Bahan kuliah IF4020 Kriptografi

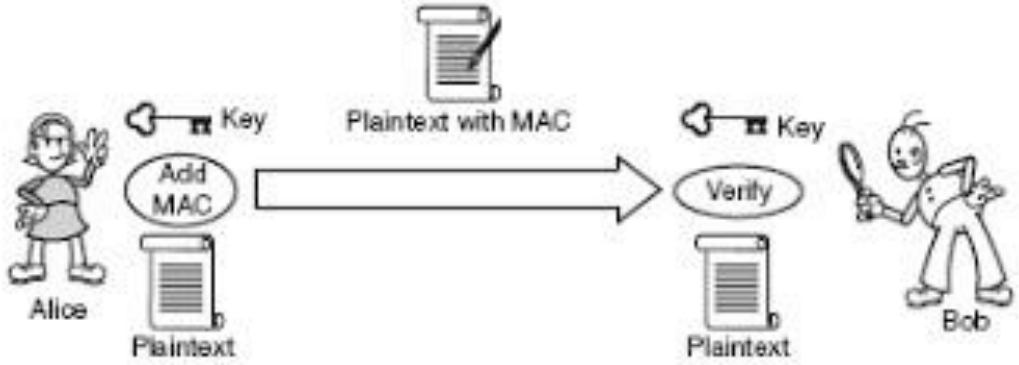


Figure 9.3 Message Authentication Code (MAC)

MAC
(*Message Authentication Code*)

Oleh: Rinaldi Munir

Program Studi Informatika
Sekolah Teknik Elektro dan Informatika
ITB – 2024

Definisi

- MAC (*message authentication code*): kode kecil berukuran tetap (*fixed*) yang dihasilkan dari pesan dan kunci untuk mengotentikasi pengirim dan memeriksa integitas pesan.

$$MAC = C_K(M)$$

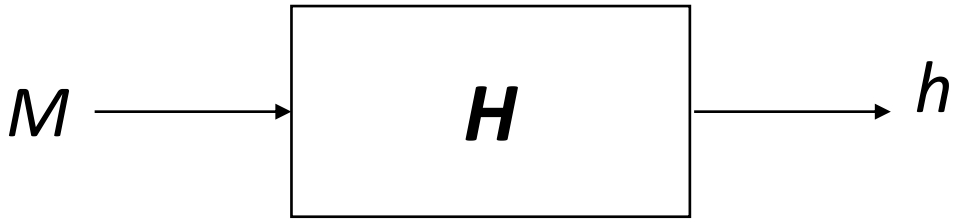
MAC = kode otentikasi pesan

C = algoritma *MAC*

K = kunci rahasia

- Nama lainnya: *Message Integrity Code* (MIC)

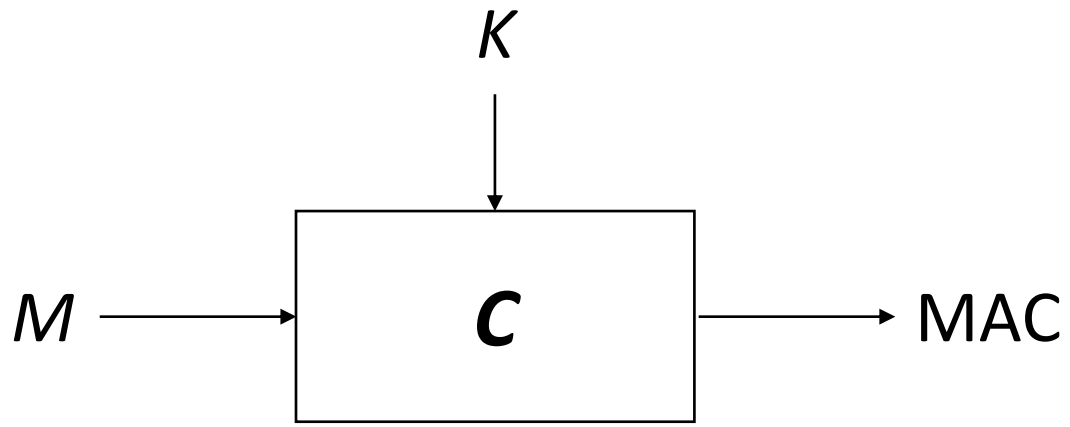
Perbedaan Algoritma MAC dengan Fungsi Hash biasa



$h = H(M)$

Message digest dengan fungsi *hash*

(tidak membutuhkan kunci)



$\text{MAC} = C_K(M)$

MAC dengan fungsi *hash*

(membutuhkan kunci)

- *MAC* dilekatkan (*embed*) pada pesan sebagai “signature”
- *MAC* digunakan untuk memeriksa integritas (keaslian) pesan dan otentikasi pengirim.
- Jika MAC yang dikirim sama dengan MAC yang dihitung oleh penerima, maka pesan masih asli.

