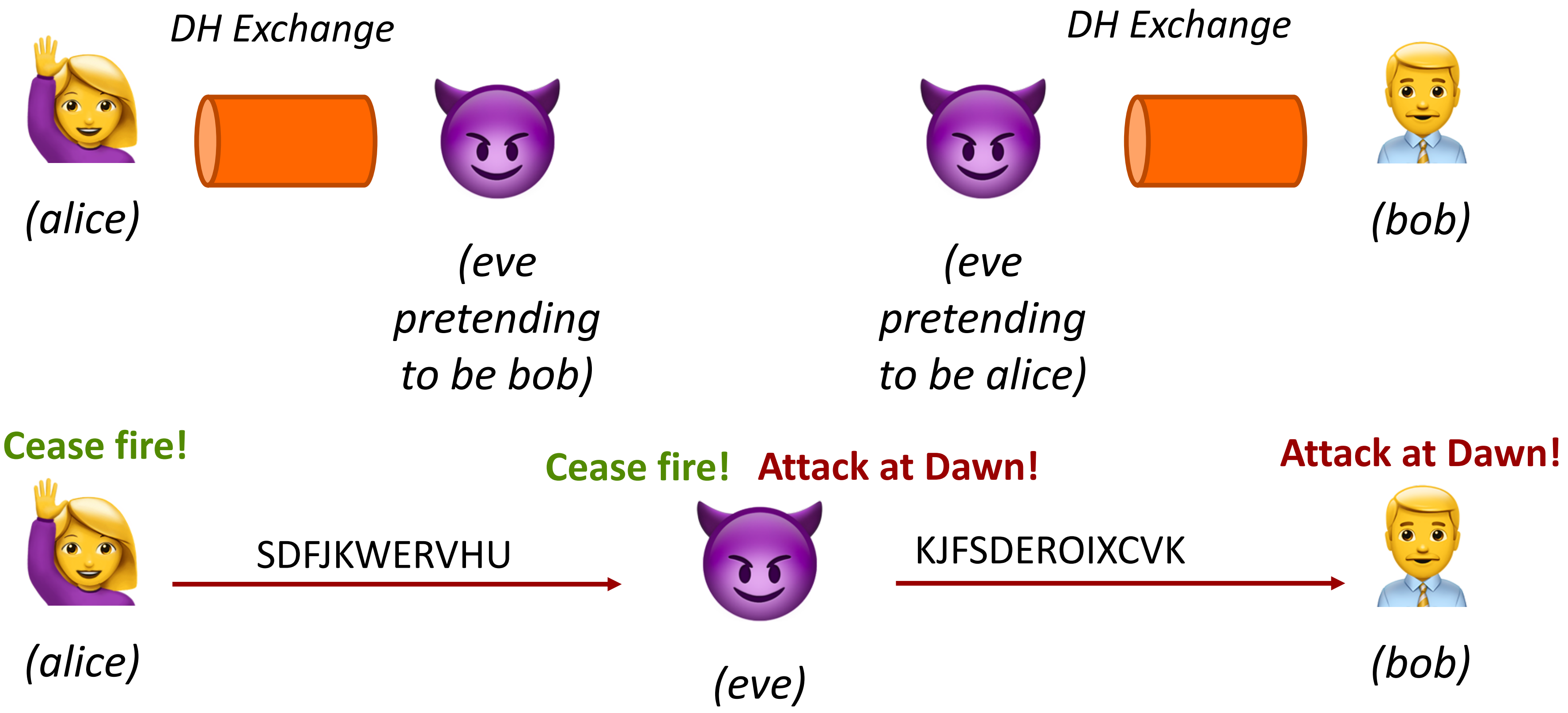
Authenticated Encryption

Motivation

What if the attacker *actively manipulates* data instead of passively observing it?

Motivation

How do we protect against this scenario?



DH Exchange

(alice)

(eve pretending to be bob)

DH Exchange

(eve pretending to be alice)

(bob)

Cease fire!

(alice)

SDFJKWERVHU

Cease fire! **Attack at Dawn!**

(eve)

KJFSDEROIXCVK

Attack at Dawn!

(bob)

Warning!

