PERANCANGAN SISTEM PENGAMANAN DAN OTENTIKASI PENGIRIMAN PESAN MENGGUNAKAN MODIFIKASI ALGORITMA ASIMETRI RSA

SYSTEM DESIGN OF SECURITY AND AUTHENTICATION DELIVERY MESSAGE USING MODIFIED RSA ASYMMETRIC ALGORITHM

[1]Fachrur Rozi [2] Leanna Vidya Yovita [3]Nur Andini

[1,2,3]Prodi S1 Teknik Telekomunikasi, Fakultas Teknik Elektrokomunikasi, Universitas Telkom

[1]harimausumatra11@gmail.com [2]leanna@telkomuniversity.ac.id [3]nurandini@telkomuniversity.ac.id

---

Pesan merupakan sesuatu hal yang bersifat rahasia, dimana pesan yang ditujukan oleh pengirim hanya boleh dibaca oleh penerima. Salah satu cara melakukan pengamanan pada pesan adalah dengan cara melakukan enkripsi pada pesan yang akan dikirim. Algoritma RSA merupakan salah satu algoritma kriptografi asimetri yang menggunakan dua kunci yang berbeda yaitu kunci publik dan kunci private. Kunci publik bersifat tidak rahasia sedangkan kunci private bersifat rahasia. Kedua kunci tersebut diatur sedemikian rupa sehingga memiliki hubungan dalam suatu persamaan aritmatik modulo. Proses enkripsi dan dekripsi menggunakan algoritma RSA memiliki beberapa kekurangan dua diantaranya adalah jumlah bilangan prima yang digunakan dalam pembangkitan pasangan kunci adalah dua dan terpisahnya proses otentikasi dari sistem enkripsi. Untuk mengatasi kedua kekurangan tersebut dirancanglah sistem pengamanan dan otentikasi pengiriman pesan menggunakan modifikasi algoritma asimetri RSA ini. Modifikasi yang digunakan adalah menggunakan modifikasi algoritma RSA n-prime dimana jumlah bilangan prima yang digunakan dalam pembangkitan pasangan kunci lebih dari 2 dan metode penggabungan proses otentikasi dalam sistem enkripsi dan dekripsi.

Kata kunci : kriptografi, RSA, modifikasi RSA, RSA n-prime

Message is something that is confidential, in which a message intended by the sender can only be read by the recipient. One way to do security at the message is to encrypt a message to be sent. RSA algorithm is one that uses asymmetric cryptographic algorithm which uses two different keys that public key and private key. The public key is not confidential, while the private key is confidential. Both keys are arranged such that they have a relationship in a modulo arithmetic equations. The process of encryption and decryption using RSA algorithm has some shortcomings, two of them is the number of primes that are used in the generation of the key pair is two and the separation of the authentication process of encryption systems. To overcome these shortcomings, it was a system designed messaging security and authentication using RSA asymmetric algorithm modification. Modifications are used is a modified n-prime RSA algorithm in which the number of primes that are used in the generation of key pairs more than 2 and a method of merging process in the system authentication encryption and decryption.

Keywords: cryptography, RSA, modified RSA, n-prime RSA

---

1. Pendahuluan

Dunia jaringan telekomunikasi kini berkembang begitu pesat. Saat ini perkembangan tidak hanya pada jalur komunikasi suara, namun juga pada jalur komunikasi data. Jarak antara si pengirim dan si penerima bukan lagi menjadi suatu masalah yang besar pada saat ini. Hal ini tentu saja menuntut adanya proses pengamanan pada data yang akan dikirim.Untuk menunjang kebutuhan akan proses pengamanan ini, salah satu caranya adalah dengan menggunakan proses enkripsi pada data yang dikirim. Dengan menggunakan proses ekripsi pada data yang dikirim, si pengirim bisa menjaga kerahasiaan pesan yang dikirimnya agar tidak diketahui oleh orang lain selain si penerima. Pada dasarnya, proses pengamanan mengunakan algoritma enkripsi RSA sudah baik, namun masih memiliki beberapa kekurangan, beberpapa diantaranya adalah panjang kunci yang digunakan dimana jika menggunakan kunci yang pendek maka akan gampang difaktorkan dan tidak terpisahnya proses otentikasi terhadap proses enkripsi yang dapat mengakibatkan kesamaan chipertext yang diterima dari hasil enkripsi siapa saja. Oleh karena itu pada tugas akhir ini dilakukan perancangan sistem pengamanan dan otentikasi pengiriman pesan menggunakan modifikasi algoritma enkripsi asimetri RSA. Modifikasi yang akan digunakan berupa gabungan fungsi modifikasi *n-prime* dalam pembangkitan kunci setiap user dan modifikasi penggabungan otentikasi dan enkripsi pada pesan yang akan dikirim.

2. Dasar Teori

A. RSA

RSA merupakan salah satu algoritma enkripsi asimetri. RSA di bidang kriptografi adalah sebuah algoritma pada enkripsi kunci public. Keamanan dari algoritma RSA terletak pada sulitnya memfaktorkan bilangan yang besar menjadi faktor-faktor prima. Penfaktoran dilakukan untuk memperoleh kunci pribadi. Selama penfaktoran bilangan belum ditemukan, maka selama itu pula keamanan algoritma RSA terjamin.

B. Properti algoritma RSA

Besaran-besaran yang digunakan pada algoritma RSA [5] :

1. p dan q bilangan prima (rahasia)
2. $n = p \times q$ (tidak rahasia)
3. $\phi(n) = (p-1)(q-1)$ (rahasia)
4. e (kunci enkripsi) (tidak rahasia)
5. d (kunci dekripsi) (rahasia)
6. m (plainteks) (rahasia)
7. c (cipherteks) (tidak rahasia)

C. Algoritma Membangkitkan Pasangan Kunci RSA

Urutan langkah untuk membangkitkan pasangan kunci dalam algoritma RSA[7]:

1. Pilih dua buah bilangan prima sembarang, p dan q.
2. Hitung $n = p \times q$ (sebaiknya $p \neq q$, sebab jika $p = q$ maka $n = p^2$ sehingga p dapat diperoleh dengan menarik akar pangkat dua dari n).
3. Hitung $\phi(n) = (p-1)(q-1)$.
4. Pilih kunci enkripsi, e, yang relatif prima terhadap $\phi(n)$.
5. Bangkitkan kunci dekripsi (d) dengan menggunakan persamaan $e \times d \equiv 1 \pmod{\phi(n)}$. Perhatikan bahwa $e \times d \equiv 1 \pmod{\phi(n)}$ ekivalen dengan $e \times d = 1 + k\phi(n)$, sehingga d dapat dihitung dengan persamaan :

$$d = \frac{1 + k\phi(n)}{e} \quad (1)$$

Dalam hal ini, k adalah suatu konstanta (1, 2, 3, ...) yang dicoba-coba hingga menghasilkan nilai d yang bulat.

