Building of Informatics, Technology and Science (BITS)
Volume 2, No 1, June 2020 Page 39-43
ISSN 2684-8910 (media cetak)
ISSN 2685-3310 (media online)

Modifikasi Pembangkit Kunci Algoritma RSA Dengan Menerapkan Algoritma Blum Blum Shub (BBS)

Chandra Frenki Sianturi

Prodi Teknik Informatika, STMIK Budi Darma, Medan, Indonesia
Email: chandrafrenkisianturi@gmail.com

Abstrak

RSA mempunyai dua kunci, yaitu kunci publik dan kunci rahasia. RSA mendasarkan proses enkripsi dan dekripsinya pada konsep bilangan prima dan aritmetika modulo. Baik kunci enkripsi maupun dekripsi keduanya merupakan bilangan bulat. Kunci enkripsi tidak dirahasiakan dan diberikan kepada umum (sehingga disebut dengan kunci publik), namun kunci untuk dekripsi bersifat rahasia (kunci privat). Untuk menemukan kunci dekripsi, dilakukan dengan memfaktorkan suatu bilangan bulat menjadi faktor-faktor primanya. Oleh sebab itu untuk meperkuat keamanan dari algoritma RSA perlu dilakukan peroses pembuatan kunci dilakukan dengan memfaktorkan suatu bilangan bulat menjadi faktor-faktor primanya pada umunya. Dikarenakan haltersebuat sudah umum makanya perlu dilakukan modifikasi kunci pada algoritma RSA. Modifikasi dilakukan dengan menggunkana algoritma lain yang menggunakan bilangan random Dengan memasukkan bilangan random, dianggap dapat menghilangkan kemungkinan penyerang menebak hasil dengan mengetahui algoritmanya.Salah satu algoritma tersebut adalah algoritma Algoritma Blum Blum Shub (BBS). Tujuan dalam menggunakan algoritma Algoritma Blum Blum Shub (BBS) ini adalah agar kunci yang dihasilkan lebih sulit ditebak sehingga mempersulit kriptanalis dalam membaca pesan atau informasi tersebut. Berdasrkan dari hasil proses di atas bahwasannya Algoritma RSA bisa digunakan untuk pembentukan kunci dengan menggunkan algoritma Blum Blum Shub sebagai kunci yang digunkan pada algoritma RSA.

Kata Kunci: Modifikasi, Kunci, Kriptografi, RSA

Abstract

RSA has two keys, namely public key and secret key. RSA bases its encryption and decryption process on the concepts of prime numbers and modulo arithmetic. Both encryption and decryption keys are integers. The encryption key is not kept secret and given to the public (so called the public key), but the key for decryption is confidential (private key). To find the decryption key, it is done by factoring an integer into its prime factors. Therefore to strengthen the security of the RSA algorithm it is necessary to process the key making by factoring an integer into its prime factors in general. Because this is so common, a key modification to the RSA algorithm needs to be done. Modifications are carried out by using another algorithm that uses random numbers. By entering random numbers, it is considered able to eliminate the possibility of an attacker guessing the results by knowing the algorithm. One of these algorithms is the Blum Blum Shub Algorithm (BBS) algorithm. The purpose of using the Blum Blum Shub (BBS) algorithm is to make the key more difficult to guess, making it difficult for cryptanalysts to read the message or information. Based on the results of the above analysis that the RSA algorithm can be used for key formation by using the Blum Blum Shub algorithm as a key used in the RSA algorithm.

Keywords: Modification, Key, Cryptography, RSA.

