Pretty Good Security

Stretching the value of a single byte.

Lab4

Pretty Good Security: Freshness and Integrity

Remember

There is no 100% security

Security, like all engineering, involves tradeoffs

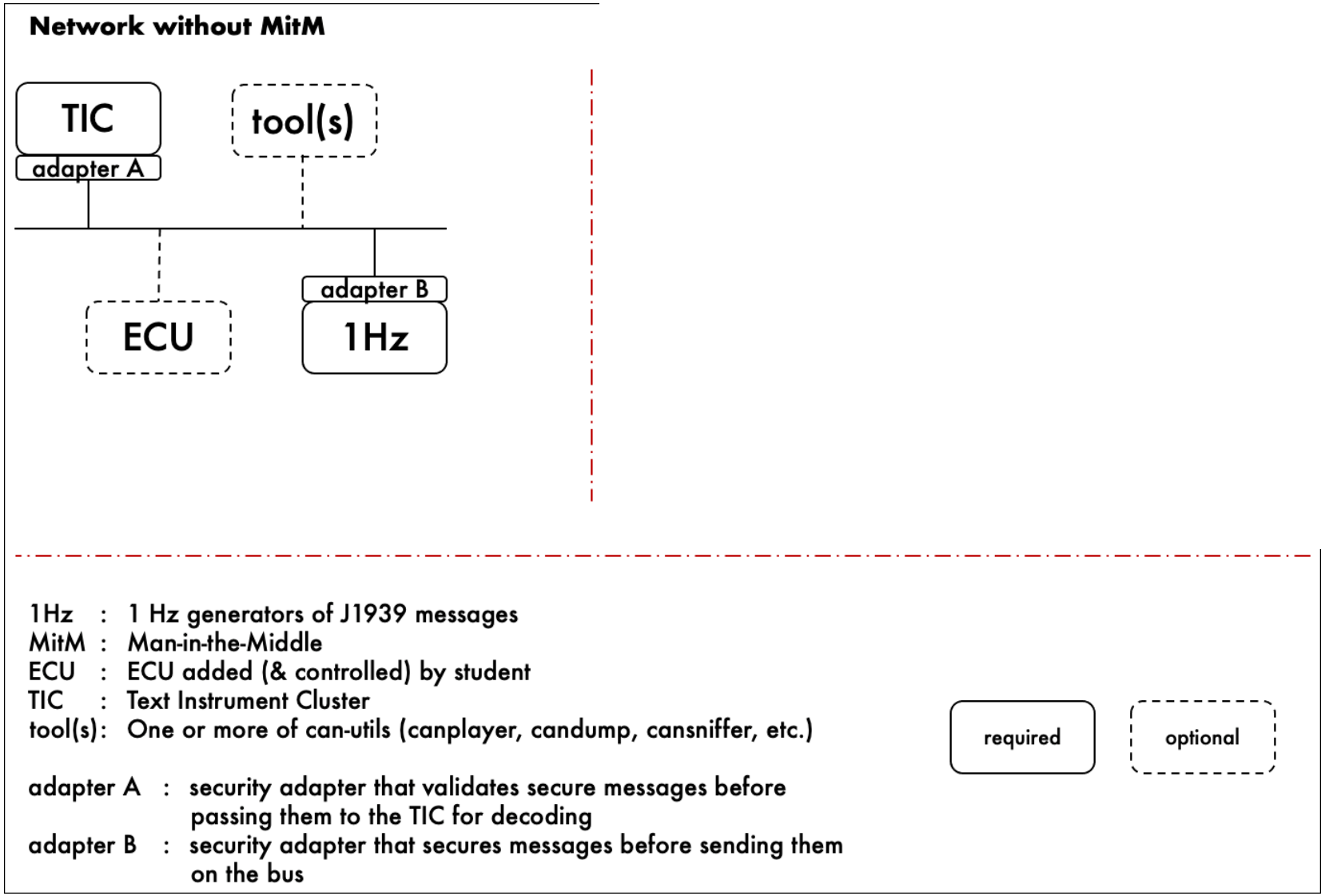
Know what you are trying to secure

The adversary…

State Sponsored

