Evaluasi Penggunaan Enkripsi Galois/Counter Mode (GCM) untuk Pengamanan Data dalam Cloud Storage

Nadira Rahmananda Arifandi - 18221059

Program Studi Sistem dan Teknologi Informasi

Sekolah Teknik Elektro dan Informatika

Institut Teknologi Bandung, Jalan Ganesha 10 Bandung

E-mail (gmail): nadiraaraa@gmail.com

Abstract*— Di tengah perkembangan era digital, cloud storage telah menjadi solusi populer untuk penyimpanan data karena fleksibilitas dan aksesibilitasnya yang tinggi. Namun, penggunaan cloud storage memicu banyak jenis risiko keamanan data bagi para penggunanya. Salah satu metode yang paling populer untuk melindungi data dalam cloud storage adalah menggunakan algoritma enkripsi seperti Galois/Counter Mode (GCM). GCM adalah algoritma enkripsi blok simetris yang menawarkan kombinasi enkripsi dan autentikasi dalam satu proses, memastikan bahwa hanya pengguna yang berwenang yang dapat mengakses dan mengubah data. Penelitian ini mengevaluasi efektivitas GCM dalam mengamankan data pada cloud storage dengan menganalisis proses. keunggulan, tantangan, dan risiko yang dihadapi. Keunggulan GCM meliputi kecepatan, kemampuan paralelisme, fleksibilitas, dan kebebasan penggunaan tanpa hak paten. Di sisi lain, keterbatasan yang dimiliki GCM meliputi risiko kolisi kunci, batasan ukuran pesan, kebutuhan hardware yang tinggi, dan risiko tinggi atas serangan pemalsuan. Untuk meningkatkan keamanan dan efisiensi penggunaan GCM pada cloud storage, terutama dalam mengenkripsi data yang terlalu besar, disarankan implementasi modifikasi dalam bentuk pemecahan pesan dan kunci autorisasi menjadi beberapa bagian dan melakukan enkripsi secara terpisah untuk menghindari pengulangan IV (intialization vector). Implementasi saran ini diharapkan dapat semakin mengoptimalkan penggunaan GCM dalam layanan cloud storage di masa depan.

***Keywords*— Cloud Storage, Enkripsi, GCM, Keamanan Data**

I. PENDAHULUAN

Di tengah perkembangan era digital, ketergantungan manusia terhadap pengaksesan dan penyimpanan secara digital meningkat pesat. Kebutuhan penyimpanan seperti dokumen, foto, dan berbagai aplikasi untuk membantu keseharian manusia membutuhkan tempat penyimpanan yang masif. Untuk menangani kebutuhan penyimpanan, cloud storage muncul sebagai solusi populer. Cloud storage merupakan alternatif penyimpanan data secara eksternal melalui internet yang menawarkan fleksibilitas dan aksesibilitas yang tinggi. Namun, data yang disimpan dalam cloud ini bersifat rentan terhadap berbagai bentuk serangan siber, baik ketika penyimpanan maupun ketika transmisi, sehingga penggunaan cloud storage membawa ancaman yang signifikan terhadap keamanan data para penggunanya.

Beberapa tahun terakhir ini, terjadi banyak kejadian kebocoran data di berbagai platform besar seperti Google. Kebocoran data Google pada tahun 2023 telah membocorkan data pribadi sensitif lebih dari 50 juta penggunanya yang dapat mengakibatkan kerugian privasi dan finansial bagi pengguna-pengguna ini[1]. Hal ini mengilustrasikan pentingnya mengevaluasi pengamanan data dari tempat kita menyimpan data.

Salah satu metode yang selama ini digunakan untuk melindungi data dalam cloud storage ataupun tempat penyimpanan data lain adalah dengan melakukan enkripsi data. Teknik enkripsi mengubah data yang disimpan berdasarkan algoritma yang digunakan, sehingga data dapat dijaga kerahasiaannya dari pihak yang tidak memiliki akses untuk memecahkan enkripsi tersebut. Salah satu teknik enkripsi bernama Galois/Counter Mode(GCM) merupakan salah satu enkripsi Di antara berbagai teknik enkripsi yang tersedia, Galois/Counter Mode (GCM) menonjol sebagai salah satu algoritma enkripsi terautentikasi yang telah direkomendasikan dalam NIST SP 800-38D[2]. Algoritma GCM merupakan enkripsi blok simetris yang menggabungkan enkripsi dan autentikasi dalam satu proses, memberikan tingkat keamanan yang relatif tinggi. Algoritma ini digunakan dalam berbagai cloud storage ternama, seperti AWS, Google Cloud[3], dan Microsoft Azure.

