Rancangan Skema Pengawasan Otentikasi Akun

Leo Lambarita Nadeak - 13515041
Program Studi Teknik Informatika
Sekolah Teknik Elektro dan Informatika
Institut Teknologi Bandung, Jl. Ganesha 10 Bandung 40132, Indonesia
lnadeak97@gmail.com

***Abstract*—Zaman sekarang otentikasi tidak tertutup hanya pada satu perangkat saja. Lebih dari satu perangkat dapat masuk ke dalam sistem menggunakan satu akun yang sama. Telah banyak metode yang dilakukan untuk mengimplementasikan arsitektur sistem seperti ini, namun bagaimana dengan keamanannya? Pada makalah ini, penulis mencoba mengajukan penggunaan MAC dan Diffie-Hellman dalam meningkatkan keamanan akun yang dapat digunakan pada berbagai perangkat atau *client* yang berbeda. Yang mana keamanan yang ingin ditingkatkan terletak pada cara bagaimana sistem maupun pemilik akun dapat mengawasi aktifitas otentikasi yang terjadi pada akun.**

***Keywords*—MAC, Diffie-Hellman, Otentikasi, Device, Client**

I. PENDAHULUAN

Pada zaman sekarang seseorang dapat memiliki lebih dari satu device elektronik, seperti *smartphone,* laptop, desktop PC, dan lainnya. Dan setiap device pastinya dapat masuk ke dalam sistem dengan satu akun saja, karena dengan metode ini memberikan kemudahan kepada pengguna sistem. Metode ini dapat ditemukan pada sistem *google*. Setiap akun dapat masuk ke dalam sistem melalui berbagai aplikasi dan juga berbagai device berbeda. Ketika memasuki sistem melalui device tertentu, status Otentikasi pada device lain tidak dihapus, melainkan *google* hanya memberikan notifikasi kepada pemilik akun bahwa terdapat aktifitas Otentikasi menggunakan akun pengguna pada device lain.

Pengiriman notifikasi kepada pengguna dapat membantu pengguna mengawasi akun yang dimilikinya, mengingat terdapat kemungkinan Otentikasi akun pada device lain yang ternyata bukan milik pengguna. Dan jika hal ini memang terjadi, dapat dilakukan penanganan lebih lanjut untuk menghalangi penyerang menggunakan akun lebih lanjut. Namun masih terdapat celah yang perlu diperhatikan jika hal ini terjadi. Penanganan dapat dilakukan ketika pengguna telah membaca notifikasi dan jika memang setuju untuk melakukan penanganan tersebut. Tidak tertutup kemungkinan bahwa penyerangan sudah terjadi bahkan sebelum pengguna sadar adanya notifikasi tersebut atau setuju melakukan notifikasi.

Dalam makalah ini dibahas sebuah skema sederhana untuk mengimplementasikan alur dan juga mencoba untuk menangani celah yang dijelaskan sebelumnya dengan memanfaatkan *Message Authentication Code* (MAC), lebih tepatnya *Hash-Based Message Authentication Code (HMAC)*. Selain itu digunakan juga algorimta pembagian kunci enkripsi simetri untuk meningkatkan keamanan siste, yaitu algoritma Diffie-Hellman. Algoritma ini digunakan untuk mencegah penyerangan dari pihak ketiga yang memungkinkan penyerang untuk mengetahui kunci yang digunakan dalam algoritma MAC.

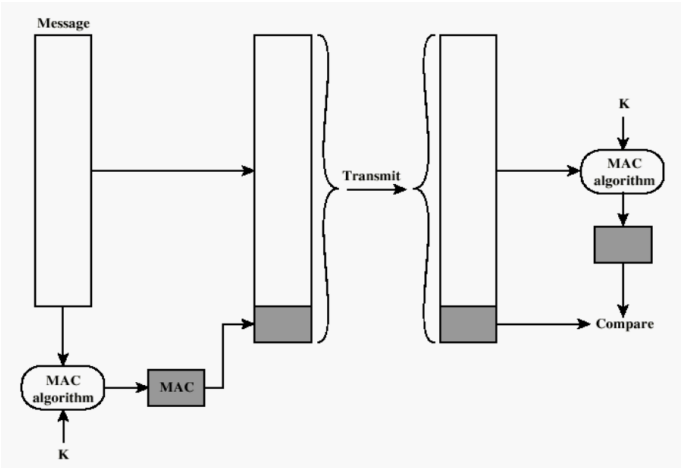
II. DASAR TEORI

A. Message Authentication Code (MAC)

MAC adalah fungsi satu arah yang menghasilkan sebuah nilai, yang disebut nilai hash, berdasarkan kunci rahasia tertentu [1]. MAC digunakan dalam komunikasi untuk mengecek otentikasi pengirim dan integritas pesan yang dipertukarkan.

