STUDI PENERAPAN BEBERAPA ALGORITMA KRIPTOGRAFI PADA IPsec

Muhammad Fiqri Muthohar – NIM : 13506084

Program Studi Teknik Informatika, Sekolah Teknik Elektro dan Informatika, Institut Teknologi Bandung
Jl. Ganesha 10, Bandung
E-mail : fiqri@arc.itb.ac.id , viqronization@gmail.com

Abstrak

Makalah ini membahas tentang beberapa penerapan algoritma kriptografi pada protokol *IPsec* (IP secure). IPsec sendiri merupakan sebuah kumpulan protokol jaringan yang menyediakan enkripsi dan autentikasi untuk pesan IP pada layer jaringan di internet. Dua protokol mayor pada IPsec adalah Authentication Header(AH) yang menyediakan autentikasi dan proteksi terhadap integritas dan Encapsulating Security Payload(ESP) yang menyediakan enkripsi beserta autentikasi serta proteksi integritas IP payload yang sifatnya optional.

Pada makalah ini akan dibahas tentang algoritma-algoritma kriptografi yang digunakan meliputi pembahasan beberapa algoritma yang digunakan pada enkripsi data pada IPsec ini yaitu HMAC(*keyed-Hash Message Authentication Code*)-SHA1(*Secure Hash Algorithm*), TripleDES(*Data Encryption Standard*)-CBC, dan HMAC-MD5-96. Dalam makalah ini juga akan dibahas bagaimana algoritma kriptografi itu diterapkan dalam proses pengiriman data. Tentu saja IPsec tidak lepas dari usaha-usaha dari pihak-pihak lain untuk membongkar sistem keamananannya, dalam makalah ini akan dibahas juga beberapa jenis serangan yang pernah dilakukan terhadap sistem keamanan dari IPsec ini terutama pada bagian enkripsi data dari IPsec ini.

***Kata kunci**: IPsecure, jaringan, enkripsi, algoritma, serangan, HMAC-SHA1,TripleDES-CBC, HMAC-MD5-96.*

1. Pendahuluan

Pengiriman data dan penyimpanan data melalui media elektronik memerlukan suatu proses yang dapat menjamin keamanan dan keutuhan dari data yang dikirimkan tersebut. Data tersebut harus tetap rahasia selama pengiriman dan harus tetap utuh pada saat penerimaan di tujuan. Untuk memenuhi hal tersebut, dilakukan proses penyandian (enkripsi dan dekripsi) terhadap data yang akan dikirimkan. Enkripsi dilakukan pada saat pengiriman dengan cara mengubah data asli menjadi data rahasia sedangkan dekripsi dilakukan pada saat penerimaan dengan cara mengubah data rahasia menjadi data asli. Jadi data yang dikirimkan selama proses pengiriman adalah data rahasia, sehingga data asli tidak dapat diketahui oleh pihak yang tidak berkepentingan. Data asli hanya dapat diketahui oleh penerima dengan menggunakan kunci rahasia.

Pertukaran data dalam sebuah jaringan tentu tidak lepas dengan apa yang disebut dengan protokol jaringan. Masing-masing protokol memiliki fungsi dan spesifikasi yang spesifik ketika dibuat atau diusulkan. Beberapa contoh protokol yang ada seperti TCP (*Transmission Control Protocol*), IP (*Internet Protocol*), SMTP (*Simple Mail Transfer Protocol*), HTTP (*HyperText Transfer Protocol*), dan masih banyak lagi protokol-protokol yang ada.

Dalam masing-masing protokol pengiriman data berbeda-beda satu sama lain, dan data yang dikirimkan terkadang berupa plainteks sehingga jika dilakukan pengendusan paket (*packet sniffing*) maka data yang dikirimkan dapat dilihat dan jika beberapa paket tersebut digabungkan maka dapat didapat kumpulan informasi yang cukup banyak karena tidak adanya proses enkripsi data.

