Optimasi Penggunaan AES dan 3DES pada Blackberry Enterprise Solution

Yosef Ardhito Winatmoko/13509052[1]
Program Studi Teknik Informatika
Sekolah Teknik Elektro dan Informatika
Institut Teknologi Bandung, Jl. Ganesha 10 Bandung 40132, Indonesia
[1]13509052@std.stei.itb.ac.id

***Abstraksi—Blackberry Enterprise Solution* (BES) adalah sebuah sistem terintegrasi yang dikembangkan oleh perusahaan *Research in Motion* (RIM) untuk mempermudah penggunaan produk – produk milik RIM dalam kegiatan bisnis. BES menyediakan dua pilihan pengamanan data yaitu AES dan 3DES. Makalah ini membahas bagaimana mendapatkan status keamanan terbaik yang mungkin didapatkan oleh pengguna dengan memanfaatkan AES dan 3DES yang disediakan oleh BES. Optimasi dilakukan pada komponen *device* dan *server* yang memiliki berbagai kemungkinan konfigurasi modul kriptografi.**

Index Terms*— 3DES, AES, *Blackberry Enterprise Solution, master encryption key

I. PENDAHULUAN

Pada saat ini Negara Indonesia merupakan pengguna Blackberry terbesar di Asia Tenggara[1]. Kebutuhan akan pembangunan server di Indonesia terus didengungkan oleh menkominfo Indonesia. Pihak Research In Motion(RIM) yang merupakan pengembang Blackberry sendiri tidak menanggapi permintaan tersebut dengan serius. Mereka beranggapan bahwa Blackberry sendiri merupakan perangkat yang aman dan cocok untuk pengusaha yang membutuhkan tingkat sekuritas yang tinggi.

Sistem Operasi Blackberry menyediakan dua metode pengamanan pesan yaitu 3DES(*triple DES*) dan AES[2]. Kedua metode pengamanan data tersebut juga diimplementasi dalam proyek besar RIM yaitu *Blackberry Enterprise Solution*(BES). Proyek BES ini sudah dijalankan ke publik dan terdapat berbagai konsekuensi mengenai keamanan data dari BES ini[3]. BES sendiri merupakan server yang terintegrasi (*enterprise server*) bersama dengan Blackberry Desktop Software dan Device Software yang mendukung 3DES dan AES.

Blackberry Enterprise Solution (BES) terdiri dari 6 komponen penyusun penting[4] yaitu:

1. *Blackberry Enterprise Server*: server utama untuk semua hubungan antara *Blackberry Smartphone*, *Enterprise Application*, dan jaringan nirkabel.
2. *Blackberry Mobile Data System* (Blackberry MDS): *Framework* utama dalam pengembangan aplikasi untuk BES. Beberapa konten *Blackberry* MDS antara lain *developer tools*, *administrative service*, dan perangkat lunak untuk perangkat *Blackberry*.
3. *Blackberry Smartphone*: perangkat nirkabel yang terintegrasi dengan *Blackberry Enterprise Server* dan berfungsi untuk memanipulasi data dan suara.
4. *Blackberry Enabled Device*: Kombinasi antara berbagai fitur utama dari *Blackberry* yaitu *push mail* dan kemampuan untuk terhubung ke *Blackberry Enterprise Server*.
5. *Blackberry Alliance Program*: Program dari RIM untuk menciptakan kolaborasi antara berbagai pengembang Blackberry yang saling independen untuk bekerja sama menyediakan aplikasi, *services*, dan solusi untuk implementasi BES.
6. *Blackberry Solution Services*: Membantu berbagai perusahaan mencapai tujuan masing-masing dengan menyediakan bantuan teknis mengenai Blackberry. Berbagai layanan tersebut antara lain *Blackberry Technical Support Services*, *Blackberry Education Services*, dan *RIM Professional Services.*

Kemananan adalah salah satu hal yang paling ditonjolkan dari layanan BES. Selain memberikan pilihan untuk pengguna antara AES atau 3DES, validasi terhadap teknologi keamanan juga dilakukan dengan standard FIPS 140-2. FIPS 140-2 atau *Federal Information Processing Standard Publication 140-2*[5] adalah standard yang dipublikasikan oleh pemerintah Amerika Serikat untuk melakukan akreditasi terhadap modul kriptografi. FIPS 140-2 merupakan bentuk standard khusus yang diterapkan untuk modul – modul kriptografi yang melibatkan perangkat lunak bersamaan dengan perangkat keras.[6]

Dari dokumentasi NIST ditemukan bahwa Blackberry hanya mendapatkan tingkat keamanan level 1(*lowest level of security*)[7]. Sesuai dokumentasi FIPS, Blackberry hanya menyediakan modul – modul kriptografi yang sederhana. Sebagai gambaran, keamanan level 1 adalah sama dengan keamanan data yang disediakan oleh *Personal Computer*(PC) biasa. Makalah ini akan membicarakan setiap fitur keamanan pada BES dan bagaimana mengembangkannya terutama dari sisi *Smartphone*. Tujuan dari makalah ini adalah untuk mendapatkan tingkat keamanan paling tinggi yang mungkin didapatkan oleh pengguna saat memanfaatkan

