Implementasi NTRU Cryptosystem

Josep Andre Ginting / 13517108[1]
Program Studi Teknik Informatika
Sekolah Teknik Elektro dan Informatika
Institut Teknologi Bandung, Jl. Ganesha 10 Bandung 40132, Indonesia
[1]13517108@std.stei.itb.ac.id

***Abstract*—Algoritma *NTRU Cryptosystem* merupakan salah satu dari banyak algoritma kunci publik. Meskipun algoritma ini masih kurang populer, algoritma ini memiliki beberapa kelebihan dibandingkan dengan algoritma kriptografi kunci publik lainnya. Salah satu kelebihan dari algoritma ini dibanding algoritma kriptografi kunci publik lainnya adalah dari segi pembangkitan kunci yang lebih cepat, efisiensi dalam proses enkripsi dan dekripsi, penggunaan memori yang rendah, dan juga ketahanan dari serangan yang berorientasi pada komputer kuantum.**

***Keywords*—NTRU , NTRUEncrypt, kriptografi, kriptografi kunci publik, implementasi.**

I. Pendahuluan

Terdapat banyak algoritma kriptografi kunci publik. Dari semua algoritma kriptografi kunci publik yang ada, terdapat beberapa algoritma yang sangat populer dan banyak diimplementasikan secara umum, antara lain algoritma *RSA* (*Rivest-Shamir-Adleman*), *Diffie-Hellman, Elgamal,* dan *ECC (Elliptic Curve Cryptography)*. Selain dari algoritma-algoritma kriptografi kunci publik tersebut, terdapat banyak algoritma lain yang kurang populer dan jarang diadaptasi secara luas, salah satunya adalah *NTRU Cryptosystem*. *NTRU* adalah kriptosistem kunci publik pertama yang tidak didasarkan pada faktorisasi atau masalah logaritma diskrit.

NTRU Cryptosystem sendiri terdiri dari dua algoritma yaitu *NTRUEncrypt* yang digunakan untuk enkripsi/ dekripsi dan juga *NTRUSign* yang digunakan untuk tanda tangan digital (*Digital Signature*). Kali ini hanya akan dibahas implementasi dari algoritma kriptografi kunci publik *NTRUEncrypt Cryptosystem* untuk enkripsi dan dekripsi pesan.

II. Dasar Teori

A. Kriptografi

Kriptografi adalah ilmu yang mempelajari teknik-teknik matematika yang berhubungan dengan aspek keamanan informasi seperti kerahasiaan (*confidentiality*), integritas data (*data integrity*), otentikasi (*authentication*), serta anti penyangkalan (*non repudiation*). kriptografi juga sering disebut sebagai ilmu dan seni untuk menjaga keamanan pesan.
Dalam kriptografi terdapat beberapa terminologi, yaitu:

1. Pesan/ plainteks (*plainteks*): informasi yang dapat dibaca dan dimengerti maknanya (baik dipersepsi secara visual maupun audial).
   Nama lain: plainteks (*plainteks*), *plain-image, plain-video, plain-audio*
2. Pengirim (*sender*): pihak yang mengirim pesan.
3. Penerima (*receiver*): pihak yang menerima pesan.
4. Cipherteks (*ciphertext*): pesan yang telah disandikan sehingga tidak bermakna lagi.
   Tujuan: agar pesan tidak dapat dibaca oleh pihak yang tidak berhak.
   Nama lain: kriptogram (*cryptogram*)
5. Enkripsi (*encryption*): proses menyandikan plainteks menjadi cipherteks.
   Nama lain: *enciphering*
6. Dekripsi (*decryption*): proses mengembalikan cipherteks menjadi plainteks semula.
   Nama lain: *deciphering*
7. *Cipher*: algoritma enkripsi dan dekripsi, atau fungsi matematika yang digunakan untuk enkripsi dan dekripsi pesan.
8. Kunci: parameter yang digunakan di dalam enkripsi dan dekripsi.
9. Penyadap (*eavesdropper*): orang/ mesin yang mencoba menangkap pesan selama ditransmisikan.
   Nama lain: *enemy, adversary, intruder, interceptor, bad guy*
10. Kriptanalisis (*cryptanalysis*): ilmu dan seni untuk memecahkan chiperteks menjadi plainteks tanpa mengetahui kunci yang digunakan. Pelakunya disebut kriptanalis.
11. Kriptologi (*cryptology*): studi mengenai kriptografi dan kriptanalisis.

