STUDI KASUS MENGENAI IPSec BERKAITAN DENGAN TEORI DAN PRAKTEK MENGENAI KRIPTOGRAFI

Satria Putra Sajuthi – NIM : 13503099

Program Studi Teknik Informatika, Institut Teknologi Bandung
Jl. Ganesha 10, Bandung
E-mail : if13099@students.if.itb.ac.id

Abstrak

Makalah ini membahas mengenai kesenjangan yang muncul antara kriptografi secara standar dan teori dengan implementasi kriptografi di aplikasi nyata. Untuk membahas perbedaan tersebut, makalah ini mengambil contoh kasus IPSec. IP Security(IPSec) merupakan suatu protocol keamanan yang terletak pada layer Internet Protocol pada protocol stack dan banyak digunakan pada komunikasi data di jaringan. Makalah ini akan membahas mengenai kecenderungan user untuk mengabaikan peringatan-peringatan mengenai konfigurasi yang baik pada IPSec sehingga pada akhirnya akan memudahkan untuk melancarkan serangan-serangan terhadap user. Pada makalah ini juga dibahas mengenai bagaimana serangan-serangan yang dapat dilakukan terhadap konfigurasi tersebut.Pada bagian akhir akan dibahas mengenai alasan-alasan yang menyebabkan terjadinya situasi ini serta rekomendasi untuk perbaikan.

Kata kunci: IPSec, Encryption, ESP

1. Pendahuluan

Kebutuhan untuk melakukan enkripsi data sudah disadari dengan baik oleh komunitas riset mengenai kriptografi. Karena kesadaran itulah mereka menciptakan suatu standar dan spesifikasi enkripsi yang mengatur mengenai tingkat keamanan suatu data. Tetapi proses untuk mengadopsi standar yang dibuat tidak dapat dilakukan dalam waktu yang cepat dan juga bukan merupakan suatu proses yang mudah. Secara alami, dibutuhkan waktu untuk mengubah suatu teori menjadi standar, suatu standar menjadi produk, dan mensosialisasikan produk tersebut ke *end-user*. Masalah ini bertambah dengan adanya penyimpangan ide yang mula-mula disampaikan oleh peneliti kemudian sampai ke tangan pengembang dan akhirnya ke tangan pengguna. Kurangnya integritas ini dapat menyebabkan munculnya masalah-masalah lain yang berupa penyerangan pada celah keamanan yang terbuka. Fokus pada paper ini adalah mengenai proteksi integritas dan enkripsi pada IPsec yang merupakan suatu protokol penting dan banyak digunakan untuk menyediakan keamanan di *layer* IP(*Internet Protocol*).

2. Latar Belakang

2.1 IPSec

IPSec didefinisikan di RFC 2401-2412, memberikan keamanan/*security* pada *layer* IP(Internet Protocol). Implementasi dari IPSec sudah ada pada Sistem Operasi Microsoft Windows 2000 dan Windows XP serta pada Linux dengan kernel versi 2.6 ke atas. Selain itu banyak juga projek-projek open source yang mendukung implementasi dari IPSec. IPSec bekerja pada network layer dalam sistem 7 layer OSI. Tidak seperti SSL dan TLS yang bekerja pada transport layer, IPSec memberikan fleksibilitas karena Ipsec dapat digunakan untuk memproteksi protokol berbasis TCP maupun UDP. Kelemahannya adalah tingkat kompleksitas yang tinggi dan overhead processing serta tidak dapat bergantung kepada TCP untuk menjamin reliabilitas dan fragmentasi paket.

Protokol IPsec dapat dioperasikan dalam 2 mode yaitu mode *transport* dan mode *tunnel*. Dalam mode *tunnel*, proteksi kriptografi disediakan untuk keseluruhan IP *datagram*. Pada intinya, keseluruhan *datagram* ditambah *security field* diperlakukan sebagai suatu payload dari IP *datagram* yang lebih luar yang *header*-nya disebut sebagai *outer header*. Datagram

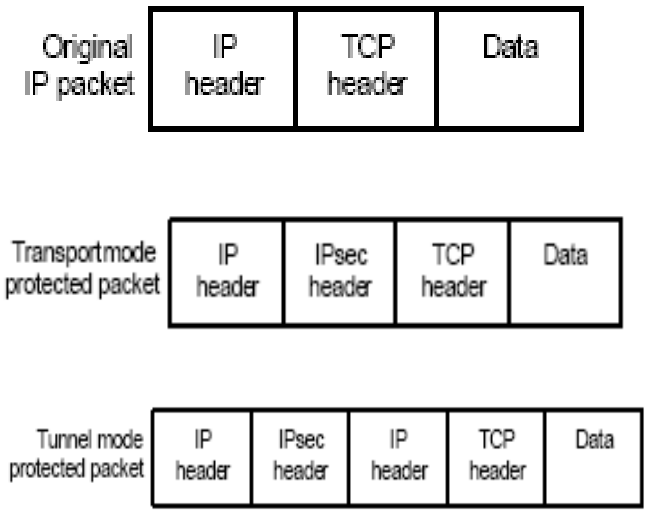
yang original atau inner disebut sebagai *encapsulated*(terbungkus) di dalam *outer* IP datagram. Dalam mode *tunnel*, pemrosesan IPsec dilakukan pada *security gateway* dan tidak ada *endpoint host*. *Gateway* tersebut dapat berupa *perimeter firewalls* atau *router*. Penggunaan *gateway* ini dimaksudkan agar host tidak perlu mengerti protokol IPsec dan keamanan didapatkan dari gateway-to-gateway atau keamanan di level gateway daripada end-to-end (di tingkat host). Jika pada mode tunnel keseluruhan paket IP dienkripsi, pada mode *transport* yang dienkripsi hanya *payload*(*message*) dari paket IP. Mode ini biasanya digunakan untuk komunikasi host-to-host.

IPsec menyediakan mekanisme otentikasi dan proteksi integritas dan/atau confidentiality services untuk data pada *network layer* melalui protokol AH dan ESP. ESP menyediakan mekanisme untuk confidentiality dan pada umumnya menggunakan algoritma blok cipher yang beroperasi pada mode CBC(*chain block chaining*). Pada mode tunnel, keseluruhan *inner IP datagram* dienkripsi dan membentuk bagian dari payload untuk *outer datagram*. IPSec dapat dikonfigurasi dengan berbagai macam cara. Salah satu cara yang paling banyak digunakan adalah membuat Virtual Private Network(VPNs), dimana IPsec dikonfigurasi untuk menggunakan protokol ESP dalam mode tunnel untuk menyediakan *confidentiality*. ESP juga dapat digunakan untuk menyediakan proteksi integritas untuk trafik VPN yang dapat juga disediakan oleh AH.

