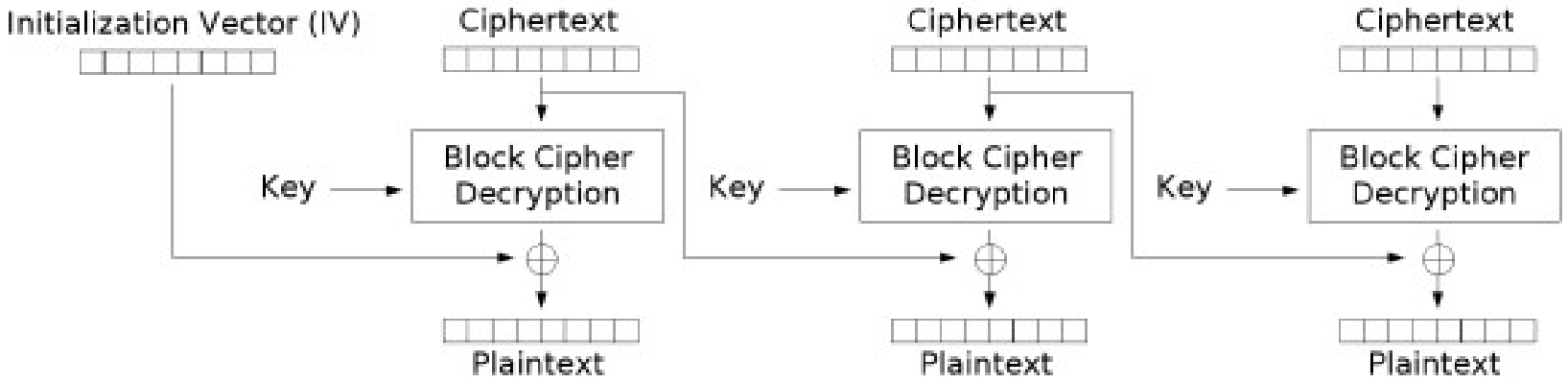


Modes other than ECB result in pseudo-randomness

The third image is how the image might appear encrypted with CBC, CTR or any of the other more secure modes—indistinguishable from random noise. Note that the random appearance of the third image does not ensure that the image has been securely encrypted; many kinds of insecure encryption have been developed which would produce output just as "random-looking".[*citation needed*]



