Jurnal Mahajana Inforamasi, Vol.1 No 1, 2016
e-ISSN: 2527-8290

KOMBINASI ALGORITMA ONE TIME PAD CIPHER DAN ALGORITMA BLUM BLUM SHUB DALAM PENGAMANAN FILE

Tomoyud Sintosaro Waruwu
Program Studi Sistem Informasi STMIK Methodis Binjai
tomoyud@gmail.com

Abstrak

Kriptografi adalah ilmu menyamarkan pesan sehingga hanya dikenal oleh pengirim dan penerima. Salah satu algoritma yang cukup aman dan sulit untuk memecahkan adalah algoritma one-time pad cipher. Namun, algoritma ini sangat tergantung pada keacakan kunci yang digunakan dan kunci hanya digunakan satu kali saja. Algoritma blum blum shub adalah sebuah algoritma untuk menghasilkan angka acak yang baik dan sulit untuk memprediksi. kombinasi dari kedua algoritma ini diharapkan untuk menciptakan lebih banyak dan lebih baik keamanan data dan dijamin

Kata Kunci : *Kriptografi, One Time Pad Cipher, Blum blum Shub*

1. PENDAHULUAN

Semakin pesatnya kemajuan teknologi memberikan berbagai kemudahan bagi setiap pihak dalam melakukan pertukaran informasi. Namun, kemudahan ini juga membawa ancaman karena banyak pihak yang tidak berwenang yang berusaha untuk mengambil informasi tersebut untuk kepentingan pribadi atau organisasi.

Untuk mengatasi ancaman ini, banyak pihak yang berusaha untuk menyandikan informasi yang mereka miliki sehingga pesan tersebut tidak memilik makna dan sulit untuk dipecahkan.

Salah satu cara yang dapat diterapkan dalam menyandikan pesan atau informasi tersebut adalah melakukan kriptografi. Kriptografi merupakan metode untuk mengamankan data, baik berupa teks maupun gambar. Metode ini dilakukan dengan melakukan penyandian pesan kedalam bentuk yang tidak dipahami oleh orang lain maupun pihak ketiga.

Secara umum ada dua tipe algoritma kriptografi berdasarkan kuncinya yaitu algoritma asimetris dan simetris. Algoritma asimetris terdiri atas 2 buah kunci yaitu kunci publik dan kunci privat. Kunci publik untuk melakukan enkripsi sedangkan kunci privat untuk melakukan dekripsi.

Sedangkan algoritma simetris adalah algoritma yang memiliki kunci enkripsi dan dekripsi yang sama. Kriptografi kunci simetris memiliki berbagai macam metode algoritma, salah satunya adalah algoritma One Time Pad Cipher.

Algoritma One Time Pad Cipher merupakan algoritma berjenis symmetric key yang artinya bahwa kunci yang digunakan untuk melakukan enkripsi dan deskripsi merupakan kunci yang sama. Dalam proses enkripsi, algoritma ini menggunakan cara stream cipher dimana cipher tersebut berasal dari hasil XOR antara bit plaintext.

Untuk mendapatkan hasil yang enkripsi yang maksimal, One Time Pad Cipher membutuhkan sebuah kunci random yang hanya digunakan dalam satu kali siklus pengiriman pesan. Dalam mendapatkan sebuah kunci random, terdapat beberapa algoritma yang telah teruji. Salah satu algoritma tersebut adalah blum blum shub. Tujuan dalam menggunakan algoritma blum blum shub ini adalah agar kunci yang dihasilkan lebih sulit ditebak sehingga mempersulit kriptanalis dalam membaca pesan atau informasi tersebut.

2. TINJAUAN PUSTAKA

2.1. Kriptografi

Kriptografi berasal dari bahasa Yunani dan terdiri atas dua kata, *kryptos* yang berarti tersembunyi dan *graphein* yang berarti menulis. Kriptografi pada awalnya didefinisikan sebagai ilmu dan seni untuk menjaga kerahasiaan pesan dengan cara 2 menyandikannya ke dalam bentuk yang tidak dapat dimengerti lagi maknanya.

Namun demikian, kriptografi berkembang sehingga tidak hanya terbatas pada menyandikan pesan, tetapi juga memberikan aspek keamanan lain. Oleh karena itu, definisi kriptografi diperbarui menjadi ilmu dan seni untuk meningkatkan aspek keamanan pesan.