Encryption without *authentication* is almost certainly **wrong**…

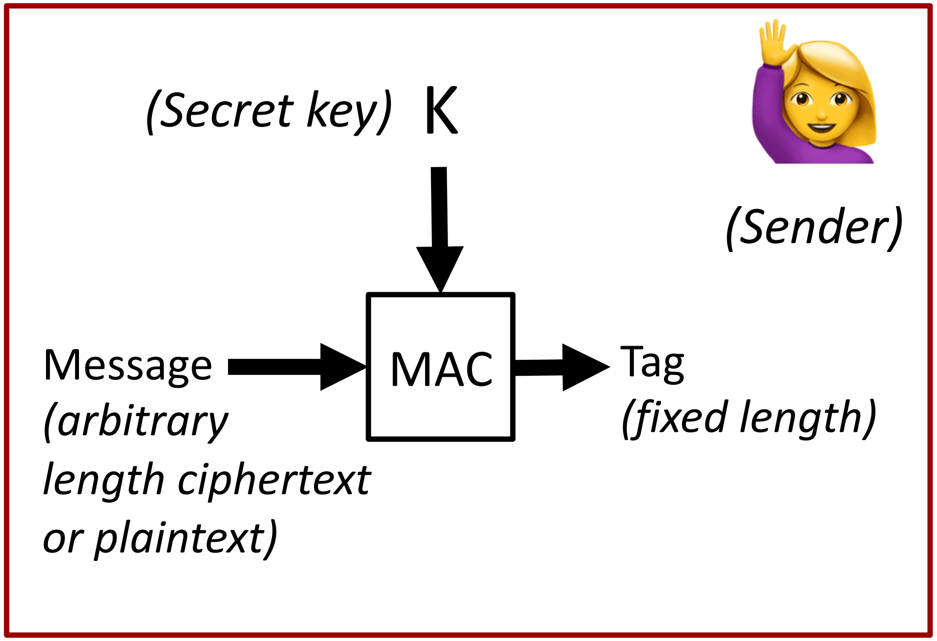
Attackers don’t need to *decrypt* to *modify* ciphertext

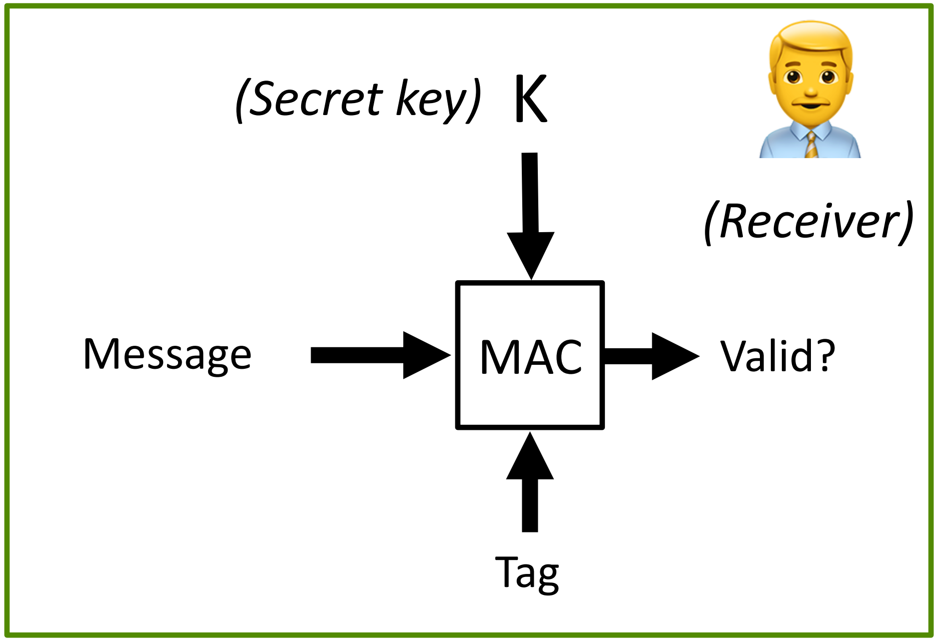
Authentication

- Goal: Add information to message that only the real sender (not Eve!) could have computed
- Authentication for symmetric-key encryption
  - **"Message Authentication Codes"**
  - MACs are generated and verified with the *same* key
- Authentication for public-key encryption
  - **"Signatures"**
  - Signatures are generated with *private* key and verified with *public* key

Message Authentication Code (MAC)

