Analisa Kinerja Teknik dan Algoritma Keamanan SMS

Tugas Akhir Mata Kuliah Keamanan Informasi dan Jaringan EL5241

Ayu Rosyida Zain

(23214315)

Magister Teknik Elektro
Sekolah Teknik Elektro dan Informatika
Institut Teknologi Bandung
2015

ABSTRAK

Saat ini autentikasi pesan menjadi sangat penting sebagai salah satu alat pengamanan. Tetapi, tanpa beberapa mekanisme keamanan, sulit untuk mengirim data dalam cara yang aman. Salah satu layanan komunikasi yang populer adalah layanan pesan singkat (SMS). Kriptografi adalah salah satu kategori utama keamanan komputer yang mengubah informasi dari bentuk yang biasa (terbaca) kedalam bentuk yang tidak terbaca dengan menggunakan teknik enkripsi dan dekripsi. Untuk menjamin keamanan dari teks yang dikirim, banyak algoritma enkripsi yang tersedia. Pada makalah ini akan fokus membahas pada keamanan pesan dan mengamankan data teks saat proses transmisi dalam jaringan. Data yang akan ditransmisikan dari pengirim ke penerima dalam jaringan harus dienkripsi menggunakan suatu algoritma enkripsi. Algoritma kunci simetris dipilih karena bisa mengurangi masalah overhead komputasi dan perhitungan algoritma dan meningkatkan kinerja enkripsi. Makalah ini membahas mekanisme autentikasi yang lebih efisien untuk data tekstual. makalah ini akan menyajikan tinjauan teknik keamanan untuk SMS dan perbandingan kinerja algoritma enkripsi simetris yang paling umum seperti DES, 3DES, RC4, Blowfish dan AES (Rijndael).

Kata Kunci : *DES, 3DES, RC4, Blowfish, AES (Rijndael), Enkripsi, Kriptografi*

1. PENDAHULUAN

Keamanan jaringan komputer semakin berkembang akhir-akhir ini seiring dengan penggunaan akses internet yang semakin meluas. Dari beberapa kasus *hacking* atau penyadapan akses internet pada media telekomunikasi membuat banyak pengguna yang mulai sadar dan peduli dengan hak privasi mereka. Saat ini berbagai jenis alat komunikasi memberikan kemampuan pengiriman pesan atau data teks secara cepat dan sederhana. Dan fitur fundamental yang terdapat di setiap telepon selular sekarang ini salah satunya adalah *Short Message Service* (SMS). SMS merupakan salah satu fitur untuk berkirim pesan singkat antar pengguna dengan menggunakan provider selular. Meskipun banyak aplikasi *instant messaging* bermunculan, SMS masih menjadi salah satu fitur yang lebih disukai karena generalitasnya yang dapat mencakup semua jenis telepon selular. Dengan semakin banyaknya pengguna telepon selular dan banyaknya pengiriman SMS antar pengguna, sekuritas pengiriman SMS menjadi isu yang patut untuk diperhatikan. Industri SMS menjadi salah satu titik rawan untuk serangan ancaman sekuritas. Layanan SMS ini kini sudah berkembang menjadi layanan penting karena penggunaannya di bidang bisnis seperti mobile banking dan juga komunikasi kehidupan sehari-hari. SMS menjadi layanan nirkabel popular diseluruh dunia dan mayoritas pengguna mengirim dan menerima SMS yang berisi data penting atau data pribadi.

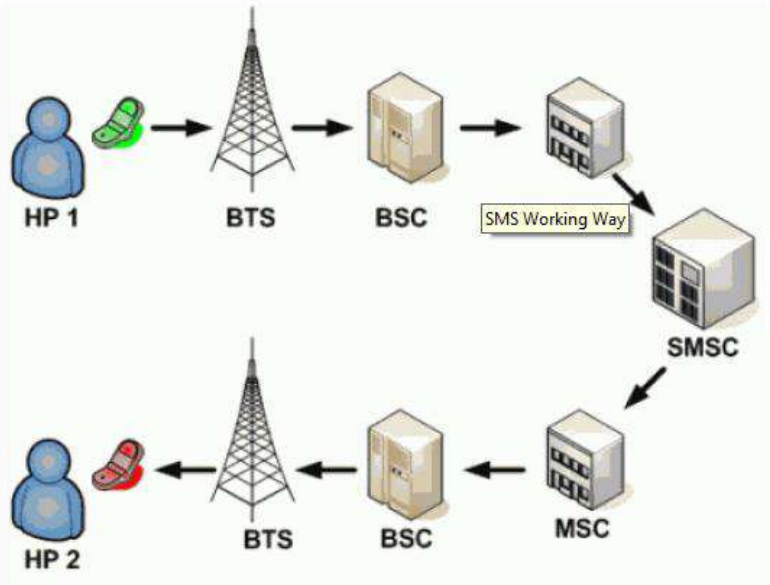
SMS bekerja dalam jaringan nirkabel. Dalam aplikasinya, pentrasmisian SMS membutuhkan beberapa komponen khusus untuk mengirimkan pesan sampai ke tujuan. Komponen yang diperlukan untuk melakukan komunikasi SMS diantaranya adalah : BTS (*Base Transceiver Station*), MSC (*Mobile Switching Center*),SMSC (*SMS Service Center*) komponen yang paling krusial adalah SMSC adalah sebuah perangkat yang terpasang pada jaringan utama SMSC ini berfungsi untuk menerima SMS dan menelusuri nomor tujuan, dan mengirimkannya ke perangkat tujuan (Telepon Seluler). SMSC ini juga berperan sebagai penyimpanan sementara untuk SMS. Jadi, jika nomor tujuan tersebut tidak aktif, SMS tersebut akan tersimpan pada SMSC dan SMSC akan mengirimkannya kembali jika perangkat tujuan telah aktif kembali. Sebagai tambahan, SMSC akan memberikan notifikasi kepada pengirim apakah pengiriman SMS tersebut berhasil ataupun tidak. Namun, karena keterbatasan memori penyimpanan, SMSC tidak dapat menyimpan SMS untuk jangka waktu yang lama.

