Implementasi Algoritma ECDH dan AES untuk Pengamanan Pesan SMS pada Telepon Seluler

Implementation of EDCH and AES Algorithms for Security of SMS Messages on Mobile Phone

Muhammad Rafi'i[1], Jazi Eko Istiyanto[2]
[1,2]Program Studi Ilmu Komputer S2/S3 FMIPA UGM
[1,2]Jurusan Ilmu Komputer dan Elektronika, FMIPA UGM, Yogyakarta
e-mail: [1]**rappa1979@gmail.com**, [2]jaziugm@gmail.com

Abstrak

Saat ini SMS menjadi kebutuhan bagi pengguna telepon seluler untuk berkomunikasi dengan orang lain. Tetapi pengguna telepon seluler tidak menyadari bahwa pesan yang dikirim bisa disadap atau dirubah oleh pihak yang tidak diinginkan. Untuk itu dibutuhkan suatu sistem keamanan dalam melakukan pengiriman pesan SMS yaitu kriptografi. Dengan keterbatasan sumber daya pada telepon seluler, maka implementasi teknik kriptografi simetrik sangat cocok untuk memenuhi kebutuhan keamanan pesan SMS. Pada kriptografi simetrik terdapat suatu kunci simetrik untuk proses enkripsi dan dekripsi. Supaya pertukaran kunci simetrik aman pada jalur publik maka dibutuhkan suatu protokol untuk pertukaran kunci.

Penelitian ini mengimplementasikan kriptografi simetrik AES untuk enkripsi dan dekripsi pesan sedangkan untuk pertukaran kunci menggunakan protokol Ecliptic Curve Diffie-Hellman(ECDH). Pada penelitian ini protokol ECDH digunakan untuk membuat kunci bersama (shared key) yang nantinya digunakan oleh algoritma AES untuk enkripsi dan dekripsi pesan. Berdasarkan pengujian sistem dapat dilihat bahwa sistem dapat berjalan dengan baik dan kedua algoritma dapat diimplementasikan untuk enkripsi SMS pada telepon seluler berbasis Android. Dalam penelitian ditemukan bahwa pesan menjadi bertambah panjang sebanyak n state dikalikan dengan 4/3 serta pesan dapat dijaga kerahasiaan dan otentikasinya.

Kata kunci—ECDH, AES, simetris, pertukaran kunci, Android

Abstract

SMS is currently a need for mobile phone users communicated with others. But cell phone users do not realize that the messages sent can be tapped or altered by unintended parties. That requires a security system in sending SMS messages is cryptography. With limited resources on the mobile phone, then the implementation of the symmetric cryptographic technique is suitable to meet the security needs of an SMS message. On symmetric cryptography, there is a symmetric key for encryption and decryption. Symmetric key exchange to secure the public path we need a protocol for key exchanges.

This research implements the AES symmetric cryptography for encryption and decryption of messages while applying the key exchange protocol Ecliptic Curve Diffie-Hellman.This protocol is used to create a shared key that will be used for encryption and decryption of messages. Based on testing of the system indicates that the system can function well, and both the encryption algorithm can be implemented for SMS on mobile phones based on Android. In the research found that the length of the message to be increased as much as n state multiplied by 4/3 and can be kept confidential, and authentication.

Keywords— ECDH, AES, symmetric, key-exchange, Android

1. Pendahuluan

Kriptografi memainkan peran penting pada keamanan dari transmisi data. Salah satu transmisi data yang ada adalah SMS. SMS adalah singkatan dari Short Message Service. SMS ini adalah teknologi yang memungkinkan pengiriman dan penerimaan pesan antara ponsel telepon. SMS pertama kali muncul di Eropa pada tahun 1992. Kemudian itu diadopsi ke teknologi nirkabel seperti CDMA dan TDMA (Medani, dkk, 2011). Pesatnya kemajuan pada komunikasi *mobile* telah mengubah SMS sebagai *tools* untuk pesan bisnis dan sosial. Dengan SMS, orang dapat dengan mudah berbagi pesan pribadi dan resmi dengan biaya yang terjangkau. SMS memungkinkan transmisi hingga 1120 bit pesan alfanumerik antara ponsel dan eksternal sistem. Menggunakan SMS Center (SMS-C) untuk operasi *routing* dalam sebuah jaringan dan dapat ditransmisikan ke jaringan lain melalui SMS gateway (Mohammad, dkk, 2009). Penggunaan SMS terancam masalah keamanan, seperti penyadapan, penangkapan (*interception*) dan modifikasi. Hal ini disebabkan pesan SMS yang dikirimkan masih berbentuk plaintext antara stasiun *mobile* dan SMS center yang menggunakan jaringan nirkabel. Isi SMS yang tersimpan dalam sistem operator jaringan dapat dengan mudah dibaca oleh personil mereka ataupun oleh yang tidak berhak. Oleh karena itu, adalah sebuah kebutuhan untuk menyediakan enkripsi tambahan pada pesan yang ditransmisikan (Chavan dan Sabness, 2012).

