Diyala University
College of Engineering
Computer & Software
Engineering Department

Fourth Year 2012/2013

Cryptographic Data Integrity Algorithms

Chapter 4/Part2

Message Authentication Codes: MAC

PRESENTED BY
DR. ALI J. ABBOUD

4. Objectives

- **Authentication Functions.**
- **More to authentication than simple Encryption?.**
- **Authentication based on Hash Functions.**
- **Message authentication code (MAC).**
- **Basic uses of MAC.**
- **MAC based on Block Ciphers.**
- **MAC based on Hash Functions.**
- **Authentication Encryption.**
- **CCM Approach.**

4.1 Authentication Functions

- Three types of authentication exist
  - Message encryption – the ciphertext serves as authenticator
  - Message authentication code (MAC) – a public function of the message and a secret key producing a fixed-length value to serve as authenticator
    - This does not provide a digital signature because A and B share the same key
  - Hash function – a public function mapping an arbitrary length message into a fixed-length hash value to serve as authenticator
    - This does not provide a digital signature because there is no key

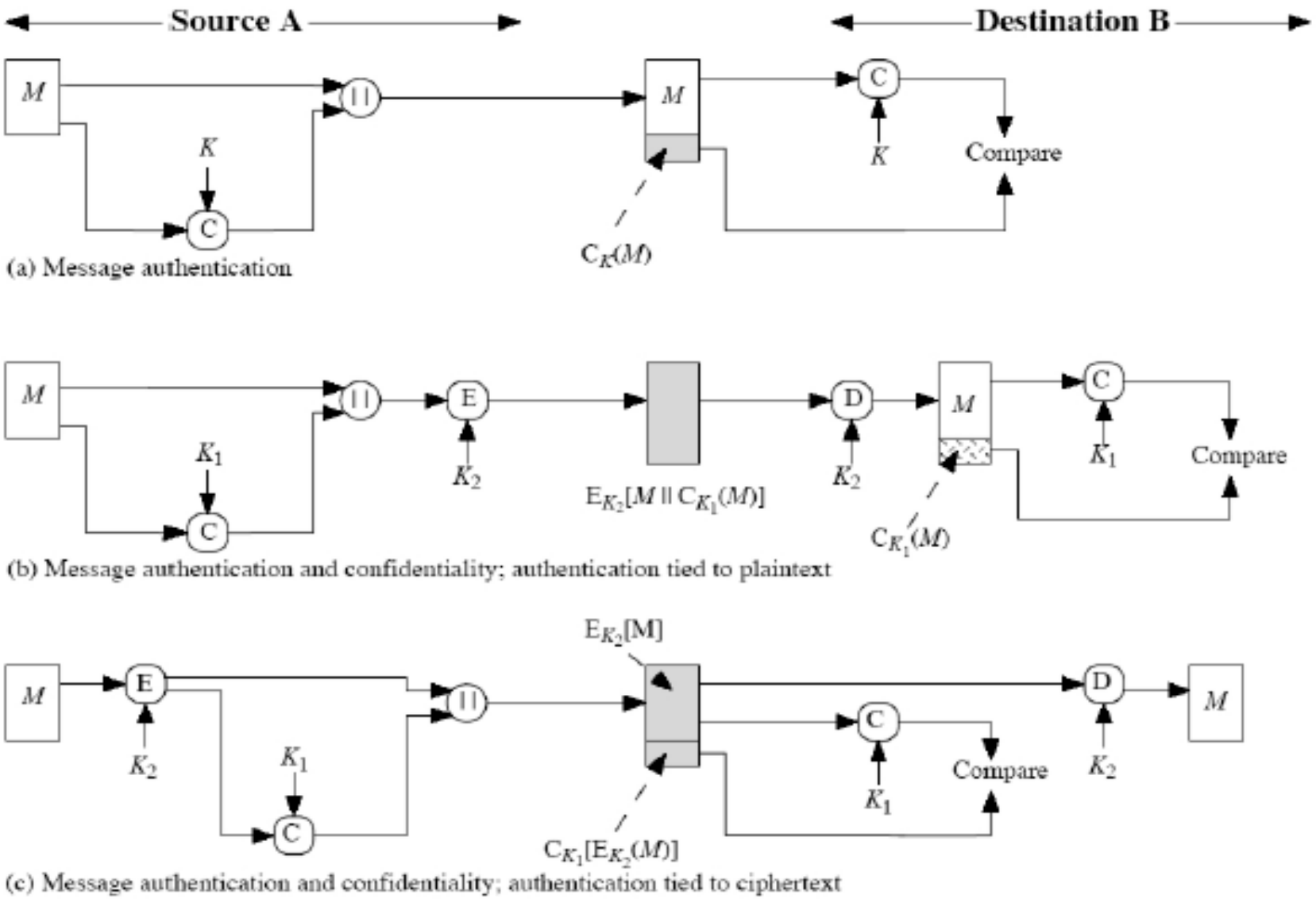
4.2 More to authentication than simple encryption?

