Aplikasi Pengaman Pertukaran SMS pada Perangkat Android dengan Metode ECHD dan AES

Hendra[1)]
Sukiman [2)]
Jurusan Teknik Informatika STMIK IBBI Medan[1,2)]
Jl. Sei Deli No. 18 Medan 20214 Indonesia
Telepon 061-4567111
E-mail : hendra.soewarno@gmail.com[1)] sukiman_liu@yahoo.com[2)]

.

Abstrak

Pertukaran pesan melalui Short Message Service (SMS) merupakan suatu layanan yang populer dikalangan pemakai telepon bergerak di Indonesia. Pengiriman SMS dari satu perangkat ke perangkat lainnya melalui SMS Center untuk menyimpan dan menyampaikan SMS ke perangkat tujuan. Enkripsi SMS hanya tersedia dilakukan antara Mobile Station dan Base Transciever Station, sedangkan pada bagian lainnya pesan SMS terbuka sama sekali, sehingga memungkinkan serangan berupa penyamaran, penyadapan maupun modifikasi. Aplikasi pengaman pesan SMS end-to-end ini dikembangkan dengan mengkombinasikan metode Elliptic curve Diffie-Hellman dan Advanced Encryption Standard dengan ukuran key 256-bit untuk pertukaran kunci publik, pembuatan share secret, enkripsi dan dekripsi pesan SMS rahasia dengan menggunakan share secret. Berdasarkan hasil pengujian dengan prototipe aplikasi, metode usulan dapat meningkatan keaslian, kerahasiaan, dan integritas pesan SMS, tetapi membawa konsekuensi berkurangnya jumlah karakter yang dapat dikirim dari 160 karakter menjadi 111 karakter untuk satu pesan SMS.

Kata Kunci : Keamanan SMS , Elliptic curve Diffie-Hellman, Advanced Encryption Standard.

1. Pendahuluan

SMS merupakan salah satu layanan yang populer dan praktis pada telepon bergerak. Layanan ini digunakan oleh berbagai kalangan baik hanya sekedar untuk pertukaran pesan pribadi maupun oleh institusi bisnis untuk kegiatan seperti konfirmasi pemesanan barang, konfirmasi pengiriman, sampai kepada konfirmasi rekening pembayaran. Layanan SMS juga digunakan pada transaksi perbankan yaitu digunakan untuk pengiriman informasi saldo dan PIN untuk transaksi e-Banking.

SMS sendirinya memiliki berbagai kelemahan yaitu SMS dibangun dengan sistem dan program yang sama, dan SMS bisa melakukan *roaming* jaringan setempat hingga ke jaringan asing. Komunikasi SMS memungkinkan pengiriman SMS *spoofing* dalam bentuk penyamaran ataupun manipulasi informasi seperti alamat atau data lainnya yang menyerupai pemakai pada umumnya. Kelemahan dari SMS lainnya adalah isi SMS yang dikirim terbuka di sistim penyedia jasa dan pegawainya sehingga beresiko terhadap penyadapan dan modifikasi[1,2,3]. Beberapa penelitian terkait dengan keamanan SMS mengusulkan pemakaian metode kriptografi asimetris RSA untuk pengiriman SMS[4, 5] maupun kombinasi antara metode kriptografi simetris dan asimetris[6]

Metode kriptografi simetris seperti AES memiliki kinerja yang lebih baik dibandingkan dengan RSA, tetapi membutuhkan pertukaran key melalui jalur yang aman. Penelitian ini mengusulkan kombinasi metode ECDH untuk pertukaran kunci publik eliptik melalui pesan SMS dan pembuatan *share secret* yang nantinya digunakan sebagai kunci enkripsi dan dekripsi pesan SMS rahasia menggunakan metode AES.

Secara umum, tujuan dari penelitian ini adalah mendapatkan suatu rancangan aplikasi pengirim dan penerima SMS yang meningkatkan kesulitan bagi *cracker* untuk membaca, mengubah maupun membuat SMS *spoofing* pada lintasan jalur pengiriman sms antara dua perangkat berbasis Android.

