Implementasi MAC dalam Autentikasi Pesan dengan Nilai HMAC dari Lampiran

Filbert Wijaya - 13518077
Program Studi Teknik Informatika
Sekolah Teknik Elektro dan Informatika
Institut Teknologi Bandung, Jalan Ganesha 10 Bandung
E-mail : filbertwijaya@gmail.com

***Abstract*—Pengiriman pesan merupakan hal yang setiap hari dilakukan oleh pengguna aplikasi pengiriman pesan maupun surat elektronik (*email*). Pesan-pesan yang dikirim dapat disertakan dengan sebuah lampiran untuk memperjelas pesan yang dikirim. Untuk memastikan pesan dan lampiran yang dikirim sesuai dengan pesan dan lampiran yang diterima, pesan dan lampiran harus memiliki sebuah metode untuk memastikan keaslian isi dari pesan dan lampiran tersebut. Salah satu metode untuk memastikan keaslian isi dari pesan dan lampiran dapat menggunakan *hash-based message authentication code* (HMAC) yang merupakan salah satu bentuk dari *message authentication code* (MAC). HMAC dapat melakukan autentikasi dan memastikan keaslian data (*data integrity*) dari pesan maupun lampiran yang dikirim. Dalam makalah ini, diimplementasikan autentikasi pesan yang menyimpan nilai HMAC dari lampirannya**

***Keywords*—autentikasi, HMAC, lampiran, MAC, pesan**

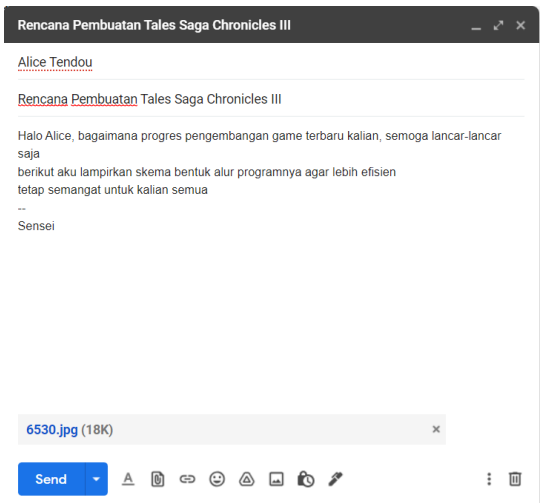
I. PENDAHULUAN

Pengiriman pesan dari satu pihak ke pihak lain memiliki kemungkinan untuk mengirimkan lampiran pendamping pesan untuk memperjelas maksud dari pesan yang ingin di kirim. Setiap pesan yang dikirim pasti sebuah teks tetapi lampiran yang dikirim dapat bervariasi seperti dokumen, gambar, file audio, maupun file video. Lampiran yang bervariasi menyebabkan proses verifikasi secara visual atau secara kasat mata tidak sepenuhnya dapat dipercaya. Verifikasi secara visual memiliki keterbatasan terutama jika lampiran yang diterima berupa stego-data yang berisi program berbahaya bagi penerimanya.

Proses komunikasi yang dilakukan harus memiliki mekanisme keamanan tertentu untuk memastikan pesan dan lampiran yang dikirim sesuai dengan pesan dan lampiran yang diterima. Salah satu bentuk mekanisme untuk menangani hal tersebut dengan menggunakan MAC (*Message Authentication Code*) dan HMAC (*Hash-based Message Authentication Code*) sebagai salah satu bentuk dari MAC. Dengan implementasi tersebut, pesan dan lampiran yang dikirim disertakan dengan sebuah kode untuk memverifikasi keaslian dari pesan dan lampiran tersebut sehingga akan menghindari adanya serangan yang dilakukan saat mengirim atau *Man in The Middle Attack*.