- Often one needs alternative authentication schemes than just encrypting the message
  - Sometimes one needs to avoid encryption of full messages due to legal requirements
  - Encryption and authentication may be separated in the system architecture
  - If a message is broadcast to several destinations in a network (such as a military control center), then it is cheaper and more reliable to have just one node responsible to evaluate the authenticity – message will be sent in plain with an attached authenticator
  - If one side has a heavy load, it cannot afford to decrypt all messages – it will just check the authenticity of some randomly selected messages
  - *If the message is sent encrypted, it is of course protected over the network. However, once the receiver decrypts the message, it is no longer secure. Using a different type of authentication protects the message also on the local computer*

4.3 Message authentication code (MAC)

- To generate the MAC of a message M, Alice gives M and the secret key K to a MAC function C: MAC=$C_K(M)$
  - Alice will send M plus the MAC to Bob
  - Bob has the same secret key K and generates the MAC himself to check the match
- Typical attacks on MACs
  - Produce an illegitimate message with the same signature as a given (or chosen) legitimate one
  - Produce a valid MAC for an illegitimate message
- Requirements for MACs
  - The MAC function is in general many-to-one – messages are arbitrarily long and the MAC has fixed length, thus there will be more than one message with the same MAC
  - Computationally easy to compute the MAC
  - Knowing M and $C_K(M)$ it is computationally infeasible to construct another message M' with $C_K(M')= C_K(M)$
  - $C_K(M)$ is uniformly distributed – if the attacker chooses a random bit pattern of length n, the chances of it being the correct signature is 2^{-n}
  - If M' is obtained from M by certain transformations (even switching one bit), then the probability that the two have the same MAC is 2^{-n}

4.4 Basic uses of MAC



4.4 Basic uses of MAC

$A \rightarrow B: M \| C_K(M)$

- Provides authentication
  - Only A and B share K

(a) Message authentication

$A \rightarrow B: E_{K_2}[M \| C_{K_1}(M)]$

- Provides authentication
  - Only A and B share K_1
- Provides confidentiality
  - Only A and B share K_2

(b) Message authentication and confidentiality: authentication tied to plaintext

$A \rightarrow B: E_{K_2}[M] \| C_{K_1}(E_{K_2}[M])$

- Provides authentication
  - Using K_1
- Provides confidentiality
  - Using K_2

(c) Message authentication and confidentiality: authentication tied to ciphertext