Rumus sederhana dari MAC dapat dilihat pada persamaan (1) di bawah dengan h adalah nilai hash, C adalah fungsi hash atau algoritma MAC yang digunakan, dan k adalah kunci rahasia.

$$h = C(k) \quad (1)$$



Gambar 1. Alur autentikasi pesan menggunakan MAC

Implementasi MAC seperti yang dapat dilihat pada gambar 1, tidak merahasiakan pesan yang dikirimkan. Nilai hash MAC disisipkan pada pesan sebelum dikirimkan. Pada sisi penerima nilai hash MAC dihitung kembali kemudian dicek pada nilai hash MAC yang disisipkan pada pesan. Jika nilai hashnya sama, maka disimpulkan pesan yang dikirimkan valid atau terotentikasi, dan jika tidak dapat disimpulkan pesan sudah tidak

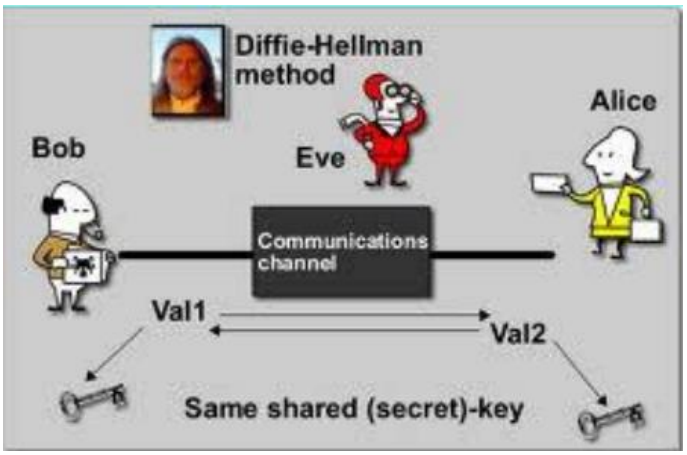
valid, tidak terotentikasi, atau terdapat perubahan oleh pihak ketiga yang tidak diharapkan.

Salah satu jenis algoritma MAC adalah *Hash-Based Message Authentication Code (HMAC)*, algoritma MAC yang menggunakan fungsi hash satu-arah seperti MD5, SHA-1, dan lainnya dalam menghasilkan nilai hash. Secara umum algoritma HMAC adalah sebagai berikut: [1]

- A dan B berbagi kunci rahasi K.
- Kemudian A dan B saling bertukar pesan M.
- A menyambung pesan M dengan K, dan kemudian A menghitung nilai MAC dari pesan yang dikirimkan dengan fungsi hash satu-arah yang diinginkan. Misal nilai MAC yang dihasilkan adalah M' dengan M' = H(M, K)
- A mengirim M dan M' kepada B.
- B melakukan hal yang sama dengan yang dilakukan A sebelum mengirim pesan. Kemudian B membandingkan hasil perhitungannya dengan pesan M' yang diterima dari A. Hasil perbandingan ini kemudian digunakan untuk menguji otentikasi A.

B. Algoritma Diffie-Hellman

Algoritma Diffie-Hellman adalah salah satu algoritma pembagian kunci cipher antar agen yang berkomunikasi. Dengan algoritma ini, kunci cipher simetri dapat dibagikan kepada seluruh entitas terkait tanpa mengirimkan kunci pada channel komunikasi seperti yang terlihat pada gambar 2 di bawah.



Gambar 2. Analogi algoritma Diffie-Hellman

Agar algoritma ini memungkinkan, setiap pihak perlu menyepakati bilangan prima n dan g dengan ketentuan g < n, yang mana kedua bilangan ini sama sekali tidak perlu rahasia [2]. Bilangan ini kemudian digunakan untuk menghitung kunci cipher yang mereka gunakan dalam berkomunikasi.