Internet Protocol Security (IPsec) adalah sebuah kumpulan protokol yang digunakan untuk mengamankan komunikasi *Internet Protocol* (IP) dengan autentikasi dan enkripsi pada setiap aliran data paket IP. Implementasi IPsec beroperasi pada sebuah host atau sebuah *security gateway environment* yang memberikan proteksi kepada trafik IP. Proteksi yang ditawarkan berdasar pada requirements yang didefinisikan oleh sebuah *Security Policy Database* (SPD) yang terpercaya dan terpelihara oleh seorang pengguna atau administrator sistem, atau oleh sebuah aplikasi yang beroperasi di dalam batasan yang diberikan oleh salah satu diantara user maupun administrator sistem. Secara umum, paket dipilih dalam satu diantara tiga mode pemrosesan yang ada yang berdasar pada IP dan informasi header layer transport (Selector) yang kemudian dibandingkan dengan entri pada database (SPD). Setiap paket kemudian diberikan servis sekuritas IPsec, dibuang, atau diizinkan untuk melalui IPsec, berdasar pada basisdata kebijakan yang dapat diterapkan sesuai dengan apa yang diidentifikasikan oleh Selector.

2. IPsec

2.1. Hal yang dilakukan oleh IPsec

IPsec menyediakan servis keamanan pada layer IP dengan memungkinkan sistem untuk memilih protokol keamanan yang dibutuhkan, memilih algoritma yang akan digunakan untuk servis, dan meletakkan

ditempatnya kunci kriptografi yang dibutuhkan untuk menyediakan servis yang diminta. IPsec dapat digunakan untuk melindungi satu atau lebih ”jalur” antara pasangan host, antara pasangan *gateway* keamanan, ataupun antara sebuah *gateway* keamanan dengan sebuah host. *Gateway* keamanan ini digunakan dalam dokumentasi IPsec untuk mengacu pada sebuah *intermediate system* yang mengimplementasikan protokol IPsec. Sebagai contoh, sebuah *router* atau sebuah tembok api (*firewall*) yang mengimplementasikan IPsec disebut sebagai *gateway* keamanan.

Set servis keamanan yang dapat disediakan oleh IPsec antara lain adalah kontrol akses, integritas nirkoneksi, autentikasi asal data, penolakan replay paket (sebuah bentuk dari integritas urutan parsial), kerahasiaan (enkripsi), dan kerahasiaan laju trafik secara terbatas. Karena servis-servis tersebut berada pada layer IP, mereka dapat digunakan pada layer protokol yang lebih tinggi, seperti TCP, UDP, ICMP, BGP, dan yang lain.

IPsec DOI juga mendukung negoisasi kompresi IP, termotivasi oleh observasi yang menunjukkan ketika enkripsi dilakukan dalam IPsec, dapat mencegah kompresi efektif oleh protokol layer yang lebih rendah.

2.2. Cara Kerja IPsec

IPsec menggunakan dua protocol untuk menyediakan sekuritas trafik, yaitu *Authentication Header* (AH) dan *Encapsulating Security Payload* (ESP). Kedua protocol ini dijelaskan lebih lanjut pada RFC masing-masing protokol [KA98a, KA98b].

IP Authentication Header (AH) [KA98a] menyediakan integritas nirkoneksi, autentikasi asal data, dan sebuah servis anti-replay yang sifatnya pilihan.

Protokol Encapsulating Security Payload (ESP) dapat memberikan keamanan (enkripsi) dan keamanan laju trafik secara terbatas. ESP dapat juga menyediakan integritas nirkoneksi, autentikasi asal data, dan servis anti-replay. (Satu atau kumpulan servis sekuritas tersebut harus diaplikasikan setiap kali ESP diinvokasi).

Kedua-duanya, AH dan ESP, adalah kendaraan untuk kontrol akses, berdasarkan distribusi kunci kriptografi dan manajemen laju trafik relatif terhadap protokol keamanan tersebut.

Protokol tersebut dapat diaplikasikan secara sendiri atau dengan kombinasi satu dengan yang lainnya untuk menyediakan sekumpulan servis sekuritas pada IPv4 dan IPv6 sesuai dengan yang diinginkan. Masing-masing protokol mendukung dua mode penggunaan yaitu mode transport dan mode tunnel. Dalam mode transport, protokol menyediakan proteksi terutama untuk protokol layer yang lebih tinggi. Sedangkan dalam mode tunnel, protokol diaplikasikan pada paket IP yang di-tunnel.