4.5 MAC based on Block Ciphers

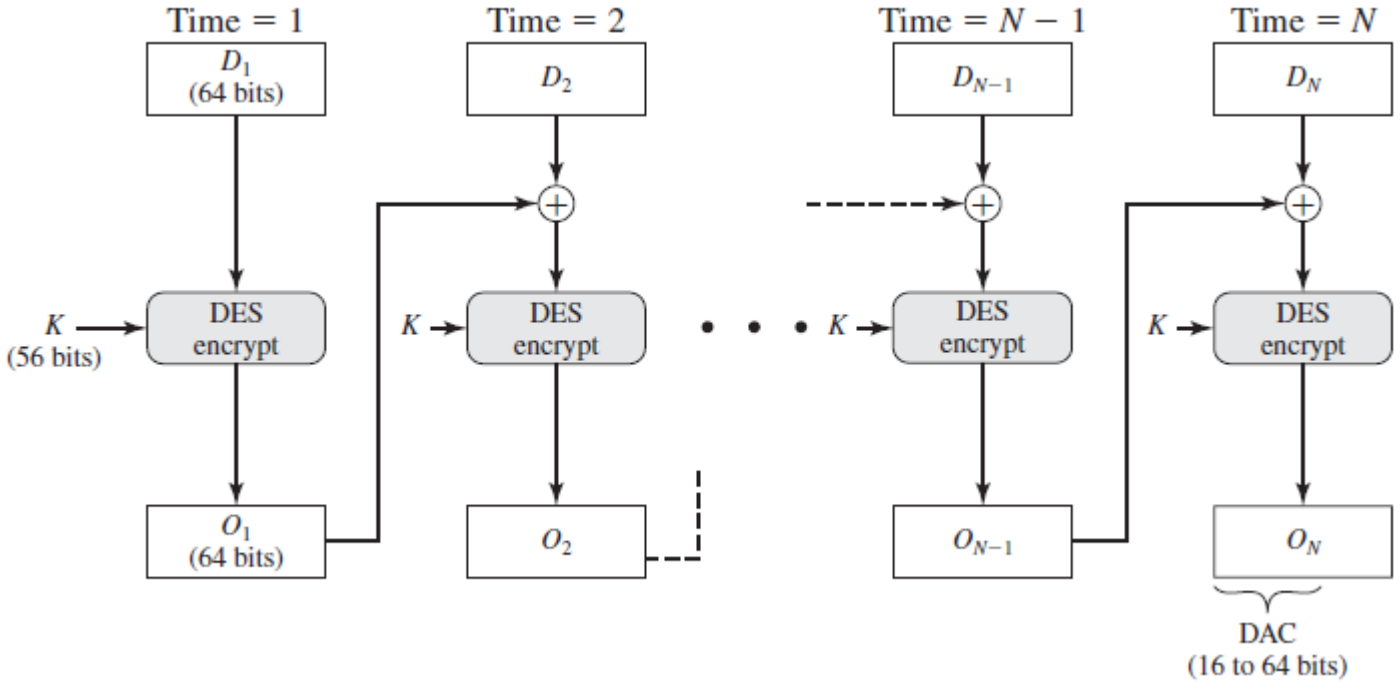
Data Authentication Algorithm

The **Data Authentication Algorithm** (DAA), based on DES, has been one of the most widely used MACs for a number of years. The algorithm is both a FIPS publication (FIPS PUB 113) and an ANSI standard (X9.17). However, as we discuss subsequently, security weaknesses in this algorithm have been discovered, and it is being replaced by newer and stronger algorithms.

The algorithm can be defined as using the cipher block chaining (CBC) mode of operation of DES (Figure) with an initialization vector of zero. The data (e.g., message, record, file, or program) to be authenticated are grouped into contiguous 64-bit blocks: $D_1, D_2, \ldots, D_N$. If necessary, the final block is padded on the right with zeroes to form a full 64-bit block. Using the DES encryption algorithm E and a secret key K, a data authentication code (DAC) is calculated as follows

$$
\begin{aligned}
O_1 &= \mathrm{E}(K, D) \\
O_2 &= \mathrm{E}(K, [D_2 \oplus O_1]) \\
O_3 &= \mathrm{E}(K, [D_3 \oplus O_2]) \\
&\bullet \\
&\bullet \\
&\bullet \\
O_N &= \mathrm{E}(K, [D_N \oplus O_{N-1}])
\end{aligned}
$$

4.5 MAC based on Block Ciphers



The DAC consists of either the entire block O_N or the leftmost M bits of the block, with $16 \leq M \leq 64$.

4.6 MAC based on Hash Functions

HMAC algorithm

- Interest in recent years in developing a MAC based on a hash function
  - MD5 and SHA-1 run faster than symmetric block ciphers such as DES
  - Code for hash functions widely available
  - No export restrictions for cryptographic hash functions
    - Cryptographic functions (even those used in MAC) restricted
- Hash values not intended for MAC – they are not protected by secret keys
  - Some protection needs to be built on top of the hash value
- The one approach that gained wide support is HMAC (RFC 2104) included in IP security and SSL
- Requirements for HMAC
  - Use existing hash functions
  - The hash function can be easily replaced by another one – treat the hash function as a black box
  - Preserve the performance of the hash function
  - Use and handle keys in a simple way
  - Well understood cryptographic analysis of the strength of the authentication mechanism

