# CRYPTOGRAPHICALLY SECURE PSEUDORANDOM NUMBER GENERATOR MENGGUNAKAN BLUM BLUM SHUB DAN RSA UNTUK ALGORITMA RC4

**Ahmad Luthfi[1], Iwan Iwut Tritoasmoro[2], Koredianto Usman[3]**

[1]Teknik Telekomunikasi, Fakultas Teknik Elektro, Universitas Telkom

**Abstrak**

**Cryptographically Secure Pseudorandom Number Generator (CSPRNG) adalah pembangkit bilangan acak yang dapat menghasilkan bilangan yang tidak mudah diprediksi pihak lawan. Pembangkit tersebut cocok untuk kriptografi misalnya digunakan untuk pembangkitan elemen-elemen kunci. Kunci inilah yang memegang peranan sangat penting dalam masalah keamanan kriptografi. Semakin sulit kunci itu ditebak oleh pihak lawan maka kriptografi tersebut akan semakin aman dari serangan. Tidak seperti Pseudorandom Number Generator (PRNG) lainnya yang biasanya kurang aman terhadap serangan, Cryptographically Secure Pseudorandom Number Generator (CSPRNG) hadir dengan performansi lebih baik sehingga dapat mengatasi hal tersebut. Hal ini dimungkinkan karena Cryptographically Secure Pseudorandom Number Generator (CSPRNG) dirancang berdasarkan operasi matematika yang sulit seperti pemfaktoran bilangan menjadi faktor prima, logaritma diskrit dan sebagainya.**

**Blum Blum Shub dan modified RSA merupakan dua pembangkit bilangan acak Cryptographically Secure Pseudorandom Number Generator (CSPRNG). Kedua macam pembangkit bilangan acak ini dapat menghasilkan bilangan yang tidak mudah diprediksi (secure) pihak lawan karena secara statistik memiliki sifat-sifat yang bagus seperti lolos uji keacakan statistik dan tahan terhadap serangan (attack) yang serius. Kedua pembangkit bilangan acak ini akan difungsikan sebagai key untuk enkripsi pada algoritma RC4.**

**Setelah itu akan dilakukan pengujian dan analisis terhadap algoritma RC4 dengan parameter seperti variansi, distribusi frekuensi, waktu proses, avalanche effect dan brute force attack. Dengan demikian dapat diketahui performansi dari kedua pembangkit bilangan acak tersebut dan dapat diambil kesimpulan pembangkit mana yang mempunyai kinerja terbaik.**

**Dari hasil analisis terhadap parameter tersebut diperoleh bahwa RC4 dengan Blum Blum Shub memiliki keunggulan dibandingkan modified RSA dalam hal waktu proses yang lebih cepat, variansi yang lebih kecil, distribusi frekuensi yang lebih merata. Sedangkan untuk parameter avalanche effect, kedua pembangkit bilangan acak ini memiliki performansi yang sama.**

**Kata Kunci : CSPRNG, Blum Blum Shub, modified RSA, RC4**

**Abstract**

**Cryptographically Secure Pseudorandom Number Generator (CSPRNG) is a random number generator that can generate the unpredictable number from the attacker. The generator is suitable for cryptography for example used for generating key elements. This key has important role on security of cryptography. If the key is increasingly difficult predictabled, the cryptography is increasingly secure from the attacker. Not likely any other Pseudorandom Number Generator (PRNG) that usually is not enough secure from attack, Cryptographically Secure Pseudorandom Number Generator (CSPRNG) come with better performance so that can solve this matter. This is possible because Cryptographically Secure Pseudorandom Number Generator (CSPRNG) is built based on difficult mathematical operation for example getting the primes factor of any number, discrete logarithm and so on.**

**Blum Blum Shub and modified RSA are two random number generator of Cryptographically Secure Pseudorandom Number Generator (CSPRNG). Both of this random number generator can generate the unpredictable number from the attacker because have good characteristic statistically like pass the random test and powerful from serious attack. Both of this random number generator will be functioned as key for encryption in RC4 algorithm.**

**After that all, test and analysis will be done for RC4 algorithm with parameter such as variance, frequency distribution, time processing, avalanche effect and brute force attack. So that the performance from both of the random number generator can be known and which generator has the best performance can be concluded.**

**From the analysis for those parameter can be obtained that RC4 using Blum Blum Shub has better performance than modified RSA for these case such as faster time processing, smaller variance, smooth frequency distribution. Whereas for avalanche effect parameter, both of these key has same performance.**

**Keywords : CSPRNG, Blum Blum Shub, modified RSA, RC4**

# BAB I
# PENDAHULUAN

## 1.1 Latar Belakang

*Cryptographically Secure Pseudorandom Number Generator* merupakan suatu pembangkit yang dapat menghasilkan bilangan acak yang tidak mudah diprediksi oleh lawan sehingga cocok untuk kriptografi. Pembangkit ini memiliki karakteristik yang bagus seperti lolos uji keacakan statistik. Di samping itu juga tahan terhadap serangan yang serius yang bertujuan untuk memprediksi bilangan acak yang dihasilkan.