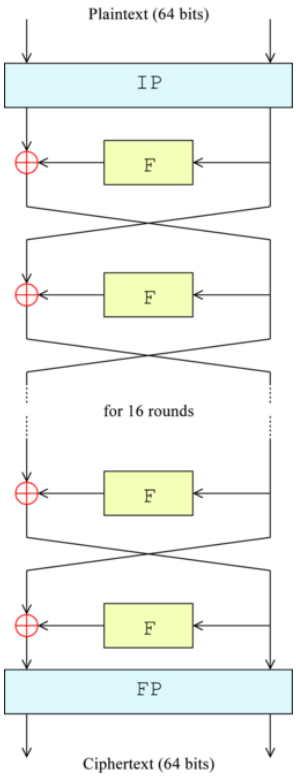
sistem BES sebagai bantuan dalam menjalankan kegiatan bisnis. Hal ini perlu dilakukan karena konfigurasi yang diberikan oleh Blackberry termasuk rumit dan sulit dimengerti oleh orang yang belum mengerti kriptografi.

II. DASAR TEORI

A. *Data Encryption Standard (DES)*

DES adalah salah satu jenis dari *block cipher* yang dikembangkan oleh IBM pada tahun 1977. DES merupakan hasil pengembangan algoritma *Lucifer* dan berperan sebagi salah satu algoritma yang dominan digunakan. DES sebenarnya adalah sebuah standard, bukan algoritma. Algoritma yang biasa dimaksud dengan DES sebenarnya adalah DEA (*Data Encryption Algorithm*). Pada DEA, panjang sebuah blok adalah 64 bit dan menggunakan kunci sepanjang 64 bit, walaupun sebenarnya hanya digunakan 56 bit saja karena 8 bit lainnya digunakan untuk memeriksa *parity*. Pemotongan kunci dilakukan untuk setiap bit pada posisi kelipatan 8 yang berarti bit kunci pada posisi 8, 16, 24, 32, 40, 48, 56, dan 64 dibuang dari kunci.

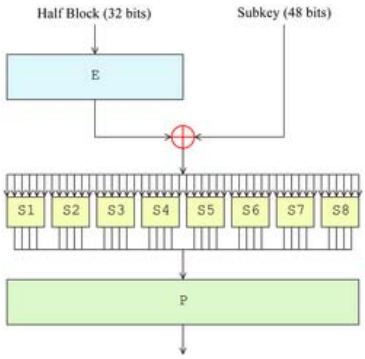
Secara umum, struktur algoritma DES ditunjukkan oleh Gambar 1. Terdapat 16 langkah pemrosesan atau biasa disebut ronde (*round*). Keseleruhan proses mengubah *plaintext* sepanjang 64 bit menjadi *ciphertext* sepanjang 64 bit juga. Sebelum dilakukan pemrosesan terhadap *plaintext*, algoritma DES terlebih dahulu melakukan *IP(Initial Permutation)* dan *FP(Final Permutation)* yang sebenarnya tidak terlalu signikan pada proses penyandian pesan tetapi diimplementasikan pada berbagai perangkat keras yang menggunakan DES. FP merupakan invers dari IP, berfungsi meniadakan efek yang ditimbulkan oleh IP.



Gambar 1. Struktur umum pada algoritma DES

Algoritma DES menggunakan konsep jaringan Feistel pada baik pada proses *enciphering* maupun *deciphering*. Secara sederhana, Feistel membagi blok menjadi dua bagian besar masing – masing 32 bit dan diproses secara bergantian. Kelebihan dari jaringan Feistel adalah bahwa proses enkripsi dan dekripsi akan sama, dengan demikian hanya diperlukan sebuah algoritma saja dalam implementasinya. Hal ini cukup signifikan terutama untuk perangkat keras yang harus menghemat memori. Perbedaan hanya terletak pada urutan bit kunci enkripsi dibalik pada proses dekripsi.

Jaringan Feistel mengacak urutan bit pada setiap potongan blok sebesar 32 bit dengan menggunakan kotak subtitusi tertentu, kemudian menggabungkannya dengan sebagian bit kunci yang telah digeser – geser posisinya. Hal ini dilakukan dengan menggunakan operasi XOR (*exclusive or*). Hasil daripada operasi ini akan dikombinasikan lagi dengan pasangan blok berukuran 32 bit. Hal penting yang perlu diperhatikan adalah pada iterasi terakhir, tidak dilakukan penukaran antar 32 bit blok berpasangan. Inilah fungsi jaringan Feistel yang menyebabkan kesamaan algoritma untuk enkripsi dan dekripsi. Proses ini diilustrasikan pada Gambar 2.