4.6 MAC based on Hash Functions

HMAC algorithm

- Idea: append a secret key to the message and compute the hash value
  - To avoid a brute-force attack, apply the hash twice to mangle thoroughly the bits of the key with those of the message
- H=embedded hash function
- IV=initial value to the hash function
- M=message input to HMAC (including the padding specific to the hash function)
- Y_i=i-th block of M
- L=number of blocks in M
- b=number of bits in a block
- n=length of the hash code
- K=secret key, if its length is greater than b – will be given as input to the hash function to produce n-bit key
- K^+=K padded with 0 on the left to make a b-bit key, if the original length of K is smaller than b
- ipad=0x36 repeated b/8 times
- opad=0x5C repeated b/8 times

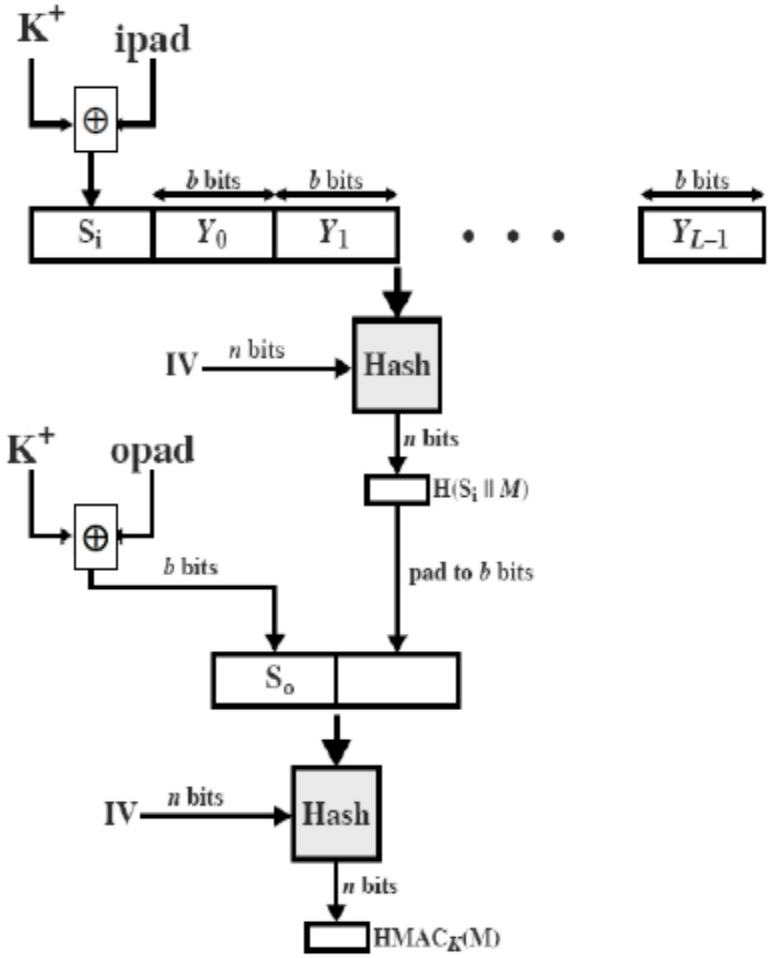
$$\mathbf{HMAC_K(M)=H[\ (K^+ \oplus opad)\ ||\ H[(K^+ \oplus ipad)\ ||\ M]\ ]}$$

4.6 MAC based on Hash Functions

HMAC algorithm

- H=embedded hash function
- IV=initial value input to hash function
- M=message input to HMAC (including the padding specific to the hash function)
- Y_i= the i-th block of M
- L=number of blocks in M
- b=number of bits in a block
- n=length of hash code produced by the embedded hash function
- K=secret key, if its length is greater than b – will be given as input to the hash function to produce n-bit key
- K^+=K padded with 0 on the left to make a b-bit key, if the original length of K is smaller than b
- ipad=0x36 repeated b/8 times
- opad=0x5C repeated b/8 times

HMACK(M)=H[*(K+⊕ opad) || H[(K+⊕ ipad) || M]*]



4.6 MAC based on Hash Functions

Strength of HMAC

- Brute-force attack requires an effort on the level 2^{n-1} for a key of length n
- Birthday attack
  - The main idea in this attack is that Eve can compute the hash values of many messages and try to find a match
  - In HMAC she is unable to do that because the hash is protected by a secret key
  - Eve will have to rely on messages that she observes on the link: for MD5 she will have to wait in average for 2^{64} messages generated using the same key
    - On a 1 Gbps-link she needs to observe a continuous stream of messages with no change in the key for about 250 000 years
    - With SHA-1 2^{80} messages are needed
    - For HMAC, using MD5 is secure (and fast)