Algoritma kriptografi sendiri dapat dikelompokkan menjadi tiga, yaitu:

1. Algoritma kriptografi simetri (*symmetric-key cryptography*)
   Ciri dari algoritma kriptografi simetri adalah kunci enkripsi sama dengan kunci dekripsi. Contoh dari algoritma kriptografi simetri adalah *data Encryption Standard* (DES), *Advanced Encryption Standard (*AES*)*, dll.
2. Algoritma kriptografi nir-simetri (*asymmetric-key cryptography*)
   Sering juga disebut kriptografi kunci publik (*public-key cryptography*). Terdapat dua kunci yang digunakan yaitu kunci enkripsi dan kunci dekripsi, dimana kunci enkripsi tidak sama dengan kunci dekripsi. Kunci enkripsi bersifat tidak rahasia (*public*

key), sedangkan kunci dekripsi bersifat rahasia (*private key*). Contoh dari algoritma kriptografi simetri adalah RSA (*Rivest-Shamir-Adleman*), *Elgamal, Diffie-Hellman, ECC (Elliptic Curve Cryptography),* dll.

3. Fungsi Hash
   Dilakukan dengan cara mengkompresi pesan ukuran sembarang menjadi *message-digest* berukuran *fixed.* Bersifat *irreversible* (tidak bisa dikembalikan menjadi pesan semula). Fungsi Hash berguna untuk memeriksa integritas dari pesan.

B. Kriptografi Kunci Publik

Kriptografi kunci publik (*public-key cryptography*) disebut juga kriptografi nir-simetri (*asymmetric-key cryptography*) karena kunci enkripsi tidak sama dengan kunci dekripsi. Istilah 'publik' muncul karena kunci untuk enkripsi diumumkan kepada publik (tidak rahasia). Hanya kunci privat yang rahasia, hanya pemilik kunci privat yang mengetahui kuncinya sendiri. Dibandingkan dengan kriptografi kunci simetri, kriptografi kunci publik memiliki beberapa keuntungan yaitu:

1. Hanya kunci privat yang perlu dijaga kerahasiaannya oleh setiap entitas yang berkomunikasi. Tidak ada kebutuhan mengirim kunci kunci privat sebagaimana pada sistem simetri.
2. Pasangan kunci publik/ kunci privat tidak perlu diubah, bahkan dalam periode waktu yang panjang.
3. Dapat digunakan untuk mengamankan pengiriman kunci simetri.
4. Beberapa algoritma kunci-publik dapat digunakan untuk memberi tanda tangan digital pada pesan.

C. NTRU Cryptosystem

NTRU Cryptosystem adalah sebuah kriptografi kunci-publik *open-source* yang menggunakan *lattice-based cryptography* untuk mengenkripsi dan mendekripsi pesan. *NTRU Cryptosystem* dikembangkan pada tahun 1996 oleh ahli matematika Joffrey Hoffstein, Jill Pipher, dan Joseph H. Silverman. *NTRU Cryptosystem* sendiri terdiri dari dua algoritma yaitu *NTRUEncrypt Cryptosystem* yang digunakan untuk enkripsi/ dekripsi dan juga yang digunakan untuk tanda tangan digital (*Digital Signature*). Tidak seperti sistem kriptografi kunci-publik populer lainnya (*RSA* dan *ECC*), kriptografi kunci-publik *NTRUEncrypt Cryptosystem* tahan terhadap serangan menggunakan *Shor's algorithm*. *Shor's algorithm* sendiri adalah sebuah *polynomial-time quantum computer algorithm* (serangan berbasis komputer kuantum) untuk faktorisasi integer.