IPsec mengizinkan pengguna (atau administrator sistem) untuk mengontrol granularitas di mana servis keamanan ditawarkan. Sebagai contoh, seseorang dapat menciptakan sebuah tunnel terenkripsi tunggal untuk membawa semua trafik diantara dua gateway keamanan atau tunnel terenkripsi yang terpisah dapat dibuat untuk setiap koneksi TCP di antara pasangan host yang saling berkomunikasi melalui gateway tersebut. Manajemen IPsec harus menggabungkan fasilitas untuk menspesifikasikan:

- Servis sekuritas mana yang akan digunakan dan dengan kombinasi seperti apa
- Granularitas di mana sebuah proteksi sekuritas seharusnya diterapkan
- Algoritma yang digunakan untuk menghasilkan sekuritas berdasar kriptografi

Karena servis sekuritas tersebut digunakan untuk berbagi nilai rahasia (kunci kriptografi), IPsec bergantung pada sejumlah mekanisme terpisah untuk meletakkan kunci tersebut pada tempatnya. Kunci tersebut digunakan untuk autentikasi/intregritas dan servis enkripsi.

2.3. Beberapa cara penerapan IPsec

Terdapat beberapa jalan untuk mengimplementasikan IPsec di sebuah host atau pada konjungsi dengan sebuah router maupun firewall untuk menciptakan sebuah gateway keamanan. Beberapa contoh umum adalah sebagai berikut:

a. Integrasi IP sec ke dalam implementasi IP native. Ini membutuhkan akses ke IP source code dan dapat diterapkan baik terhadap host maupun gateway keamanan.

b. Implementasi “Bump-in-the-stack” (BITS), dimana IPsec diimplementasikan di dalam sebuah implementasi tumpukan protokol IP yang ada, di antara IP native dan driver jaringan lokal. Akses source code untuk tumpukan IP tidak dibutuhkan dalam konteks ini, membuat pendekatan implementasi ini cocok untuk digunakan dengan sistem legacy. Pendekatan ini, ketika diadopsi biasanya diletakkan di host.

c. Penggunaan processor kriptografi luar adalah fitur desain yang umum pada sistem keamanan network yang digunakan oleh militer dan beberapa dari sistem komersil. Hal ini sering disebut sebagai implementasi “Bump-in-the-wire” (BITW). Implementasi seperti itu dapat didesain untuk melayani baik sebuah host maupun sebuah gateway maupun keduanya. Biasanya perangkat BITW beralamatkan IP. Ketika mendukung sebuah host tunggal, BITW cukup analog dengan implementasi BITS. Tetapi ketika mendukung sebuah firewall ataupun router, BITW harus beroperasi seperti layaknya gateway keamanan.

2.4. Pemilihan Algoritma dalam IPsec

Untuk implementasi IPsec dapat berinteroperasi, maka harus mendukung satu atau lebih algoritma keamanan yang umum. Algoritma keamanan digunakan secara

aktual untuk setiap implementasi ESP atau asosiasi keamanan AH ditentukan oleh mekanisme negosiasi, seperti Internet Key Exchange (IKE) atau sebelum pembentukannya.

2.4.1. Algoritma enkripsi dan autentikasi pada ESP (Encapsulating Security Payload)

Gambar 1. Ilustrasi ESP

Berikut ini adalah daftar algoritma enkripsi dan autentikasi untuk protocol ESP(*Encapsulating Security Payload*) IPsec. Daftar ini berdasar pada RFC4835 yang dikeluarkan pada April 2007.