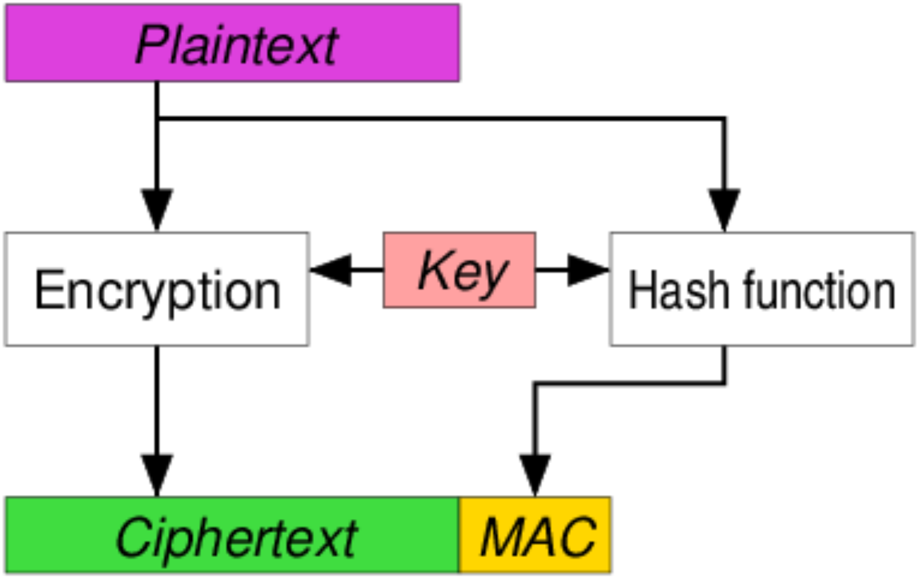
- Small piece of information used to verify message integrity / authenticity ("Tag")
- Key is *shared secret* between Alice and Bob





Message Authentication Code (MAC)

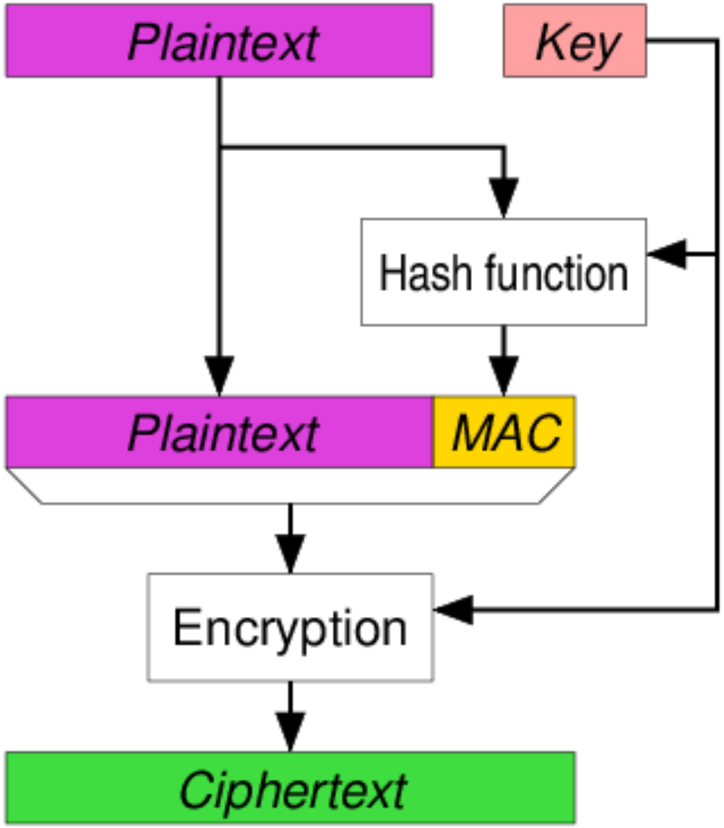
How to combine ciphertext with a MAC?



- **Authenticate and Encrypt**
  - Used by SSH
  - Authenticate and encrypt plaintext *separately*
  - $C = E(K_C,P)$ and $t=MAC(K_M,P)$
  - Send C and t

Message Authentication Code (MAC)

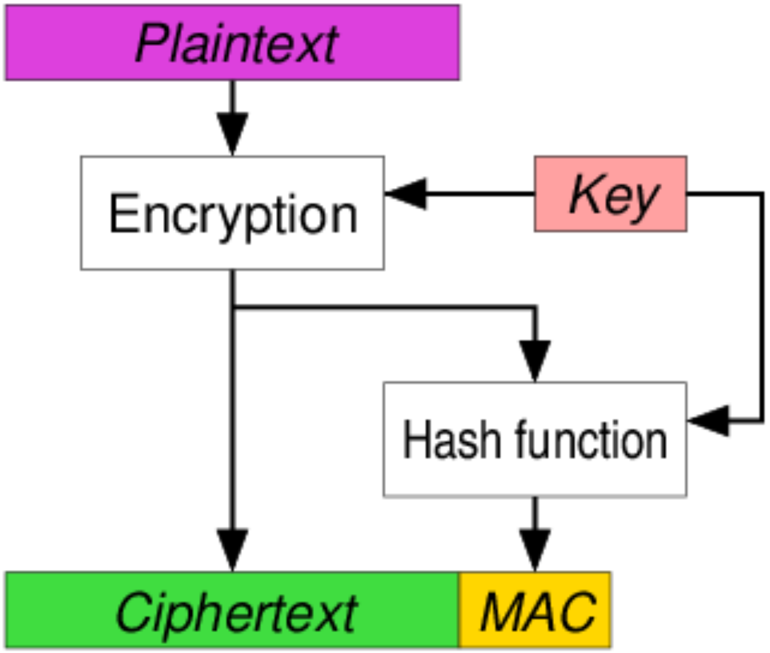
How to combine ciphertext with a MAC?