1. PENDAHULUAN

Untuk berbagai alasan, keamanan dan kerahasiaan sangat dibutuhkan dalam komunikasi data. Terdapat beberapa usaha untuk menangani masalah keamanan data rahasia yang dikirimkan melalui *internet*, diantaranya adalah menggunakan teknik kriptografi dan steganografi. Teknik pengamanan data sudah banyak dikembangkan pada saat ini, hal tersebut tentu semakin memudahkan semua pihak dalam melakukan pengamanan data. Salah satu yang banyak dipergunakan saat ini adalah sistem pengamanan berbasis komputerisasi yang dapat dipergunakan kapan saja dan dimana saja. Pada kriptografi, terdapat proses enkripsi yang mengubah teks polos menjadi *ciphertext* dan proses *dekripsi* yang mengubah *ciphertext* menjadi teks polos kembali algoritma ini juga dapat memanfaatkan kunci yang dimasukkan dari luar. Teknik kriptografi dapat menimbulkan kecurigaan pada pihak ketiga yang tidak berhak menerima informasi karena pesan disamarkan dengan cara mengubah pesan yang asli menjadi seolah-olah tidak terbaca. Selanjutnya pihak ketiga tersebut akan memiliki keinginan untuk mengetahui isi pesan rahasia tersebut dan berusaha memecahkan informasi yang sebenarnya.

RSA mempunyai dua kunci, yaitu kunci publik dan kunci rahasia. RSA mendasarkan proses enkripsi dan dekripsinya pada konsep bilangan prima dan aritmetika modulo. Baik kunci enkripsi maupun dekripsi keduanya merupakan bilangan bulat. Kunci enkripsi tidak dirahasiakan dan diberikan kepada umum (sehingga disebut dengan kunci publik), namun kunci untuk dekripsi bersifat rahasia (kunci privat). Untuk menemukan kunci dekripsi, dilakukan dengan memfaktorkan suatu bilangan bulat menjadi faktor-faktor primanya. Kenyataannya, memfaktorkan bilangan bulat menjadi faktor primanya bukanlah pekerjaan yang mudah. Karena belum ditemukan algoritma yang efisien untuk melakukan pemfaktoran. Cara yang bisaa digunakan dalam pemfaktoran adalah dengan menggunakan pohon faktor. Jika semakin besar bilangan yang akan difaktorkan, maka semakin lama waktu yang dibutuhkan. Jadi semakin besar bilangan yang difaktorkan, semakin sulit pemfaktorannya, semakin kuat pula algoritma RSA[1].

Oleh sebab itu untuk meperkuat keamanan dari algoritma RSA perlu dilakukan peroses pembuatan kunci dilakukan dengan memfaktorkan suatu bilangan bulat menjadi faktor-faktor primanya pada umunya.

Submitted: **23/06/2020**; Revised: **27/06/2020**; Accepted: **29/06/2020**; Published**: 30/06/2020**

Building of Informatics, Technology and Science (BITS)
Volume 2, No 1, June 2020 Page 39-43
ISSN 2684-8910 (media cetak)
ISSN 2685-3310 (media online)

Dikarenakan haltersebuat sudah umum makanya perlu dilakukan modifikasi kunci pada algoritma RSA. Modifikas dilakukan dengan menggunkana algoritma lain yang menggunakan bilangan random Dengan memasukkan bilangan random, dianggap dapat menghilangkan kemungkinan penyerang menebak hasil dengan mengetahui algoritmanya. Telah banyak algoritma generator bilangan acak yang diusulkan dan digunakan hingga saat ini. Algoritma-algoritma tersebut menggunakan berbagai pendekatan berbeda untuk menghasilkan bilangan random seacak mungkin. Hingga saat ini kita tidak dapat membuat suatu generator bilangan acak yang dapat menghasilkan bilangan acak secara murni[2].

Salah satu algoritma tersebut adalah algoritma Algoritma Blum Blum Shub (BBS). Tujuan dalam menggunakan algoritma Algoritma Blum Blum Shub (BBS) ini adalah agar kunci yang dihasilkan lebih sulit ditebak sehingga mempersulit kriptanalis dalam membaca pesan atau informasi tersebut[3]. Algoritma Blum Blum Shub merupakan suatu metode yang berfungsi men-generate bilangan acak secara proses matematis dengan output yang dihasilkan adalah deretan angka biner. Selain itu perlu dilakukan penambahan metode Chaotic Function yang bekerja secara linier XOR pada saat bit-bit bilangan acak dihasilkan. Keunggulan lain dari metode Chaotic Function yang digunakan ini dikarenakan waktu proses bisa bertambah makin cepat.