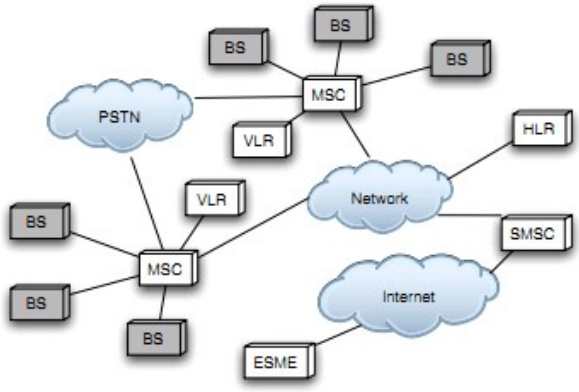
Adapun pembatasan masalah yang dilakukan pada rancangan aplikasi adalah aplikasi hanya beroperasi pada perangkat Android 2.2 keatas.

2. Metodologi penelitian

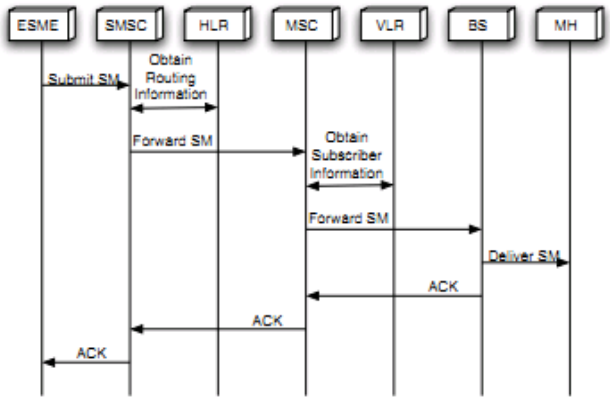
Pengiriman SMS

Ada dua metode untuk mengirim sebuah pesan text ke suatu perangkat bergerak, yaitu melalui suatu perangkat bergerak ataupun melalui suatu *External Messaging Entities* (ESMEs). ESMEs terdiri dari sejumlah besar perangkat berbeda dan memiliki berbagai antarmuka seperti email, portal messaging berbasis web yang terkoneksi pada jaringan telepon bergerak melalui internet maupun kanal dedicated tertentu. Pesan awalnya dikirim ke suatu server yang menangani trafik SMS yang

dikenal sebagai *Short Messaging Service Center* (SMSC). Suatu *provider* yang mendukung pesan text harus memiliki paling sedikit satu SMSC pada jaringan mereka. SMSC perlu untuk menentukan bagaimana pesan disampaikan ke perangkat target. SMSC menanyakan kepada suatu basis data *Home Location Register* (HLR) yang menyimpan data pemakai dan informasi lokasi. Melalui interaksi dengan elemen lainnya, HRL menentukan routing informasi ke tujuan. Jika SMSC menerima balasan bahwa target tidak dapat dicapai, maka pesan akan disimpan untuk dikirim nantinya, jika sebaliknya maka akan dibalas dengan alamat *Mobile Switching Center* (MSC) yang tersedia untuk melayani. Ketika suatu pesan tiba dari SMSC ke MSC, MSC menanyakan kepada suatu basis data *Visitor Location Register* (VLR) yang akan mengembalikan suatu duplikat informasi dari perangkat target ketika dia tidak berada pada HLR-nya. MSC kemudian mengirim pesan kepada Base Station (BS) untuk disampaikan ke target[6]. Proses pengiriman SMS ditunjukan oleh Gambar 1(a) dan 1(b).