Ada empat tujuan mendasar dari ilmu kriptografi yan juga merupakan aspek keamanan informasi yaitu:

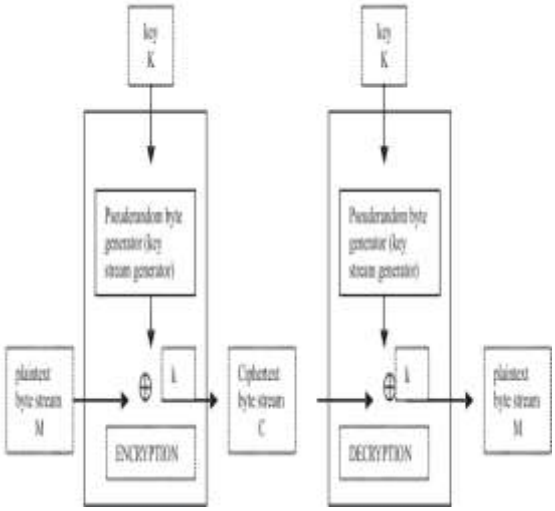
1. Kerahasiaan
   Layanan yang digunakan untuk menjaga isi informasi dari siapapun kecuali yang memiliki otoritas atau kunci rahasia untuk membuka/mengupas informasi yang telah disandi.
2. Integritas Data
   Berhubungan dengan penjagaan dari perubahan data secara tidak sah. Untuk menjaga integritas data, sistem harus memiliki kemampuan untuk mendeteksi manipulasi data dari pihak-pihak yang tidak berwenang, antara lain penyisipan, penghapusan, dan pensubsitusian data lain kedalam data yang sebenarnya.
3. Autentikasi
   Berhubungan dengan identitas/ pengenalan, baik secara kesatuan sistem atau informasi itu sendiri. Dua pihak yang saling berkomunikasi harus saling memperkenalkan diri.
4. Non repudiasi
   Tidak ada penyangkalan adalah usaha untuk mencegah terjadinya penyangkalan terhadap pengiriman/terciptanya suatu informasi oleh yang mengirimkan/membuat.

2.2. Algoritma One Time Pad Cipher

Algoritma One Time Pad Cipher adalah sebuah metode yang menerapkan algoritma kunci simetris atau proses enkripsi dan dekripsi menggunakan kunci yang sama. Kerahasian kunci merupakan faktor utama dalam penentuan keamanan atau pesan yang dikirimkan. Algoritma One Time Pad Cipher diciptakan oleh Mayor J. Maugboune dan Gilbert Vernam pada tahun 1917.

Algoritma One Time Pad Cipher adalah algoritma kriptografi yang sederhana dan mudah diimplementasikan karena hanya melakukan operasi XOR dan sudah dinyatakan oleh oleh para ahli kriptografi sebagai ”*Perfect encryption Algorithm”*.

Proses enkripsi maupun dekripsi, algoritma One Time Pad Cipher melakukan secara karakter per karakter atau sering juga disebut dengan metode stream cipher. Pada stream cipher, bila terjadi kesalahan selama transmisi maka kesalahan pada teks enkripsi penerima akan terjadi tepat di tempat kesalahan tersebut terjadi. Hasil enkripsi merupakan nilai yang didapatkan dari XOR antara plaintext dengan bit key.



Gambar 1. Proses Enkripsi dan Dekripsi Algoritma One Time Pad Cipher

Pada algoritma One Time Pad Cipher, plain text diubah kedalam ASCII. Nilai ASCII kemudian akan diubah kedalam barisan biner yang pada akhirnya akan dilakukan operasi XOR. Fungsi untuk melakukan proses enkrispi adalah

melakukan operasi XOR antara plaintext dengan kunci dan fungsi untuk melakukan dekripsi adalah melakukan operasi XOR antara cipher text dengan kunci. Secara sederhana, algoritma kriptografi One Time Pad Cipher dapat dituliskan sebagai berikut:

Untuk enkripsi :

$C_i = P_i \text{ XOR } K_i$

Untuk dekripsi :

$P_i = C_i \text{ XOR } K_i$

Yang dalam hal ini

Pi = Karakter plainteks

Ki = Karakter Kunci

Ci = Karakter Cipherteks

Sebagai contoh, akan dilakukan proses enkripsi terhadap plaintext **TOMOYUD** dengan menggunakan kunci secara yang didapat secara random yaitu **CVBKOHG**.