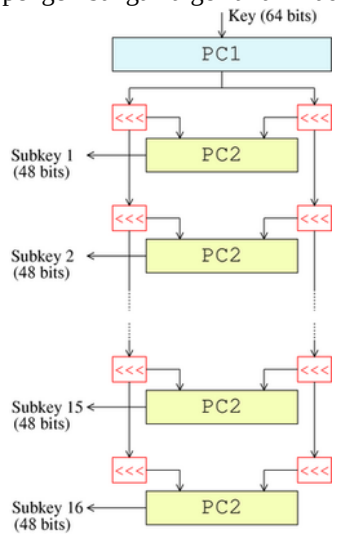


Gambar 2. Implementasi jaringan Feistel pada DES

Sesuai paragraf sebelumnya, jaringan Feistel melakukan operasi XOR antara salah satu blok 32 bit dengan sebagian kunci. Sebagian kunci tersebut disebut sebagai *subkey* dan berukuran 48 bit. *Subkey* diperoleh dengan melakukan pergeseran terhadap kunci asli dengan urutan sesuai Gambar 3. Pertama, 56 bit kunci awal dibagi menjadi dua masing – masing 28 bit. Selanjutnya setiap bagian melakukan *left shift* sebanyak 1 atau 2 bit, tergantung pada langkah ke berapa (disimbolkan dengan "<<<"). Kemudian masing – masing bagian diambil 24 bit paling kiri dan bagian lain 24 bit bagian kanan untuk membentuk *subkey* sepanjang 48 bit.

Berbagai usaha kriptanalis sudah dipublikasikan untuk menyerang algoritma DES. Bagaimanapun algoritma yang sering digunakan untuk menyerang DES sampai saat ini adalah dengan algoritma *bruteforce*. Bahkan dengan algoritma tersebut, DES dianggap sudah tidak aman lagi karena sudah terdapat sebuah perangkat keras yang dapat memecahkan pesan yang disandikan dengan DES dalam hitungan hari menggunakan algoritma *bruteforce*. DES

dianggap memiliki jumlah bit kunci yang terlalu sedikit sehingga perlu dilakukan perbaikan terhadap algoritma ini. Salah satu pengembangan algoritma ini adalah 3DES.



Gambar 3. Urutan penggunaan *subkey* pada DES

B. *Triple Data Encryption Standard (3DES)*

3DES atau *Triple* DES adalah algoritma kriptografi *Triple* DEA yaitu menggunakan algoritma DEA sebanyak 3 kali. 3DES menggunakan sekumpulan kunci yang biasanya terdiri dari 3 kunci biasa diistilahkan K_1, K_2, dan K_3 masing – masing 64 bit termasuk 8 bit *parity*, berarti masing – masing 56 bit kunci efektif. Persamaan 1 menunjukkan implementasi DES sebanyak 3 kali dengan 3 kunci yang berbeda. Ciphertext diperoleh dengan menggunakan enkripsi *plaintext* dengan K_1, dekripsi hasil sebelumnya dengan K_2, dan terakhir enkripsi dengan K_3 seluruhnya menggunakan algoritma DES

$$ciphertext = E_{K3}(D_{K2}(E_{K1}(plaintext)))$$

Persamaan 1. Penggunaan 3 kali DES dengan 3 kunci yang berbeda untuk melakukan enkripsi pada 3DES

. Untuk proses dekripsi cukup membalik skema enkripsi seperti ditunjukkan oleh Persamaan 2.

$$plaintext = D_{K1}(E_{K2}(D_{K3}(ciphertext)))$$

Persamaan 2. Dekripsi DES sebanyak 3 kali pada 3DES

Hal penting yang perlu diperhatikan adalah pemilihan 3 buah kunci pada algoritma ini. Ada 3 pilihan kombinasi kunci yaitu:

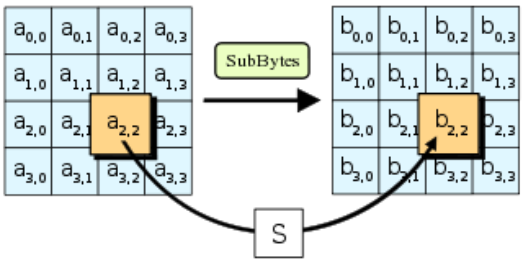
- 3 buah kunci berbeda
- 2 buah kunci berbeda, kunci ketiga menggunakan kunci pertama
- menggunakan sebuah kunci untuk 3 kali algoritma DES.

Pemilihan kunci yang digunakan sangat penting karena menentukan jumlah kemungkinan kombinasi kunci jika ada kriptanalis yang mencoba melakukan serangan terhadap algoritma ini. Biarpun begitu, modifikasi terhadap algoritma ini juga dilakukan terhadap urutan enkripsi dan dekripsi yang dilakukan untuk setiap algoritma DES pada 3DES. Pada Persamaan 1 menggunakan urutan enkripsi-dekripsi-enkripsi tetapi penggunaan urutan dekripsi-enkripsi-dekripsi juga sering digunakan.