Makalah ini bertujuan untuk mengevaluasi efektivitas dari algoritma enkripsi Galois/Counter Mode (GCM) dalam konteks pengamanan data pada cloud storage. Evaluasi ini akan mencakup analisis keunggulan dan tantangan GCM

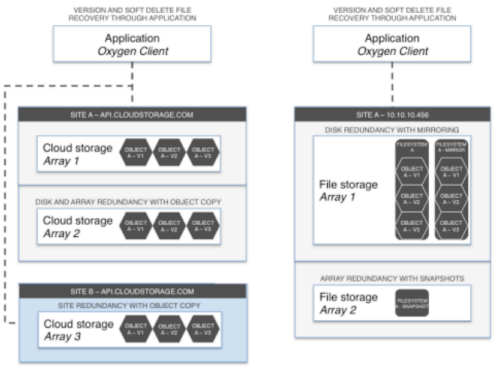
dalam melindungi data terhadap berbagai ancaman keamanan, serta kinerjanya memproses dan menyimpan data. Makalah ini diharapkan dapat memberikan wawasan yang mendalam mengenai berbagai aspek algoritma enkripsi GCM sebagai enkripsi dalam layanan-layanan cloud storage, serta rekomendasi untuk implementasi yang optimal di masa depan.

II. TINJAUAN PUSTAKA

Berikut merupakan penjabaran dari teori-teori yang relevan terhadap penulisan makalah ini.

A. Cloud Storage

Salah satu perkembangan krusial dalam bidang komputer adalah cloud computing, yaitu tipe pemrosesan data dalam bentuk software ataupun resource yang disediakan melalui internet[4]. Salah satu layanan yang ditawarkan cloud computing ini adalah cloud storage; layanan penyimpanan data digital yang memungkinkan pengguna untuk menyimpan, mengelola, dan mengakses data melalui internet. Dengan adanya cloud storage, pengguna dapat mengurangi kebutuhan penyimpanan dalam device mereka, karena banyak dari data yang mereka butuhkan dapat dialihkan ke penyimpanan cloud yang mungkin fisiknya dapat berada di kota ataupun negara berbeda. Layanan ini memberikan fleksibilitas dalam penyimpanan data yang kebutuhannya terus meningkat itu.



Gambar 1. Ilustrasi Penyimpanan Cloud dibandingkan Penyimpanan di Tempat[5]

Gambar di atas mengilustrasikan penyimpanan data pada cloud serta pada hardware device di tempat. Seperti yang dapat dilihat di atas, penyimpanan data yang dilakukan pada cloud disimpan pada suatu tempat yang tidak kita ketahui, dan data ini akan diduplikasi dalam server atau lokasi yang berbeda untuk menjaga keamanan dalam kasus terjadinya kegagalan dalam salah satu server. Kemudian, ketika data diakses, akan dilakukan duplikasi paruh data ynag dibutuhkan menggunakan layanan seperti API agar data dapat dilihat dan dimanipulasi pengguna. Di sisi lain, penyimpanan di tempat menyimpan semua data dalam satu tempat yang sama, dengan teknik mirroring untuk menduplikasi data dalam rangka memastikan data tetap dapat diakses ketika terjadi kerusakan pada satu bagian penyimpanan. Kemudian, data akan diambil dalam bentuk snapshot dalam suatu bagian file storage terpisah ketika sedang diakses untuk mendukung pengaksesan dan manipulasi data secara cepat.

Cloud storage menawarkan berbagai keunggulan pada penggunanya seperti yang dijabarkan di bawah ini:

- Skalabilitas yang tinggi, dalam arti pengguna cloud storage dapat dengan mudah menambah dan mengurangi kapasitas penyimpanan sesuai kebutuhannya tanpa perlu melakukan perubahan atau penambahan fisik terhadap hardware yang mereka miliki.
- Fleksibilitas pengaksesan data, dalam arti pengguna cloud storage dapat mengakses data menggunakan device yang berbeda, selama mereka memiliki akses terhadap cloud storage mereka. Hal ini berarti data dapat diakses dalam dua device pada saat yang bersamaan. Sebagai tambahan, fleksibilitas ini juga menjaga keberadaan data jika terjadi kerusakan terhadap hardware device milik pengguna.
- Back-up data yang efisien. Cloud storage memiliki sistem penyimpanan data back-up yang sangat efisien dibandingkan penyimpanan tradisional karena penyedia layanan akan menawarkan redundansi data untuk memastikan data tetap aman dan lengkap meskipun terjadi kegagalan pada salah satu perangkat keras penyimpan data.

Di sisi lain, cloud storage memiliki tantangan dan risiko keamanan sebagai berikut[6]:

- Risiko kebocoran data. Data yang disimpan dalam cloud disimpan dalam tempat penyimpanan massal bersama dengan data-data milik pengguna lain. Keamanan ini dapat terganggu jika terdapat pengaksesan terhadap data oleh pihak yang seharusnya tidak memiliki akses baik oleh pegawai cloud computing atau oleh hacker dalam proses penyerangan siber. Oleh karena itu, dibutuhkan regulasi dan sistem penyimpanan yang kuat untuk mencegah kebocoran data.
- Peretasan dalam transmisi. Proses pengiriman data ketika mengakses cloud storage menuju device pengguna juga menghadapi berbagai risiko pengaksesan atau bahkan perubahan dalam data yang sedang ditransmisikan.
- Lokalitas data. Penyimpanan data dalam lokasi yang tidak diketahui oleh pengguna dapat menimbulkan permasalahan akibat adanya perbedaan regulasi data dan privasi yang dimiliki negara-negara berbeda.
- Intrusi data. Karena data yang disimpan dalam cloud storage disimpan bersama data pengguna-pengguna lian, terdapat risiko terjadinya intrusi data, dimana suatu pengguna dapat mengakses data pengguna lain dalam satu cloud storage melalui loop hole dalam keamanan dan regulasi data jika tidak terdapat pembatas data antar-pengguna yang kuat.
- Keterbatasan dalam availability. Sebuah cloud storage perlu semaksimal mungkin meyediakan layanan di setiap waktu terhadap penggunanya. Namun, penyedia layanan