Cipher Block Chaining (CBC) mode encryption

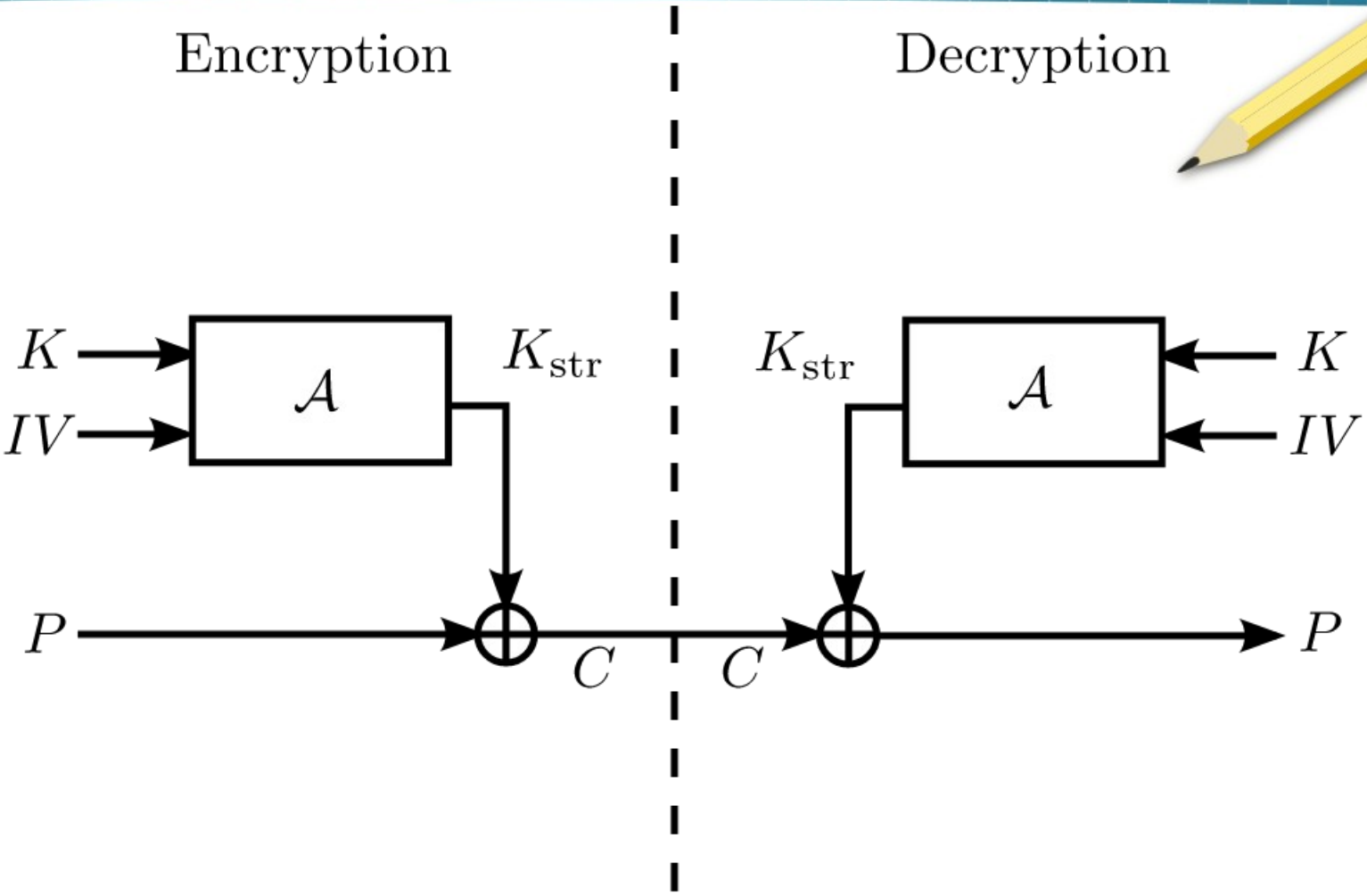


Cipher Block Chaining (CBC) mode decryption

Why different block cipher modes?

- Simplicity, security
  - API and lack of education drives many decisions about cipher modes
- Malleability
  - Can be good (error correction, deniability)
  - Can be bad
- Parallelism
- Random access
- Authentication
- Patents?

Stream ciphers...



https://commons.wikimedia.org/wiki/File:Stream_cipher.svg

WEP encryption

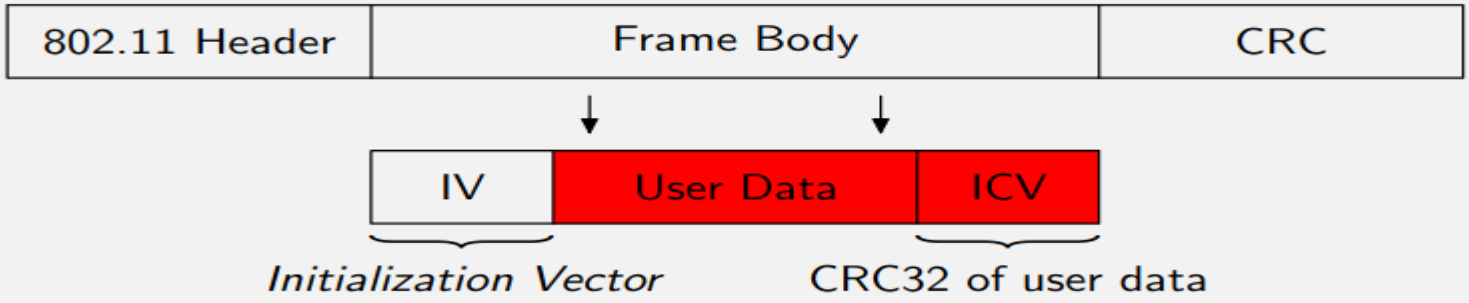
“Wired Equivalent Privacy”

-Have to be physically in a building to plug in, have to know the passphrase to join WiFi (or do you?)