IPsec didesain untuk menyediakan keamanan
berbasis kriptografi yang memiliki karakteristik *interoperable* dan berkualitas. Layanan keamanan yang disediakan mencakup *access control,connectionless integrity*, *data origin authentication*, proteksi dari *replay attack* (*sequence integrity*), *data confidentiality* dan *traffic flow confidentiality*. Layanan tersebut disediakan pada IP *layer* sehingga mendukung proteksi untuk IP *layer* dan *layer* lain di atasnya.

ESP dalam mode tunnel menyisipkan informasi dalam bentuk header diantara outer IP header dan inner datagram yang dienkripsi. Header ESP ini mengindikasikan algoritma dan kunci yang digunakan untuk memproteksi payload dalam 32-bit field yang dinamakan *Security Payload Index(SPI)*. Header ESP ini juga mengandung 32-bit bilangan berurutan untuk mencegah pengulangan paket; ketika ESP digunakan dalam mode encryption-only, bilangan berurutan ini di-*ignored* oleh implementasi IPsec. ESP dalam mode tunnel dapat menambahkan field otentikasi setelah payload yang dienkripsi. *Field* ini berisi nilai MAC jika proteksi integritas digunakan.

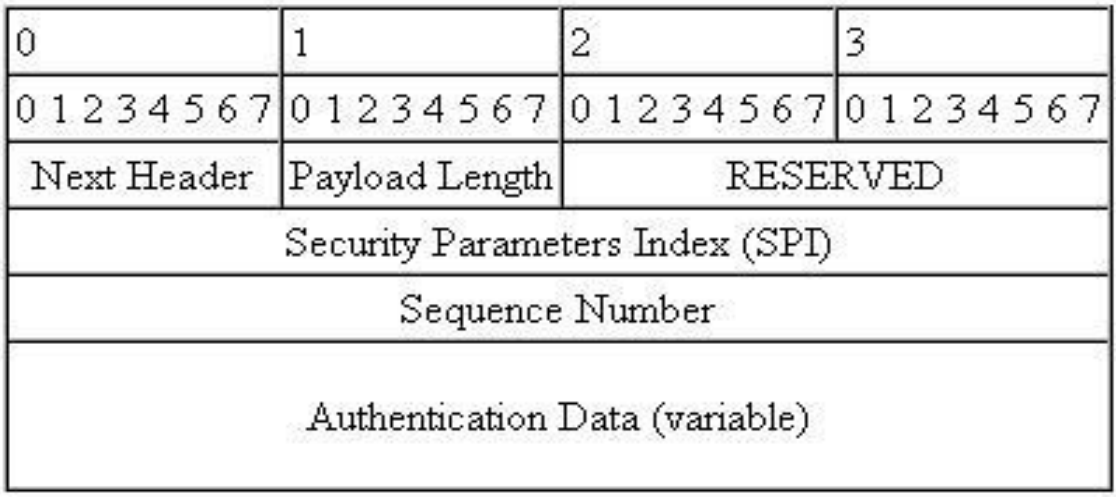
Protokol AH(Authentication Header) juga menyediakan proyeksi integritas dan pencegahan pengulangan paket walaupun scope dari aplikasinya berbeda dari ESP.



Gambar 1. Transport Mode dan Tunnel Mode

2.2 Authentication Header

AH didefinisikan sebagai protokol IP yang diberi nomor protokol 51. Dengan demikian, *field* protokol pada *header* paket IP yang diproteksi akan memiliki nilai 51 yang menunjukkan bahwa *header* yang mengikutinya adalah sebuah *header* AH. *Field* pada *header* AH diperlihatkan pada Gambar 2.



Gambar 2. Header AH

Pesan berisi informasi yang akan dikirim melalui jaringan akan dipecah-pecah menjadi paket-paket IP. Hal yang pertama dilakukan oleh IPsec dalam pemrosesan paket tersebut adalah memeriksa apakah paket tersebut akan diberikan layanan keamanan dengan protokol AH. Hal ini dilakukan dengan mencocokan paket yang bersangkutan dengan entri pada *Security Policy Database* (SPD). Apabila terdapat entri pada SPD, IPsec akan menentukan sebuah *Security Association* (SA)
berisi informasi mengenai algoritma MAC (*Message Authentication Code*) yang digunakan untuk paket tersebut sesuai dengan *source* dan *destination* paket yang bersangkutan. Setiap pasangan *source* dan *destination* memiliki Security Association yang berbeda yang ditentukan secara manual sebelumnya maupun secara otomatis melalui protokol IKE (*Internet Key Exchange*).

Informasi mengenai algoritma MAC yang digunakan dalam SA ditambahkan pada *header*
AH yaitu pada *field* SPI (Security Parameter
Index) setelah sebelumnya memberi nilai pada
field Next header, Payload length dan Reserved. *Field* Next header diberi nilai numerik dari tipe protokol dari data yang diproteksi (misalnya TCP atau UDP pada *transport mode*, atau IP pada *tunnel mode*).
Selain itu, *field* Payload length diberi nilai sesuai panjang *header* AH, kemudian *field* Reserved diberi nilai kosong (*null value*).

Field Sequence number pada awalnya diberi
nilai kosong, untuk kemudian ditambahkan satuper-satu untuk setiap paket terproteksi yang dikirimkan. Sequence number ini tidak dapat berulang sehingga apabila sudah mencapai nilai maksimum 232 maka SA untuk pengiriman pesan yang berjalan harus dimatikan terlebih dahulu dan dibentuk SA baru untuk pengiriman pesan lanjutan. Hal ini dimaksudkan untuk mencegah terjadinya *replay attack* atas paket pesan.

Field Authentication data adalah *field* yang panjangnya variabel berisi hasil dari fungsi pengecekan integritas yang disebut ICV (*Integrit Check Value*). Protokol AH pada IPsec tidak mendefinisikan (algoritma) *authenticator* tertentu tetapi mengharuskan implementasi *authenticator* untuk menjamin interoperabilitas antara implementasi IPsec yang berbeda-beda. Dua *authenticator* yang harus diimplementasikan adalah HMAC-SHA-1-96 13) dan HMAC-MD5-96 14). Kedua fungsi merupakan fungsi MAC dengan kunci rahasia yang keluarannya di*truncate* menjadi 96 bit .