2. Tinjauan Pustaka

Pengamanan pesan sms bisa dilakukan dengan dua teknik. Teknik pertama adalah dengan menggunakan kompresi dimana algoritma yang bisa digunakan antara lain; *Huffman Coding, Run Length Encoding, Arithmatic Encoding dan Dictionary Based Encoding*. Sedangkan untuk teknik enkripsi yang bisa digunakan adalah Elgamal kriptosistem, RSA kriptosistem dan ECC kriptosistem (Chavan dan Sabness, 2012).

Pengamanan pesan dengan menggunakan metode kompresi bisa dilakukan dengan menggunakan algoritma *Modified Half-Byte.* Dimana dibutuhkan sebuah proses kompresi dan dekompresi yang cepat pada data yang tidak terlalu besar dan hal itu sesuai dengan karakterisktik algoritma *Half byte*. Pengiriman SMS dapat ditingkatkan hingga mencapai 36% dibandingkan dengan pengiriman pesan menggunakan SMS biasa, semakin banyak karakter diinputkan maka hasil kompresi semakin tinggi. Pesan yang terkirim hanya bisa dibuka atau dibaca dengan menggunakan aplikasi khusus karena pesan tersebut sudah dikompres menjadi byte-byte yang tidak beraturan, sehingga apabila SMS yang dikirim dicegat atau diambil oleh pihak lain maka pesan tidak dapat terbaca (Hermawan, 2009).

Selain teknik kompresi, pesan sms juga bisa diamankan dengan teknik enkripsi (Ahomhara, dkk, 2010). Teknik enkripsi untuk pengamanan pesan ada dua yaitu teknik enkripsi asimetris dan teknik enkripsi simteris. Pengamanan pesan dengan menggunakan teknik enkripsi simetrsi sudah banyak dilakukan, misalnya dengan menggunakan teknik enkripsi simteris dengan menggunakan algoritma RC6, kekurangan dari algoritma RC6 ini adalah pesan menjadi lebih besar karena harus bekerja pada 8 bit dan dibutuhkan *padding* untuk memenuhi panjang blok (Permana, 2008). Selain algoritma RC6, algoritma *Blowfish* dan Quadsigroup juga bisa digunakan untuk mengamankan pesan, seperti pada aplikasi SafeSMS. Pengirim bisa memilih salah satu algoritma untuk mengenkripsi pesan yang kemudian melakukan *authentication* dengan algoritma SHA-1 dan penerima mendekrip dengan kunci yang hanya diketahui oleh kedua belah pihak (Hassinen dan Laitinen, 2005).

Algoritma enkripsi simetris lainnya adalah AES. AES adalah salah satu teknik enkripsi simetris yang popular dan banyak digunakan dalam berbagai aplikasi. AES terdiri dari 128 bit, 192 dan 256 bit. AES dapat dimplementasikan pada mobile device (Pitchaiah, 2012). Untuk perbandingan dengan algoritma simetris lainnya dalam hal performa AES masih kalah dengan Blowfish akan tetapi lebih aman [9]. Sedangkan dengan DES, AES menang telak dari segi keamanan dan segi kecepatan AES juga lebih unggul dari DES (Sumitra, 2013).

Selain teknik enkripsi simteris, pengamanan pesan bisa juga dilakukan dengan menggunakan teknik enkripsi asimetris yaitu algoritma RSA dan IBC, penggunaan algoritma IBC diimplementasikan bersamaan dengan algoritma RSA dikarenakan adanya kelemahan dari algoritma IBC yang kinerjanya sangat lambat dan tidak cocok untuk aplikasi telepon seluler sehingga diperlukan algoritma RSA untuk menutupi kelemahan tersebut. RSA dan IBC adalah salah satu algoritma enkripsi kunci publik (Enamy, 2007).

Algoritma enkripsi kunci publik digunakan untuk menutupi kelemahan dari enkripsi kunci simteris. Kelemahan enkripsi kunci simteris terletak pada distribusi kunci dan sangat susah dalam manajemen kuncinya (Abomhara, dkk, 2010).

Manajemen kunci dianggap bagian tersulit dalam kriptografi. Merancang algoritma kriptografi yang aman memang sulit tapi menjaga rahasia kunci jauh lebih sulit. Sebab *cryptanalysis* akan menyerang kunci kriptosistem melalui manajemen kunci yang digunakan. Untuk pertukaran kunci bisa digunakan algoritma ECC dan mengkombinasikannya dengan algoritma enkripsi El-Gamal. Kombinasi dari kedua algoritma tersebut lebih sering disebut dengan Algoritma ECC-Elgamal (Fahmy, 2005). Selain dari ECC pertukaran kunci juga bisa menggunakan algoritma Diffie-Hellman (Hassinen, 2005).

Penelitian yang dilakukan Hendarsyah membahas tentang modifikasi protokol *Diffie-Hellman* dan modifikasi algoritma RC4. Kedua algoritma tersebut diimplementasikan untuk keamanan pesan SMS. Penelitian ini menyatakan bahwa dengan adanya modifikasi protocol Diffie-Hellman yang menyertakan otentikasi maka proses pertukaran kunci dapat diamankan dari serangan *Man In The Middle Attack*. Selain itu dengan adanya modifikasi algoritma RC4 maka pesan dapat diamankan dari serangan *known-plaintext attack.*(Hendarsyah, 2010).