Network Configuration

Simple Network for this Lab

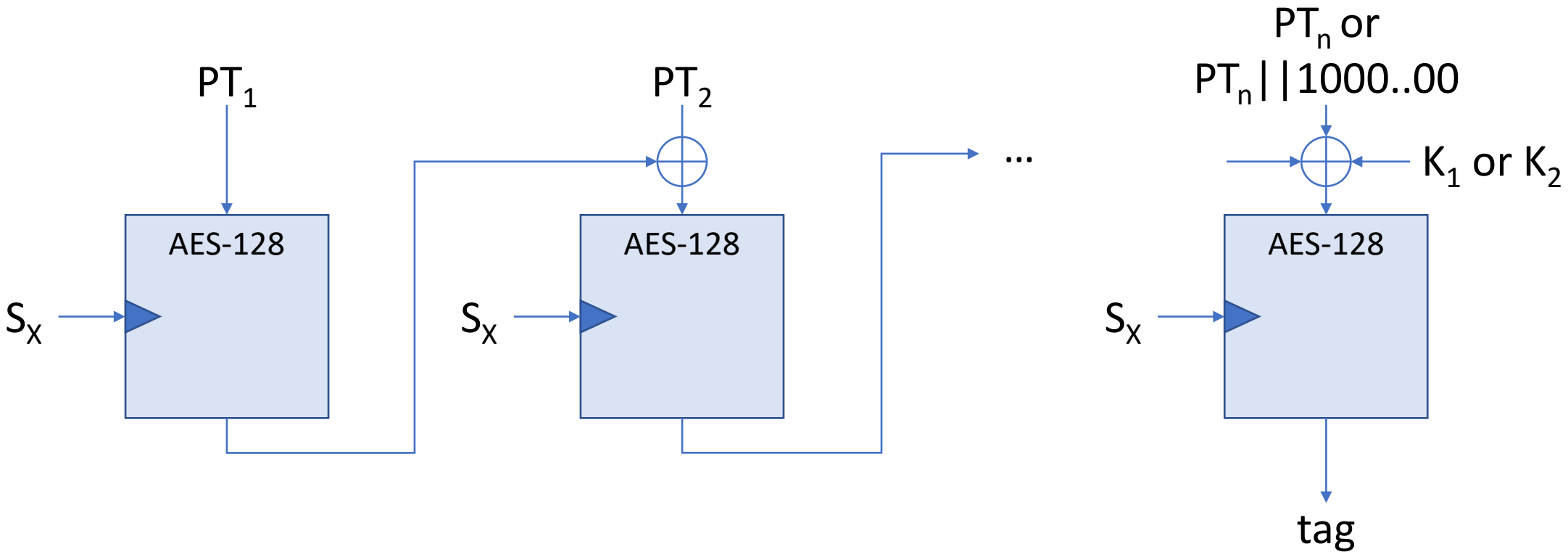


Historical Reference

- Personal experience – good enough security
  - Small footprint for overhead – only uses 1 byte
  - Use a 128-bit MAC, but truncated
  - Use a “big enough” freshness value, but only a portion is explicit