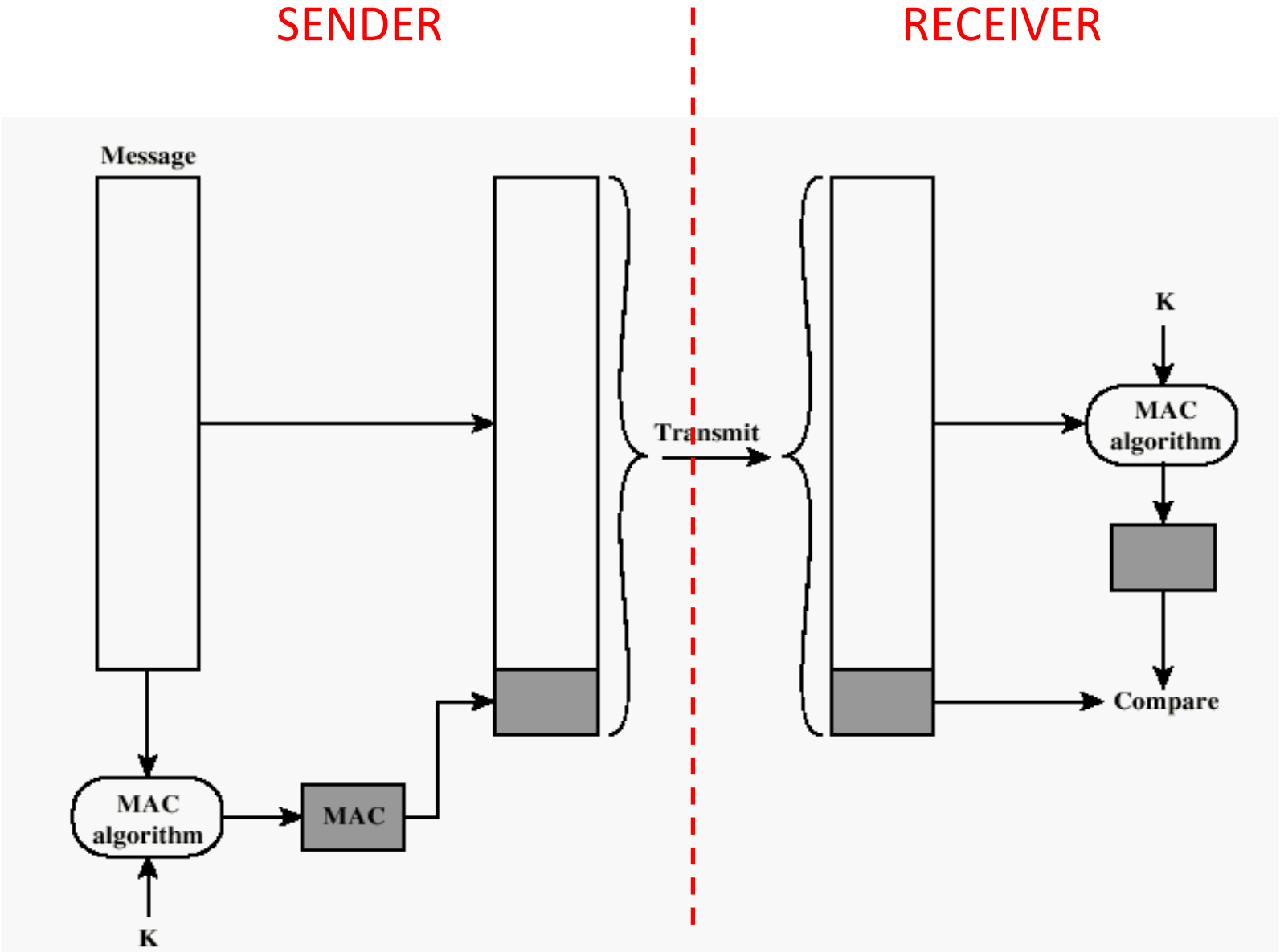
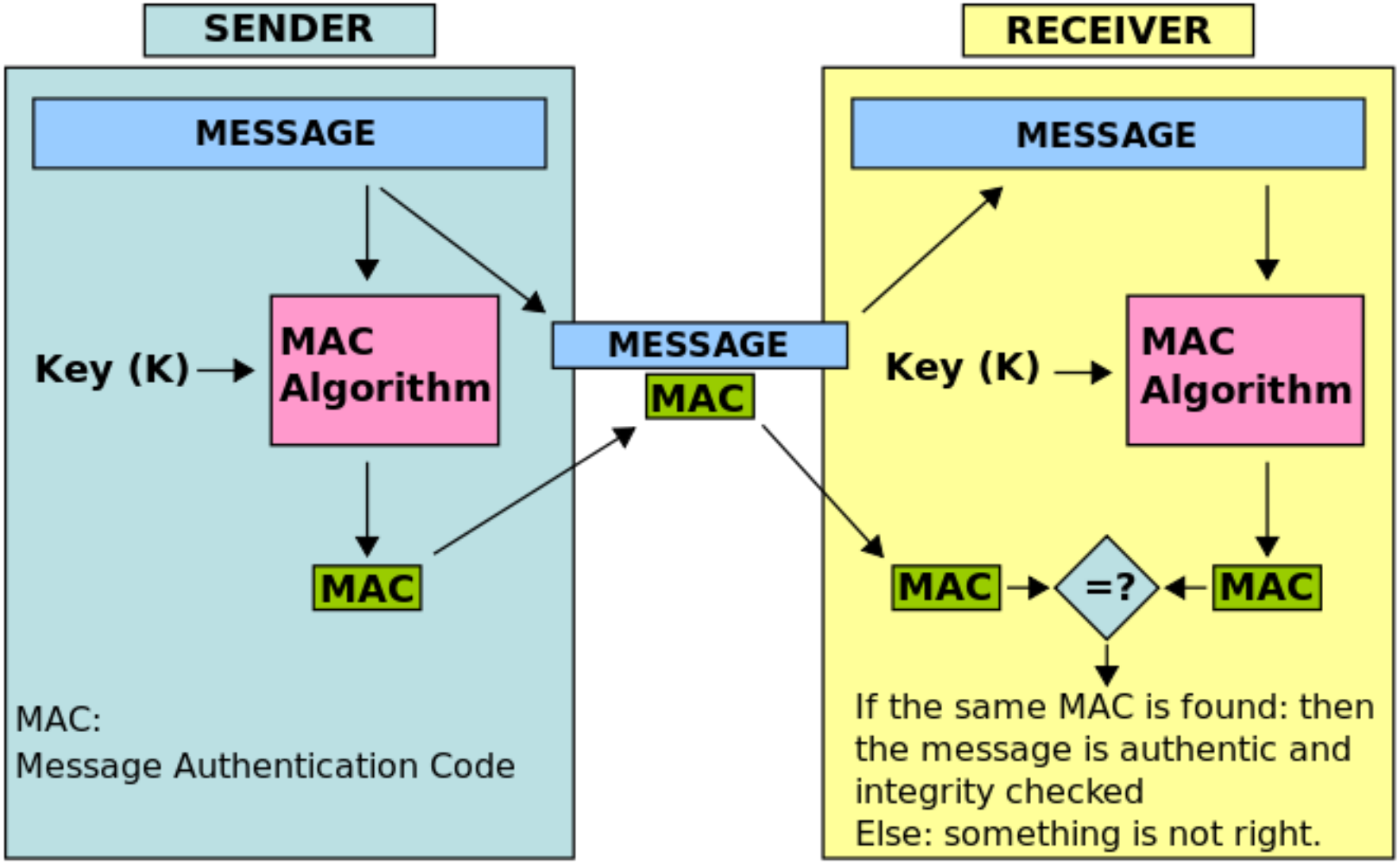


