Lecture 8: Block Cipher Modes of Operation

COSC362 Data and Network Security

Book 1: Chapters 7 and 12 – Book 2: Chapter 20

Spring Semester, 2021

Motivation

- Block ciphers encrypt SINGLE blocks of data.
- In applications, MANY blocks are encrypted sequentially.
- Breaking the plaintext into blocks and encrypting each separately are generally insecure.
- Many *modes of operation* standardised with different security and efficiency.
- Using block ciphers to provide authentication/integrity and/or confidentiality.

Outline

Outline

Why Different Modes?

- Designed to provide *confidentiality* for data OR *authentication* (and integrity) for data OR both:
  - Modes for confidentiality normally include *randomisation*.
- Different modes have:
  - Different efficiency properties
  - Different communication properties

Randomised Encryption

- Problem: the same plaintext block is encrypted to the same ciphertext block EVERY time:
  - Allowing patterns to be found in a long ciphertext.
- Prevention: randomising encryption schemes:
  - By using an *initialisation vector IV* which propagates through the entire ciphertext.
  - *IV* may require to be either unique or random.
- Another way to vary encryption:
  - By including a variable state which is updated with each block.

Efficiency

Features impacting on efficiency for practical usage (but not necessarily on security):

- Parallel processing:
  - Multiple plaintext blocks are encrypted in parallel.
  - Multiple ciphertext blocks are decrypted in parallel.
- Error propagation:
  - A bit error which occurs in the ciphertext results in multiple bit errors in the plaintext after decryption.

Padding

- Requiring the plaintext to consist of COMPLETE blocks.
- NIST suggests a padding method:
  - Append a single ’1’ bit to the data string.
  - Pad the resulting string by as few ’0’ bits, possibly none, as are necessary to complete the final block.
- Padding bits removed unambiguously if usage of this padding method is known:
  - Remove all trailing ’0’ bits after the last ’1’ bit.
  - Remove the single ’1’ bit.

Notations

- Plaintext message P (n blocks in length)
- t-th plaintext block P_t, for $1 \leq t \leq n$
- Ciphertext message C
- t-th ciphertext block C_t, for $1 \leq t \leq n$
- Key K
- Initialisation vector IV

All modes can be applied to any block cipher.
Example: AES when blocks are 128 bits in length.

NIST Standards

- ECB, CBC, CFB and OFB originally standardised for use with DES (1980).
- CTR mode initially added for use with AES (2001).
- SP 800-38A (2001): Confidentiality Modes
  - ECB, CBC, CFB and OFB.
  - An addendum defines Ciphertext Stealing.
- SP 800-38B (2005): CMAC Mode for Authentication.
- SP 800-38C (2004, updated 2007): CCM Mode.
- SP 800-38D (2007): Galois/Counter Mode (GCM).
- SP 800-38E (2010): XTS-AES Mode for Storage Devices.
- SP 800-38F (2012): Key Wrapping.
- SP 800-38G (2016): Format-Preserving Encryption.