Requirement	Encryption Algorithm
MUST	NULL
MUST	AES-CBC 128 bit
MUST -	TripleDES-CBC
SHOULD	AES-CTR
SHOULD NOT	DES-CBC

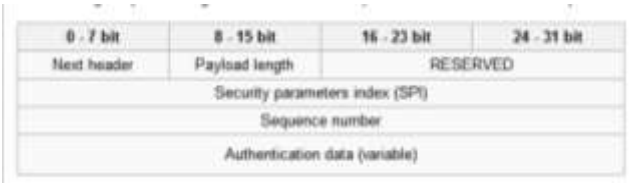
Tabel 1. Algoritma enkripsi ESP

Requirement	Authentication Algorithm
MUST	HMAC-SHA1-96
SHOULD+	AES-XCBC-MAC-96
MAY	NULL
MAY	HMAC-MD5-96

Tabel 2. Algoritma autentikasi ESP

Kelemahan telah terlihat pada SHA-1 dan MD5 tapi hal ini tidak mempengaruhi penggunaan baik SHA-1 maupun MD5 yang menggunakan HMAC

2.4.2. Algoritma enkripsi pada AH (Authentication Header)



Gambar 2. Ilustrasi AH

Berikut ini adalah daftar algoritma yang digunakan pada protokol AH (Authentication Header) IPsec. Daftar ini berdasar pada RFC4835 yang dikeluarkan pada April 2007.

Requirement	Algorithm
MUST	HMAC-SHA1-96
SHOULD+	AES-XCBC-MAC-96
MAY	HMAC-MD5-96

Tabel 3. Algoritma yang diterapkan pada AH

2.4.3. Keterangan "requirement"

MUST menjelaskan bahwa apa yang didefinisikan adalah sebuah kebutuhan mutlak dari spesifikasi yang ada.

SHOULD menjelaskan bahwa terdapat beberapa alasan yang benar dalam beberapa hal untuk mengabaikan beberapa hal di dalamnya, tetapi implikasi secara keseluruhan harus dipahamio dan dipertimbangkan secara matang sebelum memilih pilihan yang lain.

SHOULD NOT menjelaskan bahwa mungkin terdapat beberapa alasan yang benar dalam beberapa kondisi ketika beberapa bagiannya dapat diterima ataupun berguna, tetapi implikasi penggunaan secara keseluruhan harus dipahami dan dipertimbangkan secara betul-betul sebelum mengimplementasikan apa yang diberi label ini.

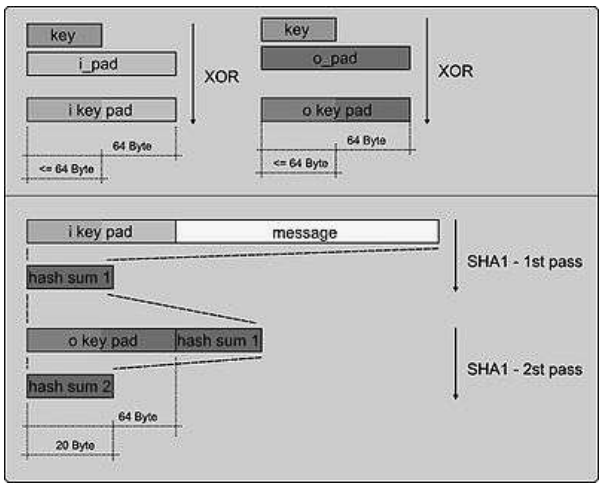
MAY menjelaskan bahwa ini bersifat benar-benar pilihan. Sebuah penyedia dapat memilih untuk memasukkan pilihan ini karena kebutuhan pasar atau karena penyedia berkeyakinan bahwa dengan penambahan ini dapat meningkatkan produk sedangkan penyedia lain menghilangkan pilihan ini.

2.5. Pertimbangan keamanan

Keamanan dari sistem berbasis kriptografi bergantung pada kekuatan dari algoritma kriptografi yang dipilih serta kekuatan dari kunci yang digunakan bersama dengan algoritma yang dipilih tersebut. Kemanan juga bergantung pada administrasi dan perancangan protokol yang digunakan oleh sistem untuk memastikan tidak ada jalan tanpa kriptografi untuk melewati keamanan dari sistem secara keseluruhan.

Dalam pemilihan algoritma kriptografi yang digunakan pada AH dan ESP, algoritma yang digunakan dengan label "MUST" atau "SHOULD" belum ditemukan celah pada saat ini dan riset kriptografi yang ada sampai saat ini membuat percaya bahwa algoritma tersebut akan aman untuk beberapa waktu ke depan.