- **Authenticate, then Encrypt**
  - Used by TLS
  - Authenticate plaintext, then encrypt {plaintext, tag}
  - $t=MAC(K_M,P)$ then
  - $C = E(K_C, \{P|t\})$
  - Send C (t is part of C)

Message Authentication Code (MAC)

How to combine ciphertext with a MAC?



- **Encrypt, then Authenticate**
  - Used by IPSec
  - Standard ISO/IEC 19772:2009
  - Encrypt plaintext, then authenticate ciphertext
  - $C = E(K_C, P)$ then $t = MAC(K_M, C)$
  - Send C and t

Message Authentication Code (MAC)

How to combine ciphertext with a MAC?

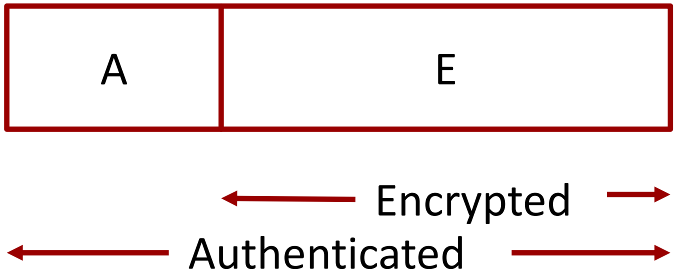
- **Which to choose?**
  - Authenticate and Encrypt
  - Authenticate, then Encrypt
  - Encrypt, then Authenticate – **Modern Best Practice**
- *Consider what the receiver does to reverse process*
- When you receive a message, the **very first thing** you do should be to authenticate it
  - Anything else risks **CERTAIN DOOM** (eventually)

Message Authentication Code (MAC)

- Position Statement: “Doom Principle”
  - https://moxie.org/blog/the-cryptographic-doom-principle/
- Example 1: Padding Oracle Attack (Vaudenay attack against CBC)
  - Trick receiver into revealing last byte of message by brute forcing padding byte, and then repeat for next to last byte, etc…
  - Successful on “Authenticate, then Encrypt” method because decryption happens first!
- Example 2: SSH Plaintext Recovery Attack
  - SSH has to decrypt first block to know message length
  - Attacker can substitute in arbitrary block and recipient will decrypt it and use attacker value as a message length
  - Successful on “Authenticate and Encrypt” because decryption happens first!

AEAD

- We can do better still! What if authentication was *part of* our encryption scheme, and not a separate step?
- **Authenticated Encryption with Associated Data (AEAD)**
  - Messages have two parts – example: emails
    - Content (encrypt!)
    - Metadata (authenticate, but plaintext)



AEAD Modes

- Galois Counter Mode (**GCM**) – **Good**!
  - Not patent encumbered
  - SSH, TLS 1.2, OpenVPN
  - Standardized in ISO/IEC 19772:2009
  - Can be used by itself (authentication-only): **GMAC**
- Many other AEAD modes
  - EAX, OCB 2.0, CCM, Key Wrap, …

Modes of Operation

Remember our Block Cipher *Modes of Operation?*

Encryption-Only

No Authentication

- Counter (**CTR**) – **Best!**
- Cipher Block Chaining (**CBC**) – **Good**
- Electronic Code Book (**ECB**) – **Don't use!**
- Also ran: CFB, OFB, XTS, …

MACs – Message Integrity Only, No Encryption

- **GMAC** - **Good**
- **HMAC** – **Good**
  - *But why are you just authenticating and not encrypting?*
- Also ran: ALG1-6, CMAC

Modes of Operation

Remember our Block Cipher *Modes of Operation?*

Authenticated Encryption (Encrypt + Auth)