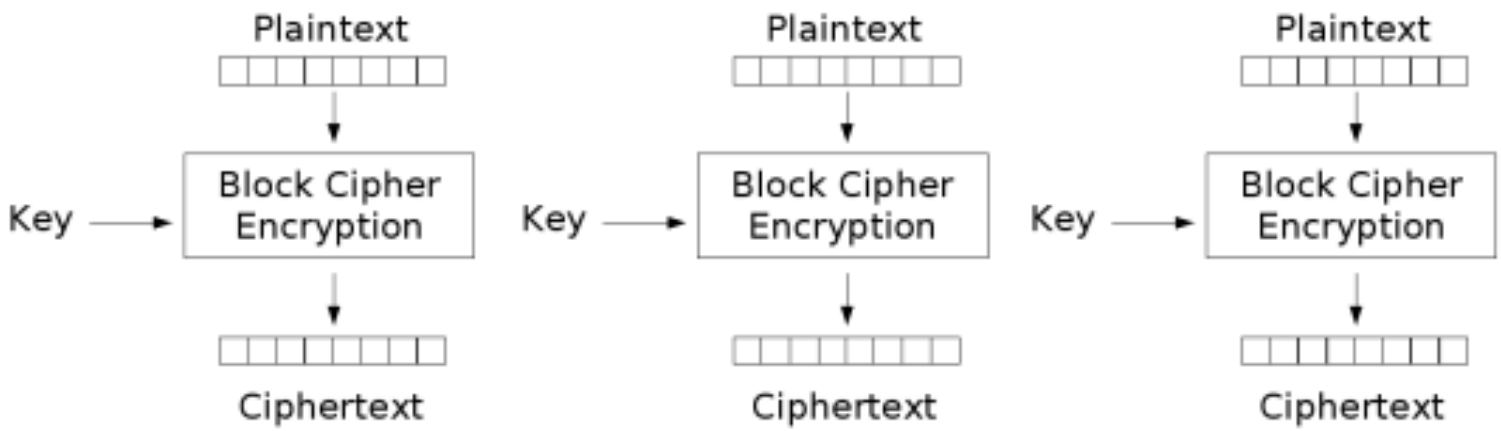
Outline

ECB Mode Encryption

Basic mode of a block cipher.

Encryption:

- $C_t = E(P_t, K)$
- Plaintext block P_t is encrypted with key K to produce ciphertext block C_t

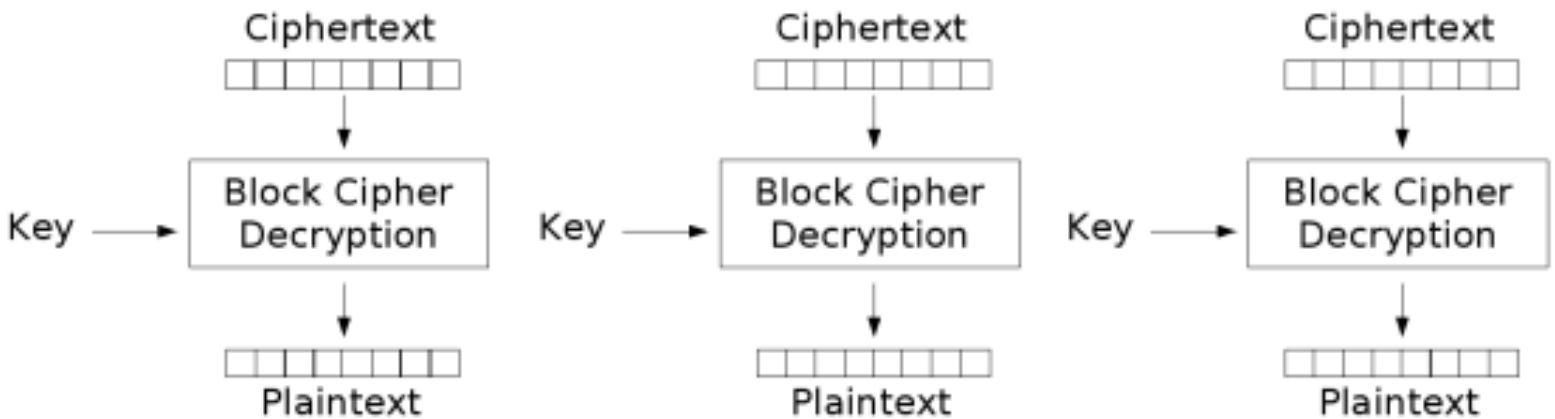


Electronic Codebook (ECB) mode encryption

ECB Mode Decryption

Decryption:

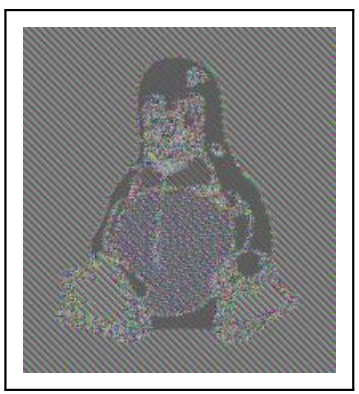
- $P_t = D(C_t, K)$
- Ciphertext block C_t is decrypted with key K to produce plaintext block P_t