3. Implementasi HMAC-SHA1-96



Gambar 3. Ilustrasi penggunaan HMAC-SHA-1-96

Implementasi HMAC-SHA1-96 diterapkan pada IPsec pada mekanisme autentikasi berkunci dalam konteks Encapsulating Security Payload dan pada Authentication Header. Tujuan dari penggunaan HMAC-SHA-1-96 adalah meyakinkan bahwa paket asli dan tidak dapat dirubah pada saat singgah.

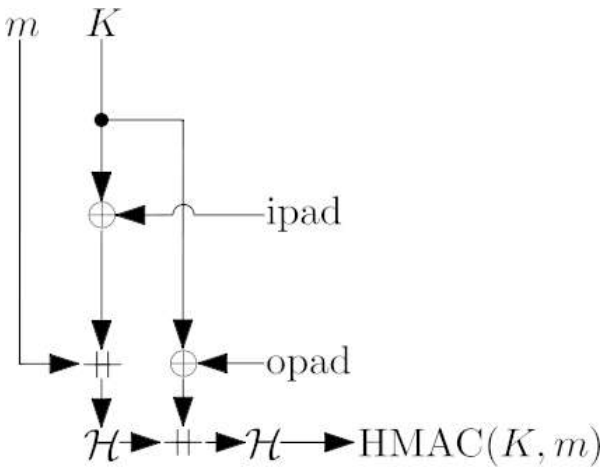
HMAC adalah algoritma autentikasi kunci rahasia. Integritas data dan autentikasi asal data yang disediakan oleh HMAC bergantung pada cakupan distribusi dari kunci rahasia. Jika hanya asal dan tujuan yang mengetahui kunci HMAC, ini menyediakan baik autentikasi asal data dan integritas data paket-paket yang dikirimkan diantara keduanya. Algoritma HMAC juga menyediakan framework untuk memasukkan berbagai macam algoritma hash seperti SHA-1.

HMAC-SHA-1-96 beroperasi pada ukuran data blok 64 byte. Padding yang dibutuhkan dispesifikasikan dalam [FIPS-180-1] dan merupakan bagian dari algoritma SHA-1. Jika SHA-1 dibuat sebgaimana yang dispesifikasikan pada [FIPS-180-1] tidak perlu dilakukan penambahan padding tambahan sebagaimana yang diperhatikan pada HMAC-SHA-1-96. Dengan mengacu pada "padding paket secara implisit" yang didefinisikan pada AH tidak dibutuhkan padding paket secara implicit.

HMAC-SHA-1-96 menghasilkan nilai pengautentikasi 160-bit. Nilai 160 bit ini dapat dipotong sebagai mana dijelaskan pada [RFC-2104]. Untuk digunakan pada ESP maupun AH, sebuah nilai terpotong menggunakan 96 bit awal harus didukung. Ketika dilakukan pengiriman, nilai yang terpotong tersebut diletakkan di dalam bagian autentikator. Ketika diterima, keseluruhan 160 bit dihitung dan kemudian 96 bit pertama dibandingkan dengan nilai yang diletakkan pada bagian autentikator. Tidak ada panjang nilai autentikator yang didukung oleh HMAC-SHA-1-96.

Panjang nilai 96 bit dipilih karena merupakan panjang kunci default pada autentikator yang dispesifikasikan pada AH dan memenuhi persyaratan keamanan yang dideskripsikan pada [RFC-2104].

3.1. Definisi algoritma HMAC



Gambar 4. Ilustrasi algoritma HMAC

HMAC didefinisikan membutuhkan sebuah fungsi kriptografi hash yang dilambangkan dengan H dan sebuah kunci rahasia K. Diasumsikan H sebagai fungsi kriptografi hash dimana data di-hash denan mengiterasikan fungsi kompresi dasar pada blok data. Kita tentukan B sebagai panjang byte dari sebuah blok (pada kasus ini panjang B=64) dan L adalah panjang dari output fungsi hash (untuk MD5 L=16 dan untuk SHA-1 L=20). Panjang kunci K dapat kurang dari sama dengan panjang B yang merupakan panjang blok dari fungsi hash. Aplikasi yang menggunakan kunci lebih panjang dari panjang B bytes kemudian akan melakukan hash dengan menggunakan fungsi H terhadap kunci dan kemudian menggunakan hasil L byte string sebagai kunci actual dari HMAC. Dalam setiap kasus, panjang kunci minimal yang direkomendasikan untuk K adalah L byte yang sesuai dengan panjang dari hasil keluaran fungsi hash.