2. METODE PENELITIAN

2.1 Kriptografi

Kriptografi (cryptography) berasal dari Bahasa Yunani, yaitu cryptos dan graphia yang berarti 'penulisan rahasia'. Kriptografi adalah ilmu ataupun seni yang mempelajari bagaimana membuat suatu pesan yang dikirim oleh pengirim dapat disampaikan kepada penerima dengan aman. Kriptografi merupakan bagian dari suatu cabang ilmu matematika yang disebut kriptologi (cryptology). Kriptografi bertujuan menjaga kerahasian informasi yang terkandung dalam data sehingga informasi tersebut tidak dapat diketahui oleh pihak yang tidak sah. Perancang algoritma kriptografi disebut kriptografer. Kriptanalisis (cryptanalysis) adalah suatu ilmu dan seni membuka (breaking) ciphertext menjadi plaintext tanpa mengetahui kunci yang digunakan. Pelaku kriptanalisis disebut kriptanalis (cryptanalyst). Kriptanalis merupakan lawan kriptografer. Persamaan kriptanalis dan kriptografer adalah bahwa kedua sama-sama menerjemhkan ciphertext menjadi plaintext[4].

2.2 Algoritma RSA

Sebagai algoritma kunci publik, RSA mempunyai dua kunci, yaitu kunci publik dan kunci rahasia. RSA mendasarkan proses enkripsi dan dekripsinya pada konsep bilangan prima dan aritmetika modulo. Baik kunci enkripsi maupun dekripsi keduanya merupakan bilangan bulat. Kunci enkripsi tidak dirahasiakan dan diberikan kepada umum (sehingga disebut dengan kunci publik), namun kunci untuk dekripsi bersifat rahasia (kunci privat)[5]. Besaran-besaran yang digunakan pada algoritma RSA:

1. p dan q bilangan prima (rahasia)
2. r = p . q (tidak rahasia)
3. ϕ (r) = (p – 1)(q – 1) (rahasia)
4. PK (kunci enkripsi) (tidak rahasia)
5. SK (kunci dekripsi)
6. X (*plainteks*) (rahasia)
7. Y (*cipherteks*) (rahasia)

Langkah-langkah proses enkripsi adalah sebagai berikut :

1. *Plaintext* diubah ke dalam bentuk bilangan. Untuk mengubah *plaintext* yang berupa huruf menjadi bilangan dapat digunakan kode ASCII dalam sistem bilangan decimal.
2. *Plaintext* m dinyatakan menjadi blok-blok x1,x2,x3,...., sedemikian sehingga setiap blok merepresentasikan nilai di dalam selang [0, n-1] , sehingga transformasinya menjadi satu ke satu.
3. Setiap blok mi dienkripsi menjadi blok ci dengan rumus $y_i = x_i^{PK} \bmod r$

Langkah-langkah proses enkripsi adalah sebagai berikut :

1. Setiap blok *cipherteks* y_i didekripsi kembali menjadi blok x_i dengan rumus $x_i = y_i^{SK} \bmod r$.
2. Kemudian blok-blok m1,m2,m3,...., diubah kembali ke bentuk huruf dengan melihat kode ASCII hasil dekripsi

2.3 Algoritma Blum-Blum-Shub

Blum-Blum Shub (BBS) merupakan suatu Pseudo Random Number Generator yang diajukan pada tahun 1986 oleh Lenore Blum, Manuel Blum dan Michael Shub[6]. BBS memiliki bentuk persamaan:

$X_{n+1} = x_n^2 \ Mod \ n$ (1)

Diman x adalah fungsi Carmichael $x(M) = x(p.) = lcm(p-1, -1)$

dengan m merupakan hasil dari perkalian dua buah bilangan prima besar p dan q, serta output-nya dalam Least Significant Bit dari Xn dimana hal yang sama sebagai parity dari Xn. Dua buah bilangan prima p dan q harus

Submitted: **23/06/2020**; Revised: **27/06/2020**; Accepted: **29/06/2020**; Published: **30/06/2020**

