Even fresh message can be replayed

Repurpose the AES block cipher to secure a stream.

Lab3

Encrypting Most Everything, but using Stream Cipher

Remember

There is no 100% security

Security, like all engineering, involves tradeoffs

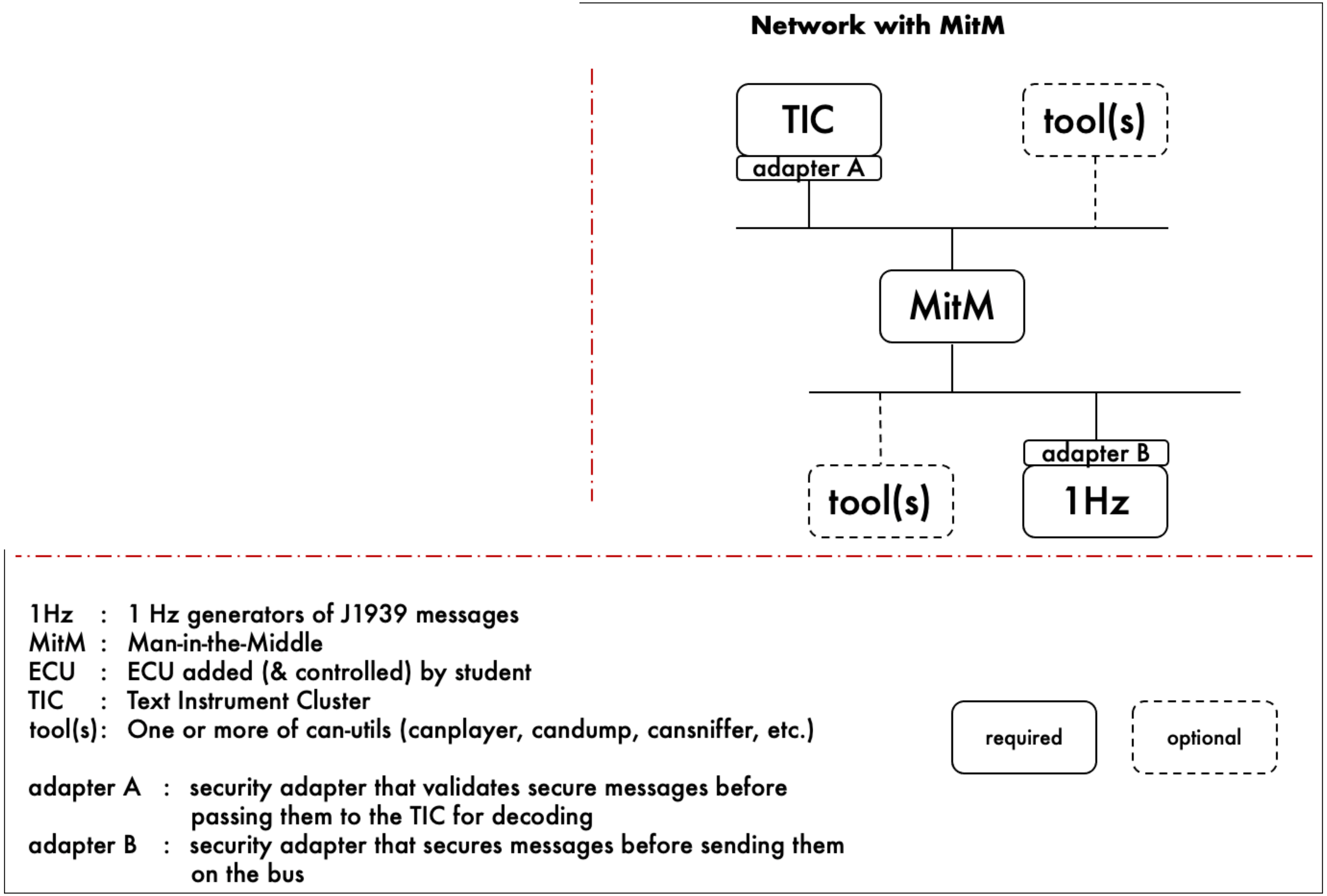
Know what you are trying to secure

The adversary…

State Sponsored