ICV dihitung dengan menjalankan fungsi yang
didefinisikan oleh *authenticator* pada SA dengan parameter kunci rahasia dan seluruh paket IP yang terproteksi termasuk *header* AH. Namun demikian, *field-field*

pada *header* IP yang sifatnya berubah-ubah seperti TTL (*Time To Live*) dan *header checksum* serta *field* Authentication data itu sendiri diberi nilai kosong terlebih dahulu sebelum dimasukkan ke dalam fungsi. Setelah nilai ICV tersebut didapat maka nilainya dimasukkan dalam *field* Authentication data. Nilai-nilai *field* dari *header* IP yang berubah juga dimasukkan kembali ke dalam *field* masing-masing.

Pemrosesan AH selesai sampai titik ini dan paket IP yang terproteksi siap untuk dikirimkan.Bergantung pada ukuran paket yang dihasilkan,paket tersebut mungkin terfragmentasi dalam perjalanannya. Hal ini tidak menjadi masalah dan akan ditangani oleh pihak penerima.

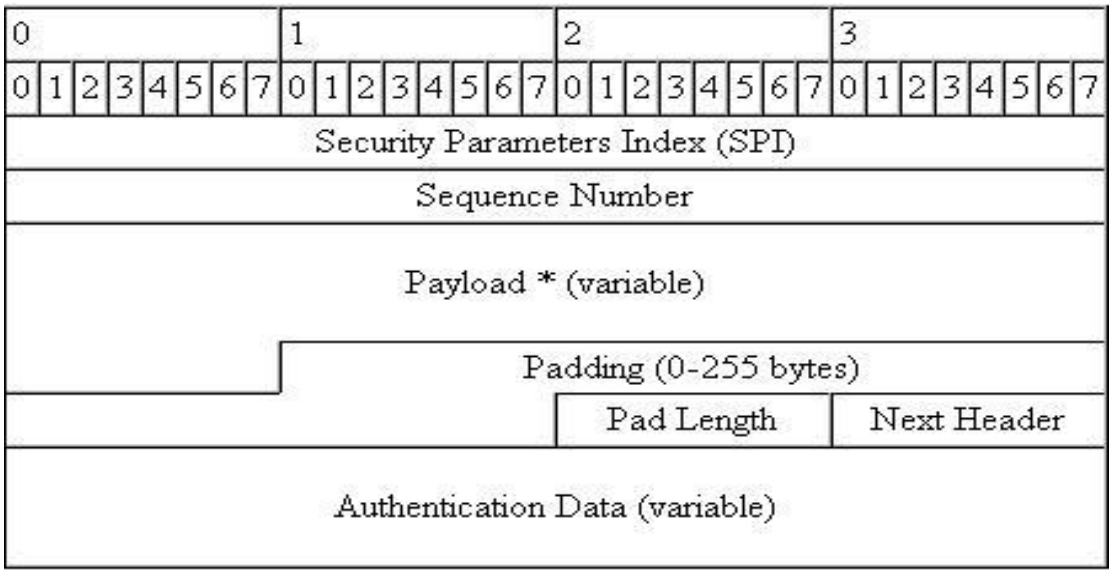
Pihak penerima pesan harus melakukan *reassembly* atas paket yang terfragmentasi.
Setelah paket utuh diterima, hal pertama yang
dilakukan adalah menentukan SA yang digunakan untuk memproteksi paket tersebut. SA yang digunakan ditentukan melalui *field*
Destination dan *field* Protocol pada header IP
serta *field* SPI pada *header* AH. Bila tidak ditemukan SA yang sesuai, maka paket tersebut di-*discard*.

Kemudian, pengecekan Sequence number dilakukan. Sama halnya dengan sebelumnya,
apabila paket yang diperiksa gagal memenuhinya maka paket tersebut di-*discard*. Paket yang gagal memenuhi pengecekan Sequence number menunjukkan bahwa paket terproteksi tersebut telah di-*replay* oleh pihak lain. Proses yang teakhir dilakukan adalah pengecekan ICV. ICV yang terdapat pada *field*
Authentication data pada *header* AH
disimpan dan *field* tersebut diberi nilai kosong.
Field pada *header* IP yang berubah-ubah juga diberi nilai kosong. Setelah itu, algoritma *authenticator* yang didefinisikan SA dijalankan pada paket tersebut dan hasilnya dibandingkan dengan ICV yang disimpan sebelumnya. Jika hasilnya sama maka paket tersebut terautentikasi dan paket IP yang diproteksi dapat di-*restore* dan diperlakukan sebagaimana paket IP biasanya.

2.3 Encapsulating Security Payload

ESP didefinisikan sebagai protokol IP yang diberi nomor protokol 50. Dengan demikian, *field* protokol pada *header* paket IP yang diproteksi akan memiliki nilai 50 yang menunjukkan bahwa *header* yang mengikutinya adalah sebuah *header*
ESP. *Field* pada *header* ESP diperlihatkan pada Gambar 3.

Sebagaimana halnya pada protokol AH, paket
yang akan diproses harus ditentukan terlebih
dahulu apakah akan diberi layanan keamanan
IPsec. Serupa dengan yang dilakukan pada AH,informasi paket yang diterima dicocokan pada entri di SPD (*Security Policy Database*). Apabila paket tersebut terdapat pada entri SPD maka paket tersebut akan diberikan layanan keamanan ESP. Sebuah *Security Association* (SA) kemudian akan ditentukan dari definisi yang ditentukan secara manual sebelumnya maupun secara otomatis dengan protokol IKE. Namun demikian, berbeda dengan protokol AH yang hanya menentukan *authenticator* (algoritma autentikasi yang digunakan), SA pada protokol ESP juga menentukan sebuah *encryptor* (algoritma enkripsi paket yang digunakan). Hal ini disebabkan ESP tidak hanya menyediakan fitur *data integrity* tetapi juga fitur data *confidentiality*. Fitur *data confidentiality* akan dijelaskan pada bagian



Gambar 3. ESP *Packet*

selanjutnya.