Didefinisikan dua buah string tetap dan berbeda ipad dan opad sebagai berikut ('i' merupakan singkatan dari "inner" dan 'o' merupakan singkatan dari "outer").

ipad = byte 0x36 diulang B kali

opad = byte 0x5C diulang B kali

untuk menghitung HMAC dari data text tersebut dilakukan dengan cara berikut:

H(K XOR opad, H(K XOR ipad, text))

Dengan penjelasan sebagai berikut:

1. Tambahkan angka nol hingga akhir dari K untuk menciptakan string dengan panjang B byte. Sebagai contoh, jika K panjangnya adalah 20byte dan B=64, maka K akan ditambahkan dengan 44 nol byte 0x00
2. XOR-kan string dengan panjang B byte yang dikomputasikan pada langkah pertama dengan ipad
3. Tambahkan teks aliran data ke B byte string yang dihasilkan dari langkah kedua
4. Lakukan fungsi H apda aliran yang dihasilkan pada langkah ketiga

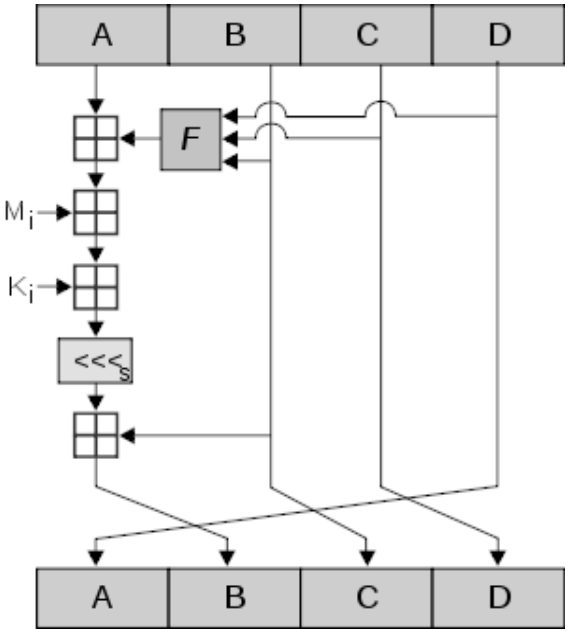
5. XOR kan B byte string yang dikomputasikan pada langkah pertama dengan opad
6. Tambahkan hasil dari fungsi H pada langkah keempat ke B byte string yang dihasilkan pada langkah kelima
7. Lakukan fungsi H kepada aliran data yang dihasilkan pada langkah keenam dan keluarkan hasilnya.

3.2. implementasi SHA-1 dengan HMAC

Sebenarnya SHA-1 telah ditemukan kelemahan-kelemahannya. Namun, hal ini tidak mempengaruhi penggunaan HMAC-SHA-1-96 sebagai sebuah metode kriptografi yang dipilih pada IPsec ini.

4. implementasi HMAC-MD5-96

HMAC-MD5-96 beroperasi pada ukuran data blok 64 byte. Padding yang dibutuhkan dispesifikasikan dalam [RFC-1321] dan merupakan bagian dari algoritma MD5. Jika MD5 dibuat sebgaimana yang dispesifikasikan pada [RFC-1321] tidak perlu dilakukan penambahan padding tambahan sebagaimana yang diperhatikan pada HMAC-MD5-96. Dengan mengacu pada "padding paket secara implisit" yang didefinisikan pada AH tidak dibutuhkan padding paket secara implisit.