(a) Jaringan SMS



(b) Aliran SMS

Gambar 1. Penyederhanaan Jaringan SMS dan Aliran SMS[1]

Keamanan SMS

GSM menyediakan mekanisme keamanan untuk memastikan kerahasiaan dan integritas dari layanan, mekanisme ini ditempatkan antara perangkat bergerak dengan jaringan operator dengan menggunakan algoritma A5, tetapi berdasarkan percobaan serangan kriptanalisis secara realtime terhadap keluaran algoritma A5/1 selama percakapan dua menit, kunci rahasia dapat dipecahkan dalam waktu satu detik, dan pada serangan kedua membutuhkan output dari algoritma A5/1 selama dua detik dari percakapan, dan kunci rahasia berhasil dipecahkan dalam waktu beberapa menit.[7]

Pesan SMS dikirim dengan mekanisme simpan dan arahkan melalui SMSC. Dimana akan diusahakan untuk mengirim pesan ke tujuan jika perangkat berada dalam jangkauan, jika tidak pesan akan disimpan dan dicoba sampai pesan berhasil disampaikan maupun pesan tersebut kadaluarsa. Pesan SMS tersimpan sebagai *plaintext* pada sistim operator maupun perangkat bergerak, sehingga privasi dari pesan SMS tidak dapat dijamin pada sistim operator, maupun pada saat pesan tersimpan diperangkat. Interkoneksi antar *provider* membawa resiko terhadap serangan SMS *spoofing*. Kebutuhan keamanan pada layanan berbasis SMS hanya dapat terpenuhi ketika suatu solusi terkait dengan isu keamanan *end-to-end* tersedia, dimana parameter utama dari keamanan adalah keaslian, kerahasiaan, integritas dan *non-repudiation*.[3]

Keaslian adalah layanan yang dapat membuktikan bahwa identitas yang diklaim dari suatu pelaku komunikasi adalah *valid*, kerahasiaan adalah layanan dapat melindungi data dari pihak yang tidak berhak, integritas adalah layanan memastikan bahwa selama data dikirim tidak dapat diberubah, dan *non-repudiation* adalah layanan memastikan bahwa pelaku tidak dapat menolak telah melakukan pengiriman ataupun menerima suatu pesan.[4]

Elliptic curve Cryptography (ECC)

Ide dari pemakaian Elliptic curve dan kriptografi awalnya diperkenalkan oleh Victor Miller dan N. Koblitz sebagaimana alternatif terhadap sistim kunci public seperti DSA dan RSA. Ketika sistim kriptografi kunci publik (RSA, dan RSA) bekerja secara langsung pada bilangan integer besar, ECC bekerja melalui titik pada kurva eliptik. *Elliptic curve Discrete Log Problem* (ECDLP) misalkan carilah k, berdasarkan P dan Q=kP, problem ini sangat kompleks perhitungannya untuk nilai k yang besar membuat *problem* tersebut harus dipecahkan dalam waktu f*ull-exponential*, sedangkan pada RSA dan DSA merupakan masalah faktorisasi atau *Discrete Log Problem* yang dapat dipecahkan dalam waktu sub-exponential.[8] Hal ini berarti bahwa secara nyata parameter yang lebih kecil dapat digunakan pada ECC dibandingkan dengan RSA dan DSA, dengan ukuran key yang lebih kecil berarti komputasi yang lebih

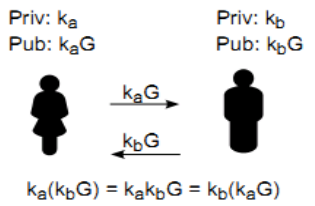
cepat, sebaliknya algoritma untuk memecahkan ECC adalah lebih lambat dibandingkan dengan algoritma memecahkan kriptografi RSA dan DSA. Dengan suatu ECC 256-bit dapat memberikan keamanan yang setara dengan kunci public RSA 2072-bit.[9,10]

Kelompok kurva eliptik yang digunakan didalam kriptografi adalah didenisikan melalui dua jenis dari *field* yaitu: GF(p), dimana p adalah suatu prima dan GF(2^m) dimana masing-masing elemen adalah suatu binary polymonial dengan derajat m (yang mana dapat dinyatakan sebagai suatu string m-bit karena masing-masing koefisien adalah 0 atau 1). GF(2^m) yang lebih sesuai untuk implementasi pada hardware.

Kekuatiran tentang kekayaan intelektual telah memperlambat penerapan ECC karena sejumlah optimasi dan algoritma khusus telah dipatenkan oleh individu maupun perusahaan dalam beberapa tahun terakhir ini. Perusahaan Canada, Certicom Inc memegang lebih dari 130 patent terkait dengan kurva eliptik dan kriptografi kunci public.

