Jurnal Riset Sistem Informasi Dan Teknik Informatika (JURASIK)
Volume 8, Nomor 1, Februari 2023, pp 142-154
Terakreditasi Nomor 204/E/KPT/2022 | ISSN: 2527-5771/EISSN: 2549-7839
https://tunasbangsa.ac.id/ejurnal/index.php/jurasik

Analisis Enkripsi Kriptografi Asimetris Algoritma RSA Berbasis Pemrograman *Batch* pada Media *Flashdisk*

Nathanael Berliano Novanka Putra[1], Fikra Amalia Raihana[2], Willem Michael Albert Mondong[3*], Aqwam Rosadi Kardian[4]
[1,2,3,4]Politeknik Siber dan Sandi Negara, Jawa Barat, Indonesia
*Email: [1]nael.ano101@gmail.com, [2]fikaraihana15@gmail.com, [*3]michaelmondong6@gmail.com, [4]aqwam@staff.jak-stik.ac.id*

Abstract

Flashdisk is an external data storage device that is connected to a USB port. Data security is necessary considering that flash drives can be accessed by anyone physically. Therefore, we need a program or application that can act as a secure intermediary for document transfer encryption where users can use flash drives not only to exchange information but also to secure that information. Asymmetric Cryptography is a type of cryptography that utilizes 2 types of keys, namely public keys and private keys. In this study, the RSA Algorithm was chosen to perform encryption because of its better level of security than other asymmetric algorithms, and it can still keep up with technological developments, especially in the security aspect. In this research, an experiment will be carried out to apply asymmetric cryptography using the RSA algorithm in the OpenSSL program to encrypt documents on flash media. The results of the study stated that asymmetric cryptography with the RSA algorithm could perform good encryption on documents contained in flash drives that had used the caraka.bat program.

Keywords*: flash, encryption, cryptography, batch, RSA*

Abstrak

Flashdisk adalah sebuah alat penyimpanan data eksternal yang dihubungkan pada port USB. Pengamanan data diperlukan mengingat bahwa flashdisk dapat diakses oleh siapa pun secara fisik. Oleh karena itu, diperlukan suatu program atau aplikasi yang dapat berperan sebagai perantara yang aman untuk enkripsi transfer dokumen dimana pengguna dapat menggunakan flashdisk bukan hanya untuk bertukar informasi tapi juga mengamankan informasi tersebut. Kriptografi Asimetris adalah jenis kriptografi yang memanfaatkan 2 jenis kunci yaitu public key dan private key. Dalam penelitian ini, Algoritma RSA dipilih untuk melakukan enkripsi karena tingkat keamanannya yang lebih baik dibanding algoritma asimetris lainnya, serta masih dapat mengimbangi perkembangan teknologi khususnya dalam aspek keamanan. Pada penelitian ini akan dilakukan eksperimen untuk menerapkan kriptografi asimetris menggunakan algoritma RSA pada program OpenSSL untuk enkripsi dokumen pada media flashdisk. Hasil penelitian menyatakan bahwa kriptografi asimetris dengan algoritma RSA dapat melakukan enkripsi dengan baik pada dokumen yang terdapat di flashdisk yang telah menggunakan program caraka.bat

Kata kunci*: flashdisk, enkripsi, kriptografi, batch, RSA*

1. PENDAHULUAN

Komputer merupakan alat yang sangat dibutuhkan oleh banyak instansi dan perusahaan-perusahaan milik negara maupun swasta untuk dapat bekerja [1]. Salah satu faktor terpenting dalam pekerjaan adalah pertukaran informasi. Dalam pertukaran informasi tersebut dibutuhkan jalur yang aman untuk dapat melakukan komunikasi dengan lebih baik. Media yang pada umumnya digunakan perusahaan untuk melakukan pertukaran data atau informasi tersebut adalah

dengan menggunakan USB Flash Drive atau flashdisk. USB flash drive atau juga sering dikenal sebagai flashdisk merupakan teknologi penyimpanan berbentuk sebuah stik[2]. Dengan menggunakan teknologi flashdisk, pengguna dapat melakukan pertukaran data dan informasi dengan lebih praktis. Namun dibalik praktisnya penggunaan flashdisk, ada berbagai ancaman kebocoran data maupun informasi yang dapat terjadi, sehingga untuk mencegah dan meminimalisir terjadinya hal yang tidak diinginkan, dibutuhkan suatu flashdisk yang memiliki fungsi sebagai media pertukaran informasi sekaligus dapat melakukan pengamanan terhadap data yang akan didistribusikan.