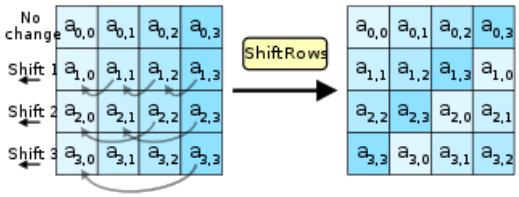
C. *Advanced Encryption Standard (AES)*

AES adalah standard yang digunakan setelah DES dinyatakan tidak aman. AES dibuat oleh Joan Daemen dan Vincent Rijmen. Berbeda dari DES, AES tidak menggunakan jaringan Feistel. AES dirancang dengan menggunakan prinsip jaringan subtitusi-permutasi. Panjang satu blok pada AES adalah 128 bit dan terdapat pilihan panjang kunci 128 bit (AES-128), 192 bit (AES-192) dan 256 bit (AES-256), namun versi 192 bit jarang digunakan pada kehidupan nyata. AES menggunakan 4x4 matriks byte dalam melakukan *enciphering* dan *deciphering*.

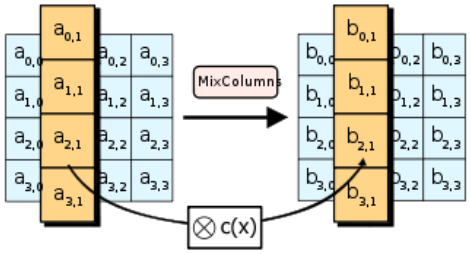
AES terdiri dari sekumpulan proses transformasi yaitu ekspansi kunci, inisialisasi, dan 3 proses transformasi yang diulang-ulang yaitu subtitusi, *shift*, *mix columns*, dan *round key*. Subtitusi adalah mengganti bit pada setiap posisi dengan suatu bit lain yang ditentukan oleh sebuah *s-box* yang sudah terdefinisi sebelumnya, ditunjukkan oleh Gambar 4. *Shift* adalah menggeser setiap baris pada matriks sejumlah tertentu kolom. Baris pertama tidak digeser sama sekali, baris kedua digeser sebanyak satu kolom, baris ketiga digeser sebanyak dua kolom, dan baris keempat digeser sebanyak tiga kolom. Proses *shift* diilustrasikan oleh Gambar 5. Transformasi *mix columns* melakukan XOR pada setiap kolom matriks dengan matriks lain yang sudah didefinisikan sesuai ilustrasi Gambar 6. Operasi *mix columns* tidak dilakukan pada iterasi terakhir. Terakhir adalah *addroundkey* dimana dilakukan XOR dengan kunci pada ronde tersebut, seperti diilustrasikan oleh Gambar 7.



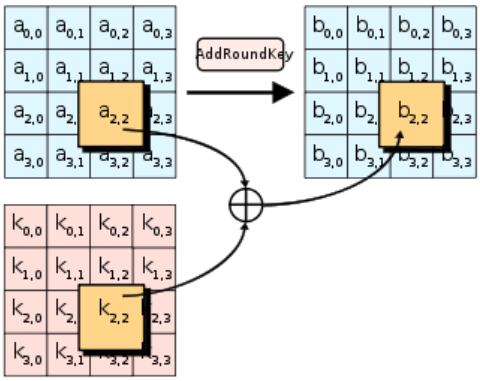
Gambar 4. Transformasi subtitusi pada AES



Gambar 5. Penggeseran baris pada AES



Gambar 6. Melakukan XOR terhadap setiap kolom matriks pada AES

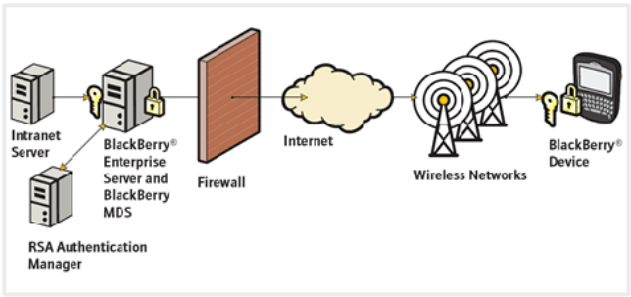


Gambar 7. Melakukan XOR matriks dengan kunci pada AES

III. BLACKBERRY ENTERPRISE SOLUTION

A. Arsitektur Keamanan pada BES

Penerapan AES dan Triple DES pada BES dilakukan di *Blackberry Enterprise Server* dan *Blackberry Device*. *Device* yang dimaksud disini bisa berupa *smartphone Blackberry*, *smartphone Blackberry-enabled*, dan *Blackberry Desktop Software*. Komunikasi antara *device* dengan *server* sepenuhnya dilakukan terenkripsi. Proses dekripsi hanya dapat dilakukan oleh *server* dan *device* seperti ditunjukkan oleh Gambar 8.