juga harus menanggapi berbagai kejanggalan atau kegagalan yang dapat membahayakan pengguna dengan serius, sehingga maintenance atau access blocking mungkin diperlukan dalam waktu ke waktu, sehingga pengguna tidak dapat mengakses layanan pada waktu-waktu tertentu tersebut.

B. Keamanan Data

Keamanan data merupakan tingkat kualitas penyimpanan dan manajemen data untuk menghindarkan data dari hal-hal yang tidak diinginkan. Untuk mencapai keamanan data yang baik, segala proses data harus memenuhi 5 unsur berupa confidentiality, integrity, availability, authentication, dan non-repudiation[7] :

- Confidentiality atau kerahasiaan berarti data tidak boleh bisa dimengerti oleh pihak yang tidak berkepentingan.
- Integrity atau keaslian berarti data tidak boleh berubah tanpa pengetahuan dan persetujuan pemilik data.
- Availability atau ketersediaan berarti data harus dapat diakses ketika dibutuhkan.
- Authentication atau pengenalan berarti pengirim data harus dapat dipastikan atau divalidasi.
- Non-repudiation atau kenirsangkalan berarti data yang telah dikirim tidak dapat disangkal oleh pengirim data tersebut.

Kebocoran data (data breach) adalah insiden signifikan yang menyerang keamanan data di mana data pengguna yang seharusnya bersifat privat berhasil diakses oleh seseorang yang tidak berwenang. Dalam beberapa tahun ke belakang ini, kebocoran data sangat marak terjadi akibat berbagai faktor-faktor seperti serangan siber, kelalaian manusia, atau kegagalan sistem[3]. Tidak hanya pada perusahaan-perusahaan cloud storage berskala kecil, kebocoran data ini juga banyak terjadi pada perusahaan besar yang dahulu diasumsikan bersifat aman seperti Google dan Amazon.

C. Enkripsi Kriptografi

Enkripsi adalah metode mengontrol keamanan yang digunakan untuk menjaga kerahasiaan data agar tidak dapat dipahami oleh pihak yang tidak berkepentingan. Proses enkripsi merupakan proses transformasi matematis dari data untuk menghasilkan kumpulan data berbentuk lain yang tidak menggambarkan data asalnya, disebut ciphertext. Data ini kemudian akan didekripsi ketika ingin diakses agar kembali dapat dimengerti, menghasilkan plaintext. Proses enkripsi ini bersifat reversible, sehingga data apa pun yang telah dienkripsi pasti dapat dikembalikan menjadi data asalnya menggunakan suatu kunci.[8]

Terdapat dua jenis kunci enkripsi, yaitu symmetric-key encryption dan asymmetric-key encryption. Dalam symmetric-key encryption, kunci yang digunakan untuk mengenkripsi dan mendekripsi merupakan kunci yang sama. Sedangkan dalam asymmetric-key encryption, kunci yang digunakan untuk mengenkripsi dan mendekripsi berbeda untuk meningkatkan keamanan lebih lanjut. Kunci yang berbeda ini biasa disebut public key dan private key.

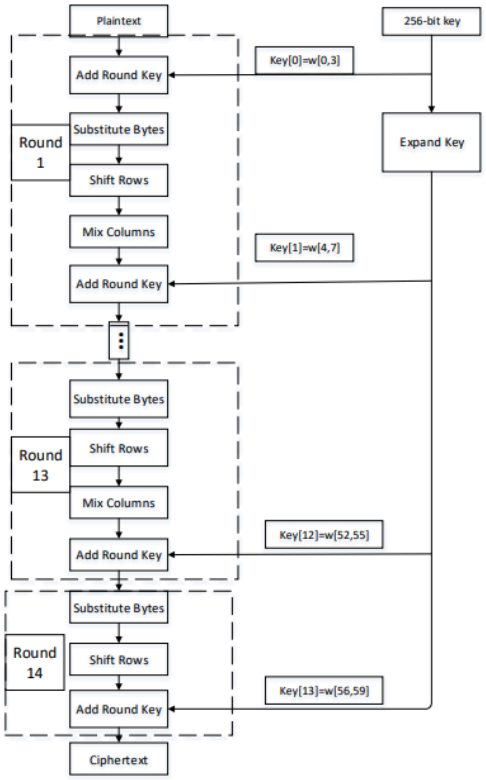
Implementasi kriptografi untuk mengamankan data kini digunakan dalam hampir tiap aplikasi. Berbagai algoritma kriptografi telah dikembangkan dan diimplementasi untuk kebutuhan yang berbeda-beda. Untuk mengatur tiap algoritma ini, dibuat standarisasi keamanan yang akan menilai keamanan algoritma untuk mengamankan suatu data.

