ISSN: 2614-6754 (print)
ISSN: 2614-3097(online)

Halaman 12173-12181
Volume 7 Nomor 2 Tahun 2023

Metode Pengamanan Komunikasi E-Mail pada Perangkat Mobile Berbasis Android dengan Menggunakan Metode Hybrid Cryptosystem

Andri[1], Halimatussa'diah[2], Julizal[3]

[1,2,3]Program Studi Informatika, Fakultas Teknik, dan Ilmu Komputer, Universitas Indraprasta PGRI

Email: andriecitra@gmail.com[1], gbhock300679@gmail.com[2], julizal.ram@gmail.com[3]

Abstrak

Beberapa tahun yang lalu, jika seseorang ingin terhubung ke internet seperti surat elektronik (email), browsing web, chatting dan lain-lain, Itu masih tergantung pada komputer dengan koneksi sambungan tetap. Namun, seiring dengan pesatnya perkembangan teknologi dari waktu ke waktu sekarang dapat dilakukan tanpa harus bergantung pada komputer yang menggunakan saluran tetap lagi. Saat ini hampir semua layanan internet dapat dinikmati secara mobile di mana pun dan kapan pun menggunakan perangkat seluler seperti notebook, smartphone, tablet PC dan sebagainya. Salah satu layanan internet paling populer yang digunakan pada perangkat seluler berbasis Android adalah email dengan fitur push email-nya. Dengan perangkat seluler yang terhubung ke internet, layanan email dapat digunakan secara luas oleh berbagai kelompok untuk bertukar informasi dan berkolaborasi, baik untuk kepentingan individu, kelompok, institusi dan perusahaan. Disadari atau tidak, penggunaan email untuk bertukar informasi dan berkolaborasi tidak hanya terbatas pada informasi biasa, tetapi juga informasi sensitif, yaitu informasi yang memiliki nilai kerahasiaan yang jika jatuh ke pihak lain yang tidak berhak dapat merugikan pihak-pihak tertentu. Dalam penelitian ini penulis mengusulkan metode keamanan komunikasi email pada perangkat mobile berbasis Android menggunakan metode hybrid cryptosystem yang merupakan kombinasi dari algoritma kriptografi simetris, algoritma kriptografi asimetris, fungsi hash dan sistem generasi kunci acak yang diharapkan dapat memenuhi semua aspek keamanan informasi.

Kata Kunci : *Email, Android, Linux, Internet, hybrid cryptosystem*

Abstract

A few years ago, if someone wanted to connect to the internet to use services such as electronic mail (e-mail), web browsing, chatting, etc., it was still very dependent on computers with fixed line connections. However, along with the rapid development of technology from time to time it can now be done without having to rely on computers that use fixed line again. At present almost all internet services can be enjoyed on a mobile basis wherever and whenever using mobile devices such as notebooks, smartphones, tablet PCs and so on. One of the most popular internet services used on Android-based mobile devices is e-mail with its push e-mail feature. With mobile devices connected to the internet, e-mail services can be widely used by various groups to exchange information and collaborate, both for the benefit of individuals, groups, institutions and corporations. Whether we realize it or not, the use of e-mail to exchange information and collaborate is not only limited to ordinary information, but also sensitive information, which is information that has a value of confidentiality which if it falls to other parties that are not entitled can harm the parties certain ones. In this study the authors propose a method of e-mail communication security on Android-based mobile devices using the hybrid cryptosystem method which is a combination of symmetric cryptographic algorithms, asymmetric cryptographic algorithms, hash functions and random key generation systems that are expected to meet all aspects of information security.

Keywords: *Email, Android, Linux, Internet, hybrid cryptosystem*

PENDAHULUAN

Dengan makin meningkatnya mobilitas masyarakat, peranan perangkat mobile sebagai alat berkomunikasi dan bertukar informasi juga semakin berkembang pesat. Belakangan ini, salah satu platform perangkat mobile yang sedang marak digunakan yaitu Android. Android adalah sistem operasi open source berbasis Linux untuk perangkat mobile yang menyediakan kemudahan dan keleluasaan bagi para pengembang aplikasi untuk membangun aplikasi pada perangkat mobile. Salah satu layanan internet yang paling populer digunakan pada perangkat mobile berbasis Android adalah e-mail dengan fitur push e-mail-nya. Dengan perangkat mobile yang terhubung dengan internet, layanan e-mail dapat digunakan secara luas oleh berbagai kalangan untuk saling bertukar informasi dan berkolaborasi, baik untuk kepentingan individu, kelompok, institusi maupun korporasi. Disadari atau tidak, pemanfaatan e-mail untuk bertukar informasi dan berkolaborasi, tidak hanya terbatas pada informasi yang bersifat biasa saja, tetapi juga informasi yang sensitif, yaitu informasi yang memiliki nilai kerahasiaan yang apabila jatuh kepada pihak lain yang tidak berhak dapat merugikan pihak-pihak yang tertentu.