Elliptic curve Diffie-Hellman (ECDH)

ECDH adalah *anonymous protokol key agreement* yang memperbolehkan dua pihak yang masing-masing memiliki pasangan kunci publik dan kunci privat kurva eliptik membuat suatu *share secret* melalui kanal yang tidak diamankan, *share secret* tersebut nantinya dapat digunakan sebagai kunci untuk kriptografi kunci simetris. Misalkan Alice ingin membuat suatu *share secret* bersama bob, tetapi kanal yang tersedia memungkinkan penyadapan oleh pihak ketiga. Awalnya perlu disetujui suatu domain parameter, dalam hal ini adalah (p, a, b, G, n, h) untuk prima dan (m, f(x), a, b, G, n, h) adalah untuk binari. Masing-masing pihak memilih kunci privat k_a (suatu bilangan bulat yang dipilih secara acak dalam jangkauan [1, n-1]) dan sebuah kunci publik k_aG (dimana $k_aG = k_a \times G$). Sehingga pasangan kunci untuk Alice adalah (k_a, k_aG) dan pasangan kunci untuk Bob adalah (k_b, k_bG), dan masing-masing pihak melakukan pertukaran kunci publik(k_aG dan k_bG) , dan kemudian masing-masing menghitung sebuah *share secret* dengan melakukan perkalian kunci privat dengan kunci publik pihak lawan, karena $k_ak_bG = k_a(k_bG) = k_b(k_aG)$. Satu-satunya informasi yang diekspose oleh Alice dan Bob adalah kunci publik, maka tidak ada pihak selain Alice dan Bob yang mengetahui kunci privat mereka masing-masing, kecuali pihak tersebut dapat memecahkan *Elliptic curve Discrete Algorithm problem*. *Share secret* bersama Bob dan Alice juga aman, tidak ada pihak lain selain Alice dan Bob dapat menghitung *share secret* kecuali pihak tersebut dapat memecahkan *Elliptic curve Diffie-Hellman problem*. Kebanyakan protokol standard berbasis ECHD menurunkan suatu kunci simetris dari k_ak_bG menggunakan *hash-based key derivation function*. Pembuatan share secret ditunjukan pada Gambar 2.



Gambar 2. Pembuatan share secret ECDH[8]

Elliptic curve Domain parameter

Pembuatan parameter domain tidak selalu dilakukan oleh masing-masing partisipan karena untuk mendapatkan suatu kurva yang aman membutuhkan perhitungan yang memakan waktu dan sulit untuk diimplementasikan. Sebagai hasilnya berbagai badan standar menerbitkan parameter domain kurva eliptik untuk beberapa ukuran field yang umum, sehingga domain parameter sering dikenal sebagai "standard curves" atau "named curve"; suatu nama kurva dapat mengacu berdasarkan nama atau berdasarkan pengenal unik yang didefinisikan pada dokumen standard. Domain parameter memiliki dua jenis parameter yaitu parameter yang sesuai dengan kurva Koblitz(k) dan parameter terverifikasi yang dipilih secara random(r). Parameter k lebih mudah diimplementasikan, sedangkan parameter r memberikan kekuatan yang ekstrim. Pemakaian kurva standar adalah terkait dengan alasan keamanan, karena tidak semua kurva eliptik adalah aman untuk digunakan, dan pemakaian kurva standar juga untuk menjaga *interoperable* antara solusi ECC.[9]

Pemakaian ECDH pada Android

Android menggunakan pustaka Java Bouncy Castle (BC) untuk mengimplementasikan fungsi kriptografi melalui *package* java.security. BC versi 1.46 mendukung ECC pada Android versi 4.0 (Ice Cream Sandwich). Berdasarkan data jumlah perangkat Android yang mengakses Google Play per Januari 2013, dapat terlihat bahwa perangkat menggunakan Android versi 2.3.x (Ginger bread) kebawah masih mendominasi pasar dengan porsi lebih kurang 60%. Perlu diketahui bahwa Android versi 2.3.x kebawah[11] menggunakan BC versi 1.45 yang belum mendukung implementasi ECC sehingga diperlukan pustaka eksternal Spongy Castle(SC) versi 1.47 yang merupakan pustaka BC yang dipaket ulang untuk Android.[13]