Kenyataannya SMS memiliki beberapa masalah seperti lebih rentan terhadap hacking, penyadapan dan pencurian akses lainnya[3]. Celah keamanan terbesar pada pengiriman SMS adalah pada saat SMS tersebut tersimpan pada SMSC. Sehingga, jika dilakukan serangan pada server SMSC, akan mengakibatkan pesan yang dikirim akan dapat dibaca orang lain yang tidak berhak. Komunikasi yang tidak dilindungi menimbulkan kerentanan terhadap keamanan yang cukup serius karena dibeberapa kasus komunikasi menggunakan sms berisi data yang bersifat rahasia. Salah satu metode yang cukup efisien adalah penggunaan autentikasi untuk melindungi dari serangan yang tidak diinginkan selama proses transmisi pesan. Tujuan utama dari mekanisme keamanan adalah untuk memberikan privasi pesan yang menjamin kerahasiaan, integritas dan tidak ada pengulangan data[12]. Fitur utama dari keamanan jaringan itu sendiri adalah menyediakan autentikasi yang efisien dengan menggunakan teknik kriptografi yaitu dengan melakukan enkripsi pada pesan yang akan dikirimkan. Dengan melakukan enkripsi, maka pesan yang terbajak dari SMSC tidak dapat diterjemahkan langsung oleh pelaku.

Teknik kriptografi itu sendiri ada 2 jenis berdasarkan perbedaan teknik enkripsinya yaitu enkripsi asimetris dan enkripsi simetris. Enkripsi asimetris biasa disebut kunci enkripsi publik dimana ada 2 kunci yang digunakan untuk proses enkripsi yaitu kunci publik dan kunci privat. Sedangkan enkripsi simetris hanya menggunakan 1 kunci untuk enkripsi dan dekripsi. Pada makalah ini lebih fokus pada penggunaan algoritma kunci simetris yang dapat mengurangi masalah *overhead* komputasi dan perhitungan algoritma dan meningkatkan kinerja enkripsi. Algoritma enkripsi simetris juga ada beberapa jenis diantaranya DES, 3DES, RC4, Blowfish and AES (Rijndael). Dan makalah ini akan membandingkan kinerja masing-masing algoritma simetris tersebut.

2. SMS (*SHORT MESSAGING SERVICE)*

Short Messaging Service (SMS) merupakan salah satu fitur dari GSM yang dikembangkan dan distandarisasi oleh ETSI. Pada saat kita mengirim pesan SMS dari handphone, maka pesan SMS tersebut tidak langsung dikirim ke handphone tujuan, akan tetapi terlebih dahulu dikirim ke SMS Center (SMSC) dengan prinsip *Store and Forward*, setelah itu baru dikirimkan ke handphone yang dituju. Proses pengiriman SMS dapat dilihat pada gambar 1.



Gambar 1. Skema Proses SMS

SMS bekerja dalam jaringan nirkabel. Dalam aplikasinya, pentrasmisian SMS membutuhkan beberapa komponen khusus untuk mengirimkan pesan sampai ke tujuan. Komponen yang diperlukan untuk melakukan komunikasi SMS diantaranya adalah :

a. BTS (*Base Transceiver Station*)

   BTS ini merupakan sebuah perangkat yang memfasilitasi komunikasi nirkabel antara perangkat user dengan jaringan. Perangkat user ini dapat meliputi telepon selular, komputer dengan koneksi internet nirkabel, dan lain-lain.

b. MSC (*Mobile Switching Center*)

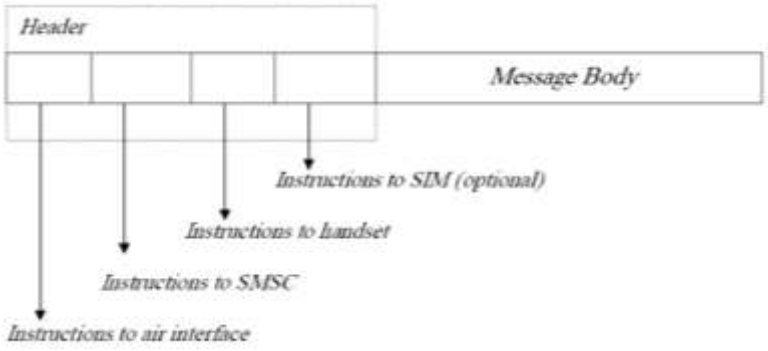
   MSC ini adalah sebuah noda layanan pengiriman utama bagi GSM/CDMA. Perangkat ini berfungsi untuk *routing* panggilan suara, SMS, FAX, maupun *conference call*.

c. SMSC (*SMS Service Center*)

   SMSC adalah sebuah perangkat yang terpasang pada jaringan utama SMSC ini berfungsi untuk menerima SMS dan menelusuri nomor tujuan, dan mengirimkannya ke perangkat tujuan (Telepon Seluler). SMSC ini juga berperan Makalah IF2120 Matematika Diskrit – Sem. I Tahun 2013/2014.

2.1 STRUKTUR PESAN SMS

Struktur pesan dalam sebuah paket SMS dapat dilihat pada gambar dibawah berikut :



Gambar 2. Struktur Pesan SMS[13]

Pada Gambar 2 dapat terlihat bahwa pada sebuah paket pesan SMS terdiri dari header dan body. Header pesan terdiri dari instruksi-instruksi kepada komponen-komponen yang bekerja dalam jaringan SMS. Pada instruksi-instruksi tersebut, terdapat informasi yang diperlukan selama pengiriman pesan seperti informasi validitas pesan, dan informasi informasi lainnya. Pada bagian message body, terdapat isi dari pengirim pesan yang akan dikirimkan. Panjang isi pesan pada sebuah paket SMS berukuran maksimal 160 karakter, dimana setiap karakter memiliki panjang 7 bit. Beberapa aplikasi standar telepon selular dapat mendukung panjang pesan dengan karakter sepanjang 8 bit (panjang pesan maksimum 140 karakter) dan karakter yang lebih panjang lainnya seperti 16 bit, namun karakter sepanjang 8 bit dan 16 bit ini tidak didukung oleh semua aplikasi standar telepon selular. Pada umumnya karakter sepanjang 8 bit dan 7 bit digunakan untuk menampilkan data seperti gambar dan simbol. JAVA ME (*JAVA Mobile Edition*) adalah salah satu jenis bahasa pemrograman JAVA yang diperuntukkan untuk pengembangan aplikasi java agar dapat berjalan pada perangkat seluler yang memiliki keterbatasan memori dan tampilan (Johannes, 2010). Dengan menggunakan JAVA ME, dapat dikembangkan aplikasi SMS yang memiliki banyak fitur dibandingkan dengan aplikasi SMS standar telepon seluler. Struktur pesan dalam bahasa JAVA ME dikenal ada dua jenis yang diturunkan dari *interface Message*, yaitu *interface TextMessage* dan *BinaryMessage*. *Interface TextMessage* berfungsi untuk mengirim dan menerima pesan dalam bentuk teks sama seperti fungsi aplikasi SMS standar ponsel. Sedangkan *interface BinaryMessage* berfungsi untuk mengirim dan menerima pesan dalam bentuk *binary*. Pengiriman pesan SMS umumnya hanya dapat dilakukan satu kali oleh sebuah telepon seluler. Namun seiring dengan kemajuan teknologi, beberapa telepon seluler mampu mengirimkan beberapa paket SMS dalam satu pesan. Yang dilakukan oleh telepon seluler agar terlihat dapat mengirim beberapa paket SMS dalam satu pesan adalah dengan melakukan

konkatinasi [2] Dengan menggunakan fitur ini, seolah-olah pengguna telepon seluler dapat mengirim paket SMS lebih dari 160 karakter untuk satu buah pesan. Namun yang sebenarnya dilakukan oleh telepon seluler adalah mengirimkan paket-paket SMS tersebut lebih dari satu kali dan kemudian paket-paket SMS tersebut disatukan agar menjadi satu buah pesan. Proses penyambungan beberapa pesan agar menjadi satu buah pesan memerlukan informasi tambahan, oleh karena itu panjang satu buah pesan tersebut akan menjadi lebih kecil[8].