Apabila informasi yang sensitif dikirimkan melalui e-mail maka dibalik berbagai kemudahan yang ditawarkan oleh layanan e-mail, keamanan merupakan isu yang tidak dapat dihindari karena internet menggunakan sistem jaringan terbuka yang sangat rentan terhadap kemungkinan pemanfaatan dan penyalahgunaan informasi oleh pihak-pihak yang tidak berhak. Selain itu, sistem e-mail tidak memiliki proteksi terhadap keutuhan e-mail sehingga isi e-mail dapat diubah atau dimodifikasi baik pada saat transmisi maupun pada saat tersimpan di e-mail server tanpa terdeteksi. E-mail juga tidak memiliki sistem otentikasi pengirim sehingga tidak bisa menjamin bahwa e-mail dikirim oleh pengirim yang sesuai dengan alamat e-mail pengirim sehingga pemilik alamat e-mail pengirim dapat menyangkal bahwa ia yang telah mengirim e-mail. Untuk melindungi informasi yang sensitif yang dikomunikasikan melalui e-mail, maka perlu diimplementasikan segala hal yang berkaitan dengan keamanan informasi pada e-mail. Aspek-aspek keamanan informasi tersebut meliputi aspek kerahasiaan (confidentiality), keutuhan data (data integrity), keaslian (authentication), dan tidak dapat dilakukan penyangkalan (non-repudiation)[Menezes 1996].

METODE

Tujuan dari penelitian ini yaitu untuk membangun sebuah sistem pengamanan komunikasi e-mail pada perangkat mobile berbasis Android menggunakan hybrid cryptosystem yang diharapkan dapat memenuhi seluruh aspek keamanan informasi. Berdasarkan tujuan tersebut, maka metode yang digunakan dalam penelitian ini adalah metode penelitian dan pengembangan (research and development). Metode penelitian dan pengembangan adalah metode penelitian yang digunakan untuk menghasilkan produk tertentu dan menguji keefektifan produk tersebut[Sugiyono 2013]. Dalam konteks ini, produk yang dimaksud tidak hanya berbentuk hardware seperti buku, modul, alat bantu laboratorium dan sebagainya, tetapi juga bisa berupa software seperti program untuk pengolahan data, pembelajaran di kelas, perpustakaan, laboratorium ataupun model-model untuk pendidikan, evaluasi, manajemen dan sebagainya.

Hasil dan Pembahasan

Electronic Mail

1. Pengertian *Electronic Mail*

Electronic mail atau biasa disebut dengan e-mail merupakan salah satu layanan internet yang sangat populer dan paling banyak digunakan baik di kepentingan individu, kelompok, institusi maupun korporasi. E-mail digunakan untuk saling bertukar informasi atau mengirim pesan antara seseorang dengan orang lainnya yang terpisahkan oleh jarak melalui perangkat telekomunikasi. E-mail beroperasi seperti halnya surat kertas atau konvensional dengan layanan pos. Seseorang dapat menulis pada kertas dan

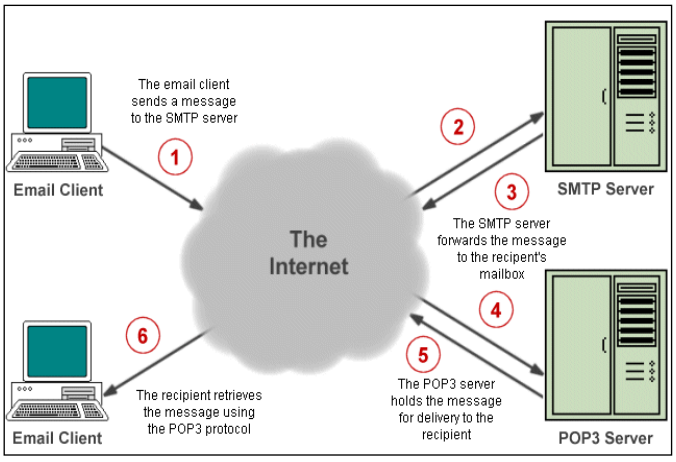
menempatkannya pada amplop. Jika orang tersebut membubuhkan nama dan alamat yang benar di depan amplop dan menempatkannya dalam kotak surat, maka orang tersebut dapat mengharapkan layanan pos mengirimkan surat ke tempat yang benar. Jika orang tersebut menempatkan alamat pengirim di amplop, penerima dapat membalas dengan menempatkan alamat orang tersebut di depan amplop sebagai tujuan surat tersebut dikirim.