Deskripsi dari algoritma *NTRU*:

1. Notasi
   Secara khusus, operasi *NTRU* didasarkan pada objek dalam cincin polinomial yang terpotong (*truncated polynomial ring*) $R = \mathbb{Z}[X] / (X^N - 1)$ dengan perkalian konvolusi dan semua polinomial di dalam ring memiliki koefisien bilangan bulat dan derajat paling banyak $N - 1$:

$$F = \sum_{i=0}^{N-1} F_i X^i = [F_0, F_1, ..., F_{N-1}]$$

NTRU Cryptosystem merupakan keluarga dari kriptosistem berparameter, dimana setiap sistem ditentukan oleh tiga parameter integer (N, p, q) yang mewakili derajat maksimal $N - 1$ untuk semua polinom di cincin terpotong R, masing-masing modulus dan modulus besar, dimana diasumsikan bahwa N adalah bilangan prima, q selalu lebih besar dari p, serta p dan q saling relatif prima; dan empat set polinomial L_f, L_g, L_ϕ, L_m dengan derajat paling banyak $N - 1$. Notasi yang ada pada algoritma *NTRUEncrypt* dapat dilihat pada tabel 2.1.

Tabel 2.1. Notasi dan keterangan pada *NTRUEncrypt Cryptosystem*

Notasi	Keterangan
N	Polinomial di cincin (*Ring*) R memiliki derajat N-1. (Tidak rahasia)
q	Modulus besar. (Tidak rahasia)
p	Modulus kecil. (Tidak rahasia)
f	Polinomial yang merupakan kunci privat.
g	Polinomial yang digunakan untuk menghasilkan kunci publik h dari f (Rahasia, dibuang setelah penggunaan awal)
h	Polinomial yang merupakan kunci publik.
r	Polinomial "*blinding*" acak (Rahasia, dibuang setelah penggunaan awal)
d	Koefisien

2. Pembangkitan Kunci
   Untuk membuat kunci *NTRU*, secara acak dipilih dua polinomial f dan g dengan derajat paling banyak $N - 1$ dan dengan koefisien dalam {-1, 0, 1}. Dimana polinomial f harus memenuhi persyaratan tambahan yang memiliki invers modulo q dan modulo p (dihitung dengan algoritma Euclidean) sehingga memenuhi $F_p \cdot f = 1 \ (mod\ p)$ dan juga $F_q \cdot f = 1 \ (mod\ q)$. f dan f_p (dan g) merupakan kunci private. Sedangkan kunci publik h dapat dihasilkan dengan cara: $h = F_q \cdot g \ (mod\ q)$.
3. Enkripsi
   Pesan diletakkan dalam bentuk polinomial m. Pilih secara acak polinomial r dengan koefisien kecil untuk mengaburkan pesan. Dengan kunci publik h, pesan terenkripsi e dihitung:
   $$e = r \cdot h + m \ (mod\ q).$$
4. Dekripsi
   Untuk bisa mendapatkan pesan m, pertama harus dilakukan penggandaan pesan terenkripsi e dan

bagian dari kunci privat f:

$$a = f \cdot e \,(mod\, q)$$

kemudian dipilih koefisien a dalam interval $[-q/2,\ q/2]$. Sekarang a dapat diperlakukan sebagai polinomial dengan koefisien integer untuk memulihkan pesan. Langkah berikutnya adalah dengan menghitung:

$$b = a \ (mod\, p) = f \cdot m \,(mod\, p)$$

untuk mendapatkan pesan m kembali, dilakukan perkalian b dan f_p:

$$c = f_p \cdot b = f_p \cdot f \cdot m \ (mod\, p)$$

dimana $c = m \,(mod\, p)$.

5. Keamanan

   Untuk dapat menghindari serangan kisi, serangan *brute force*, dan serangan *meet-in-the-middle*, polinomial f dan g harus memiliki sekitar 72 koefisien bukan nol. Berdasarkan penelitian terbaru, nilai parameter-parameter pada *NTRUEncrypt* yang dianggap aman adalah seperti gambar tabel dibawah.

