Implementasi *Digital Signature* dan *Message Encryption* pada *Chatroom*

Afrizal Sebastian - 13520120
Program Studi Teknik Informatika
Sekolah Teknik Elektro dan Informatika
Institut Teknologi Bandung, Jalan Ganesha 10 Bandung
E-mail (gmail): 13520120@std.stei.itb.ac.id

***Abstract*—Saat ini, keamanan dalam pertukaran pesan merupakan salah satu *concern* dari pengguna *chatroom*. Pengguna tidak tahu dengan hal yang terjadi pada saat pengiriman pesan. Untuk mengatasi permasalahan tersebut, perlu dibuat suatu pengamanan pesan dan otentikasi pesan yang menjaga *integrity* pesan. Dalam makalah ini, akan dilakukan penerapan *message encryption* dan *digital signature* pada pesan untuk memastikan keamanan pesan tetap terjaga. *Message encryption* yang digunakan adalah AES dan penggunaan *ECDSA* serta SHA-3 pada implementasi *digital signature*.**

Keywords—ECDSA, Digital Siganture, AES, enkripsi, dekripsi, SHA-3, Keccak

I. PENDAHULUAN

Di era saat ini, informasi yang ada sangat mudah mengalir dari satu orang ke orang lainnya. Informasi tersebut dapat mengandung infromasi yang penting maupun tidak. Orang-orang ingin agar informasi yang mereka kirim dan terima masih tetap terjaga integritasnya. Informasi yang ada dapat mengalir dari transaksi, penggunaan media sosial, dll. Informasi juga dapat mengalir dari media *chatroom*. *Chatroom* merupakan platform dua atau lebih pengguna pada suatu "ruangan" untuk salin bertukar pesan dan informasi.

Chatroom menjadi wadah untuk beberapa orang saling bertukar informasi pribadi maupun umum. Pada saat ini, penggunaan *chatroom* sudah sangat sering digunakan oleh orang-orang untuk bertukar pesan dan informasi yang bisa bersifat pribadi maupun tidak. Pengguna tentutnya tidak ingin adanya kebocoran informasi pada saat penggunaan *chatroom* tersebut.

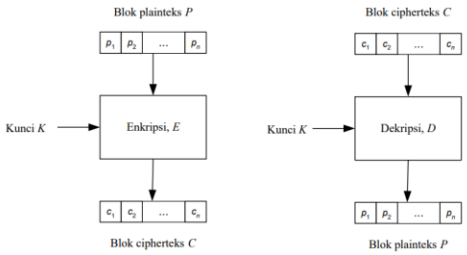
Pesan yang saling bertukar pada *chatroom* akan sangat berbahaya jika terjadi perubahan pada pesan maupun kebocoran pesan penting. Oleh karena itu, perlu dilakukan mekanisme enkripsi pesan pada saat pengiriman agar pesan tidak dapat dimengerti oleh seseorang yang berada pada jaringan. Jika pesan yang dikirimkan terjadi perubahan, maka dengan mekanisme *digital signature* dapat melakukan otentikasi terhadap kebenaran dari pesan serta pengirimnya. Dengan dua mekanisme tersebut pesan yang terkirim dapat terjamin kebenarannya dan akan sulit untuk dibaca maupun diubah oleh penyerang.

Pada makalah ini, akan dibahas dan diimplementasikan penggunana enkripsi-dekripsi pesan dan tanda tangan digital pada suatu *chatroom* untuk beberapa *client*. Dengan beberapa asumsi yaitu hanya jaringan yang bersifat tidak aman, sedangkan *server* dan *client* bersifat aman.

II. DASAR TEORI

A. *Cipher Blok*

Cipher blok akan membagi plainteks yang ada menjadi blok-blok but dengn panjang yang sama dengan panjang yang umunya 64 bit, 128 bit, 256 bit, dsb. Enkripsi akan dilakukan pada setiap blok plainteks dengan menggunakan bit-bit kunci.