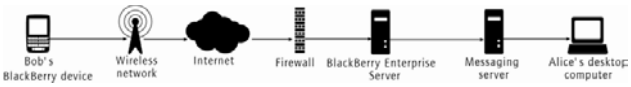


Gambar 8. Abstraksi Arsitektur *Blackberry Enterprise Solution*

AES pada BES baru diimplementasikan pada versi 4.0(*server*, *device*, dan *desktop software*). Ada dua proses enkripsi yang terjadi terkait transfer pesan di BES. Proses yang pertama adalah saat pengguna mengirimkan pesan dari *device* dan *server* menggunakan salah satu algoritma kriptografi kunci simetrik untuk melakukan enkripsi dan dekripsi pesan sesuai urutan proses berikut:

1. *Device* melakukan kompresi terhadap pesan
2. *Device* melakukan enkripsi menggunakan kunci pesan tertentu
3. *Device* melakukan enkripsi terhadap kunci pesan menggunakan *master encryption key* yang unik untuk setiap *device.*
4. *Device* mengirim kunci pesan dan pesan yang terenkripsi.
5. *Server* menerima kunci pesan dan pesan yang terenkripsi dari *device*
6. *Server* melakukan dekripsi kunci pesan menggunakan *master encryption key* yang terdaftar untuk *device*.
7. *Server* melakukan dekripsi pesan dengan pesan kunci yang sudah terdekripsi
8. *Server* melakukan dekompresi terhadap pesan yang sudah terdekripsi.

Proses yang berlangsung pada saat suatu *device* menerima pesan mirip dengan proses saat *device* mengirimkan pesan. Skema umum komunikasi antara *device*, *server*, dan *desktop application* diilustrasikan oleh Gambar 9. Contoh pada Gambar 9 mengilustrasikan pengiriman pesan yang dilakukan dari sebuah Blackberry *device* ke *desktop application*.



Gambar 9. Jalur komunikasi pengiriman pesan antara Blackberry *device*, *desktop*, dan *server* pada BES

B. Penggunaan Master Encryption Key

Salah satu komponen penting proses pengiriman dan penerimaan pesan pada BES adalah *master encryption key*. *Master encryption key* bersifat unik untuk masing-masing *device*. Kunci unik ini disipan di *server* dan *device* dan keduanya harus sama. Jika pada saat ingin mengirimkan pesan ternyata kunci enkripsi yang tersimpan di *device* dan *server* berbeda maka semua pesan akan dihapus karena proses dekripsi tidak akan memberikan hasil pesan yang sesuai. Pilihan lain adalah dengan membuat sebuah *master encryption* key baru. Penyimpanan *master encryption key* ditentukan oleh tiga komponen yaitu konfigurasi basis data, *messaging server*, *device flash memory state*. BES sendiri mendukung tiga jenis aplikasi basis data yaitu:

- IBM Lotus® Domino® server: kunci disimpan di *Blackberry database profile*.
- Microsoft® Exchange server: kunci disimpan pada mailbox pengguna di program desktop.
- Novell® GroupWise® server: kunci tidak disimpan.

Salah satu fitur yang menarik pada BES adalah pengaksesan kunci yang sebelumnya pernah digunakan karena *previous key* disimpan selama 7 hari secara bawaan di *flash memory* dan kita dapat menentukan *pending key*, kunci yang akan digunakan pada penggantian *master encryption key* berikutnya. Kunci ini sendiri dapat dengan mudah dibangkitkan oleh pengguna menggunakan dua pilihan metode yaitu *desktop* dan *device*. Secara umum, pembangkitan *master encryption key* dengan metode ini memiliki proses sebagai berikut:

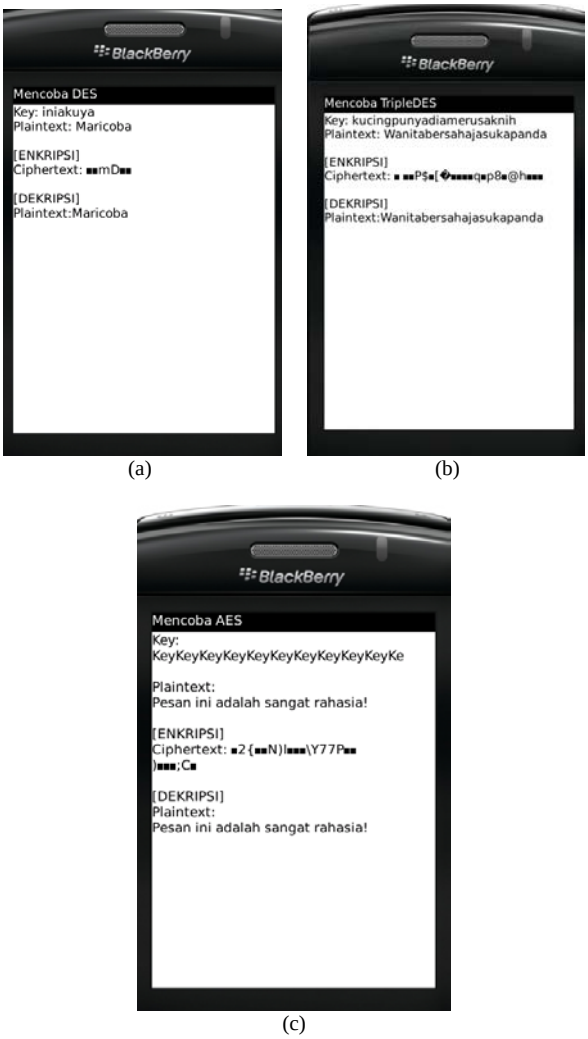
1. Pengguna diminta untuk menggerakkan tetikus.
2. Posisi tetikus baru di layar kemudian diambil sebagai *seed* untuk membangkitkan 3 byte semi-acak. Posisi baru akan diambil jika tetikus dinyatakan bergerak dari posisi sebelumnya.
3. Demikian diulangi proses ini dengan interval bilangan acak antara 50 sampai 150 milisekon sampai mendapatkan 384 byte bilangan semi-acak.
4. *Desktop application* kemudian menerima 384 byte semi-acak dari MSCAPI(*Microsoft Crypto API)*. Dengan demikian saat ini sudah terdapat 768 byte semi-acak.
5. *Desktop application* melakukan *hashing* terhadap 768 byte semi-acak yang dimiliki menggunakan SHA-512, memori yang tidak lagi berguna dikosongkan.
6. Disini *desktop application* akan menggunakan 256-bit pertama untuk membangkitkan *master encryption key* jika konfigurasi yang dipilih adalah dengan AES. Sementara jika yang digunakan adalah 3DES maka pembangkitan *master encryption key* hanya menggunakan 128-bit pertama. Setelah itu, semua bit memori yang tidak lagi digunakan segera dibersihkan.

C. Modul Kriptografi pada BES

Untuk fungsional kriptografi, BES menggunakan RIM *cryptographic* API. Untuk developer, API ini tersedia pada *package* net.rim.device.api.crypto. API ini sudah termasuk ke dalam *developer tools* Blackberry berupa *plugin* untuk IDE Eclipse versi Galileo. API ini sudah menyediakan berbagai algoritma untuk pengamanan data termasuk algoritma 3DES dan AES yang digunakan sebagai algoritma enkripsi dan dekripsi di BES. Masing – masing algoritma memiliki 2 komponen utama:

- Kelas Key: Kelas yang merepresentasikan kunci untuk algoritma kriptografi tertentu. Termasuk di dalamnya adalah kelas DESKey, TripleDESKey, dan AESKey.
- Kelas Engine: kelas yang menampung algoritma enkripsi dan dekripsi *array of byte.* Masing – masing algoritma DES, TripleDES, dan AES memiliki kelas engine baik untuk enkripsi maupun dekripsi.

Masing – masing kelas memiliki fungsi yang memroses sekumpulan *array of byte* untuk dienkripsi atau didekrispi. Pada BES, algoritma DES tidak digunakan melainkan algoritma TripleDES yang digunakan. TripleDES pada BES hanya menggunakan dua buah kunci, maka dari itu kunci pertama digunakan dua kali yaitu sebagai kunci pertama dan kunci ketiga. Untuk menguji fungsional modul kriptografi pada API RIM yang digunakan oleh BES, dibuat 3 buah program sederhana dengan menggunakan *Blackberry developer tools*, dikembangkan dalam IDE Eclipse dengan *Blackberry 9550 Simulator*. Hasil dari pengujian sederhana ini ditunjukkan oleh Gambar 10, masing – masing menggunakan algoritma yang berbeda yaitu (a) DES, (b) 3DES, dan (c) AES.



Gambar 10. Pengujian implementasi algoritma (a) DES, (b) 3DES, dan (c) AES pada RIM *Cryptographic* API

IV. ANALISIS FITUR KEAMANAN BES

Analisis keamanan BES yang dilakukan menggunakan komponen:

- BlackBerry® Enterprise Server Express for Microsoft Exchange v4.1.7 (*trial*)
- Desktop Software v6.1.0 B38 (*Multilanguage*)
- BlackBerry® Bold™ 9700
- Blackberry OS v5.0
- Firmware: v4.3.0.104 (Platform 2.6.0.59)
- *Cryptographic Kernel*: v3.8.5.11c

Jika dilihat dari jalur komunikasi dan proses yag berlangsung saat sebuah pesan dikirimkan pada sebuah BES, *master encryption key* merupakan komponen yang paling penting untuk dijaga keamanannya. Penyimpanan *master encryption key* sendiri harus dijamin aman dari berbagai serangan. *Master encryption key* ini hanya boleh berada pada *server* dan *device* itu sendiri dan tidak boleh diikutsertakan pada pesan yang dikirimkan. Jika *master encryption key* diikutsertakan pada proses pengiriman pesan maka besar kemungkinannya kriptanalis akan melakukan penyerangan dengan metode *meet-in-the-*

middle karena jalur komunikasi sangat terbuka terutama jika pengiriman pesan dilakukan dengan metode *wireless*.