Advanced Encryption Standard (AES)

AES merupakan standar enkripsi kunci simetris yang diadopsi oleh pemerintah US pada tahun 2001. Standar tersebut meliputi tiga ukuran key AES-128, AES-192, dan AES-256. AES melakukan enkripsi dan dekripsi dalam blok ukuran 128-bit. AES cipher telah dianalisa secara ketat dan sekarang digunakan secara luas. Algoritma AES bekerja dimulai dengan suatu bilangan acak, dimana kunci dan data dienkripsi melalui empat

tahapan proses matematika. Keempat tahapan tersebut disebut sebagai SubBytes, ShiftRows, MixColumns, dan AddRoundKey. Menurut National Institute of Standards anda Technology (NIST) AES-256 memiliki kekuatan yang setara dengan RSA 15,360 bit dan ECC 512-bit untuk tingkat keamanan yang sama.

Tabel 1. Rekomendasi ukuran key menurut NIST

Symmetric key size (bits)	RSA and DH key size (bits)	EC key size (bits)
128	3072	256
192	7680	384
256	15360	521

Base64-encoding

Permasalahan penyimpan kunci private, pengiriman kunci publik dan pesan hasil enkripsi sebagai data binari mengalami masalah *trailing zero* maupun karakter ASCII yang tidak dapat dicetak, sehingga data perlu dikonversi ke format Base64. Suatu format Base64 adalah kodefikasi dimana setiap 6-bit diganti dengan suatu alpabet 64 karakter yang terdiri dari A-Z, a-z, 0-9 dan + serta /, dengan suatu = sebagai karakter padding. Panjang dari suatu string dalam format Base64- dapat dihitung dengan formula 1.

Base64 = (Bytes + 2 - ((Bytes + 2) MOD 3)) / 3 * 4
….....1)

3. Hasil dan Perancangan

Analisa

Pertukaran kunci publik kurva eliptik dilakukan melalui media SMS yang memiliki keterbatasan 160 karakter, maka perlu ditentukan parameter domain standar yang sesuai. Hasil pengujian dengan parameter dari kurva Koblitz(k) rekomendasi Certicom.[9] ditunjukan pada Tabel 2.

Tabel 2. Ukuran kunci publik

Parameter domain	Ukuran kunci publik dalam format Base64
secp192k1	98
secp224k1	110
secp256k1	**122**

Berdasarkan hasil pengujian maka dipilih parameter domain secp256k1 yang nantinya menghasilkan *share secret* berukuran 256-bit. Langkah berikutnya adalah menentukan ukuran maksimal karakter per-SMS berdasarkan ciphertext hasil enkripsi AES-256 dalam format base64- yang ditunjukan pada Tabel 3.

Tabel 3. Ukuran ciphertext pesan SMS

Ukuran plaintext SMS	Ukuran ciphertext	Ukuran ciphertext Base64
80 s/d 95	96	130
96 s/d 112	**112**	**154**
112 s/d 127	128	175

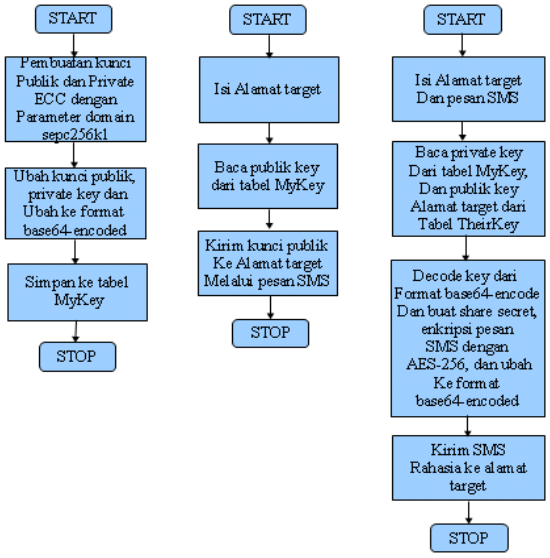
Berdasarkan hasil maka panjang maksimal plaintext pesan SMS adalah 112 karakter.

