Applied Cryptography

Lecture 2

Arnab Roy

13 March, 2025

University of Innsbruck

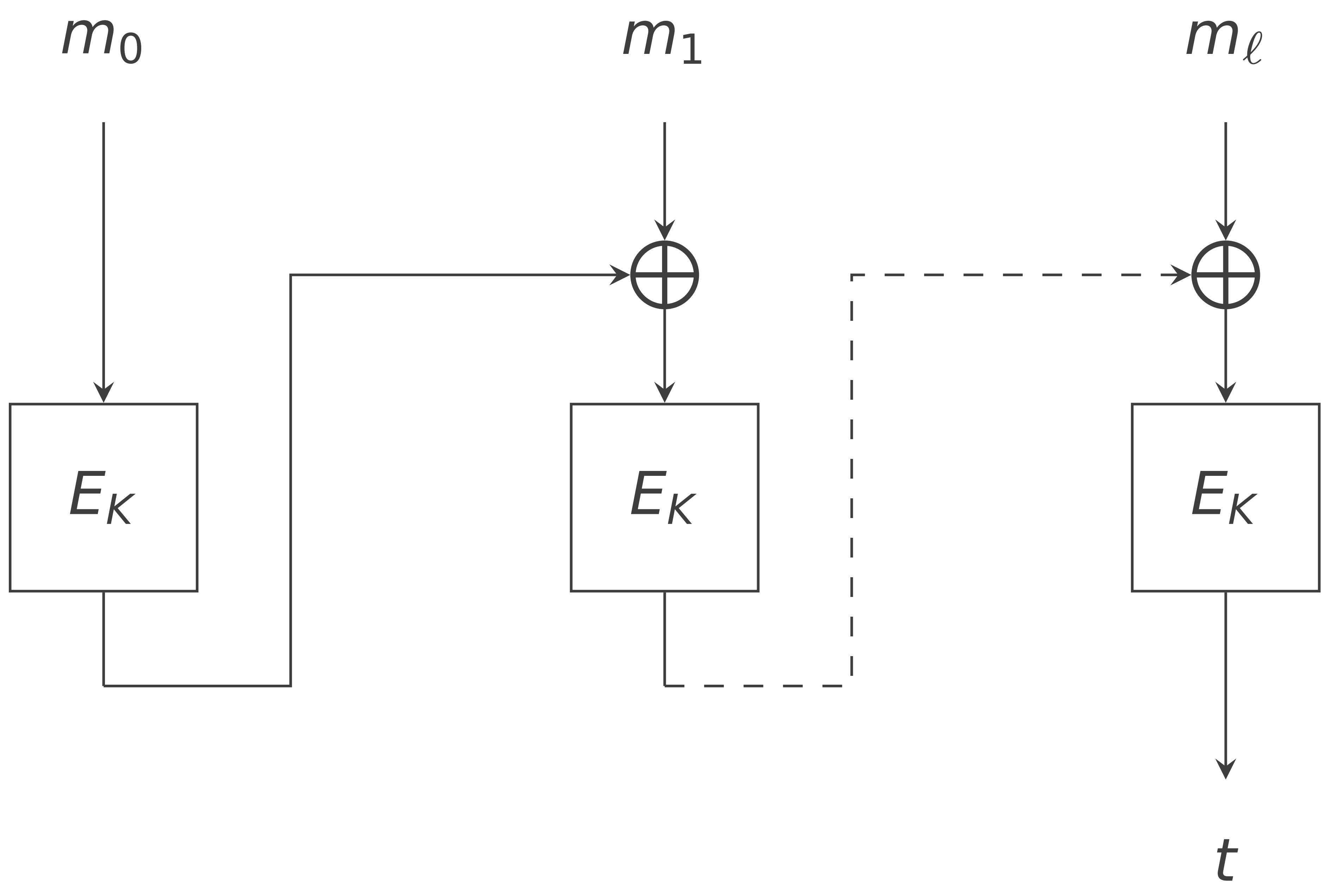
Message Authentication Code

- A MAC is a tuple of PPT (probabilistic polynomial time) algorithms (Gen, Tag, Vrfy)
  - Gen: takes an integer n as input and outputs a secret key k such that $|k| \geq n$
  - Tag: Takes k and message m as inputs and generates a tag t i.e. Tag(k, m) $\rightarrow t$
  - Vrfy: A deterministic algorithm that takes k, m, t as inputs and outputs $b =$ Verfy(k, m, t)
- $b = 0$ means the verification failed and the corresponding m, t pair is not valid; $b = 1$ means that t is a valid tag.
- Note: Tag can be a non-deterministic algorithm