MAC melakukan autentikasi dari pesan berdasarkan kunci rahasia yang sebelumnya telah disepakati oleh pihak pengirim dan penerima baik dengan menggunakan pertukaran key Diffie-Hellman (*Diffie-Hellman key exchange*) atau metode lain yang disepakati oleh kedua pihak.

Dalam praktiknya, autentikasi pesan dan lampiran tidak selalu menggunakan MAC karena MAC tidak dapat memastikan siapa yang mengirim pesan tersebut seperti yang dapat dilakukan oleh tanda tangan digital (*digital signature*) dan mekanisme keamanan yang disediakan MAC gagal jika algoritma MAC dan key yang disepakati terbongkar atau diketahui pihak ketiga. Tetapi MAC masih dapat digunakan karena pihak ketiga tidak cukup mengetahui salah satu antara algoritma MAC atau key yang disepakati tetapi harus keduanya seperti pada HMAC-MD5 dimana meskipun MD5 sudah dinyatakan tidak aman tetapi HMAC-MD5 masih dapat digunakan karena nilai keynya yang dapat disesuaikan.



Gambar 1. Surat elektronik sebagai contoh pesan yang memiliki lampiran

II. DASAR TEORI

A. *Fungsi Hash Kriptografis*

Fungsi hash merupakan fungsi yang memapatkan sebuah string dari input bebas menjadi sebuah string dengan panjang tetap [3]. Sebuah fungsi hash dapat dikatakan sebagai fungsi hash kriptografis jika fungsi hash tersebut berfungsi dalam melakukan kegiatan kriptografi seperti enkripsi. Fungsi hash kriptografis berfungsi untuk menjamin aspek-aspek keamanan

seperti autentikasi, tanda tangan digital, *pseudo random number generator*, dan lain-lain. Fungsi hash kriptografis memiliki beberapa sifat berikut[4]:

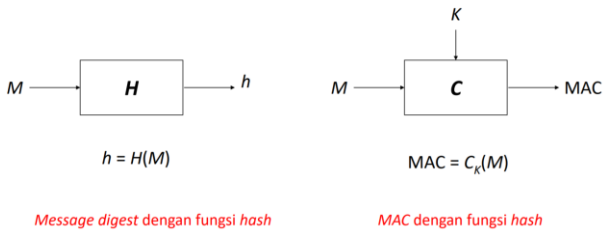
- Sulit untuk menemukan input a sehingga H(a) = y untuk sembarang output y (*preimage resistance*)
- Sangat sulit untuk menemukan dua input berbeda a dan b sehingga H(a) = H(b) (*collision resistance*)
- Sulit dalam menemukan input kedua b sedemikian sehingga H(b) = y untuk input a dan output y = H(a) (*second preimage resistance*)

Penerapan fungsi hash kriptografis memiliki beberapa keuntungan sebagai berikut[4]:

- Menjaga integritas data. Perubahan satu bit pada masukan akan menghasilkan nilai hash yang sangat berbeda sehingga menunjukan bahwa fungsi hash sangat peka terhadap perubahan input. Nilai hash yang ditambahkan pada data yang dihash dapat digunakan untuk mencocokan apakah nilai hash yang ditambahkan sama dengan hasil komputasi. Kesamaan nilai hash akan menentukan jika data mengalami perubahan atau tidak.
- Mempermudah verifikasi. Untuk mencocokan suatu data yang sangat besar atau mencocokan sebuah pesan dari *broadcaster* (pihak yang melakukan *broadcast*) tidak perlu mencocokan keseluruhan data atau pesan yang dapat memperlambat kinerja atau mengkonsumsi banyak waktu, tetapi hanya perlu mencocokan nilai hashnya saja karena nilai hash yang umumnya lebih pendek daripada data atau pesan yang dikirim.
- Normalisasi panjang data. Nilai hash memiliki Panjang yang tetap sehingga akan mempermudah proses penyimpanan. Sebagai contoh, penyimpanan password pengguna yang berbeda akan menyulitkan pada saat penyimpanan sehingga nilai hash lebih cocok untuk disimpan karena nilai hash sangat sulit untuk menemukan nilai hash dengan dua input yang berbeda dan nilai hash memiliki panjang sama.