3. Landasan Teori

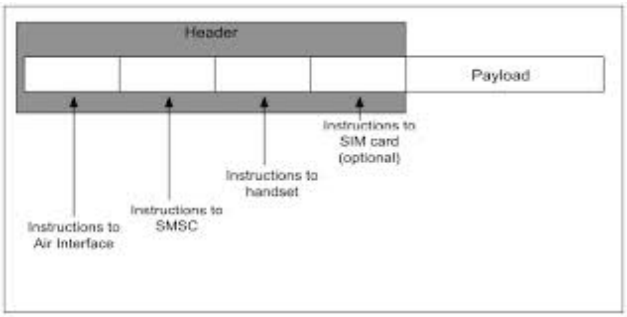
3.1 Short Message Service

Arsitektur *Short Message Service* pada jaringan GSM di ilustrasikan pada Gambar 1 (Chavan dan Sabness, 2012). Pada Gambar 1 ini, pesan singkat pertama dikirim dari M*obile Station* (MS) A ke *Service Message-Service Center* (SM-SC) melalui *Base Station System* (BSS), *Mobile Switching Center* (MSC), dan kemudian *Inter Working* MSC (IWMSC). SM-SC kemudian meneruskan pesan ke jaringan GSM melalui GSM tertentu MSC disebut *Short Message Service Center Gateway* MSC (SMS GMSC). SM-SC dapat menghubungkan ke beberapa jaringan GSM dan untuk beberapa GMSC-GMSC SMS dalam jaringan GSM. Menurut protokol jelajah GSM, SMS GMSC menentukan lokasi MSC dari penerima pesan dan meneruskan pesan itu ke MSC. MSC kemudian menyiarkan pesan melalui BSS ke tujuan MS B (Chavan dan Sabness, 2012; Bodic, 2005).

Gambar1 Arsitektur SMS pada jaringan GSM(Chavan dan Sabness, 2012)

Struktur Pesan

Desain paket pesan SMS terbilang sederhana. Struktur paket SMS bisa dilhat pada Gambar 2. (Mahmoud dkk, 2009).



Gambar 2 Struktur paket sms (Mahmoud, dkk, 2009)

Pada Gambar pesan SMS terbagi menjadi unsur-unsur berikut yang mana hanya user data saja yang terlihat pada telepon seluler penerima :

1. Header, berfungsi sebagai identifikasi tipe pesan :
    - Instruction to Air Surface
    - Instruction to SMSC
    - Instruction to Phone
    - Instruction to SIM Card
2. User Data, isi pesan (*payload*).

Skema Kode Teks

Teks merupakan bagian dari pesan yang bisa dikodekan menurut abjad teks. Dua skema pengkodean teks yang dapat digunakan dalam SMS adalah GSM 7-*bit* alphabet standard, maksimum 160 karakter dan *Universal Character Set* (UCS2). Sedangkan jumlah teks 140 karakter yang terdapat dalam segmen pesan merupakan skema kode 8-*bit* (Bodic, 2005).

3.2 Keamanan SMS

SMS dikirim sebagai teks biasa dimana privasi dari isi SMS tersebut tidak dapat dijamin, tidak hanya melalui jaringan, tetapi juga saat pesan tersebut disimpan pada handset (Chavan dan Sabness, 2012).

Parameter utama keamanan adalah Otentikasi, Kerahasiaan, Integritas, penyangkalan pada pesan (Chavan dan Sabness, 2012). Adapun ancaman keamanannya adalah *Man in The Middle Attack, Reeplay Attack, Message Disclosure, Spamming, DOS, SMS Phone Crash, SMS Virus* dan *SMS Pishing*.

ECDH

Untuk generasi kunci rahasia bersama antara A dan B menggunakan ECDH , keduanya harus setuju atas parameter domain EC. Keduanya akhirnya memiliki sepasang kunci yang terdiri dari kunci d privat (integer dipilih secara acak yang bernilai lebih kecil dari n , dimana n adalah urutan kurva) dan satu lagi adalah kunci publik Q = d * G (G adalah titik pembangkit) . Biarkan (d_A , Q_A) menjadi pasangan kunci publik-privat A dan (d_B , Q_B) menjadi kunci privat - publik B.

1. A Menghitung $K_A = (X_A , Y_A) = d_A * Q_B$
2. B Menghitung $K_B = (X_B , Y_B) = d_B * Q_A$
3. Sejak $d_A * Q_B = d_A d_B G = G = d_B d_A d_B * Q_A$. Oleh karena $K_A = K_B$ dan karenanya $X_A = X_B$. (Dimana G adalah titik pembangkit). Oleh karena itu *share secret* adalah K_A .Karena hampir mustahil untuk menemukan kunci pribadi d_A atau d_B dari KA kunci publik (Ahriwal dan Ahke, 2013).

AES

Advanced Encryption Standard (AES) mulai digunakan 1997. NIST mengumumkan bahwa AES digunakan sebagai pengganti dari algoritma enkripsi Data Encryption Standard (DES) yang telah lama dan kurang aman (Bodic, 2005). AES telah menjadi block chiper dimana dapat memrposes blok 128 bit dari input yang berupa plaintext dalam suatu waktu. AES juga telah pendukung pengaturan kunci 128, 192, dan 256 bit serta lebih efisien daripada DES (Hermawanm 2009).