D. Galois/Counter Mode

Enkripsi Galois/Counter Mode (GCM) adalah mode enkripsi yang menggabungkan autentikasi (validasi pengguna yang mengakses data) dan enkripsi dalam satu operasi. GCM merupakan algoritma yang banyak digunakan karena efisiensinya ketika digunakan dalam pengiriman data dan kebutuhan resource implementasinya yang relatif sedikit. Algoritma ini dirancang untuk digunakan bersama algoritma enkripsi blok simetris seperti AES.

Algoritma GCM terdiri dari dua operasi utama, yaitu Counter Mode (CTR) dan Galois Field (GF). Operasi CTR dilakukan terlebih dahulu untuk mengenkripsi data menjadi ciphertext menggunakan algoritma enkripsi blok simetris. Kemudian, dilakukan GF, yaitu operasi autentikasi dengan cara menghasilkan tag autentikasi menggunakan kombinasi data, kunci autentikasi, dan nilai nonce (number used once).

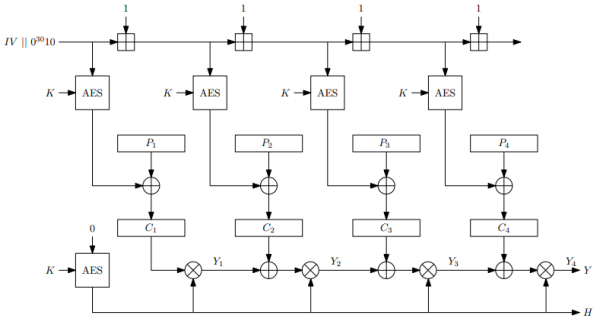
Kini, algoritma GCM merupakan tipe enkripsi blok yang paling banyak digunakan di dunia. Penggunaan algoritma GCM merupakan bagian dari TLS 1.2(2008). Algoritma ini juga banyak digunakan bersama RSA membentuk enkripsi RSA GCM untuk mengenkripsi data dalam cloud storage seperti pada layanan AWS, Google Cloud, dan Microsoft Azure, baik dalam penyimpanan statisnya (encryption at rest), maupun ketika transmisi data (end-to-end encryption). Penggunaan algoritma ini diatur dalam standar NIST SP800-38D[9].



Gambar 4. Ilustrasi Enkripsi GCM[10]

Berikut merupakan tahapan enkripsi GCM sesuai ilustrasi di atas:

- Inisiasi kunci
- Inisiasi IV (Initialization Vector)
- Inisiasi AAD (Addiitonal Autheticated Data)
- Membangkitkan blok counter
- Enkripsi tiap blok plaintext menggunakan algoritma enkripsi blok seperti AES dalam mode counter.
- Enkripsi counter dengan kunci simetris untuk menghasilkan enkripsi kunci simetris.
- Operasi XOR plaintext dengan keystream untuk menghasilkan cyphertext.
- Melakukan operasi multiplikasi GF untuk menghasilkan tag autentikasi.
- Menghasilkan hash key dengan cara mengenkripsi nilai nol dengan kunci simetris
- Melakukan hashing AAD dan ciphertext untuk menghasilkan tag autentikasi.



Gambar 5. Ilustrasi Enkripsi GCM[11]

Berikut merupakan sebuah gambaran algoritma untuk melakukan enkripsi dan dekripsi GCM yang telah dijabarkan di atas:

```
def encryptGCM (plaintext, key, IV, AAD):
    H = Encrypt_Block(0, key)  # Hash key
    Y0 = Initialize_Counter(IV)
    ciphertext = []
    counter = Y0

    # Start encryption
    for block in plaintext:
        keystream = Encrypt_Block(counter, key)
        ciphertext_block = XOR(block, keystream)
        ciphertext.append(ciphertext_block)
        counter = Increment_Counter(counter)

    # Start authentication process
    auth_tag = Galois_Field_Multiply(H, AAD, ciphertext)

    return (ciphertext, auth_tag)

def decryptGCM(ciphertext, key, IV, AAD, auth_tag):
    H = Encrypt_Block(0, key)
    Y0 = Initialize_Counter(IV)
    plaintext = []
    counter = Y0

    # Start decryption
    for block in ciphertext:
        keystream = Encrypt_Block(counter, key)
        plaintext_block = XOR(block, keystream)
        plaintext.append(plaintext_block)
        counter = Increment_Counter(counter)

    # Start authentication process
    expected_auth_tag = Galois_Field_Multiply(H, AAD, ciphertext)
    if auth_tag != expected_auth_tag:
        raise Exception("Authentication failed")

    return plaintext
```

III. ANALISIS DAN PEMBAHASAN

Sejak sekitar tahun 2010, algoritma GCM, terutama algoritma AES-GCM telah marak digunakan untuk melakukan enkripsi data dalam cloud storage, jaringan komunikasi, dan keamanan perangkat keras. Selama implementasinya, algoritma ini telah mengalami banyak perubahan dan peningkatan kualitas untuk terus memastikan keamanan data

yang maksimal. Walaupun algoritma ini tentunya tetap memiliki kelemahannya, AES-GCM tetap digunakan oleh banyak penyedia layanan cloud storage ternama karena alasan-alasan tertentu.