Langkah pertama adalah mengubah plaintext tersebut kedalam bentuk ASCII kemudian dilakukan proses konversi kedalam bentuk biner.

T = 84 = 01010100
O = 79 = 01001111
M = 77 = 01001101
O = 79 = 01001111
Y = 89 = 01011001
U = 85 = 01010101
D = 68 = 01000100

Kemudian lakukan proses yang sama terhadap kunci yang telah disiapkan sebelumnya.

c = 67 = 00100011
v = 86 = 01010110
b = 66 = 01000010
k = 75 = 01001011
o = 79 = 01001101
h = 72 = 01001000
g = 71 = 01000111

Jika setiap karakter pada plaintext maupun kunci telah diubah kedalam bentuk binary, langkah selanjutnya adalah melakukan operasi XOR antara plaintext dengan kunci

Tabel 1

Plaintext	Kunci	Hasil XOR
T = 01010100	c = 01100011	00110111 = 7
O = 01001111	v = 01110110	00111001 = 9
M = 01001101	b = 01100010	00101111 = /
O = 01001111	k = 01101011	00100100 = $
Y = 01011001	o = 01101111	00110110 = 6
U = 01010101	h = 01101000	00111101 = '='
D = 01000100	g = 01100111	00100011 = #

Dari penjelasan diatas, ada beberapa persyataran utama agar algoritma One Time Pad Cipher dapat digunakan secara maksimal :

1. Pemilihan kunci harus dilakukan secara acak agar tidak mudah ditebak
2. Jumlah karakter kunci harus sama panjang dengan karakter plain text.
3. Jika kunci tidak dapat diproduksi ulang maka algoritma dinyatakan aman.

2.3. Algoritma Blum Blum Shub

Pembangkit bilangan acak yang cocok untuk kriptografi adalah *cryptographically secure pseudorandom generator* (*CSPRNG*). Persyaratan dari CSPRNG adalah:

1. Secara statistik ia mempunyai sifat-sifat yang bagus (lolos uji keacakan statistik)
2. Tahan terhadap serangan (attack) yang serius. Serangan ini bertujuan untuk memprediksi bilangan acak yang dihasilkan.

Blum Blum Shub (*BBS*) adalah *CSPRNG* yang paling sederhana dan paling mangkus (secara kompleksitas teoritis). *BBS* dibuat pada tahun 1986 oleh Lenore Blum, Manuel Blum dan Michael Shub. Algoritma *BBS* dapat dijelaskan secara singkat sebagai berikut:

1. Pilih dua buah bilangan prima rahasia *p* dan *q*, yang masing-masing kongruen 3 modulo 4 (dalam praktek bilangan prima yang digunakan cukup besar).

2. Kalikan keduanya menjadi $n = pq$. Bilangan *n* ini disebut **bilangan bulat Blum.**

3. Pilih bilangan bulat acak lain, *s*, sebagai umpan sedemikian sehingga: (i) $2 \leq s \leq n$ (ii) *s* dan *n* relatif prima kemudian hitung $x0 = s2 \bmod n$

4. Barisan bit acak dihasilkan dengan melakukan iterasi berikut sepanjang yang diinginkan

a. Hitung $x_i = x_{i-1}^{\,e} \bmod n$ degan $x_0 = s$.

b. z_i = bit *LSB* (*Least Significant Bit*) dari x_i

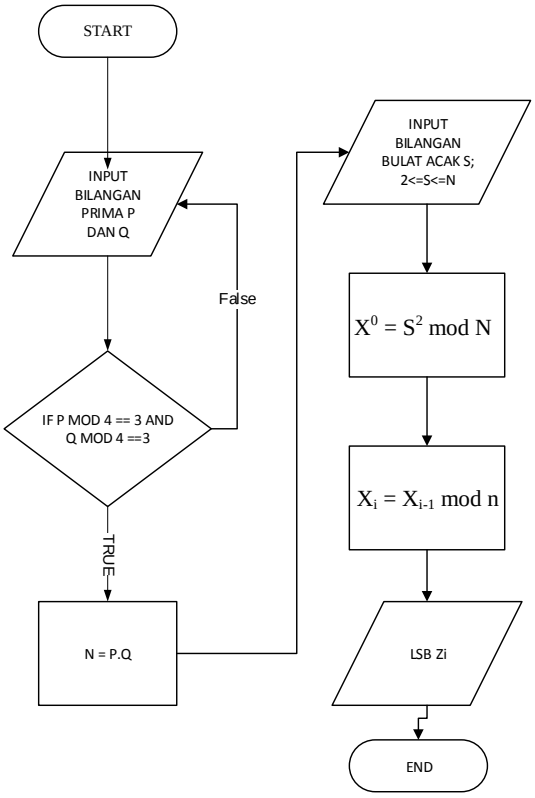
5. Barisan bit acak yang dihasilkan adalah *z*1, *z*2, *z*3, ...