B. *Message Authentication Code (MAC)*

MAC merupakan sebuah nilai hash dari sebuah pesan dimana fungsi hash yang dilakukan membutuhkan sebuah kunci untuk menghasilkan nilai hashnya. MAC dapat digunakan metika melakukan komunikasi melalui jaringan atau saluran yang tidak aman dimana pesan yang disampaikan dapat diperiksa keasliannya[2]



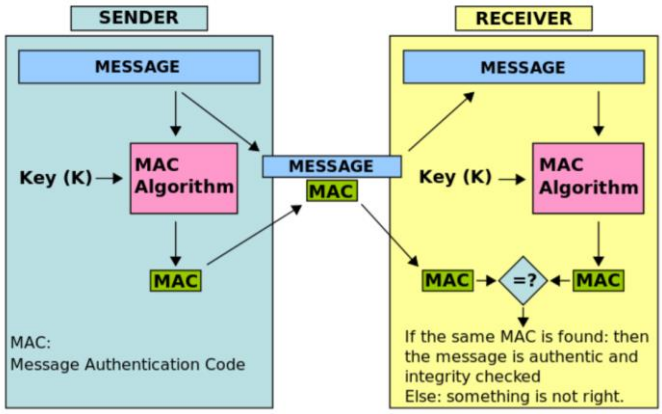
Gambar 2. Perbedaan fungsi hash dengan MAC [1]

Kebutuhan adanya sebuah kunci rahasia memungkinkan untuk mencegah sembarang pihak untuk membangkitkan nilai hash yang benar sehingga hal tersebut menjadi perbedaan utama antara MAC dengan fungsi hash biasa. Dengan demikian, MAC mengatasi kelemahan dari hash dimana sembarang pihak dapat melakukan manipulasi pesan dan penghasilkan nilai hash yang terlihat benar jika dapat mengetahui fungsi hash yang digunakan oleh pengirim dan penerima. Tetapi untuk memanipulasi pesan yang dihash menggunakan MAC, sembarang pihak harus mengetahui juga kunci yang telah disepakati oleh pengirim dan penerima pesan. Oleh karena itu MAC memiliki manfaat dalam autentikasi pesan yang diterima.

Biasanya, MAC diimplementasi dengan persamaan berikut:

$$MAC_K(m) = H(m + K)$$

Dengan H merupakan fungsi hash (seperti MD-5, SHA-1, dan lain-lain.), m adalah pesan asli dari pengirim, K adalah kunci yang disepakati oleh pengirim dan penerima, dan + adalah operasi pelekatan (*append*).



Gambar 3. Mekanisme kerja MAC [1]

C. *Hash-based Message Authentication Code (HMAC)*

MAC memiliki kelemahan terhadap length extension attack sehingga sembarang pihak hanya perlu mengetahui nilai MAC dari pesan asli dan panjang dari pesan asli tanpa perlu mengetahui isi dari pesan asli tersebut untuk menghitung MAC dari pesan yang dimanipulasi[5]. Untuk mengatasi kelemahan MAC, salah satu solusi yang meningkatkan keamanan dari MAC adalah HMAC (*Hash-based* MAC atau *Keyed-hash MAC*).