Bilangan acak (*random*) banyak digunakan dalam kriptografi, misalnya untuk pembangkitan elemen-elemen kunci pada algoritma OTP, pembangkitan *initialization vector* (*IV*) pada algoritma kunci simetri, pembangkitan parameter kunci pada sistem kriptografi kunci publik dan sebagainya. Yang dimaksud acak di sini adalah bilangan yang tidak mudah diprediksi oleh pihak lawan. Sayangnya sangat sulit memperoleh bilangan acak dalam praktek kriptografi. Tidak ada prosedur komputasi yang benar-benar menghasilkan bilangan acak secara sempurna. Bilangan acak yang dihasilkan dengan rumus-rumus matematika adalah bilangan acak semu, karena bilangan acak tersebut dapat berulang kembali secara periodik. Pembangkit bilangan acak semacam itu disebut *Pseudorandom Number Generator* (PRNG).

*Blum Blum Shub* dan *modified* RSA adalah dua macam pembangkit bilangan acak yang termasuk dalam *cryptographically secure pseudorandom number generator* (CSPRNG), difungsikan sebagai *key* untuk enkripsi pada algoritma RC4. Algoritma RC4 saat ini banyak diterapkan pada *software* aplikasi yang memakai RC4 sebagai algoritma enkripsi, seperti aplikasi *secure sms* pada ponsel yang menggunakan J2ME dengan Netbeans IDE, dan juga digunakan secara luas pada sistem keamanan seperti protokol SSL (*Secure Socket Layer*).

Mengingat potensi penggunaan RC4 di masa kini dan masa yang akan datang, maka analisis dan peningkatan performansi RC4 merupakan topik penelitian yang penting dan menarik, sehingga diharapkan akan memperkuat algoritma enkripsi RC4.

## 1.2 Tujuan Penulisan Tugas Akhir

Berikut ini beberapa tujuan dari penulisan Tugas Akhir ini, antara lain :

1. Melakukan simulasi dan analisis terhadap *Blum Blum Shub* dan *modified* RSA sebagai pembangkit bilangan acak *Cryptographically Secure Pseudorandom Number Generator* (CSPRNG)
2. Menggunakan kedua pembangkit bilangan acak tersebut pada algoritma RC4 sebagai *key* untuk enkripsi dan membandingkan diantara keduanya sehingga diketahui manakah yang mempunyai performansi lebih baik

## 1.3 Perumusan Masalah

Dalam Tugas Akhir ini terdapat beberapa masalah dan dirumuskan sebagai berikut :

1. Pembangkitan bilangan acak menggunakan *Blum Blum Shub* dan *modified* RSA
2. Perbandingan jenis-jenis CSPRNG (*Blum Blum Shub* dan *modified* RSA) dan memilih jenis CSPRNG yang terbaik dilihat dari beberapa parameter dan kondisi
3. Proses pembuatan *ciphertext* menggunakan algoritma RC4
4. Teknik penggabungan CSPRNG (*Blum Blum Shub* dan *modified* RSA) dengan algoritma RC4 dilihat dari beberapa parameter dan kondisi
5. Keuntungan dan kerugian pada proses penggabungan tersebut serta parameter yang mempengaruhinya
6. Teknik membuat simulasi untuk proses pembangkitan bilangan acak

## 1.4 Batasan Masalah

Dalam penulisan Tugas Akhir ini batasan-batasan dan ruang lingkupnya adalah :

1. Informasi yang akan dienkripsi (*plaintext*) adalah *character* (huruf, angka, atau simbol) dengan format .txt dan *image* dengan format .bmp 3 layer RGB
2. Metode CSPRNG yang digunakan adalah *Blum Blum Shub* dan *modified* RSA
3. Menggunakan algoritma RC4 untuk proses enkripsi dan dekripsi
4. Tidak membahas aspek keamanan *integrity*, *authentication* dan *nonrepudiation*
5. Tidak membahas penyadapan pada kanal transmisi dan asumsi tidak ada bit yang *error* pada saat transmisi
6. Simulasi dilakukan dengan menggunakan software Matlab 2007a