Sama halnya dengan sistem pengiriman surat kertas melalui layanan pos, namun e-mail tidak menggunakan kertas melainkan menggunakan suatu aplikasi berbentuk program komputer dengan media komunikasi berupa jaringan komputer atau internet. E-mail memungkinkan seseorang menuliskan pesan berupa teks dan lampiran file (attachment), mengidentifikasi siapa saja yang ingin orang tersebut kirimi dengan menuliskan alamat e-mail seseorang di bagian tujuan pengiriman e-mail dan mengirimkan ke alamat tersebut. Dengan mengirimkan e-mail, seseorang dapat mengerjakan hal yang sama seperti mengirimkan e-mail kepada layanan pos. Layanan ini mengirim e-mail dan kemudian orang yang dikirimi pesan memeriksa kotak masuk (inbox), kemudian dia menerima pesan yang telah orang lain kirimkan.

2. Sistem Pengiriman E-mail

Dalam proses kirim dan terima pesan e-mail selalu menggunakan standar TCP/IP dengan standar IMF (Internet Message Format) untuk menentukan header yang digunakan untuk mengenkapsulasi isi pesan. Proses pengiriman pesan menggunakan protokol SMTP (Simple Mail Transport Protocol) dimana pengirim e-mail melalui Mail User Agent (MUA) atau biasa disebut dengan e-mail client akan mengirimkan e-mail Mail Transfer Agent (MTA) atau SMTP server. MUA adalah bagian dari sistem e-mail yang umumnya diketahui oleh pengguna. MUA adalah suatu program yang berinteraksi dengan pengguna dan tugasnya adalah menangani pesan-pesan e-mail yang masuk dan keluar. Sedangkan MTA bertanggung jawab untuk memindahkan atau mentransfer sebuah surat dari satu sistem ke sistem yang lain.

Setelah e-mail tiba di email server kemudian sesuai dengan alamat penerima e-mail, e-mail akan diteruskan ke e-mail server penerima yang berupa POP3 (Post Office Protocol versi 3) server. Selanjutnya dengan menggunakan protocol POP3 e-mail client penerima akan mengunduh e-mail yang masuk ke mailbox-nya yang berada di POP3 server. Sebagai catatan, untuk dapat mendapatkan pesan e-mail maka akun e-mail seseorang harus terdaftar terlebih dahulu di e-mail server atau POP3 server. Secara umum proses kirim dan terima e-mail diilustrasikan pada di bawah ini.



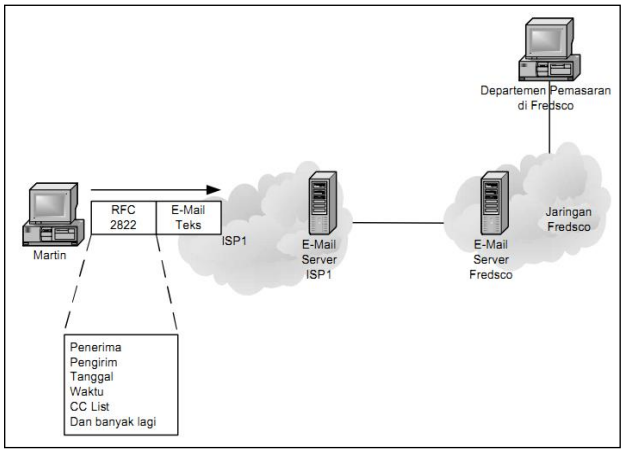
Gambar II-1. Proses kirim dan terima e-mail.

3. **Protokol dan Standar *E-mail***

E-mail memiliki protokol dan standar yang sering digunakan dalam pengiriman dan penerimaan pesan, agar pesan tersebut sampai tujuan. Standar dan protokol utama yang umumnya digunakan dalam layanan *e-mail* yaitu sebagai berikut.

a. RFC 2822 atau *Internet Message Format*

Saat pengirim *e-mail* mengirim *e-mail* dan kemudian *e-mail server* meneruskannya, sebenarnya tidak hanya teks isi *e-mail* saja yang ditransmisikan, namun *header* juga ikut ditransmisikan. *Header e-mail* terdiri dari beberapa *field* yang terdiri *e-mail address* dari pengirim, penerima, tanggal, waktu, daftar *carbon copy* (CC) dan lain sebagainya seperti yang digambarkan di bawah ini.



Gambar II-2. *Header e-mail* berdasarkan RFC 2822

b. RFC 821 atau Simple Mail Transport Protocol

Salah satu protokol *e-mail* yang populer adalah SMTP. Pada proses pengiriman *e-mail*, *e-mail client* dan *e-mail server* menggunakan protokol SMTP melalui port 25 untuk mengatur proses pengiriman dan penerimaan pesan. Jika akan mengirimkan suatu *e-mail*, maka SMTP *clinet* pada *e-mail client* akan membuka kanal dua arah ke SMTP *server*. Dalam hal ini SMTP *server* bisa merupakan tujuan akhir, namun kadang bisa juga menjadi perantara antara komputer penerima dengan komputer pengirim atau berupa gerbang yang menghubungkan komunikasi SMTP dengan protokol lain [Postel 1982].