Salah satu keuntungan dari HMAC adalah kekuatan kriptografis yang kuat sehingga jika HMAC yang tidak aman merupakan akibaat dari kesalahan pada penggunaan fungsi hash, bukan pada proses pembuatan MAC [2]. Beberapa tujuan dari penggunaan HMAC sebagai berikut[2]:

- Dapat menggunakan fungsi hash apa pun yang tersedia tanpa modifikasi
- Mempertahankan kinerja dari fungsi hash tanpa degradasi yang signifikan

- Memudahkan penggantian fungsi hash dengan fungsi hash yang lebih aman saat dibutuhkan
- Mendapatkan pemahaman analisis kriptografis mengenai autentikasi berdasarkan asumsi pada fungsi hash yang digunakan

Mempermudah verifikasi. Untuk mencocokan suatu data

Dengan demikian HMAC memiliki rumus sebagai mekanisme kerja sebagai berikut:

$$HMAC(K,m) = H((K' \oplus opad) + H((K' \oplus ipad)+m))$$

K '= H(K), jika Panjang K lebih besar dari *block size* atau K'= K

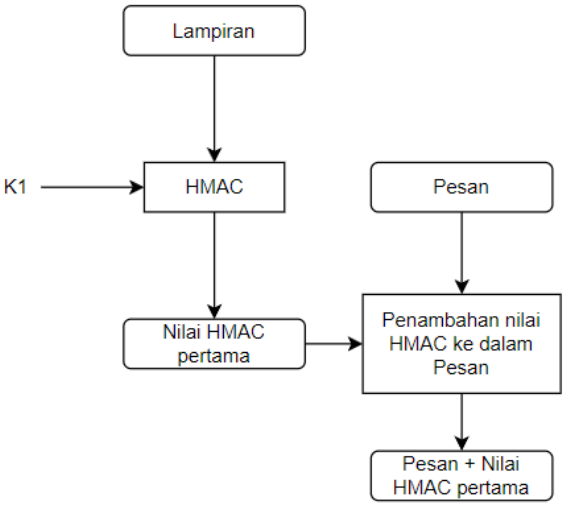
Dimana H merupakan fungsi hash, m adalah pesan asli, K adalah kunci yang disepakati, K' adalah kunci sepanjang *block size* yang diturunkan dari nilai K, + adalah operasi pelekatan (append), ⊕ merupakan fungsi XOR, opad adalah padding luar sebesar block size yang berisi nilai 0x5c yang berulang, dan ipad adalah padding dalam sebesar block size yang berisi nilai 0x36 yang berulang.

III. Implementasi

Metode yang digunakan akan memanfaatkan perbandingan dua buah nilai HMAC yang terlampir pada pesan dengan dua buah nilai HMAC yang dihasilkan dari pesan dan lampiran yang diterima.

A. Prosedur pembuatan HMAC pada lampiran

Prosedur pembuatan HMAC pada lampiran terdiri dari 2 buah proses utama, yaitu fungsi HMAC dengan key pertama (K1) yang disepakati dan penambahan nilai HMAC pada lampiran (HMAC pertama) ke dalam pesan yang akan dikirim, prosedur ini digambarkan dalam Gambar 4. yang menunjukan prosedur pembuatan HMAC pada lampiran.



Gambar 4. Skema Pembuatan nilai HMAC pada lampiran hingga penambahan nilai HMAC kedalam pesan

1). Fungsi HMAC pada lampiran (HMAC pertama)

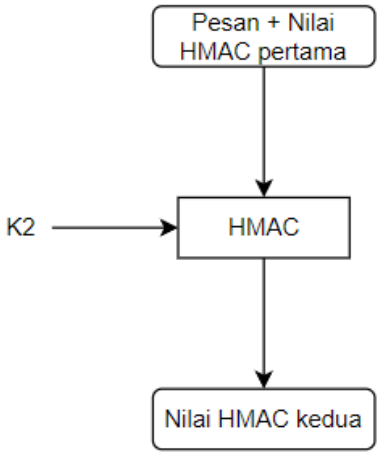
Lampiran yang akan diproses akan dibaca menurut byte demi byte sehingga lampiran dapat menerima jenis file apa pun. Pemilihan metode pembacaan didasarkan dengan semua jenis data yang tersimpan secara digital dapat dibaca dalam bentuk byte maupun biner. Kemudian key yang digunakan (K1) akan dikonversi dari bentuk string menjadi bentuk byte untuk menyesuaikan lampiran saat melakukan proses HMAC. Proses HMAC yang dilakukan berupa fungsi hash SHA-256 dan menghasilkan nilai HMAC pertama yang memiliki panjang tetap.