Electronic Codebook (ECB) mode decryption

ECB Mode Properties

Randomised	✘
Padding	Required
Error propagation	Errors propagate within blocks
IV	None
Parallel encryption	✓
Parallel decryption	✓



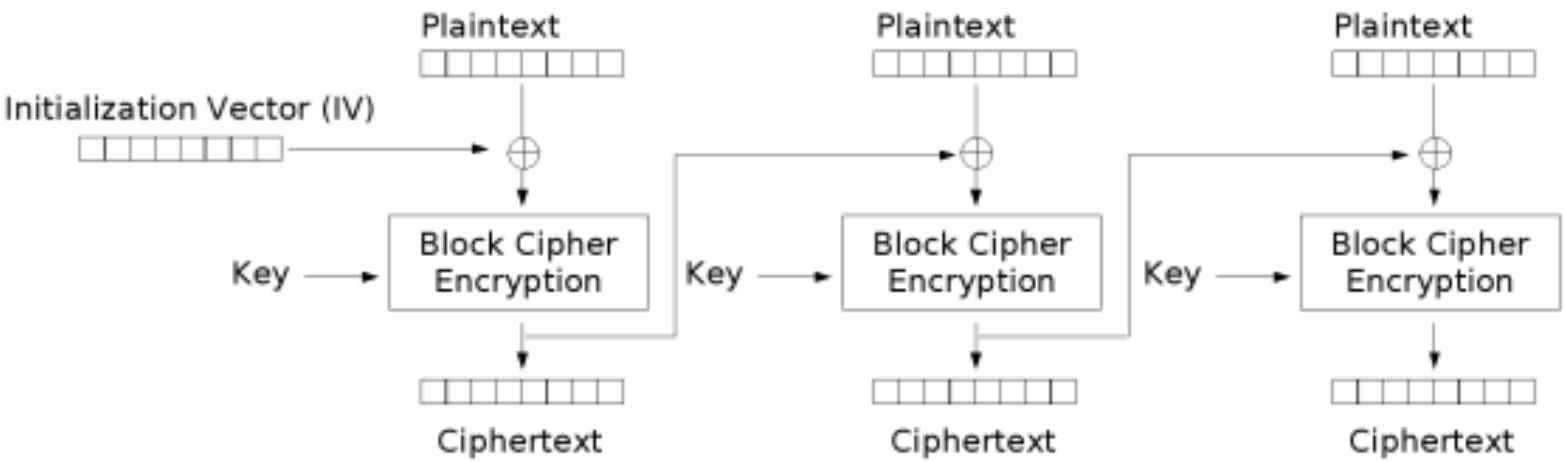
Deterministic $\Rightarrow$ not normally used for bulk encryption.

CBC Mode Encryption

"Chaining" blocks together.

Encryption:

- $C_t = E(P_t \oplus C_{t-1}, K)$ s.t. $C_0 = IV$
  - IV is chosen at random and sent together with ciphertext blocks
- P_t is XORed with previous ciphertext block C_{t-1}, and encrypted with key K to produce C_t