Nilai numerik informasi mengenai *Security Association* (SA) yang telah ditentukan kemudian dimasukkan ke dalam *field* Security Parameter Index (SPI) sebagaimana dilakukan pada protokol AH. Demikian pula dengan *field* Sequence number akan diinisiasi dengan nilai nol dan di-*increment* setiap pemrosesan paket pesan yang diproteksi. Seperti pada protokol AH, *field* ini diperlukan untuk untuk menjamin keamanan pengiriman paket dari *replay-attack*.

Field IV (Initialization Vector), Protected data, Pad, dan Pad length akan berisi nilai hasil pemrosesan *data confidentiality*. *Field-field* ini akan dijelaskan pada bagian berikutnya. Untuk sementara, *field-field* tersebut diasumsikan telah diisi nilai sebagaimana mestinya.

Kemudian, *field* Next header diberi nilai sesuai dengan nilai numerik protokol dari *payload* paket yang terproteksi. Dengan demikian, apabila ESP digunakan dalam *transport mode* dengan *payload* berupa paket TCP maka akan memiliki nilai 6 dan bila digunakan dalam *tunnel mode* (*payload* berupa paket IP) maka akan memiliki nilai 4.

Terakhir, sebagaimana dalam protokol AH, *field* Authentication data akan berisi nilai yang digunakan dalam pengecekan integritas paket atau ICV (*Integrity Check Value*). Seperti halnya pada protokol AH, *authenticator* yang harus diimplementasi dalam protokol ESP adalah HMAC-MD5-96 dan HMAC-SHA-96 untuk menjamin interoperabilitas antar implementasi IPsec yang berbeda-beda. Semua *field* dari *field* SPI hingga *field* Next header kemudian dimasukkan ke dalam fungsi *hash* yang telah ditentukan pada SA dan hasilnya dimasukkan ke dalam *field* Authentication data.

Sampai di sini pemrosesan fitur *data integrity* dari paket protokol ESP selesai dijalankan. Paket kemudian siap untuk dikirimkan melalui saluran komunikasi. Fragmentasi paket juga mungkin terjadi sebagaimana pada paket AH dan akan ditangani pula oleh pihak penerima paket tersebut.

Pihak penerima pesan harus melakukan *reassembly* atas paket yang terfragmentasi. Setelah paket utuh diterima, hal pertama yang dilakukan adalah menentukan SA yang digunakan untuk memproteksi paket tersebut. SA yang digunakan ditentukan melalui *field* Destination dan *field* Protocol pada *header* IP serta *field* SPI pada *header* ESP. Bila tidak ditemukan SA yang sesuai, maka paket tersebut di-*discard*.

Kemudian seperti pada protokol AH, pengecekan Sequence number dilakukan. Apabila paket yang diperiksa gagal memenuhinya maka paket tersebut di-*discard*. Paket yang gagal memenuhi pengecekan Sequence number menunjukkan bahwa paket terproteksi tersebut telah di-*replay* oleh pihak lain. Proses berikutnya yang dilakukan adalah pengecekan ICV. ICV yang terdapat pada *field* Authentication data pada *header* ESP. Setelah itu, algoritma *authenticator* yang didefinisikan SA dijalankan pada paket tersebut dari *field* SPI hingga *field* Next header dan hasilnya dibandingkan dengan ICV yang disimpan sebelumnya. Jika hasilnya sama maka paket tersebut terautentikasi.

Pemrosesan *data integrity* untuk paket yang diterima selesai sampai di sini dan akan dilanjutkan pada pemrosesan *data confidentiality* yang akan dijelaskan pada bagian selanjutnya.

2.4 Enkripsi mode CBC dalam ESP

Pertama-tama, inner datagram yang akan diproteksi diperlakukan sebagai rangkaian byte. Rangkaian ini di-padding dengan pola bytes tertentu dan ditambahkan dengan satu byte header. Padding dikonstruksi sedemikian rupa sehingga jumlah total bytes dalam rangkaian byte merupakan kelipatan dari jumlah byte dalam satu blok sesuai dengan algoritma enkripsi.(Misalnya 8 byte untuk DES dan 16 byte untuk AES), dan padding tersebut dapat dipisahkan tanpa menyebabkan ambiguitas.

Asumsikan rangkaian byte setelah padding terdiri dari q blok, setiap blok terdiri dari n bit(dimana n = 64 untuk DES dan n = 128 untuk AES). Blok-blok tersebut dinotasikan dengan P1,P2,.....,Pq. Simbol K digunakan untuk menyatakan kunci yang digunakan untuk algoritma blok cipher dan e_K dan d_K digunakan untuk menotasikan fungsi enkripsi dan dekripsi suatu blok dengan menggunakan kunci K. Kemudian enkripsi mode CBC dalam ESP berlanjut sebagai berikut. Pertama, dipilih secara acak initialization vector(IV) sebesar n-bit. Kemudian blok ciphertext dibangkitkan menurut persamaan:

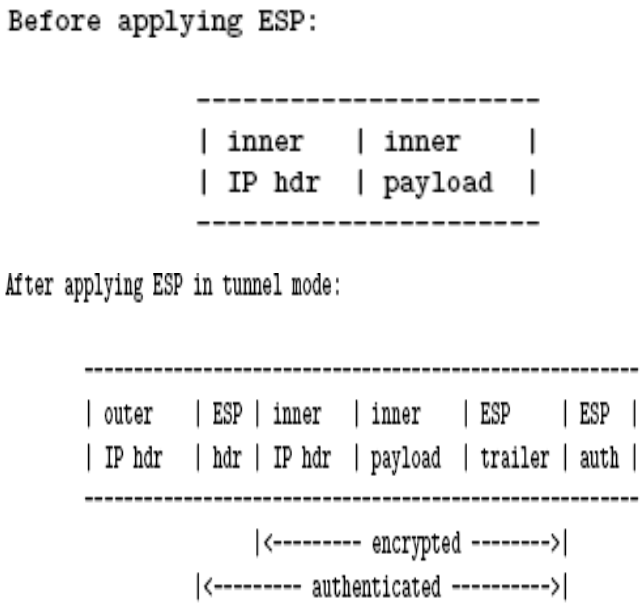
$$C_0 = IV, \quad C_i = e_K(C_{i-1} \oplus P_i), \quad (1 \le i \le q).$$

Bagian yang terenkripsi dari outer datagram didefinisikan sebagai rangkaian blok sejumlah q + 1 dari $C_0,C_1,....,C_q$.