Proses Enkripsi AES

Secara umum Enkripsi AES terdiri dari empat langkah (Stalling, 2005; FIPSP, 2001):

1. ***SubBytes***, merupakan langkah substitusi *non-linear* di mana setiap *byte* diganti sesuai dengan tabel tertentu. Pada langkah *SubBytes*, setiap byte pada array akan diubah dengan menggunakan *S-Box Rijndael*. *S-Box* yang digunakan, diturunkan dari invers multiplikatif terhadap GF (2^8), yang diketahui mempunyai sifat non-linear.
2. ***ShiftRows***, merupakan langkah transposisi *byte* di mana masing-masing baris *byte* dirotasi dengan jumlah pergeseran tertentu. Jumlah pergeseran baris adalah 0, 1, 2 dan 3 *byte* untuk masing-masing baris pertama, kedua ketiga dan keempat.
3. ***MixColumns***, merupakan operasi pencampuran empat *byte* pada masing-masing kolom untuk menghasilkan keluaran empat *byte*. Masing-masing kolom dibuat menjadi polinom GF (2^8) dan kemudian dikalikan dengan modulo dengan polinomial konstan.
4. ***AddRoundKey***, mengoperasikan *byte-byte* input dengan round key.Round key didapat melalui *algoritma key schedule*. Proses kombinasi dilakukan dengan menggunakan operasi bitwise XOR.

Proses Dekripsi AES

Sama seperti pada proses enkripsi, proses dekripsi AES juga terdiri empat langkah (Stalling, 2005; FIPSP, 2001):

1. ***InvShiftRows*** adalah transformasi byte yang berkebalikan dengan transformasi ShiftRows. Pada transformasi InvShiftRows, dilakukan pergeseran bit ke kanan sedangkan pada ShiftRows dilakukan pergeseran bit ke kiri. Pada baris kedua, pergeseran bit dilakukan sebanyak 3 kali, sedangkan pada baris ketiga dan baris keempat, dilakukan pergeseran bit sebanyak dua kali dan satu kali.
2. ***InvSubBytes*** juga merupakan transformasi bytes yang berkebalikan dengan transformasi SubBytes. Pada InvSubBytes, tiap elemen pada *state* dipetakan dengan menggunakan tabel *inverse S-Box*.
3. Pada ***InvMixColumns***, kolom-kolom pada tiap *state* (*word*) akan dipandang sebagai polinom atas GF(2^8) dan mengalikan modulo $x^4 + 1$ dengan polinom tetap $a^{-1}(x)$ yang diperoleh dari :

$$a^{-1}(x) = \{0B\}x^3 + \{0D\}x^2 + \{09\}x + \{0E\} \quad (1)$$

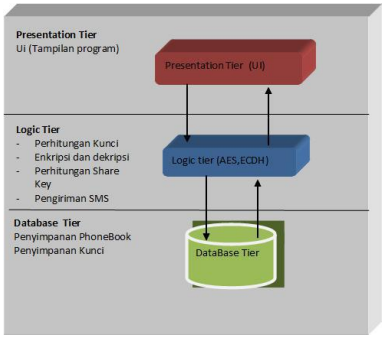
4. Transformasi ***Inverse AddRoundKey*** tidak mempunyai perbedaan dengan transformasi ***AddRoundKey*** karena pada transformasi ini hanya dilakukan operasi penambahan sederhana dengan menggunakan operasi bitwise XOR.

4. Metode Penelitian

Metode penelitian yang digunakan untuk mengimplementasikan algoritma kriptografi ini diantaranya adalah Perancangan Sistem dan Implementasi Sistem. Perancangan sistem terbagi menjadi dua yaitu arsitektur sistem dan rancangan sistem.

4.1 Arsitektur sistem

Arsitektur sistem yang digunakan untuk membuat aplikasi ini adalah 3-tiers. Arsitektur sistem bisa dilihat pada gambar 3.



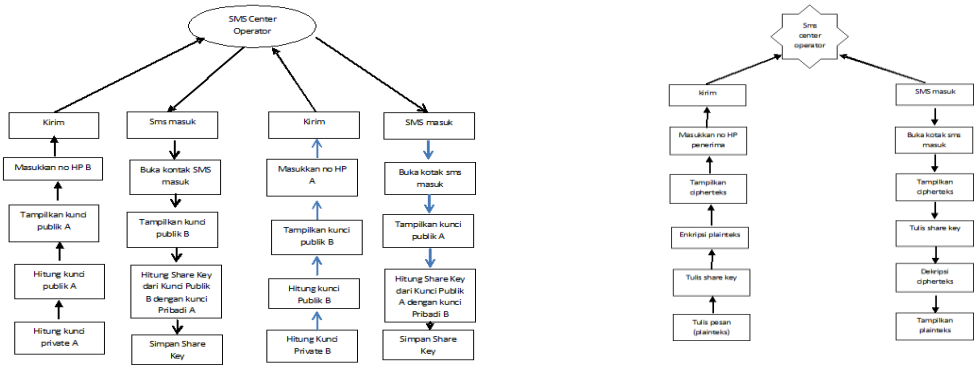
Gambar 3 Arsitektur sistem

Pada gambar 3 terlihat ada 3 tier, tier pertama adalah Presentation Tier yang mana pada tier ini berisikan tampilan untuk interaksi antara pengguna dengan sistem. Pada tier kedua adalah Logic tier yang berisi inti dari program seperti perhitungan kunci, ennkripsi-dekripsi, dan pengiriman sms serta perhitungan share key. Yang terakhir adalah Database Tier, tier ini berfungsi sebagai penyimpanan data dari hasil yang diperoleh oleh Logic Tier. Adapun yang disimpan oleh tier ini adalah data phonebook dan data pasangan kunci.