Algoritma diffie-hellman adalah sebagai berikut: [2]

1. Misal pihak yang ingin berkomunikasi adalah Alice dan Bob.
2. Alice memilih bilangan bulat acak yang besar a dan kemudia mengirimkan pesan X kepada Bob, yang mana X diperoleh dengan menggunakan persamaan (2) dibawah.

$$X = g^a \bmod n \quad (2)$$

3. Bob memilih bilangan bulat acak besar b dan mengirim pesan Y kepada Alice, yang mana Y diperoleh melalui komputasi dengan persamaan (3) dibawah.

$$Y = g^b \bmod n$$

4. Kemudian Alice memperoleh kunci K melalui perhitungan dengan persamaan (4), sedangkan Bob memperoleh kunci K melalui perhitungan dengan persamaan (5).

$$K = Y^a \bmod n \quad (4)$$

$$K = X^b \bmod n \quad (5)$$

Penyerangan berupa penyadapan pesan yang dikirimkan antara Alice dan Bob masih belum dapat menghasilkan kunci. Karena informasi yang dapat diperoleh penyadap hanyalah informasi n, g, X dan Y. Sementara penghitungan kunci masih memerlukan informasi a dan b. Nilai a dan b dapat diperoleh dengan melakukan perhitungan logaritma diskrit, yang tentunya sulit untuk dilakukan [2]. Sehingga tingkat keamanan algoritma ini berdasarkan kompleksitas logaritma diskrit yang diperlukan untuk mendapatkan nilai a dan b.

C. Otentikasi

Otentikasi atau yang lebih umum dikenal dalam bahasa inggris yaitu *Authentication* adalah suatu metode yang dilakukan untuk memvalidasi identitas seorang pengguna ketika ingin mengakses sebuah sistem. Pada kebanyakan sistem, perlu dilakukan otentikasi terlebih dahulu agar pengguna dapat menggunakan fungsionalitas lainnya.

III. RANCANGAN SISTEM

Sistem yang ingin diajukan melibatkan dua pihak utama yaitu client dan server. Client yang dimaksud merupakan aplikasi yang ada pada device/perangkat pengguna dalam mengakses atau mengirimkan *request* kepada server sistem.

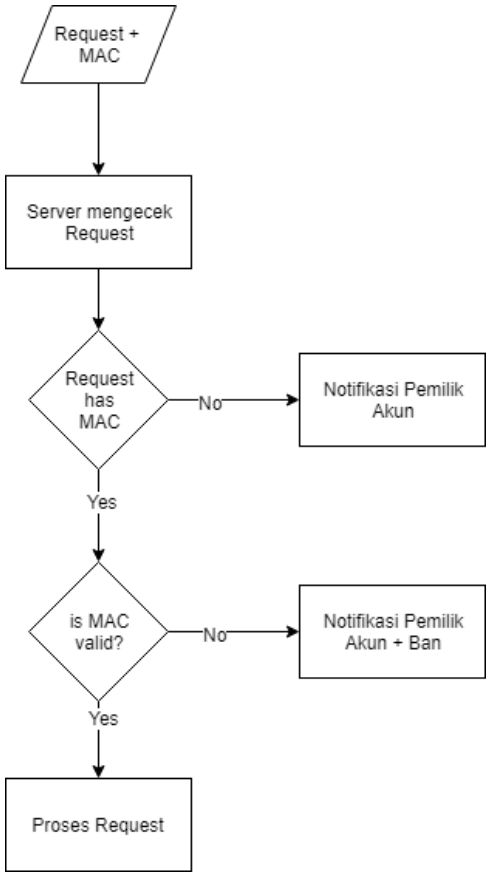
Secara sederhana setiap kali mengakses data kepada server, client harus mengirimkan request yang telah disisipi nilai MAC yang valid. Oleh karena itu client perlu menyimpan data kunci rahasia yang digunakan pada algoritma MAC. Pada sisi server, server mengecek nilai MAC dengan melakukan penghitungan nilai MAC menggunakan pesan dan kunci yang disimpan pada server berdasarakan id client yang mengirimkan request.

Jika request client tidak mengandung nilai MAC maka disimpulkan terdapat aktifitas autentikasi pada client atau perangkat baru. Ketika hal ini terjadi notifikasi dikirimkan ke pengguna yang menyatakan bahwa ditemukan aktifitas autentikasi pada perangkat baru.