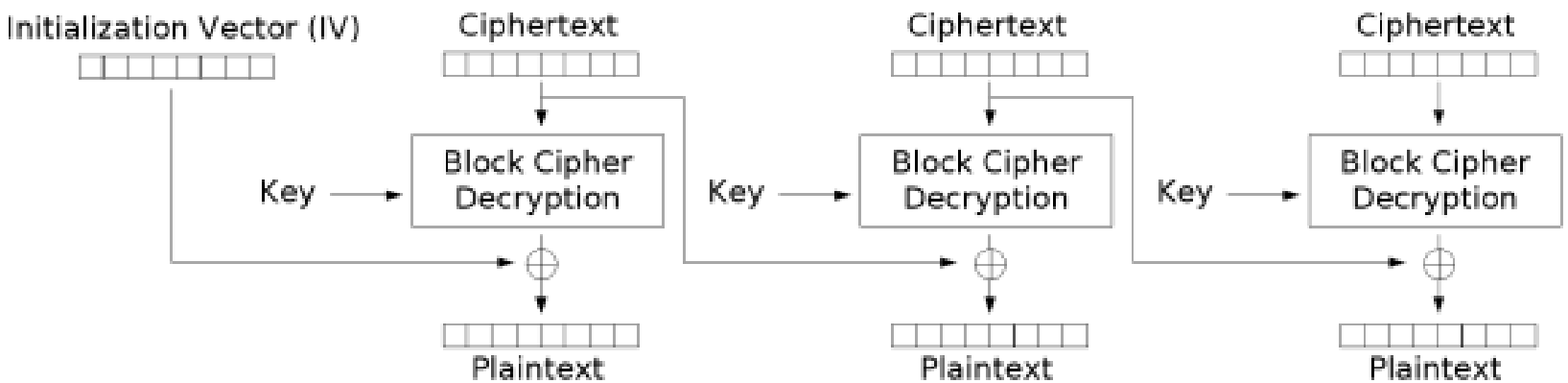


Cipher Block Chaining (CBC) mode encryption

CBC Mode Decryption

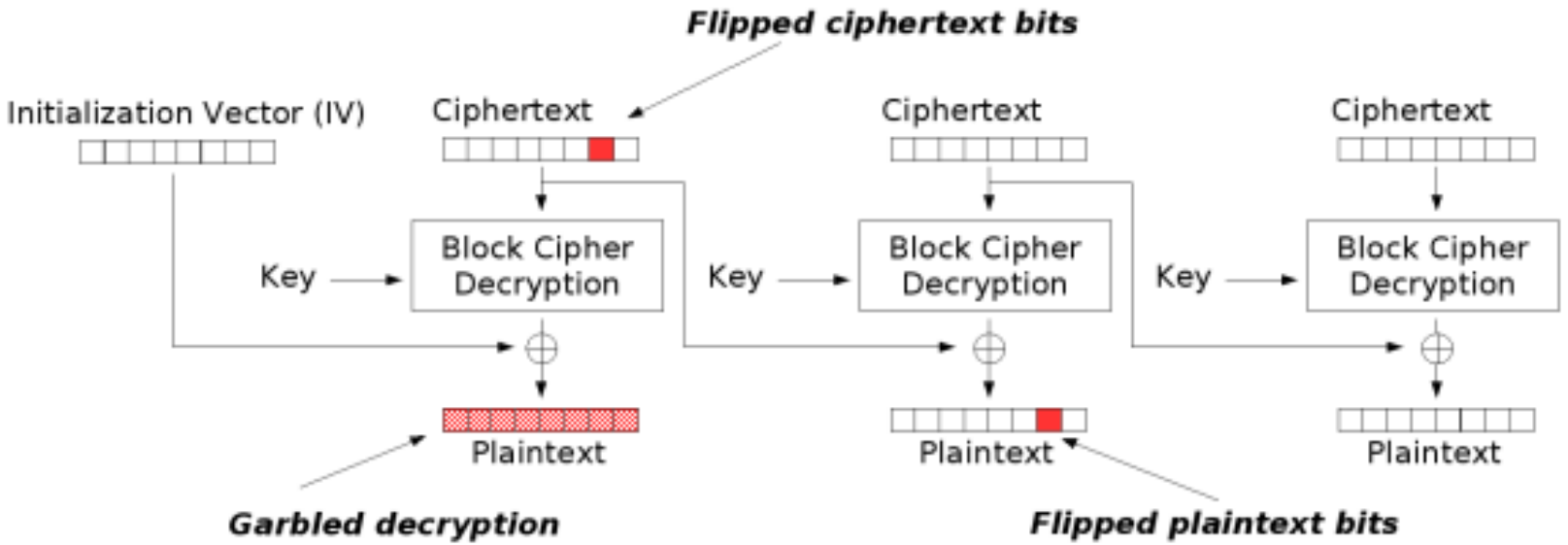
Decryption:

- $P_t = D(C_t, K) \oplus C_{t-1}$ s.t. $C_0 = IV$
- C_t is decrypted with key K, and XORed with previous ciphertext block C_{t-1} to produce P_t



Cipher Block Chaining (CBC) mode decryption

CBC Mode Error Propagation



Modification attack or transmission error for CBC

CBC Mode Properties

Randomised	✓
Padding	Required
Error propagation	Errors propagate within blocks and into specific bits of next block
IV	Must be random
Parallel encryption	✘
Parallel decryption	✓

- Commonly used for bulk encryption.
- Common choice for channel protection in TLS up to TLS 1.2.