4.7 Authentication Encryption

- Authenticated encryption (AE) is a term used to describe encryption systems that simultaneously protect confidentiality and authenticity (integrity) of communications.
- Many applications and protocols require both forms of security, but until recently the two services have been designed separately.
- There are four common approaches to providing both confidentiality and encryption for a message **M** :

  - **HtE: Hash-then-encrypt.** First compute the cryptographic hash function over M as $h = H(M)$. Then encrypt the message plus hash function: $E(K, (M||h))$.
  - **MtE: MAC-then-encrypt.** Use two keys. First authenticate the plaintext by computing the MAC value as $T = MAC(K_1, M)$. Then encrypt the message plus tag: $E(K_2, (M \| T)$. This approach is taken by the SSL/TLS protocols
  - **EtM: Encrypt-then-MAC.** Use two keys. First encrypt the message to yield the ciphertext $C = E(K_2, M)$. Then authenticate the ciphertext with $T = MAC(K_1, C)$ to yield the pair (C, T). This approach is used in the IPsec protocol
  - **E&M: Encrypt-and-MAC.** Use two keys. Encrypt the message to yield the ciphertext $C = E(K_2, M)$. Authenticate the plaintext with $T = MAC(K_1, M)$ to yield the pair (C, T). These operations can be performed in either order. This approach is used by the SSH protocol

4.7 Authentication Encryption

- **Both decryption and verification are straightforward for each approach. For HtE, MtE, and E&M, decrypt first, then verify. For EtM, verify first, then decrypt. There are security vulnerabilities with all of these approaches .**

- **The HtE approach is used in the Wired Equivalent Privacy (WEP) protocol to protect WiFi networks. This approach had fundamental weaknesses and led to the replacement of the WEP protocol.**

- **Counter with Cipher Block Chaining-Message Authentication Code (CCM) is an example of approach that is standardized by NIST for authenticated encryption systems.**

- **The CCM mode of operation was standardized by NIST specifically to support the security requirements of IEEE 802.11 WiFi wireless local area networks but can be used in any networking application requiring authenticated encryption.**

- **CCM is a variation of the encrypt-and-MAC approach to authenticated encryption.**

- **The key algorithmic ingredients of CCM are the AES encryption algorithm , the CTR mode of operation , and the CMAC authentication algorithm.**

- **A single key is used for both encryption and MAC algorithms**

4.8 CCM Approach

The input to the CCM encryption process consists of three elements:

1. Data that will be both authenticated and encrypted. This is the plaintext message P of data block.
2. Associated data A that will be authenticated but not encrypted. An example is a protocol header that must be transmitted in the clear for proper protocol operation but which needs to be authenticated.
3. A nonce N that is assigned to the payload and the associated data. This is a unique value that is different for every instance during the lifetime of a protocol association and is intended to prevent replay attacks and certain other types of attacks.

4.8 CCM Approach

SP 800-38C defines the authentication/encryption process as follows.