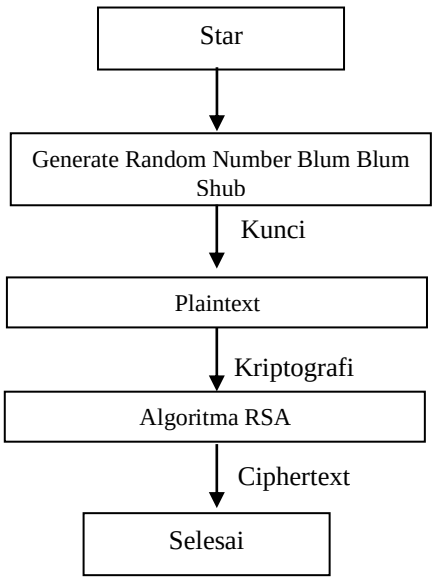
Building of Informatics, Technology and Science (BITS)
Volume 2, No 1, June 2020 Page 39-43
ISSN 2684-8910 (media cetak)
ISSN 2685-3310 (media online)

kongruen terhadap 3 mod 4 dan Greatest Common Divisor (GCD) harus kecil. Generator ini sering digunakan untuk aplikasi kriptografi, karena generator ini tidak begitu cepat. Bagaiman pun juga, generator ini mempunyai bukti keamanan yang kuat, dimana berhubungan dengan kualitas generator karena sulitnya faktorisasi integer. Berikut langkah- langkah algoritma dari BBS:

1. Hitung $X_{n+1} = x_n^2 \ Mod \ n$
2. Hasilkan zi = bit - bit yang diambil dari xi. Bit yang diambil bisa merupakan LSB (Least Significant Bit) atau hanya satu bit atau sebanyak j bit (j tidak melebihi log2 (log2 n)). Bilangan bit acak dapat digunakan langsung atau di-format dengan aturan tertentu, sedemikian hingga menjadi bilangan bulat

3. ANALISA DAN PEMBAHASAN

Analisa terdapat suatu algoritma dapat bertujuan untuk melihat faktor efesiensi dan efektifitas dari algoritma yang sedang dianalisa , dapat dilakukan dengan melihat sisi waktu tempu dari suatu algoritma,proses atatu langkah langkah atau satuan waktu yang ditempuh dari suatu algoritma dalam menyelesaikan suatu masalah.kriptografi merupakan metode dengan menyandikan file teks menjadi yang sulit atau bahkan tidak dipahami melalui proses enkripsi, untuk memperoleh kembali informasi yang dapat dengan proses enkripsi, untuk memperoleh kembali informasih yang asli dan dapat dilakukan dengan proses enkripsi yang tentunya dapat digunakan dengan kunci yang benar, Untuk melindungi file teks darinpihak pihak yang tidak berkepintangan tersebut maka diperlukan enkripsi dan denkripsi agar dapat dilakukan dengan baik, dibutuhan suatu algoritma untuk endkripsi dan dekripsi salah satunya algoritma RSA. RSA mempunyai dua kunci, yaitu kunci publik dan kunci rahasia. RSA mendasarkan proses enkripsi dan dekripsinya pada konsep bilangan prima dan aritmetika modulo. Baik kunci enkripsi maupun dekripsi keduanya merupakan bilangan bulat. Kunci enkripsi tidak dirahasiakan dan diberikan kepada umum (sehingga disebut dengan kunci publik), namun kunci untuk dekripsi bersifat rahasia (kunci privat). Oleh sebab itu untuk meperkuat keamanan dari algoritma RSA perlu dilakukan peroses pembuatan kunci dilakukan dengan memfaktorkan suatu bilangan bulat menjadi faktor-faktor primanya pada umunya. Dikarenakan haltersebuat sudah umum makanya perlu dilakukan modifikasi kunci pada algoritma RSA. Modifikas dilakukan dengan menggunkana algoritma lain yang menggunakan bilangan random Dengan memasukkan bilangan random, dianggap dapat menghilangkan kemungkinan penyerang menebak hasil dengan mengetahui algoritmanya. Telah banyak algoritma generator bilangan acak yang diusulkan dan digunakan hingga saat ini. Algoritma-algoritma tersebut menggunakan berbagai pendekatan berbeda untuk menghasilkan bilangan random seacak. Salah satu algoritma tersebut adalah algoritma Algoritma Blum Blum Shub (BBS). Tujuan dalam menggunakan algoritma Blum Blum Shub (BBS) ini adalah agar kunci yang dihasilkan lebih sulit ditebak sehingga mempersulit kriptanalis dalam membaca pesan atau informasi tersebut. Adapun proses modipikasi Kunci algoritma RSA dengan menggunkan Blum Blum Shub dijelaskan mengenai rancangan sistem penambahan modifikasi pada metode kriptografi yang diteliti. Secara garis besar, proses yang dilakukan pada penelitian ini digambarkan dengan block diagram berikut:



Gambar 1. Diagram Modipikasi Kunci Algoritma RSA

1. Proses Pembentukan Kunci Dengan menggunkan Algoritma Blum Blum Shub
   a. Menentukan 2 bilangan prima sembarang dan acak yang diwakilkan oleh variable p dan variabel q. Misalkan $p = 13$ dan $q = 17$
   b. Menghitung modulus (n) dengan formula n = p * q, dimana nilai p = 13 dan nilai q = 17.

Submitted: **23/06/2020**; Revised: **27/06/2020**; Accepted: **29/06/2020**; Published: **30/06/2020**

Building of Informatics, Technology and Science (BITS)
Volume 2, No 1, June 2020 Page 39-43
ISSN 2684-8910 (media cetak)
ISSN 2685-3310 (media online)

n = p*q
n = 13*17
n = 221

c. Menghitung nilai m yang akan digunakan untuk mencari *enciphering exponet* (e).
m = (p – 1) * (q – 1)
m = (13 – 1) * (17 – 1)
m = (12) * (16)
m = 192

d. Menghitung nilai *e* dengan formula : gcd(e,m) = 1, dengan syarat e = bilangan prima dan 1 < e < m. dimisalkan e = 5, maka formula gcd(5,192) = 1 bernilai *true*.

e. Menghitung nilai *deciphering exponent* (d) dengan menggunakan formula e * d = 1 mod (m), dimana nilai e dan m didapatkan dari langkah sebelumnya e = 5 dan m = 192.
e * d = 1 mod (m)
d = 1 + (k * m) / e
d = 1 (k * 192) /5
dengan nilai k = *integer* sembarang, maka dimisalkan nilai d yang akan diambil adalah d yang bernilai *integer*. Nilai d yang diambil kali ini adalah d = 77.

f. Dari langkah diatas, nilai n, e, dan d telah ditemukan yang berarti juga pasangan kunci telah terbentuk. Pasangan kunci publik (n,e) = (221, 5) Pasangan kunci rahasia (n,d) = (221, 77)

2. Dimisalkan terdapat himpunan karakter : "SAY" yang akan disandikan menggunakan kunci yang telah dibentuk pada langkah sebelumnya, maka terlebih dahulu karakter tersebut dikonversi kedalam bentuk numerik, proses konversi bisa menggunakan teknik tersendiri dari penggunaan atau menggunakan tabel ASCII decimal berikut :

Tabel 1. Koversi char SAY ke ASCII Desimal

Char	Ascii Desimal
S	83
A	65
Y	89

Dari tabel . hasil konversi setiap baris dijadikan satu deret, akan menghasilkan plaintext ASCII decimal (M) dari karakter yang akan dienkrip sebagai berikut : M = 836589

3. Untuk menjalankan proses enkripsi, digunakan kunci publik yang telah dibentuk sebelumnya yaitu kunci publik (221,5), dengan formula $C = M^e \bmod N$. Namun sebelum melakukan perhitungan, terlebih dahulu dilakukan pemecahan deret plaintext ASCII menjadi blok yang panjang digit setiap bloknya kurang dari panjang digit n. Pada contoh kali ini, (RSA) modulus yang digunakan adalah = 221 (2 digit), jadi untuk setiap blok dibatasi maksimal 2 digit per blok. Dilakukan pemenggalan 3 digit per blok karena karakter yang digunakan dalam proses enkripsi adalah karakter ASCII yang setiap karakternya 8 bit, agar maksimal bit 8 digit (2^3). Hasil perhitungan enkripsi dengan formula $C = M^e \bmod N$ terlihat pada tabel 2. dibawah ini.