Koneksi SMTP *client* dan *server* diawali dengan proses inisialisasi, SMTP *server* akan memberikan status bisa digunakan atau tidak. Jika tidak bisa digunakan maka koneksi diputus dan jika statusnya bisa digunakan SMTP *client* bisa memulai pengiriman kumpulan perintah yang diperlukan seperti menentukan alamat pengirim, alamat tujuan, serta pesan yang akan disampaikan. Setelah pesan dikirimkan oleh SMTP *server*, SMTP *client* bisa meminta koneksinya diputus atau dimulai untuk pengiriman *e-mail* lainnya. Pada pengiriman sebuah *e-mail*, SMTP *client* bertanggung jawab hanya sampai SMTP *server* dan memberikan informasi bahwa proses pengiriman telah selesai. Hal ini bukan berarti pesan tersebut telah dikirimkan dan diterima oleh penerima yang dimaksud.

c. RFC 1939 atau *Post Office Protocol* versi 3

POP3 merupakan protokol yang digunakan untuk mengambil *e-mail* dari *mailbox* pada *e-mail server* dan menyimpannya pada komputer lokal pengguna POP3 dengan menggunakan port 110 pada TCP/IP [Myers 1996]. Jika ada POP3 *client* yang akan menggunakan layanan POP3 *server*, maka koneksi antara keduanya berlangsung. Setelah terkoneksi, POP3 *server* akan memberikan sebuah pesan sambutan yang kemudian dilanjutkan pada tahap berikutnya yaitu tahapan otentikasi, dimana POP3 *client* harus mengidentifikasikan dirinya ke POP3 *server* dengan mengirimkan *user id* dan *password e-mail*.

Jika otentikasi berhasil dan sesuai dengan *e-mail* yang tersedia di POP3 *server*, maka pengguna akan mengambil data yang dibutuhkan dalam koneksi tersebut dan dilanjutkan dengan tahapan transaksi. Pada tahapan transaksi, pengguna bisa menggunakan beberapa perintah untuk berinteraksi dengan POP3 *server*, misalnya menampilkan daftar *e-mail* yang tersedia dalam *mailbox*. Semua pesan yang dikirimkan

dalam koneksi POP3 berupa kode ASCII dan format pesan email yang dikirimkan diasumsikan sesuai dengan standar pada RFC 822.

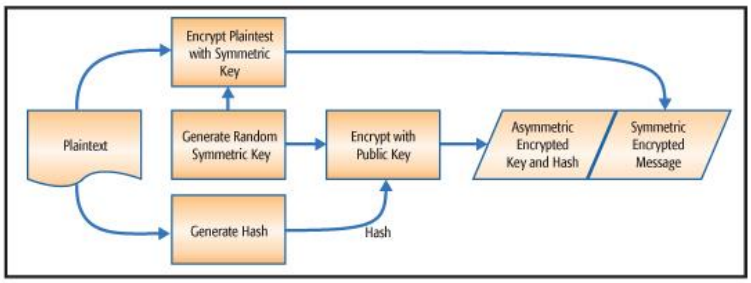
Hybrid Cryptosytem

Protokol kriptografi merupakan protokol yang dibangun dengan melibatkan beberapa algoritma kriptografi. Pentingnya penggunaan protokol kriptografi dalam proses komunikasi rahasia pada dasarnya adalah untuk mencegah atau mendeteksi terjadinya penyadapan dan pemodifikasian pesan oleh pihak yang tidak berhak yang ingin mengambil keuntungan dari pesan pada komunikasi tersebut. Sedangkan peruntukan penggunaan protokol kriptografi beraneka ragam, tergantung pada tujuan yang ingin dicapai.

Tujuan dari penggunaan protokol kriptografi saat ini biasanya yaitu untuk berbagi komponen rahasia dalam menghitung sebuah nilai, membangkitkan rangkaian bilangan acak dan meyakinkan identitas orang lain dan lain-lain. Protokol kriptografi yang berkembang saat ini kebanyakan menggabungkan penggunaan antara algoritma kriptografi simetrik dan asimetrik. Penggabungan 2 (dua) algoritma tersebut menghasilkan sistem yang disebut dengan hybrid cryptosystem[Rasmi 2011]. Pembuatan protokol dengan menggunakan hybrid cryptosystem dimaksudkan untuk memecahkan permasalahan key establishment (mekanisme penggunaan kunci yang disepakati) selain masalah kerahasiaan pesan.

