MAC-then-Encrypt or Encrypt-then-MAC?

- Method 1: Encrypt-then-MAC
  - First compute Enc(K_1, M)
  - Then MAC the ciphertext: MAC(K_2, $\underline{\text{Enc}(K_1, M)}$)
- Method 2: MAC-then-encrypt
  - First compute MAC(K_2, M)
  - Then encrypt the message and the MAC together: $\underbrace{\text{Enc}(k_1, \overset{M'}{M} \,||\, \text{MAC}(K_2, M))}_{\text{decrypt}} \rightarrow \underline{M} \,||\, \text{MAC}$
- Which is better?
  - In theory, both are secure if applied properly
  - MAC-then-encrypt has a flaw: You don't know if tampering has occurred until after decrypting
    - Attacker can supply arbitrary tampered input, and you always have to decrypt it RSA timing attack
    - Passing attacker-chosen input through the decryption function can cause side-channel leaks
- **Always use encrypt-then-MAC** because it's more robust to mistakes

TLS 1.0 “Lucky 13” Attack

- TLS: A protocol for sending encrypted and authenticated messages over the Internet

2nd, M'

- TLS 1.0 uses MAC-then-encrypt: $\text{Enc}(k_1, M \,||\, \text{MAC}(k_2, M))$
  - The encryption algorithm is AES-CBC
- The Lucky 13 attack abuses MAC-then-encrypt to read encrypted messages
  - Guess a byte of plaintext and change the ciphertext accordingly

receiver

  - The MAC will error, but the time it takes to error is different depending on if the guess is correct

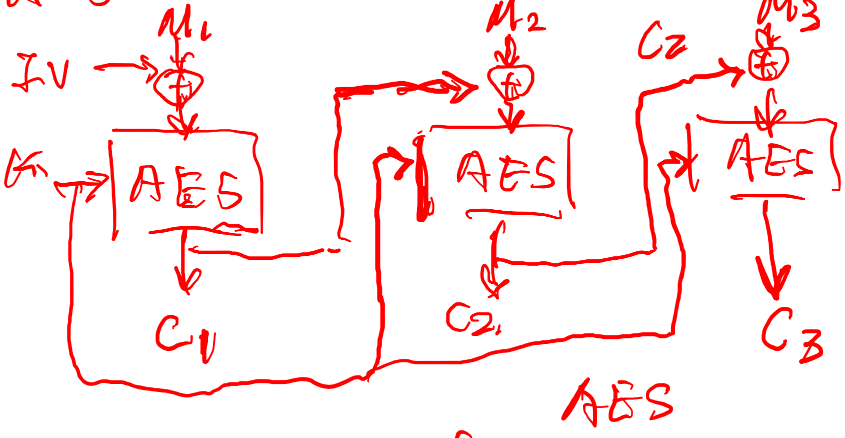
MAC + timing attack

  - Attacker measures how long it takes to error in order to learn information about plaintext
  - TLS will send the message again if the MAC errors, so the attacker can guess repeatedly

1 8t T1
2 8t T2

- Takeaways
  - Side channel attack: The algorithm is proved secure, but poor implementation made it vulnerable
  - Always encrypt-then-MAC

AES - CBC.

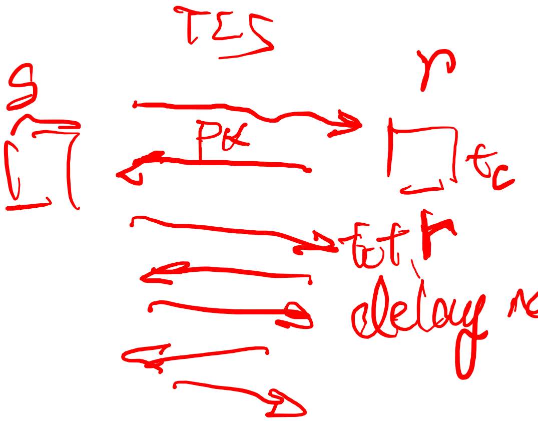


No timing attack

Defense:

① fixed comparison time

② random comparison time,



delay response time,

Big Data.

Authenticated Encryption: Summary

= no group key.

- Authenticated encryption: A scheme that simultaneously guarantees confidentiality and integrity (and authenticity) on a message
- Approach: Combine schemes that provide confidentiality with schemes that provide integrity and authenticity
  - MAC-then-encrypt: $\mathrm{Enc}(K_1, M \,||\, \mathrm{MAC}(K_2, M))$
  - Encrypt-then-MAC: $\mathrm{MAC}(K_2, \mathrm{Enc}(K_1, M))$
  - Always use Encrypt-then-MAC because it's more robust to mistakes

Computational time

MAC ≤ Symmetric Encryption ≤ asymmetric Encryption

Signature

HC 1