Pada security gateway (juga mempunyai kunci K), payload dari outer datagram dapat didapatkan kembali dengan menggunakan persamaan:

$$P_i = C_{i-1} \oplus d_K(C_i), \quad (1 \le i \le q).$$

Padding dan header dapat dilepaskan untuk mendapatkan inner datagram yang original.



Gambar 4. Paket IP sebelum dan sesudah dienkripsi

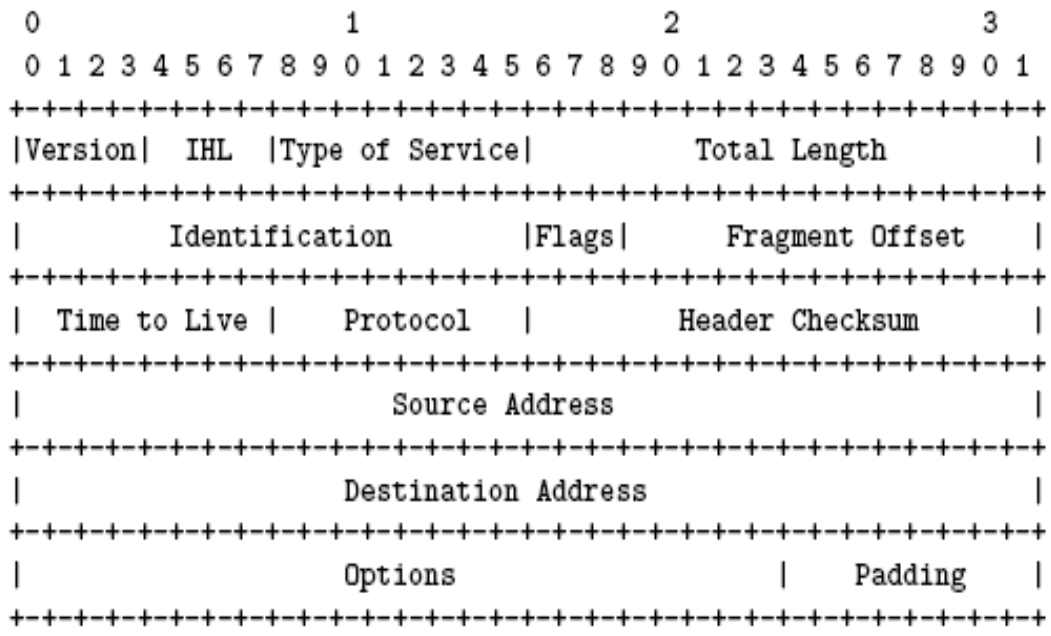
2.5 Bit Flipping Attack

Mode CBC mempunyai kelemahan yang dikenal dengan nama *bit flipping vulnerability*. Misalkan attacker mempunyai ciphertext dari $C_0,C_1,....,C_q$ kemudian mem-flip bit spesifik j dalam C_{i-1} dan memasukkan ciphertext yang telah dimodifikasi ke dalam network. Setelah diterima dan didekripsi, bit yang di-flip

akan ditransformasikan menjadi bit yang bersesuain dalam blok plaintext P_i. Hal ini dapat dilihat dengan mengamati persamaan dekripsi $Pi = Ci_i1+dK(Ci)$. Dari persamaan dapat dilihat bahwa attacker dapat mengkontrol perubahan dengan cari mem-flip bit pada cipherteks dan memasukkan ciphertext yang telah dimodifikasi.

Masalah bagi attacker adalah perubahan pada ciphertext mengakibatkan perubahan pada plaintext yang bersesuaian secara acak. Jika perubahan dilakukan pada inisialization vector maka tidak ada perubahan yang akan terjadi pada plaintext. Serangan jenis ini dapat dicegah dengan menggunakan mekanisme proteksi integritas seperti algoritma MAC.

2.6 IP Datagram Header



Gambar 5. IP Datagram Header

Eksekusi dari serangan pada ESP dalam mode tunnel tergantung pada struktur dari header IP datagram dan urutan field-field header diproses. Layout dari header IP datagram dapat dilihat pada gambar 5.

Internet Header Length(IHL) field mempunyai panjang 4 bit dan bernilai antara 5 sampai 15. Field ini menandakan panjang dari header. Nilai 5 menandakan panjang header adalah 20 bytes dan tidak ada tambahan bit opsional. Jika IHL mempunyai nilai yang lebih besar dari 5, maka ada tambahan bit opsional pada field Options. Field ini dapat bernilai sampai 40 bytes dan dapat digunakan untuk membawa instruksi tambahan atau informasi.

Field protocol mempunyai panjang 8 bits. Field ini mengindikasikan protokol apa yang digunakan pada layer yang lebih atas yang dibawa ke *IP datagram payload*. Himpunan nilai yang mungkin diberikan oleh host ketika membuat datagram tergantung dari konfigurasi host dan protokol yang didukung. Minimal protokol yang didukung mencakup ICMP, TCP dan UDP. Umumnya akan didukung 2 sampai 3 protokol lagi selain protokol di atas. Ketika suatu IP datagram mencapai destination host, field protokol akan diperiksa. Setelah itu akan diteruskan ke layer yang lebih atas sesuai dengan nilai protokol pada field tersebut. Jika field protocol mempunyai nilai yang tidak didukung oleh host yang menerima, maka akan mengirimkan pesan ICMP "*protocol unreachable*".

Field Header checksum dengan panjang 16 bit (2 byte) dibentuk dengan menginterpretasi header termasuk option field sebagai rangkaian 16 bit words.

Rangkaian tersebut dijumlahkan dengan menggunakan aritmatik one's complement dan mengambil hasilnya dalam bentuk one's complement. Header checksum menyediakan verifikasi apakah informasi yang digunakan ketika dilakukan pemrosesan datagram benar atau tidak. Jika header checksum bernilai gagal, maka datagram akan dibuang oleh entitas yang mendeteksi *error*.