Security of MAC

- Adversarial model
  - Adaptive chosen message attack
  - Adversary can obtain tags corresponding to the messages of here choice e.g. for each chosen m_i she obtains t_i
- Security Goal: existential unforgeability
  - It is computationally difficult for an adversary to forge a tag for a new message that she has not queried before. Suppose the forgery is m, t i.e. t is the tag for the message m
  - Note that $m \neq m_i$

CBC MAC



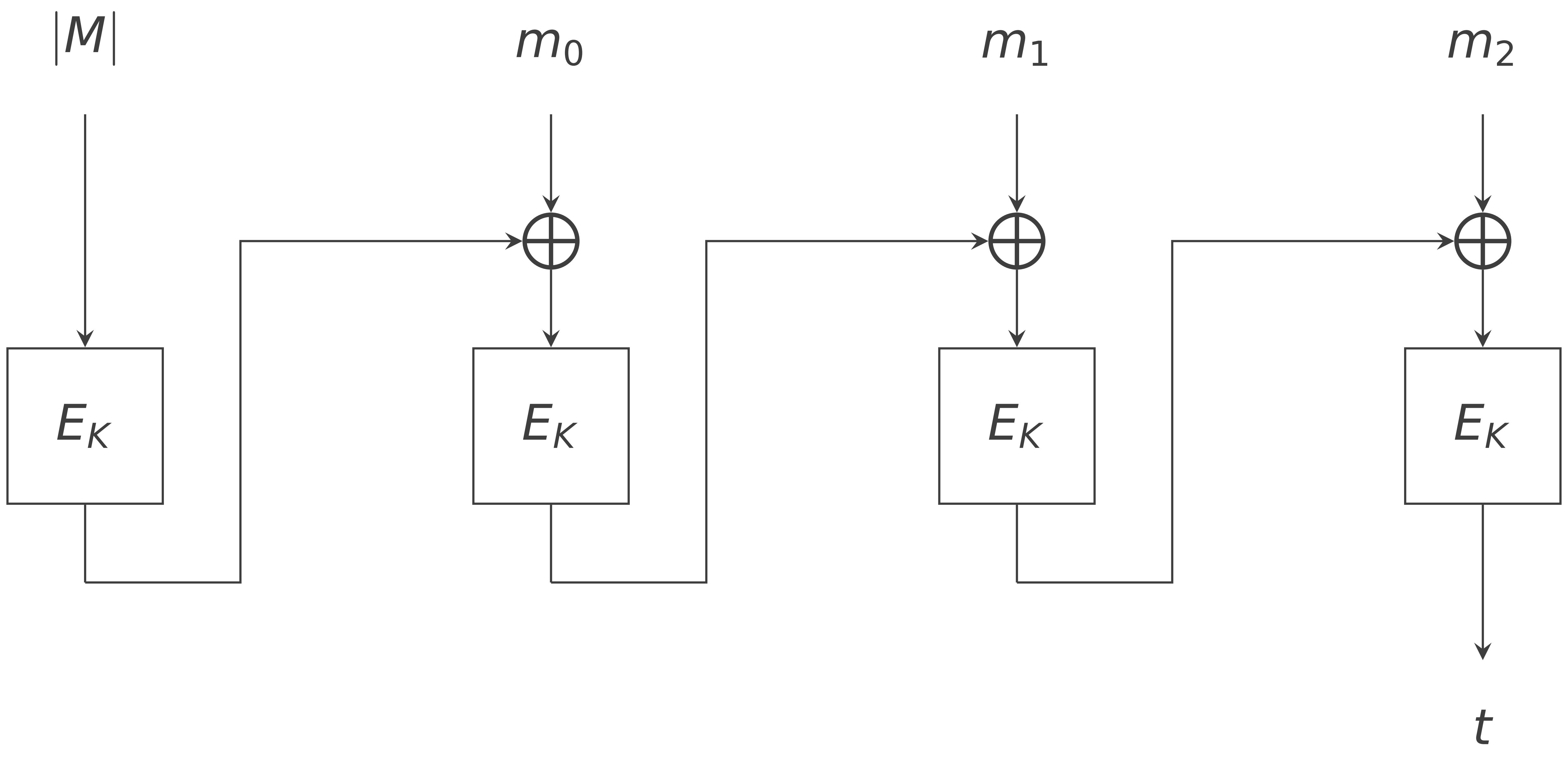
Difference from CBC encryption mode:

- No IV
- Only the final block output is considered

Security of CBC MAC

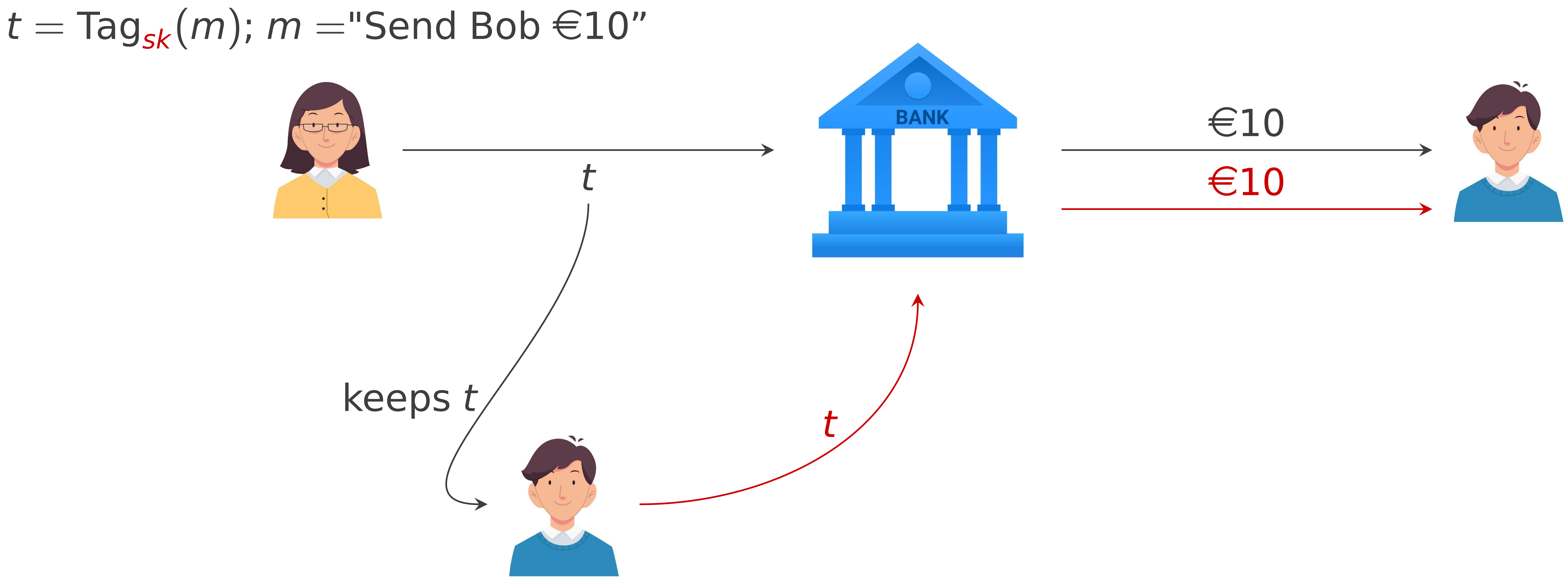
- Note that variable message length is allowed here i.e. adversary can obtains tags corresponding to two messages that are of different lengths
- CBC MAC is not secure if variable message length is allowed
- Can you think of a forgery attack?
- How to fix this?