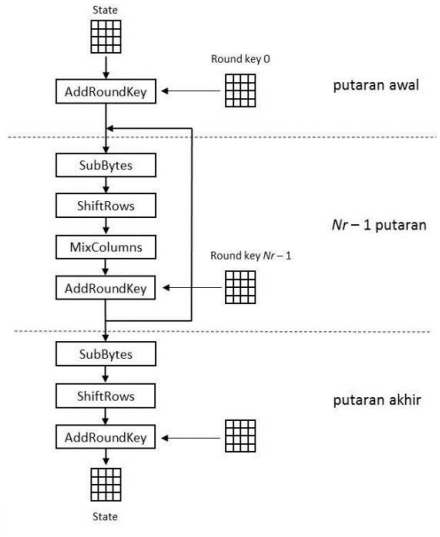
Gambar 1. Skema umum *cipher* blok

Terdapat lima mode operasi pada operasi *cipher* blok, yaitu *Electronic Code Book (ECB), Cipher Block Chaining (CBC), Cipher Feedback (CFB), Output Feedback (OFB),* dan *Counter Mode*

B. *Advanced Encryption Standard (AES)*

AES merupakan kriptograsi simetri yang berbasis *cipher* blok dengan menggunakan algoritma *Rijndael*. Algoritma ini mendukung kunci 128 bit sampai dengan 256 bit dengan step 32 bit tiap kenaikan panjang kunci.

Algoritma *Rijndael* beroperasi dalam orientasi *byte* dan tidak menggunakn jaringan Feistel. *Enciphering* dilakukan dalam sejumlah putaran. Setiap putaran mengunakan kunci internal yang berbeda.



Gambar 2. Skema AES (alogritma *Rijndael*)

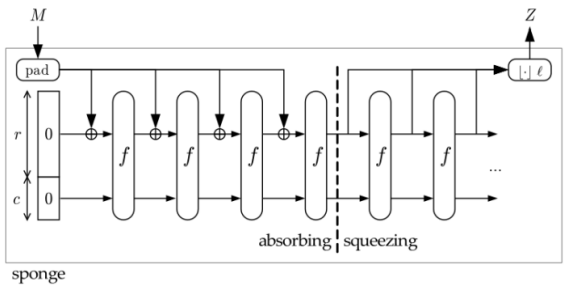
C. *Fungsi Hash*

Fungsi hash merupakan suatu fungsi yang mengkompresi pesan dengan berukuran sembara menjadi sebuat string dengan ukuran yang *fixed* atau tetap. Fungsi hash bersifat satu arah, tidak dapat dikembalikan mejadi pesan semula (*irreversible*). Sifat yang terdapat pada fungsi hash, yaitu *collision resistance, preimage resistance,* dan *second preimage resistance*

D. *SHA-3*

SHA-3 merupkan salah satu fungsi hash yang menggunakan algoritma *Keccak*. Terdapat 3 fase pada algoritma ini, yaitu :

1. Praproses, menyesuaikan panjang pesan dengan panjang blok yang akan digunakan dengan menambahkan *padding*
2. Fase Penyerapan, setiap blok akan di-XOR-kan dengan panjang yang sama pada *state* S lalu masuk ke dalam fungsi permutasi untuk membentuk *state* S berikutnya.
3. Fase Pemerasan, mengambil setiap panjang blok dari *state* S hingga panjang *digest* yang diinginkan.



Gambar 3. Skema SHA-3 (algoritma *Keccak*)

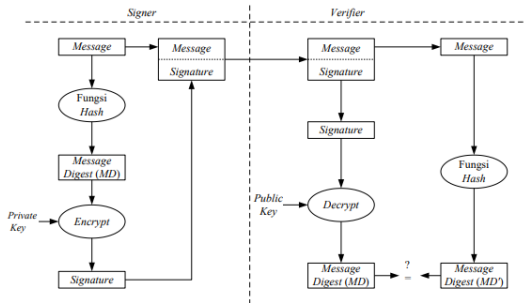
E. *Tanda Tangan Digital*

Tanda tangan digital adalah nilai kriptografis yang bergantung pada isi pesan dan kunci. Tanda tangan digital seseorang akan selalu berbeda pada setiap pesan yang berbeda.