Network Configuration

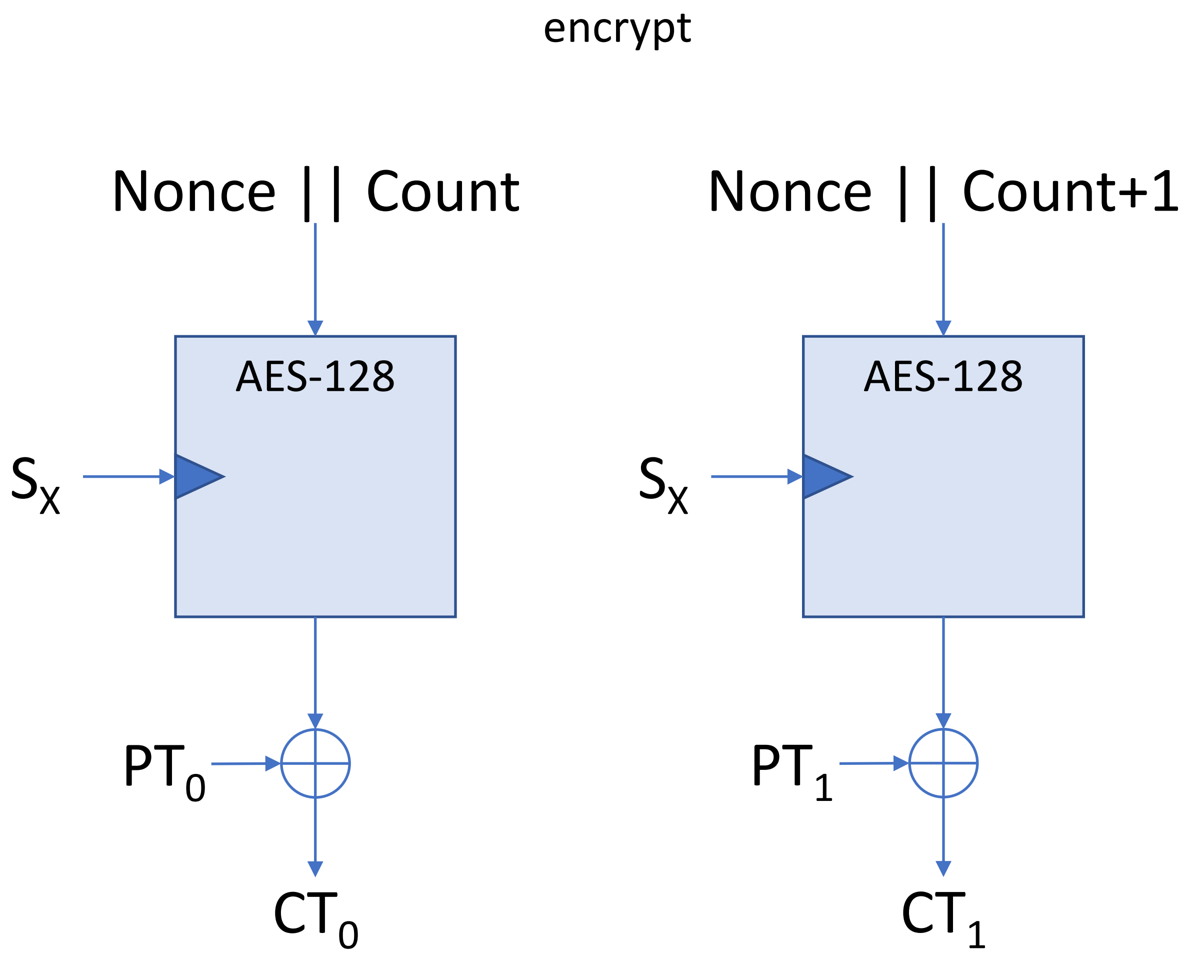
Use just MitM configuration in this lab



Historical Reference

- Personal experience – brainstorming sessions early in the development of product security
  - Make it “easy”, “just encrypt everything”
  - “128-bit encryption is unbreakable”
  - It must fit in a standard CAN frame