2). Penambahan nilai HMAC pertama kedalam pesan

Nilai HMAC pertama akan dilampirkan pada baris baru setelah baris terakhir pada pesan dengan menambahkan atribut <att>…</att> untuk menandakan nilai HMAC pertama.

B. Prosedur pembuatan HMAC pada pesan dengan nilai HMAC pada lampiran

Prosedur pembuatan HMAC pada pesan dengan nilai HMAC pada lampiran (HMAC kedua) hanya memiliki proses HMAC dengan key kedua (K2), prosedur ini digambarkan dalam Gambar 5. Yang menunjukan prosedur pembuatan HMAC pada pesan dengan nilai HMAC pada lampiran.



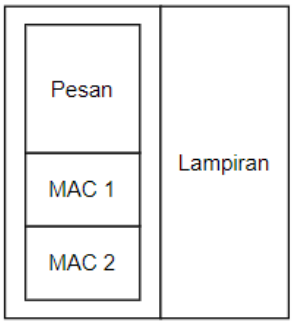
Gambar 5. Skema Pembuatan nilai HMAC pada pesan dengan nilai HMAC pada lampiran.

Pesan yang telah ditambahkan dengan nilai HMAC pertama akan melakukan proses HMAC dengan key kedua (K2) yang disepakati, jika key kedua tidak ada, key kedua akan disamakan dengan key pertama (K1). HMAC yang dilakukan berupa fungsi hash SHA-256 dan menghasilkan nilai HMAC pertama yang memiliki panjang tetap

C. Bentuk Pesan yang Dikirim dan Prosedur Pemeriksaan MAC

Pesan dan lampiran yang akan dikirim akan disertakan dengan dua buah nilai HMAC yang telah dibuat sebelumnya dan keduanya dilekatkan pada pesan sehingga pesan dan lampiran yang dikirim seperti Gambar 6 yang menunjukan bentuk pesan yang dikirim. MAC 1 merupakan nilai HMAC

pertama yang dihasilkan dari lampiran dan MAC 2 merupakan nilai HMAC kedua yang dihasilkan dari pesan yang telah dilekatkan dengan nilai HMAC pertama.

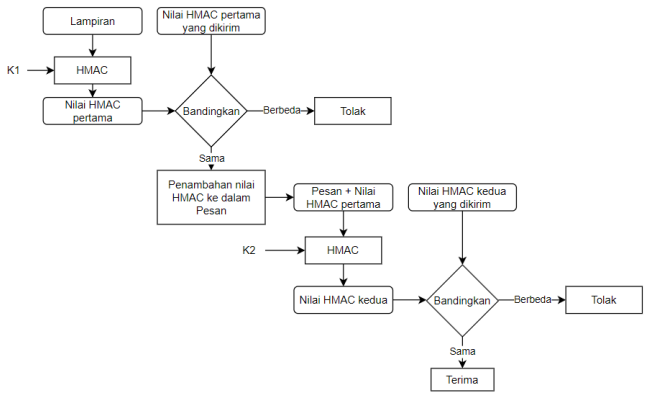


Gambar 6. Skema Bentuk Pesan dan Lampiran yang akan dikirim

Pesan dan lampiran yang diterima akan diperiksa dengan melakukan ekstraksi MAC 1 dan MAC 2 dari pesan. Setelah melakukan ekstraksi, lampiran yang disertakan akan diproses dengan cara yang sama dengan pada saat membuat MAC 1 pada saat sebelum mengirim pesan. Nilai HMAC yang dihasilkan dari proses akan dibandingkan dengan MAC 1 yang diterima, jika kedua nilai tersebut berbeda, program akan memberikan peringatan dan menyatakan bahwa lampiran yang dikirim berbeda dengan lampiran yang diterima.