Terdapat dua proses pada tandan tangan digital, ya itu :

1. Memberi tanda tangan pada pesan (*signing*)
2. Memeriksa keabsahan tanda tangan digital (*verification*)

Pesan akan dienkripsi dengan kunci privat si pengirim dan didekripsi dengan kunci public si pengirim. Dengan car aini, maka kerahasiaan pesan dan otentikasi si pengrim pesan dapat dicapai. Jadi, pesan yang dienkripsi oleh kunci privat berarti menandatangai pesan dan pesan yang didekripsi dengan kunci public berarti memverifikasi tanda tangan tersebut.



Gambar 4. Skema umum proses *digital siganture*

F. *ECDSA*

ECDSA merupakan suatu algoritma yang digunakan untuk melakukan penandatanganan suatu pesan digital. ECDSA merupakan pengembangan dari algoritma DSA dalam melakukan tanda tangan digital. Algoritma ini menggunakan *elliptic curve* pada suatu medan berhingga F_p.

Terdapat tiga proses pada algoritma ini yaitu :

1. Pembentukan *key*, pada proses ini akan dibentuk pasanagan kunci privat dan kunci publik.
2. Penandatangan pesan, pada proses ini pesan akan ditandatangani dengan menggunakan kunci privat yang terbentuk sebelumnya.
3. Verifikasi tanda tangan digital, pada proses ini tanda tangan digital yang ada pada pesan akan diverifikasi dengan menggunakan kunci public yang berpasangan dengan kunci privat sebelumnya.

III. IMPLEMENTASI

Implementasi *digital signature* dan *message encryption* menggunakan bahasa *python*. Program dibuat dengan berbasis GUI dengan library *Tkinter*. Enkripsi pesan yang digunakan menggunakan library *Crypto* pada *python* yaitu AES.

A. *Implementasi SHA-3*

Berikut ini adalah fungsi-fungsi yang digunakan untuk menimplementasikan SHA-3.

```
def Keccak(message : str, ):
    ⇨ Fungsi yang mengembalikan message digest dari message masukan yang ada.
def KeccakPermutation(A : List[List[int]], b:int):
    ⇨ Fungsi yang digunakan untuk melakukan permutasi sebanyak b putaran dan mengembalikan matriks hasil permutasi
def KeccakRound(A : List[List[int]], RCnow, b:int):
    ⇨ Fungsi yang digunakan untuk melakukan permutasi pada satu putaran.
def rot(x, y):
    ⇨ Fungsi yang digunakan untuk melakukan rotasi pada setiap putarannya.
```

B. Implementasi ECDSA

Berikut ini adalah *method-method* pada kelas ECDSA yang digunakan untuk menimplementasikan ECDSA.

```
class ECDSA:
    ⇨ Kelas yang digunakan untuk mengimplementasikan ECDSA
def __init__(self):
    ⇨ Method constructor untuk kelas ECDSA.
def gcd(self, a, b):
    ⇨ Method untuk mendapatkan greatest common divisor dari a dan b.
def modulus(self, a, b):
    ⇨ Method yang digunakan untuk mendapatkan nilai modulus dari a dan b dengan memperhatikan bilangan negative.
def invModulus(self, a, b):
    ⇨ Method yang digunakan untuk mendapatkan nilai invers modulus dari a dan b.
def pointAdd(self, P, Q, a, d, mod):
    ⇨ Method yang digunakan untuk melakukan penjumlahan titik P dan Q pada kurva dengan parameter a dan d pada modulus mod.
def pointMultiply(self, P, k, a, d, mod):
    ⇨ Method yang digunakan untuk melakukan perkalian titik P dan Q pada kurva dengan parameter a dan d pada modulus mod.
def generateKeyPair(self):
    ⇨ Method yang digunakan untuk mengasilkan pasangan private key dan public key.
def getPublicKey(self, privateKey):
    ⇨ Method yang digunakan untuk mendapatkan nilai public key dari private key yang ada.
def hashing(self, message):
    ⇨ Method yang digunakan untuk melakukan hash pada pesan dengan menggunakan Keccak.
def sign(self, message, privateKey):
    ⇨ Method yang digunakan untuk menandatangani pesan menggunakan private key.
def verify(self, message, publicKey, signature):
    ⇨ Method yang digunakan untuk melakukan verifikasi digital signature.
```

