Analisis Perbandingan Algoritma RSA dan Diffie-Hellman untuk Pertukaran Kunci

Reynald Alexander G - 13509006
Program Studi Teknik Informatika
Sekolah Teknik Elektro dan Informatika
Institut Teknologi Bandung, Jl. Ganesha 10 Bandung 40132, Indonesia
13509006@std.stei.itb.ac.id

***Abstract*—Diffie Helman dan RSA merupakan algoritma asimetri yang sudah sering dipergunakan secara umum. Terutama dalam hal pertukaran kunci, algoritma-algoritma seperti Diffie-Hellman tersebut digunakan. Algoritma-algoritma tersebut menggunakan bilangan-bilangan yang disebut kunci privat untuk membangkitkan kunci yang ada. Berikut di makalah ini akan dibahas mengenai kekuatan, kerumitan, serta celah yang mungkin saja ada untuk masing-masing algoritma.**

***Index Terms*—enkripsi, kriptografi, kriptanalis, playfair**

I. PENDAHULUAN

Informasi merupakan hal yang sangat krusial di kehidupan manusia terutama di era modern ini. Informasi memiliki peran dan kekuatan yang besar bagi masing-masing hidup anggota masyarakat. Tak heran jika seringkali terjadi pencurian pesan sandi lewat ataupun sejenisnya. Oleh karena itu dibutuhkanlah metode yang aman untuk dapat mengirim maupun menerima pesan. Oleh karena itulah digunakanlah salah satu metode yang dinamakan kriptografi. Yang mana kriptografi merupakan metode menyembunyikan isi pesan dengan kunci tertentu agar yang bisa mengetahui isi pesan hanyalah yang berwenang saja. Sebagai tambahan informasi, kriptografi ini memiliki beberapa terminologi dasar yakni pengirim pesan, penerima pesan, plainteks, cipherteks, enkripsi, dekripsi, dan kunci. Sedangkan, di era modern ini, pengamanan pesan saja tidaklah cukup karena sudah sering juga terjadi pencurian kunci dekripsi.

Oleh karena itulah digunakanlah salah satu metode yang masih di dalam naungan kriptografi yang dinamakan dengan metode pertukaran kunci. Di mana di era modern ini kelemahan – kelemahan dalam kriptografi sangatlah tidak bisa ditolerir karena hal tersebut bisa saja menyebabkan kerugian yang sangat besar bagi pihak yang bersangkutan. Maka dari itu analisis dan perbandingan algoritma pertukaran kunci yang ada diperlukan untuk menentukan mana algoritma yang lebih baik dan yang mana bisa dikembangkan lebih lanjut lagi untuk dapat dipergunakan kembali.

II. DASAR TEORI

Diffie-Hellman dan RSA termasuk dalam algoritma kriptografi asimetri. Arti dari algoritma asimetri adalah kunci enkripsi dan kunci dekripsi adalah hal yang berbeda. Berbeda halnya dengan algoritma kriptografi simetri yang menggunakan kunci yang sama untuk mengenkripsi pesan ataupun mendekripsi pesan. Contohnya saja adalah chiper substitusi tunggal seperti di bawah ini :

A	B	C	D	E	F	G	H	I	J	K	L	M
R	Z	B	U	Q	K	F	C	P	Y	E	V	L
N	O	P	Q	R	S	T	U	V	W	X	Y	Z
S	N	G	W	O	X	D	J	I	A	H	T	M

Tabel 1. Tabel substitusi abjad

Kita ambil contoh CAESAR sebagai plainteks. Maka hasil enkripsi dari kata CAESAR menggunakan tabel di atas adalah BRQXRO, demikian juga apabila kita chiperteks BRQXRO hendak didekripsi maka menggunakan pembacaan terbalik dari tabel di atas maka hasilnya adalah CAESAR. Adapun sesungguhnya algoritma simetri ini terbagi menjadi dua buah kelas berdasarkan waktunya. Kelas-kelas tersebut adalah algoritma kriptografi klasik dan algoritma kriptografi modern.

Menurut Ir Rinaldi Munir contoh-contoh algoritma yang berada di golongan klasik adalah chiper substitusi tunggal, Viginere Chiper, Playfair Chiper, dan Enigma Chiper. Chiper-chiper golongan tersebut sudah obsolet atau dengan kata lain tidak dapat dipercaya sebagai algoritma yang amana. Sedangkan algoritma yang termasuk ke dalam golongan modern misalnya adalah stream chiper, blok chiper(keduanya bermain di dalam lingkup bit ataupun blok dari sekumpulan bit), serta algoritma asimetri juga termasuk dalam algoritma kriptografi modern.