6. Hasil dari algoritma di atas:
   - Kunci Public adalah pasangan (e, n)
   - Kunci Private adalah pasangan (d, n)

Catatan: n tidak bersifat rahasia, namun ia diperlukan pada perhitungan enkripsi/dekripsi.

D. Algoritma Enkripsi/Dekripsi RSA

Enkripsi [6]

1. Ambil kunci publik penerima pesan, e, dan modulus n.
2. Nyatakan plainteks m menjadi blok-blok m_1 , m_2 , …, sedemikian sehingga setiap blok merepresentasikan nilai di dalam selang $[0, n-1]$.
3. Setiap blok m_i dienkripsi menjadi blok c_i dengan rumus:

$$c_i = m_i^e \bmod n \quad (2)$$

Dekripsi [6]

1. Setiap blok cipherteks c_i didekripsi kembali menjadi blok m_i dengan rumus:

$$m_i = c_i^d \bmod n \quad (3)$$

E. Modified RSA menggunakan metode *n-prime*

Modified RSA menggunakan metode *n-prime* pada dasarnya hampir sama dengan algoritma RSA biasa namun yang membedakannya adalah masukan bilangan primanya tergantung keinginan user. Modifikasi ini dilakukan untuk menutupi kekurangan RSA dimana jika nilai $n = p \,.\, q$ sudah dapat difaktorkan maka selama itu pulalah keamaan RSA dapat diandalkan.

Prinsip cara kerja modified RSA menggunakan metode *n-prime* [3] :

1. User menentukan jumlah bilangan prima yang akan digunakan.
2. Misalkan user memilih menggunakan 4 bilangan prima sembarang p, q, r, s.
3. Hitung $n = p \times q \times r \times s$ (sebaiknya $p \neq q \neq r \neq s$, sebab jika $p = q = r = s$ maka $n = p^4$ sehingga p dapat diperoleh dengan menarik akar pangkat empat dari n).
4. Hitung $\phi(n) = (p-1)(q-1)(r-1)(s-1)$.

5. Pilih kunci enkripsi, *e*, yang relatif prima terhadap $\phi(n)$.
6. Bangkitkan kunci dekripsi (*d*) dengan menggunakan persamaan $e \times d \equiv 1 \pmod{\phi(n)}$. Perhatikan bahwa $e \times d \equiv 1 \pmod{\phi(n)}$ ekivalen dengan $e \times d = 1 + k\phi(n)$, sehingga *d* dapat dihitung dengan persamaan (1), Dalam hal ini, *k* adalah suatu konstanta (1, 2, 3, ...) yang dicoba-coba hingga menghasilkan nilai *d* yang bulat.
7. Hasil dari algoritma di atas:
   - Kunci public adalah pasangan (*e*, *n*)
   - Kunci private adalah pasangan (*d*, *n*)

Catatan: *n* tidak bersifat rahasia, namun ia diperlukan pada perhitungan enkripsi/dekripsi.

3. Pembahasan

A. Implementasi Sistem Pembangkitan Pasangan Kunci

Pada sub-bab ini dilakukan implementasi sistem pembangkitan pasangan kunci dimana terdapat seorang user yang ingin membangkitkan pasangan kunci RSA miliknya dengan menggunakan algoritma modified RSA *n-prime*.Pada implementasi ini dilakukan skenario user memilih nilai *np*=4 dengan p_1 sd p_4 = (13, 23, 43, 83) dan nilai *e* yang dipilih = 76831.

a) Masukan nilai *n-prime*

Pada tahap ini sistem meminta masukan nilai *n-prime* kepada user dan user memilih menggunakan jumlah *n-prime* sebanyak 4. Sehingga sistem memiliki nilai *np* = 4.

b) Masukan nilai bilangan prima p_i dimana i =1 sampai dengan *np*

Pada tahap ini sistem meminta masukan nilai p_i kepada user dan user memilih menggunakan nilai bilangan prima sebagai berikut: (13, 23, 43, 83).

c) Menghitung nilai *n* dan $\phi(n)$

Pada tahap ini sitem menghitung nilai *n* dan $\phi(n)$ sehingga diperoleh nilai *n* = 1067131 dan $\phi(n)$ = 909216

d) Masukan nilai *e* dimana $1<e<\phi(n)$ dan gdc $(e,\phi(n)) = 1$

Pada tahap ini sistem meminta masukan nilai *e* kepada user dan user memilih menggunakan nilai *e* = 76831. Kemudian sistem melakukan pengecekan terhadap nilai *e* apakah nilai masukan *e* yang digunakan user memenuhhi syarat *e* yang akan digunakan sistem.

e) Menghitung nilai *d*

Pada tahap ini sistem menghitung nilai *d* dengan menggunakan persamaan (1). Dimana nilai *k* merupakan konstanta tetap dengan nilai berkelanjutan (1, 2, 3, ...) yang dicoba sehingga sistem memperoleh nilai *d* bilangan bulat. Setelah melakukan perhitungan sistem memperoleh nilai *d* = 742783.

f) Keluaran pasangan kunci

Pada tahap ini sistem menampilkan hasil pasangan kunci yang telah dihasilkan dengan kombinasi nilai - nilai masukan dari user.

Adapun keluaran pasangan kunci yang telah dihasilkan sebagai berikut :

Kunci public = (76831 , 1067131)
Kunci private = (742783 , 1067131)

B. Implementasi Sistem Enkripsi Modified RSA

Pada sub-bab ini dilakukan implementasi sistem enkripsi dimana terdapat 2 orang user (user A dan user B) yang telah membangkitkan menggunakan algoritma modifikasi RSA. User A ingin mengirimkan message „transaksi di chiken attack jam 15.00" kepada user B.
Dengan data masing-masing user sebagai berikut :

Tabel 1 Data User Implementasi Sistem

User	Jumlah n-prime (*np*)	Nilai input bilangan prima (p_i , i=1 sd *np*)	Kunci Public (*e* , *n*)	Kunci Private (*d* , *n*)
A	2	(4327 , 3217)	(655787, 13919959)	(1250339, 13919959)
B	3	(6581, 47 , 83)	(345677, 25672481)	(11770533, 25672481)

a) Proses Konversi Message – Angka

Pada proses ini message „transaksi di chiken attack jam 15.00" dikonversi menjadi deretan angka menggunakan tabel maping angka – message sehingga hasil keluarannya sebagai berikut :