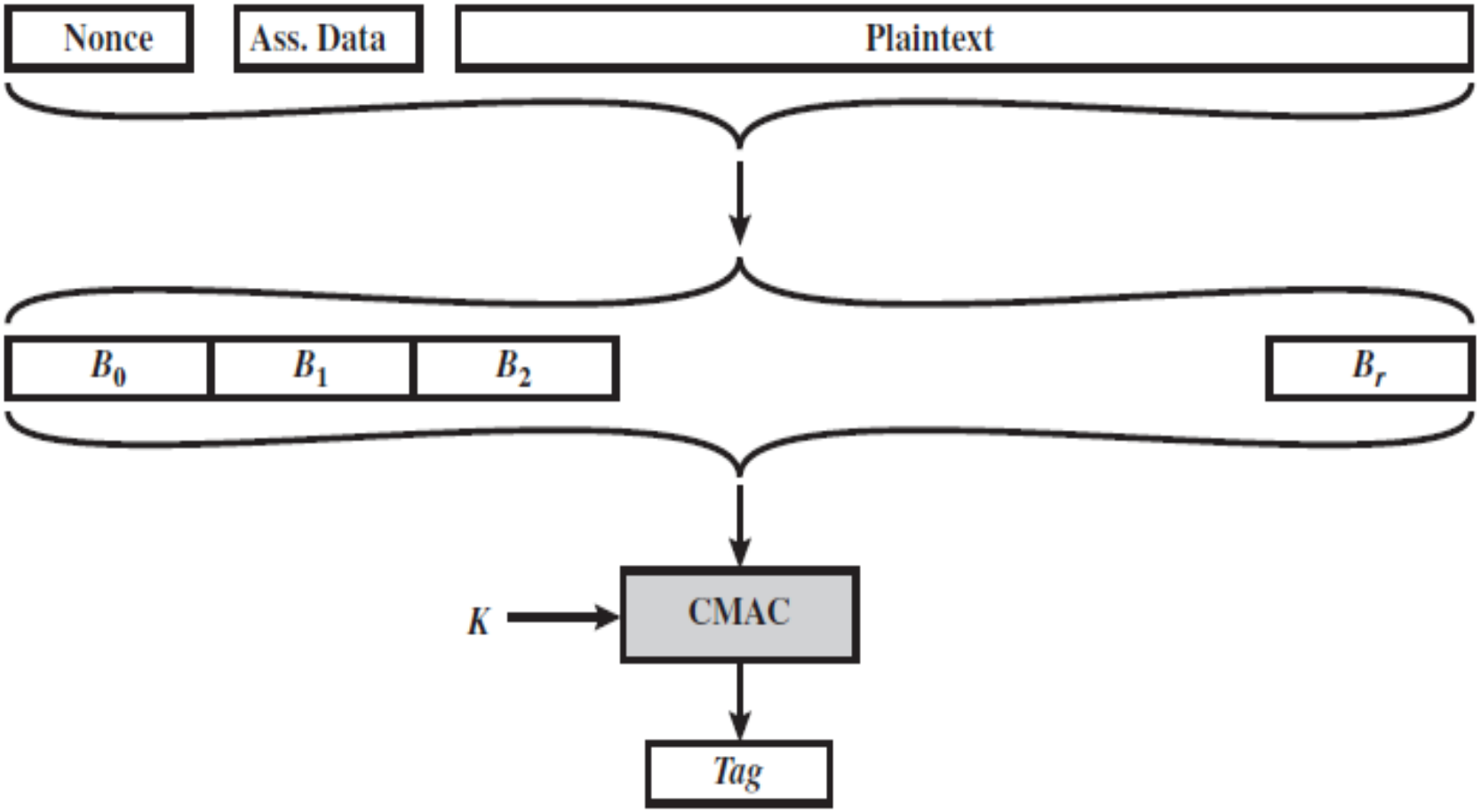
1. Apply the formatting function to (N, A, P) to produce the blocks $B_0, B_1, \ldots, B_r$.
2. Set $Y_0 = \mathrm{E}(K, B_0)$.
3. For $i = 1$ to r, do $Y_i = \mathrm{E}(K, (B_i \oplus Y_{i-1}))$.
4. Set $T = \mathrm{MSB}_{Tlen}(Y_r)$.
5. Apply the counter generation function to generate the counter blocks $Ctr_0, Ctr_1, \ldots, Ctr_m$, where $m = \lceil Plen/128 \rceil$.
6. For $j = 0$ to m, do $S_j = \mathrm{E}(K, Ctr_j)$.
7. Set $S = S_1 \| S_2 \| \cdots \| S_m$.
8. Return $C = (P \oplus \mathrm{MSB}_{Plen}(S)) \| (T \oplus \mathrm{MSB}_{Tlen}(S_0))$.

4.8 CCM Approach

For decryption and verification, the recipient requires the following input: the ciphertext C, the nonce N, the associated data A, the key K, and the initial counter Ctr_0. The steps are as follows.