Contoh dari blok chiper salah satunya adalah DES (Data Encryption Standard) yang menggunakan blok berukuran 64 bit dan putaran sebanyak 16. Contoh lainnya adalah AES (Advanced Encryption Standard) yang merupakan hasil pengembangan dari Data Encryption Standard, DES sudah tidak dipercaya lagi karena telah berhasil dipecahkan menggunakan

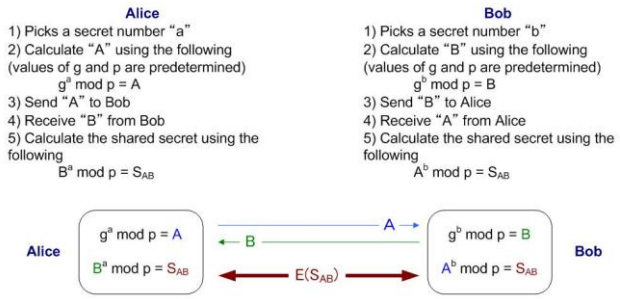
exhaustive search oleh proyek *Electronic Frontier Foundation(EFE)* pada tahun 1998 sebagai akibat dari pendeknya kunci dan jumlah putaran. Oleh karena itulah AES menggunakan jumlah panjang kunci hingga 256 bit dan ukuran blok hingga 128 bit dan blok-S menggunakan metoda Rijndael yang menerapkan metoda substitusi dan permutasi tertentu yang tidak akan dibahas di makalah ini.

Menurut buku Ir Rinaldi Munir, algoritma-algoritma kriptografi asimetri atau lebih terkenal dengan algoritma kunci privat-kunci publik ditinjau dari penerapannya terdiri dari tiga macam yakni enkripsi dan dekripsi, *Digital Signature*, dan pertukaran kunci. Adapun konsep keamanan utama dari tipe algoritma ini adalah operasi aritmetik dari bilangan yang bernilai besar dengan operasi yang digunakan biasanya adalah perpangkatan atau semacamnya. Contoh-contoh dari algoritma asimetri ini adalah RSA, *ElGamal*, Diffie-Hellman, *KnapSack*, Rabin, GOST, DSA. Sedangkan untuk topik kali ini akan lebih spesifik dijelaskan pada Diffie-Hellman dan RSA yang merupakan contoh algoritma kriptografi untuk pertukaran kunci.

Diffie-Hellman merupakan protokol pertukaran kunci untuk yang dikembangkan oleh Whitfield Diffie and Martin Hellman pada tahun 1976. Seperti yang dijelaskan di www.rsa.com bahwa protokol membiarkan dua pengguna untuk saling bertukar kunci melalui media perantara yang tidak aman tanpa ada rahasia sebelumnya.

Metoda yang digunakan untuk pertukaran kunci di sini adalah dengan menggunakan mesin generator kunci publik. Misalnya saja ada dua orang peserta yang hendak saling bertukar pesan, umpamakan saja Bob dan Alice. Ketika Alice akan bertukar kunci, Alice menggunakan kunci privat a kemudian mesin penggenerasi kunci menghasilkan kunci publik dengan menggunakan kalkulasi $n = g^{ab} \bmod p$ dengan p merupakan bilangan bulat besar prima dan g merupakan bilangan bulat yang nilainya lebih kecil dari p dengan syarat untuk setiap n yang ada dari 1 hingga p-1 inklusif, terdapat hasil pangkat oleh k dari g yang mana $n = g^k \bmod p$. Demikian pula Bob juga menggunakan kunci privat b dan mesin penggenerasi kunci menghasilkan $n = g^{ba} \bmod p$. Dari kedua persamaan tersebut dapat disimpulkan bahwa secara tidak langsung Bob dan Alice telah saling bertukar kunci privat. Untuk lebih jelasnya adalah adanya $g^{ba} \bmod p = g^{ab} \bmod p = k$.