Untuk mengerti serangan ini, harus lebih dahulu dimengerti langkah-langkah yang dilakukan oleh Internet Protocol(IP) ketika memproses suatu datagram. Dalam Linux, langkah-langkah tersebut adalah sebagai berikut. Pertama kali ketika datagram diterima akan dilakukan basic check pada field Version dan field IHL. Selanjutnya akan dilakukan pengecekan pada field Header Checksum. Setelah itu, untuk mengetahui panjang datagram dapat dilakukan dengan mengecek field Total Length. Datagram akan dibuang jika salah satu dari pengecekan di atas gagal. Selanjutnya akan dilakukan pemrosesan pada field option jika pada field IHL tersedia options. Setelah pemrosesan berhasil, maka akan dilakukan routing. Setelah paket sampai ke destination address, maka field Protocol akan diperiksa untuk menentukan protokol tujuan pada layer yang lebih tinggi. Field TTL akan diperiksa jika paket didistribusikan melalui router, jika TTL mencapai nilai 0, maka paket tersebut akan dibuang.

Dari penjelasan di atas, serangan tersebut datagram harus memastikan kesesuaian dengan payload dengan header checksum.

2.7 ICMP

Bagian ini akan menjelaskan mengenai ICMP(Internet Control Message Protocol). ICMP merupakan bagian vital dari implementasi Internet Protocol. ICMP akan melaporkan masalah yang terjadi pada network ke host, rute yang dicoba dan sekumpulan diagnosa yang dilakukan.

Jika ada datagram bermasalah yang diterima oleh host, host tersebut akan membuat ICMP message. Message ini akan berisi IP header dari datagram yang bermasalah dan panjang bytes dari payload.

3. Serangan berbasis Destination Address Rewriting

Pada bagian ini akan dibahas mengenai serangan two phase. Kasusnya pada Encryption only ESP mode tunnel. Asumsi attacker mengetahui IP target dan destination address(DestAddr) dari inner datagram target. AttAddr menandakan alamat attacker.

3.1 *First Phase*

Fase pertama dari serangan ini didasarkan pada manipulasi field destination address dalam inner datagram. Field ini terletak pada word ke 5 pada ip header dan membentuk plaintext 32 bit pertama dalam blok plaintext ketiga dalam rangkaian bit-bit plaintext mode CBC. 32 bit kedua dalam blok ini merupakan 32 bit pertama dari payload inner datagram. Langkah-langkah serangan adalah sebagai berikut:

1. Dapatkan ESP protected outer datagram dari jaringan.
2. Modifikasi blok ciphertext C_2. 32 bit pertama dari blok ini di XOR kan dengan 32 bit mask M = DestAddr XOR AttAddr untuk mendapatkan blok C'_2.
3. **repeat**:

- Modifikasi blok C'_2 pada 32 bit terakhir dengan merubah bit-bit tersebut secara acak. C''_2 merupakan blok yang sudah diubah.
- Masukkan datagram yang sudah diubah ke dalam jaringan.

Until datagram diterima oleh attacker di AttAddr.

Setiap injected datagram sekarang mempunyai AttAddr sebagai destination address dari inner datagram. Jadi ketika security gateway menerima outer datagram yang telah dimodifikasi kemudian mendekripsinya, hasil dekripsi tersebut kemudian akan dikirimkan ke alamat attacker.

3.2 *Second Phase*

Attacker yang telah berhasil melakukan fase pertama tidak perlu mengulanginya lagi untuk mendapatkan hasil dekripsi dari inner datagram. Attacker akan

menggunakan kembali C_0,C_1,C''_2,C_3 dari outer datagram yang didapatkan dari fase pertama kemudian membaginya menjadi q-6 blok ciphertext dari payload yang terenkripsi dari datagram target dan diakhiri dengan 3 blok C_{q-2},C_{q-1},Cq dari target awal. Ditambahkan dengan padding jika diperlukan.

Attacker kemudian menggunakan rangkaian byte yang dimodifikasi sebagai payload terenkripsi dari outer datagram. Konstruksi ini(seperti dalam gambar 6) memastikan bahwa pada saat dekripsi dilakukan oleh security *gateway*, *payload* akan di-padding dengan benar dan diinterpretasikan sebagai *inner datagram* dengan header yang valid dan *destination address* yang sesuai dengan AttAddr. *Datagram* ini akan dirutekan kepada mesinnya *attacker*. Dari *datagram* yang diterima, sejumlah 64*(q-6) bit plaintext dari target *datagram* baru dapat di-*recover*.

4. Serangan berbasis IP Options Processing

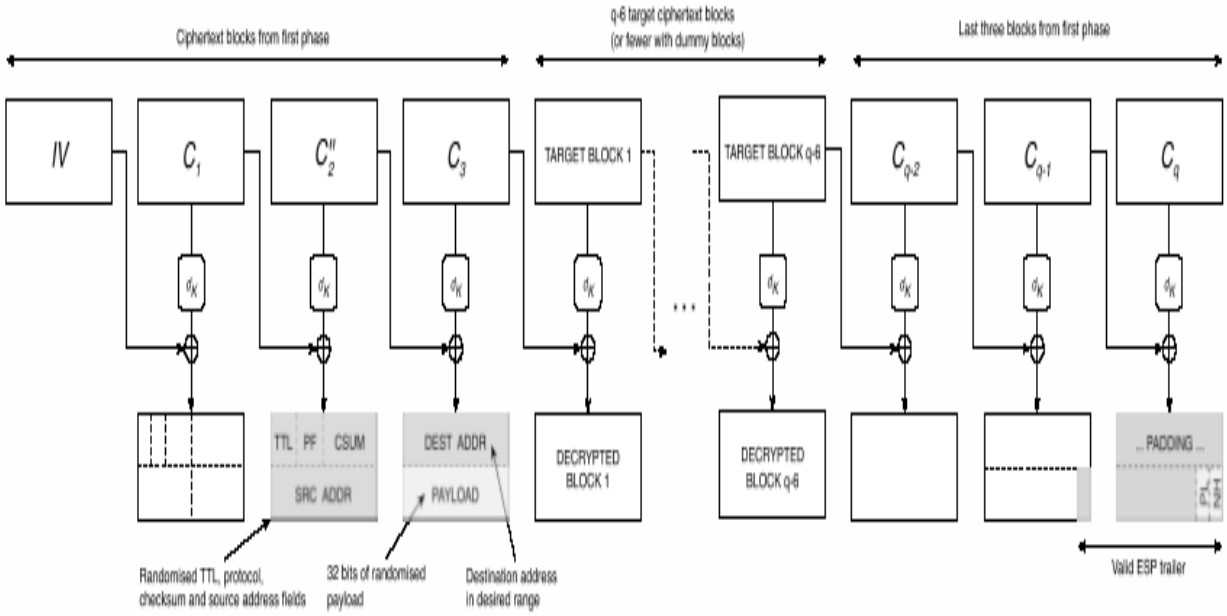
Serangan ini mengeksploitasi suatu cara implementasi IP yang membuat ICMP messages ketika memproses option field yang tidak sesuai format.