Jika request client tidak mengandung nilai MAC yang valid maka disimpulkan terjadi autentikasi yang tidak diharakan. Karena jika nilai MAC tidak valid, dapat disimpulkan bahwa kunci yang digunakan untuk menghitung nilai MAC bukanlah kunci yang seharusnya. Dan jika hal ini terjadi, tanpa perlu respon dari pengguna, perangkat dapat ditangguhkan untuk sementara waktu agar tidak dapat mengakses akun pengguna. Karena *request* memang terindikasi berasal dari pengguna yang tidak diharapkan karena telah menggunakan kunci yang salah. Selain itu notifikasi perlu dikirimkan kepada pengguna yang menyatakan bahwa ditemukan aktifitas autentikasi yang

berkemungkinan berasal dari pihak ketiga yang tidak diharapkan.

Secara garis besar, skema sistem yang digunakan dapat dilihat pada gambar 3 di bawah.



Gambar 3. Skema Rancangan Sistem

Untuk mengimplementasikan rancangan sistem diatas diperlukan satu device utama sebagai tujuan pengiriman notifikasi ketika dideteksi aktifitas autentikasi pada device yang tidak dikenali. Pemilihan device ini dapat dilakukan dengan cara memilih device dimana pengguna melakukan registrasi pada sistem. Ketika registrasi sukses, kunci rahasia unik untuk berdasarkan id client dibangkitkan secara acak kemudian dikirimkan kepada client.

Cara lain selain registrasi yang dapat digunakan adalah memilih device pertama yang masuk ke dalam sistem. Device pertama yang dimaksud adalah device yang masuk ke sistem ketika server sistem tidak menyimpan data device manapun yang telah masuk ke dalam sistem.

IV. Percobaan dan Analisis

A. Analisis Implementasi

Imlementasi skema sistem yang dibahas tidaklah rumit. Secara umum implemetasi dapat dilakukan sebagaimana komunikasi client dan server diimplementasikan pada umumnya. Perbedaanya terletak pada penambahan penghitungan dan penyisipan nilai MAC setiap kali client mengirimkan request pada server. Dan pada server perlu ditambahkan *middleware* untuk melakukan pengecekan nilai MAC pada request.

Selain penambahan kedua hal di atas, secara fungsional sistem perlu juga ditambahkan fungsionalitas untuk mengirimkan notifikasi kepada client atau device utama milik pengguna. Hal ini memerlukan *cost* yang cukup besar apabila sistem pada dasarnya tidak memiliki fitur notifikasi sebelumnya. Namun fungsionalitas ini dapat ditukar dengan fungsionalitas lain yang lebih *feasible* untuk diimplementasikan jika memang diperlukan, seperti mengubah notifikasi menjadi *logging* pada server. Pengecekan berkala dapat dilakukan pada log untuk mengamati aktifitas yang tidak diharapkan. Dan tentunya skema seperti ini akan sangat memakan *resource* ketika jumlah pengguna sistem sudah sangat banyak.

Yang juga dapat menjadi permasalahan dalam implementasi adalah identitas client yang disimpan pada server untuk mengidentifikasi sistem, yang mana identitas ini juga perlu dikirimkan pada *request* dari client, sehingga server dapat tahu menggunakan kunci mana tiap – tiap *request.* Hal ini menjadi masalah mengingat device yang ada cukup beragam, dan identitas yang dimiliki dapat berbeda. Sehingga perlu dilakukan pemilihan identitas yang disimpan pada server, dapat berupa IP, identitas aplikasi (browser atau lainnya), ataupun identitas lainnya.

B. Analisis Keamanan

Analisis keamanan dapat dilihat melalui analisis sensitivitas ataupun analisis lain yang dapat digunakan untuk menganlisis keamanan dari algoritma MAC yang diberikan. Hasil analisis ini memiliki hasil sama dengan hasil analisis fungsi hash yang digunakan dalam algoritma MAC yang digunakan.

Analisis lain yang dapat dilakukan adalah analisis terhadap kemungkinan penyerangan yang mungkin terjadi pada sistem. Tidak tertutup kemungkinan terjadi penyerangan dari pihak ketiga yang dapat membaca kunci rahasia algoritma MAC. Seperti menyadap komunikasi ketika dilakukan pembagian kunci rahasis MAC kepada client. Atau pengaksesan data pada *memory* client secara illegal untuk memperoleh kunci MAC.

Perlu dirancang algoritma penyebaran kunci yang lebih mumpuni untuk mengurangi kemungkinan bocornya kunci rahasia MAC, seperti penggunaan algoritma Diffie Hellman.