Apabila terdapat lebih dari dua partisipan yang hendak saling bertukar kunci, partisipan-partisipan tersebut saling bertukar g^x dengan x adalah kunci privat yang dimiliki masing-masing partisipan. Apabila partisipan berjumlah 3 yakni A,B dan C. Maka A mengirimkan $g^a \bmod p$ kepada b dan b menghitung nilai $g^{ab} \bmod p$ menggunakan kunci privatnya yakni b, kemudian menyerahkannya kepada C sehingga C dapat menghitung nilai $g^{abc} \bmod p$ menggunakan kunci privatnya yakni c. Pada akhirnya C menggunakan $g^{abc} \bmod p$ sebagai kunci privat terbaginya. Demikian pula teknik yang sama dapat dilakukan untuk B, yakni C mengirimkan $g^c \bmod p$ kepada A, setelah itu A menghitung $g^{ca} \bmod p$ menggunakan kunci privat a lalu memberikannya kepada B sehingga B bisa menghitung $g^{cab} \bmod p = g^{abc} \bmod p$ sebagai kunci privat terbaginya. Berikut adalah gambar yang diambil dari voipsa.org :



Gambar 1. Skema pertukaran kunci Diffie Hellman

Menurut Fred Hazan and Frank Rundatz di searchsecurity.techtarget.com, algoritma RSA merupakan algoritma kunci privat kunci publik yang dikembangkan oleh Ron Rivest, Adi Shamir, dan Leonard Adleman pada tahun 1977. Orang-orang tersebut adalah para ilmuwan yang berada di MIT (*Massachuset Institute of Technology*). RSA telah banyak digunakan di berbagai hal misalnya saja di web browser yang dikembangkan oleh microsoft dan netscape.

Konsep utama keamanan dari RSA adalah susah pemfaktoran bilangan-bilangan besar menjadi faktor-faktor primanya. Terdapat besaran-besaran yang penting di algoritma RSA yakni :

1. p dan q bilangan prima	(rahasia)
2. n = p . q	(tidak rahasia)
3. φ (n) = (p - 1) (q - 1)	(rahasia)
4. e (kunci enkripsi)	(tidak rahasia)
5. d (kunci dekripsi)	(rahasia)
6. m (plainteks)	(rahasia)
7. chiperteks	(tidak rahasia)

Tabel 2. Besaran – besaran di dalam RSA

Teknik operasi pembangkitan kunci pada RSA adalah sebagai berikut

1. Memilih dua bilangan prima berbeda p dan q.
   - Untuk alasan keamanan, bilangan bulat p dan q dipilih secara random.
2. Compute $n = pq$. Hitung n = p . q
   - *n* digunakan sebagai modulus dari kunci publik dan kunci privat.
3. Hitung $\varphi(n) = (p-1)(q-1)$, di mana φ is fungsi Euler tosien.
4. Pilih sebuah bilangan bulat *e* sehingga $1 < e < \varphi(n)$ dan faktor pembagi terbesar dari $(e, \varphi(n)) = 1$; i.e., *e* dan $\varphi(n)$ are relatif prima.

- *e* digunakan sebagai eksponen kunci publik.
- *e* mempunyai panjang bit yang pendek dan berat Hamming yang ringan menghasilkan hasil yang lebih efisien dalam enkripsi - umumnya 0x10001 = 65,537. Namun demikian, semakin kecil nilai *e* (such as 3) semakin kecil pula tingkat keamanan di hal – hal tertentu.

5. Tentukan *d* sebagai:

$$d \equiv e^{-1} \pmod{\varphi(n)}$$

i.e., *d* is the invers perkalian of *e* mod φ(*n*).

- *d* disimpan sebagai eksponen kunci privat.

Sedangkan teknik enkripsi dan dekripsinya adalah :

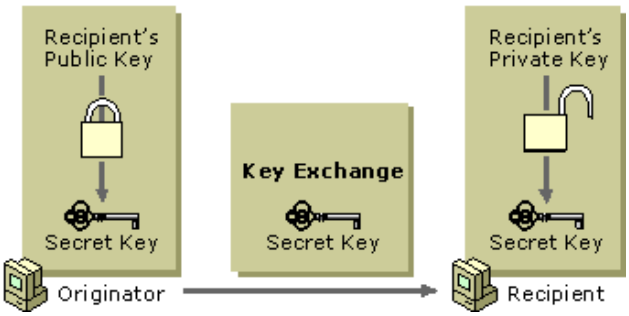
Enkripsi :