Sebagai contoh, kita memilih p=11 dan q=19 sehingga n=pq = 209, selanjutnya kita pilih s=3. Kemudian untuk nilai e didapatkan dari rumus GCD(e,m)=1, oleh karena itu kita harus mendapatkan nilai m terlebih dahulu. m= (p-1)-(q-1) sehingga didapat m=(11-1)*(19-1), m=180. Kmudian untuk mencari nilai e, GCD(e,m), GCD(e,180)=1 sehingga nilai e=7. Setelah kita mendapatkan semua variabel, maka gunakan ketentuan yang ada untuk membangkitkan kunci tersebut dengan $X_{0=}$s, sehingga $X_{0=}$3.

$x_i = x_{i-1}^{e} \bmod n$ dengan x_0

demikian seterusnya. Perhatikan bahwa nilai *xi* yang mungkin hanya terletak antara 1 sampai 209 saja, sehingga pada suatu saat barisan tersebut akan berulang. Akibatnya, barisan bit yang dihasilkan pun juga akan berulang.

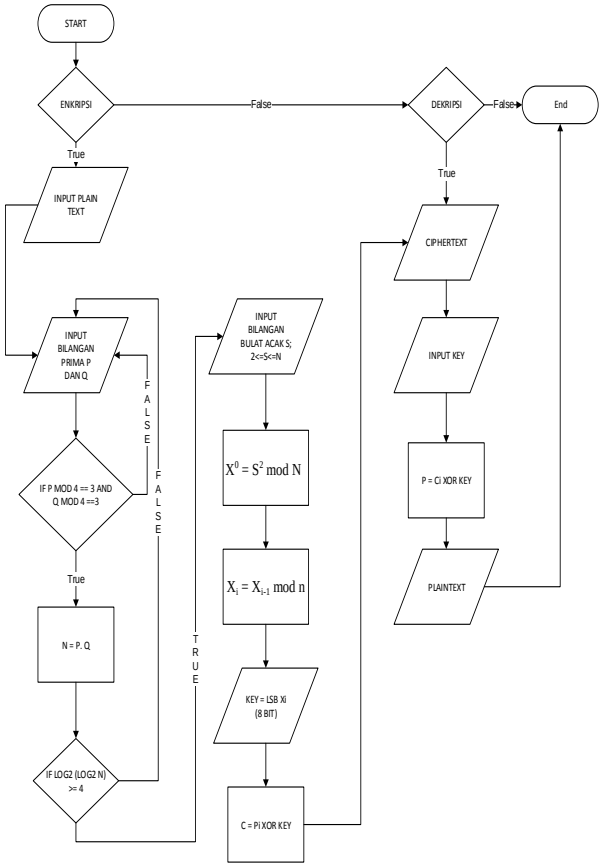
Jadi untuk nilai *n* yang kecil, maka *CSPRNG Blum Blum Shub* dapat dikatakan tidak aman, karena jika penyerang sudah mengetahui pola periodenya, maka tidak akan sulit untuk menebak bit yang dibangkitkan berikutnya. Sebenarnya untuk *n* besar sekalipun, jika pemilihan *s* tidak bagus, maka bisa terjadi periodenya kecil. Jika ini terjadi, penyerang juga bisa mengetahui barisan bit-bit yang dibangkitkan berikutnya. Bilangan acak yang diinginkan tidak harus 1 bit lsb, tetapi bisa juga j buah bit.



Gambar 2. Flowchart Algoritma Blum Blum Shub

3. HASIL DAN PEMBAHASAN

Pada tahap pertama, penulis akan mengimplementasikan penggunaan algoritma One Time Pad Cipher dan algoritma blum blum shub dalam pengamanan file.