Secure CBC MAC



- Here $M = m_0 || m_1 || m_2$ and $|M|$ is the length of the message

Replay Attack

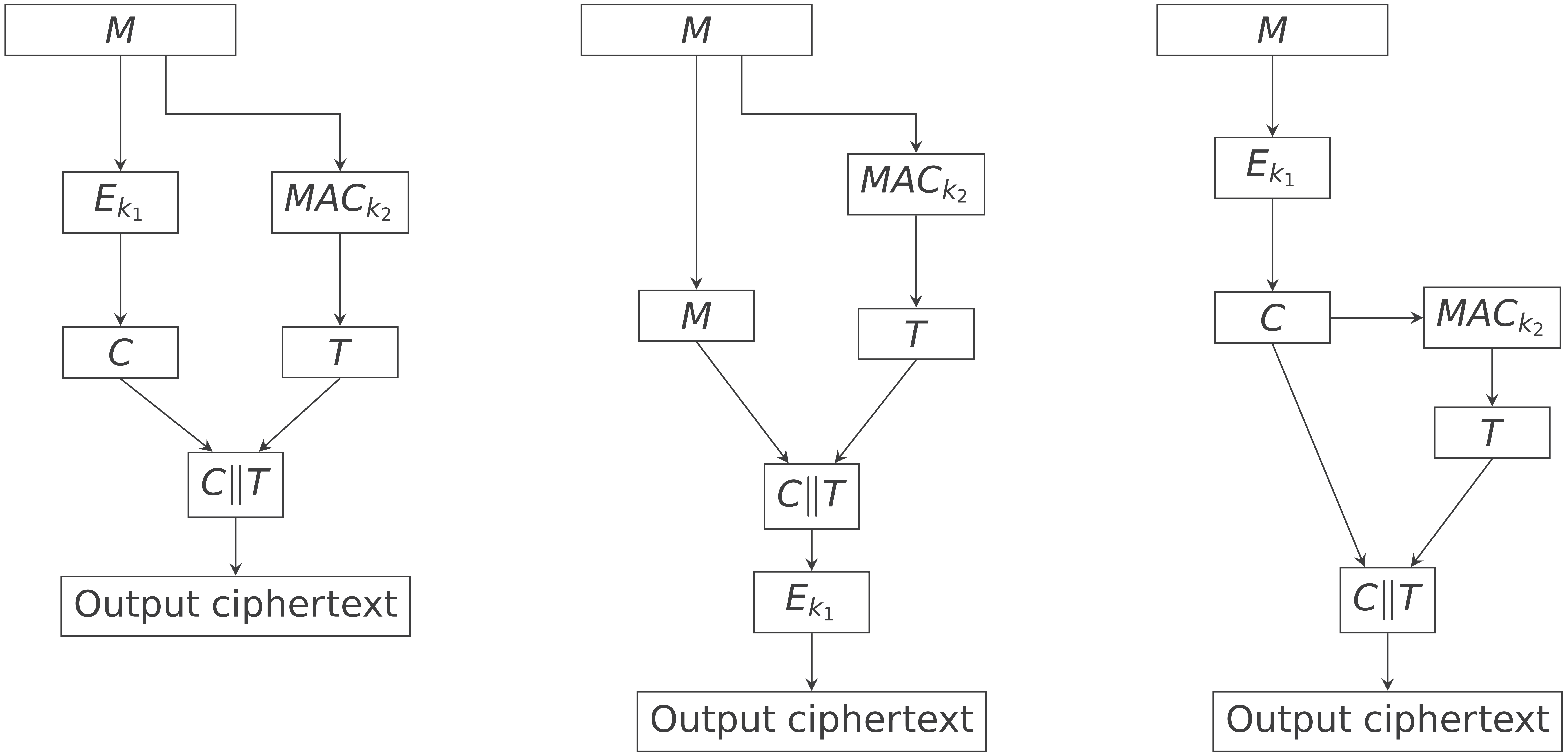


- Can not be protected with a stateless MAC
- Requires additional inputs: time-stamp, nonce

Authenticated Encryption

- Generic AE composition combines encryption and authentication
- There are 3 generic combinations possible
  - Encrypt and MAC (E&M)
  - Encrypt then MAC (EtM)
  - MAC then Encrypt (MtE)
- Modern approach: dedicated AE scheme (NOT covered in this course)

Generic Composition



Security of AE

- E&M: $C||T \leftarrow E_{k_1}(M)||MAC_{k_2}(M)$
  - Provides integrity to plaintext; No integrity to cipher text
  - It was used in SSH
  - It can not provide secure AE in general
- EtM: $C||T \leftarrow E_{k_1}(M)||MAC_{k_2}(C)$
  - It is secure when block cipher is secure and MAC is unforgeable under chosen message attack
  - Used in IPSec
- MtE: $E_{k_1}(M||MAC_{k_2}(M))$
  - No integrity if cipher text; until decryption is complete there is no way to assure the authenticity of message
  - Can not provide secure AE in general
  - It was used in TLS 1.2