1. Ambil kunci publik penerima pesan, e, dan modulus n.
2. Ubah plainteks m menjadi blok-blok sehingga merepresentasikan selang [0,n – 1].
3. Setiap blok m_i dienkripsi menjadi blok c_i dengan rumus $c_i = m_i^e \bmod n$

Dekripsi :

Setiap blok c didekripsi menjadi blok m dengan rumus $m_i = c_i^d \bmod n$

Berdasar technet.microsoft.com, penerapan RSA di dalam pertukaran kunci adalah dengan cara mengenkripsi kunci privat dari pesan dengan menggunakan kunci publik hasil pembangkitan dari RSA dan pesan berisi kunci itu dapat dibuka hanya dengan kunci privat hasil pembangkitan RSA yang dimiliki oleh penerima pesan. Berikut kurang lebih skema dari pertukaran kunci tersebut:



Gambar 2. Skema pertukaran kunci dengan RSA

III. PERBANDINGAN DAN ANALISIS KOMPLEKSITAS ALGORITMA

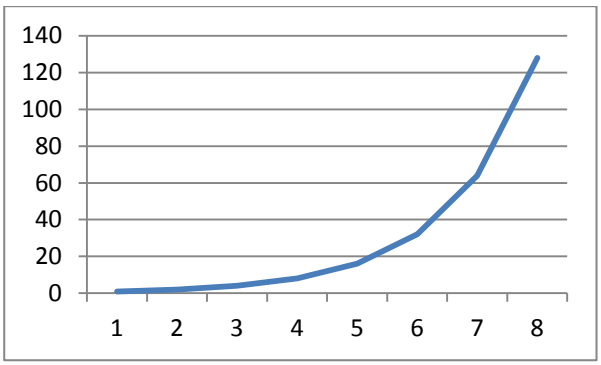
Prinsip kerja utama dari algoritma Diffie-Hellman adalah dengan menggunakan permasalahan logaritma diskrit. Permasalahan logaritma diskrit dapat dinotasikan sebagai $g^a \equiv A \pmod p$. Permasalahan utama dari hal ini adalah penentuan nilai a. Karena pada prinsipnya, untuk beberapa nilai a, hasil modulus yang dihasilkan adalah A, untuk lebih jelasnya kita perhatikan contohnya. Untuk p bernilai 17 dan untuk g bernilai 3 dan a bernilai 4, kita dapat mencari nilai A yakni $3^4 \bmod 17 = 13$. Sehingga persamaan di atas menjadi $3^4 \equiv 13 \pmod{17}$. Pencarian nilai 13 tadi disebut sebagai eksponensial diskrit. Sekarang kita ubah, yakni yang tidak diketahui adalah nilai a, sedangkan untuk $3^{16} \equiv 1 \pmod{17}$, maka $3^{4+16n} \equiv 13 \pmod{17}$ untuk n = 0, 1, 2, ... dengan kata lain nilai a tidak hanya memiliki 1 nilai saja, tetapi memiliki banyak kemungkinan nilai.

Di dalam Diffie-Hellman, nilai a yang digunakan biasanya memiliki panjang yang luar biasa, demikian juga dengan nilai p yang digunakan. Sedangkan untuk nilai g, cukup digunakan bilangan prima dengan panjang 1 digit saja. Karena dengan pemangkatan oleh a yang panjangnya luar biasa, nilai yang dihasilkan pun akan luar biasa besar. Dengan asumsi bahwa g merupakan sebuah bilangan berdigit 1 dan pencarian nilai a adalah dengan teknik exhaustive search, berdasarkan perhitungan secara awam, ditentukanlah bahwa kompleksitas pencarian nilai a sehingga $g^a = S$ adalah secara linier dalam iterasi nilai a sedangkan bernilai eksponensial dalam operasi g^a yakni dengan semakin besarnya nilai a maka operasi perkalian akan semakin rumit(menaik secara eksponensial). Contoh sederhananya adalah misalkan kemungkinan nilai a adalah 8 dan nilai g adalah 2, dengan menggunakan prinsip di atas maka iterasi linier terjadi sebanyak 8 kali sedangkan di 8 proses tersebut dapat diuraikan sebagai berikut.

(i) a = 1	2
(ii) a = 2	2 x 2
(iii) a = 3	2 x 4
(iv) a = 4	2 x 8
(v) a = 5	2 x 16
(vi) a = 6	2 x 32
(vii) a = 7	2 x 64
(viii) a = 8	2 x 128

Tabel 3. Kompleksitas kalkulasi Diffie Hellman

Apabila angka 2, 4, 8, 16, 32, 64, 128 tersebut digambarkan dalam bentuk kurva maka bentuknya akan menaik secara eksponensial seperti di bawah ini.