4.2 Rancangan Sistem

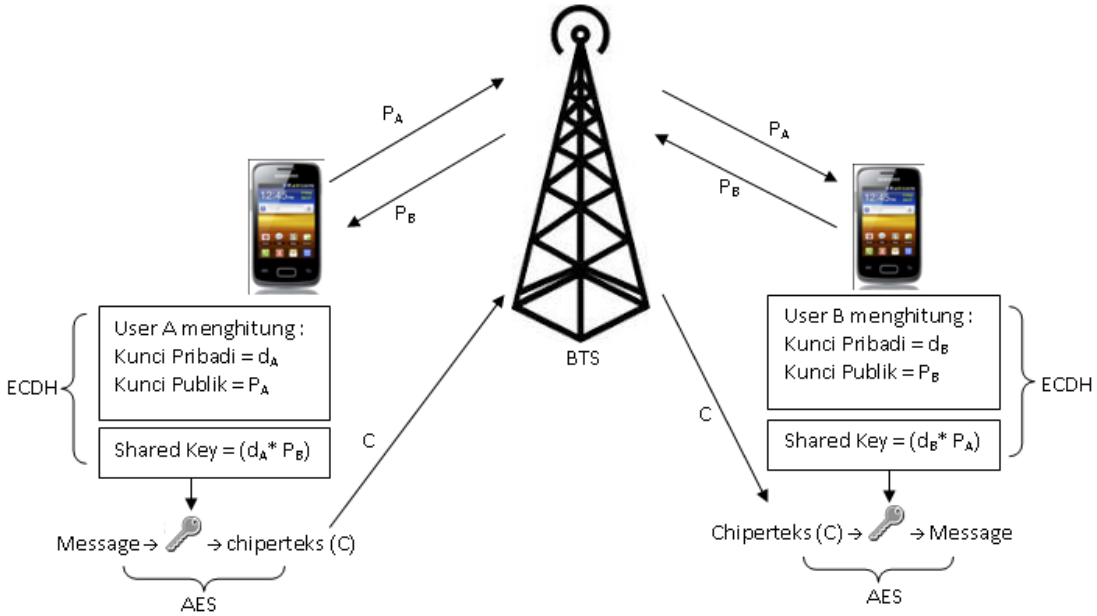
Dalam system yang dibangun terdapat dua buah rancangan yaitu rancangan pertukaran kunci dan rancangan enkripsi-dekripsi.Pada gambar 4a adalah skema proses pertukaran

kunci, dimana si A bertukar kunci dengan si B. Pada gambar 4b adalah rancangan enkripsi dan dekripsi pesan.



(a) (b)

Gambar 4 Pertukaran kunci (a) Enkripsi-dekripsi (b)



Gambar 5 Korelasi ECDH dengan AES

Dari kedua rancangan tersebut hubungan atau korelasi dari kedua algoritma bisa dilihat pada Gambar 5, dimana hasil dari perhitungan algoritma ECDH digunakan sebagai kunci enkripsi-dekripsi AES.

4.3 Implementasi

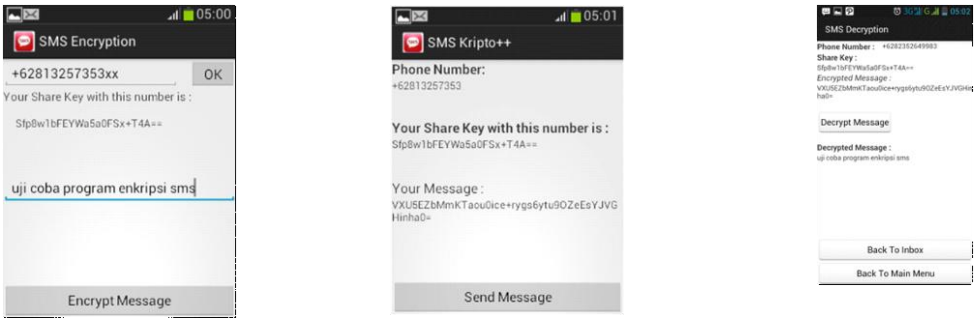
Sesuai dengan rancangan yang ada maka implementasi terbagi dua yaitu pertukaran kunci seperti yang diperlihatkan pada Gambar 6a,6b dan 6c dan proses enkripsi-dekripsi seperti yang diperlihatkan pada Gambar 7a, 7b dan 7c.