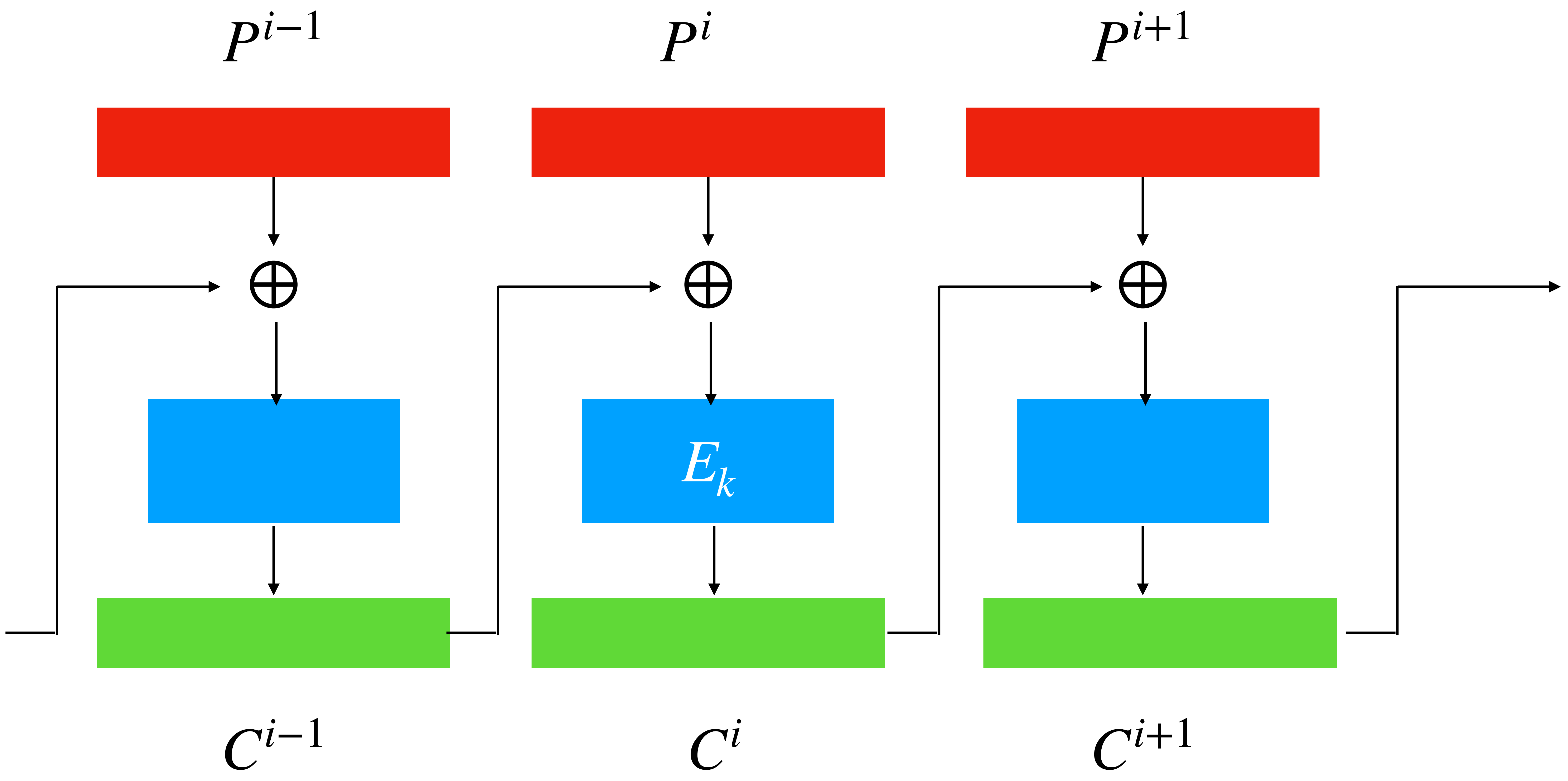
Transport Layer Security (TLS)

- A widely adopted security protocol
- It was proposed by the international standardisation organisation Internet Engineering Task Force (IETF), In 1999 the first version of TLS protocol was published
- Most recent version is TLS 1.3 published in 2018
- TLS vs. SSL: TLS evolved from SSL (Secure Socket Layer) encryption protocol. SSL was developed by Netscape. TLS 1.0 was developed as SSL 3.1
- TLS vs. HTTPS: On top of HTTP protocol TLS is implemented. For websites, TLS protected HTTPS is a standard practice.
- Why use it? TLS encryption provides web applications with data confidentiality and protect data breaches.