77755871765868766684616684606566686271845877775860688467587084465090555 5

Kemudian nilai ini dibagi menjadi blok-blok dengan format sebagai berikut :

panjang nilai setiap blok = panjang nilai „n" pada kunci pengirim – 1

Sehingga dihasilkan blok data numerik 1 sebagai berikut :

Tabel 2 Data Numerik 1

'7775587'	'1765868'
'7666846'	'1668460'
'6566686'	'2718458'
'7777586'	'0688467'
'5870844'	'6509055'
'5500000'	'0000000'

b) Proses Otentikasi

Pada proses ini akan dilakukan otentikasi menggunakan perhitungan algoritma RSA dengan rumus sebagai berikut:

$$c = x^d \ mod \ n \quad (4)$$

Dimana x adalah nilai dari setiap blok data numerik 1. [d, n] adalah pasangan kunci private pengirim. Dan c adalah hasil otentikasi dari nilai setiap blok data numerik 1.

Sehingga dihasilkan blok data numerik terotentikasi sebagai berikut:

Tabel 3 Data Numerik Terotentikasi

'575658'	'5002345'
'1910628'	'9061026'
'4839051'	'2153112'
'5718900'	'1634541'
'386285'	'2063081'
'800173'	'8840313'
'3527000'	'0'

c) Proses Pengecekan dan Pembagian Blok Baru

Pada proses ini akan dilakukan pengecekan terhadap blok data numerik terotentikasi dan kemudian dibagi menjadi blok – blok baru yang memenuhi kriteria sebagai inputan proses enkripsi dengan format dari pembentukan blok baru sebagai berikut:

panjang nilai setiap blok = panjang nilai „n" pada kunci penerima – 1

Sehingga menghasilkan blok data numerik 2 sebagai berikut:

Tabel 4 Data Numerik 2

'0575658'	'5002345'
'1910628'	'9061026'
'4839051'	'2153112'
'5718900'	'1634541'
'0386285'	'2063081'
'0800173'	'8840313'
'3527000'	'0000000'

d) Proses Enkripsi

Pada proses ini dilakukan enkripsi menggunakan perhitungan algoritma RSA, dengan rumus sebagai berikut :

$$c = x^e \ mod \ n \quad (5)$$

Dimana x adalah nilai dari setiap blok data numerik 2. [e, n] adalah pasangan kunci public penerima. Dan c adalah hasil enkripsi dari nilai setiap blok data numerik 2.

Sehingga dihasikan blok data numerik terenkripsi sebagai berikut :

Tabel 5 Data Numerik Terenkripsi

'7494939'	'12796954'
'12547648'	'14987475'
'15376617'	'22322427'
'24099017'	'17379375'
'9967867'	'19448730'
'16741222'	'19047497'
'15670713'	'0'

e) Proses Konversi Angka – Chipertext

Pada proses ini nilai blok data numerik terenkripsi akan dikonversi menjadi deretan angka sehingga hasil keluarannya sebagai berikut :

07494939127969541254764814987475153766172232242724099017173793750996786719448730167412
221904749715670713000000000

Kemudian hasil ini ini akan dikonversi menjadi deretan chipertext menggunakan tabel maping angka – chipertext sehingga dihasilkan chipertext sebagai berikut :

Q48uB04RY5H18h3n8qU4D87Hd4i7g72h2W0zA97I7J6s9j4jQ05hb6T4f71R3f5H8A6i7v

Chipertext ini lah yang akan dikirimkan oleh user A kepada user B.

C. Implementasi Sistem Dekripsi Modified RSA

Pada sub-bab ini dilakukan implementasi sistem dekripsi dimana terdapat 2 orang user (user A dan user B) yang telah membangkitkan menggunakan algoritma modifikasi RSA seperti yang tertera pada Tabel 1. User B menerima chipertext dari user A. Adapun chipertext yang diterima B sebagai berikut :

„Q48uB04RY5H18h3n8qU4D87Hd4i7g72h2W0zA97I7J6s9j4jQ05hb6T4f71R3f5H8A6i7v"

a) Proses Konversi Chipertext – Angka

Pada proses ini chipertext dikonversi menjadi deretan angka menggunakan tabel maping angka – chipertext sehingga hasil keluarannya sebagai berikut :

07494939127969541254764814987475153766172232242724099017173793750996786719448730167412219
047497156707130

Kemudian nilai ini dibagi menjadi blok-blok dengan format sebagai berikut:

panjang nilai setiap blok = panjang nilai „n" pada kunci penerima

Sehingga dihasilkan blok data numerik 1 sebagai berikut :

Tabel 6 Data Numerik 1

'07494939'	'12796954'
'12547648'	'14987475'
'15376617'	'22322427'
'24099017'	'17379375'
'09967867'	'19448730'
'16741222'	'19047497'
'15670713'	'00000000'

b) Proses Dekripsi

Pada proses ini dilakukan dekripsi menggunakan perhitungan algoritma RSA, dengan rumus sebagai berikut :

$$c = x^d \ mod \ n \quad (6)$$

Dimana x adalah nilai dari setiap blok data numerik 1. [d ,n] adalah pasangan kunci private penerima. Dan c adalah hasil dekripsi dari nilai setiap blok data numerik 1. Sehingga dihasilkan blok data numerik terdekripsi sebagai berikut:

Tabel 7 Data Numerik Terdekripsi

'575658'	'5002345'
'1910628'	'9061026'
'4839051'	'2153112'
'5718900'	'1634541'
'386285'	'2063081'
'800173'	'8840313'
'3527000'	'0'

c) Proses Pengecekan dan Pembagian Blok Baru

Pada proses ini akan dilakukan pengecekan terhadap blok data numerik terdekripsi dan kemudian dibagi menjadi blok – blok baru yang memenuhi kriteria sebagai inputan proses enkripsi. Adapun format dari pembentukan blok baru sebagai berikut:

panjang nilai setiap blok = panjang nilai „n" pada kunci pengirim

Sehingga dihasilkan blok data numerik 2 sebagai berikut:

Tabel 8 Data Numerik 2

'05756585'	'00234519'
'10628906'	'10264839'
'05121531'	'12571890'
'01634541'	'03862852'
'06308108'	'00173884'
'03133527'	'00000000'

d) Proses Otentikasi

Pada proses ini dilakukan otentikasi menggunakan perhitungan algoritma RSA, dengan rumus sebagai berikut :

$$c = x^e \bmod n \quad (7)$$

Dimana x adalah nilai dari setiap blok data numerik 2. [e ,n] adalah pasangan kunci public pengirim. Dan c adalah hasil otentikasi dari nilai setiap blok data numerik 2. Sehingga dihasilkan blok data numerik terotentikasi sebagai berikut :