Tabel 2.2. Parameter untuk keamanan *NTRUEncrypt Cryptosystem*

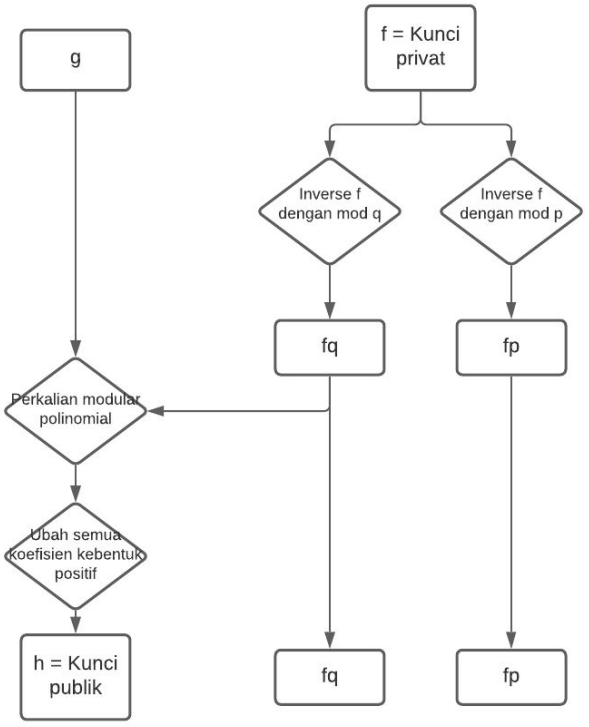
	N	q	p
Moderate Security	107	64	3
High Security	167	128	3
Highest Security	503	256	3

III. Implementasi

Implementasi dari *NTRUEncrypt Cryptosystem* menggunakan bahasa pemrograman *python*. Kode implementasi dapat diakses pada link: https://github.com/josepandre99/uas-kripto . Berikut adalah penjelasan proses implementasi operasi-operasi dasar dari *NTRUEncrypt*.

A. Pembangkitan Kunci

Untuk membangkitkan kunci, dibutuhkan beberapa parameter yaitu g yang merupakan polinomial, f yang merupakan kunci privat, serta p dan q yang akan digunakan untuk melakukan invers modulo terhadap kunci privat. Untuk lebih jelas mengenai proses pembangkitan kunci pada algoritma *NTRUEncrypt* dapat dilihat pada gambar 3.1.

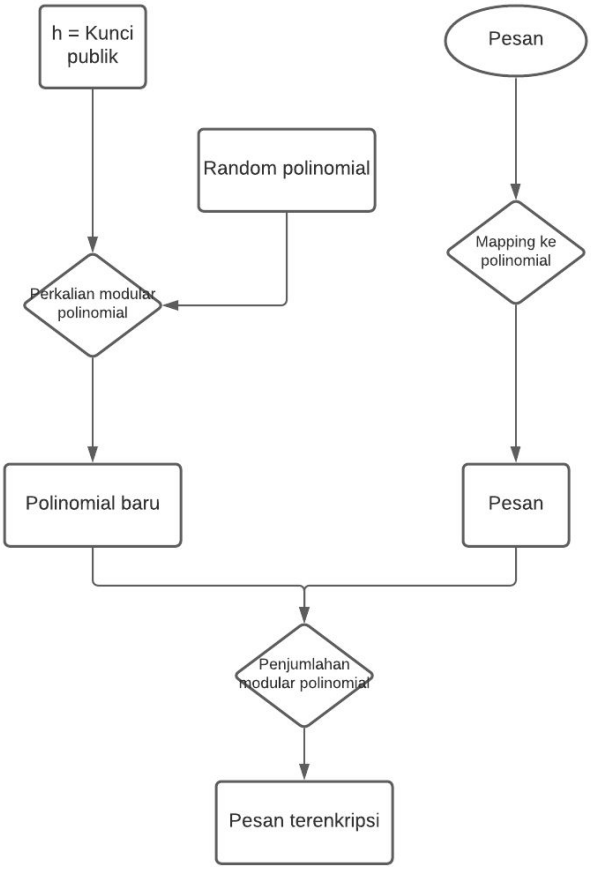


Gambar 3.1. Proses pembanhkitan kunci pada algoritma *NTRUEncrypt*

Kunci publik dapat diperoleh dengan melakukan pengubahan koefisien polinomial menjadi bentuk positif dari polinomial yang merupakan hasil dari perkalian modulo polinomial g dengan fq yang merupakan invers modulo dari kunci privat f dengan bilangan q.

B. Enkripsi

Proses enkripsi pesan pada *NTRUEncrypt* membutuhkan kunci publik dan juga sebuah polinomial acak r. Untuk lebih jelas mengenai proses enkripsi pesan pada algoritma *NTRUEncrypt* dapat dilihat pada gambar 3.2.