Gambar 5. Ilustrasi MD5

HMAC-MD5-96 menghasilkan nilai autentikator yang panjangnya 128bit. Nilai 128 bit ini dapat dipotong seperti yang dideskripsikan pada [RFC-2104]. Untuk dapat digunakan baik pada ESP maupun AH, sebuah nilai terpotong menggunakan nilai 96 bit pertama harus didukung. Ketika melakukan pengiriman, nilai yang terpotong ini diletakkan pada bagian autentikator. Ketika diterima, keseluruhan nilai 128 bit di komputasikan dan 96 bit pertama dibandingkan dengan nilai yang ada pada bagian autentikator. Seperti pada HMAC-SHA-1-96, tidak ada dukungan lain terhadap nilai yang ada pada autentikator.

Keamanan yang diberikan oleh HMAC-MD5-96 berdasar pada kekuatan HMAC, dan sedikit dibawah kekuatan MD5. Pada [RFC-2104] diklaim bahwa HMAC tidak bergantung pada property dari ketahanan kolisi kuat, yang merupakan ketika melakukan evaluasi menggunakan MD5, sebuah algoritma yang telah menunjukkan kurang lebih ketahanan kolisi dari apa yang diduga.

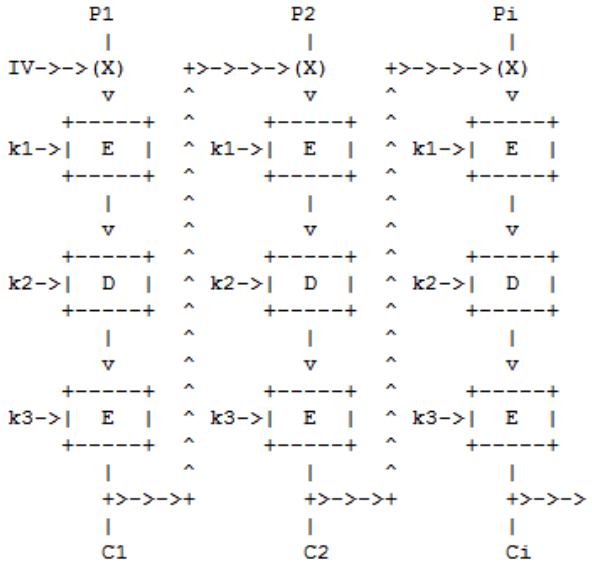
[RFC-2104] menyatakan bahwa "minimally reasonable hash function" dari "birthday attack", serangan terkuat yang diketahui terhadap HMAC tidak berguna. Untuk sebuah hash blok 64 byte seperti HMAC-MD5-96, sebuah serangan yang melibatkan keberhasilan proses 2**64 blok tidak mungkin kecuali jika hal itu ditemukan hash yang mendasari memiliki kolisi setelah memproses 2**30 blok. Sebuah hash yang memiliki karakteristik ketahanan kolisi yang lemah akan dipertimbangkan tidak dapat digunakan.

Hal lain yang harus dipertimbangkan adalah MD5 tidak pernah dikembangkan sebagai algoritma hash berkunci, sedangkan HMAC memiliki criteria tersebut dari awal.

Pada [RFC-2104] juga didiskusikan pada kemanan tambahan yang potensial yang disediakan oleh pemotongan dari hasil hash. Spesifikasi yang melibatkan HMAC sangat dianjurkan untuk melakukan pemotongan hash ini.

Jika melihat kondisi sampai saat tulisan ini dibuat, MD5 telah ditemukan kelemahan-kelemahannya. Tetapi penggunaan algoritma kombinasi dari HMAC dan MD5 sampai saat tulisan ini ditulis masih belum ditemukan kelemahan yang ada padanya.

5. implementasi TripleDES-CBC



Gambar 6. Ilustrasi TripleDES

Varian dari DES ini disebut dengan nama "TripleDES" atau sebagai DES-EDE3. Varian ini memproses setiap blok masing-masing sebanyak tiga kali, yang masing-masing menggunakan kunci yang berbeda.

DES-EDE3-CBC adalah sebuah algoritma yang merupakan varian simple dari DES-CBC.