Gambar 3. Flowchart algoritma One Time Pad Cipher dengan Algoritma Blum Blum Shub

Penulis mempunyai sebuah plaintext " PESAN RAHASIA". Tahap pertama yang kita lakukan adalah mengubah plaintext tersebut kedalam ASCII kemudian mengubahnya lagi kedalam bentuk biner. Dalam hal ini, penulis mengabaikan penggunaan spasi.

Tabel 2

Plaintext	ASCII	Notasi Biner
P	80	01010000
E	69	01000101
S	83	01010011
A	65	01000001
N	78	01001110
R	82	01010010
A	65	01000001
H	72	01001000
S	65	01000001
I	73	01001001
A	65	01000001

Pada tahap kedua, penulis akan melakukan proses pembangkitan kunci acak menggunakan algoritma blum blum shub. Panjang kunci yang digunakan harus sama dengan plaintext yang akan dijaga kerahasiaannya.

Penulis memilih p=9011 dan q=9851 sehingga n=pq=88767361. Kemudian penulis memilih s=59997801. Hitung $X_0 = S^2$ mod n sehingga $X_0=59997801^2$ mod 88767361=25834487. Pada tahap ini, hasil akhir tersebut akan diubah kedalam biner kemudian ambil kemudian ambil barisan bit sebanyak 8 (delapan)buah. Barisan bit tersebut dihitung dari barisan bit yang paling belakang.

$X_1 = X_0^2$ mod n = 25834487^2 mod 88767361
X_1 = 5839650 →10110010001101100100010

$X_2 = X_1^2$ mod n = 5839650^2 mod 88767361
X_2 = 21349213 →10100101110000110101110 1

$X_3 = X_2^2$ mod n = 21349213^2 mod 88767361
X_3 = 9397524 →10001111011001010001010 0

$X_4 = X_3^2$ mod n = 9397524^2 mod 88767361
X_4 = 52614730 →11001000101101011001001010

$X_5 = X_4^2$ mod n = 52614730^2 mod 88767361
X_5 = 63208858 →11110001000111110110011010

$X_6 = X_5^2$ mod n = 63208858^2 mod 88767361
X_6 = 18883951 →10010000000100101011011 11

$X_7 = X_6^2$ mod n = 18883951^2 mod 88767361
X_7 =83837599→10011111111010000101001111 1

$X_8 = X_7^2$ mod n = 83837599^2 mod 88767361
X_8 = 2816786 →10101011111011000100 10

$X_9 = X_8^2$ mod n = 2816786^2 mod 88767361
X_9 = 79108894 →10010110111000110110001111 0

$X_{10} = X_9^2$ mod n = 79108894^2 mod 88767361
$X_{1\,0}$= 10045745 →10011001010010010011000 1

$X_{11} = X_{10}^2$ mod n = 10045745^2 mod 88767361
$X_{1\,1}$= 42904955 →10100011101010110101111 011

$X_{12} = X_{11}^2$ mod n = 42904955^2 mod 88767361
X_{12} = 508997 →111110010001000101

Pada tahap ketiga, akan dilakukan proses enkripsi pada plaintext menggunakan kunci acak yang telah dibangkitkan sebelumnya.

Plaintext (P_1)	01010000
Kunci (K_1)	00100010
Ciphertext (C_1)	01110010

Plaintext (P_2)	01000101
Kunci (K_2)	01011101
Ciphertext (C_2)	00011000
Plaintext (P_3)	01010011
Kunci (K_3)	00010100
Ciphertext (C_3)	01000111
Plaintext (P_4)	01000001
Kunci (K_4)	01001010
Ciphertext (C_4)	00001011
Plaintext (P_5)	01001110
Kunci (K_5)	10011010
Ciphertext (C_5)	11010100
Plaintext (P_6)	01010010
Kunci (K_6)	01101111
Ciphertext (C_6)	00111100
Plaintext (P_7)	01000001
Kunci (K_7)	10011111
Ciphertext (C_7)	11011110
Plaintext (P_8)	01001000
Kunci (K_8)	00010010
Ciphertext (C_8)	01011010
Plaintext (P_9)	01000001
Kunci (K_9)	00011110
Ciphertext (C_9)	01011111
Plaintext (P_{10})	01010011
Kunci (K_{10})	00110001
Ciphertext (C_{10})	01100011
Plaintext (P_{11})	01001001
Kunci (K_{11})	01111011
Ciphertext (C_{11})	00110010
Plaintext (P_2)	01000001
Kunci (K_{12})	01000101
Ciphertext (C_{12})	01000100

Ciphertext yang masih terdiri dari notasi biner tersebut akan dikonversi kedalam ASCII kemudian diubah kedalam bentuk karakter.