4.1 *First Phase*

Serangan ini seperti diilustrasikan dalam gambar 7, dapat dirunut dengan langkah-langkah berikut:

1. Capture ESP-protected outer datagram dari network.
2. Modifikasi byte pertama dari blok IV, XORkan dengan mask yang meningkatkan IHL sehingga bernilai lebih besar dari 5; hasilnya menjadi blok C'_0.
3. **Repeat**:

- Modifikasi 32 bit terakhir dari blok blok C_2 dengan cara menset nilainya dengan nilai acak; C'_2



Gambar 6. Skema phase two

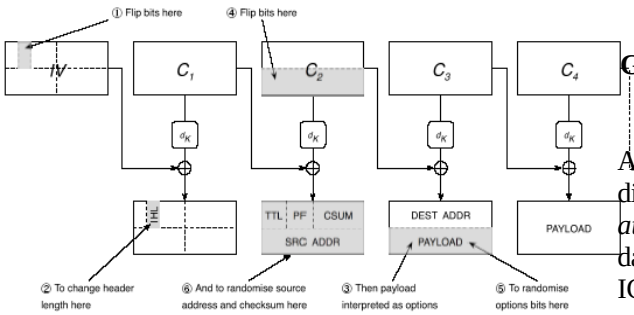
menyatakan blok yang telah dimodifikasi.

- Siapkan datagram yang dimodifikasi yang identik dengan datagram yang didapatkan pada langkah 1 kecuali blok C_0 dan C_2 diganti dengan C'_0 dan C'_2. Masukkan datagram tersebut ke network.

Until mendapatkan *message* ICMP.

4.2. *Second Phase*

Pada fase pertama di atas menghasilkan ICMP message yang berisi header dan segment payload dari inner datagram yang dimodifikasi. Jumlah payload yang di-recover tergantung dari implementasi ICMP di gateway. Trik yang digunakan mirip dengan fase kedua pada bagian 3.2 untuk mempercepat proses recovery dari payload sisa.

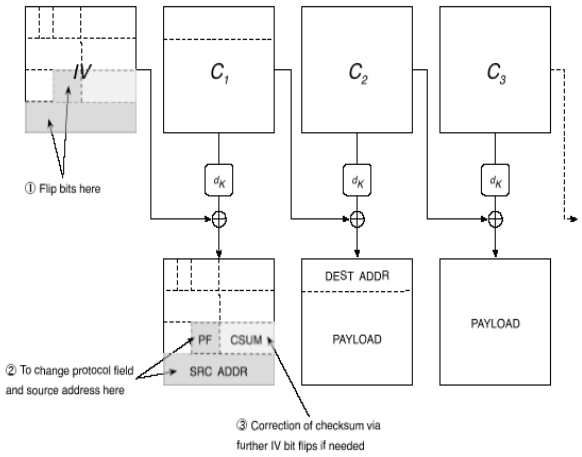


Gambar 7. Modifikasi Inner Header pada Option Processing Attack

5 Serangan berbasis manipulasi field protokol

5.1 *First Phase*

Pada bagian 2.6 tentang IP Datagram Header telah dijelaskan bahwa field protokol terletak pada byte kedua dari word ketiga dari IP header dan terletak pada blok plaintext pertama dalam rangkaian blok plaintext yang akan dienkripsi dalam mode CBC oleh ESP. *Attacker* memodifikasi isi dari field protokol dari *inner datagram* dengan cara mem-*flip* bit-bit yang bersesuaian dalam IV sehingga *field* tersebut sama dengan protokol tingkat atas yang tidak didukung oleh host tujuan. Biasanya, hal ini cukup dilakukan dengan mem-*flip most significant bit* dari *field* protokol. Ketika inner datagram tiba di host tujuan, maka akan mengirimkan pesan ICMP "*protocol unreachable*". Payload dari pesan ICMP ini akan mengandung *header* dan *segment* dari *payload inner datagram*. Jika ICMP *message* ini dapat dicapture oleh *attacker*, maka informasi mengenai *plaintext* dapat diperoleh.



Gambar 8. Modifikasi *field inner header* pada *protocol field attack*

Ada dua masalah utama yang harus dibereskan oleh *attacker*. Pertama, *attacker* harus merubah *source address* dari *inner datagram*, jika tidak maka ICMP akan merespon dengan mengirimkan *message* melalui tunnel Ipsec kepada sumbernya semula dari inner datagram. Kedua, attacker harus menyesuaikan header checksum sehingga mengandung nilai yang benar untuk inner header yang telah dimodifikasi.

Analisis dari algoritma IP *header checksum* menunjukkan bahwa mem-*flip* 1 bit dalam *header* mempunyai efek XOR 16 bit *checksum* dengan satu dari 17 kemungkinan *masks*, *mask* ini mempunyai distribusi geometrik. Tabel 1 menunjukkan kemungkinan mask beserta probabilitasnya untuk kasus bit flip pada most significant position dalam 16 bit checksum. Tabel Tj untuk posisi j dimana (0<=j<16) dapat diperoleh dengan merotasikan ke kiri 16 bit melalui 15-j posisi pada masak di tabel 1.

Mask	Probability
0000000000000001	$1/2$
0000000000000011	$1/4$
0000000000000111	$1/8$
⋮	⋮
1111111111111111	2^{-16}
1111111111111110	2^{-16}

Tabel 1. Tabel mask dan probablitasnya

5.2 Second Phase

Seperti pada bagian 4, ketika fase pertama telah selesai, fase kedua me-*recover content* dari sisa *target datagram*. Karena hubungan antara header field dengan batasan blok plaintext, fase kedua ini memerlukan ICMP untuk membawa minimal 28 bytes payload dari inner datagram.

6. Penanganan Serangan

Cara termudah untuk menjamin kekebalan terhadap serangan-serangan yang telah dideskripsikan sebelumnya adalah dengan cara mengkonfigurasi ESP untuk menggunakan *confidentiality* dan *integrity protection*. Penggunaan integrity protection dapat mementahkan serangan-serangan tersebut karena datagram yang dimodifikasi langsung ditolak dengan probabilitas yang tinggi.