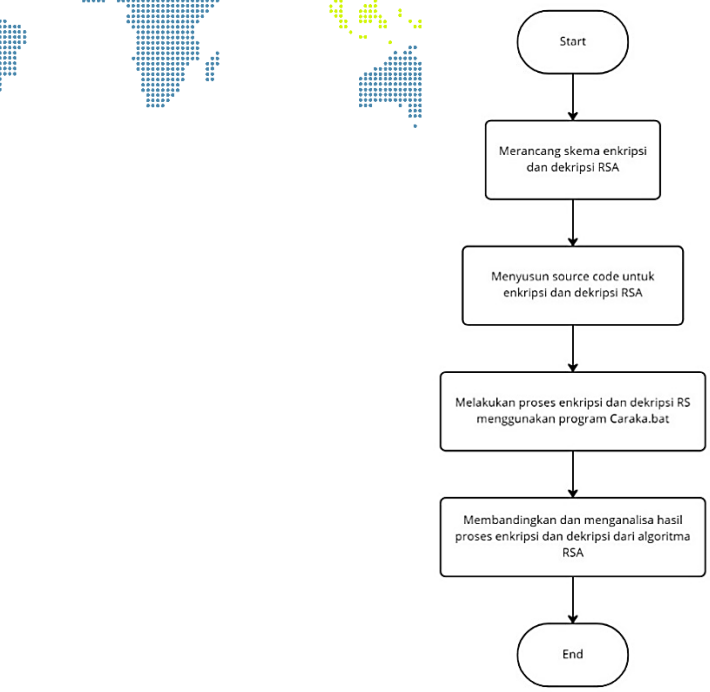
Kriptografi dapat diartikan sebagai suatu ilmu atau seni menjaga keamanan pesan. Dengan dua proses dasar kriptografi berupa enkripsi dan dekripsi. Enkripsi merupakan proses mengolah plaintext (pesan yang bisa dibaca) menjadi ciphertext (pesan acak yang tidak bisa dibaca). Dekripsi adalah kebalikan dari proses enkripsi. Yakni suatu proses mengolah ciphertext menjadi plaintext. Proses ini berlangsung menggunakan kunci yang sama dan algoritma pembalik. Berdasarkan kunci yang digunakan dalam proses enkripsi dan dekripsi, kriptografi dibedakan menjadi kriptografi kunci-simetri (symmetric-key cryptography) yang biasa disebut kriptografi kunci-privat dan kriptografi kunci- nirsimetri (asymmetric key cryptography) yang biasa disebut kriptografi kunci-publik[3]. Pada kriptografi kunci- publik terdapat sepasang kunci, satu kunci untuk proses enkripsi dan satu kunci untuk proses dekripsi. Kunci untuk proses enkripsi bersifat umum atau tidak dirahasiakan (kunci publik), oleh karena itu semua orang dapat menggunakan kunci publik. Sedangkan kunci untuk proses dekripsi bersifat rahasia (kunci rahasia) dan hanya digunakan oleh penerima pesan. Oleh karena itu, kunci enkripsi tidak boleh sama dengan kunci dekripsi. Salah satu algoritma kriptografi kunci-publik yang populer adalah algoritma RSA. RSA adalah singkatan dari ketiga nama penemunya yaitu Rivest, Shamir, dan Adleman. Keamanan algoritma RSA terletak pada sulitnya memfaktorkan bilangan yang besar menjadi faktor-faktor prima. Pemfaktoran dilakukan untuk memperoleh kunci privat. Selama pemfaktoran bilangan besar menjadi faktor-faktor prima belum ditemukan algoritma yang mangkus, maka selama itu pula keamanan algoritma RSA tetap terjamin. Pada penelitian ini, kami menggunakan program OpenSSL yang sudah memiliki library yang berkaitan dengan kriptografi[4].

Penelitian ini bertujuan melakukan analisis terhadap enkripsi transfer dokumen dengan menggunakan media flashdisk dengan metode kriptografi asimetris serta menggunakan algoritma RSA agar sebuah flashdisk dapat melakukan proses enkripsi maupun dekripsi data sehingga tercipta sebuah jalur yang aman untuk dapat melakukan pertukaran data dan informasi.

2. METODOLOGI PENELITIAN

Penelitian ini dimulai dengan tahap perancangan skema enkripsi dan dekripsi RSA dan penyusunan kode program menggunakan pemrograman batch. Berikut disajikan flowchart alur penelitian yang akan digunakan sebagai acuan penelitian ini.