Konsep hybrid cryptosystem berangkat dari adanya kelemahan dan kelebihan yang dimiliki oleh algoritma kriptografi simetrik dan asimetrik. Dalam aspek kecepatan proses enkripsi dan dekripsi, algoritma kriptografi simetrik memiliki kecepatan yang lebih baik dimana kecepatannya mencapai 1.000 kali lebih cepat dibandingkan dengan algoritma kriptografi asimetrik. Namun algoritma kriptografi simetrik tidak memiliki mekanisme key establishment sehingga kunci enkripsi harus didistribusikan kepada seluruh pihak yang berkepentingan. Pada proses distribusi kunci enkripsi inilah yang menjadi peluang bagi pihak yang tidak berwenang untuk mendapatkan kunci enkripsi tersebut. Sedangkan algoritma kriptografi asimetrik memiliki mekanisme untuk melakukan pertukaran kunci enkripsi secara aman. Dalam rangka saling melengkapi kelebihan dan kekurangan masing-masing dari sistem enkripsi simetrik dan asimetrik, maka dapat digunakan mekanisme hybrid cryptosystem untuk berkomunikasi dengan aman dengan mengkombinasi sistem enkripsi simetrik dan asimetrik.

Selain itu, masih terdapat permasalahan lain yaitu masalah keaslian pesan, keaslian entitas dan penyangkalan entitas. Sehingga untuk mengantisipasi semua permasalahan tersebut, banyak protokol yang menggabungkan berbagai sistem kriptografi, yaitu algoritma kriptografi simetrik, asimetrik, fungsi hash dan sistem pembangkit kunci acak. Secara umum mekanisme hybrid cryptosystem yang mengkombinasikan sistem algoritma kriptografi simetrik, asimetrik, fungsi hash dan sistem pembangkit kunci acak dapat dilihat pada gambar di bawah ini.



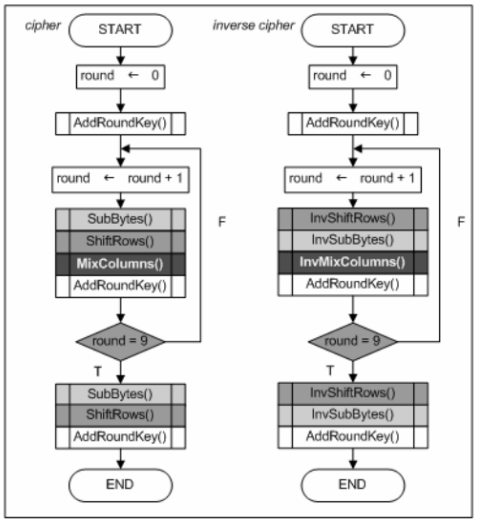
Gambar II-3. Mekanisme hybrid cryptsystem

Algoritma *Advanced Encryption Standard* (AES)

Algoritma Rijndael diumumkan oleh *National Institute of Standards and Technology* (NIST) sebagai *Federal Information Processing Standards* (FIPS) publikasi 197 (FIPS 197) pada tanggal 26 November 2001 setelah proses standardisasi selama 5 tahun, dimana ada 15 desain enkripsi yang disajikan dan dievaluasi, sebelum algoritma Rijndael terpilih sebagai

yang paling cocok sebagai *Advanced Encryption Standard* (AES). AES adalah nama standar yang diterbitkan NIST untuk kategori keamanan komputer[NIST 2001]. Standar algoritma ini oleh penerbitnya ditujukan bagi informasi sensitif di lingkungan pemerintah federal Amerika Serikat yang membutuhkan perlindungan kriptografi.

AES menyandikan data dalam empat langkah dasar yaitu, langkah nonlinear, langkah dispersi, langkah difusi dan penambahan kunci. Nonlinearisasi diperoleh dengan memakai tabel subsitusi nonlinear. Dispersi dilakukan lewat permutasi *byte-byte* data dari kolom *array* yang berbeda. Langkah difusi menyandikan data menjadi kombinasi linear dari *byte-byte* data dalam satu kolom array tersebut. Penambahan kunci dilakukan dengan operasi XOR antara data dengan kunci. Keempat langkah tersebut memiliki nama khusus dalam algoritma yang diterangkan AES sesuai yang diilustrikan pada gambar berikut.



Gambar II-4. Diagram alir *cipher* dan *inverse cipher* AES

Algoritma AES adalah algoritma kriptografi simetrik *block cipher* dimana kunci rahasia yang sama digunakan untuk menyandikan data maupun untuk memperoleh kembali data tersebut dari data tersandinya. Istilah AES-128 merujuk pada algoritma Rijndael dengan panjang blok data dan panjang kunci 128 bit. AES sendiri menggunakan panjang blok data 128 bit, tetapi panjang kunci bisa berbeda-beda (AES-128, AES-192, dan AES-256).