A. Keunggulan Penggunaan GCM

Berikut merupakan beberapa keunggulan dari penggunaan GCM:

- Kecepatan. GCM dirancang untuk kinerja yang tinggi, menjadikannya sangat efisien untuk enkripsi dan dekripsi data dengan throughput tinggi dengan cara menggunakan block cipher dengan perhitungan matematis yang menghasilkan angka relatif kecil. Hal ini sangat dibutuhkan untuk penyimpanan dalam cloud karena banyaknya jumlah pengguna serta besarnya data yang perlu disimpan dan diakses sehari-hari. Kecepatan AES-GCM telah terbukti secara berulang kali bahwa enkripsi dan dekripsi data dapat dilakukan secara lebih cepat dibandingkan algoritma lainnya seperti CBC, CCM, CWC, EAX, dan OCB[12].

- Paralelisme. Operasi enkripsi dan autentikasi di GCM dapat dilakukan secara paralel, meningkatkan kecepatan pemrosesan pada perangkat keras yang mendukung, dalam arti operasi generasi dan enkripsi dapat dilakukan sebelum operasi pada blok sebelumnya selesai. Hal ini sangat diperlukan untuk mendukung kebutuhan transformasi yang besar akibat panjangnya key yang digunakan[10].

- Kombinasi Enkripsi dan Autentikasi. Fitur ini bersifat krusial dalam penggunaan cloud storage, karena data harus hanya dapat diakses oleh pengguna yang terautentikasi. Penggunaan satu algoritma untuk enkripsi dan autentikasi meningkatkan keamanan data karena tidak perlu adanya penyimpanan kunci enkripsi tambahan setelah melakukan autentikasi atau sebaliknya.

- Fleksibilitas. Algoritma AES-GCM dapat menggunakan IV dengan panjang yang beragam, yang kemudian dapat di-hash untuk mendapatkan panjang IV yang sesuai, sehingga implementasinya dapat dengan mudah disesuaikan dengan kebutuhan sistem.

- Kebebasan penggunaan. Salah satu alasan lain algoritma GCM sangat populer untuk digunakan adalah karena algoritma ini dapat digunakan secara bebas. Algoritma ini telah diakui secara internasional dan tidak memiliki hak paten, sehingga siapa pun dapat menggunakannya [12].

B. Kelemahan dan Risiko Penggunaan GCM

Berikut merupakan beberapa keunggulan dari penggunaan GCM:

- Weak Key Collision. Weak key collision adalah kejadian ketika nonce yang digunakan untuk mengenkripsikan data digunakan secara berulang, sehingga kedua ciphertext dapat dioperasikan untuk menghasilkan fungsi hash atau key yang digunakan. AES-GCM menggunakan kunci simetris dan IV dalam proses enkripsi dan autentikasinya. Dalam skenario terburuk, weak key collision dapat menyebabkan perubahan data dan pengaksesan data oleh pihak yang tidak berwenang. Untuk menghindari hal ini, para pengguna AES-GCM biasa menggunakan counter mode, di mana tiap IV dibangkitkan secara berurutan untuk mengurangi kejadian collision. Namun, berhubung IV yang digunakan AES-GCM secara aktual adalah 96 bit, penggunaan IV selain ukuran itu dapat menyebabkan pengulangan IV dalam counter mode. Hal ini mengakibatkan besar IV yang sesungguhnya dapat digunakan tetaplah 96 bit. Bukan angka yang kecil, tetapi hal ini merupakan risiko signifikan terutama dalam cloud storage di mana banyak pengguna mungkin melakukan pengunggahan dan pengaksesan banyak sekali data secara bersamaan[13].

- Keterbatasan Ukuran Pesan. Algoritma AES-GCM memiliki keterbatasan dalam ukuran pesan yang dapat dienkripsi, yaitu 64 GB. Batasan ini disebabkan oleh penggunaan IV sebesar yang unik dan counter sebesar 32-bit dalam mode operasi GCM. Suatu pesan yang melebihi batas ini dapat mengakibatkan pengulangan counter, yang secara signifikan meningkatkan risiko collision yang membahayakan keamanan data terenkripsi. Oleh karena itu, dibutuhkan pemrosesan yang membagi pesan enkripsi menjadi berbagai bagian lebih kecil[14].