Ketika menggunakan Diffie Hellman dalam membagikan kunci antara server dan client, terjadi kebutuhan lebih dimana baik client maupun server perlu menghitung kunci setiap kali melalukan penghitungan MAC. Selain itu sebelum pengiriman request, perlu dilakukan juga pertukaran informasi antara client dan server untuk bertukar nilai yang digunakan untuk menghitung kunci rahasia MAC. Sehingga performasi yang dapat diberikan sistem dapat berkurang dari yang sebelumnya. Namun penggunaan algoritma Diffie-Hellman merupakan teknik yang efektif sehingga penyerang sulit untuk menemukan kunci rahasia yang digunakan dalam algoritma MAC sistem.

Untuk mengimplementasikan penggunaan algoritma Diffie-Hellman, client tidak lagi menyimpan kunci rahasia namun yang perlu disimpan adalah nilai bilangan prima n dan g yang digunakan nantinya untuk menghitung kunci rahasia. Dapat dilihat bahwa keamanan dari sistem meningkat yang mana penyerang tidak dapat menyadap komunikasi client dan server ketika pengiriman kunci dilakukan, atau menyerang dengan mengakses secara illegal data pada client untuk

memperoleh kunci rahasia. Mengingat kunci sudah tidak disimpan lagi ataupun dikirimkan pada *channel* komunikasi, melainkan melalui perhitungan menggunakan bilangan acak.

Untuk mengingkatkan keamanan sistem, bilangan acak yang digunakan dalam menghitung kunci digunakan hanya sekali pemrosesan *request* saja. Ketika perlu memroses *request* lain yang baru, bilangan acak perlu dibangkitkan kembali. Dengan kata lain bilangan acak disimpan oleh masing – masing server ataupun client hany untuk sementara waktu saja, hingga server berhasil mengecek validitas *request* yang diterimanya.

V. KESIMPULAN DAN SARAN

A. Kesimpulan

MAC untuk mengawasi aktifitas otentikasi sebuah akun pada sistem tertentu merupakan pilihan yang cukup baik. Selain implementasi yang cukup sederhana, integritas pesan dapat terjaga meskipun kerahasiaan pesan tidak disembunyikan. Menimbang hal ini, penggunaan MAC akan sangat berguna pada *request* client yang bertujuan untuk melakukan pengubahan pada data yang disimpan dalam sistem seperti *update*, *store*, ataupun *delete* pada *database.* Dengan mengecek integritas pesan, *request* dapat dilihat apakah berasal dari client yang terotentikasi tanpa harus membatasi device yang dapat mengakses sistem.

Dengan skema sistem yang dijelaskan sebelumnya, keamanan akun dapat ditingkatkan. Pemberitahuan kepada pengguna ketika sistem mendeteksi perilaku yang tidak seharusnya pada akun pengguna dapat mencegah penyerang untuk mengakses akun pengguna tanpa sepengetahuan pemilik akun.

B. Saran

Skema sistem yang diajukan masih memiliki banyak hal yang dapat dikembangkan. Baik dari sisi arsitektur, keamanan, maupun performansi yang diberikan. Sehingga pengembangan lebih lanjut dari sistem yang diajukan masih dapat dilakukan untuk mencapai sistem yang efisien dan aman bagi pengguna.

VI. UCAPAN TERIMA KASIH

Penulis ingin mengucapkan rasa syukur atas rahmat dan karunia Tuhan Yang Maha Esa sehingga makalah ini bisa terselesaikan. Penulis juga ingin mengucapkan terima kasih kepada Bapak Dr. Ir. Rinaldi Munir, M.T. selaku dosen pengampu IF4020 Kriptografi yang telah membimbing penulis dalam mempelajari kriptografi sehingga dapat menulis makalah ini. Dan juga penulis ingin mengucapkan terima kasih kepada keluarga dan teman yang telah memberikan dorongan selama ini.

DAFTAR PUSTAKA

[1] Munir, Rinaldi. 2019. Slide Kuliah IF4020 Kriptografi: MAC (Message Authentication Code).

[2] Munir, Rinaldi. 2019. Slide Kuliah IF4020 Kriptografi: Algoritma Pertukaran Kunci Diffie-Hellman.

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, 10 Mei 2019

Leo Lambarita Nadeak
13515041