BES dengan basis data menggunakan Microsoft Exchange menyimpan *master encryption key* pada sebuah folder *hidden* bernama BlackBerryHandheldInfo dan menyimpan dua macam informasi penting yaitu:

- konfigurasi *device*, termasuk di dalamnya *master encryption key*.
- *binary form* dengan *tag* untuk menandai *master encryption key* tertentu. *Tag* dapat berisi: 0x6002 (*pending*), 0x6003 (*current*), and 0x6004 (*previous*)

Keamanan pada folder ini cukup terjamin karena isi daripada folder ini sendiri dienkripsi menggunakan *grand master key* yang dimiliki oleh Blackberry. Informasi mengenai *grand master key* sangat sedikit karena merupakan kunci enkripsi untuk seluruh kunci lain yang dimiliki oleh *device* Blackberry. Permasalahannya adalah *master encryption key* hanya akan dibangkitkan pada saat inisialisasi *device* atau *server* jika pengguna tidak melakukan pembangkitan secara manual padahal keamanan pengiriman pesan sangat bergantung pada *master encryption key*.

Penyimpanan *master encryption key* memiliki dua status yang sebenarnya kurang efektif yaitu kunci yang *pending* atau *previous.* Kunci dengan kedua status ini sebenarnya kurang memberikan manfaat dan justru membuat resiko kebocoran *master encryption key* menjadi semakin besar. Penyimpanan *pending* seharusnya tidak perlu dilakukan karena tidak memberikan dampak yang signifikan jika kita sudah menyiapkan *master encryption key* dan menyimpannya terlebih dahulu daripada membangkitkan suatu *master encryption key* dan secara langsung menggunakan kunci tersebut.

Fitur kedua yang perlu diperhatikan adalah pada BES terdapat pilihan untuk menggunakan salah satu dari dua pilihan algoritma enkripsi-dekripsi yaitu 3DES dan AES. AES mungkin memang termasuk fitur yang tergolong baru karena baru dapat digunakan oleh BES minimal versi 4.0. Pada BES, 3DES hanya menggunakan dua buah kunci DES. Karena setiap kunci DES memiliki panjang 56-bit maka 3DES dengan dua kunci memiliki 2^{112} (5,19 x 10^{33}) kemungkinan kunci. NIST sendiri pernah menyatakan bahwa 3DES dengan dua kunci hanya memiliki keamanan setara panjang kunci 80-bit karena metode penyerangan *known-plaintext attack* mudah dilakukan pada BES. Unsur ini cukup penting karena pada BES, kita dengan mudah dapat melihat *plaintext* dan *ciphertext*-nya, tidak ada metode untuk menyembunyikan kedua hal tersebut. Dengan demikian kita dapat menganggap keamanan 3DES pada BES setara kunci dengan 2^{80} (1,24 x 10^{24}).

Misalkan kita mengambil salah satu *processor* PC non-*server* paling cepat saat ini yaitu Intel Core i7 *Extreme Edition* 3960X (*Hex core*) 3.33 Ghz yang dapat melakukan 177,730 *Million instructions per second* (MIPS). Dengan demikian *processor* tersebut dapat melakukan *processor* dapat melakukan 56,08 x 10^{15} iterasi per-tahun. 1,7 x 10^{7} tahun untuk melakukan iterasi ke seluruh kemungkinan solusi. Jika dibandingkan dengan AES-256 yang digunakan oleh BES, hasil yang diperoleh akan jauh berbeda karena AES menggunakan 256 bit kunci. Pembangkitan kunci baik 3DES maupun AES juga sudah sangat baik dengan memanfaatkan *master encryption key* dalam memilih bilangan semi-acak.

Fitur ketiga sebenarnya merupakan batasan yang terdapat pada RIM *Cryptographic* API, dimana jumlah key harus sesuai dengan ketentuan setiap algoritma, tidak ada ekspansi (penambahan). Kunci harus ditulis lengkap dan menggunakan listrik untuk memegani kunci. Penggunaan memori pada BES pada saat melakukan enkripsi dan dekripsi banyak terbuang karena memaksakan jumlah bit yang harus sesuai dengan ketentuan panjang kunci dari algoritma yang digunakan. 128-bit untuk 3DES dan 256-bit untuk AES. Proses enkripsi dan dekripsi pada BES menggunakan jauh lebih banyak memori dengan alasan keteracakan poin yang didapatkan padahal hasil pengacakan juga tidak jauh berbeda.