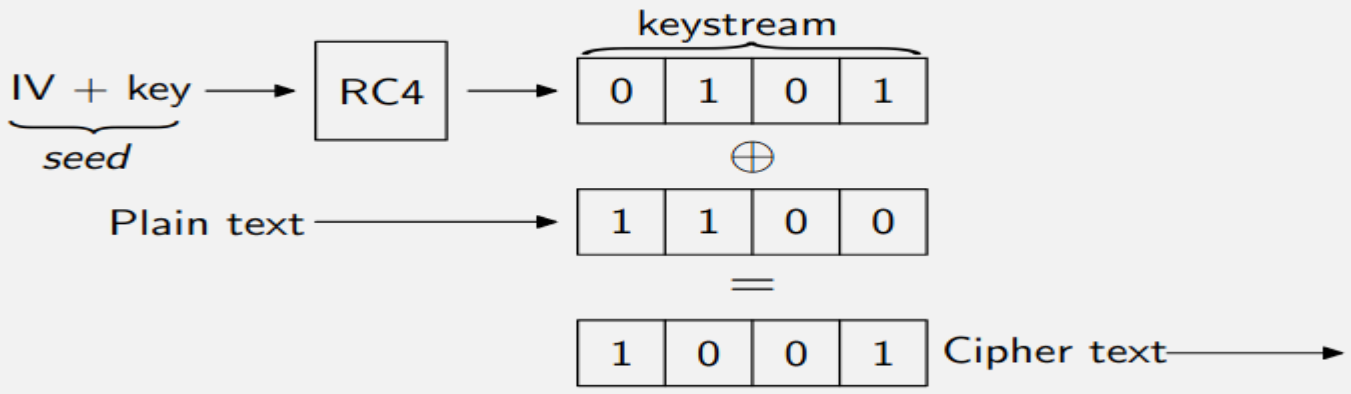
RC4, 40-bit key, 24-bit IV

Following are from:
https://jedcrandall.github.io/courses/cse468fall2022/wep/198fbe890b692e5296fcf7ad1b015e653ec9.pdf

Data frame format



Encryption



If cipher-text & plain-text pair is known, their XOR is a keystream.
Known plain-text (LLC/SNAP headers) in IP packets:

802.11 header	0xAA	0xAA	0x03	0x00	0x00	0x00	0x08	0x00

$\oplus$

802.11 header	Cipher-text

$=$

8 bytes of keystream

Can recover 8 bytes of keystream by eavesdropping a packet.

- Can encrypt (and transmit) 8 bytes of arbitrary data.