AES-128 CMAC **RFC 4493**

Generate keyed tag

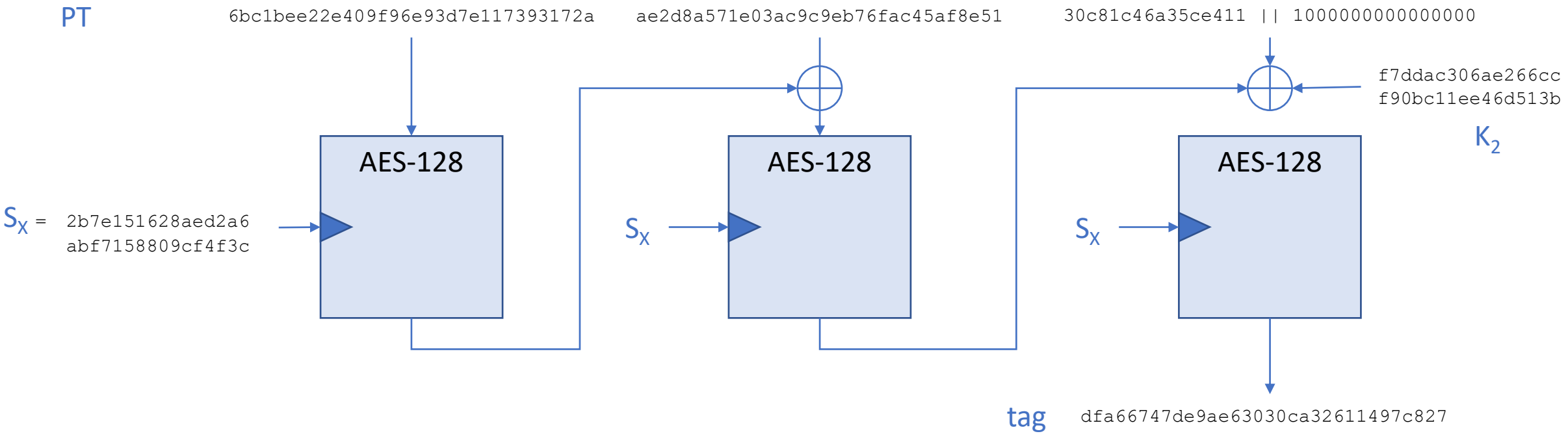


S_X: symmetric key for entity "x"
PT: plaintext
CT: ciphertext

CMAC: cipher message authentication code
Tag: fixed-size, keyed, cryptographic hash of plaintext (128 bits for AES-128 CMAC)

AES-128 CMAC | **NIST 800-38B** | **RFC 4493**

Generate keyed tag
Example 3 from RFC 4493



https://nvlpubs.nist.gov/nistpubs/specialpublications/nist.sp.800-38b.pdf
https://www.rfc-editor.org/rfc/rfc4493.html

CMAC example

```
> cat -n cmac.py
     1  #
     2  # reference:
     3  # https://cryptography.io/en/latest/hazmat/primitives/mac/cmac/
     4
     5  from cryptography.hazmat.primitives import cmac
     6  from cryptography.hazmat.primitives.ciphers import algorithms
     7
     8  Sx = bytes.fromhex("00000000 11111111 22222222 33333333")
     9  c  = cmac.CMAC(algorithms.AES(Sx))
    10
    11  data = bytes.fromhex("00 11 22 33 44 55 66")
    12  c.update(data)
    13  tag = c.finalize()
    14  print("tag - has length %d" % (len(tag)))
    15  print(tag.hex(" ", 4))
    16  tagprime = tag[0]
    17  print("tagprime")
    18  print("%02x" % (tagprime))
    19
    20
    21  data2 = bytes.fromhex("01 11 22 33 44 55 66")
    22  c2 = cmac.CMAC(algorithms.AES(Sx))
    23  c2.update(data2)
    24  tag2 = c2.finalize()
    25  print("\n\ntag2")
    26  print(tag2.hex(" ", 4))
    27
```

```
> python3 cmac.py
tag - has length 16
70122c50 987d75ad f9be6249 3fd8ef04
tagprime
70


only one bit difference in data...
see the new CMAC value, tag2:
9a93ee34 d6ee5e86 6e37ac06 50fad4ad
```

Freshness

- Explicit
  - All freshness values appear in the message
- Implicit
  - None of the freshness values appear in the message
  - (nodes keep track by counting messages or use some other value on the bus)
- Hybrid
  - Say freshness is 32-bit value, but only 4 bits appear in the message
  - Make the least-significant bits the explicit portion

Lab

- Use Su = 00000000 11111111 22222222 33333333