Representasi Data

Susunan *byte* dan bit data diturunkan dari urutan input 128 bit (blok). Bit-bit tersebut ditandai mulai dari 0 sampai dengan 127 ($0 \leq i \leq 128$). Setiap urutan 8 bit (byte) diperlakukan sebagai entitas tunggal, sebagai elemen *finite field* dengan representasi polinomial

$$b(x) = b_7x^7 + b_6x^6 + b_5x^5 + b_4x^4 + b_3x^3 + b_2x^2 + b_1x + b_0 \qquad (2.1)$$

Sebagai contoh {01100011} direpresentasikan oleh persamaan polinomial $x^6 + x^5 + x + 1$. *Byte* tersebut bernilai heksadesimal {63}, dimana pengindeksan bit dalam *byte* dan penomoran barisan *byte* dalam blok dapat dilihat lebih jelas dalam Tabel II-1 berikut ini.

Tabel II-1. Pengindeksan aliran input

bit	0	1	2	3	4	5	6	7	8	9	10	11	12	13	14	15	16	17	18	19	20	21	22
	in_0								in_1								in_2						
indeks	7	6	5	4	3	2	1	0	7	6	5	4	3	2	1	0	7	6	5	4	3	2	1
	$s_{0,0}$								$s_{1,0}$								$s_{2,0}$						

Cipher AES dilakukan pada *array byte* dua dimensi yang disebut *state*. Blok data disusun dalam *state* yang terdiri atas empat baris-*Nb byte* (*Nb* = panjang blok/2 adalah 4 untuk AES-128). Setiap *byte* diberi dua indeks yang menyatakan posisinya, yang dinyatakan sebagai $s_{r,c}$ atau $s[r,c]$, dengan nomor baris r dalam interval $0 \le r < 4$, sedangkan nomor kolom c dalam $0 \le c < Nb$. Data dalam *state* menginformasikan hasil setiap tahap transformasi (*intermediate result*).

Input disalin ke *state array* pada permulaan *cipher* dan *inverse cipher*, kemudian *state* diperbarui pada akhir setiap transformasi (Gambar II-1). Nilai *state* pada transformasi yang terakhir kemudian disalin ke *output* kembali dengan pengindeksan yang sama seperti pada Tabel II-1. *State* juga dapat dilihat sebagai *word* empat *byte*, dengan nomor baris r dari $s_{r,c}$ menyatakan indeks dari keempat *byte* dalam setiap *word*. Dengan kata lain, *state* ekivalen dengan *array* dari empat *word* yang berindeks c (nomor kolom dari $s_{r,c}$) seperti dinyatakan di bawah ini.

$$w_0 = s_{0,0}s_{1,0}s_{2,0}s_{3,0} \qquad w_2 = s_{0,2}s_{1,2}s_{2,2}s_{3,2}$$
$$w_1 = s_{0,1}s_{1,1}s_{2,1}s_{3,1} \qquad w_3 = s_{0,3}s_{1,3}s_{2,3}s_{3,3}$$

Operasi Dasar

Walaupun secara keseluruhan hasil antara setiap tahap transformasi melibatkan *state* (1 blok), unit dasar operasi AES adalah *byte*. Setiap *byte* sebagai elemen *finite field* dapat dijumlah maupun dikalikan.

1. Penjumlahan

   Penjumlahan dua elemen *finite field* diimplementasikan sebagai operasi XOR per bit. Sebagai konsekuensinya, pengurangan adalah operasi yang identik. Ekspresi berikut ini adalah ekivalen antara satu dengan lainnya (notasi polinomial, biner, dan heksadesimal).

   $(x^6 + x^4 + x^2 + x + 1) + (x^7 + x + 1) = x^7 + x^6 + x^4 + x2$
   $\{01010111\} \oplus \{10000011\} = \{11010100\}$
   $\{57\} \oplus \{83\} = \{d4\}$

2. Perkalian

   Perkalian elemen *Galois Field* (28) (notasi ●) dalam representasi polinomial adalah perkalian dengan modulo $m(x) = x^8 + x^4 + x^3 + x + 1$. Bilangan modulo $m(x)$ adalah *irreducible polynomial* GF(28).

   3. $xb(x) = b_8x^8 + b_7x^7 + b_6x^6 + b_5x^5 + b_4x^4 + b_3x^3 + b_2x^2 + b_1x^1 + b_0x$ (2.2)

   Perkalian $x \bullet b(x)$ dapat diwujudkan sebagai *left shift* yang diikuti XOR kondisional dengan {1b}, jika $b_8 = 1$, maka XOR dilakukan, jika $b_8 = 0$, maka XOR tidak dilakukan. Exclusive-OR kondisional tersebut tidak lain adalah operasi modulo dengan $m(x)$. Serangkaian *left shift* yang disusul operasi XOR tersebut dapat digunakan untuk perkalian antar elemen *finite field*.