Nilai HMAC yang sudah sesuai dengan MAC 1 yang diterima akan disatukan dengan pesan. Pesan yang dilekatkan dengan nilai HMAC pertama yang sesuai dengan MAC 1 yang diterima akan diproses dengan cara yang sama seperti pada saat membuat nilai HMAC kedua pada saat sebelum mengirim pesan. Nilai HMAC kedua yang dihasilkan akan dibandingkan dengan MAC 2, jika kedua nilai tersebut berbeda, program akan memberikan peringatan dan menyatakan bahwa pesan yang dikirim berbeda dengan pesan yang diterima. Pada tahap ini, lampiran yang dikirim sudah dipastikan sama dengan lampiran yang diterima berdasarkan proses sebelumnya.

Kedua proses tersebut akan berjalan seperti seharusnya jika key yang digunakan untuk memeriksa pesan dan lampiran sama seperti key yang digunakan saat membangkitkan nilai HMAC saat mengirim pesan dan lampirannya. Keseluruhan proses digambarkan pada Gambar 7.



Gambar 7. Keseluruhan proses pemeriksaan keaslian (autentikasi) pesan dan lampirannya.

IV. Hasil Pengujian dan Analisis

Bagian ini berisi pengujian terhadap ketahanan terhadap beberapa jenis manipulasi lampiran yang dicontohkan dengan file gambar dan ketahanan terhadap beberapa jenis manipulasi teks pesan dan kesimpulan berdasarkan hasil pengujian

A. Pengujian

1). Pengujian I: Uji Autentikasi Pesan dan Lampiran Asli

Pengujian dilakukan dengan lampiran berupa gambar alice.jpg seperti yang terlihat pada Gambar 8. dan teks isi pesan terlampir dalam kotak text. Autentikasi dilakukan dengan menggunakan key yang sama dengan key yang digunakan untuk membangkitkan MAC saat mengirim pesan

> Charging the lines with the force of a furious storm Fast as the lighting phantoms swarm Two hundred miles at nightfall Taken within a day Thus earning their name Earning the fame They are the panzer elite, born to compete, never retreat (ghost division) Living or dead, always ahead, fed by your dread

Gambar 8. Gambar original “alice.jpg”

Sumber: https://bluearchive.fandom.com/wiki/Tendou_Alice

TABLE I. HASIL PENGUJIAN I

Kunci 1 (K1)	Alice
Kunci 2 (K2)	Wangy
HMAC 1	b64c2a657dc33f1a4ac88b5c5ad3da6e3 5e4f0962e4abff799862622b342b847
HMAC 2	3138ff420439de553c52313e3aeeaa7a0 bef7c410a5fc3a695037f59d0cb216a
Hasil	Terautentikasi

Hasil pengujian I terlihat pada Tabel I. didapatkan bahwa metode autentikasi pesan dan lampiran yang diajukan berhasil menautentikasi pesan dan lampiran original yang tidak dimanipulasi.

2). Pengujian II: Uji Manipulasi Pesan

Pengujian dilakukan dengan lampiran berupa gambar alice.jpg seperti yang terlihat pada Gambar 8. dan teks isi pesan terlampir dalam kotak text dengan perubahan teks setelah

> Charging the lines with the force of a furious storm Fast as the lighting phantoms swarm Two hundred miles at nightfall Taken within a day Thus earning their name Earning the fame They are the panzer elite, born to complete, never retreat (ghost division) Living or dead, always ahead, fed by your dread

pembangkitan MAC ditandai dengan teks berwarna merah. Autentikasi dilakukan dengan menggunakan key yang sama dengan key yang digunakan untuk membangkitkan MAC saat mengirim pesan