C. Implementasi Server

Berikut ini adalah fungsi-fungsi yang digunakan untuk menimplementasikan server.

```
def pad_data(data:bytes):
    ⇨ Melakukan padding pada pesan untuk melakukan enkripsi.
def broadcast(message, publicKey) :
    ⇨ Melakukan broadcast atau mengirim pesan ke semua client yang terhubung ke server
def handle(client):
    ⇨ Menerima pesan dari suatu client dan melakukan broadcast terhadap pesan tersebut.
def receive():
    ⇨ Menerima client yang akan bergabung dan melakukan beberapa setup untuk client dan server.
```

D. Client

Berikut ini adalah *method-method* pada kelas *Client* yang digunakan untuk menimplementasikan client.

```
class Client :
    ⇨ Kelas yang digunakan untuk mengimplementasikan Client.
def __init__(self, host, port):
    ⇨ Method constructor untuk kelas Client. Melakukan setup untuk socket, nickname, dan private key-public key.
def generateKeyPair(self):
    ⇨ Method yang digunakan untuk mendapatkan pasangan private dan public key dengan menggunakan ECDSA
def gui_loop(self) :
    ⇨ Method yang digunakan untuk membentuk GUI untuk chatroom.
def sign(self, messages : str, privateKey : str):
    ⇨ Method yang digunakan untuk melakukan sign pada pesan yang akan dikirim menggunakan
```

private key.

`def pad_data(self, data:bytes):`

⇨ Method yang digunakan untuk melakukan *padding* pada pesan untuk melakukan enkripsi.

`def write(self):`

⇨ Method yang digunakan untuk mengirim pesan dengan melakukan tanda tangan dan enkripsi terlebih dahulu.

`def stop(self):`

⇨ Method yang digunakan untuk memberhentikan *client*.

`def verify(self, signed_message : str, publicKey1 : str, publicKey2 : str):`

⇨ Method yang digunakan untuk melakukan verifikasi terhadap tanda tangan yang ada.

`def unpad_data(self, data : bytes):`

⇨ Method yang digunakan untuk membuang *padding* pada pesan.

`def countSetup(self):`

⇨ Method yang digunakan untuk melakukan pemantauan *setup* pada *client*.

`def recieve(self):`

⇨ Method yang digunakan untuk menerima pesan dari *server* dan menampilkannya pada GUI dengan terlebih dahulu melakukan dekripsi dan verifikasi pada pesan.

E. Alur Program

Pertama *server* akan berjalan terlebih dahulu pada IP dan *port* yang akan digunakan. Server akan membentuk *key* yang akan digunakan untuk melakukan enkripsi dan dekripsi pesan. Setelah itu, *client* dapat terhubung ke *server*. Saat *client* terhubung ke *server*, *server* akan meminta *public key* untuk penandatangan pesan dan *nickname* serta memberikan kunci yang akan digunakan untuk enkripsi-dekripsi pesan ke *client*.

Pada saat *client* ingin melakukan pengiriman pesan. Pesan terlebih dahulu ditandatangani dengan *private key* milik *client* lalu melakukan enkripsi dengan *key* yang diterima dari *server*. Lalu *server* akan menerima pesan lalu meneruskan ke setiap *client* yang terhubung ke *server*. Saat pesan diterima oleh *client*, *pesan* akan terlebih dahulu didekripsi lalu melakukan verifikasi pada tanda tangan. Jika terbukti keabsahan pesan, maka pesan akan ditampilkan pada kolom *chatroom*. Jika sebaliknya, maka pesan tidak akan ditampilkan dan memberikan peringatan ke *client* bahwa pesan tidak sah.

IV. EKSPERIMEN DAN ANALISIS

Berikut ini adalah hasil pengujian terhadap hasil implementasi yang dilakukan pada *chatroom*.