Fitur keempat yang sebenarnya adalah pilihan pengguna, jenis *server* yang digunakan. Terdapat tiga macam basis data yang dapat digunakan pada BES yaitu

- *IBM Lotus® Domino® server*
- *Microsoft® Exchange server*
- *Novell® GroupWise® server*

Setiap jenis basis data memiliki keunggulan dan kelemahan masing – masing. Diantara keunggulan dan kelemahan yang dimiliki setiap jenis basis data, terdapat beberapa sifat yang mencolok. *Novell* belum sepenuhnya di *support* oleh BES sementara IBM Lotus tidak menyediakan sarana untuk melakukan enkripsi secara wireless sementara saat ini mobilitas sangat dijunjung oleh pengguna.

V. Optimasi Keamanan BES

Banyak sekali konfigurasi keamanan yang dapat diatur pada BES karena BES sendiri memiliki tiga komponen utama yang masing – masing memiliki konfigurasi keamanan yang berbeda. Dari hasil analisis yang dilakukan maka untuk mendapatkan kemungkinan keamanan tertinggi pada BES dengan memanfaatkan modul kriptografi RIM API terutama algoritma 3DES dan AES dapat dilakukan dengan mengimplementasikan konfigurasi dan kegiatan berikut:

1. Menggunakan AES sebagai algoritma enkripsi pesan. AES memberikan tingkat keamanan yang jauh lebih baik daripada 3DES pada BES yang hanya menggunakan dua buah kunci. 3DES cukup digunakan untuk masalah adaptasi terhadap kompabilitas versi *server* tertentu.
2. Membangkitkan *master encryption key* yang berbeda secara berkala. Pembangkitan *master encryption key* dapat dilakukan baik dari sisi *server* maupun *device*. Cara terbaik adalah selalu membangkitkan *master encryption key* untuk setiap persiapan pemindahan informasi secara berkala antara *server* dengan *device*. Proses pembangkitan pun dilakukan oleh kedua pihak,

tergantung pihak mana yang akan mengirimkan informasi. Bagaimanapun komunikasi mengenai *master encryption key* ini tidak boleh dilakukan bersamaan dengan pesan karena memperbesar kemungkinan penyerangan oleh pihak ketiga.

3. Mematikan fungsi penyimpanan *master encryption key* yang berstatus *pending* untuk meminimalisasi kemungkinan penyerangan yang berhasil.
4. Menggunakan *Microsoft Exchange* sebagai basis data yang digunakan oleh BES. *Microsoft Exchange* menyediakan fitur keamanan yang lebih baik dari dua pilihan lainnya misalnya *content protection* dan *compression*, *password keeper* dan *device wipe*, dan yang paling penting adalah kemudahan pembangkitan *encryption key* secara nirkabel.

DAFTAR REFERENSI

1. http://tekno.kompas.com/read/2011/11/15/08135042/Kemkominfo.Terus.Desak.RIM.Bangun.Server.di.Indonesia (diakses 26 Februari 2012)
2. http://docs.blackberry.com/en/admin/deliverables/25763/Algorithms_to_encrypt_data_1682588_11.jsp (diakses 27 Februari 2012)
3. http://www.blackberryforums.com.au/forums/general-bes-discussion/1374-bes-encryption-algorithms-impact-blackberry-smartphone-users.html (diakses 28 Februari 2012)
4. http://us.blackberry.com/business/types/enterprise/BlackBerry_Enterprise_Solution.pdf (diakses 28 Februari 2012)
5. http://csrc.nist.gov/cryptval/140-2.htm (diakses 27 Februari 2012)
6. http://en.wikipedia.org/wiki/FIPS_140-2 (diakses 28 Februari 2012)
7. http://csrc.nist.gov/groups/STM/cmvp/documents/140-1/140val-all.htm (diakses 28 Februari 2012)
8. http://testlab.sit.fraunhofer.de/downloads/certificates/Certification_Report-06-104302.pdf (diakses 18 Maret 2012)
9. http://www.gen-i.co.nz/services/Documents/Service-Resources/BlackBerry_Enterprise_Solution_Security_version_4.pdf (diakses 18 Maret 2012)
10. http://docs.blackberry.com/en/admin/deliverables/4133/BB_Ent_Soln_Security_4.1.6_STO.pdf (diakses 19 Maret 2012)
11. http://csrc.nist.gov/publications/nistpubs/800-57/sp800-57-Part1-revised2_Mar08-2007.pdf (diakses 19 Maret 2012)
12. http://us.blackberry.com/apps-software/server/5/compare.jsp (diakses 19 Maret 2012)
13. Munir, Rinaldi. 2006. Kriptografi. Bandung: Informatika

PERNYATAAN

Dengan ini saya menyatakan bahwa makalah yang saya tulis ini adalah tulisan saya sendiri, bukan saduran, atau terjemahan dari makalah orang lain, dan bukan plagiasi.

Bandung, 20 Maret 2012

Yosef Ardhito Winatmoko - 13509052