TABLE II. HASIL PENGUJIAN II

Kunci 1 (K1)	Alice
Kunci 2 (K2)	Wangy
HMAC 1	b64c2a657dc33f1a4ac88b5c5ad3da6e3 5e4f0962e4abff799862622b342b847
HMAC 1 pada saat pemeriksaan	b64c2a657dc33f1a4ac88b5c5ad3da6e3 5e4f0962e4abff799862622b342b847
HMAC 2	3138ff420439de553c52313e3aeeaa7a0 bef7c410a5fc3a695037f59d0cb216a
HMAC 2 pada saat pemeriksaan	abd5232a3cf7c7b372872818847f208b2 71061f4bc93aaa634620ce2a9afbabb
Hasil	Tidak terautentikasi karena pesan telah termodifikasi

Hasil pengujian II terlihat pada Tabel II. didapatkan bahwa perubahan sebuah karakter akan mengubah keseluruhan dari nilai HMAC sehingga perubahan.

3). Pengujian III: Uji Manipulasi Lampiran

Pengujian dilakukan dengan mengubah lampiran yang semula berupa gambar alice.jpg seperti yang terlihat pada Gambar 8. menjadi sebuah stego-file yang mirip dengan Gambar 8. yaitu "alice.jpg" pada Gambar 9. dan teks isi pesan sama seperti pada pengujian I. Autentikasi dilakukan dengan menggunakan key yang sama dengan key yang digunakan untuk membangkitkan MAC saat mengirim pesan

Gambar 9. Gambar "alice.jpg" yang sebenarnya merupakan stego-file

TABLE III. HASIL PENGUJIAN III

Kunci 1 (K1)	Alice
Kunci 2 (K2)	Wangy
HMAC 1	b64c2a657dc33f1a4ac88b5c5ad3da6e3 5e4f0962e4abff799862622b342b847
HMAC 1 pada saat pemeriksaan	70041094d4786d7e037dcaa2869737a8 67bd329c1740a82bbdf3cddceeb20337
HMAC 2	3138ff420439de553c52313e3aeeaa7a0 bef7c410a5fc3a695037f59d0cb216a
HMAC 2 pada saat pemeriksaan	- (tidak dibangkitkan)
Hasil	Tidak terautentikasi karena lampiran telah termodifikasi

Hasil pengujian III terlihat pada Tabel III. didapatkan bahwa MAC dapat membuktikan adanya perubahan pada bit-bit lampiran yang menyebabkan perubahan nilai HMAC yang dibangkitkan sehingga dua buah lampiran yang tidak bisa dibedakan secara visual tetap dapat diketahui berdasarkan kedua nilai HMAC yang berbeda meskipun sudah menggunakan key yang sama. Nilai HMAC 2 tidak dibangkitkan saat pemeriksaan karena nilai HMAC 1 yang diterima berbeda dengan nilai HMAC 1 yang dibangkitkan saat pemeriksaan sehingga tidak perlu untuk membangkitkan nilai HMAC 2.

4). Pengujian IV: Uji Kunci Berbeda

Pengujian dilakukan dengan lampiran berupa gambar alice.jpg seperti yang terlihat pada Gambar 8. dan teks isi pesan sama seperti pada pengujian I. Autentikasi dilakukan dengan menggunakan key yang berbeda dengan key yang digunakan untuk membangkitkan MAC saat mengirim pesan

TABLE IV. HASIL PENGUJIAN IV

Kunci 1 (K1)	Alice
Kunci 2 (K2)	Wangy
K1 pada saat pemeriksaan	Hina
K2 pada saat pemeriksaan	Papan
HMAC 1	b64c2a657dc33f1a4ac88b5c5ad3da6e3 5e4f0962e4abff799862622b342b847
HMAC 1 pada saat pemeriksaan	407bdb6ca221d16840f551fd370c844f4 ebaec1c1e5f0fe11aac3de292910a0c
HMAC 2	3138ff420439de553c52313e3aeeaa7a0 bef7c410a5fc3a695037f59d0cb216a
HMAC 2 pada saat pemeriksaan	- (tidak dibangkitkan)
Hasil	Tidak terautentikasi karena nilai HMAC 1 yang dihasilkan berbeda

Hasil pengujian IV terlihat pada Tabel IV. didapatkan bahwa penggunaan kunci yang berbeda menghasilkan nilai HMAC yang berbeda sehingga pesan yang diterima dapat dicurigai bukan berasal pihak yang telah menyepakati kunci rahasianya.