Perancangan

Secara umum aplikasi yang dirancang terdiri dari 7 komponen utama yaitu:

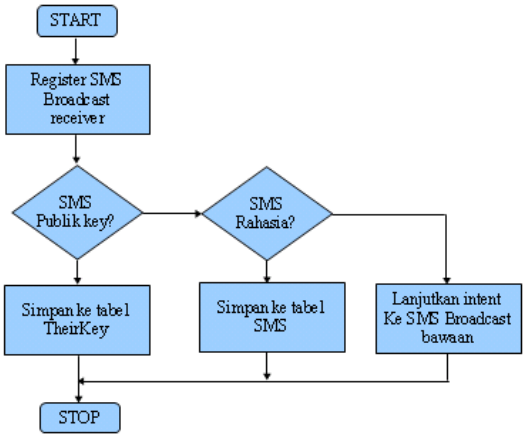
1. Class MyCrypto yang terdiri dari metoda generate kunci publik, kunci privat, *share secret* dengan metode ECDH dan proses enkripsi dan dekripsi dengan AES-256.
2. Class MyDBHelper yang terdiri dari metoda inisialisasi struktur basis data, dan membuka serta menutup basis data.
3. Activity untuk inisialisasi kunci publik dan kunci privat pemakai.
4. Activity untuk pengiriman sms kunci publik.
5. Activity untuk pengiriman ciphertext sms.
6. Broadcast receiver yang berfungsi melakukan penerimaan kunci publik maupun ciphertext sms dan menyimpannya ke database.
7. Activity yang berfungsi menampilkan plaintext dari ciphertext sms.

Awalnya masing-masing pihak perlu membuat sepasang kunci menurut algoritma ECDH yaitu kunci publik dan kunci privat dan disimpan pada perangkat masing-masing. Kemudian masing-masing pihak melakukan pertukaran kunci publik yang nantinya akan digunakan untuk pembuatan *share secret,* selanjutnya *share secret* digunakan untuk proses enkripsi maupun dekripsi pesan SMS menggunakan enkripsi simetris AES-256 sebagaimana yang ditunjukan pada Gambar 3.



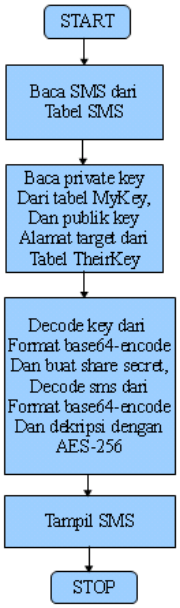
Gambar 3. Pembuatan key dan pengiriman SMS rahasia

Penerimaan pesan SMS pada perangkat dilakukan melalui suatu Broadcast Receiver yang menggunakan intent filter untuk menyaring broadcast yang dipancarkan oleh sistim. Broadcast receiver melakukan pemeriksaan untuk membedakan antara pesan SMS kunci publik, pesan SMS rahasia, dan pesan SMS biasa. Agar pesan SMS kunci publik dapat dibedakan dengan SMS rahasia, maka pada SMS kunci publik diberi awalan karakter @ dan memiliki panjang 123 (122 + 1) karakter. Sedangkan untuk dapat membedakan antara pesan SMS rahasia dengan pesan SMS biasa, maka sesaat pesan SMS diterima akan diperiksa nomor identitas pengirim dengan menanyakan tabel ThierKey, jika nomor identitas tersedia, maka pesan SMS yang diterima akan disimpan ke tabel SMS, dan sebaiknya proses dilanjutkan ke broadcast receiver bawaan Android sebagaimana yang ditunjukan pada Gambar 4.



Gambar 4. Penerimaan SMS kunci publik dan SMS rahasia

Hasil proses dekripsi SMS rahasia ditunjukan pada Gambar 5.