Selain proses pengiriman pesan, sebuah aplikasi SMS juga harus memiliki proses penerimaan pesan. Untuk membuat aplikasi SMS pada sebuah telepon seluler yang tentu saja juga sudah terinstal aplikasi SMS, pengembang harus mendefinisikan *port* aplikasi SMS yang akan dibangun. Fungsi pendefinisian nomor *port* ini adalah agar pesan yang akan dikirim sampai pada aplikasi SMS yang dibangun dan bukan pada aplikasi SMS standar ponsel yang memiliki nomor *port* 0. Penggunaan nomor *port* bergantung pada jenis aplikasi SMS yang hendak dibangun. Jika aplikasi SMS yang hendak dibangun adalah aplikasi SMS yang berfungsi mengirimkan pesan pada waktu tertentu, maka tidak perlu mendefinisikan nomor *port* pada aplikasi SMS. Hal ini dikarenakan pesan memang ditujukan untuk masuk pada aplikasi SMS standar ponsel. Namun jika aplikasi SMS yang akan dibangun memiliki fungsi khusus yang tidak dimiliki aplikasi SMS standar ponsel, maka perlu dilakukan pendefinisian nomor *port*. Informasi nomor *port* yang telah didefinisikan akan dibawa bersama paket pesan yang dikirim oleh pengirim. Sehingga panjang maksimal paket pesan akan berkurang untuk menampung informasi nomor *port* [2].

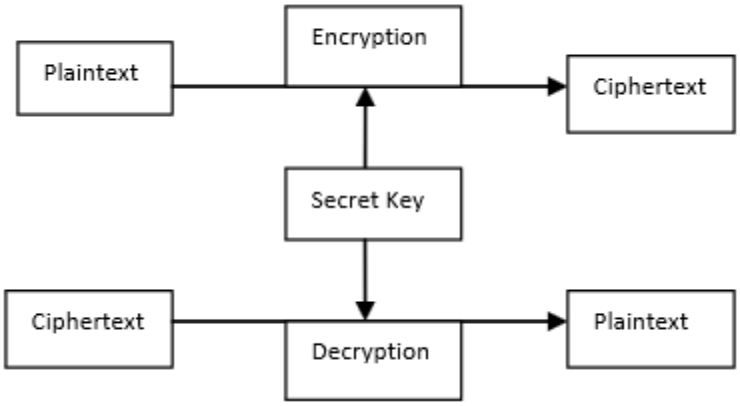
3. KRIPTOGRAFI

Kriptografi adalah suatu ilmu pengetahuan yang mempelajari teknik-teknik yang berkaitan dengan keamanan informasi, teknik-teknik yang digunakan pada umumnya menggunakan dasar pengetahuan matematika[1]. Kriptografi bukanlah satu-satunya jalan dalam menjaga keamanan dokumen tetai kriptografi menyediakan kumpulan teknik untuk menjaga dokumen.

Secara garis besar kriptografi dibagi menjadi 2 jenis, kriptografi klasik dan kriptografi modern. Perbedaan mendasar yang terdapat pada dua jenis tersebut adalah pada kriptografi modern algoritmanya beroperasi pada mode bit, sedangkan kriptografi klasik beroperasi pada mode karakter. Teknik kriptografi modern dibagi menjadi 2 jenis secara umum, yaitu algoritma kunci simetris dan asimetris.

a. Algoritma Kriptografi Kunci Simetris

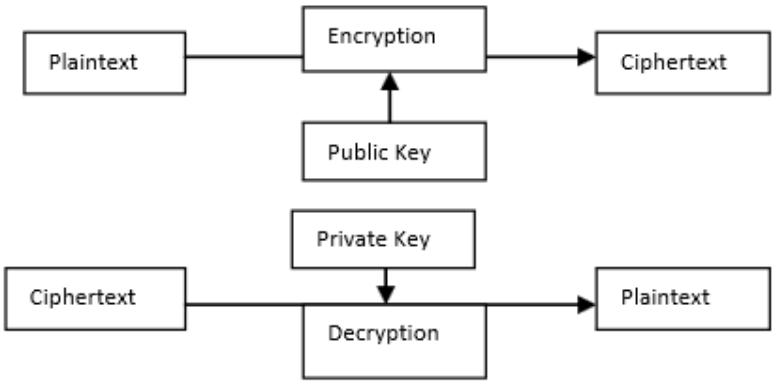
Pada algoritma ini, kunci yang digunakan dalam proses dekripsi dan enskripsi merupakan kunci yang sama. Berdasarkan pemrosesan bit, algoritma kunci simetris dibagi menjadi dua bagian, yaitu: algoritma block chipper yang melakukan pemrosesan bit per-blok dan algoritma stream chipper yang memproses blok secara mengalir atau per-bit.



Gambar 3. Algoritma Kunci Simetris[1]

b. Algoritma Kriptografi Kunci Asimetris

Proses enkripsi dan dekripsi pada algoritma kriptografi kunci asimetris menggunakan kunci yang berbeda. Algoritma ini menggunakan kunci enkripsi yang bersifat public atau tidak rahasia, namn menggunakan kunci dekripsi yang bersifat privat atau rahasia. Kunci dekripsi pada umumnya merupakan perhitungan dari kunci enkripsi yang bukan merupakan pemetaan satu ke satu, sebuah kunci dekripsi dapat memiliki beberapa kunci enkripsi. Dalam penggunaanya, algoritma kriptografi kunci publik tidak hanya digunakan untuk menyembunyikan pesan, tetapi dapat juga digunakan untuk melakukan autentikasi dokumen.