Figure 3.1 Message Authentication Using a Message Authentication Code (MAC)


SENDER
RECEIVER
MESSAGE
MESSAGE
Key (K)
MAC Algorithm
MESSAGE
MAC
Key (K)
MAC Algorithm
MAC
MAC
=?
MAC
MAC:
Message Authentication Code
If the same MAC is found: then the message is authentic and integrity checked
Else: something is not right.

Kegunaan MAC

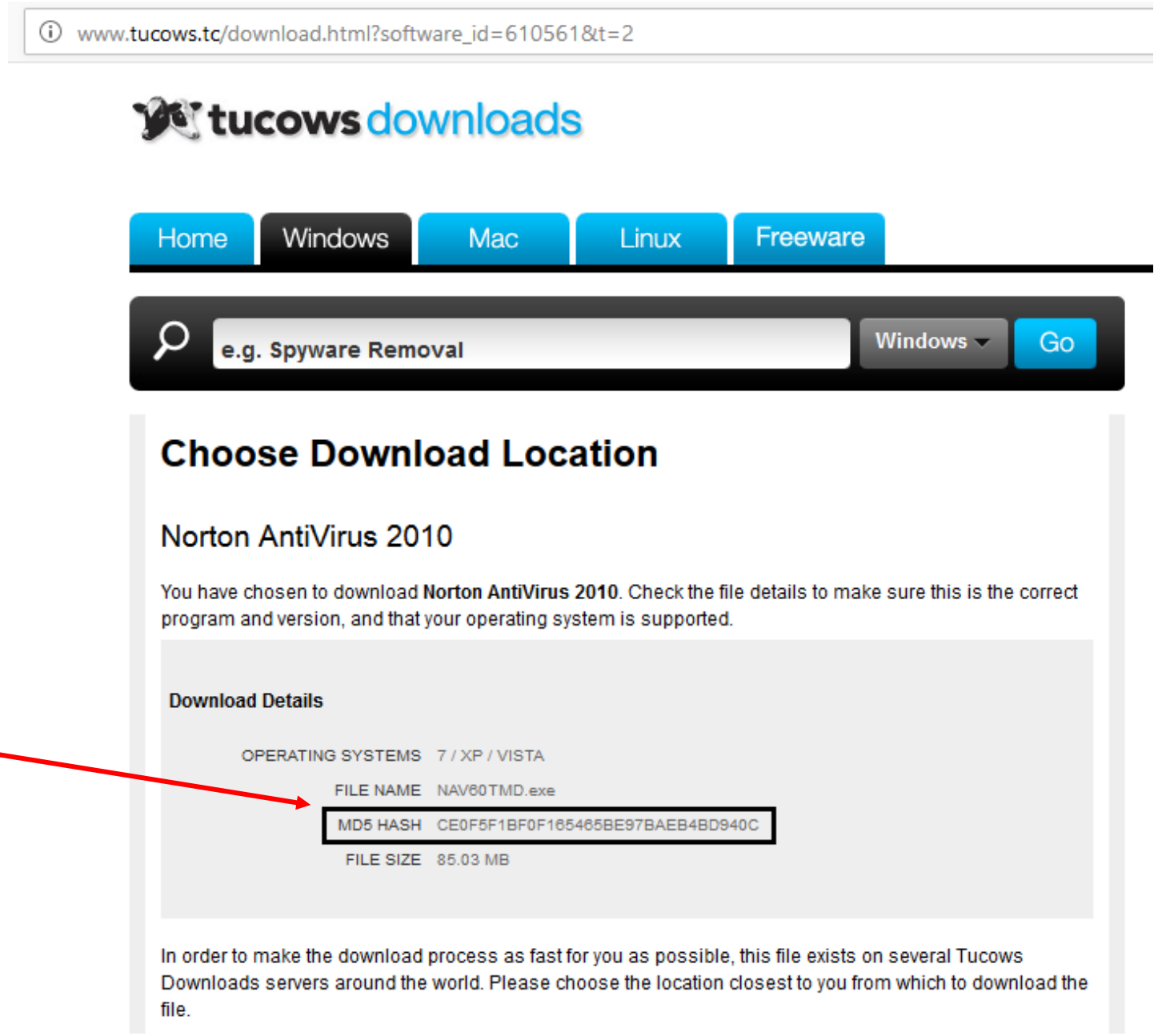
1. Seperti *message digest*, MAC digunakan untuk memeriksa keaslian pesan, dokumen, dsb. (*cryptographic checksum*)

   → Menjaga integritas (keaslian) pesan terhadap perubahan oleh pihak lawan, misalnya akibat serangan *hacker*, virus, dsb.