Key Expansion

Key Expansion merupakan *routine* untuk menghasilkan *Round Key*, kunci yang ditambahkan pada setiap *round*. Dari empat *word cipher key*, *K*, akan dihasilkan 44 *word* ekspansinya, w_i, dengan $0 \le i < 44$. Empat *word* ekspansi pertama adalah *cipher key* itu sendiri. Setiap *word* berikutnya, $w[i]$, dihasilkan dari operasi XOR *word* sebelumnya, $w[i-1]$, dengan word *Nk* posisi sebelumnya, $w[i-Nk]$. *Nk* sama dengan 4 untuk AES-128. Apabila i kelipatan dari Nk, maka sejumlah transformasi dilakukan pada $w[i-Nk]$ sebelum operasi XOR di atas, diikuti oleh XOR dengan *word* konstanta *round*, Rcon.

Transformasi yang pertama adalah *SubWord()*, *word* $w[i-4]$ tersebut dipetakan ke nilai S-Box-nya. Keluaran *SubWord()* kemudian dipermutasi secara siklik oleh fungsi *RotWord()*, *word* $[a_0, a_1, a_2, a_3]$ akan menjadi $[a_1, a_2, a_3, a_0]$ setelah *RotWord()*. Konstanta Rcon di atas adalah *word* $[\{02\}^{i-1}, \{00\}, \{00\}, \{00\}]$.

ISSN: 2614-6754 (print)
ISSN: 2614-3097(online)
Halaman 12173-12181
Volume 7 Nomor 2 Tahun 2023

Enkripsi

Proses *cipher* (Gambar II-2) berlangsung dalam rentetan empat fungsi pembangun (primitif), *SubBytes()*, *ShiftRows()*, *MixColumns()*, dan *AddRoundKey()*. Rentetan tersebut dijalankan sebanyak *Nr* - 1 sebagai *loop* utama (*Nr* = 10 untuk AES-128). Setiap *loop* disebut *round*. *AddRoundKey()* dieksekusi sebagai *round* inisial sebelum *loop* utama. Setelah *loop* utama tersebut berakhir (sembilan *round*), *SubBytes()*, *ShiftRows()*, *MixColumns()* dan *AddRoundKey()* dieksekusi secara berturutan sebagai *final round*.

1. *AddRoundKey()*

   Penjumlahan dilakukan antara *state* dengan *round key* hasil ekspansi. Persamaan berikut ini menjabarkan penjumlahan tersebut.

   $$[s'_{0,c}, s'_{1,c}, s'_{2,c}, s'_{3,c}] = [s_{0,c}, s_{1,c}, s_{2,c}, s_{3,c}] \oplus [w_{round*4+c}] \quad (2.3)$$

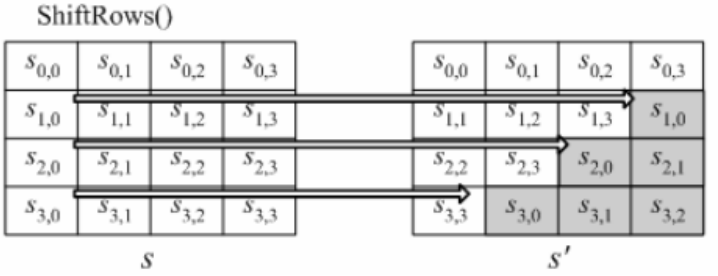
   dengan $0 \le c < 4$ (penjumlahan per blok).

2. *SubBytes()*

   Tabel *substitusi byte* untuk langkah nonlinear tersedia sebagai S-Box. Transfomasi yang telah ditabelkan tersebut mengambil invers multiplikatif GF(28) setiap *byte*, kemudian diikuti dengan transformasi *affine*.

3. *ShiftRows()*

   ShiftRows() merupakan langkah permutasi yang dieksekusi lewat pergeseran siklik tiga baris terakhir *state* (baris pertama, $r = 0$, tidak digeser). Baris kedua digeser siklik ke kanan sekali, baris ketiga dua kali, baris keempat tiga kali seperti diilustrasikan pada Gambar II-5 berikut.



Gambar II-5. Transformasi *ShiftRows()*

4. *MixColumns()*

   Difusi diperoleh lewat transformasi MixColumns() yang mengoperasikan *state* kolom-demi-kolom.

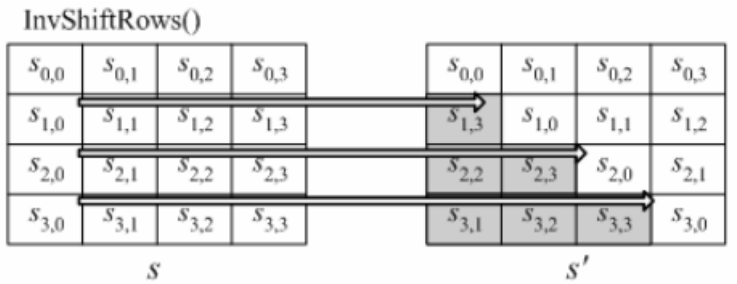
   $$s'_{0,c} = (\{02\} \bullet s_{0,c}) \oplus (\{03\} \bullet s_{1,c}) \oplus s_{2,c} \oplus s_{3,c}$$
   $$s'_{1,c} = s_{0,c} \oplus (\{02\} \bullet s_{1,c}) \oplus (\{03\} \bullet s_{2,c}) \oplus s_{3,c}$$
   $$s'_{2,c} = s_{0,c} \oplus s_{1,c} \oplus (\{02\} \bullet s_{2,c}) \oplus (\{03\} \bullet s_{3,c})$$
   $$s'_{3,c} = (\{03\} \bullet s_{0,c}) \oplus s_{1,c} \oplus s_{2,c} \oplus (\{02\} \bullet s_{3,c}) \quad (2.4)$$