- GCM – **Good!**
- CCM – **Good!**
- Also-ran: EAX, OCB 2.0, Key Wrap, …

Repeating the Warning…

Encryption without *authentication* is almost certainly **wrong**…

Attackers don’t need to *decrypt* to *modify* ciphertext

Meet a Cryptographer

- **Phillip Rogaway**
- Professor, Computer Science
  UC Davis
- Winner of Levchin prize for cryptography: http://levchinprize.com/

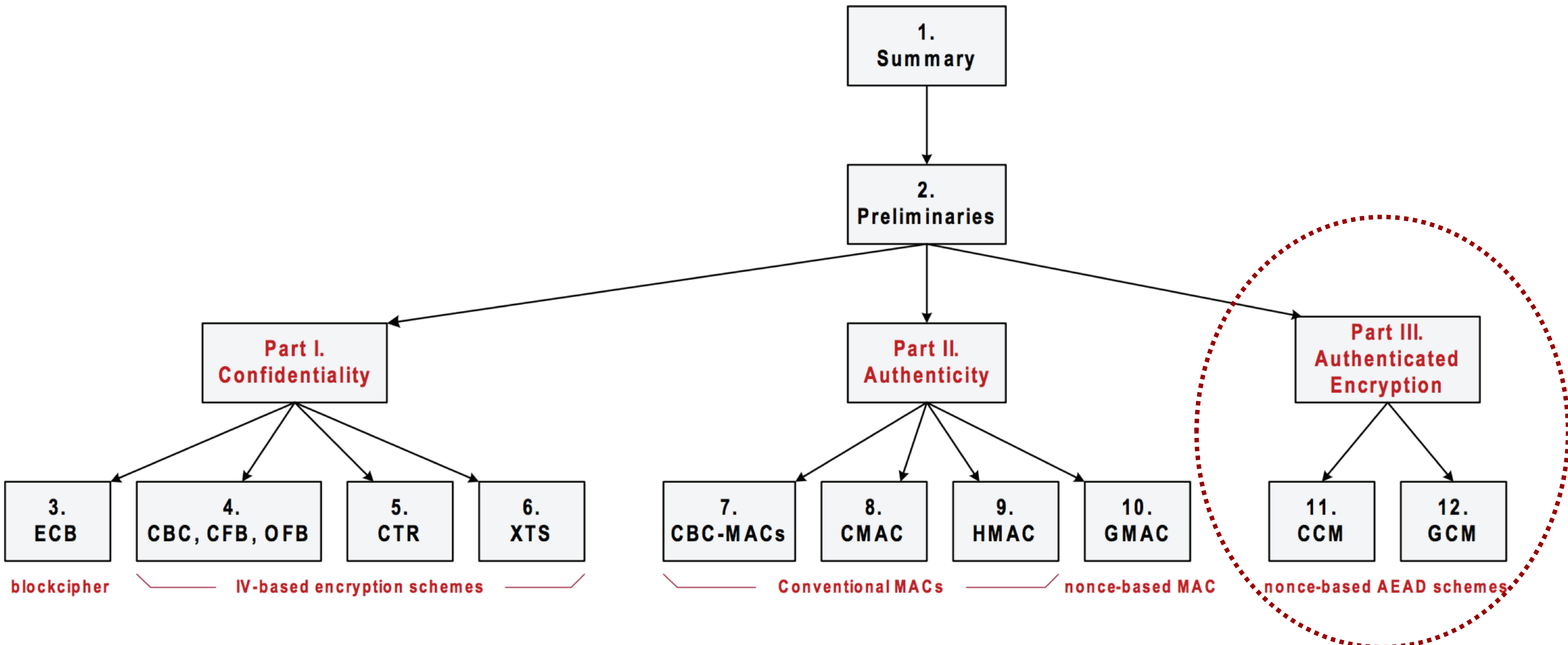


Figure 1.1: **Roadmap**. The chart shows organization and logical dependencies among the chapters and parts of this documents.

Rogaway, P. "Evaluation of Some Blockcipher Modes of Operation", February 2011
http://web.cs.ucdavis.edu/~rogaway/papers/modes.pdf
153 pages of details…

Authentication

- Goal: Add information to message that only the real sender (not Eve!) could have computed
- Authentication for symmetric-key encryption
  - **"Message Authentication Codes"**
  - MACs are generated and verified with the *same* key
- Authentication for public-key encryption
  - **"Signatures"**
  - Signatures are generated with *private* key and verified with *public* key

Signatures

- RSA-based signatures
- Digital Signal Algorithm (**DSA**)
- Elliptic Curve Digital Signature Algorithm (**ECSDA**)