- Jika MAC yang dihitung oleh penerima pesan = MAC yang melekat pada pesan, berarti pesan masih asli.
- Jika pemilik pesan menggunakan fungsi *hash* satu-arah biasa (seperti MD5 atau SHA), maka pihak lawan dapat menghitung *message digest* yang baru dari dokumen yang sudah diubah, lalu menggantinya.

  Tetapi, jika digunakan *MAC*, pihak lawan tidak dapat melakukan hal ini karena ia tidak mengetahui kunci yang asli untuk menghitung MAC.

Hacker bisa mengganti file dengan file lain, mengganti nilai MD5 semula dengan nilai MD5 yang baru. Pengunduh file tidak dapat menyadarinya.



2. Mengotentikasi pengirim pesan

- Selain memeriksa keaslian pesan, MAC dapat digunakan untuk mengotentikasi pengirim pesan, bahwa pesan memang berasal dari pengirim yang sesunguhnya (asli).

- Hal ini karena hanya pengirim dan penerima pesan yang mengetahui kunci K.

- Jika K pengirim pesan tidak sama dengan K penerima pesan, maka MAC yang dihitung oleh penerima pesan pasti tidak sama dengan MAC yang melekat pada pesan yang diterimanya.

Kapan menggunakan MAC?

- Bila hanya otentikasi pesan saja yang diperlukan, sedangkan kerahasiaan pesan tidak ditekankan.

- Namun, jika dikombinasikan dengan enkripsi pesan, maka enkripsi pesan dan perhitungan MAC menggunakan kunci yang terpisah.
- MAC dapat dihitung setelah pesan dienkripsi atau sebelum pesan dienkripsi. Umumnya lebih baik MAC dihitung sebelum pesan dienkripsi.

- Perlu dicatat bahwa MAC tidak sama dengan “digital signature” karena ia tidak menyediakan layanan nir-penyangkalan (*non-repudiation*).

- Digital signature akan dibahas pada pokok bahasan berikutnya

Algoritma MAC

- Sebagaimana fungsi hash, MAC dihasilkan dengan oleh algoritma yang melakukan kompresi pesan yang bersifat *irreversible*.

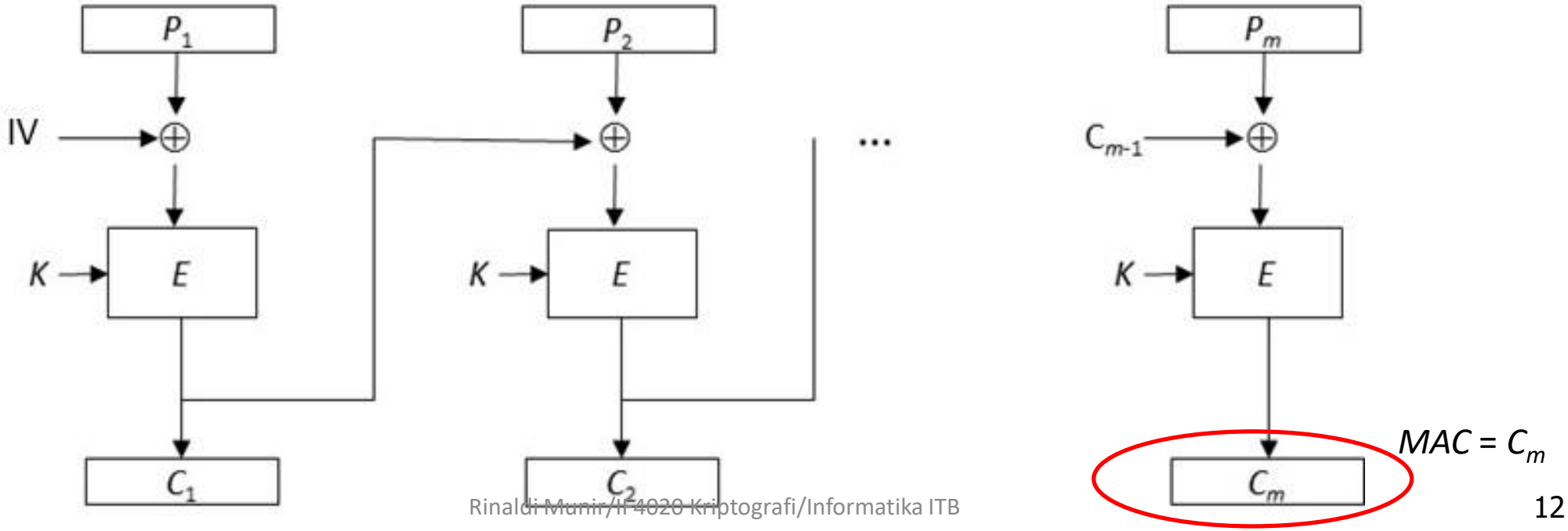
- Algoritma MAC dalah fungsi banyak-ke-satu (many-to-one function), artinya potensial banyak pesan memiliki MAC yang sama
- Namun menemukan dua atau lebih pesan yang memiliki MAC yang sama adalah sangat sulit.
- Persyaratan algoritma MAC:
  1. Diberikan sebuah pesan dan MAC, maka sangat sukar menemukan pesan lain yang memiliki MAC yang sama.
  2. MAC dari berbagai pesan seharusnya terdistribusi secara *uniform*
  3. MAC seharusnya bergantung rata pada semua bit-bit pesan

- MAC dapat dihasilkan dengan menggunakan:
    1. Algoritma MAC berbasis *block cipher*
    2. Algoritma MAC berbasis fungsi *hash* satu arah yang sudah ada