Gambar 4. Algoritma Kunci Asimetris[1]

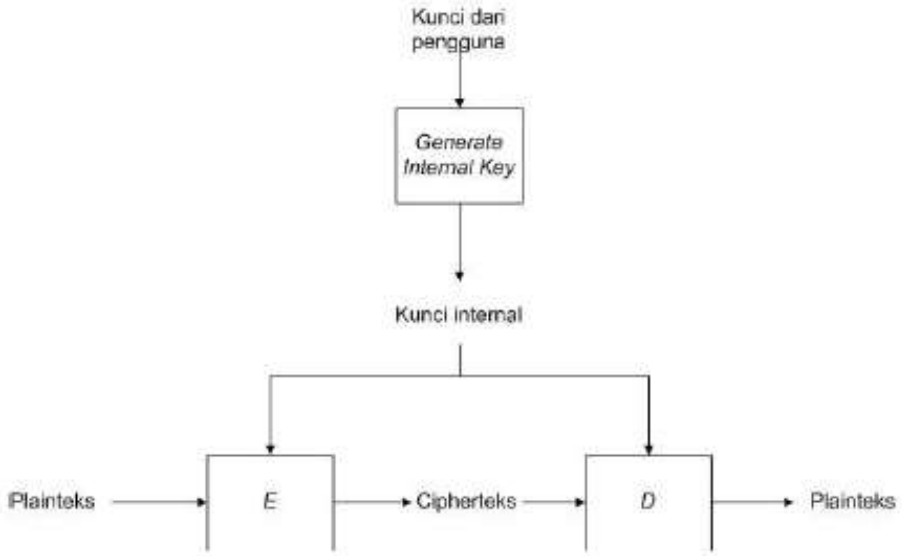
Tujuan kriptografi adalah untuk mencegah dan mendeteksi orang yang tidak bertanggung jawab melakukan hal-hal yang menggangu seperti membaca data rahasia atau mengubah suatu data penting. Untuk tujuan ini, kriptografi menyediakan empat aspek keamanan yaitu :

1. Kerahasiaan (*confidentiality*)
   Layanan yang digunakan untuk menjaga isi pesan dari siapapun yang tidak memiliki hak untuk membacanya.
2. Integritas data (*data integrity*)
   Layanan yang menjamin bahwa pesan masih asli/utuh atau belum pernah dimanipulasi selama pengiriman.
3. Otentikasi (*authentication*)
   Layanan untuk mengidentifikasi kebenaran pihak-pihak yang berkomunikasi (*user authentication*) dan untuk mengidentifikasi kebenaran sumber pesan (*data origin authentication*).
4. Nirpenyangkalan (*non-repudiation*)
   Layanan untuk mencegah pihak yang berkomunikasi melakukan penyangkalan, yaitu pengirim pesan menyangkal melakukan pengiriman atau penerima pesan menyangkal telah menerima pesan.

Algoritma kriptografi melibatkan proses pengubahan pesan menjadi tersembunyi atau tidak dikenali isi dan maksudnya. Pesan yang belum diubah tersebut disebut dengan plainteks dan pesan yang telah diubah dengan cipherteks. Proses pengubahan plainteks menjadi cipherteks disebut denga enkripsi dan proses pengembalian cipherteks menjadi plainteks disebut dengan dekripsi.

3.1 BLOCK CHIPPER

Block cipher adalah suatu tipe algoritma kriptografi kunci simteris yang mengubah plainteks yang dibagi dalam blok-blok dengan panjang yang sama menjadi cipherteks yang meiliki panjang blok yang sama. Ukuran panjang blok dapat beragam bergantung kepada algoritma yang digunakan, ukuran yang sering digunakan adalah 64 bit dan menuju 128 bit. Seperti seua akgoritma kunci simetri, proses enkripsi yang dilakukan akan menggunakan suatu input dari user yang disebut sebagai kunci rahasia. Kunci rahasia ini juga akan dipakai ketika melakukan proses dekripsi. Cara kerja secara umum dari block cipher dapat dilihat pada gambar 5.



Gambar 5. Skema Cara Kerja Blok Cipher[1]

Dalam penggunaannya block cipher dikombinasikan dengan suatu teknik yang dinamakan mode operasi dari block cipher. Mode operasi yang sederhana dan sering digunakan adalah mode Electronic Code Book(ECB). Pada mode ECB setiap blok pada plainteks dienrkripsi satu persatu secara independen. Hasil enkripsi masing-masing blok tidak mempengaruhi blok yang lain. Prses enkripsi pada mode ini sangat sederhana, setiap blok plainteks dienkripsi dengan fungsi enkripsi secara terpisah. Seperti halnya dalam proses enkripsi, dalam proses dekripsi, masingmasing blok-blok cipherteks dikenakan dengan fungsi dekripsi secara independen.

Dalam melakukan perancangan block cipher, beberapa prinsip harus dipertimbangkan. Prinsip-prinsip tersebut yaitu:

1. Prinsip Confusion dan Diffusion dari Shannon.

   Tujuan dari prinsip confusion adalah untuk menyembunyikan hubungan apapun yang ada antara plainteks, cipherteks, dank unci. Sehingga dapat membuat kripanalis kesulitan dalam menemukan pola-pola pada cipherteks. Tujuan dari prinsip diffusion adalah menyebarkan pengaruh satu bit plainteks atau kunci ke sebanyak mungkin cipherteks, sehingga dengan berubahnya satu bit plainteks dapat mengubah cipherteks yang sulit untuk diprediksi.
2. Iterated Cipher