Jurnal Riset Sistem Informasi Dan Teknik Informatika (JURASIK)
Volume 8, Nomor 1, Februari 2023, pp 142-154
Terakreditasi Nomor 204/E/KPT/2022 | ISSN: 2527-5771/EISSN: 2549-7839
https://tunasbangsa.ac.id/ejurnal/index.php/jurasik



Gambar 1. Flowchart Penelitian

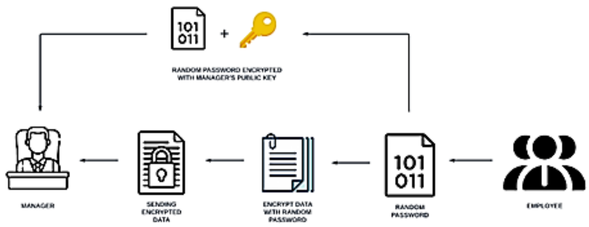
Dalam penelitian ini, skema enkripsi dan dekripsi pada algoritma RSA dimodifikasi dengan menambahkan random password yang berguna untuk menambah faktor keamanan data dan sebagai fungsi keamanan berlapis untuk meningkatkan keamanan dokumen. Random password akan dibuat dengan menggunakan pemrograman batch yang akan dijelaskan lebih lanjut. Selain itu, agar penelitian dapat dilakukan maka PC/Laptop yang akan digunakan harus sudah mempunyai OpenSSL agar program batch dapat berjalan sebagaimana mestinya. OpenSSL adalah alat sumber terbuka untuk menggunakan protokol Secure Socket Layer (SSL) Cryptographic Library.

Skenario pengamanan dokumen dapat diibaratkan sebagai seorang manager dalam sebuah perusahaan sebagai penerima (receiver) dan karyawan-karyawan dalam perusahaan tersebut sebagai pengirim (sender). Manager akan menyebarkan public key kepada karyawan-karyawan di perusahaan. Karyawan akan menerima public key milik manager dan disimpan dalam PC/Laptop. Setiap entitas termasuk manager dan karyawan harus memiliki random password milik sendiri yang sudah dibuat sebelumnya oleh masing-masing entitas.

Sebagai contoh. seorang karyawan berniat untuk mengirimkan sebuah berkas kepada manager maka karyawan akan melakukan enkripsi pada berkas yang ingin dikirimkan dengan random password yang ia buat sebelumnya. Setelah berkas terenkripsi, random password yang digunakan untuk mengenkripsi berkas sebelumnya dienkripsi dengan menggunakan public key yang sebelumnya didapatkan dari manager.

Manager mendapatkan sebuah berkas yang sudah terenkripsi dan kunci yang dienkripsi untuk membuka berkas tersebut. Manager akan menggunakan private

key milik manager untuk membuka kunci yang terenkripsi dengan kunci publik milik manager. Kunci random password pun didapatkan lalu manager akan menggunakannya untuk mendekripsi dokumen terenkripsi yang sebelumnya dikirimkan.



Gambar 2. Alur Pengiriman Data

Tahap berikutnya adalah penyusunan kode program untuk enkripsi dan dekripsi algoritma RSA pada pemrograman batch. Pemrograman Batch adalah pemrograman dasar yang terdapat pada sistem operasi Microsoft Windows yang berisi sekumpulan perintah command line pada Microsoft Windows[7]. Program Batch memiliki ekstensi .bat atau .cmd dan pada umumnya berfungsi sebagai program administrator. Berikut adalah source code dari program batch yang telah dirancang sedemikian rupa:

```
@ECHO OFF
TITLE CARAKA

:MAINMENU
CLS
ECHO
"^!!!!!!!!!!!!!!!!!!!!!!!!!!!!!!!!!!!!!!!!!!!!!!!!!!!!!!!!!!!!!!!!!!!!!!!!!!!!!!!!!
!!!!!!!!!!!!:"
ECHO "
YJ:^^^^^^^^^^^:::::::^^^^^^^^^::^^^^^::::::::^^^^^^^^^^^^^::^^^^^:::::^^^::::^^^^^^^::^
^^^^^^^^^:P7"
ECHO "   Y7              ..:...        :        .  ...                :.       .  .     . ..
:.           Y7"
ECHO "   Y?          :7?7!!!7?5^       7#:      .~GG~!?5P!          ~&~      .^5B~:   .!P7^
~&^            Y7"
ECHO "   Y?         ?G!.         ..      ~?BP        G#     Y@~        ^?GB        5&.   :~!:
^?GB            Y7"
ECHO "   Y?        !@~                ^7 ^@?       P#.   .5B.       :? .&Y       Y&.^!^.
:? .&Y           Y7"
ECHO "   Y?        5&.               :5^^^5@^      P#~!GP~        .5~^^J@!      Y&!PG!
.5~^^J@!         Y7"
ECHO "   Y?        7@J              .J^^^^:P#.     P#  ~BP:       J~^^^:Y&:     Y&. !GG!
J~^^^:Y&:        Y7"
ECHO "   Y?         ?#P!:....^?~ .P~      :&G.    G&.. .5&5^. .57      .B#:    Y@:    7#P^.
.57     .B#:       Y7"
ECHO "   Y?          .~77!~~^^. :!?~^    .^!?!:.^7?~^.  ^?7~:~?~^     ^!?!: ^!?~^
.?7~:~?~^     ^!?!:      Y7"
ECHO "   Y7
Y7"
ECHO "
?Y77777777777777777777777777777777777777777777777777777777777777777777777777777777
7777777775!"
ECHO "
...................................................................................
........  "
ECHO Welcome!
ECHO 1) Generate Key
ECHO 2) Generate Password
ECHO 3) Encrypt Dokumen
ECHO 4) Decrypt Dokumen
```