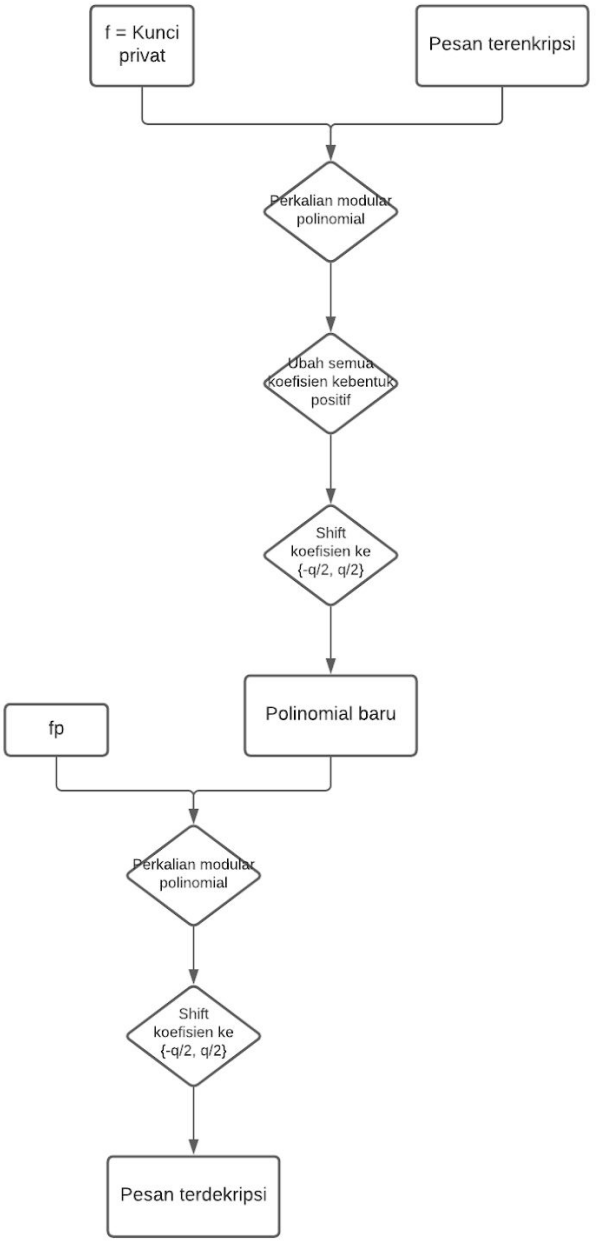


Gambar 3.2. Proses enkripsi pada algoritma *NTRUEncrypt*

Kunci publik h pada awalnya akan dikalikan dengan polinom acak r hingga menghasilkan polinom baru. Polinom baru tersebut kemudian dijumlahkan dengan pesan yang sudah di *mapping* ke dalam bentuk polinom.

C. Dekripsi

Proses dekripsi pesan pada *NTRUEncrypt* membutuhkan kunci privat dan juga sebuah polinomial fp yang merupakan hasil invers modulo dari kunci privat f dengan bilangan p. Untuk lebih jelas mengenai proses enkripsi pesan pada algoritma *NTRUEncrypt* dapat dilihat pada gambar 3.2.



Gambar 3.3. Proses dekripsi pesan pada algoritma *NTRUEncrypt*

Kunci private f pada awalnya akan dikalikan dengan pesan yang terenkripsi, kemudian koefisiennya di shift ke dalam bentuk $\{-q/2, q/2\}$ sehingga membentuk polinom baru. Polinom baru tersebut akan dikalikan dengan fp dan di shift lagi kedalam bentuk $\{-q/2, q/2\}$ sehingga diperoleh pesan yang sudah terdekripsi.

IV. Eksperimen, Analisis Keamanan dan Analisis Kinerja

A. Eksperimen

Dari hasil implementasi algoritma *NTRUEncrypt Cryptosystem*, akan dilakukan enkripsi dan dekripsi pada pesan teks dengan ukuran 62 *Bytes* dengan isi pesan seperti pada tabel 4.1. Pengujian dilakukan dengan *NTRUEncrypt* dengan parameter *Highest Security* (N=503, q=,256 p=3).