Pada DES-EDE3-CBC sebuah *initialization vector* (IV) di XOR kan terhadap 64 bit pertama dari blok plainteks(p1). Fungsi DES berkunci diulangi sebanyak tiga kali, sebuah enkripsi (Ek1) kemudian dekripsi (Dk2) kemudian diikuti dengan enkripsi lagi (Ek3), dan kemudian dihasilkan (C1) untuk blok tersebut. Setiap iterasi menggunakan 3 buah kunci yang independen yaitu : k1, k2, dan k3.

Untuk blok-blok selanjutnya, blok cipherteks hasil dari proses sebelumnya di-XOR-kan dengan plainteks yang actual (Pi). Kemudian fungsi enkripsi DES-EDE3 menghasilkan cipherteks (Ci) dari blok tersebut.

Sedangkan untuk melakukan dekripsi urutan dari fungsi dibalik. Dekripsi menggunakan k3, enkripsi menggunakan k2, kemudian dekripsi dengan menggunakan k1 dan kemudian hasilnya di XOR kan dengan blok cipherteks sebelumnya.

Ketika ketiga kunci yang digunakan adalah sama, DES-EDE3-CBC tidak ubahnya sama dengan DES-CBC.

Pada algoritma DES terdapat 64 kunci lemah, termasuk yang disebut kunci semi lemah dan kuci yang mungkin lemah. Pemilihan satu secara acak tak berarti.

Tetapi kejadian dimana dua buah kunci 64 bit sama (k1==k2 atau k2==k3 atau k1==k2==k3), maka implementasi TripleDES ini dapat disamakan dengan DES. Dalam implementasinya, harus ditolak penggunaan kunci yang seperti ini.

6. Kesimpulan

Sampai saat tulisan ini dibuat, sejauh pengamatan dari penulis ketiga algoritma ini masih dapat diterapkan pada protokol IPsec.

Namun dalam waktu yang akan datang algoritma-algoritma ini akan kembali pengimplementasiannya pada protokol IPsec ini. Dan kategori dari masing-masing algoritma akan dapat berubah maupun algoritma tersebut tidak diizinkan digunakan kembali dalam implementasi IPsec.

Keamanan dari HMAC-SHA-1-96 dan HMAC-MD5-96 terdapat pada algoritma HMAC karena baik SHA-1 dan MD5 diketahui memiliki masalah-masalah yang menyebabkan keamanannya dipertanyakan.

7. Daftar Pustaka

[RFC-4835], http://tools.ietf.org/html/rfc4835 waktu akses 30 Maret 2009, 19.00

[RFC-2451], http://tools.ietf.org/html/rfc2451 waktu akses 30 Maret 2009, 19.00

[RFC-2403], http://tools.ietf.org/html/rfc2403 waktu akses 30 Maret 2009, 19.00

[RFC-2404],http://tools.ietf.org/html/rfc2404 waktu akses 30 Maret 2009, 19.00

[RFC-2104],http://tools.ietf.org/html/rfc2104 waktu akses 30 Maret 2009, 19.00

[RFC-2119],http://tools.ietf.org/html/rfc2119 waktu akses 30 Maret 2009, 19.00

http://www.networksorcery.com/enp/topic/ipsecsuite.htm waktu akses 30 Maret 2009, 19.00

[RFC-4301], http://www.networksorcery.com/enp/rfc/rfc4301.txt waktu akses 30 Maret 2009, 19.00

[RFC-2401], http://www.faqs.org/rfcs/rfc2401.html waktu akses 30 Maret 2009, 19.00

[FIPS-180-1], http://www.cryptography.com/resources/whitepapers/misc/FIPS180-1.txt waktu akses 30 Maret 2009, 19.00

http://en.wikipedia.org/wiki/HMAC waktu akses 2 April 2009 22.00

http://en.wikipedia.org/wiki/SHA_hash_functions waktu akses 2 April 2009 22.00

http://en.wikipedia.org/wiki/MD5 waktu akses 2 April 2009 22.00

http://en.wikipedia.org/wiki/Triple_DES waktu akses 2 April 2009 22.00

http://en.wikipedia.org/wiki/Data_Encryption_Standard waktu akses 2 April 2009 22.00