(a) (b) (c)

Gambar 6 Pembuatan kunci (a), Pengiriman kunci (b), Import kunci (c)

Pada Gambar 5a terlihat user A men-generate kunci terlebih dahulu kemudian mengirimkannya ke user B dengan memasukkan nomer telepon user B (Gambar 5b). Pada user B setelah kunci publik user A diterima maka user B akan meminta sistem untuk melakukan perhitungan guna mendapatkan *shared key* yang nantinya digunakan untuk enkripsi-dekripsi pesan dan disimpan ke dalam phonebook (Gambar 5c).

Setelah mendapatkan *shared key* maka user B akan mengirimkan pesan kepada user A, setelah memasukkan nomer telepon beserta *shared key* maka user B kemudian meminta sistem untuk mengenkripsi pesan (gambar 6a) dan hasil dari proses enkripsi tersebut dikirimkan kepada user A seperti terlihat pada gambar 6b. Setelah pesan diterima user A dari user B maka user A kemudian meminta system untuk mendekripsi pesan yang diterima dari user B. Sistem kemudian memproses pesan tersebut dan mendekripsi pesan berdasarkan *shared key* yang ada pada sistem. Hasil dari dekripsi tersebut oleh sistem akan ditampilkan pada layar seperti terlihat pada gambar 6c.



(a) (b) (c)

Gambar 7 Menulis sms (a), Pengiriman hasil enkripsi sms (b), Dekripsi sms (c)

5. Hasil dan Pembahasan

5.1 Hasil pertukaran kunci

Dalam pengujian terdapat user A dan user B, dimana kedua user ini menggunakan dua nomer telepon seluler yang berbeda, percobaan menggunakan telepon seluler. Tabel 1 merupakan tabel pengujian dari pertukaran kunci ECDH.

Tabel 1 Pengujian Pertukaran Kunci

Kunci Pribadi A	Kunci Pribadi B	Kunci Publik A	Kunci Publik B	Hasil Perhitungan Share Key Pada	
				A	B
S: 5138376d4aef6914f79bc650c240bea3	S: d552b7ee779fbbf8d362a9afebcb5f15	X: 26bccbe0f3240170d247432e7ee95093 Y: 9e169f416fdec7b5c0a724d2fe7d559f	X: 6dbdfc65efa0dee9c796a995190b7d7c Y: 95b6f0c320ffc9ab204a9c74d895812e	dtkO8kKQw+/rKp93v8npYg== (sama)	dtkO8kKQw+/rKp93v8npYg== (sama)
S: 2d6bb24fb5556b877fa206af12751a6e	S: 43c538acdc0d8a79202f17af342e11e9	X: b2033083ba59edbce31baa96ddaf4e51 Y: 902a3bd3b79cc4efd334097d4dfca9b1	X: 255edea7dcc66f5df9ca6f991c31cbae Y: 4c1837d503791b8601cf3245d0f69d49	M9fLY9TBBMmI7fr1ORSknQ== (sama)	M9fLY9TBBMmI7fr1ORSknQ== (sama)

Dari Tabel 1 dapat dilihat bahwa pertukaran kunci publik hanya mengetahui kunci publik dari user A dan user B sehingga penyerang akan susah untuk menyerang karena tidak mengetahui paramater-parameter untuk menghitung kunci publik dan kunci bersama (*shared key)*.

5.2 Hasil enkripsi dan dekripsi

Hasil ujicoba enkripsi dan dekripsi pesan yang dilakukan pada telepon seluler bisa dilihat pada Tabel 2.

Tabel 2. Pengujian enkripsi dan dekripsi pesan

Pesan	Share Key	Hasil Enkripsi (Cipherteks)	Hasil Dekripsi (Plainteks)
Hari ini ujian jam 10	dtkO8kKQw+/rKp93	MW8naJDRHXd3Bw3o5WDWV84BEo1ROBW5T3KA2eQxq6G=	Hari ini ujian jam 10
Transfernya ke Mandiri aja ya a/n Adnan Suhendar 9000013966461	9WnKxmJrvRveoyUD	tfO9aosWgbpo67udJC96DMXIW3ywwOuFYZGRo6Q4G60/NNe7W0WoGAO5C3ZPIf+T/DJMfOMKUeZJsNVKAknlbg==	Transfernya ke Mandiri aja ya a/n Adnan Suhendar 9000013966461
Maaf,pulsa anda tidak mencukupi untuk permintaan transfer Pulsa	w+GIFi5I8dZLCcug	Y4hbh32I792mZOKECwnncqw6BDTYQzEaM3WsKXr+eEYLmF6SdVBClo/mkh+UaBGf6A6M8rd1+VSwCIGoUz5o3Q==	Maaf,pulsa anda tidak mencukupi untuk permintaan transfer Pulsa

Pada Tabel 2 terlihat bahwa proses enkripsi dan dekripsi bisa berjalan dengan baik. Apabila shared key berubah maka program tidak dapat mendekripsi pesan dan hasilnya terlihat seperti pada Tabel 3.