A. Eksperimen

Terdapat 3 pengujian yang akan dilakukan terhadap hasil implementasi, yaitu *:*

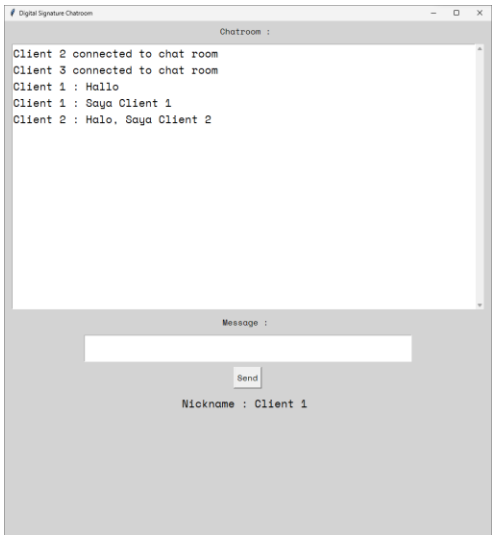
1. Pengujian pada pesan dan tanda tangan yang valid.
2. Pengujian terhadap pesan yang terubah.
3. Pengujian dengan menggunakan wireshark.

Hasil dari pengujian tersebut dilakuakan untuk mengetahui kebenaran tanda tangan digital dan hasil enkripsi pesan pada jaringan yang ditangkap menggunakan wireshark.

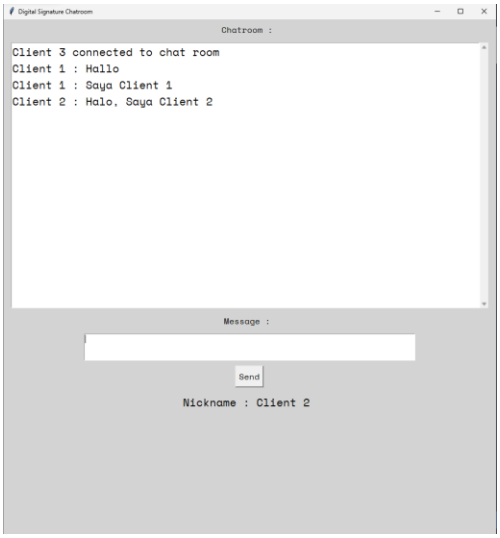
1. Pengujian pada pesan dan tanda tangan yang valid.

   Keluaran yang diharapkan adalah pesan dapat ditampilkan pada kolom *chatroom*.

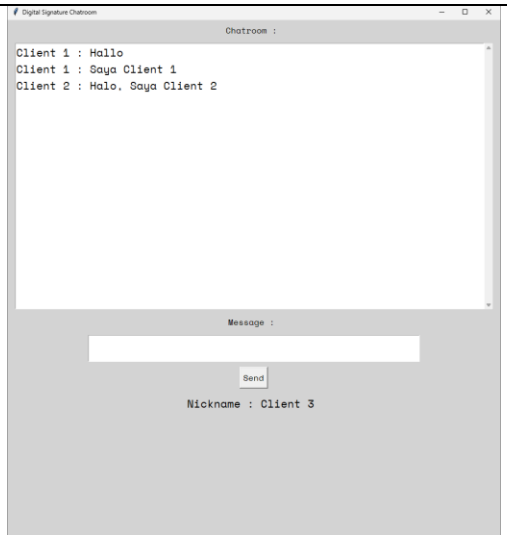
Client 1



Client 2



Client 3



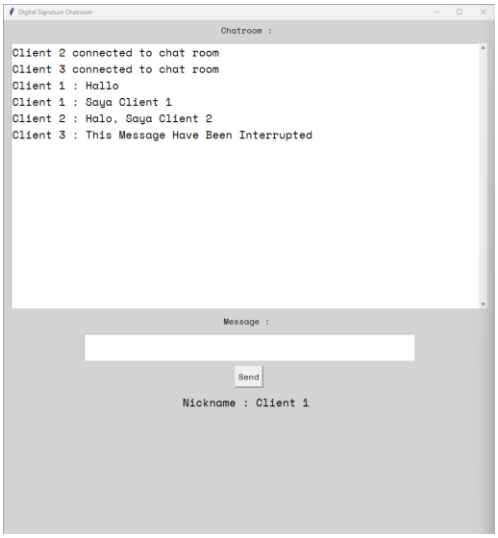
Hasil yang didapatkan adalah pesan yang dikirim oleh *client 1* dan *client 2* dapat ditampilkan pada kolom *chatroom* setiap client yang terhubung ke *server*.