Tabel 9 Data Numerik Terotentikasi

'7775587'	'1765868'
'7666846'	'1668460'
'6566686'	'2718458'
'7777586'	'688467'
'5870844'	'6509055'
'5500000'	'0'

e) Proses Konversi Angka – Message

Pada proses ini data numerik terotentikasi dikonversi menjadi deretan angka sehingga hasil keluarannya sebagai berikut :

77755871765868766684616684606566686271845877775860688467587084465090555 5

Kemudian hasil tersebut dikonversi menjadi deretan karakter menggunakan tabel maping angka – message sehingga dihasilkan message sebagai berikut :

transaksi di chiken attack jam 15.00

Message ini lah yang ingin disampaikan oleh user A kepada user B.

D. Hasil Pengujian Sistem

Setelah dilakukan beberapa kali pengujian terhadap sistem maka diperoleh hasil sebagai berikut:

a) Hasil Pengujian Brute Force Attack

Pada pengujian ini dilakukan pembandingan perkiraan hasil perhitungan kemungkinan percobaan pemfaktoran dari nilai *n* pada kunci yang telah dibangkitkan. Pada pengujian ini nilai *n* yang digunakan adalah bilangan 26273 (5 digit angka).

Jika nilai *n* merupakan perkalian 2 buah bilangan (*np*=2) maka dengan menggunakan perhitungan 2 kombinasi dari 26273 angka (C(26273;2)) diperoleh nilai kemungkinan percobaan pemfaktoran sebanyak 345148401 kemungkinan percobaan. Jika nilai *n* merupakan perkalian 3 (*np*=3) buah bilangan maka dengan menggunakan perhitungan 3 kombinasi dari 26273 angka (C(26273;3)) diperoleh nilai kemungkinan percobaan pemfaktoran sebanyak 3022924745425 kemungkinan percobaan. Dan karena nilai *np* dirahasiakan maka untuk melakukan percobaan pemfaktoran *n* sebesar 26273 (5digit) terdapat lebih dari 14 kemungkinan nilai *np* yang digunakan (2^{14}= 16384) yang mengakibatkan lebih dari 8,53 x 10^{50} (C(26273;14) = 8,53 x 10^{50}) kemungkinan kombinasi pemfaktoran nilai *n*.

Hal ini membuktikan dengan menggunakan modified *n-prime* pada pembangkitan kunci akan mempersulit kunci untuk difaktorkan menggunakan Brute Force Attack.

b) Hasil Pengujian Avalanche Effect

Pada pengujian ini dilakukan beberapa kali pengujian terhadap sistem enkripsi dan dekripsi modifikasi RSA yang telah dirancang dengan melakukan perubahan karakter masukan dan dianalisis perubahan yang terjadi pada chipertext yang dihasilkan. Dan juga dilakukan perbandingan hasil pengujian yang sama terhadap sistem enkripsi dan dekripsi RSA yang belum termodifikasi. Sehingga di peroleh nilai rata-rata sebagai berikut:

Tabel 10 Hasil pengujian avalanche effect

Nilai Avalanche Effect	Sistem Enkripsi RSA	Sistem Enkripsi Modifikasi RSA
Pembangkitan pasangan kunci RSA	0,1690631 %	0,2142406 %
Pembangkitan pasangan kunci modifikasi RSA n-prime	0,198316 %	0,228038 %

Dari Tabel 2 terlihat bahwa rata-rata nilai avalanche effect dari sistem enkripsi modified RSA kurang dari 50%, sedangkan „nilai avalanche effect optimal untuk suatu metoda kriptografi adalah 50%" [7] , dan „suatu algoritma bisa dikatakan memenuhi kriteria SAC (*Strict Avalanche Criterion*) apabila rata-rata perubahan bit keluaran terhadap berubahnya karakter masukan setidaknya adalah 50%" [5] . Nilai avalanche effect yang diperoleh pada sistem enkripsi modified RSA ini tidak mencapai nilai optimal disebabkan oleh adanya proses pembagian blok-blok dalam perhitungan sistem, yang mengakibatkan jika terjadi 1 bit perubahan pada nilai inputan maka hanya akan berpengaruh pada blok-blok yang berdekatan dengan bit yang berubah, sedangkan nilai dalam blok-blok yang lain akan bernilai tetap karna tidak terpengaruh oleh perubahan tersebut.

c) Hasil Pengujian Chi-Square

Pada pengujian ini dilakukan perhitungan dan perbandingan nilai chi-square dari keluaran sistem enkripsi dekripsi modifikasi RSA sehingga diperoleh nilai sebagai berikut:

Tabel 11 Hasil pengujian chi-square

Nilai Chi-Square	Sistem Enkripsi Modifikasi RSA
Pembangkitan pasangan kunci RSA	0,166999
Pembangkitan pasangan kunci modifikasi RSA n-prime	0,763792

Berdasarkan tabel nilai chi-square maka untuk α = 5% maka nilai x^2 maksimal harus sebesar 3,481. Dari hasil uji chi-square untuk masing-masing pasangan user nilai yang dihasilkan lebih kecil dari 3,481. Sehingga dapat dikatakan tidak ada hubungan antara dominannya sebuah karakter pada data masukan dengan karakter yang memiliki frekuensi kemunculan terbanyak pada chipertext yang dihasilkan oleh sistem.

d) Hasil Pengujian Waktu Performansi Sistem

Pada pengujian ini dihitung waktu yang dibutuhkan sistem enkripsi dan dekripsi modifikasi RSA yang telah dirancang untuk mengubah message menjadi chipertext dan dilakukan perbandingan terhadap waktu yang dibutuhkan sistem enkripsi dan dekripsi RSA yang belum di modifikasi dan juga dilakukan perbandingan terhadap sistem enkripsi dan dekripsi RSA dengan penyisipan otentikasi Hash. Dimana pada sistem enkripsi dan dekripsi RSA dengan penyisipan otentikasi Hash digunakan perulangan perhitungan xor 8bit sesuai dengan bit karakter inputan. Setalah dilakukan beberapa kali percobaan diperoleh nilai rata-rata waktu sebagai berikut:

Tabel 12 Hasil pengujian waktu performansi sistem

Waktu yang dibutuhkan sistem mengolah message menjadi chipertext	Sistem Enkripsi RSA	Sistem Enkripsi RSA Dengan Penyisipan Otentikasi Hash	Sistem Enkripsi Modifikasi RSA
Pembangkitan pasangan kunci RSA	0,337114 detik	0,384744 detik	0,758558 detik
Pembangkitan pasangan kunci modifikasi RSA n-prime	0,381089 detik	0,411596 detik	0,666938 detik

Dari hasil yang diperoleh pada pengujian ini waktu yang dibutuhkan sistem enkripsi modifikasi RSA dalam mengubah pesan menjadi chipertext lebih lama dibandingkan sistem enkripsi RSA maupun sistem enkripsi RSA dengan penyisipan otentikasi Hash. hal ini disebabkan oleh adanya penambahan proses otentikasi yang menggunakan perhitungan algoritma RSA yang melakukan pemangkatan sebanyak *x* kali dimana x merupakan nilai *d* pada kunci private pengirim pesan dan proses pengecekan dan pembagian blok baru yang terjadi antara proses konversi message – angka dan proses enkripsi.

4. Kesimpulan

Berdasarkan hasil analisis terhadap sistem yang telah dirancang dapat disimpulkan bahwa.Semakin besar nilai n-prime yang digunanakan dalam pembangkitan pasangan kunci maka akan semakin sulit pasangan kunci tersebut untuk di serang dengan brute force attack. Keluaran sistem enkripsi dan dekripsi modified RSA bergantung pada pasangan kunci user pengirim dan user penerima dan panjang pesan yang akan dikirim. Setelah diselidiki nilai avalanche effect dari sistem enkripsi modified RSA masih belum mencapai nilai diatas 50% untuk kasus perubahan 1 karakter ataupun perubahan 1 kata, namun nilai yang diperoleh menunjukkan peningkatan kualitas dibandingkan menggunakan sistem enkripsi RSA. Setelah diselidiki dengan metoda chi-square maka diperoleh nilai x^2 < 3,481 untuk hubungan antara data masukan dengan chipertext yang dihasilkan. Setelah diselidiki waktu yang dibutuhkan oleh sistem membutuhkan waktu dengan nilai rata-rata 0,7 detik dalam mengolah message menjadi chipertext sesuai dengan skenario yang dilakukan saat percobaan penilitian.

Daftar Pustaka

[1] Ayale, amare anagaw, 2013. "*A Modified RSA Encription Tecnique Based on Multiply public keys*", International Journal of Innovative Research in Computer an Communication Engineering, Vol 1, Issue 4, June 2013.

[2] Gunarto, 2000. "*Uji X^2*", Universitas Gunadarma.

[3] Ivy, B.Persis Urbana, 2012. "*A Modified RSA Criptosystem Based on 'n' Prime Number*", International journal of enginering an computer science ISSN:2319-7242 volume 1 issue 2 Nov 2012, page no. 63-66.

[4] Khyoon, Assad Ibraheem. "*Modification on the Algorithm of RSA Criptography System*", Electronic Departement College of Enginering Diala University.

[5] Munir, rinaldi, 2004. "*Algoritma RSA dan ElGamal*", Depatetemen Teknik Informatika Intitut Teknologi Bandung.

[6] Schneider, Bruce. 1996. "*Aplied Cryptography*", Jhon Wiley & Sons.

[7] Siregar, Ivan Michael, 2011. "*Membongkar Source Code Berbagai Aplikasi Android*", Yogyakarta, gava media, hal. 299-311.

[8] Zikrullah, Muhammad. 2011. "*Design of Security and Authentication System For Documents Delivery by Digital Signature Using Combined Ripemd-160 Hash Algorithm and Asymmetry RSA Encryption*", Fakultas Elektro dan Komunikasi Institut Teknologi Telkom Bandung.

LAMPIRAN

1. Tabel Maping Angka – Message

' = 5	~ = 6	! = 7	@ = 8	# = 9
$ = 10	% = 11	^ = 12	& = 13	* = 14
(= 15	) = 16	_ = 17	+ = 18	A = 19
B = 20	C = 21	D = 22	E = 23	F = 24
G = 25	H = 26	I = 27	J = 28	K = 29
L = 30	M = 31	N = 32	O = 33	P = 34
Q = 35	R = 36	S = 37	T = 38	U = 39
V = 40	W = 41	X = 42	Y = 43	Z = 44
` = 45	1 = 46	2 = 47	3 = 48	4 = 49
5 = 50	6 = 51	7 = 52	8 = 53	9 = 54
0 = 55	- = 56	= = 57	a = 58	b = 59
c = 60	d = 61	e = 62	f = 63	g = 64
h = 65	i = 66	j = 67	k = 68	l = 69
m = 70	n = 71	o = 72	p = 73	q = 74
r = 75	s = 76	t = 77	u = 78	v = 79
w = 80	x = 81	y = 82	z = 83	= 84
* = 85	+ = 86	\ = 87	; = 88	, = 89
. = 90	/ = 91	, = 92	- = 93	| = 94
: = 95	" = 96	< = 97	> = 98	? = 99