Tabel 4.1. Pesan teks berukuran 62 *Bytes* untuk pengujian enkripsi dan dekripsi

Kriptografi adalah ilmu dan seni untuk menjaga keamanan pesan.

Dalam pengujian, sudah dipastikan hasil enkripsi dan dekripsi sudah benar. Hasil pengukuran waktu eksekusi baik pada proses enkripsi maupun dekripsi.

Tabel 4.2. Pengukuran waktu eksekusi dan dekripsi

Proses	Waktu Eksekusi (Second)
Enkripsi	0.06183171272277832
Dekripsi	0.15259099006652832

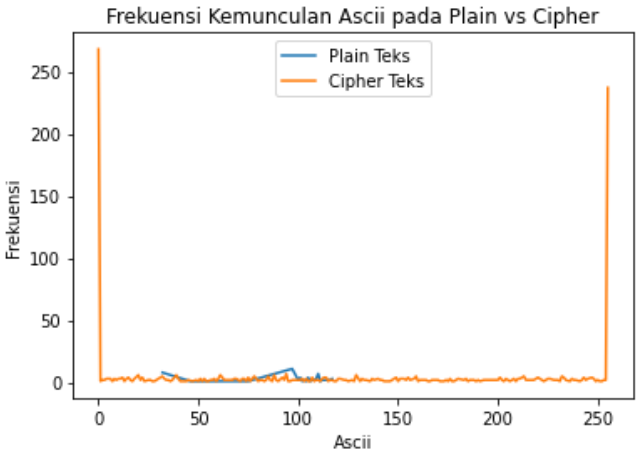
Berdasarkan tabel 4.2 yang menampilkan perbandingan ukuran waktu eksekusi pada saat melakukan enkripsi dan juga dekripsi, dapat disimpulkan bahwa proses dekripsi jauh lebih lama dibandingkan proses enkripsi, dimana pada contoh eksperimen ini, waktu eksekusi dekripsi pesan lebih dari 2 kali waktu eksekusi enkripsi pesan.

Tabel 4.3. Hasil pengukuran ukuran pesan awal, setelah enkripsi, dan setelah dekripsi

Ukuran Pesan (Bytes)		
Pesan Awal	Setelah Enkripsi	Setelah Dekripsi
62	1006	62

Dari hasil pada tabel 4.3, dapat dilihat bahwa ukuran pesan hasil enkripsi (cipherteks) cukup besar, yaitu lebih besar 62 kali dibanding pesan asli. Diperoleh juga kesimpulan bahwa ukuran pesan akhir setelah dekripsi sama dengan ukuran pesan awal, dimana berarti tidak ada padding yang ditambahkan pada hasil akhir dekripsi.

Selain membandingkan waktu eksekusi proses enkripsi dan dekripsi serta membandingkan ukuran file awal, hasil enkripsi dan hasil dekripsi, akan dibandingkan juga perbandingan kemunculan karakter ASCII pada pesan dengan cipherteks.



Gambar 4.1. Perbandingan frekuensi kemunculan Ascii pada Plainteks dan Cipherteks

Berdasarkan gambar 4.1, dapat diketahui bahwa persebaran Ascii cipherteks cukup merata kecuali pada karakter dengan orde 0 dan orde 255, hal itu disebabkan karena padding yang ditambahkan pada proses enkripsi adalah karakter pada orde 0 dan 255.

B. Analisis Keamanan

Seperti yang sudah dijelaskan sebelumnya pada Bab II mengenai keamanannya, algoritma *NTRUEncrypt* memiliki beberapa parameter yang sudah diteliti oleh pembuat algoritma ini, yaitu *Moderate Security, High Security,* dan *Highest Security* dengan nilai parameternya masing-masing. Berikut merupakan analisis keamanan dari masing-masing parameter keamanan tersebut:

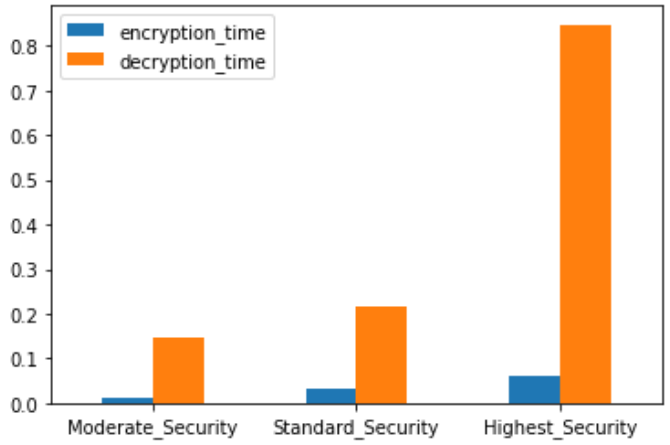
1. *Moderate Security* (N=107, q=64, p=3)
   Dengan parameter ini, maka akan diperoleh panjang kunci privat sepanjang 340 bit dan kunci public sepanjang 642 bit. Level keamanan dari parameter ini berdasarkan serangan *mid-in-the-middle* adalah keamanan kunci sebesar 2^{50} dan keamanan pesan sebesar $2^{26.5}$.
2. *High Security* (N=167, q=128, p=3)
   Dengan parameter ini, maka akan diperoleh panjang kunci privat sepanjang 530 bit dan kunci public sepanjang 1169 bit. Level keamanan dari parameter ini berdasarkan serangan *mid-in-the-middle* adalah keamanan kunci sebesar $2^{82.9}$ dan keamanan pesan sebesar $2^{77.5}$.
3. *Highest Security* (N=503, q=256, p=3)
   Dengan parameter ini, maka akan diperoleh panjang kunci privat sepanjang 1595 bit dan kunci public sepanjang 4024 bit. Level keamanan dari parameter ini berdasarkan serangan *mid-in-the-middle* adalah keamanan kunci sebesar 2^{285} dan keamanan pesan sebesar 2^{170}.

Berdasarkan penelitian, algoritma *NTRUEncrypt* dengan parameter yang sudah disebutkan sebelumnya dianggap sudah aman untuk sebagian besar serangan. Di Sebagian besar aplikasi komersial yang menggunakan algoritma ini, menggunakan parameter N = 251, q = 128, dan p = 3, dan

dianggap sudah aman untuk menghindari serangan berupa *lattice attacks, brute force attacks,* dan *meet-in-the-middle attacks*.

C. Analisis Kinerja

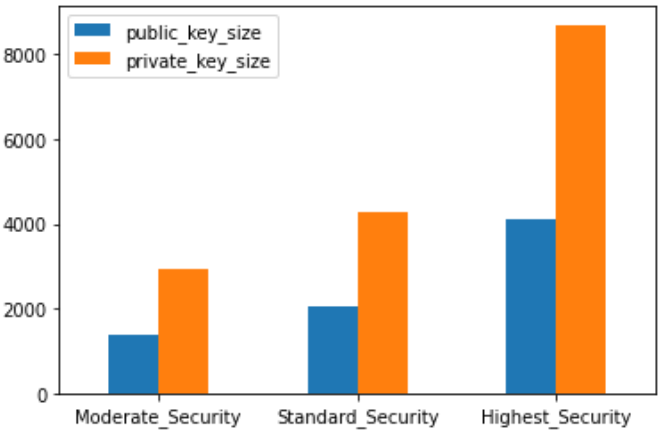
Akan dibandingkan kinerja dari enkripsi dan dekripsi yang dilakukan dengan algoritma *NTRUEncrypt* dengan menggunakan tingkat keamanan yang berbeda-beda yaitu *Moderate Security, High Security,* dan *Highest Security.*



Gambar 4.2. Perbandingan kecepatan enkripsi dan dekripsi dalam detik (second) pada beberapa tingkat keamanan

Dari gambar 4.2, dapat diketahui bahwa waktu yang dibutuhkan untuk melakukan proses dekripsi jauh lebih lama dibanding proses enkripsi. Perbandingan lama waktu melakukan enkripsi dengan tingkat keamanan yang berbeda tidaklah signifikan, sedangkan lama waktu melakukan dekripsi dengan tingkat keamanan yang berbeda cukup jauh berbeda, terutama pada tingkat keamanan *Highest Security* dimana waktu melakukan dekripsi sangat jauh berbeda dibanding tingkat keamanan lain. Kecepatan proses enkripsi dengan tingkat keamanan yang berbeda bertambah secara linear sesuai dengan bertambahnya tingkat keamanan, sedangkan untuk proses dekripsi, pertambahan waktu bertambah secara kuadratik.