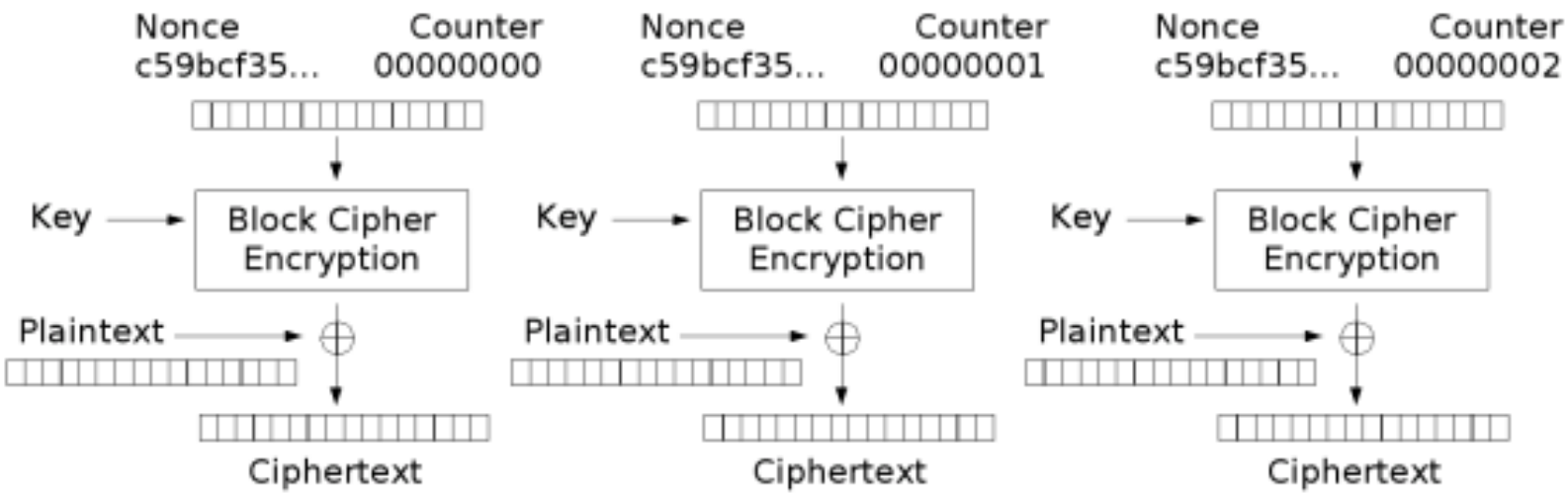
CTR Mode

- *Synchronous stream cipher mode* (see later).
- A counter and a nonce are used, initialised using a randomly chosen value N:
  - $T_t = N||t$ is the concatenation of the nonce N and block number t
  - $O_t = E(T_t, K)$
  - This is XORed with the plaintext block P_t
- Propagation of channel errors: A one-bit change in the ciphertext produces a one-bit change in the plaintext at the SAME location.

CTR Mode Encryption

Encryption:

- $C_t = O_t \oplus P_t$
- Plaintext block P_t is XORed with O_t

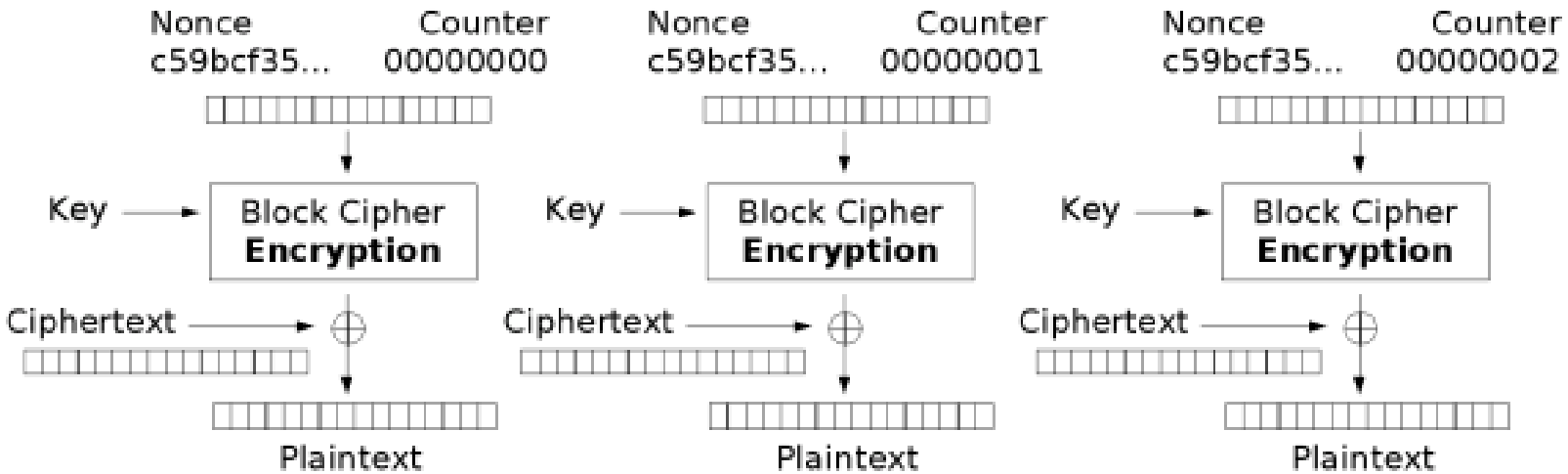


Counter (CTR) mode encryption

CTR Mode Decryption

Decryption:

- $P_t = O_t \oplus C_t$
- Ciphertext block C_t is XORed with O_t



Counter (CTR) mode decryption

CTR Mode Properties

Randomised	✓
Padding	Not required
Error propagation	Errors occur in specific bits of current block
IV	Nonce must be unique
Parallel encryption	✓
Parallel decryption	✓

- Good for access to specific plaintext blocks without decrypting the whole stream.
- Basis for authenticated encryption in TLS 1.2.

Outline

Message Integrity

- Ensuring that messages are not altered in transmission.
- Treating *message integrity* and *message authentication* as the same thing.
- Preventing an adversary from re-ordering, replacing, replicating and deleting message blocks in order to alter the received message.
- Proving message integrity is independent from using encryption for confidentiality.

Message Authentication Code (MAC)

- Cryptographic mechanism to ensure message integrity.
- $T = \mathrm{MAC}(M, K)$:
  - Inputs: arbitrary-length message M and secret key K
  - Output: (short) fixed-length *tag* T
- Parties Alice and Bob share a common key K.
- Alice wants to send a message M to Bob:
  - Alice computes $T = \mathrm{MAC}(M, K)$
  - Alice sends the message M and adjoins its tag T
  - Bob computes $T' = \mathrm{MAC}(M', K)$ on received message M' and checks that $T' = T$