AES-128 CTR

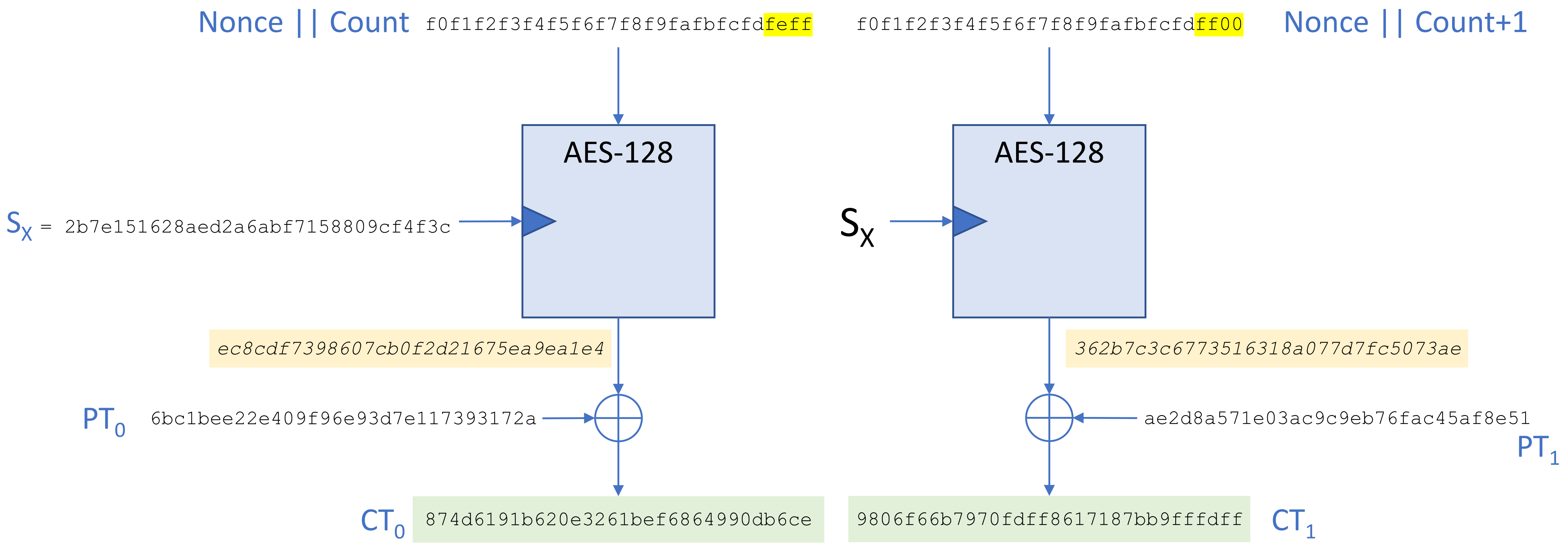


S_X: symmetric key for entity "x"
PT: plaintext
CT: ciphertext

CTR: counter mode

AES-128 CTR **NIST 800-38A**

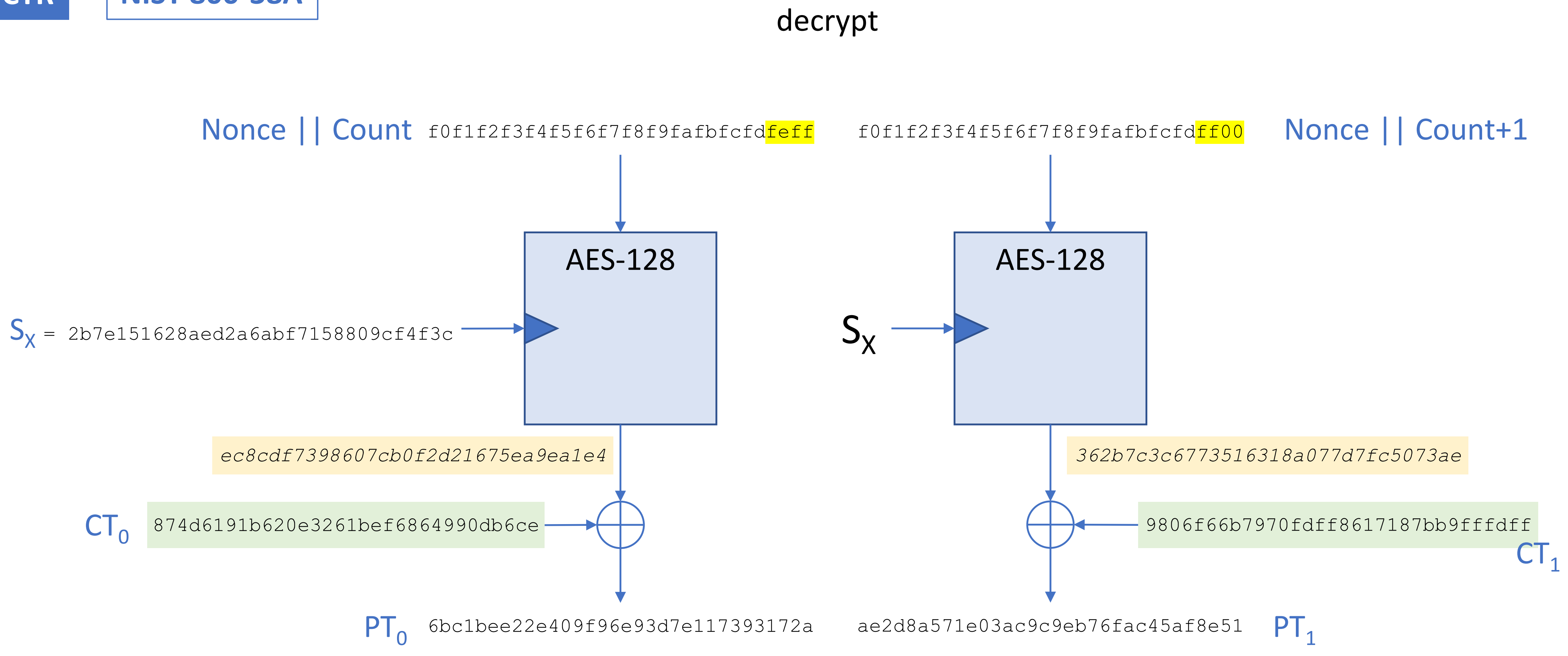
encrypt



https://nvlpubs.nist.gov/nistpubs/Legacy/SP/nistspecialpublication800-38a.pdf

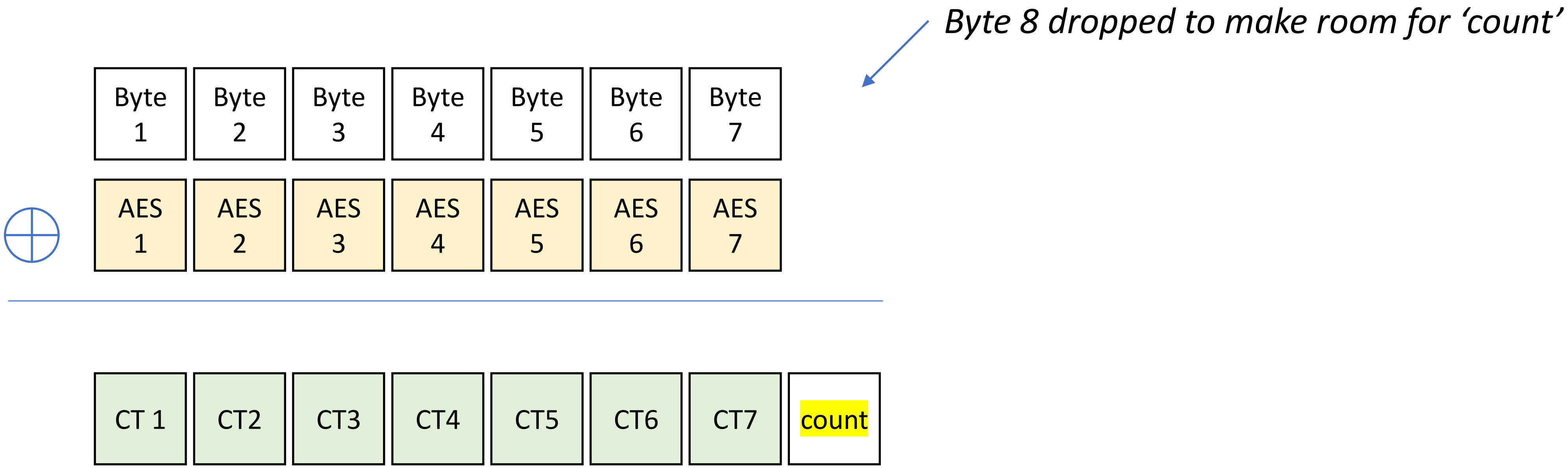
AES-128 CTR

NIST 800-38A

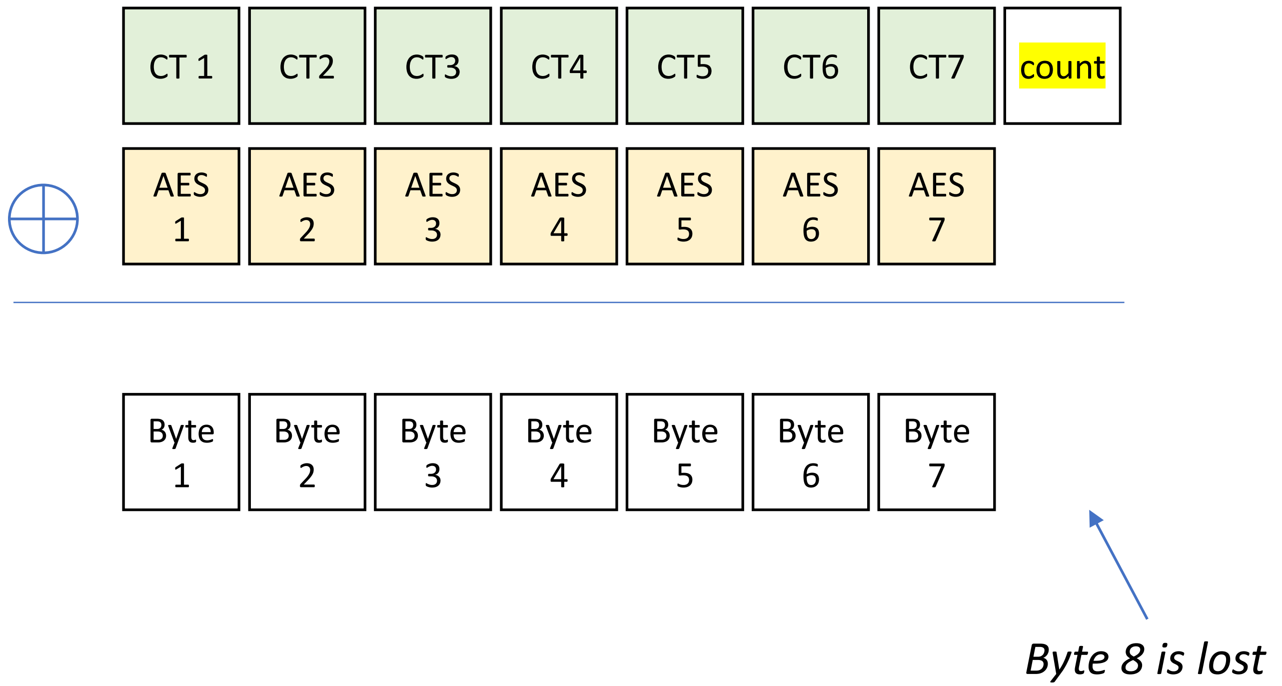


https://nvlpubs.nist.gov/nistpubs/Legacy/SP/nistspecialpublication800-38a.pdf

CTR Example, Encrypt



CTR Example, Decrypt



CTR Example

```
> cat -n ctr.py
#
# reference:
# https://cryptography.io/en/latest/hazmat/primitives/symmetric-encryption/

from cryptography.hazmat.primitives.ciphers import Cipher, algorithms, modes