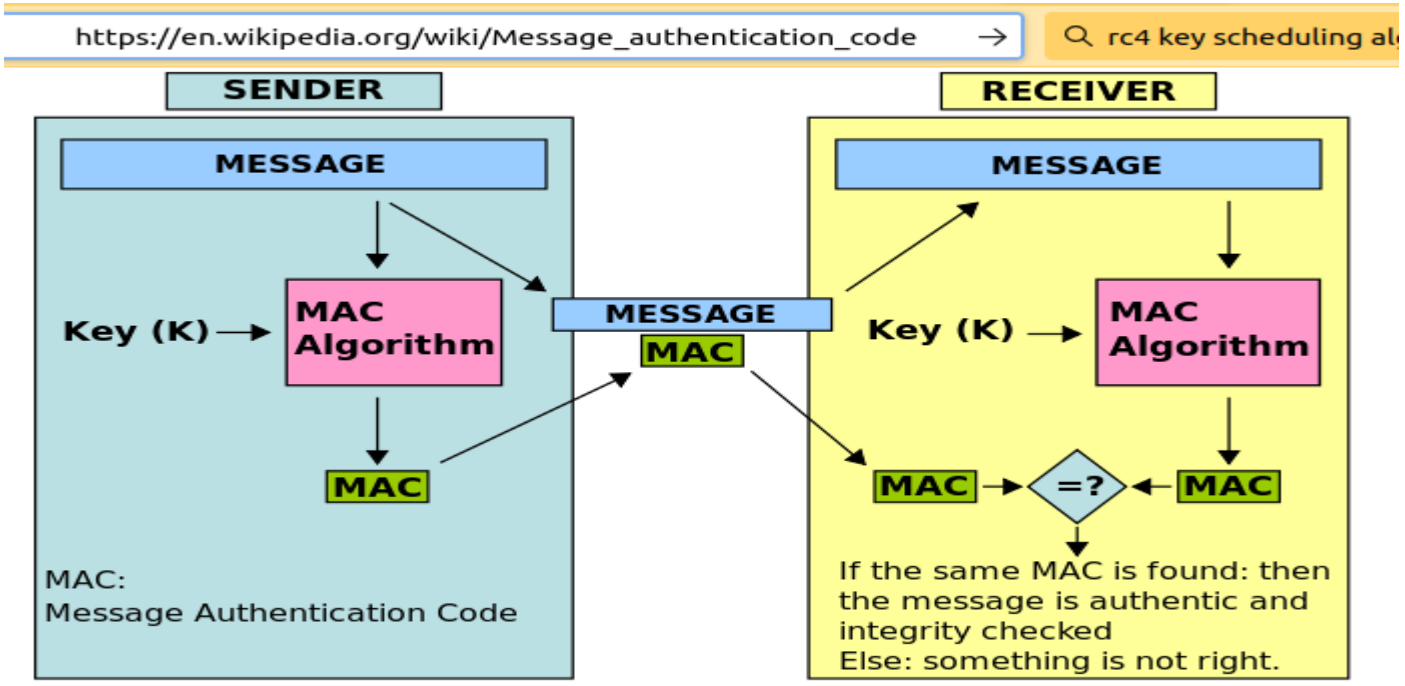
rc4-3.py

Possible to create statistical biases in the Key Scheduling Algorithm (KSA)

More info:

https://www.youtube.com/watch?v=2o3Hs-JDWLs

Message authentication codes...


https://en.wikipedia.org/wiki/Message_authentication_code
rc4 key scheduling al
SENDER
RECEIVER
MESSAGE
MESSAGE
Key (K)
MAC Algorithm
MESSAGE
MAC
Key (K)
MAC Algorithm
MAC
MAC
=?
MAC
MAC:
Message Authentication Code
If the same MAC is found: then the message is authentic and integrity checked
Else: something is not right.

Example: HMAC

- Can make a MAC based on a key, K, and a secure hash function, H
- Why not just use MAC = H(key || message)?

Example: HMAC

- Can make a MAC based on a key, K, and a secure hash function, H
- Why not just use MAC = H(key || message)?
  - Length extension attacks
- People tried other things like MAC = H(key || message || key)
  - Also has problems

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

This definition is taken from RFC 2104:

$$\mathrm{HMAC}(K, m) = \mathrm{H}\left((K' \oplus opad) \parallel \mathrm{H}\left((K' \oplus ipad) \parallel m \right) \right)$$

$$K' = \begin{cases} \mathrm{H}(K) & \text{if } K \text{ is larger than block size} \\ K & \text{otherwise} \end{cases}$$

H is a cryptographic hash function.

m is the message to be authenticated.

K is the secret key.

K' is a block-sized key derived from the secret key, *K*; either by padding to the right with 0s up to the block size, or by hashing down to less than or equal to the block size first and then padding to the right with zeros.

‖ denotes concatenation.

⊕ denotes bitwise exclusive or (XOR).

opad is the block-sized outer padding, consisting of repeated bytes valued 0x5c.

ipad is the block-sized inner padding, consisting of repeated bytes valued 0x36.[3]

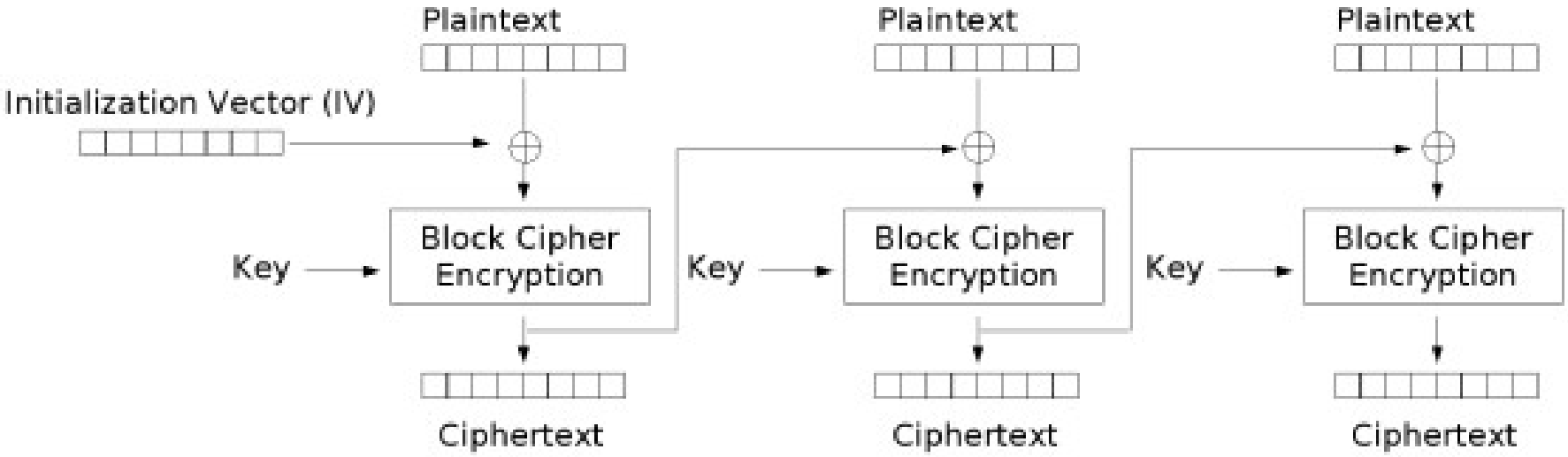
What to authenticate?