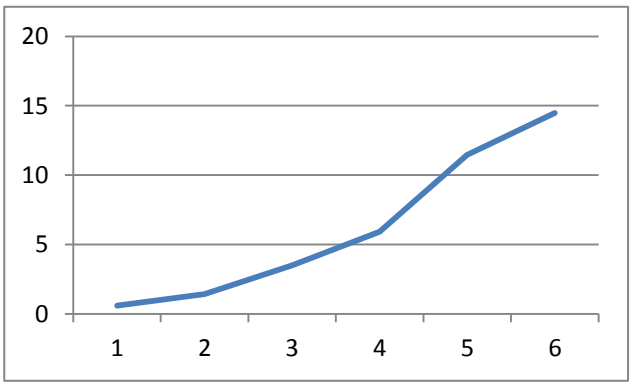


Gambar 3. Grafik eksponensial 2^n

Sehingga dapat disimpulkan bahwa kompleksitas dari persoalan Diffie Hellman bukanlah eksponensial semata melainkan merupakan gabungan dari linier dan eksponensial.

Berbeda dengan prinsip kerja Diffie-Hellman yang menggunakan prinsip DLP(*Discrete Logarithm* Problem), prinsip kekuatan RSA diletakkan pada sulitnya mencari faktor prima dari sebuah bilangan yang sangat besar. Dalam kasus ini, faktor prima yang dimaksud adalah p dan q sebagai nilai awal pembangkit kunci. Di mana p tidak boleh sama dengan q. Apabila diketahui sebuah nilai n, dan kompleksitas *primality test*(ujicoba apakah bilangan yang dipilih prima atau bukan) yang paling mangkus memiliki kompleksitas sebesar O(n) = log(n) maka dapatlah dihitung bahwa kompleksitas dari algoritma RSA ini adalah O(n) = n . log(n) dengan n adalah hasil kali p dan q yang merupakan batas iterasi apabila pencarian faktor prima menggunakan *exhaustive search* dan log(n) adalah kompleksitas algoritma untuk menguji apakah bilangan yang dipilih merupakan bilangan prima atau bukan.

Sebagai gambaran kita ambil sebuah nilai n adalah 14, bilangan uji dimulai dari angka 1,dst. Sehingga membentuk kurva seperti di bawah ini:



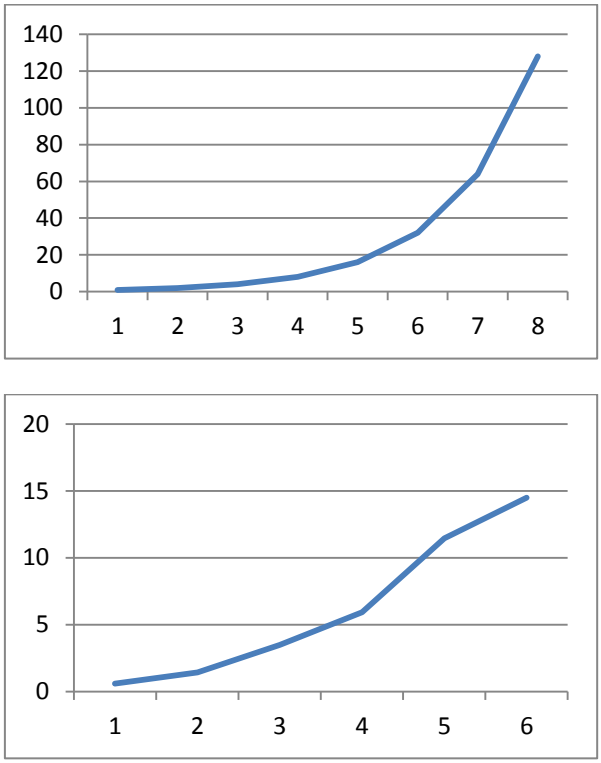
Gambar 4. Grafik n . log(n)

Dari grafik dapat kita lihat bahwa grafik tersebut memiliki kecenderungan menaik seperti layaknya graf eksponensial. Oleh karena itu apabila hendak dipecahkan secara *bruteforce* terutama untuk nilai p dan q yang memiliki panjang yang sangat luar biasa dapat memakan waktu yang sangat lama hingga bertahun-tahun atau bahkan lebih dari 1 abad meskipun telah mengerahkan seluruh sumber daya komputasi yang ada karena selama ini belum ada algoritma yang mangkus untuk mencari faktor primadari sebuah bilangan kecuali dengan *exhaustive search* dan *primality testing*.