count = 0  # will range thru 0..255

Sx =    bytes.fromhex("00000000 11111111 22222222 33333333")
nonce = bytes.fromhex("00000000 00000000 00000000 000000")
count = count + 1
iv = bytes(nonce) + count.to_bytes(1, "big")
print("iv")
print(iv.hex(" ", 4))

cipher = Cipher(algorithms.AES(Sx), modes.ECB())
encryptor = cipher.encryptor()
ctiv = encryptor.update(iv) + encryptor.finalize()
print("ctiv")
print(ctiv.hex(" ", 4))

data = bytes.fromhex("dead cafe beef 0102")
ct = bytes([a ^ b for a, b in zip(ctiv, data)])
print("ct")
print(ct.hex(" ", 4))

print("now, to decrypt...")
pt = bytes([a ^ b for a, b in zip(ctiv, ct)])
print("pt")
print(pt.hex(" ", 4))
```

```
> python3 ctr.py | cat -n
iv
00000000 00000000 00000000 00000001
ctiv
b01731f7 55efc4f5 ed9a1e20 a9963816
ct
6ebafb09 eb00c5f7
now, to decrypt...
pt
deadcafe beef0102
```

Lab

- Use Su = 00000000 11111111 22222222 33333333

- What is a valid message?
  - Established by Policy and the security needs
  - How do we detect invalid messages?
  - When we detect an issue, what do we do?
  - Without policies:
    - Unwanted data makes it through.
    - Sometimes it is malformed bytes that may exceed bounds during conversion to engineering units

Data Plane / Control Plane

- Taken from enterprise networking

- For <u>our</u> purposes:
    - Data Plane – the messages moving content (e.g., ground speed)
    - Control Plane – the message moving configuration (e.g., NAME claim)

- With this lab we bring in control plane for last exercise