Tabel 3

Notasi Biner	ASCII	Ciphertext
01110010	114	r
00011000	24	CAN
01000111	71	G
00001011	11	VT
11010100	212	’
00111100	60	<
11011110	222	fi
01011010	90	Z
01011111	95	_
01100011	99	c
00110010	50	2
01000100	68	D

4. KESIMPULAN

Dari penjelasan diatas, ada beberapa kesimpulan yang penulis dapatkan yaitu:

1. Algoritma One Time Pad Cipher adalah algoritma yang sederhana karena menggunakan operasi XOR dalam proses enkripsi data.
2. Kunci yang digunakan hanya dipakai satu kali dan panjang kunci harus sama dengan plaintext
3. Kunci yang dibangkitkan harus merupakan kunci acak sehingga mempersulitkan kriptanalis dalam mengetahui pesan tersebut.
4. Algoritma blum blum shub adalah *cryptographically secure pseudorandom number generator* (CSPRNG) yang paling sederhana dan paling mangkus (secara kompleksitas teori)
5. Keamanan algoritma blum blum shub terletak pada sulitnya memfaktorkan n. Nilai n tidak perlu rahasia dan dapat diumumkan secara publik
6. Algoritma blum blum shub tidak dapat diprediksi dari arah kiri (*unpredictable to the left*) dan tidak dapat diprediksi dari arah kanan (*unpredictable to the right)* artinya jika diberikan barisan bit yang dihasilkan oleh algoritma blum blum shub, kriptanalis tidak dapat memprediksi barisan bit sebelumnya dan barisan bit sesudahnya.

5. SARAN

Saran yang diberikan penulis pada penelitian ini adalah :

1. Untuk menjaga keamanan dari algoritma *blum blum shub* sebaiknya bilangan besar p dan q adalah bilangan prima dengan panjang lebih besar dari 2^{1024} dan menggunakan algoritma pencarian

bilangan prima yang tepat sehingga waktu yang dibutuhkan untuk menghasilkan bilangan acak semakin efesien

2. Untuk penelitian lebih lanjut diharapkan proses enkripsi dan dekripsi dapat mencakup pada data image, suara maupun video

DAFTAR PUSTAKA

Agustanti, Sri Primaini, *Pengamanan Kunci Enkripsi One Time Pad menggunakan Enkripsi RSA*, Jurnal Media Teknik, 2010

Blum, L., Blum, M., Shub, M., *A Simple Unpredictable Pseudo-Random Number Generator*, Society For Industrial and Applied Mathematics, 1986.

Blum, Manuel., Micali, Silvio, *How to Generate Cryptographically Strong Sequence of Pseudo Random Bits.*

Junod, Pascal, *Cryptographic Secure Pseudo-Random Bits Generation: The Blum Blum Shub Generator*, 1999.

Leung, Debbie W., *Quantum One Time Pad Cipher*, Quantum Information and Computation, 2001

Menezes, A., Oorschot, van P., Vanstone, S., *Handbook of Applied Cryptography*, CRC Press, 1996

Munir, Rinaldi, *Algoritma Enkripsi Citra dengan pseudo One-Time Pad yang menggunakan sistem chaos*, Konferensi Nasional Informatika, 2011

Munir, Rinaldi, *Kriptografi*, Penerbit Informatika, Bandung,2005

Rogaway, Phillip, *A Software-Optimized Encryption Algorithm*, Journal Of Cryptology, 1998,

Sholeh, M., Hamokwarong, J.V., *Aplikasi Kriptografi dengan metode One Time Pad Cipher dan Metode Permutasi*, Momentum, 2011

Sidorenko, Andrey,. Schoenmakers, Berry, *Concrete Security of The Blum-Blum-Shub Pseudorandom Generator*, IMA International Conference, 2005

Smart, Nigel, *Cryptography (An Introduction)*, McGraw-Hill Education, 2003

Stalling, William, *Cryptography and Network Security, Principle and Practice*, Pearson Education, 2003