TLS Functionalities

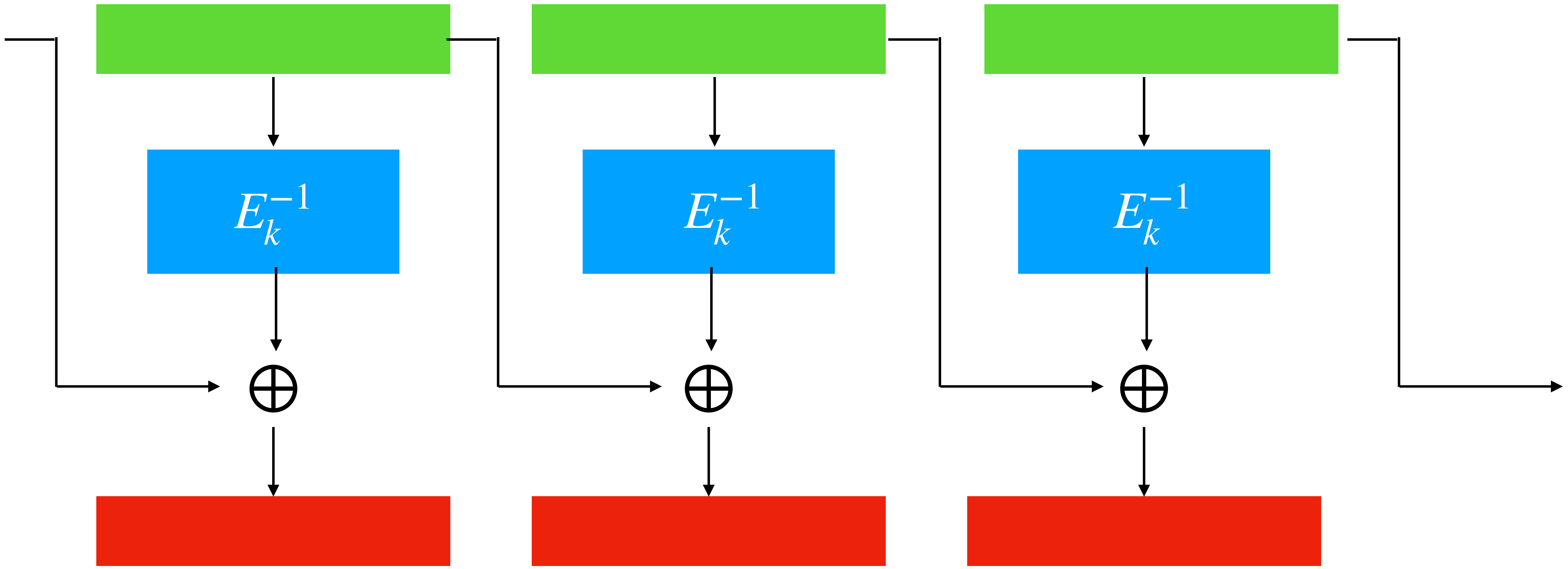
- TLS provides
  1. **Confidentiality**: by encryption of data
  2. **Authenticity**: ensures that the parties exchanging messages or data, are who they claim to be
  3. **Integrity**: ensures that the data is not tampered with or forged
- **TLS certificate**: The TLS certificate is installed on a web server. Such a certificate is issued by a CA (certificate authority). The certificate contains important information like holder's (server) identity, public key etc. [More on this in latter Lecture]
- References for TLS are given on the last slide