Selain membandingkan kecepatan enkripsi dan dekripsi pada beberapa tingkat keamanan yang berbeda, akan dibandingkan juga ukuran kunci publik dan kunci privat yang dihasilkan dari tingkat keamanan yang berbeda-beda yaitu *Moderate Security, High Security,* dan *Highest Security.*



Gambar 4.3. Perbandingan ukuran dalam byte kunci publik dan kunci private pada beberapa tingkat keamanan

Seperti yang terlihat pada gambar 4.3, pertambahan ukuran kunci publik dan kunci privat pada *NTRUEncrypt* dengan pertambahan tingkat keamanan bertambah secara linear, dimana ukuran kunci private berukuran sekitar dua kali ukuran kunci publik.

V. Keunggulan NTRUEncrypt

Berdasarkan beberapa penelitian terkait dan eksperimen yang telah dilakukan algoritma *NTRUEncrypt* memiliki beberapa kelebihan dibanding algoritma kriptografi lainnya, yaitu:

1. Enkripsi dan dekripsi yang lebih efisien, baik dalam implementasi perangkat keras maupun perangkat lunak.
2. Pembuatan kunci yang jauh lebih cepat memungkinkan menggunakan kunci "sekali pakai" (karena secara komputasi kuncinya "murah" untuk dibuat).
3. Penggunaan memori yang rendah memungkinkannya digunakan dalam aplikasi seperti perangkat seluler dan *smart-card*.

VI. Kesimpulan dan Saran

Algoritma *NTRUEncrypt* berhasil diimplementasi dengan baik dan dapat berjalan dengan semestinya. *NTRUEncrypt Cryptosystem* cocok digunakan

Untuk pengembangan selanjutnya, mungkin dapat diimplementasikan *NTRUSign* yang dibuat berdasarkan *NTRU* untuk tanda tangan digital (*digital signature*). Selain itu dapat juga dicari dan dianalisis parameter-parameter lain yang dapat digunakan untuk meningkatkan tingkat keamanan dari *NTRUEncrypt* sehingga akan lebih banyak pilihan parameter yang dapat digunakan sesuai dengan kebutuhan.

VII. Acknowledgment

Puji syukur ke hadirat Tuhan Yang Maha Esa yang telah memberikan kesempatan kepada penulis untuk menyelesaikan makalah ini dengan tepat waktu. Penulis berterima kasih kepada dosen pengampu mata kuliah IF4020 Kriptografi, bapak Dr. Ir. Rinaldi Munir, M.T., yang telah memberikan wawasan dan pengetahuan mengenai kriptografi sehingga implementasi dari algoritma *NTRUEncrypt Cryptosystem* ini dapat dibuat. Penulis juga berterima kasih kepada semua pihak yang turut membantu dalam menulis makalah ini. Dengan makalah ini, penulis berharap dapat membantu pengembangan wawasan pembaca.

Referensi

[1] Munir, R. 2020, "Pengantar Kriptografi (2020)". Slide tersedia di: http://informatika.stei.itb.ac.id/~rinaldi.munir/Kriptografi/2020-2021/Pengantar-Kriptografi-(2020).pdf.

[2] Munir, R. 2020, "Kriptografi kunci-publik". Slide tersedia di:http://informatika.stei.itb.ac.id/~rinaldi.munir/Kriptografi/2020-2021/Kriptografi-Kunci-Publik-2020.pdf.

[3] Hoffstein, J., Pipher, J., & Silverman, J. H. (1998, June). NTRU: A ring-based public key cryptosystem. In *International Algorithmic Number Theory Symposium* (pp. 267-288). Springer, Berlin, Heidelberg.

[4] The Essence of NTRU: Key generation, Encryption, Decryption. https://medium.com/tixlorg/the-essence-of-ntru-key-generation-encryption-decryption-7c0540ef8441. Diakses tanggal 20 Desember 2020.

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, 21 Desember 2020

Josep Andre Ginting
13517108