Gambar 5. Menampilkan hasil dekripsi SMS rahasia

Struktur Database

Data terkait dengan aplikasi disimpan pada suatu basis data berbasis SQLite3 yang merupakan aplikasi basis data standard pada perangkat Android, adapun struktur data masing-masing tabel terkait dengan aplikasi disajikan pada Tabel 1, Tabel 2 dan Tabel 3.

Tabel 4. Struktur tabel MyKey

Nama field	Tipe Data	Keterangan
_id	Integer	PK, autoincr
PublicKey	Text	
PrivateKey	Text	

Tabel 5. Struktur tabel TheirKey

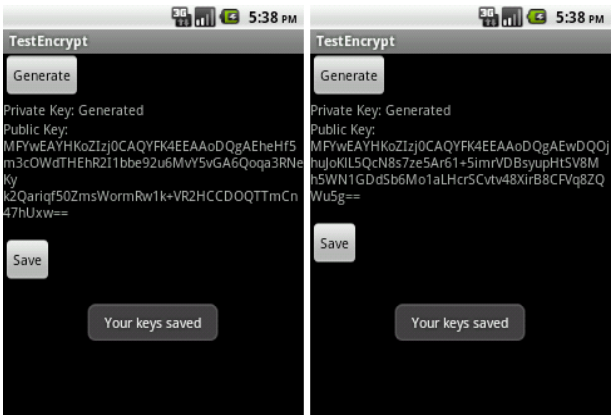
Nama field	Tipe Data	Keterangan
_id	Integer	PK, autoincr
Nomor	Text	
PublicKey	Text	

Tabel 6. Struktur tabel SMS

Nama field	Tipe Data	Keterangan
_id	Integer	PK, autoincr
Jenis	Integer	0=sent, 1=received
Nomor	Text	
Waktu	Integer	long timestamp
Pesan	Text	
Baca	Integer	0=unread, 1 = read

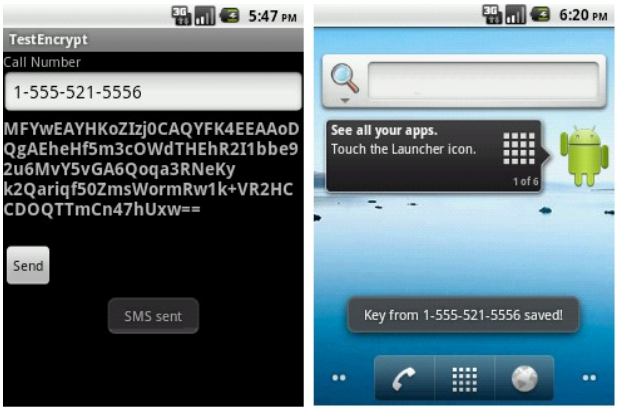
Pengujian

Pengujian dilakukan dengan menggunakan prototipe aplikasi yang dijalankan dengan menggunakan emulator pada System Developement Toolkit Android. Hasil pengujian dilakukan pada proses pembuatan kunci ditunjukan pada Gambar 6.



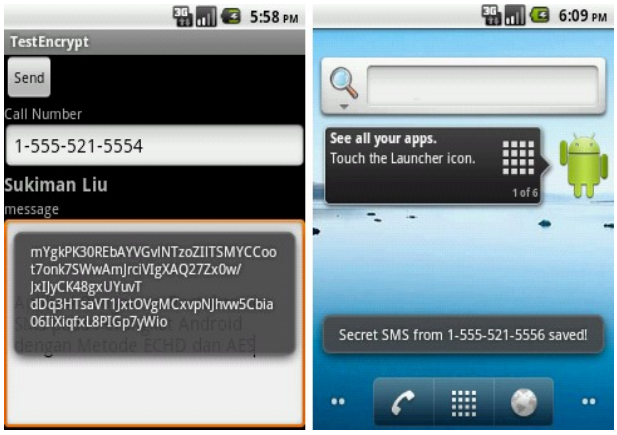
Gambar 6. Pembuatan kunci publik dan kunci privat.

Hasil pengujian proses pengiriman dan penyimpanan kunci publik ditunjukan Gambar 7.