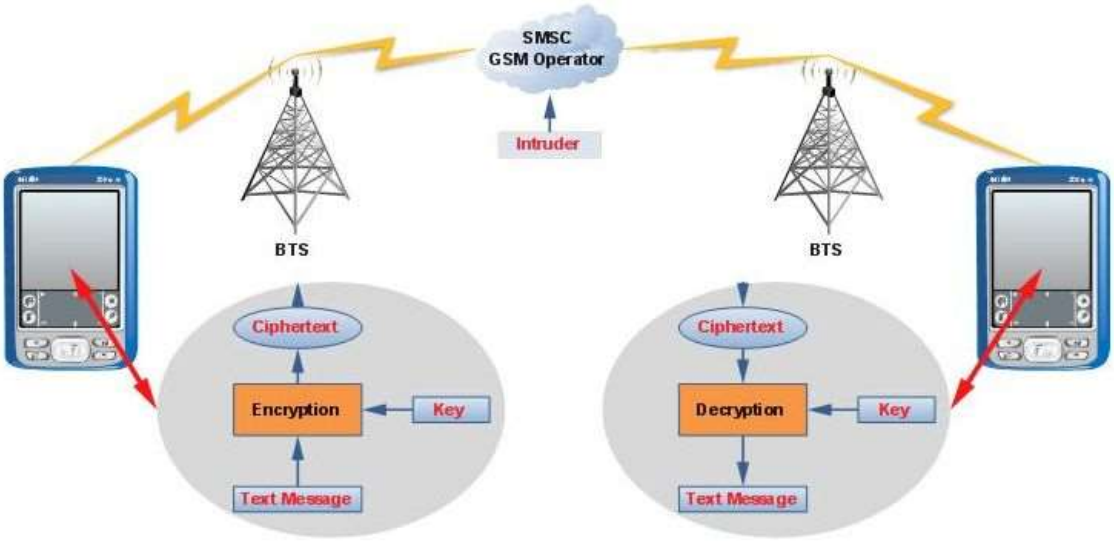
   Untuk menambah keamanan, pada algoritma-algoritma block cipher dilakukan iterasi pada pemrosesan setiap blok, pada setiap rotasi dari iterasi tersebut digunakan fungsi transformasi yang sama namun memakai kunci yang berbeda yang disebut dengan kunci internal. Kunci internal pada umumnya merupakan hasil dari kunci yang dimasukkan oleh pengguna yang dikomputasi menggunakan suatu fungsi tertentu. Dengan adanya iterasi tersebut keamanan akan semakin terjamin, namun performansi akan berkurang karena adanya waktu lebih yang dibutuhkan untuk melakukan iterasi. Block cipher yang menerapkan konsep iterasi ini disebut juga dengan iterated block cipher.
3. Kunci Lemah

   Suatu hal yang perlu dihindari dalam melakukan perancangan algoritma kriptografi adalah kunci yang dapat menghasilkan cipherteks yang mirip atau serupa dengan plainteks.

4. ENKRIPSI PADA SMS

Pada jaringan selular lebar *bandwith* sangat terbatas, sehingga algoritma enkripsi selain harus memenuhi standar keamanan, juga harus menjaga agar hasil file enkripsi tetap kecil. Bagian yang dapat dimanipulasi dari pesan SMS hanyalah bagian payload, yang artinya dengan dilakukan enkripsi akan terjadi penambahan byte yang mengurangi kapasitas payload untuk pesan yang belum terenkripsi (plainteks).

Oleh karena keterbatasan tersebut, solusi yang ditawarkan untuk enkripsi SMS adalah kerahasiaan algoritmanya, yang ditawarkan oleh penyedia layanannya atau juga yang disediakan oleh perangkat *mobile* tertentu.



Gambar 6. Proses Enkripsi pada SMS

Enkripsi SMS itu adalah proses keamanan dengan pengkodean pesan untuk membuat mereka nonreadable. Langkah-langkah dapat digambarkan sebagai berikut:

Step 1: Get secrete message.
Step 2: Determine the message recipient.
Step 3: Compress the secrete message.
Step 4: Check the compressed message length.
Step 5: Encrypt the compressed message using encryption algorithm.
Step 6: Add signature to the message.
Step 7: Send the secrete message.

Gambar 7. Langkah Enkripsi SMS

5. Review Algoritma

Pada makalah ini, berbagai jenis algoritma simetris telah dievaluasi. Dan untuk menerapkan algoritma yang tepat dalam aplikasi sms diperlukan mengetahui kekuatan dan keterbatasan masin-masing algoritma. Penilaian algoritma yang ada dinilai dari beberapa parameter tertentu yang diperlukan. Parameter mungkin termasuk adalah sebagai berikut :

- Arsitektur

  Mendefinisikan struktur dan operasi yang diterapkan pada algoritma, karakteristik dan bagaimana mereka diimplementasikan.

- Keamanan

  Ukuran afirmatif dari kekuatan sistem dalam menolak serangan adalah elemen yang diinginkan dari setiap algoritma enkripsi memiliki distinguishability (dibangun dengan menggabungkan substitusi dengan transposisi berulang kali). Keamanan algoritma enkripsi tergantung pada ukuran utama yang digunakan untuk menjalankan enkripsi: umumnya, semakin besar ukuran kunci maka enkripsi semakin kuat. Panjang kunci diukur dalam bit.
- Fleksibilitas

  Mendefinisikan apakah algoritma ini mampu dimodifikasi sesuai dengan kebutuhan.
- Skalabilitas

  Ini adalah salah satu elemen utama yang dianalisa dari suatu algoritma enkripsi. Skalabilitas tergantung pada parameter tertentu seperti Memory Usage, tingkat enkripsi, kinerja hardware Software, efisiensi komputasi.
- Keterbatasan (Serangan Dikenal)

  Mendefinisikan seberapa baik algoritma bekerja dengan menggunakansumber daya komputer yang ada dan seberapa rentan terhadap berbagai jenis serangan.

5.1 ALGORITMA DES

DES (Data Encryption Standard) adalah algoritma *cipher* blok yang popular karena dijadikan standard algoritma enkripsi kunci-simetri, meskipun saat ini standard tersebut telah digantikan dengan algoritma yang baru, *AES,* karena *DES* sudah dianggap tidak aman lagi. Sebenarnya *DES* adalah nama standard enkripsi simetri, nama algoritma enkripsinya sendiri adalah *DEA (Data Encryption Algorithm),* namun nama *DES* lebih populer daripada *DEA.* Algoritma *DES* dikembangkan di *IBM* dibawah kepemimpinan W.L. Tuchman pada tahun 1972. Algoritma ini didasarkan pada algoritma *Lucifer* yang dibuat oleh Horst Feistel. Algoritma ini telah disetujui oleh *National Bureau of Standard (NBS)* setelah penilaian kekuatannya oleh *National Security Agency (NSA)* Amerika Serikat. DES adalah blok cipher yang menggunakan 64 bit teks biasa dengan 16 putaran dan Panjang Kunci 56-bit. Sebenarnya kuncinya adalah 64 bit (sama dengan ukuran blok), tetapi dalam setiap byte ada 1 bit di yang dipilih sebagai '*parity*' bit, dan tidak digunakan untuk mekanisme enkripsi. 56 bit dipermutasi menjadi 16 sub kunci masing-masing dengan panjang 48-bit. Algoritma yang sama digunakan dengan urutan terbalik untuk dekripsi.