Dari hasil di atas dapat disimpulkan bahwa problem RSA bukanlah problem polinomial dan memiliki kompleksitas eksponensial. Problem eksponensial akan semakin rumit dengan semakin besarnya nilai n yang tentunya di dalam RSA nilai n dipengaruhi oleh nilai p dan nilai q.

IV. Analisis Perbandingan Kekuatan Algoritma

Dengan menggunakan ulasan mengenai kompleksitas di atas, apabila kita hanya memperhatikan nilai O(n) saja yang mana untuk Diffie-Hellman dapat ditarik kesimpulan bahwa O(n) = n . 2^n ,sedangkan untuk RSA nilai O(n) = n . log(n) maka Diffie-Hellman lah yang memiliki keunggulan di atas RSA karena secara grafik Diffie-Hellman lebih menaik secara eksponensial karena pada dasarnya grafik y = 2^x merupakan grafik yang sudah menaik secara eksponensial dan dipertegas lagi dengan kehadiran n.



Gambar 5. Perbandingan grafik Diffie-Hellman dan RSA

Pada dasarnya dua algoritma tersebut akan mungkin dapat ditembus secara operasi matematika apabila terjadi keluaran bilangan yang bernilai kecil(panjang bilangan kurang). Misalnya saja pada Diffie-Hellman, bisa saja nilai kunci privat ataupun nilai q yang dipilih kurang besar. Pada algoritma RSA pun hal yang sama seperti

kurang besarnya nilai p ataupun q serta pemilihan nilai p dan q yang tidak sesuai, yakni ketika p = q. Hal-hal tersebut biasanya terjadi karena adanya gangguan dari pihak luar, seperti yang akan dijelaskan berikutnya.

Namun pada dasarnya, permasalahan keamanan dari algoritma-algoritma kriptografi pertukaran kunci tersebut tidak hanya terletak pada kompleksitas perhitungan saja. Terdapat berbagai macam serangan yang bisa melumpuhkan keamanan dari algoritma Diffie-Hellman ataupun RSA.

Seperti yang dikatakan oleh Jean-Francois Raymond, serangan terhadap protokol Diffie-Hellman dapat dikategorikan menjadi tiga, yakni :

1. Denial of service Attacks
   Penyerang berupaya agar pengguna gagal dalam menggunakan protokol yang ada dengan memberikan komputasi ataupun komunikasi yang tak berguna.
2. Outsider Attacks
   Penyerang berupaya untuk memotong komunikasi protokol, misalnya saja menambahkan, menghapus, atau mengganti pesan dari protokol.
3. Insider Attacks
   Bisa saja terjadi bahwa salah satu peserta di protokol DH membuat protokol yang bisa dengan mudah diserang untuk memperoleh kunci privat dari peer. Serangan dilakukan bisa saja dengan menggunakan perangkat lunak berbahaya seperti virus atau semacamnya.

Salah satu serangan yang sering dibahas oleh banyak situs di internet adalah Man in the Middle Attacks, penjelasan dari teknik ini adalah bahwa pihak ketiga umpamakan C, melakukan komunikasi dengan peserta A seolah-olah sebagai B dan dia memberikan kunci privat $g^{y'}$ kepada A sedangkan A mengira bahwa kunci yang benar adalah $g^{y'}$ dan C melakukan komunikasi dengan B dan memberikan kunci $g^{x'}$ seolah-olah dia sebagai A, sedangkan B mengira bahwa C adalah A dan mengira $g^{x'}$ merupakan kunci yang benar. Demikian selanjutnya C berkomunikasi dengan A dan B shared private key C menerima pesan dari salah satu partisipan kemudian mampu mendekripsi, mengaksesnya, membaca, bahkan mengubah isi pesannya kemudian mengenkripsinya kembali dan memberikannya ke pihak partisipan satunya.