## 1.5 Metodologi Penelitian

Metodologi penelitian yang dilakukan dalam Tugas Akhir ini meliputi :

1. Penelitian dilakukan dalam bentuk simulasi program menggunakan software Matlab 2007a sehingga dimungkinkan untuk mengamati variable-variabel input dan meneliti pengaruhnya terhadap performansi *cryptography*
2. Pengumpulan data-data penunjang diperoleh dari hasil simulasi yang dilakukan dan dari data-data yang diperoleh dari *paper* pada daftar pustaka
3. Melakukan studi *literature* dengan mempelajari permasalahan yang berkaitan dengan sistem *cryptography*
4. Proses pengujian dilakukan dengan *input* berupa *character* (huruf, angka, atau simbol). Selain dari performansinya sendiri, dilakukan analisis statistik terhadap keluaran untuk mengetahui performansi dari *cryptography* tersebut

## 1.6 Sistematika Penulisan

Susunan penulisan dalam Tugas Akhir ini adalah sebagai berikut :

**BAB I PENDAHULUAN**

Bab ini membahas tentang latar belakang, maksud dan tujuan, permasalahan, batasan masalah, metodologi penelitian dan sistematika penulisan.

**BAB II DASAR TEORI**

Bab ini berisi teori dasar mengenai konsep kriptografi, metode pembangkitan bilangan acak CSPRNG yang difungsikan sebagai *key* untuk enkripsi pada algoritma RC4.

**BAB III PERANCANGAN DAN IMPLEMENTASI**

Bab ini berisi tahapan perancangan dan pemodelan sistem yang dilanjutkan dengan implementasi hasil perancangan ke dalam fungsi dan prosedur untuk masing-masing algoritma.

**BAB IV PENGUJIAN DAN ANALISIS**

Bab ini berisi hasil analisis dari hasil implementasi berdasarkan pengujian sistem secara keseluruhan dengan parameter-parameter yang telah ditentukan.

**BAB V KESIMPULAN DAN SARAN**

Bab ini berisi kesimpulan dari hasil analisis yang telah dilakukan dan saran-saran untuk pengembangan selanjutnya.

# BAB V
# KESIMPULAN DAN SARAN

## 5.1 Kesimpulan

1. Waktu untuk proses *key generation* menunjukkan bahwa *key without* CSPRNG membutuhkan waktu tercepat, diikuti oleh *Blum Blum Shub* dan *modified* RSA. Karena pada *modified* RSA terdapat operasi perpangkatan (*e*) dan modulus (*n*) sepanjang 512 *bit.*
2. Pada *key generation* akan dihasilkan tiga jenis *key* dengan distribusi frekuensi yang berbeda-beda. *Modified* RSA memiliki distribusi frekuensi sedikit lebih merata dibandingkan *Blum Blum Shub*. Hal ini berarti ditinjau dari sisi kunci, *modified* RSA memiliki performansi sedikit lebih bagus daripada *Blum Blum Shub*. Hal ini diperkuat juga oleh nilai variansi dari *key* tersebut, *modified* RSA memiliki variansi sebesar 0.0001 sedangkan *Blum Blum Shub* 0.0002. Makin kecil nilai suatu variansi makin merata distribusi frekuensinya yang berarti makin tinggi tingkat keacakan karakternya. Untuk *key without* CSPRNG yang merupakan *default* algoritma RC4, memiliki nilai variansi yang sangat besar yaitu 0.0115 dan distribusi frekuensi yang tidak merata sehingga tingkat keacakan karakternya pun rendah.
3. *Brute force attack* untuk *Blum Blum Shub* dan *modified* RSA memerlukan waktu relatif lebih lama daripada *key without* CSPRNG karena *Blum Blum Shub* dan *modified* RSA menggunakan bilangan prima pada proses *key generation*-nya sehingga *attacker* harus mengetahui pasangan bilangan prima *p* dan *q* sehingga dihasilkan *n* dimana $n = p \times q$. Jika *n* yang dihasilkan sesuai, berarti *p* dan *q* yang didapatkan benar. Sedangkan *key without* CSPRNG dihasilkan hanya dengan *padding* hingga panjang *key* menjadi 256 *bytes* tanpa memakai bilangan prima.
4. Uji keperiodikan terhadap ketiga *generator* mendapatkan hasil bahwa *Blum Blum Shub* memiliki koefisien korelasi dengan rataan sebesar 0.7476, *modified RSA* 0.7442 dan *without CSPRNG* 0.9971.
5. Tahap *preprocessing* pada algoritma RC4 untuk masing-masing data (*text* dan *image*) mempunyai karakteristik yang berbeda-beda sehingga mengakibatkan waktu prosesnya pun berbeda pula. Selain itu ukuran file (untuk *text*) dan ukuran *pixel* (untuk *image*) juga mempengaruhi waktu proses enkripsi dekripsi. Data *image* membutuhkan waktu enkripsi relatif lebih lama daripada data *text*.