Recall: CBC Mode of Encryption



Note: Each blue box denotes block cipher with secret key k

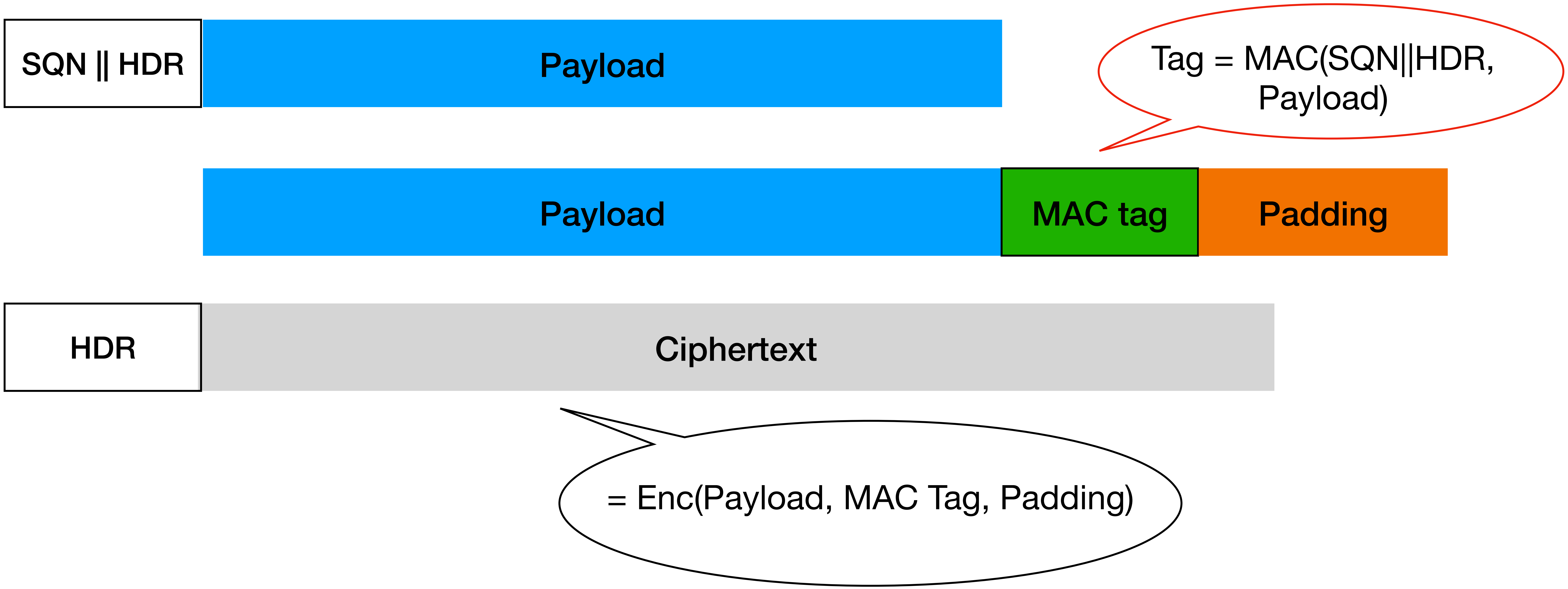
Recall: CBC Decryption



Note: Each blue box denotes decryption using a block cipher with secret key k

TLS Record Protocol

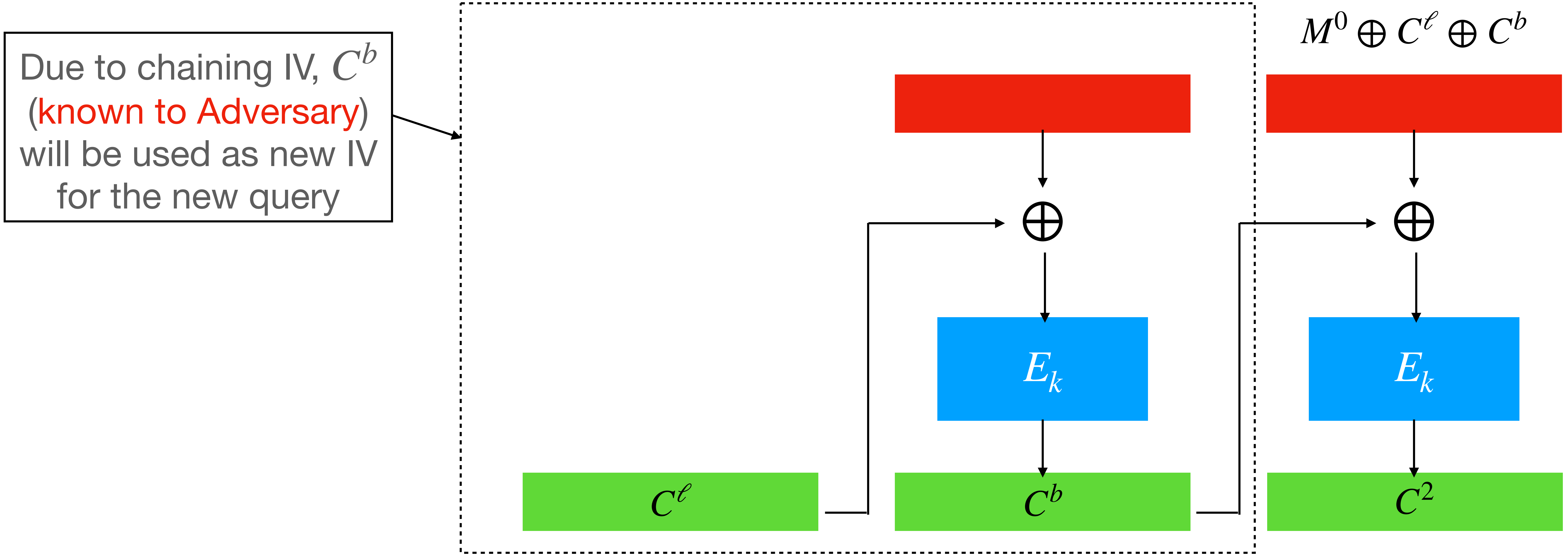
- Applies MAC-then-Encrypt



CBC Mode in TLS

- TLS 1.0: Uses chained IV
- Chained IV means the current IV is the ciphertext corresponding to the last block from the previous message
- This means IV is predictable.
- This way of using IV leads to an attack
  - It was first observed in 1999 by Rogaway against general CBC Mode
  - Dai and Moeller applied against TLS 1.0
  - Bard extended this to theoretical plaintext recovery attack in 2004/2006
  - Duong and Rizzo turned this to practical plaintext recovery attack in 2011