- Keterbatasan Hardware. Dalam implementasinya, algoritma AES-GCM memerlukan perangkat keras yang sangat canggih karena besarnya kompleksitas operasional serta kebutuhan untuk pemrosesan secara paralel dalam jumlah banyak. Penggabungan kemampuan enkripsi dan autentikasi memerlukan perhitungan cepat dan efisien untuk menjaga kecepatan tanpa mengorbankan keamanan. Selanjutnya, operasi AES-GCM seperti carry-less multiplication dan pengelolaan nonce yang unik, membutuhkan kemampuan komputasi yang besar. Jika algoritma diimplementasikan tanpa dukungan perangkat keras yang cukup, sistem dapat mengalami keterlambatan signifikan, yang kemudian dapat meningkatkan potensi kerentanan terhadap serangan data. Sebagai contohnya, algoritma ini hingga sekarang belum dapat dijalankan dalam semua OS Android, sehingga terdapat beberapa restriksi dalam penggunaannya.

- Risiko Pemalsuan Tinggi. Risiko pada AES-GCM juga melibatkan potensi pemulihan kunci autentikasi. Teknik pemalsuan ini memiliki kemungkinan berhasil yang cukup besar setelah sekitar 216 upaya pemalsuan. Untuk melakukan ini, penyerang mengambil pesan yang terautentikasi, dan menerapkan perbedaan yang dirancang dengan baik pada pesan tersebut, untuk memastikan bahwa setengah dari bit tag autentikasi tidak akan berubah. Serangan ini dilakukan secara berulang terhadap teks terenkripsi, dan jika pesan mampu mendapatkan validasi autentikasi. Ketika hal ini terjadi, penyerang akan menemukan informasi tentang kunci autentikasi

yang dapat digunakan untuk mengulangi serangan dan secara perlahan menemukan keseluruhan kunci autentikasi. Dalam enkripsi untuk cloud storage, risiko ini cukup signifikan mengingat penyimpanan dapat dilakukan pada lokasi yang sangat jauh, sehingga penyerang memiliki banyak kesempatan selama transmisi untuk melakukan penyerangan.

C. *Implementasi AES-GCM dalam cloud storage*

Dalam lingkungan cloud storage, keamanan data menjadi prioritas utama dalam menilai efektivitas enkripsi. Kemudian, kecepatan dan berat operasi data juga penting untuk diperhitungkan karena seringnya data akan diakses dan ditambahkan. Galois/Counter Mode (GCM) adalah salah satu metode enkripsi yang paling banyak digunakan dalam cloud storage karena memberikan tingkat keamanan yang tinggi dengan kinerja yang cukup efisien sekaligus mengautentikasi pengguna.

Berikut merupakan gambaran implementasi AES-GSM untuk melakukan penyimpanan pada cloud storage yang terjadi saat ini:

- Enkripsi Data Sebelum Pengunggahan. Data yang akan disimpan di cloud dienkripsi menggunakan algoritma enkripsi AES-GCM sebelum dikirim ke server penyimpanan. Proses enkripsi menghasilkan ciphertext dan tag autentikasi.

- Pengelolaan Kunci. Kunci enkripsi disimpan secara terpisah agar hanya dapat diakses oleh pemilik data. Kunci dapat disimpan secara lokal oleh pengguna atau menggunakan layanan manajemen kunci (KMS) yang disediakan oleh penyedia cloud. Protokol manajemen kunci ini juga harus memiliki mekanisme distribusi, penyimpanan, dan rotasi (penggantian secara berkala) kunci secara aman.

- Pengunggahan Data ke Cloud. Setelah data dienkripsi, ciphertext dan tag autentikasi dikirim ke server penyimpanan cloud storage. Penyedia layanan cloud kemudian menyimpan ciphertext dan tag autentikasi, sambil memastikan tag autentikasi sesuai, menandakan data tidak mengalami perubahan selama transmisi. Penyedia ini tidak memiliki akses ke kunci enkripsi, sehingga data tidak akan bisa diakses orang lain.

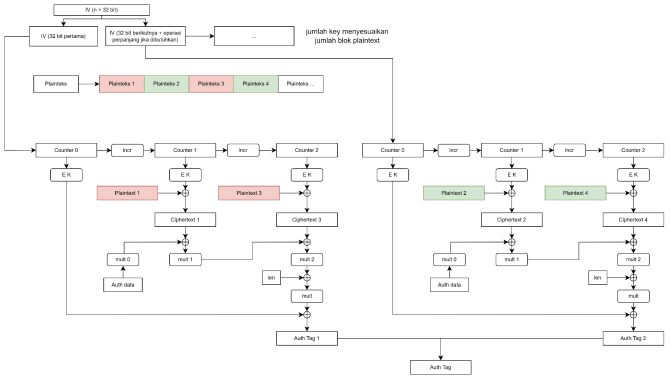
- Autentikasi Data. Saat data ingin diakses atau diunduh, tag autentikasi yang dihasilkan selama enkripsi diperiksa untuk memastikan bahwa data tidak mengalami perubahan. Jika tag autentikasi cocok dengan tag yang dihasilkan selama enkripsi, data dianggap valid dan aman dari serangan.