Tabel 2. Hasil Enkripsi

Blok ke	M	e	n	C
1	83	5	221	96
2	65	5	221	78
3	89	5	221	102

Berdasrkan dari hasil perosess di atas bahwasannya Algoritma RSA bisa digunakan untuk pembentukan kunci dengan menggunkan algoritma Blum Blum Shub sebagai kunci yang digunkan pada algoritma RSA.

4. KESIMPULAN

Berdasarkan analisa dan pembahasan yang dilakukan pada modifikasi pembangkit kunci algoritma rSA dengan menerapkan algoritma Blum Blum Shub (BBS), maka dapat disimpulkan sebagai berikut :

1. Algoritma RSA mempunyai dua teknik pembacaan yaitu teknik enkripsi (mengubah file asli menjadi file yang tidak dapat dibaca) dan teknik dekripsi (mengubah file yang tidak dapat dibaca menjadi file asli)
2. Algoritma blum blum shub adalah cryptographically secure pseudorandom number generator (CSPRNG) yang paling sederhana dan paling mangkus (secara kompleksitas teori)
3. Keamanan algoritma blum blum shub terletak pada sulitnya memfaktorkan n. Nilai n tidak perlu rahasia dan dapat diumumkan secara publik.

REFERENCES

Submitted: **23/06/2020**; Revised: **27/06/2020**; Accepted: **29/06/2020**; Published**: 30/06/2020**

Building of Informatics, Technology and Science (BITS)
Volume 2, No 1, June 2020 Page 39-43
ISSN 2684-8910 (media cetak)
ISSN 2685-3310 (media online)

[1] P. Pahrizal and D. Pratama, "Implementasi Algoritma Rsa Untuk Pengamanan Data Berbentuk Teks," *Pseudocode*, vol. 3, no. 1, pp. 44–49, 2016.
[2] T. S. Waruwu and K. Telaumbanua, "Kombinasi Algoritma OTP Cipher dan Algoritma BBS dalam Pengamanan File," *JSM STMIK Mikroskil*, vol. 17, no. 1, pp. 119–126, 2016.
[3] M. B. Sanjaya and P. A. Telnoni, "Implementasi Random Number Blum-Blum-Shub Dan Chaotic Function Untuk Modifikasi Key Generating Pada Kriptografi Aes Implementasi Blum-Blum-Shub Dan Chaotic Function Untuk Modifikasi Key Generating Pada Aes Implementation of Blum-Blum-Shub and Chaotic Func," *J. Elektro Telekomun. Terap. Desember*, vol. 1, no. 1, pp. 154–165, 2015.
[4] M. I. Zulfikar, G. Abdillah, A. Komarudin, J. Informatika, and F. Sains, "Kriptografi untuk Keamanan Pengiriman Email Menggunakan Blowfish dan Rivest Shamir Adleman (RSA)," *Semin. Nas. Apl. Teknol. Inf. 2019*, vol. 2, no. 1, pp. 19–26, 2019.
[5] A. N. Agustina, Aryanti, and Nasron, "Pengamanan Dokumen Menggunakan Metode Rsa (Rivest Shamir Adleman) Berbasis Web," *Proceeding SENDI_U*, vol. 3, no. 3, pp. 14–19, 2017.
[6] A. Sidorenko and B. Schoenmakers, "Concrete security of the blum-blum-shub pseudorandom generator," *Lect. Notes Comput. Sci. (including Subser. Lect. Notes Artif. Intell. Lect. Notes Bioinformatics)*, vol. 3796 LNCS, pp. 355–375, 2005.

Submitted: **23/06/2020**; Revised: **27/06/2020**; Accepted: **29/06/2020**; Published**: 30/06/2020**