- Plaintext?
- Ciphertext?
- Metadata?
  - E.g., message number, sender, receiver, *etc.*

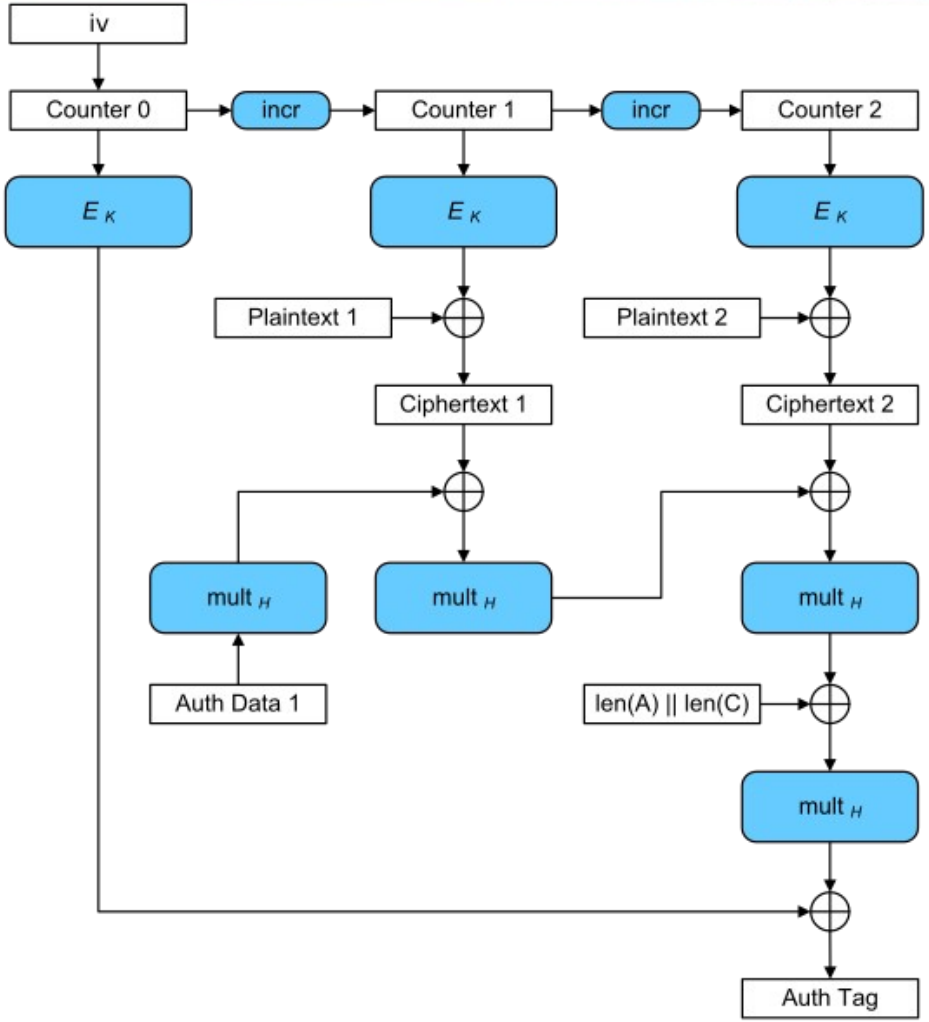
Gallois counter mode...

Why GCM?

- Both confidentiality and authenticity (integrity)
- Super fast
  - CPU pipeline reasons
  - Parallelism
    - Compare to CBC
- IVs of arbitrary length
- Stream cipher
  - Based on Counter Mode



Cipher Block Chaining (CBC) mode encryption



https://commons.wikimedia.org/wiki/File:GCM-Galois_Counter_Mode_with_IV.svg