- Dekripsi Data Setelah Pengunduhan. Pengguna yang berwenang dapat mengunduh ciphertext dari cloud dan mendekripsinya menggunakan kunci enkripsi yang sesuai. Proses dekripsi melibatkan memverifikasi tag autentikasi dan mengembalikan data asli (plaintext) jika autentikasi berhasil.

D. *Desain Peningkatan Penggunaan AES-GCM dalam Cloud Storage*

Berdasarkan analisis yang telah dilakukan, diketahui bahwa kelemahan utama AES-GCM saat ini terletak pada keterbatasan IV yang juga menyebabkan keterbatasan panjang pesan. Selain itu, terdapat juga risiko dalam bentuk manajemen kunci dan keterbasan dalam bentuk kebetuhan hardware.

Berhubung tidak ada satu solusi manajemen kunci yang bisa dianggap aman dan dibutuhkannya perubahan dan fleksibilitas dalam manajemen kunci, desain peningkatan ini tidak akan berfokus pada penyimpanan kunci AES-GCM. Kemudian karena keamanan data dianggap lebih penting dibandingkan aksesibilitas hardware akibat terus meningkatnya kualitas hardware tiap tahunnya, desain peningkatan ini juga tidak akan berfokus pada simplifikasi algoritma AES-GCM.



Gambar 6. Ilustrasi Desain AES-GCM untuk Message Besar

Ilustrasi di atas menggambarkan modifikasi terhadap algoritma AES-GCM untuk mengatasi keterbatasan jumlah IV dan panjang pesan yang dapat dienkripsi. Berikut merupakan cara kerja desain yang dirancang:

- Pemecahan IV menjadi satu atau lebih berdasarkan panjang pesan yang ada. IV kemudian menggunakan algoritma hashing dalam rangka terjadinya IV yang tidak memenuhi panjang yang seharusnya.
- Pemecahan blok-blok plaintext sesuai jumlah IV yang telah ditentukan. Pemecahan blok dilakukan secara bergantian untuk mempersulit penyerangan brute force karena penyerang tidak akan memiliki informasi jumlah pembagian yang digunakan.
- Enkripsi seperti biasa menggunakan IV dan auth data yang terpisah. Hal ini memastikan satu bagian dari enkripsi tidak dapat digunakan untuk mendapatkan informasi mengenai bagian lain dari enkripsi.

Berikut merupakan gambaran *source code* implementasi desain di atas:

```
def encryptGCM (plaintext, key, IV, AAD):
  divisions = (plaintext.len()/8192 +1)
  IV = IV*divisions[0:divisions*43]
  for i in range (plaintext.len()-16/8192):
    H = Encrypt_Block(0, key) # Hash key
```

```
    Y0 = Initialize_Counter(IV)
    ciphertext = []
    counter = Y0

     # Start encryption
    for j in range (plaintext.len()-16/32):
      block = plaintext[j*divisions*8 : (j*(divisions+1)*8)-1]
      keystream = Encrypt_Block(counter, key)
      ciphertext_block = XOR(block, keystream)
      ciphertext.append(ciphertext_block)
      counter = Increment_Counter(counter)

  # Start authentication process
   auth_tag = Galois_Field_Multiply(H, AAD, ciphertext)

   return (ciphertext, auth_tag)

def decryptGCM(ciphertext, key, IV, AAD, auth_tag):
  divisions = (plaintext.len()-16/8192 +1)
  IV = IV*divisions[0:divisions*43]

  for i in range (plaintext.len()-16/8192):
    H = Encrypt_Block(0, key) # Hash key
    Y0 = Initialize_Counter(IV)
    ciphertext = []
    counter = Y0

     # Start decryption
    for j in range (plaintext.len()-16/32):
      keystream = Encrypt_Block(counter, key)
      plaintext_block = XOR(block, keystream)
      plaintext.append(plaintext_block)
      counter = Increment_Counter(counter)

   # Start authentication process
   expected_auth_tag = Galois_Field_Multiply(H, AAD, ciphertext)
   if auth_tag != expected_auth_tag:
      raise Exception("Authentication failed")

   return plaintext
```

IV. KESIMPULAN DAN SARAN

Berdasarkan evaluasi yang telah dilakukan, dapat disimpulkan bahwa secara garis besar, algoritma GCM merupakan algoritma yang cukup memadai dan unggul dalam melakukan enkripsi terhadap data yang sensitif dan berjumlah besar. Hal-hal ini bersifat sangat penting dalam manajemen data pada cloud storage, sehingga GCM masih dinilai cocok untuk melakukan enkripsi tugas ini.

Kelemahan GCM terutama dalam implementasinya bersama AES terletak pada keterbatasan pembangkitan IV, keterbatasan besar pesan, besarnya kebutuhan hardware, serta risiko dalam manajemen kunci yang telah digunakan dalam enkripsi.