6. Distribusi frekuensi *cipher* hasil enkripsi algoritma RC4 dengan *Blum Blum Shub key* terlihat lebih merata diantara ketiga *key* tersebut. Hal ini sesuai dengan nilai variansinya, yaitu *Blum Blum Shub* 4.407 x $10^{-5}$, *modified* RSA 4.917 x $10^{-5}$ dan *key without* CSPRNG 8.243 x $10^{-5}$. Untuk *cipher* dari *plain* berupa *text* monoton, *Blum Blum Shub* tetap memberi variansi yang terbaik dibanding yang lainnya, yaitu *Blum Blum Shub* sebesar 2.070 x $10^{-5}$, *modified* RSA 3.635 x $10^{-5}$ dan *key without* CSPRNG 2.863 x $10^{-5}$.
7. *Avalanche effect* RC4 dengan *Blum Blum Shub* dan *modified* RSA menunjukkan hasil yang sama ketika dilakukan perubahan kecil terhadap *plaintext*, yaitu sebesar 3.125% yang sangat jauh dari kriteria *avalanche effect* yang baik. Dengan demikian diperlukan suatu *key* yang memiliki tingkat keacakan karakter yang tinggi.
8. *Brute force attack* RC4 dilakukan dengan mengetahui terlebih dahulu *key* yang digunakan pada algoritma RC4 tersebut. Sementara itu untuk memperoleh *key* kita harus mendapatkan prima terlebih dahulu. Terdapat sekitar $5.67 \times 10^{74}$ buah bilangan prima 256 bit. Untuk *brute force attack Blum Blum Shub* dibutuhkan waktu $3.3049 \times 10^{66}$ tahun jika digunakan 1 *computer* dan $3.3049 \times 10^{64}$ tahun jika digunakan 100 *paralel computer*. Dan untuk *brute force attack modified* RSA dibutuhkan waktu $3.3047 \times 10^{66}$ tahun jika digunakan 1 *computer* dan $3.3047 \times 10^{64}$ tahun jika digunakan 100 *paralel computer*.

## 5.2 Saran

1. Menggunakan *number generator* lainnya hingga diperoleh *avalanche effect* yang bagus, seperti CSPRNG berbasis *chaos*.
2. Menggunakan bilangan prima 512 bit dalam pembangkitan bilangan acak.
3. Dapat diaplikasikan ke dalam hardware.

# DAFTAR PUSTAKA


[1] Andi. 2003. ***Memahami Model Enkripsi dan Security Data***. Yogyakarta : Penerbit Andi.

[2] Kurniawan, Yusuf. 2004. ***Kriptografi : Keamanan Internet dan Jaringan Komunikasi***. Bandung : Penerbit Informatika.

[3] Menezes, A. 2002. ***Evaluation of Securirty Level of Cryptography: RSA Signature Schemes.*** University of Waterloo.

[4] Menezes, A.J., P.C.V. Oorschot, and S.A. Vanstone. 2001. ***Handbook of Applied Cryptography.*** CRC Press

[5] Munir, Rinaldi. 2006. ***Kriptografi.*** Bandung : Penerbit Informatika.

[6] Ronald R. Rivest. ***Finding Four Million Large Random Primes.*** Laboratory for Computer Science Massachusetts Institute of Technology Cambridge.

[7] Schneier, Bruce. 1996. ***Applied Cryptography, Second Edition***. New York : Wiley.

[8] Stallings, William. 1999. ***Cryptography and Network Security (Principle and Practice)***. New Jersey : Prentice Hall.

[9] Tamici, Halga. 2005. ***Analisa Kinerja Cryptography SHS (Secure Hash Standard) pada DSS (Digital Signature Standard)***. Bandung : Tugas Akhir STTTelkom.

[10] www.primes.utm.edu/howmany.shtml, Januari 2008.