Sedangkan serangan lainnya yang mungkin adalah dengan menggunakan kesalahan protokol, dengan konsep Serangan berdasarkan Teori Bilangan. Misalnya saja nilai $g^x = 1$ yang sesungguhnya sangat kurang mungkin untuk terjadi karena x merupakan nilai yang peserta protokol tentukan. Hal yang mungkin adalah adanya serangan perangkat lunak tertentu yang dapat mengubah nilai g^x sehingga menjadi sama dengan satu. Pada dasarnya kelemahan tersebut hanya terletak pada protokolnya dan pada Diffie-Hellman yang masih standar dan belum dimodifikasi, di zaman sekarang protokol Diffie-Hellman telah dimodifikasi dan keamanannya jauh lebih terjamin.

Sedangkan pertukaran kunci pada RSA tidak dapat dicuri dengan mudah oleh pihak ketiga karena kunci privat untuk mendekripsi pesan berisi kunci tersebut hanya dimiliki oleh penerima pesan saja. Jadi meskipun ada pihak luar C mengambil pesan berisi kunci, C tidak dapat membuka isi dari pesan tersebut. Tetapi selain itu, serangan terhadap RSA dapat dilakukan dengan Wiener's attack, serangan ini dapat dilakukan apabila terdapat weak key yang digunakan untuk pembangkitan kunci. Maka dari itulah pembangkitan nilai p dan q harus memenuhi aturan Fermat.

Selain itu Timing Attacks juga dapat dilakukan untuk menebak kunci ketika penyerang mengetahui waktu dekripsi dari beberapa chiperteks yang diketahui. Selain itu pada tahun 2003 ditemukanlah tipe serangan pada RSA dengan cara memulihkan faktorisasi dari RSA misalnya saja dari SSL dari koneksi web server. Serangan ini menggunakan prinsip modifikasi yang menggunakan *Chinese Remainder Theorem*. Cara yang paling ampuh untuk mengatasi serangan ini adalah daripada menggunakan $c^d \pmod n$, digunakanlah r random dan dihitunglah $(r^e c)^d \pmod n$. Nilai r ini selalu diubah untuk masing-masing chiperteks yang dikirim.

Adapun demi tingkat sekuritas protokol pertukaran kunci yang lebih baik. Hal-hal seperti inilah yang perlu diperhatikan untuk Diffie-Hellman :

1. Nilai kunci privat a dan b haruslah memiliki panjang yang luar biasa.
2. Nilai p haruslah memiliki panjang yang sangat panjang juga.
3. Gunakanlah fungsi hash seperti MD5 ataupun SHA1 sebagai alat untuk mengetahui keaslian pesan.

Sedangkan untuk sekuritas protokol RSA, hal berikut perlu diperhatikan, yakni :

1. Nilai p dan q haruslah memiliki panjang yang cukup.
2. Nilai p dan q tidak boleh berdekatan apalagi p = q
3. Gunakanlah kunci yang berbeda-beda untuk masing-masing chiperteks sehingga tidak mudah diserang dengan timing attack oleh pihak penyerang.

V. CONCLUSION

Dari hasil perbandingan di atas, dapat dilihat bahwa satu dengan lainnya memiliki kelemahan di celah-celah yang berbeda namun permasalahan utama dari kedua protokol pertukaran kunci itu bukan di algoritmanya melainkan terletak pada keamanan protokolnya itu sendiri. Adapun beberapa hal yang perlu diamati bahwa :

1. Algoritma Diffie-Hellman memiliki kompleksitas lebih tinggi sehingga membutuhkan komputasi yang lebih tinggi dan waktu kriptanalis yang juga lebih lama.
2. Kekuatan algoritma Diffie-Hellman ataupun RSA terutama terletak pada panjang atau pendeknya bilangan yang dibangkitkan serta apakah memenuhi persyaratan yang ada sesuai aturan algoritma masing-masing.
3. Apabila tingkat keacakan bilangan-bilangan yang dibangkitkan kurang tinggi, peluang untuk berhasil diserang pun akan semakin besar.

REFERENCES

Security Issue in the Diffie-Hellman Key Agreement Protocol Paper. *Raymond, Jean-Francois & Stiglic, Anton.*
Diktat Kuliah Kriptografi. *Munir, Ir Rinaldi*. 2005
http://www.cimt.plymouth.ac.uk/resources/codes/codes_u1_text.pdf
http://www.rsa.com/rsalabs/node.asp?id=2248
http://www.rsa.com/press_release.aspx?id=261
http://technet.microsoft.com/en-us/library/cc962035.aspx

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, 14 Mei 2012

ttd

Reynald Alexander G / 13509006