V. KESIMPULAN

Berdasarkan hasil percobaan, terbukti bahwa metode autentikasi pesan dan lampirannya dengan *message authentication code* (MAC) dapat menjamin keaslian isi pesan mau pun lampiran yang terkirim. Secara aspek keamanan, metode autentikasi memiliki ketahanan terhadap serangan modifikasi baik pesan maupun modifikasi lampiran. Modifikasi pada pesan dan lampiran, dalam kasus ini menggunakan gambar, menyebabkan kerusakan pada integritas data sehingga gagal saat melakukan autentikasi. Tetapi, metode autentikasi dengan menggunakan MAC tidak dapat membuktikan pengirim pesan tersebut karena pengirim dan penerima memiliki kunci yang sama, tidak seperti yang diterapkan pada tanda tangan digital. Oleh karena itu, penggunaan MAC dalam kehidupan sehari-hari terbatas tetapi tetap digunakan.

VI. UCAPAN TERIMA KASIH

Dalam proses penulisan makalah dengan judul "Implementasi MAC dalam Autentikasi Pesan dengan Nilai HMAC dari Lampiran", penulis menerima banyak bantuan dari sekitar baik berupa materil maupun moral. Pertama-tama, penulis mengucapkan syukur kepada Tuhan Yang Maha Esa atas rahmat-Nya hingga makalah ini selesai dibuat dengan tepat waktu.

Penulis juga berterimakasih kepada Pak Rinaldi Munir sebagai dosen pengajar IF4020 Kriptografi yang telah memberikan pengetahuan yang menjadi dasar dari pengembangan metode autentikasi pesan dan lampirannya dalam makalah ini. Penulis berterima kasih juga terutama kepada sahabat-sahabat "*Average Ganyu Enjoyer*" yang telah memberikan dukugan dalam bentuk yang tidak bisa dijelaskan dengan kata-kata selama proses pengerjaan makalah ini dan pihak-pihak lain yang telah membantu pengerjaan makalah ini secara tidak langsung.

REFERENSI

[1] Munir, Rinaldi. "MAC (Message Authentication Code)", https://informatika.stei.itb.ac.id/~rinaldi.munir/Kriptografi/2020-2021/MAC-2020.pdf (terakhir diakses pada 00.20 WIB, 19 Desember 2021)

[2] Bellare, M, Canetti, R, Krawczyk, H. "Keying hash functions for message authentication",1996

[3] R. Sobti and G. Geetha, "Cryptographic Hash functions - a review", 2012

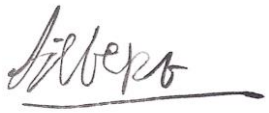
[4] Munir, Rinaldi, "Fungsi Hash", https://informatika.stei.itb.ac.id/~rinaldi.munir/Kriptografi/2020-2021/Fungsi-hash-2020.pdf (terakhir diakses pada 10.05 WIB, 18 Desember 2021)

[5] Y. Sasaki, "Cryptanalyses on a Merkle-Damg ˚ ard Based MAC — Almost Universal Forgery and Distinguishing- H Attacks."

PERNYATAAN

Dengan ini saya menyatakan bahwa makalah yang saya tulis ini adalah tulisan saya sendiri, bukan saduran, atau terjemahan dari makalah orang lain, dan bukan plagiasi.

Bandung, 19 Desember 2021



Filbert Wijaya 13518077