Applied Cryptography
CPEG 472/672
Lecture 8A

Instructor: Nektarios Tsoutsos

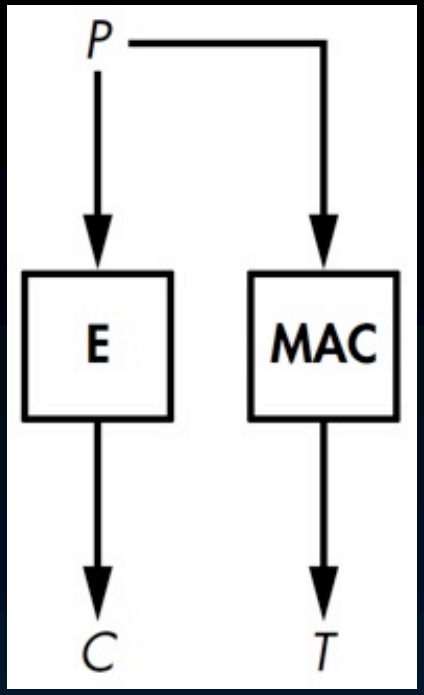
Authenticated Encryption (AE)

- We want confidentiality + integrity
  - Combine a cipher with a MAC
- Different ways to achieve this
  - Using MACs
  - Using Authenticated ciphers
- Real-life examples using AES (Thursday)
  - GCM (Galois counter mode)
  - OCB (Offset codebook)
  - SIV (Synthetic IV)

AE using MACs

- Three different constructions
  - Encrypt and MAC (EaM)
  - MAC then Encrypt (MtE)
  - Encrypt then MAC (EtM)
- What is the difference?
  - The cipher and the MAC are combined in different order
  - Different compositions have different properties
- Different keys for cipher and MAC

Encrypt and MAC (EaM)

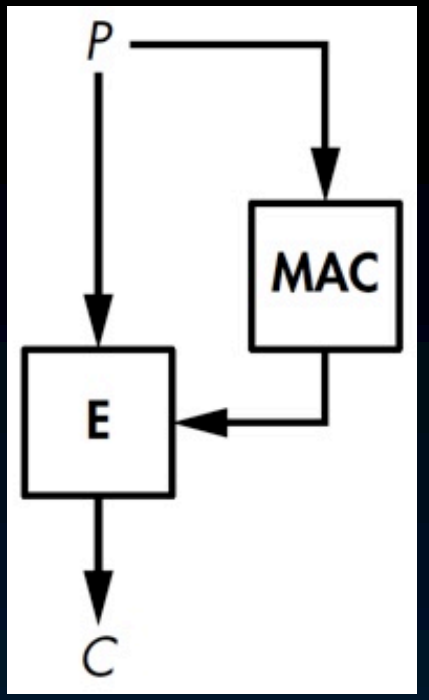


- Ctxt and MAC computed separately
  - Parallelizable
- Sender computes C=E(K1,P) and T=MAC(K2,P)
  - C || T is sent to the recipient
- Recipient computes P=D(K1,C) and T'=MAC(K2,P)
  - Check if T'==T
  - Decryption of C happens before checking T

Is EaM secure?

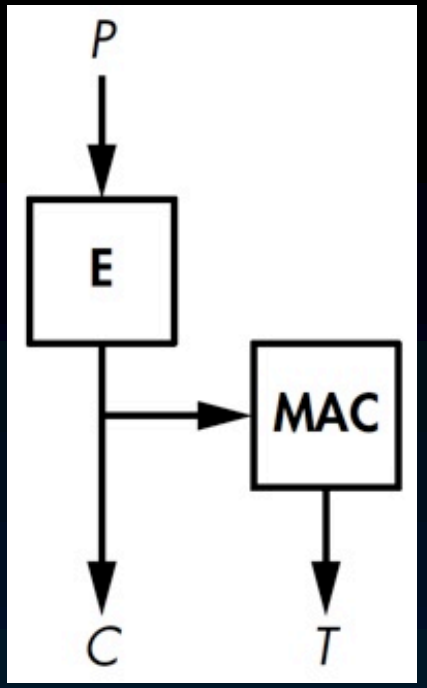
- In theory, EaM is the least secure variant
- Recall, the goal of MACs is unforgeability
  - This means a MAC could leak info about the input message
    - Confidentiality is not a goal of MACs
  - If the MAC is PRF the tag won't have leaks
- A secure MAC in EaM may leak ptxt bits
  - Makes recovering P from C easier
- EaM used in SSH: T=MAC(K,N||P)
  - MAC is HMAC-SHA-256 (no leaks)

MAC then Encrypt (MtE)



- First compute T=MAC(K2, P)
- Then encrypt C=E(K1, P || T)
- Recipient decrypts C
  - P || T=D(K1,C)
  - Decryption of C happens before checking T
- Recipient computes T'=MAC(K2, P) and compares T' with T
  - More secure than EaM
- Used in TLS before v1.3

Encrypt then MAC (EtM)



- Sender encrypts ptxt
  - C=E(K1,P)
- Sender generates a tag for C
  - T=MAC(K2,C)
  - C || T is sent to the recipient
- Recipient computes T'=MAC(K2,C) first
  - Check if T==T' and then decrypt P=D(K1,C)
  - No decryption if tag verification fails
    - Recipient cannot be decryption oracle for attacks
    - Stronger than EaM, MtE; used on IPSec

Authenticated ciphers

- Alternative to cipher + MAC construction
- The cipher returns ctxt + tag
- Notation
  - Authenticated Encryption: (C,T) = AE(K,P)
  - Authenticated Decryption: (P,err)=AD(K,C,T)
    - No plaintext if there is error in tag verification
    - Prevents chosen-ctxt queries
- If ptxt is returned, it must be encrypted by someone who knows the AE key K

AEAD

Authenticated Encryption with Associated Data

- What is associated data?
  - Data we want to authenticate by not encrypt
  - E.g., packet header must be unencrypted
- Notation
  - Encryption: (C,A,T) = AEAD(K,P,A)
  - Associated data A is part of the output
  - Tag T includes A, ctxt C does not include A
  - Decryption: (P,A,err) = ADAD(K,C,A,T)
    - Error if C or A is corrupted
  - You may leave P or A empty in AEAD

Nonces in AE

- Nonces: prevent attackers from detecting if the same ptxt is encrypted twice
  - We have seen nonces and IVs before
  - Same approach in authenticated ciphers
- Nonce must be unique for the same key
- Notation
  - (C,A,T) = AEAD(K,P,A,N)
  - (P,A,err) = ADAD(K,C,A,T,N)

Evaluation criteria for AE

- Security
  - Protect confidentiality, authenticity, integrity
  - AEAD as secure as a secure cipher and as strong as a secure MAC
  - Misuse resistance: impact of nonce reuse
- Performance
  - Number of operations, parallelization
  - Single-layer, double-layer structure
  - Streamability: can we discard already processed blocks?
    - Less memory (e.g., useful for routers)

Hands-on exercises

- Encrypt and MAC
- MAC then Encrypt
- Encrypt then MAC
- AEAD example

Reading for next lecture

- Aumasson: Chapter 8 until end of chapter
  - We will have a short quiz on the material