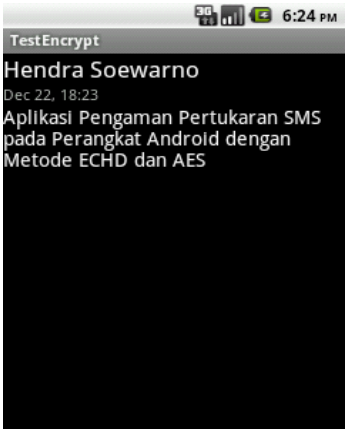


Gambar 7 Pertukaran kunci publik

Hasil pengujian pertukaran SMS rahasia ditunjukan Gambar 8 dan Gambar 9.



Gambar 8 Pertukaran SMS rahasia



Gambar 9 Hasil dekripsi SMS rahasia

Kesimpulan

Teknologi pertukaran pesan melalui SMS tidak menjamin keaslian, kerahasiaan, integritas pesan yang dikirim maupun diterima sehingga diperlukan suatu aplikasi pengamanan *end-to-end* yang dioperasikan pada perangkat pihak yang membutuhkan pertukaran pesan SMS rahasia. Aplikasi yang diusulkan menerapkan metode ECDH dengan parameter domain secp256k1 untuk pembuatan share secret, yang selanjutnya digunakan sebagai kunci untuk enkripsi dan dekripsi menggunakan metode kriptografi simetris AES-256. Berdasarkan hasil pengujian menggunakan prototipe aplikasi, sistim usulan dapat berfungsi untuk meningkatkan keamanan pertukaran pesan SMS terkait dengan isu keaslian, kerahasiaan, dan integritas SMS rahasia. Pemakaian aplikasi usulan membawa konsekuensi berkurangnya jumlah karakter pesan SMS dari 160 menjadi 112.

Daftar Pustaka

[1] W. Enck, P. Traynor, P. McDaniel, T. La Porta, Exploting Open Functionality in SMS-Capable Cellular Networks,CCS'05, November 7-11,2005, Alexandria, Virgina, USA

[2] Tarek M Mahmoud, Bahgat A. Abdel-latef, Awny A. Ahmed & Ahmed M Mahfouz, Hybrid Compression Encryption Technique for Securing SMS, International Journal of Computer Science and Security (IJCSS), Volume (3): Issue (6)

[3] SMS vulnerabilities and XMS technology, Network Security Solutions, (2009, July). Available: http://www.mynetsec.com/files/xms_mobile/SMS_Vulnerabilities_XMS_Technology_White_Paper.pdf

[4] C. Douligeris, D. N. Serpanos, Network Security: Current Status and Future Directions, IEEE Press, Wiley-interscience, Canada, 2007.

[5] Ashish Ranjan, Rjashekara Murthy S, Ramaknath Kumar P, A Review of Secure SMS Based M-Commerce, International Journal of Engineering Sciences & Emerging Technologies, Feb 2012. ISSN: 2231-6604, Volume 1, Issue 2, pp: 1-7

[6] Ikechukwyu, Salem Aljareh, SMS Security: Highlighting Its Vulnerabilites & Technique Towards Developing a Solution, ISBN: 978-1-902560-26-7 © 2012 PGNet

[7] A. Biryukov, A. Shamir, and D. Wagner, "Real Time Cryptanalysis of A5/1 on a PC," in Proceedings of the 7th International Workshop on Fast Software Encryption, 2001, pp. 1-18.

[8] S. Chang, H. Eberle, V. Gupta, N. gura, Elliptic Curve Cryptography- How it Works, http://research.sun.com/projects/crypto

[9] Certicom Research, SEC 2: Recommended Elliptic Curve Domain Parameters, Version 1.0, 20 September 2000

[10] Megha Kolhekar, Anita Jadhav, Impementation of Elliptic Curve Cryptography on Text and Image, International Journal of Enterprise Computing and Business Systems, Vol. 1 Issue 2 July 2011.

[11] Android Developer Dashboards, http://developer.android.com/about/dashboards/index.html, diakses pada 14 Januari 2013.