2. Tabel Maping Angka - Chipertext

x \ y	0	1	2	3	4	5	6	7	8	9
0	'0Z'	'a1'	'f1'	'k1'	'q1'	'v1'	'1a'	'1f'	'1k'	'1q'
1	'1v'	'A1'	'F1'	'K1'	'Q1'	'V1'	'1A'	'1F'	'1K'	'1Q'
2	'1V'	'a2'	'f2'	'k2'	'q2'	'v2'	'2a'	'2f'	'2k'	'2q'
3	'2v'	'A2'	'F2'	'K2'	'Q2'	'V2'	'2A'	'2F'	'2K'	'2Q'
4	'2V'	'a3'	'f3'	'k3'	'q3'	'v3'	'3a'	'3f'	'3k'	'3q'
5	'3v'	'A3'	'F3'	'K3'	'Q3'	'V3'	'3A'	'3F'	'3K'	'3Q'
6	'3V'	'a4'	'f4'	'k4'	'q4'	'v4'	'4a'	'4f'	'4k'	'4q'
7	'4v'	'A4'	'F4'	'K4'	'Q4'	'V4'	'4A'	'4F'	'4K'	'4Q'
8	'4V'	'a5'	'f5'	'k5'	'q5'	'v5'	'5a'	'5f'	'5k'	'5q'
9	'5v'	'A5'	'F5'	'K5'	'Q5'	'V5'	'5A'	'5F'	'5K'	'5Q'
10	'5V'	'a6'	'f6'	'k6'	'q6'	'v6'	'6a'	'6f'	'6k'	'6q'
11	'6v'	'A6'	'F6'	'K6'	'Q6'	'V6'	'6A'	'6F'	'6K'	'6Q'
12	'6V'	'a7'	'f7'	'k7'	'q7'	'v7'	'7a'	'7f'	'7k'	'7q'
13	'7v'	'A7'	'F7'	'K7'	'Q7'	'V7'	'7A'	'7F'	'7K'	'7Q'
14	'7V'	'a8'	'f8'	'k8'	'q8'	'v8'	'8a'	'8f'	'8k'	'8q'
15	'8v'	'A8'	'F8'	'K8'	'Q8'	'V8'	'8A'	'8F'	'8K'	'8Q'
16	'8V'	'a9'	'f9'	'k9'	'q9'	'v9'	'9a'	'9f'	'9k'	'9q'
17	'9v'	'A9'	'F9'	'K9'	'Q9'	'V9'	'9A'	'9F'	'9K'	'9Q'
18	'9V'	'a0'	'f0'	'k0'	'q0'	'v0'	'0a'	'0f'	'0k'	'0q'
19	'0v'	'A0'	'F0'	'K0'	'Q0'	'V0'	'0A'	'0F'	'0K'	'0Q'
20	'0V'	'b1'	'g1'	'l1'	'r1'	'w1'	'1b'	'1g'	'1l'	'1r'
21	'1w'	'B1'	'G1'	'L1'	'R1'	'W1'	'1B'	'1G'	'1L'	'1R'
22	'1W'	'b2'	'g2'	'l2'	'r2'	'w2'	'2b'	'2g'	'2l'	'2r'
23	'2w'	'B2'	'G2'	'L2'	'R2'	'W2'	'2B'	'2G'	'2L'	'2R'
24	'2W'	'b3'	'g3'	'l3'	'r3'	'w3'	'3b'	'3g'	'3l'	'3r'
25	'3w'	'B3'	'G3'	'L3'	'R3'	'W3'	'3B'	'3G'	'3L'	'3R'
26	'3W'	'b4'	'g4'	'l4'	'r4'	'w4'	'4b'	'4g'	'4l'	'4r'
27	'4w'	'B4'	'G4'	'L4'	'R4'	'W4'	'4B'	'4G'	'4L'	'4R'
28	'4W'	'b5'	'g5'	'l5'	'r5'	'w5'	'5b'	'5g'	'5l'	'5r'
29	'5w'	'B5'	'G5'	'L5'	'R5'	'W5'	'5B'	'5G'	'5L'	'5R'
30	'5W'	'b6'	'g6'	'l6'	'r6'	'w6'	'6b'	'6g'	'6l'	'6r'
31	'6w'	'B6'	'G6'	'L6'	'R6'	'W6'	'6B'	'6G'	'6L'	'6R'
32	'6W'	'b7'	'g7'	'l7'	'r7'	'w7'	'7b'	'7g'	'7l'	'7r'
33	'7w'	'B7'	'G7'	'L7'	'R7'	'W7'	'7B'	'7G'	'7L'	'7R'

34	'7W'	'b8'	'g8'	'l8'	'r8'	'w8'	'8b'	'8g'	'8l'	'8r'
35	'8w'	'B8'	'G8'	'L8'	'R8'	'W8'	'8B'	'8G'	'8L'	'8R'

◆ ◆	0	1	2	3	4	5	6	7	8	9
36	'8W'	'b9'	'g9'	'l9'	'r9'	'w9'	'9b'	'9g'	'9l'	'9r'
37	'9w'	'B9'	'G9'	'L9'	'R9'	'W9'	'9B'	'9G'	'9L'	'9R'
38	'9W'	'b0'	'g0'	'l0'	'r0'	'w0'	'0b'	'0g'	'0l'	'0r'
39	'0w'	'B0'	'G0'	'L0'	'R0'	'W0'	'0B'	'0G'	'0L'	'0R'
40	'0W'	'c1'	'h1'	'm1'	's1'	'x1'	'1c'	'1h'	'1m'	'1s'
41	'1x'	'C1'	'H1'	'M1'	'S1'	'X1'	'1C'	'1H'	'1M'	'1S'
42	'1X'	'c2'	'h2'	'm2'	's2'	'x2'	'2c'	'2h'	'2m'	'2s'
43	'2x'	'C2'	'H2'	'M2'	'S2'	'X2'	'2C'	'2H'	'2M'	'2S'
44	'2X'	'c3'	'h3'	'm3'	's3'	'x3'	'3c'	'3h'	'3m'	'3s'
45	'3x'	'C3'	'H3'	'M3'	'S3'	'X3'	'3C'	'3H'	'3M'	'3S'
46	'3X'	'c4'	'h4'	'm4'	's4'	'x4'	'4c'	'4h'	'4m'	'4s'
47	'4x'	'C4'	'H4'	'M4'	'S4'	'X4'	'4C'	'4H'	'4M'	'4S'
48	'4X'	'c5'	'h5'	'm5'	's5'	'x5'	'5c'	'5h'	'5m'	'5s'
49	'5x'	'C5'	'H5'	'M5'	'S5'	'X5'	'5C'	'5H'	'5M'	'5S'
50	'5X'	'c6'	'h6'	'm6'	's6'	'x6'	'6c'	'6h'	'6m'	'6s'
51	'6x'	'C6'	'H6'	'M6'	'S6'	'X6'	'6C'	'6H'	'6M'	'6S'
52	'6X'	'c7'	'h7'	'm7'	's7'	'x7'	'7c'	'7h'	'7m'	'7s'
53	'7x'	'C7'	'H7'	'M7'	'S7'	'X7'	'7C'	'7H'	'7M'	'7S'
54	'7X'	'c8'	'h8'	'm8'	's8'	'x8'	'8c'	'8h'	'8m'	'8s'
55	'8x'	'C8'	'H8'	'M8'	'S8'	'X8'	'8C'	'8H'	'8M'	'8S'
56	'8X'	'c9'	'h9'	'm9'	's9'	'x9'	'9c'	'9h'	'9m'	'9s'
57	'9x'	'C9'	'H9'	'M9'	'S9'	'X9'	'9C'	'9H'	'9M'	'9S'
58	'9X'	'c0'	'h0'	'm0'	's0'	'x0'	'0c'	'0h'	'0m'	'0s'
59	'0x'	'C0'	'H0'	'M0'	'S0'	'X0'	'0C'	'0H'	'0M'	'0S'
60	'0X'	'd1'	'i1'	'n1'	't1'	'y1'	'1d'	'1i'	'1n'	'1t'
61	'1y'	'D1'	'I1'	'N1'	'T1'	'Y1'	'1D'	'1I'	'1N'	'1T'
62	'1Y'	'd2'	'i2'	'n2'	't2'	'y2'	'2d'	'2i'	'2n'	'2t'
63	'2y'	'D2'	'I2'	'N2'	'T2'	'Y2'	'2D'	'2I'	'2N'	'2T'
64	'2Y'	'd3'	'i3'	'n3'	't3'	'y3'	'3d'	'3i'	'3n'	'3t'
65	'3y'	'D3'	'I3'	'N3'	'T3'	'Y3'	'3D'	'3I'	'3N'	'3T'
66	'3Y'	'd4'	'i4'	'n4'	't4'	'y4'	'4d'	'4i'	'4n'	'4t'
67	'4y'	'D4'	'I4'	'N4'	'T4'	'Y4'	'4D'	'4I'	'4N'	'4T'
68	'4Y'	'd5'	'i5'	'n5'	't5'	'y5'	'5d'	'5i'	'5n'	'5t'
69	'5y'	'D5'	'I5'	'N5'	'T5'	'Y5'	'5D'	'5I'	'5N'	'5T'