2. Pengujian terhadap perubahan pesan

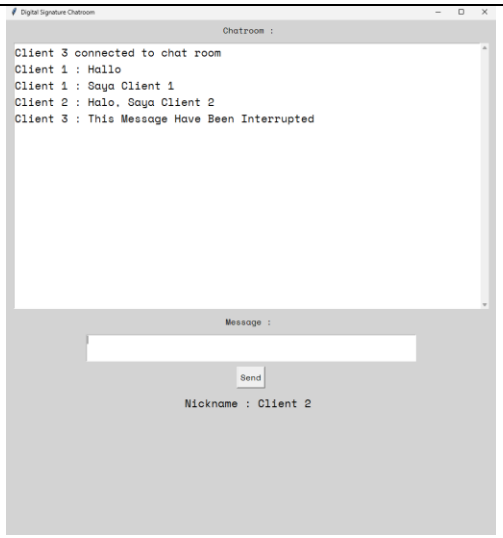
   Keluaran yang diharapkan adalah pesan yang dikirimkan tidka ditampilkan pada kolom *chatroom* dan memberikan pesan peringatan kepada setiap *client* bahwa pesan telah berubah.

   Pesan yang dikirimkan adalah "Halo Saya Client 3" dan berubah menjadi "Halo Saya Client 3 berubah"

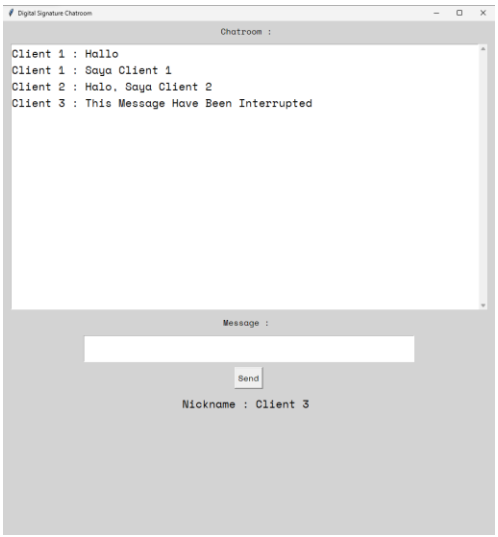
Client 1



Client 2



Client 3

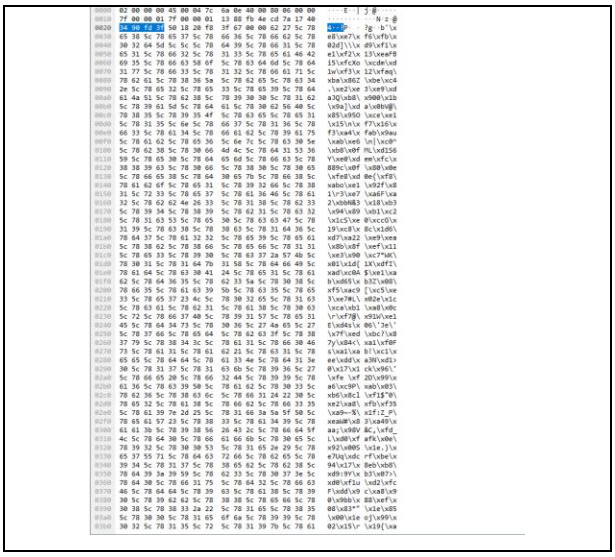


Hasil yang didapatkan adalah pesan yang tidak terkirim oleh *client 3* tidak dapat ditampilkan pada kolom *chatroom*. Pesan diganti dengan peringatan bahwa pesan telah berubah.

3. Pengujian dengan wireshark

   Keluaran yang diharapkan adalah pesan tidak dapat dibaca pada saat terkirim di jaringan.

   Pesan yang dikirim "Halo Halo Ini Client 1"



Hasil yang didapatkan pesan tidak dapat dimengerti maknya karena telah dienkripsi pada saat sebelum melakukan pengiriman.