Algoritma MAC

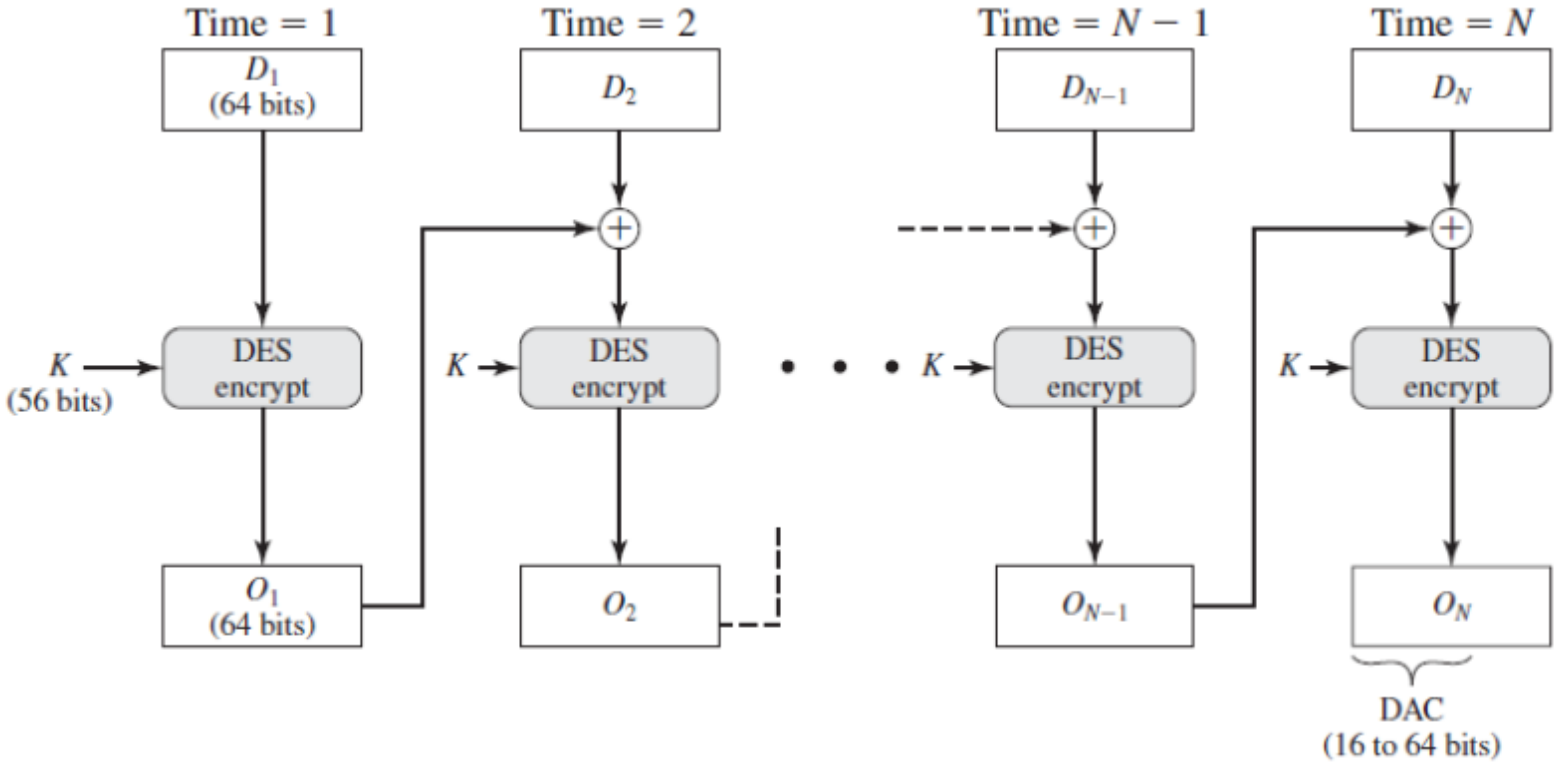
(a) Algoritma MAC berbasis *block cipher*

- *MAC* dibangkitkan dari *block cipher* dengan mode *CBC* atau *CFB*.
- Nilai *hash*-nya (yang menjadi *MAC*) adalah hasil enkripsi blok terakhir.



- Misalkan *DES* digunakan sebagai *cipher* blok, maka MAC = ukuran blok = 64 bit, dan kunci rahasia *MAC* adalah kunci DES yang panjangnya 56 bit.

- *Data Authentication Algorithm* (*DAA*) adalah algoritma *MAC* berbasis *DES-CBC* yang digunakan secara luas:



(b) Algoritma MAC berbasis fungsi *hash* satu-arah

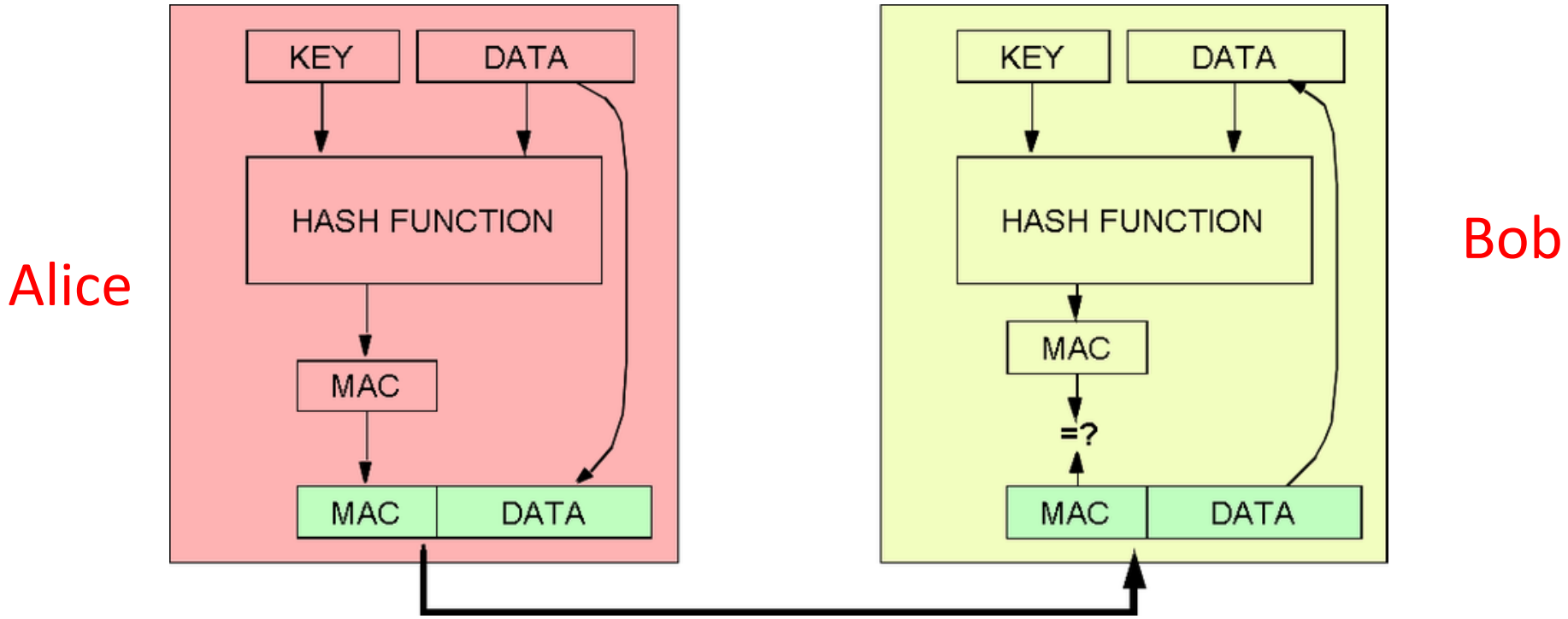
- Fungsi *hash* seperti *MD5* dan SHA dapat digunakan sebagai *MAC*
- Pesan M disambung (*concate*) dengan kunci K, lalu dihitung nilai hash dari hasil penggabungan tersebut dengan dengan fungsi hash H seperti MD5 atau SHA

$$MAC = H(M|K)$$

ket: '|' adalah simbol *concatenation*

- Panjang MAC tergantung dari fungsi hash yang digunakan. Jika menggunakan fungsi SHA-1, maka MAC yang dihasilkan adalah 160 bit

Misalkan Alice dan Bob akan saling bertukar DATA. Alice dan Bob telah berbagi sebuah kunci rahasia *KEY*.



HMAC

- HMAC adalah algoritma pembangkitan MAC menggunakan fungsi hash dan kunci.
- Dinotasikan dengan HMAC-x, x adalah nama fungsi hash yang digunakan, misalnya HMAC-SHA1, HMAC-MD5, HMAC-SHA2, dsb.
- Defenisi dari RFC 2014:

$$\mathrm{HMAC}(K, m) = \mathrm{H}\left((K' \oplus opad) \parallel \mathrm{H}\left((K' \oplus ipad) \parallel m \right) \right)$$

$$K' = \begin{cases} \mathrm{H}(K) & \text{if } K \text{ is larger than block size} \\ K & \text{otherwise} \end{cases}$$

where

H is a cryptographic hash function.

m is the message to be authenticated.

K is the secret key.

K' is a block-sized key derived from the secret key, K; either by padding to the right with 0s up to the block size, or by hashing down to less than or equal to the block size first and then padding to the right with zeros.

$\parallel$ denotes concatenation.

$\oplus$ denotes bitwise exclusive or (XOR).

$opad$ is the block-sized outer padding, consisting of repeated bytes valued 0x5c.