70	'5Y'	'd6'	'i6'	'n6'	't6'	'y6'	'6d'	'6i'	'6n'	'6t'

? ?	0	1	2	3	4	5	6	7	8	9
71	'6y'	'D6'	'I6'	'N6'	'T6'	'Y6'	'6D'	'6I'	'6N'	'6T'
72	'6Y'	'd7'	'i7'	'n7'	't7'	'y7'	'7d'	'7i'	'7n'	'7t'
73	'7y'	'D7'	'I7'	'N7'	'T7'	'Y7'	'7D'	'7I'	'7N'	'7T'
74	'7Y'	'd8'	'i8'	'n8'	't8'	'y8'	'8d'	'8i'	'8n'	'8t'
75	'8y'	'D8'	'I8'	'N8'	'T8'	'Y8'	'8D'	'8I'	'8N'	'8T'
76	'8Y'	'd9'	'i9'	'n9'	't9'	'y9'	'9d'	'9i'	'9n'	'9t'
77	'9y'	'D9'	'I9'	'N9'	'T9'	'Y9'	'9D'	'9I'	'9N'	'9T'
78	'9Y'	'd0'	'i0'	'n0'	't0'	'y0'	'0d'	'0i'	'0n'	'0t'
79	'0y'	'D0'	'I0'	'N0'	'T0'	'Y0'	'0D'	'0I'	'0N'	'0T'
80	'0Y'	'e1'	'j1'	'p1'	'u1'	'z1'	'1e'	'1j'	'1p'	'1u'
81	'1z'	'E1'	'J1'	'P1'	'U1'	'Z1'	'1E'	'1J'	'1P'	'1U'
82	'1Z'	'e2'	'j2'	'p2'	'u2'	'z2'	'2e'	'2j'	'2p'	'2u'
83	'2z'	'E2'	'J2'	'P2'	'U2'	'Z2'	'2E'	'2J'	'2P'	'2U'
84	'2Z'	'e3'	'j3'	'p3'	'u3'	'z3'	'3e'	'3j'	'3p'	'3u'
85	'3z'	'E3'	'J3'	'P3'	'U3'	'Z3'	'3E'	'3J'	'3P'	'3U'
86	'3Z'	'e4'	'j4'	'p4'	'u4'	'z4'	'4e'	'4j'	'4p'	'4u'
87	'4z'	'E4'	'J4'	'P4'	'U4'	'Z4'	'4E'	'4J'	'4P'	'4U'
88	'4Z'	'e5'	'j5'	'p5'	'u5'	'z5'	'5e'	'5j'	'5p'	'5u'
89	'5z'	'E5'	'J5'	'P5'	'U5'	'Z5'	'5E'	'5J'	'5P'	'5U'
90	'5Z'	'e6'	'j6'	'p6'	'u6'	'z6'	'6e'	'6j'	'6p'	'6u'
91	'6z'	'E6'	'J6'	'P6'	'U6'	'Z6'	'6E'	'6J'	'6P'	'6U'
92	'6Z'	'e7'	'j7'	'p7'	'u7'	'z7'	'7e'	'7j'	'7p'	'7u'
93	'7z'	'E7'	'J7'	'P7'	'U7'	'Z7'	'7E'	'7J'	'7P'	'7U'
94	'7Z'	'e8'	'j8'	'p8'	'u8'	'z8'	'8e'	'8j'	'8p'	'8u'
95	'8z'	'E8'	'J8'	'P8'	'U8'	'Z8'	'8E'	'8J'	'8P'	'8U'
96	'8Z'	'e9'	'j9'	'p9'	'u9'	'z9'	'9e'	'9j'	'9p'	'9u'
97	'9z'	'E9'	'J9'	'P9'	'U9'	'Z9'	'9E'	'9J'	'9P'	'9U'
98	'9Z'	'e0'	'j0'	'p0'	'u0'	'z0'	'0e'	'0j'	'0p'	'0u'
99	'0z'	'E0'	'J0'	'P0'	'U0'	'Z0'	'0E'	'0J'	'0P'	'0U'

Keterangan :
Koordinat (xy) = 'chipertext'
contoh :
980 (x=98, y=0) = '9Z'
885 (x=88, y=5) = 'z5'