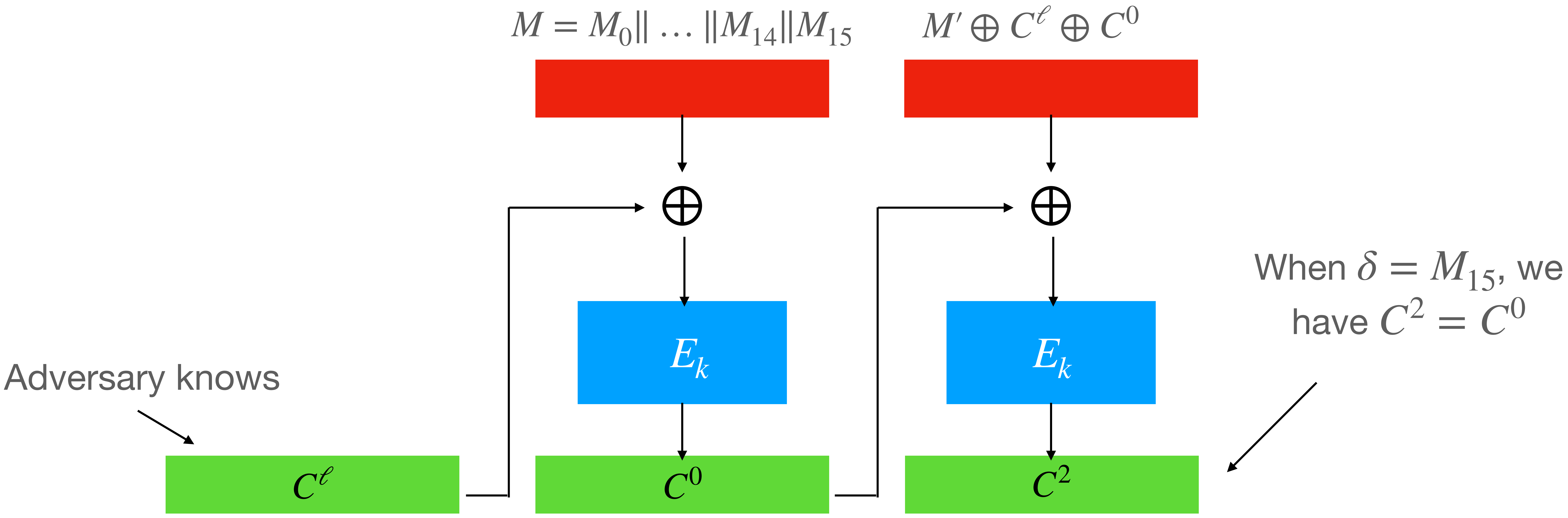
Idea of Predictable IV Attack



- When the unmarked red box has plaintext M^0, $C^b = C^2$

Predictable IV Attack: Plaintext Byte Recovery

- Assume that adversary knows 15 bytes $M_0, \ldots, M_{14}$ of a message block and trying to recover M_{15}. She uses the known 15 bytes to recover the 16th byte.
- She constructs a block$M' \oplus C^\ell \oplus C^0$ where $M' = M_0 || M_1 || \ldots || M_{14} || \delta$. Now iterate over all 256 possible values of δ. On average ~128 trials are needed to recover M_{15}



In Practice: BEAST Attack

- A chosen plaintext attack
- The assumptions behind the attack are not impractical. The BEAST (Browser Exploit Against TLS) attack was published in 2011.
- The attack was developed against TLS 1.0
- The chosen plaintext was injected using javascript that is put on a client's browser. This allows an adversary to send chosen message to a server from client's browser.
- It requires intercepting cipher text from earlier communication

Padding in TLS

- Variable length padding and maximum 256 bytes padding
- In TLS 1.0, 1.1, 1.2: Padding format is “00”, “01 01”, “02 02 02” etc. The maximum padding is a string “ff ff … ff”.
- **Padding rule**:
  - Add at least 1 byte of padding
  - If p bytes of padding is required then p copies of the number $p-1$ are added with byte representation

Decryption: Check Padding

- Two checks are required - padding check after decryption and verify MAC tag
- In TLS 1.0 error alerts `decryption_failed`. The decryption is considered invalid in two cases
    - Decrypted data is not an even multiple of block length (of the block cipher)
    - Padding is incorrect
- TLS 1.1 follows the same

THANK YOU!

Questions?

References

- More on TLS
    1. https://www.cloudflare.com/en-gb/learning/ssl/transport-layer-security-tls/
    2. https://www.ibm.com/docs/en/ibm-mq/9.2?topic=tls-how-provides-identification-authentication-confidentiality-integrity
    3. TLS 1.3 IETF documentation: https://datatracker.ietf.org/doc/html/rfc8446