Kekuatan keamanan DES tergantung pada 56 bit ukuran kunci yang menghasilkan 7,2 x 1016 kemungkinan kunci. Jika kunci sering berubah, resiko perhitungan yang tidak sah atau akuisisi dapat sangat dimoderasi. Selain itu DES menunjukkan efek *avalanche* yang kuat yaitu modifikasi miniatur dalam plaintext atau kunci. Awalnya DES dianggap aman dan sulit untuk pecahkan. Serangan brute-force menjadi subyek spekulasi pertama setelah algoritma rilis dalam domain publik, meskipun DES bertahan secara linear dan diferensial serangan yang berbeda tetapi pada tahun 1998 Electronic Frontier Foundation (EFF) merancang mesin tujuan khusus untuk "mendekripsi DES". Dalam satu demonstrasi berhasil mendapatkan kunci dari pesan terenkripsi dalam waktu kurang dari satu hari dalam kombinasi dengan aliansi pengguna komputer di seluruh dunia. Secara umum DES terbukti aman untuk perusahaan besar atau pemerintah dan lebih sederhana untuk tidak menggunakan algoritma DES. Namun untuk kompatibilitas, dan biaya upgrade, DES tetap harus diutamakan. Struktur DES tidak mendukung modifikasi. DES adalah sangat rentan terhadap serangan kriptanalisis linear, kunci lemah juga merupakan masalah besar. DES juga terkena serangan brute force.

5.2 TRIPLE-DES

DES digantikan oleh triple-DES (3DES) pada bulan November 1998, berkonsentrasi pada ketidaksempurnaan dalam DES dan tanpa mengubah struktur asli dari algoritma DES. TDES adalah versi yang jauh lebih rumit dari DES mencapai tingkat keamanan yang tinggi dengan mengenkripsi data menggunakan DES tiga kali menggunakan dengan tiga kunci DES yang berbeda. 3DES masih disetujui untuk digunakan oleh sistem pemerintahan US, tetapi telah digantikan oleh AES.

Sama seperti namanya 3DES melakukan 3 iterasi dari enkripsi DES pada setiap blok. Karena merupakan versi yang disempurnakan dari DES sehingga didasarkan pada konsep Struktur Feistel. 3DES menggunakan 64 bit teks biasa dengan 48 putaran dan Panjang Kunci dari 168-bit yang dipermutasi menjadi 16 sub kunci masing-masing panjangnya 48-bit.

TDES adalah versi yang disempurnakan dari DES, 3DES menggunakan ukuran yang lebih besar dari kunci (yaitu 168-bit) untuk mengenkripsi dibandingkan DES. Operasi DES (mengenkripsi-decrypt-encrypt) dilakukan 3 kali di 3DES dengan 2-3 kunci yang berbeda, menawarkan "112 bit keamanan", dapat menghindari serangan -meet-in-the-middle. TDES menawarkan tingkat keamanan yang tinggi dibandingkan DES. Struktur 3DES sama seperti DES, tidak mendukung modifikasi tetapi sebagai iterasi DES 3 kali sehingga ukuran kunci

diperluas ke 168 bit. 3 DES terkena diferensial dan serangan related-key. Juga rentan terhadap variasi tertentu seperti serangan meet-in-the-middle.

5.3 BLOWFISH

Blowfish dianggap sebagai algoritma enkripsi yang sangat kuat dari segi keamanan oleh Bruce Schneier, penulis Applied Cryptography, dengan struktur dan Fungsi yang berbeda dari algoritma enkripsi yang telah disebutkan lainnya. Blowfish adalah algoritma enkripsi blok cepat, padat, dan sederhana dengan panjang kunci variabel yang memungkinkan pertukaran antara kecepatan dan keamanan. Blowfish merupakan algoritma domain publik (unpatented) dan digunakan dalam SSL atau program lain. Blowfish juga merupakan kunci algoritma simetris yang terdiri dari 2 bagian: bagian ekspansi kunci dan bagian data enkripsi. Blowfish adalah blok cipher yang menggunakan 64 bit teks biasa dengan 16 putaran, sehingga panjang kunci variabel, hingga 448 bit, dipermutasi ke 18 sub kunci masing-masing 32 bit dan dapat diimplementasikan pada 32 atau 64-bit prosesor. Keamanan Blowfish terletak pada ukurannya kunci yang variabel(128-448 bit) memberikan tingkat keamanan yang tinggi, Upaya untuk pembacaan sandi Blowfish dimulai segera setelah publikasi namun upaya kriptanalisis kurang dibuat pada Blowfish daripada algoritma lainnya. Blowfish kebal terhadap serangan diferensial related-key, karena setiap bit dari kunci master melibatkan banyak kunci putaran yang sangat independen, membuat serangan tersebut sangat rumit atau tidak layak. Otonomi seperti ini sangat patut ditiru. Algoritma blowfish sangat fleksibel dan dapat di modifikasi dengan panjang kunci kelipatan dari 32 bit. Blowfish memiliki beberapa kelas kunci yang lemah. 4 putaran blowfish terkena serangan kedua diferensial orde. Jadi, keandalan Blowfish dipertanyakan karena besarnya jumlah kunci lemah.

5.4 RC4

RC4 adalah stream cipher yang dirancang pada tahun 1987 oleh Ron Rivest untuk *RSA Security*. Ini adalah variabel kunci ukuran *stream cipher*. Algoritma ini didasarkan pada permutasi acak. Di bulan september 1994, algoritma RC4 yang anonim diposting di Internet. Algoritma ini mudah untuk dijelaskan. Variabel panjang kunci adalah dari 1 sampai 256. Hal ini disebut pseudo random karena RC4 menghasilkan urutan angka yang hanya mendekati sifat-sifat nomor acak. Urutan byte yang dihasilkan tidak acak karena output selalu sama untuk setiap masukan yang diberikan. Dalam stream cipher, data dienkripsi dan didekripsi sedikit demi sedikit. Hal ini juga lebih cepat dan lebih cocok untuk aplikasi streaming.

Sehingga membutuhkan waktu yang sedikit pada CPU, pemanfaatan sumber daya yang lebih kecil dan juga mudah diimplementasikan tetapi RC4 memiliki banyak kelemahan karena ukuran kunci yang kecil. Jika ukuran kunci pendek penyerang dapat dengan mudah memperoleh kunci dan melakukan serangan terhadap data menggunakan algoritma pemulihan kunci.

5.5 AES *(RIJNDAEL)*

AES termasuk blok chipper simteris yang pertama kali diperkenalkan ole NIST . memiliki panjang kunci variabel dengan ukuran 128, 192, dan 256 bit. Memiliki ukuran blok 128 bit di 10, 12 dan 14 putaran tergantung pada ukuran kunci. Hal ini dapat diterapkan pada berbagai platform dan sudah diuji dengan baik untuk banyak keamanan aplikasi. berikt langkah-langkah yang digunakan untuk mengenkripsi blok 128-bit:

1. Mendapatkan satu set kunci putaran dari kunci cipher.
2. Inisialisasi state array dengan pesan asli (Plaintext).
3. tambahkan inisialisai kunci putaran untuk memulai state array.
4. Melakukan sembilan putaran manipulasi state.
5. Lakukan putaran kesepuluh dan manipulasi state terakhir.
6. Salin array keadaan akhir sebagai data *non readable* (Teks Cipher).