3. Tabel ASCII (American Standard Code for Information Interchange)

```
Decimal Octal Hex Binary Value
------- ----- --- ------ -----
000 000 000 00000000 NUL (Null char \0)
001 001 001 00000001 SOH (Start of Header)
002 002 002 00000010 STX (Start of Text)
003 003 003 00000011 ETX (End of Text)
004 004 004 00000100 EOT (End of Transmission)
005 005 005 00000101 ENQ (Enquiry)
006 006 006 00000110 ACK (Acknowledgment)
007 007 007 00000111 BEL (Bell \a)
008 010 008 00001000 BS (Backspace \b)
009 011 009 00001001 HT (Horizontal Tab \t)
010 012 00A 00001010 LF (Line Feed \n)
011 013 00B 00001011 VT (Vertical Tab \v)
012 014 00C 00001100 FF (Form Feed \f)
013 015 00D 00001101 CR (Carriage Return \r)
014 016 00E 00001110 SO (Shift Out)
015 017 00F 00001111 SI (Shift In)
016 020 010 00010000 DLE (Data Link Escape)
017 021 011 00010001 DC1 (XON) (Device Control 1)
018 022 012 00010010 DC2 (Device Control 2)
019 023 013 00010011 DC3 (XOFF)(Device Control 3)
020 024 014 00010100 DC4 (Device Control 4)
021 025 015 00010101 NAK (Negative Acknowledgement)
022 026 016 00010110 SYN (Synchronous Idle)
023 027 017 00010111 ETB (End of Trans. Block)
024 030 018 00011000 CAN (Cancel)
025 031 019 00011001 EM (End of Medium)
026 032 01A 00011010 SUB (Substitute)
027 033 01B 00011011 ESC (Escape)
028 034 01C 00011100 FS (File Separator)
029 035 01D 00011101 GS (Group Separator)
030 036 01E 00011110 RS (Request to Send)
(Record Separator)
031 037 01F 00011111 US (Unit Separator)
032 040 020 00100000 SP (Space)
033 041 021 00100001 ! (exclamation mark)
034 042 022 00100010 " (double quote)
035 043 023 00100011 # (number sign)
036 044 024 00100100 $ (dollar sign)
037 045 025 00100101 % (percent)
038 046 026 00100110 & (ampersand)
039 047 027 00100111 ' (single quote)
040 050 028 00101000 ( (left/opening parenthesis)
041 051 029 00101001 ) (right/closing parenthesis)
042 052 02A 00101010 * (asterisk)
043 053 02B 00101011 + (plus)
044 054 02C 00101100 , (comma)
045 055 02D 00101101 - (minus or dash)
046 056 02E 00101110 . (dot)
047 057 02F 00101111 / (forward slash)
048 060 030 00110000 0
```

```
049 061 031 00110001 1
050 062 032 00110010 2
051 063 033 00110011 3
052 064 034 00110100 4
053 065 035 00110101 5
054 066 036 00110110 6

Decimal Octal Hex Binary Value
------- ----- --- ------ -----

055 067 037 00110111 7
056 070 038 00111000 8
057 071 039 00111001 9
058 072 03A 00111010 : (colon)
059 073 03B 00111011 ; (semi-colon)
060 074 03C 00111100 < (less than)
061 075 03D 00111101 = (equal sign)
062 076 03E 00111110 > (greater than)
063 077 03F 00111111 ? (question mark)
064 100 040 01000000 @ (AT symbol)
065 101 041 01000001 A
066 102 042 01000010 B
067 103 043 01000011 C
068 104 044 01000100 D
069 105 045 01000101 E
070 106 046 01000110 F
071 107 047 01000111 G
072 110 048 01001000 H
073 111 049 01001001 I
074 112 04A 01001010 J
075 113 04B 01001011 K
076 114 04C 01001100 L
077 115 04D 01001101 M
078 116 04E 01001110 N
079 117 04F 01001111 O
080 120 050 01010000 P
081 121 051 01010001 Q
082 122 052 01010010 R
083 123 053 01010011 S
084 124 054 01010100 T
085 125 055 01010101 U
086 126 056 01010110 V
087 127 057 01010111 W
088 130 058 01011000 X
089 131 059 01011001 Y
090 132 05A 01011010 Z
091 133 05B 01011011 [ (left/opening bracket)
092 134 05C 01011100 \ (back slash)
093 135 05D 01011101 ] (right/closing bracket)
094 136 05E 01011110 ^ (caret/circumflex)
095 137 05F 01011111 _ (underscore)
096 140 060 01100000 `
097 141 061 01100001 a
098 142 062 01100010 b
099 143 063 01100011 c
100 144 064 01100100 d
101 145 065 01100101 e
102 146 066 01100110 f
103 147 067 01100111 g
104 150 068 01101000 h
105 151 069 01101001 i
```

106 152 06A 01101010 j
107 153 06B 01101011 k
108 154 06C 01101100 l
109 155 06D 01101101 m
110 156 06E 01101110 n
111 157 06F 01101111 o
112 160 070 01110000 p

Decimal Octal Hex Binary Value
------- ----- --- ------ -----

113 161 071 01110001 q
114 162 072 01110010 r
115 163 073 01110011 s
116 164 074 01110100 t
117 165 075 01110101 u
118 166 076 01110110 v
119 167 077 01110111 w
120 170 078 01111000 x
121 171 079 01111001 y
122 172 07A 01111010 z
123 173 07B 01111011 { (left/opening brace)
124 174 07C 01111100 | (vertical bar)
125 175 07D 01111101 } (right/closing brace)
126 176 07E 01111110 ~ (tilde)
127 177 07F 01111111 DEL (delete)

4. Tabel Chi-Square

	P								
DF	**0.20**	**0.10**	**0.05**	**0.025**	**0.02**	**0.01**	**0.005**	**0.002**	**0.001**
1	1.642	2.706	3.841	5.024	5.412	6.635	7.879	9.550	10.828
2	3.219	4.605	5.991	7.378	7.824	9.210	10.597	12.429	13.816
3	4.642	6.251	7.815	9.348	9.837	11.345	12.838	14.796	16.266
4	5.989	7.779	9.488	11.143	11.668	13.277	14.860	16.924	18.467
5	7.289	9.236	11.070	12.833	13.388	15.086	16.750	18.907	20.515
6	8.558	10.645	12.592	14.449	15.033	16.812	18.548	20.791	22.458
7	9.803	12.017	14.067	16.013	16.622	18.475	20.278	22.601	24.322
8	11.030	13.362	15.507	17.535	18.168	20.090	21.955	24.352	26.124
9	12.242	14.684	16.919	19.023	19.679	21.666	23.589	26.056	27.877
10	13.442	15.987	18.307	20.483	21.161	23.209	25.188	27.722	29.588
11	14.631	17.275	19.675	21.920	22.618	24.725	26.757	29.354	31.264
12	15.812	18.549	21.026	23.337	24.054	26.217	28.300	30.957	32.909
13	16.985	19.812	22.362	24.736	25.472	27.688	29.819	32.535	34.528
14	18.151	21.064	23.685	26.119	26.873	29.141	31.319	34.091	36.123
15	19.311	22.307	24.996	27.488	28.259	30.578	32.801	35.628	37.697
16	20.465	23.542	26.296	28.845	29.633	32.000	34.267	37.146	39.252
17	21.615	24.769	27.587	30.191	30.995	33.409	35.718	38.648	40.790
18	22.760	25.989	28.869	31.526	32.346	34.805	37.156	40.136	42.312
19	23.900	27.204	30.144	32.852	33.687	36.191	38.582	41.610	43.820
20	25.038	28.412	31.410	34.170	35.020	37.566	39.997	43.072	45.315