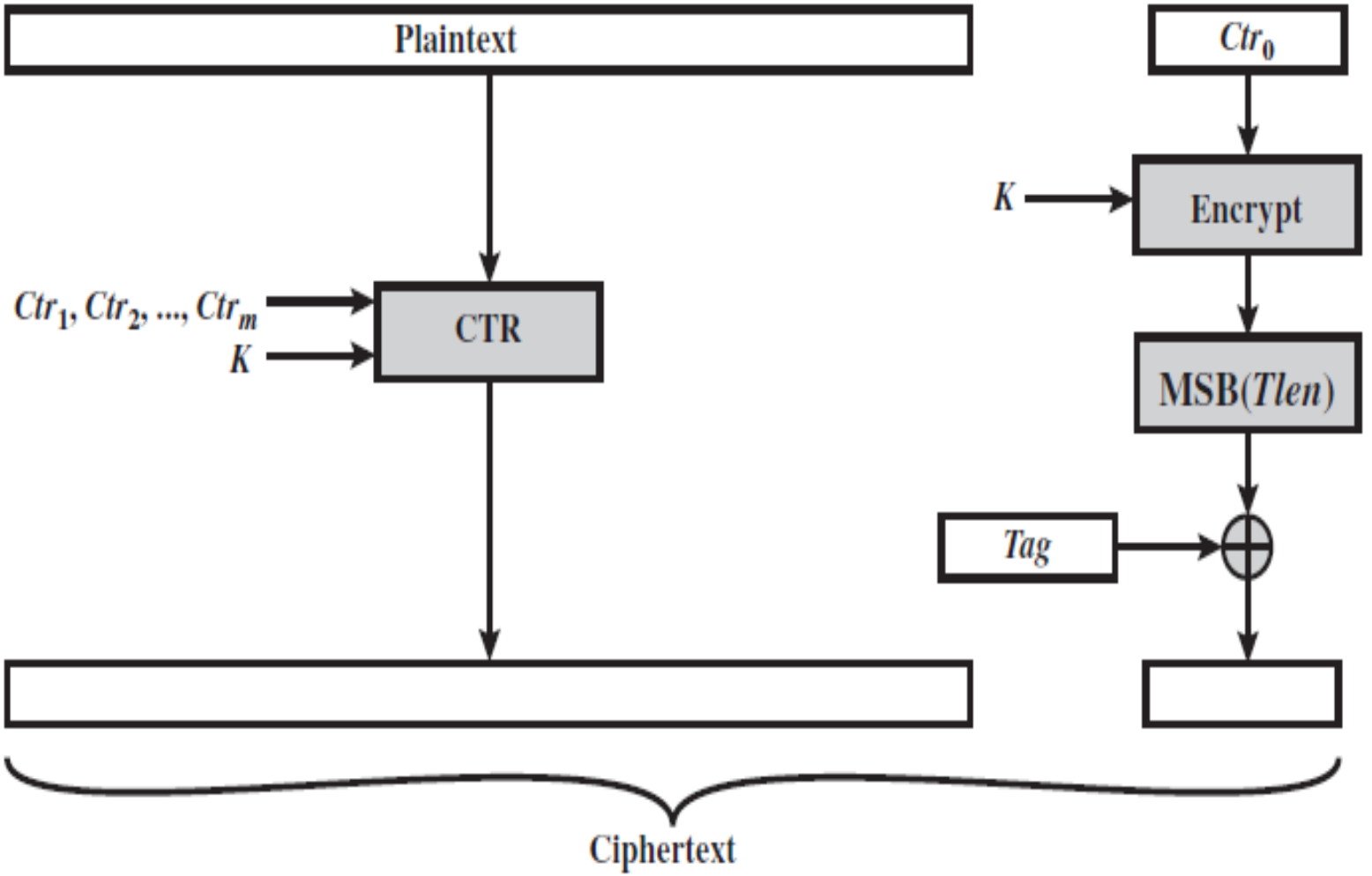
1. If $Clen \leq Tlen$, then return INVALID.
2. Apply the counter generation function to generate the counter blocks $Ctr_0, Ctr_1, \ldots, Ctr_m$, where $m = \lceil Clen/128 \rceil$.
3. For $j = 0$ to m, do $S_j = \mathrm{E}(K, Ctr_j)$.
4. Set $S = S_1 \| S_2 \| \ldots \| S_m$.
5. Set $P = \mathrm{MSB}_{Clen-Tlen}(C) \oplus \mathrm{MSB}_{Clen-Tlen}(S)$.
6. Set $T = \mathrm{LSB}_{Tlen}(C) \oplus \mathrm{MSB}_{Tlen}(S_0)$.
7. Apply the formatting function to (N, A, P) to produce the blocks $B_0, B_1, \ldots, B_r$.
8. Set $Y_0 = \mathrm{E}(K, B_0)$.
9. For $i = 1$ to r, do $Y_i = \mathrm{E}(K, (B_i \oplus Y_{i-1}))$.
10. If $T \neq \mathrm{MSB}_{Tlen}(Y_r)$, then return INVALID, else return P.

4.8 CCM Approach



(a) Authentication

4.8 CCM Approach



(b) Encryption

End of Chapter 4/Part2