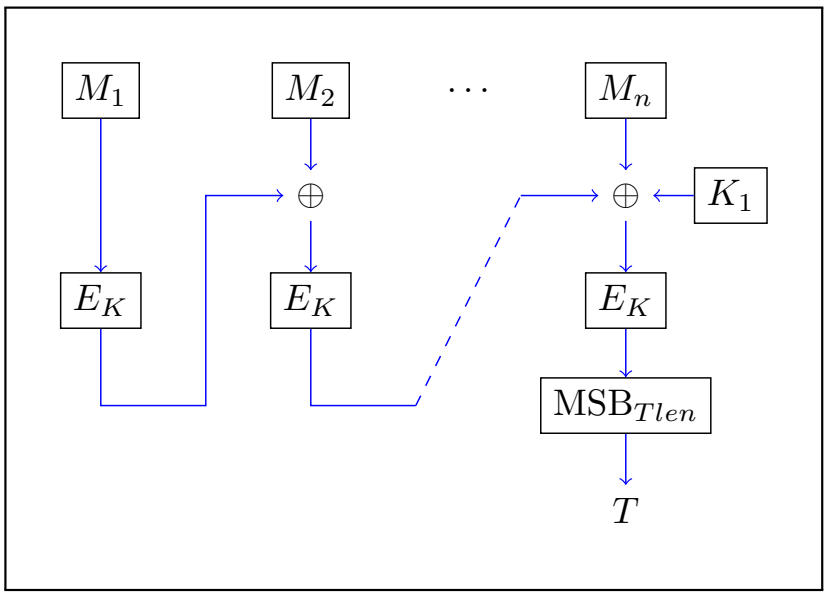
MAC Properties

- Providing sender authentication to the message:
  - Only Alice and Bob CAN produce T from M.
  - If $T' = T$ then Bob concludes that the message received M' was sent by Alice and has not been modified in transit (either intentionally or accidentally).
  - If $T' \neq T$ then Bob concludes that (M', T) was not sent by Alice.
- Basic security property: Unforgeability
  - Infeasible to produce M and T s.t. $T = \text{MAC}(M, K)$ without knowledge of K

Basic CBC-MAC

- Using a block cipher to create a MAC providing message integrity (but not confidentiality).
- IV must be fixed and public, and can be set to all zeros.
- CBC-MAC with random IV is NOT secure.
- P is the message with n blocks.
- $T = \text{CBC-MAC}(P, K)$ as follows:
  - $C_t = E(P_t \oplus C_{t-1}, K)$ for $0 \leq t \leq n$, s.t. $C_0 = IV$
  - $T = C_n$
- Unforgeable as long as the message length is fixed.

Cipher-based MAC (CMAC)



- Standardised: NIST secure version of CBC-MAC.
- 2 keys K_1, K_2 are derived from the original key K.
- K_1 or K_2 XORed with M_n (padding as necessary).
- IV set to be the all 0 block.
- CBC encryption on the message M.
- T is some number of MSB bits of final block.

CMAC

- NIST standard allows any number of bit *Tlen* to be chosen for tag T:
  - Recommendation of 64 bits to avoid guessing attacks.
- Standard recommends MAC tag T to be of length at least $\log_2(lim/R)$ with:
  - *lim* is a limit on how many invalid messages are detected before K is changed.
  - R is the acceptable probability (risk) that a false message is accepted.

Outline

Authenticated Encryption

- 2 types of input data:
  - Payload: both encrypted and authenticated.
  - Associated data: only authenticated.
- 2 modes specified in:
  - NIST SP-800-38C in 2004 for Counter with CBC-MAC (CCM) Mode
  - NIST SP-800-38D in 2007 for Galois/Counter (GCM) Mode
- Both modes use CTR mode for confidentiality but add integrity in different ways.
- Both used in TLS 1.2 and TLS 1.3 (latest versions).

Counter with CBC-MAC (CCM) Mode

Combining CBC-MAC for authentication of ALL data (payload and associated data) and CTR mode encryption for the payload:

- Inputs: nonce N for CTR mode, payload P of $Plen$ bits, and associated data A.
- Format N, A, P to produce a set of blocks.
- Compute CBC-MAC tag T for these blocks with length $Tlen$.
- Use CTR mode to compute blocks of key stream $S_0, S_1, \cdots, S_m$ where $m = \lceil Plen/128 \rceil$.
- Output: $C = (P \oplus MSB_{Plen}(S))||(T \oplus MSB_{Tlen}(S_0))$ where $S = S_1, \cdots, S_m$.

CCM Mode Format

- Complex format with restrictions w.r.t. different standards.
- Lengths of N, P are included in the 1st block.
- If A is non-zero then formatted from the 2nd block onwards including its length.
- Example: TLS 1.2
  - Tag T is 8 bytes
  - CTR mode nonce N is 12 octets
  - Max payload size is $2^{24} - 1$ bytes