Cara lainnya adalah dengan menggunakan protokol AH bersama ESP untuk memberikan integrity protection. Cara ini harus dikonfigurasi dengan hati-hati. Konfigurasi AH pada transport mode dan ESP pada tunnel masih mempunyai celah. Alternatif lain adalah dengan cara mematikan error reporting atau dengan memfilter message dengan firewall.

7. Kesimpulan

IPsec merupakan sekumpulan protokol yang rumit. Untuk mendapatkan manfaat keamanan dari IPsec kuncinya terletak pada konfigurasi, policy, seleksi algoritma dan manajemen kunci yang baik. IPsec standar dan penerapannya masih memungkinkan konfigurasi yang tidak aman untuk dipilih.

Konfigurasi yang tidak aman dalam teori maupun praktek seharusnya dikeluarkan dari IPsec standar. Gap antar standar dan pengguna memperbolehkan suatu konfigurasi yang terlalu besar untuk dijembatani walaupun banyak *warning* yang tertulis di dalam RFC.

Gap antara standar dan implementasi terjadi karena belum adanya standar baku yang mengimplementasikan suatu protokol yang ditulis di dalam RFC. Gap ini akan terus membesar bila diteruskan ke end user, karena end user tidak membaca RFC.

8. Daftar Referensi

1. R. Atkinson, \IP Encapsulating Security Payload (ESP)", RFC 1827, August 1995.

2. F. Baker, \Requirements for IPv4 Routers", RFC 1812, June 1995.

3. M. Bellare, T. Kohno and C. Namprempre, \Breaking and provably repairing the SSH authenticated encryption scheme: A case study of the Encode-then-Encrypt-and-MAC paradigm." *ACM Transactions on Information and System Security (TISSEC)*, Vol. 7, No. 2, May 2004, pp. 206{241.

4. M. Bellare and C. Namprempre, \Authenticated Encryption: Relations among notions and analysis of the generic composition paradigm." In *T. Okamoto (ed.), Advances in Cryptology { ASIACRYPT 2000*, LNCS Vol. 1976, Springer-Verlag, 2000, pp. 531{545.

5. M. Bellare and P. Rogaway, \Encode-then-encipher encryption: How to exploit nonces or redundancy in plaintexts for e±cient cryptography." In *T. Okamoto (ed.), Advances in Cryptology { ASIACRYPT 2000*, LNCS Vol. 1976, Springer-Verlag, 2000, pp.317{330.

6. S. Bellovin, \Problem Areas for the IP Security Protocols", in *Proceedings of the Sixth Usenix Unix Security Symposium*, pp. 1{16, San Jose, CA, July 1996.

7. D.Bleichenbacher, \Chosen Ciphertext Attacks Against Protocols Based on the RSA Encryption Stan-dard PKCS #1", in *H. Krawczyk (ed.), Advances in Cryptology { CRYPTO 1998*, LNCS Vol. 1462,
Springer-Verlag, 1998, pp. 1{12.

8. N. Borisov, I. Goldberg and D. Wagner, \Intercepting Mobile Communications: The Insecurity of 802.11", in *Proc. MOBICOM 2001*, ACM Press, 2001, pp. 180{189.

9. B.Canvel, A.P. Hiltgen, S. Vaudenay and M. Vuagnoux, \Password Interception in a SSL/TLS Chan-nel," in *D. Boneh (ed.), Advances in Cryptology { CRYPTO 2003*, LNCS Vol. 2729, Springer-Verlag, 2003, pp. 583{599

10. N. Doraswamy and D. Harkins. *IPsec: the new security standard for the Internet, Intranets and Virtual Private Networks (second edition)*, Prentice Hall PTR, 2003.

11. N. Ferguson and B. Schneier, \A cryptographic evaluation of IPsec." Unpublished manuscript, Feb. 1999. Available from http://www.schneier.com/paper-ipsec.html.

12. S. Frankel, R. Glenn and S. Kelly, \The AES-CBC Cipher Algorithm and Its Use with IPsec", RFC 3602, Sept. 2003.

13. S. Frankel, K. Kent, R. Lewkowski, A.D. Orebaugh, R.W. Ritchey and S.R. Sharma, \Guide to Ipsec VPNs", NIST Special Publication 800-77 (Draft), January 2005.

14. J. Katz and M. Yung, \Unforgeable encryption and chosen ciphertext secure modes of operation." In *B. Schneier (ed.), FSE 2000*, LNCS Vol. 1978, Springer-Verlag 2001, pp. 284{299.

15. S. Kent and R. Atkinson, \Security Architecture for the Internet Protocol", RFC 2401, Nov. 1998.

16. S. Kent and R. Atkinson, \IP Encapsulating Security Payload (ESP)", RFC 2406, Nov. 1998.

17. S. Kent and K. Seo, \Security Architecture for the Internet Protocol", RFC 4301 (obsoletes RFC 2401), Dec. 2005.

18. S. Kent, \IP Encapsulating Security Payload (ESP)", RFC 4303 (obsoletes RFC 2406), Dec. 2005.

19. H. Krawczyk, \The Order of Encryption and Authentication for Protecting Communications (Or:How Secure Is SSL?)", in *J. Kilian (ed.), Advances in Cryptology { CRYPTO 2001*, LNCS Vol. 2139,
Springer-Verlag 2001, pp. 310{331.

20. Internet Protocol, RFC 791, Sept. 1981.

21. C. Madson and N. Doraswamy, \The ESP DES-CBC Cipher Algorithm With Explicit IV", RFC 2405,Nov. 1998.

22. J. Manger, \A chosen ciphertext attack on RSA Optimal Asymmetric Encryption Padding (OAEP) as standardized in PKCS #1 v2.0," in *J. Kilian (ed.), Advances in Cryptology { CRYPTO 2001*, LNCS Vol. 2139, Springer-Verlag 2001, pp. 230-238.

23. C.B. McCubbin, A.A. Selcuk and D. Sidhu, \Initialization vector attacks on the IPsec protocol suite."In *9th IEEE International Workshops on Enabling Technologies: Infrastructure for Collaborative Enterprises (WETICE 2000)*, IEEE Computer Society, 2000, pp. 171{175.