B. *Analisis*

Dari hasi pengujian tersebut, hasil yang diinginkan dan hasil yan didapatkan sesuai. Pengujian 1 dan 2 dilakukan untuk melakukan verifikasi terhadap penggunaan *digital signature* dan hasil yang didapatkan untuk pengujian 1 adalah pesan ditampilkan ke kolom *chatroom* sedangkan untuk pengujian 2 digunakan untuk menampilkan pesan peringatan bahwa pesan telah berubah. Pengujian 3 dilakukan untuk melihat *package* yang terkirim dengan menggunakan *wireshark* untuk memastikan pesan terenkripsi. Sehingga, *digital signature* dan *message encryption* dapat diimplementasikan dengan baik pada makalah ini.

V. KESIMPULAN DAN SARAN

Digital signature dapat digunakan untuk mengatasi integritas pesan yang dikirim maupun yang akan diterima pada aplikasi *chatroom*. Dengan adanya *digital signature* pesan dapat diverifikasi dan terjada integritasnya. *Message encryption* juga dapat digunakan sebagai salah satu mekanisme agar pesan yang terkirim akan sulit dibaca dan dimengerti pada saat berada pada jaringan, sehingga dapat menjaga integritas pesan.

Saran untuk melakukan pengembangan program ini adalah dapat mengganti *message encryption* dengan menggunakan algorimta kunci publik, seperti RSA, ElGamal, dsb.

SOURCE CODE

Berikut ini adalah repository hasil implementasi *digital signature* dan *message encryption* pada *chatroom*.

https://github.com/afrizalsebastian/Digital-Signature-pada-Chat-Room-berbasis-GUI---Kriptografi

UCAPAN TERIMA KASIH

Puji syukur penulis kepada Tuhan Yang Maha Esa atas berkat dan rahmat-Nya makalah "Implementasi *Digital Signature* dan *Message Encryption* pada *Chatroom*" dapat selesai dengan baik dan tepat waktu.

Penulis mengucapkan terima kasih kepada Bapak Dr. Ir. Rinaldi Munir, M.T sebagai dosen pengajar Mata Kuliah Kriptografi atas bimbingan dan pengajaran terkait kuliah ini. Penulis juga mengucapkan terima kasih kepada Bapak Dr. Ir. Rinaldi Munir, M.T. yang telah menyedikan materi kuliah Kriptografi dan website yang membantu pengerjaan makalah ini. Penulias juga mengucapkan terima kasih kepada orangtua, keluarga, teman, serta semua orang yang membantu dalam pengerjaan makalah ini.

REFERENSI

[1] Munir, Rinaldi "Block Cipher". https://informatika.stei.itb.ac.id/~rinaldi.munir/Kriptografi/2022-2023/13-Block-Cipher-Bagian1-2023.pdf. Diakses pada 19 Mei 2023

[2] Munir, Rinaldi "Review Beberapa Block Cipher" https://informatika.stei.itb.ac.id/~rinaldi.munir/Kriptografi/2022-2023/17-Beberapa-block-cipher-bagian3-2023.pdf. Diakses pada 19 Mei 2023

[3] Munir, Rinaldi "Fungsi Hash" https://informatika.stei.itb.ac.id/~rinaldi.munir/Kriptografi/2022-2023/25-Fungsi-hash-2023.pdf. Diakses pada 20 Mei 2023

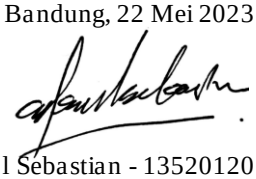
[4] Munir, Rinaldi "Tanda-tangan Digital" https://informatika.stei.itb.ac.id/~rinaldi.munir/Kriptografi/2022-2023/28-Tanda-tangan-digital-2023.pdf. Diakses pada 20 Mei 2023

[5] Triwinarko, Andy "Elliptic Curve Digital Signature Algorithm (ECDSA) Departemen Teknik Informatika ITB" https://informatika.stei.itb.ac.id/~rinaldi.munir/TA/Makalah_TA%20Andy%20T.pdf. Diakses pada 20 Mei 2023

UCAPAN TERIMA KASIH

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, 22 Mei 2023



Afrizal Sebastian - 13520120