Proses enkripsi terdiri dari serangkaian langkah untuk mengubah state array. Ada empat langkah yang dilibatkan dalam enkripsi. Diantaranya adalah sebagai berikut:

a. Sub Bytes: Operasi ini adalah substitusi sederhana yang dalam hal ini mengubah setiap bagian menjadi nilai yang berbeda. Dalam 128-bit blok diganti dengan yang blok 128-bit lain untuk substitusi proses.
b. Shift Rows: Setiap baris diputar ke kanan sebanyak nomor tertentu dalam byte.
c. Mix Columns: Array diproses secara terpisah untuk masing-masing kolom state untuk menghasilkan kolom baru. Kolom baru menggantikan kolom lama.
d. XorRoundKey: Operasi ini hanya menangkap b yang ada pada state array.

Proses dekripsi diperoleh dengan melakukan kembali semua langkah yang diambil di enkripsi menggunakan fungsi inverse seperti InvSubBytes, InvShiftRows, dan InvMixColumns. AES memiliki fleksibilitas yang besar dengan perlawanan yang efektif terhadap serangan kriptoanalisis. Keamanan AES tergantung pada variabel ukuran kunci

yang memungkinkan sampai dengan 256 bit untuk memberikan perlawanan terhadap serangan. Dari semua blok enkripsi cipher, AES lebih cepat dan fleksibel dibandingkan dengan enkripsi lainnya. Dalam hal pelaksanaan hardware AES sangat cepat dibandingkan dengan enkripsi lainnya.

5.6 MODIFIKASI AES

Untuk mengurangi masalah komputasi tinggi dan overhead maka menggunakan AES yang dimodifikasi. AES juga dikenal dengan modifikasi AES karena dengan total langkah permutasi awal mengambil dari DES untuk memperbesar kinerja enkripsi. 4 langkah modifikasi AES adalah sbb:

1. Substitution Byte
2. ShiftRow
3. Permutasi
4. AddRoundKey

Di sini kita menggunakan Permutasi sedangkan MixColumn Substitution Byte, ShiftRow, dan AddRoundKey tetap sama seperti pada AES. Permutasi banyak digunakan dalam algoritma kriptografi. Operasi permutasi menjadi menarik dan penting jika dilihat dari sudut pandang kriptografi dan arsitektur. Algoritma DES akan memberikan kami tabel permutasi. Input ke Tabel IP terdiri dari 128 bit. Algoritma AES-dimodifikasi mengambil 128 bit sebagai input. Fungsi Substitution Bytes dan ShiftRow juga diartikan sebagai 128 bit sedangkan fungsi Permutasi juga membutuhkan 128 bit. Dalam tabel permutasi masing-masing entri menunjukkan posisi tertentu dari input bit nomor yang mungkin juga terdiri dari 256 bit pada output. Saat membaca tabel dari kiri ke kanan dan kemudian dari atas ke bawah, kita mengamati bahwa bit ke 242 dari blok 256-bit berada di posisi pertama, 226 berada di posisi kedua dan seterusnyaa. Setelah menerapkan permutasi pada 128 bit kita set lagi dengan melengkapi 128 bit dan kemudian melakukan fungsi yang algoritma berikutnya. Jika kita mengambil permutasi terbalik untuk mengembalikan lagi bit asli, Hasil outputnya adalah 128-bit cipher teks. Untuk dekripsi penuh modifikasi AES algoritma proses transformasi nya adalah Inv-Bytesub, inv-ShiftRows, inv-Permutasi, dan AddRoundKey, yang dilakukan di 10 putaran seperti dalam Proses enkripsi. Berdasarkan analisis menunjukkan bahwa algoritma ini lebih baik menggunakan kinerja metrik.

1. Waktu Enkripsi

Ini adalah waktu yang algoritma enkripsi yang diperlukan untuk menghasilkan cipher teks dari plaintext a.

2. Waktu Dekripsi

   Ini adalah waktu yang diperlukan algoritma dekripsi untuk menghasilkan plaintext dari ciphertext.

3. Throughput

   Throughput mendefinisikan kecepatan enkripsi. Throughput dihitung sebagai total plaintext terenkripsi dalam KiloByte / waktu enkripsi (KB / sec). Konsumsi Power menurun maka throughput yang meningkat.

4. Waktu proses CPU

   Ini adalah waktu yang CPU didedikasikan hanya untuk proses tertentu untuk melakukan perhitungan. Hal ini mencerminkan beban dari CPU. Semakin banyak waktu CPU yang digunakan dalam proses enkripsi maka beban lebih tinggi.

5. Penggunaan Memory

   Penggunaan memori mendefinisikan bagaimana memori dikonsumsi pada saat enkripsi dan dekripsi.

Metrik kinerja di atas didefinisikan dengan menggunakan tugas berikut.

a. Hitung waktu enkripsi dan dekripsi untuk setiap algoritma menggunakan file input yang berbeda ukuran.
b. Hitung throughput untuk setiap algoritma dalam KB / Sec.
c. Sifat mengubah ukuran kunci pada waktu enkripsi / dekripsi.
d. Hasil dari perubahan ukuran file pada pemanfaatan memori.
e. Hitung waktu CPU untuk enkripsi dan dekripsi untuk setiap algoritma menggunakan file input yang berbeda ukuran.

6. ANALISA PERBANDINGAN

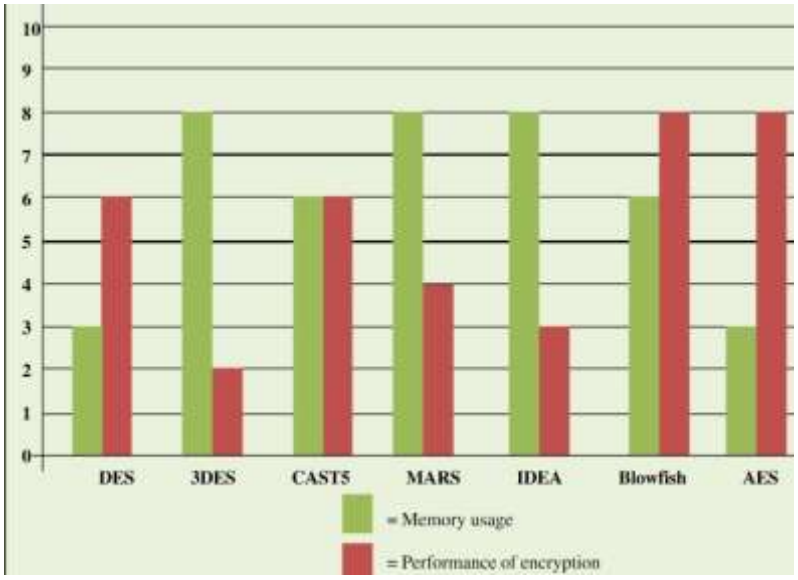
Setelah menganalisisa hasil dari beberapa makalah mengenai algoritma enkripsi simetris yang mengemukakan hasil diskusi terkait topoik diatas. Didapatkan Perbandingan yang dilakukan berdasarkan pada konsumsi daya, waktu pemrosesan dan throughput. Dimana throughput meningkatkan sedangkan konsumsi daya menurun.