Tabel 3 Hasil enkripsi jika shared key berbeda

Pesan	Share Key	Hasil Enkripsi (Cipherteks)	Share Key	Hasil Dekripsi (Plainteks)
Hari ini ujian jam 10	dtkO8kKQw+/rKp93	MW8naJDRHXd3Bw3o5WDWV84BEo1ROBW5T3KA2eQxq6G=	WR/A+JGs/87Zt5rC	null
Transfernya ke Mandiri aja ya a/n Adnan Suhendar 9000013966461	9WnKxmJrvRveoyUD	tfO9aosWgbpo67udJC96DMXIW3ywwOuFYZGRo6Q4G60/NNe7W0WoGAO5C3ZPIf+T/DJMfOMKUeZJsNVKAknlbg==	F+19i4JiiUyCnmhO	null
Maaf,pulsa anda tidak mencukupi untuk permintaan transfer Pulsa	w+GIFi5I8dZLCcug	Y4hbh32I792mZOKECwnncqw6BDTYQzEaM3WsKXr+eEYLmF6SdVBClo/mkh+UaBGf6A6M8rd1+VSwCIGoUz5o3Q==	rsKWvi1aDvkivEw8	null

5.3 Pengujian panjang pesan

Pengujian panjang pesan SMS hanya akan dilakukan pada hasil proses enkripsi dan dekripsi. Pengujian ini dilakukan untuk mengetahui seberapa besar panjang pesan setelah di enkripsi. Pada pengujian ini *shared key* yang digunakan adalah sama. Hasil pengujian bisa dilihat pada Tabel 4.

Dari Tabel 4 hasil dari enkripsi pesan bertambah panjangnya tidak seragam semakin banyak jumlah karakter tidak berarti semakin banyak hasil enkripsinya, hal ini disebabkan karakterisktik algoritma AES yang mengkodekan pesan menjadi sebuah state terdiri dari 16 karakter. Selain dari algoritma AES, algoritma enkoding base64 juga mengkibatkan bertambahnya panjang pesan. Algoritma enkoding ini mempuyai karakteristik pertambahan panjang pesan 4/3 per karakternya dan hasil total dari enkoding harus bisa dibagi dengan empat. Apabila hasil akhir enkoding belum bisa dibagi empat maka ditambahkan karakter "=" untuk menggenapinya. Seperti pada data no 1 pada Tabel 6.4 jumlah karakter sebelum enkripsi adalah 85 maka statenya ada 6, ini didapat dari perhitungan 85:16 = 5,3 berarti ada 5 state dimana 5 state sama dengan 80 karakter (5*16 = 80) jadi masih tersisa 5 karakter. Sisa karakter ini digenapi sampai menjadi 16 karakter (1 state) sehingga total karakternya adalah 96 karakter. Kemudian oleh algoritma enkoding base64, panjang pesan menjadi (96*4)/3 = 128 karakter. Karena 128 mod 4 sama dengan 0 maka hasil dari enkoding tersbut tidak akan ditambahkan karakter "=".

Pada data no 2 di Table 5 jumlah karakter sebelum enkripsi adalah 157 maka statenya adalah 157 : 16 = 9,8 karena tidak genap 16 karakter maka hasil dari pembagian tersebut digenapkan menjadi 10 state, total karakter setelah dienkripsi dengan AES adalah 160. Hasil ini kemudian di enkoding dengan base64, maka hasilnya menjadi (160*4)/3=213,3 dibulatkan menjadi 214. Hasil ini kemudian dimodulo 4 ternyata hasilnya adalah 2, karena hasilnya bukan 0 berarti masih ada karakter yang harus ditambahkan, adapun banyaknya karakter yang ditambahkan adalah 4 dikurangi hasil modulo, 4-2 = 2. Sehingga total karakter setelah dienkripsi adalah 214+2=216.

Tabel 4 Pengujian panjang pesan

No	Pesan (plaintext)	Jumlah karakter	Hasil Enkripsi (Cipherteks)	Jumlah ciphertex	Persentase (%)
1	besok diharapkan kedatangannya, hari senin jam 09.00 untuk membicarakan rapat penting	85	r57Et1vk5UVUg8N5XSRC4d70rjqkEGuPU2yaq7t1lcolKV+tggmK6ft6rWblBlnEqGcaKUCbX4p2wxtKvFpBcGDllpKTWzpRPn2mMA5uIuxsmIRehJaLJIz3os/gNjCE	128	45.00%
2	Nomor kamu : 082352649983. Kalau pulsamu habis atau dlm masa tenggang, km bs minta teman utk hub km dg Call Me.Cukup hub *808*no teman#, misal:*808*081355xx#	157	nEXh7yhI5+w4vWPBDx08Ty/1i+oQE4wMH6I2D7F5+M2YzztrQoW6bb7mox6dlx9jvhj/5cHY21VtJi3NgIITCXcyGYanysn8Strob1GYy79DGICeN2NRuM+X0zzUnKTtsbcGVZy9cKub8pvz1UYrhlHQEvsWpgfBDx/RBa9ralorJ+TRZhS5i3F3me2avUcZhmevm7QFUSfftN8vdDWm8a8X55WYKTVuD0yu0yB5RWwv2HlxIpy4Q0TYJDYHV6kD4NKPGwtEI0ASjlG/JZGh2g==	216	72.00%