Dekripsi

Setiap fungsi enkripsi di atas memiliki kebalikan, dekripsi berlangsung dengan kebalikan dari setiap primitif (*inverse cipher*) (Gambar II-4). *AddRoundKey()* dieksekusi sebagai *initial round*, diikuti sembilan *round* rentetan *InvShiftRows()*, *InvSubBytes()*, *InvMixColumns()*, dan *AddRoundKey()*. Round ke-10 yang mengikutinya tidak menyertakan *InvMixColumns()* serupa dengan *round* terakhir enkripsi.

1. *InvShiftRows()*

   Kebalikan *ShiftRows()* ini dilakukan dengan menggeser siklik ke arah berlawanan. Baris ke dua digeser siklik ke kiri sekali, baris ke tiga dua kali, baris ke empat tiga kali sesuai dengan Gambar II-6 berikut.



Gambar II-6. Transformasi *InvShiftRows()*

2. *InvSubBytes()*
   Merupakan invers dari tabel S-Box yang digunakan untuk SubBytes tersedia sebagai S-Box^{-1}.
3. *InvMixColumns()*
   Operasi *state* per kolom yang dilakukan *MixColumns()* memiliki kebalikan berupa persamaan berikut ini.

$$s'_{0,c} = (\{0e\} \bullet s_{0,c}) \oplus (\{0b\} \bullet s_{1,c}) \oplus (\{0d\} \bullet s_{2,c}) \oplus (\{09\} \bullet s_{3,c})$$
$$s'_{1,c} = (\{09\} \bullet s_{0,c}) \oplus (\{0e\} \bullet s_{1,c}) \oplus (\{0b\} \bullet s_{2,c}) \oplus (\{0d\} \bullet s_{3,c})$$
$$s'_{2,c} = (\{0d\} \bullet s_{0,c}) \oplus (\{09\} \bullet s_{1,c}) \oplus (\{0e\} \bullet s_{2,c}) \oplus (\{0b\} \bullet s_{3,c})$$
$$s'_{3,c} = (\{0b\} \bullet s_{0,c}) \oplus (\{0d\} \bullet s_{1,c}) \oplus (\{09\} \bullet s_{2,c}) \oplus (\{0e\} \bullet s_{3,c}) \quad (2.5)$$

SIMPULAN

Salah satu layanan internet yang paling populer digunakan pada perangkat *mobile* berbasis Android adalah *e-mail*. Dengan perangkat *mobile* yang terhubung dengan internet, layanan *e-mail* dapat digunakan secara luas oleh berbagai kalangan untuk saling bertukar informasi dan berkolaborasi. Disadari atau tidak, pemanfaatan *e-mail* untuk bertukar informasi dan berkolaborasi, tidak hanya terbatas pada informasi yang bersifat biasa saja, tetapi juga informasi yang sensitif, yaitu informasi yang memiliki nilai kerahasiaan yang apabila jatuh kepada pihak lain yang tidak berhak dapat merugikan pihak-pihak yang tertentu.

DAFTAR PUSTAKA

Menezes, Alfred J, et.al, Handbook of Applied Cryptography". Florida: CRC Press Inc., 1996.

Myers. J. and Rose M, RFC 1939, Post Office Protocol – Version 3. 1996. http//www.ietf.org/rfc/rfc1939.txt. Diakses 5 Juni 2012).

National Institute of Standards and Technology. 2001. Federal Information Processing Standards Publication (FIPS) 197, Advanced Encryption Standard. Washington DC

Postel, Jonathan B., RFC 821, Simple Mail Transfer Protocol. 1982. http//www.ietf.org/rfc/rfc821.txt. Diakses 5 Juni 2012).

Sugiyono. 2013. Metode Penelitian Kuantitatif, Kualitatif, dan R&D. Bandung: Alfabeta. Halaman 60

Rasmi, P S. and Paul, Varghese., "A Hybrid Crypto System based on a CircleSymmetric key Algorithm and RSA with CRT Asymmetric key Algorithm for E-commerce Applications". International Conference on VLSI, Cominunication & Instrumentation (ICVCI). 2011): 14-18.