Faktor Analisa	DES	RC4	Blowfish	AES
Panjang Kunci (bit)	64	40 - 128	1288	128,192,256
Ukuran Blok (bit)	64	32,64,128	64	128
Rasio Enkripsi	Rendah	Rendah	Tinggi	Tinggi
Ketahanan Serangan	Tidak Memadai	Kurang Aman	Kurang Aman	Sangat Aman
Fitur	Sangat umum, tidak cukup kuat	Modifikasi dari DES, keamanan memadai	Keamanan baik	Pengganti DES, keamanan baik
Konsumsi Daya	Tinggi	Tinggi	Rendah	Rendah
Implementasi Hardware dan software	lebih baik di hardware dari pada di software	tidak terlalu efisien	tidak terlalu efisien	lebih cepat dan efisien

Tabel 1. Tabel Perbandingan Algoritma Enkripsi DES, RC4, Blowfish AES

Pada makalah ini di dapatkan hasil survey dari teknik enkripsi yang terdahulu sampai yang terbaru. Dan menunjukkan bahwa algoritma AES sangat aman, cepat dan tahan terhadap serangan untuk komunikasi *mobile*. Algoritma AES mampu menyelesaikan masalah utama dalam keamanan pada komunikasi *mobile* dan enkripsi SMS di jaringan. Diagram dibawah ini menunjukkan perbandingan antara berbagai algoritma enkripsi berdasarkan skalabilitas seperti penggunaan *memory* dan performa enkripsi. Dan dari gambar tersebut dapat dianalisa bahwa AES adalah algoritma enkripsi yang terbaik di bandingkan algoritma enkripsi lainnya.

Blowfish dan AES memiliki performansi enkripsi yang sama tetapi penggunaan memory yang berbeda.



Gambar 8. Grafik Perbandingan Performa Algoritma Simetris [2]

7. KESIMPULAN

Makalah ini membahas mengenai berbagai mekanisme keamanan untuk mendukung autentikasi pesan. Makalah ini menyajikan beberapa algoritma keamanan seperti DES, 3DES, BLOWFISH, RC4 dan AES secara detail. DES dan 3DES tidak dapat menambah keamanan dan menggunakan dua atau 3 enkripsi yang sangat lambat jika di implementasikan pada software. DES memliki ukuran data yang besar dan panjang kunci yang terbatas. Sedangkan Blowfish dan RC4 bermasalah pada kunci enkripsi yang lemah. Setelah dianalisa dari semua algoritma diatas, algoritma simetris yang paling pouler adalah AES(Rijndael) dimana merupakan algoritma enkripsi yang paling fleksibel dan memiliki performa enkripsi terbaik, aman dan cepat.

Saran kedepannya dapat dilakukan perbandingan algoritma enksripsi yang lebih banyak dan dapat membuat modifikasi algoritma enkripsi yang lebih baik. Dari makalah ini selanjutnya juga dapat melakukan eksperimen yang sama pada data gambar, data Video dan data audio untuk mengembangkan teknik enkripsi yang kuat dengan kecepatan tinggi dan troughput yang baik.

DAFTAR PUSTAKA

[1] Stalling,William, *Cryptography and Network Security Principles and Practice*, 5th ed, 2011.

[2] Shujaat Khan, Mansoor Ebrahim ,*Symmetric Algorithm Survey: A Comparative Analysis*, IJCA,Januari 2013.

[3] P. Traynor, W. Enck, P. McDaniel and T. La Porta, *Mitigating Attacks on Open Functionality in SMS-Capable Cellular Networks*, IEEE/ACM Transactions on Networking, 17(1):40-2009.

[4] Jawahar Thakur, Nagesh Kumar, *DES, AES and Blowfish: Symmetric Key Cryptography Algorithms Simulation Based Performance Analysis*, IJETAE, Desember 2011.

[5] S.Jacobs and C.P. Bean, *International Journal of Science and Research (IJSR),*India Online ISSN: 2319-7064 Volume 2 Issue 4, April 2013.

[6] B. Padmavathi , S. Ranjitha Kumari, *A Survey on Performance Analysis of DES, AES and RSA Algorithm along with LSB Substitution Technique*, IJSR, April 2014.

[7] Nidhi Singha, J.P.S.Raina, *Comparative Analysis of AES and RC4 Algorithms for Better Utilization*, IJCT, 2011.

[8] Md Asif Mushtaque, *Comparative Analysis on Different parameters of Encryption Algorithms for Information*, JCSE, April 2014.

[9] Karale1,Shraddha N, Pendke,Kalyani, Dahiwale ,Prashant, *The Survey of Various Techniques & Algorithms for SMS Security*, IEEE Sponsored 2nd International Conference on Innovations in Information Embedded and Communication Systems, 2015.

[10] Traynor, W. Enck, P. McDaniel dan T. La Porta, *Mitigating Attacks on Open Functionality in SMS-Capable Cellular Networks*, IEEE/ACM Transactions on In Networking, 17(1):40-2009.

[11] Zainuddin,Zahir, Manullang,Evanita V, *E-Lerning Concept Design Of Rijndael Encryption Process*, IEEE International Conference on Teaching, Assessment and Learning for Engineering (TALE), Agustus 2013.

[12] Rahayu,Tri Puji, Yakub, Limiady,Irwan, *Aplikasi Enkripsi Pesan Teks (SMS) Pada Perangkat Handphone Dengan Algoritma Caesar Cipher*, Seminar Nasional

Teknologi Informasi dan Komunikasi 2012 (SENTIKA 2012) ISSN: 2089-9815, 10 Maret 2012.

[13] Pangestu, Tegar Aji, *Implementasi Algoritma Rijndael pada Aplikasi Android Pengirim Short Message Service (SMS) Terenkripsi*, Makalah IF2120 Matematika Diskrit, Sem. I Tahun 2013/2014.