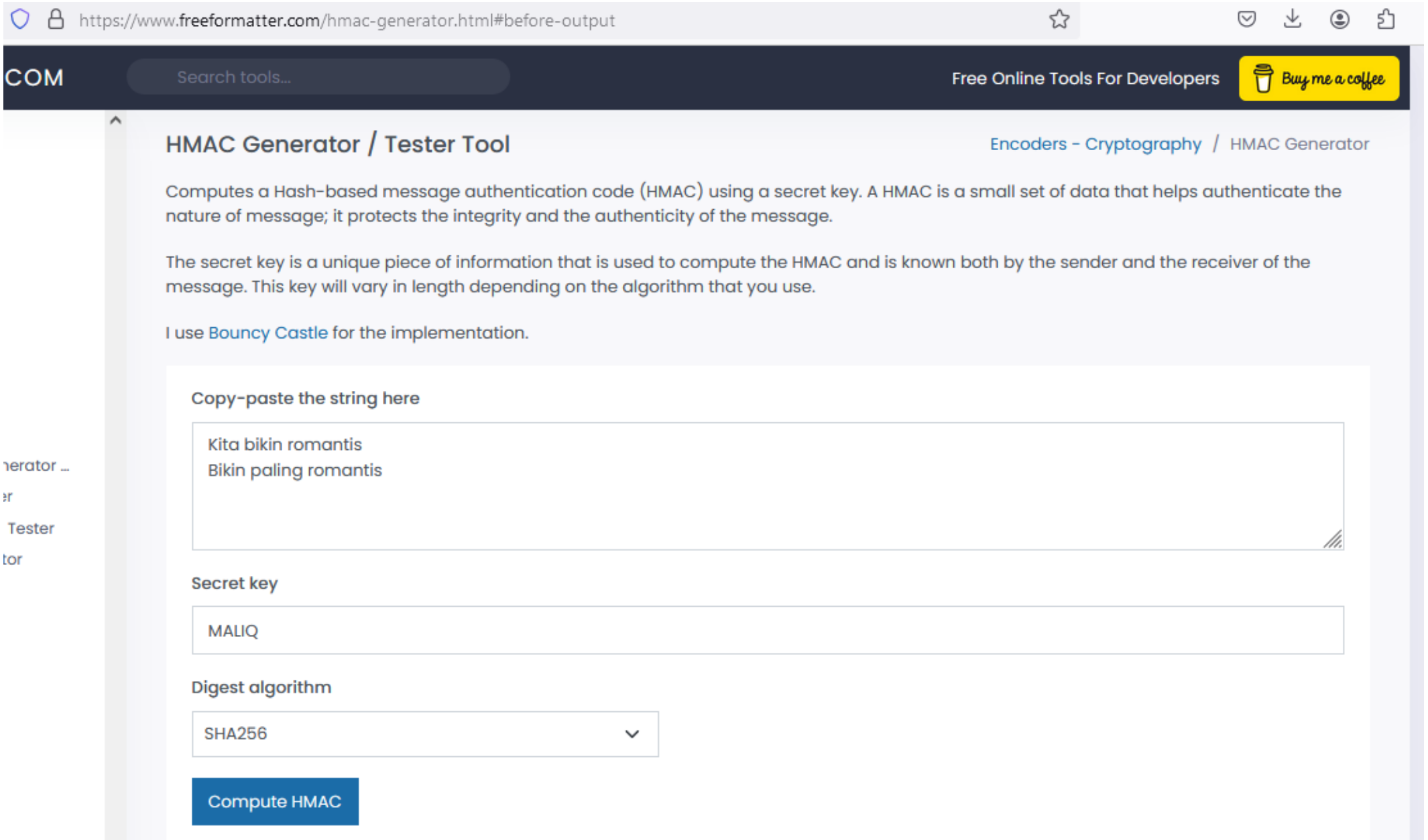
$ipad$ is the block-sized inner padding, consisting of repeated bytes valued 0x36.[3]

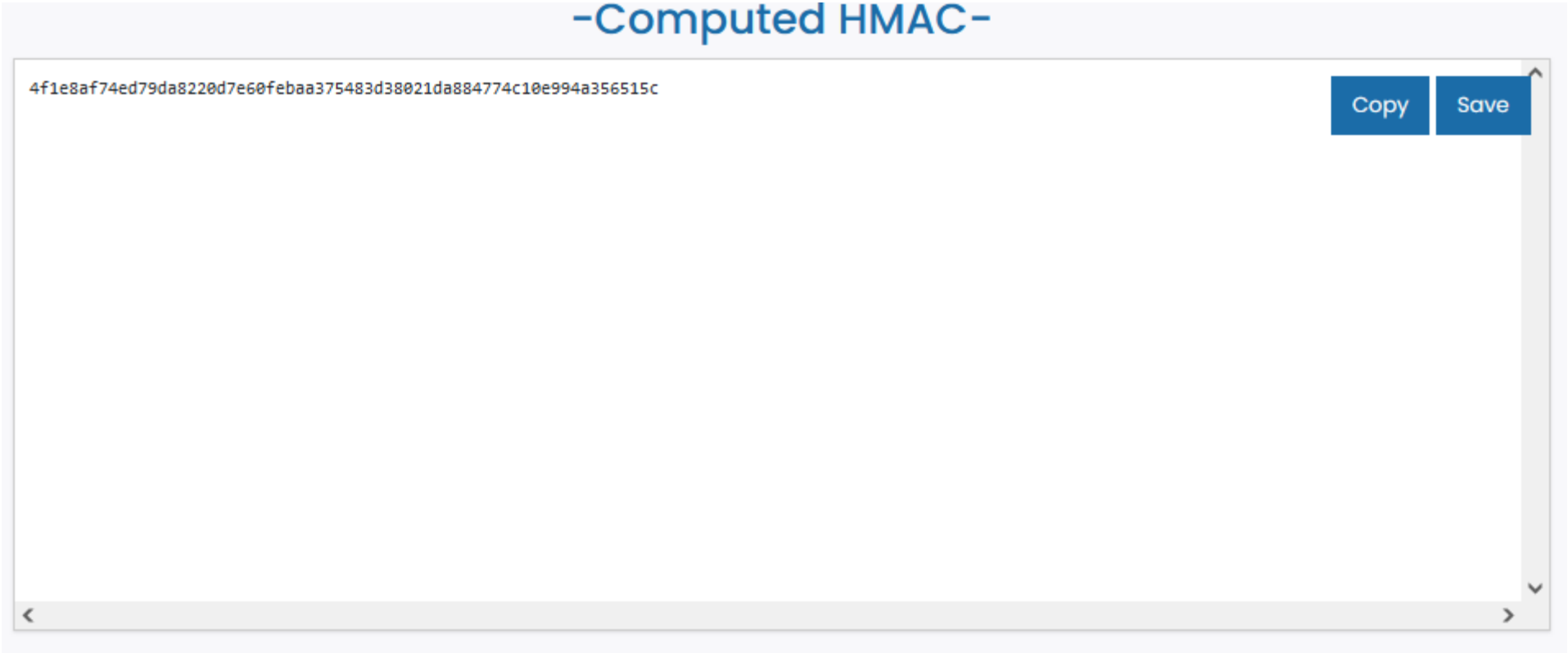
Sumber: Wikipedia

```
function hmac is
    input:
        key:        Bytes    // Array of bytes
        message:    Bytes    // Array of bytes to be hashed
        hash:       Function // The hash function to use (e.g. SHA-1)
        blockSize:  Integer  // The block size of the hash function (e.g. 64 bytes for SHA-1)
        outputSize: Integer  // The output size of the hash function (e.g. 20 bytes for SHA-1)

    // Compute the block sized key
    block_sized_key = computeBlockSizedKey(key, hash, blockSize)

    o_key_pad ← block_sized_key xor [0x5c blockSize]   // Outer padded key
    i_key_pad ← block_sized_key xor [0x36 blockSize]   // Inner padded key

    return  hash(o_key_pad ∥ hash(i_key_pad ∥ message))

function computeBlockSizedKey is
    input:
        key:        Bytes    // Array of bytes
        hash:       Function // The hash function to use (e.g. SHA-1)
        blockSize:  Integer  // The block size of the hash function (e.g. 64 bytes for SHA-1)

    // Keys longer than blockSize are shortened by hashing them
    if (length(key) > blockSize) then
        key = hash(key)

    // Keys shorter than blockSize are padded to blockSize by padding with zeros on the right
    if (length(key) < blockSize) then
        return  Pad(key, blockSize) // Pad key with zeros to make it blockSize bytes long

    return  key
```