Keterbatasan dalam IV dan besar pesan merupakan masalah utama yang dapat membatasi pencapaian algoritma ini dalam cloud storage karena data yang disimpan dalam cloud cenderung besar dan banyak. Sehingga diberikan saran untuk implementasi modifikasi algoritma dengan cara memecah plaintext dan authorization data menjadi dua atau lebih bagian. Dengan cara ini, IV yang digunakan dapat berbeda satu sama lain, dan irisan IV antara pembagian enkripsi yang berbeda tidak berpengaruh karena tidak adanya keterhubungan antara satu dengan yang lain.

Besarnya kebutuhan hardware merupakan pembatas dari penggunaan algoritma AES-GCM, tetapi semua operasi yang dilakukan dianggap penting dalam memastikan penyimpanan pada cloud storage. Maka diberikan saran untuk menggunakan algoritma AES-GCM tanpa paralelisme. Hal ini tentunya akan menyebabkan enkripsi dan penyimpanan data memiliki durasi yang lebih lama. Namun, hal ini bukan masalah karena pentingnya menjamin keamanan data, serta mengingat perkembangan hardware yang terus meningkat, sehingga dalam masa depan dapat diperkirakan masalah ini akan semakin berkurang.

Terakhir, manajemen kunci merupakan permasalahan yang dihadapi semua jenis enkripsi tanpa pengecualian. Hal ini harus diselesaikan dalam cara yang berbeda dan tertutup untuk mempersulit penyerangan data. Oleh karena itu, disarankan manajemen kunci dilakukan secara terpisah dan dengan mengimplementasikan key rotation secara berkala untuk menghindari terjadinya kebocoran data.

VIDEO LINK AT YOUTUBE *(Heading 5)*

Include link of your video on YouTube in this section.

DAFTAR PUSTAKA

[1] Heiligenstein, Michael X. Google Data Breaches: Full Timeline Through 2023. Firewall Times. 2023. https://firewalltimes.com/google-data-breach-timeline/

[2] NIST. Recommendation for Block Cipher Modes of Operation: Galois/Counter Mode (GCM) and GMAC. 2007. https://csrc.nist.gov/pubs/sp/800/38/d/final

[3] Google Cloud. Default encryption at rest. 2024. https://cloud.google.com/docs/security/encryption/default-encryption

[4] O'Brien, J. A. & Marakas, G. M. Computer Software. Management Information Systems 10th ed. 145. McGraw-Hill/Irwin. 2011.

[5] Leung, Leo. 4 reasons why cloud and on-premises storage are different, but equally good for people data. Oxygen Cloud. 2013. https://web.archive.org/web/20130925073953 /http://blog.oxygencloud.com/2013/09/09/4-reasons-why-cloud-and-on-premises-storage-are-different/

[6] S. Subashini, V. Kavitha. A survey on security issues in service delivery models of cloud computing. Journal of Network and Computer Applications Volume 34, Issue 1. 2011. ISSN 1084-8045. https://doi.org/10.1016/j.jnca.2010.07.006

[7] Dhillon and Backhouse. Information System Security Management in the New Millennium. 2000. DOI:10.1145/341852.341877

[8] Dang, Quynh dan Kevin Stine. Encryption Basics. Journal of AHIMA (American Health Information Management Association). 2011. https://tsapps.nist.gov/publication/ get_pdf.cfm?pub_id=908084

[9] Dworkin, Morris J., dkk. Advanced Encryption Standard (AES). 2001. https://www.nist.gov/publications/advanced-encryption-standard-aes

[10] Nabihah J., Ahmad. Phys.: Conf. Ser. 1019 012008. 2018. https://iopscience.iop.org/article/10.1088/1742-6596/1019/1/012008/pdf#:~:text=Advanced%20Encryption%20Standard%20with%20Galois%20Counter%20Mode%20(AES%2DGCM),computers%20and%20other%20communication%20applications

[11] Saarinen, Markku-Juhani O.. Cycling Attacks on GCM, GHASH and Other Polynomial MACs and Hashes. 2011. https://eprint.iacr.org/2011/202.pdf

[12] McGrew, David A., Viega, John. The Galois/Counter Mode of Operation (GCM). 2004. https://csrc.nist.rip/groups/ST/toolkit/BCM/documents/proposedmodes/gcm/gcm-spec.pdf

[13] Saarinen, Markku-Juhani O. Cycling Attacks on GCM, GHASH and Other Polynomial MACs and Hashes. 2011. https://eprint.iacr.org/2011/202

[14] Gueron, Shay, Kounavis, Michael E.. Intel® Carry-Less Multiplication Instruction and its Usage for Computing the GCM Mode. 2014. https://www.intel.com/content/dam/develop/external/us/en/documents/clmul-wp-rev-2-02-2014-04-20.pdf

[15] Arghire, Ionut. Google Cloud Platform Vulnerability Led to Stealthy Account Backdoors. 2023. https://www.securityweek.com/google-cloud-platform-vulnerability-led-to-stealthy-account-backdoors/

[16] IDStrong. Google Cloud Breach. 2021. https://www.idstrong.com/data-breaches/google-cloud-breach/#:~:text=How%20Did%20the%20Breach%20Occur,response%2C%20the%20investigators%20alerted%20Google

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, 12 Juni 2024

Nadira Rahmananda Arifandi