6. Kesimpulan

Berdasarkan hasil pengembangan sistem, maka dapat ditarik kesimpulan sebagai berikut:

1. Sebuah perangkat lunak yang mengimplementasikan algoritma pertukaran kunci ECDH dan algoritma enkripsi-dekripsi AES telah berhasil dibangun. Perangkat lunak tersebut dapat melakukan pengiriman pesan sms (kunci maupun chiperteks) dan penerimaan pesan sms dengan baik.
2. Proses enkripsi dan dekripsi dengan algoritma AES tidak akan bisa dilakukan tanpa adanya kunci yang dibuat dari proses pertukaran kunci menggunakan algoritma ECDH.
3. Dengan adanya kunci publik dan kunci private maka secara otomatis dapat menjaga *confidentiality* (kerahasiaan) dan *authentication* (otentikasi) pada pesan.
4. Kekurangan dari implementasi algoritma AES untuk enkripsi adalah pesan menjadi lebih panjang dari sebelumnya dikarenakan karakteristik dari algoritma AES dimana setiap statenya harus berisi 16 karakter per state dan karakteristik dari enkoding base64 dimana setiap karakter dirubah menjadi lebih panjang 4/3 per karakter sehingga biaya pengiriman sms menjadi meningkat.

Daftar Pustaka

Abomhara, M.,Khalifa, O., Zakaria, O., Zaidan, A., Zaidan, A., Alanazi, H., 2010, Suitability of Using Symmetric Key to Secure Multimedia Data: An Overview, Journal of Applied Sciences, 15, Vol.10, pp.1656-1661

Ahirwal, R. R., Ahke M., 2013, Elliptic Curve Diffie-Hellman Key Exchange Algorthm for Securing Hypertext Information on Wide Area Network, International Journal of Computer Science and Information Technologies, No.2, Vol.4, pp.363-368

Bodic, G., L., 2005, Mobile Meesaging Technologies and Services SMS, EMS and MMS Second Edition, John Willey and Sons Ltd, England

Chavan, R. R., Sabness, M., 2012, Secured Mobile Messaging, International Conference on Computing, Electronics and Electrical Technologies (ICCEET), Mumbai, 21-22 March 2012

Dennis, S. T., Jhonson, S., 2007, Cryptography for Developer, Syngress, Rockland

Enany, A., 2007, Achieving Security in Messaging and Personal Content in Symbian Phone, Thesis, Department of Interaction and System Design School of Engineering, Blekinge Institute of Technology, Sweden

Fahmy, A., 2005, Key Exchange Protocol Over Insecure Channel ,Proceeding of World Academy of Science, Engineering and Technology,Oslo, Norway,Juni 2005,34-36

Federal Information Processing Standards Publication, 2001, FIPS-197:2001 Announcing the Advanced Encryption Standard (AES), USA:National Institute Of Standards and Technology

Hassinen, M., Laitinen, P., 2005, End-to-end Encryption for SMS Message in the Health Care Domain, Connecting Medical Informatics and Bio-informatics, Vol.5, pp.316

Hassinen, M., SafeSMS - end-to-end encryption for SMS, Telecommunications, 2005. ConTEL 2005. Proceedings of the 8th International Conference on , Zagreb, Croatia, June 15-17 2005, pp.359-365

Hendarsyah, D., 2010, Implementasi Protokol Diffie-Hellman dan Algoritma RC4 untuk Keamanan Pesan SMS, Thesis, Ilmu Komputer, Universitas Gadjah Mada,

Hermawan, A.P., 2009, Kompresi Pesan SMS Menggunakan Algoritma Modified Half-Byte, Thesis, Ilmu Komputer, Universitas Gadjah Mada, Yogyakarta

Kumar, M, A., Kartikeyan, S., 2012, Investigating the Efficiency of Blowfish and Rijindael (AES) Algorithms, I.J. Computer Network and Information Security, No.2, Vol.4, pp.22-28

Mahmoud, T. M., Abdel-latef, B. A., Ahmed, A. A., Mahfouz, A. M., 2009, Hybrid Compression Encryption Technique for securing SMS, International Journal of Computer Science and Security (IJCSS), 6, 3, 473-481

Medani, A., Gani, A., Zakaria, O., Zaidan, A. A., Zaidan, B. B., 2011, Review of mobile short message service security issues and techniques toward the solution, Scientific Research an Essay, No.6, Vol.6, 1147-1165

Menezes, A. J., van Oorschot, P. C., Vanstone, S. A., 2007, Handbook of Applied Cryptography, CRC Press, USA

Permana, R. W. A.,, 2008, Implementasi Algoritma RC6 Untuk Enkripsi SMS Pada Telepon Selular, Skripsi, Departemen Informatika, Institut Teknologi Bandung, Bandung

Pitchaiah, M., Daniel, P., Praveen, 2012, Implementation of Advanced Encryption Standard Algorithm, International Journal of Scientific & Engineering Research, Issues 3, Vol.3, pp.

Stallling, William, 2005, Cryptography and Network Security Principles and Practices, 4thPrentice Hall, New Jersey

Sumitra, 2013, Comparative Analysis of AES and DES security Algorithm, International Journal of Scientific and Research Publication, Issues 1, Vol.3, pp.1-5