Sumber: Wikipedia

https://www.freeformatter.com/hmac-generator.html




-Computed HMAC-
4f1e8af74ed79da8220d7e60febaa375483d38021da884774c10e994a356515c
Copy
Save

Contoh:

M = *Halo, Bob!*

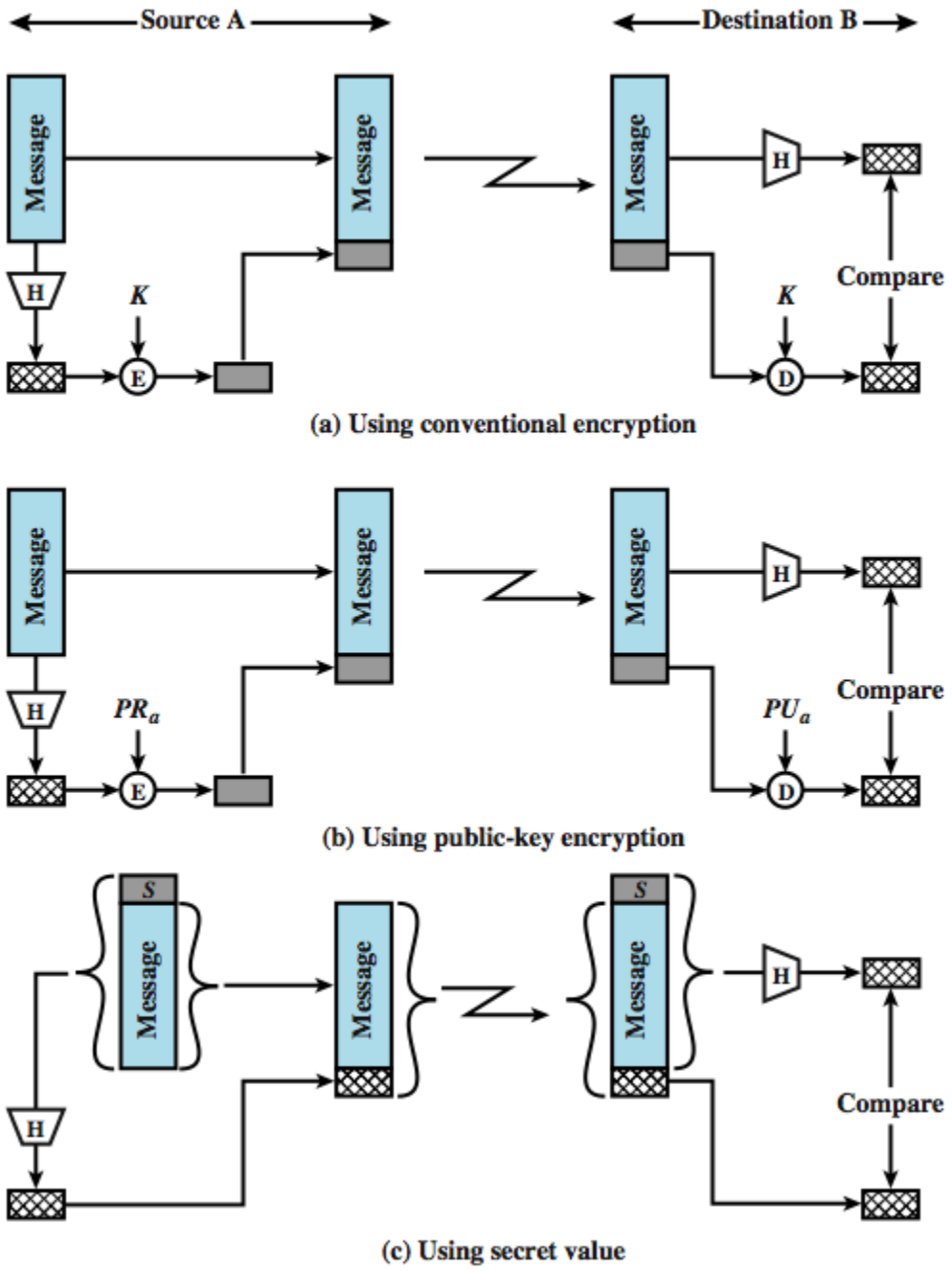
K = 12345678

Fungsi Hash: SHA-1

MAC = 6f8605c7c3a649a40abfb87b44aa21f356e931a0

Sumber: MAC online https://www.freeformatter.com/hmac-generator.html

- Selain menggunakan MAC, terdapat teknik lain untuk mengotentikasi pesan menggunakan kunci
- (a) Menggunakan fungsi hash dan enkripsi simetri
- (b) Menggunalan enkripsi dengan kriptografi kunci publik
- (c) Menggunakan nilai rahasia S yang disepakati oleh pengirim dan penerima pesan



SELAMAT BELAJAR