Jurnal Riset Sistem Informasi Dan Teknik Informatika (JURASIK)
Volume 8, Nomor 1, Februari 2023, pp 142-154
Terakreditasi Nomor 204/E/KPT/2022 | ISSN: 2527-5771/EISSN: 2549-7839
https://tunasbangsa.ac.id/ejurnal/index.php/jurasik

```
set /p menu=(1,2,3,4)? :

IF /i %menu%==1 goto GENERATEKEY
IF /i %menu%==2 goto GENERATEPASSWORD
IF /i %menu%==3 goto ENCRYPT
IF /i %menu%==4 goto DECRYPT

:GENERATEKEY
OPENSSL genrsa -out private.pem 2048
OPENSSL rsa -in private.pem -outform PEM -pubout -out public.pem
goto MAINMENU

:GENERATEPASSWORD
set /p pub=[+]Path of Public Key:
OPENSSL rand -out randompassword -hex 32
OPENSSL rsautl -encrypt -inkey %pub% -pubin -in randompassword -out
randompassword.encrypted
ECHO [V]Randompassword and randompassword.encrypted created!
PAUSE
goto MAINMENU

:ENCRYPT
set /p dokumen=[+]Path of Dokumen:
set /p drive=[+]Destination:
set /p pw=[+]Path of password:
OpenSSL enc -p -aes-256-cbc -salt -in %dokumen% -out secret.enc -pass dokumen:%pw%

COPY Caraka.bat%drive%
COPY randompassword.encrypted %drive%
SLEEP 2
COPY secret.enc %drive%
attrib +h %drive%"\secret.enc"

ECHO [V]Encrypted and moved!
PAUSE
goto MAINMENU

:DECRYPT
set /p private=[+]Path of Private Key:
set /p dokumen=[+]Result dokumenname:

OpenSSL rsautl -decrypt -inkey %private% -in randompassword.encrypted -out
randompassword
attrib -h secret.enc
OpenSSL enc -d -p -aes-256-cbc -salt -in secret.enc -out %dokumen% -pass
dokumen:./randompassword
rm secret.enc
ECHO [V]Decrypted!
PAUSE
goto MAINMENU

PAUSE
```

Kriptografi Asimetris pada program diterapkan dengan menggunakan program OpenSSL karena OpenSSL sudah memiliki library terkait kriptografi. Program terbagi menjadi beberapa bagian yaitu, MAINMENU, GENERATEKEY, GENERATEPASSWORD, ENCRYPT, COPY, dan DECRYPT. Pada menu MAINMENU program akan menampilkan halaman utama dari program yang berisikan pilihan menu yang terdiri dari menu Generate key, Generate Password, Encrypt, dan Decrypt.

2.1. Kriptografi

Kriptografi (*cryptography*) berasal dari bahasa Yunani yang terdiri dari kata kryptos yang artinya tersembunyi dan graphia yang artinya sesuatu yang tertulis sehingga kriptografi dapat juga disebut sebagai sesuatu yang tertulis secara rahasia atau tersembunyi[5]. Kriptografi merupakan ilmu yang menggunakan teknik-teknik matematis yang diimplementasikan pada aspek keamanan informasi, seperti integritas data, kerahasiaan data, dan otentikasi pengirim/penerima data. Kriptografi juga dapat digunakan untuk melakukan identifikasi pengiriman pesan dengan tanda tangan digital dan keaslian pesan dengan menggunakan sidik jari digital.

Ada beberapa istilah yang digunakan dalam kriptografi. Proses untuk mengubah teks asli menjadi teks rahasia dikenal dengan istilah enkripsi. Sebaliknya, proses untuk mengubah kembali teks rahasia menjadi teks asli disebut dekripsi. Pesan yang tid ak terenkripsi disebut sebagai teks terang(plaintext), sedangkan pesan yang telah mengalami proses enkripsi disebut teks sandi(ciphertext). Proses enkripsi dan dekripsi membutuhkan sejumlah informasi rahasia yang disebut sebagai kunci(key).

Kriptografi memiliki empat tujuan dasar yang juga dijabarkan sebagai aspek keamanan informasi, yaitu:

a) kerahasiaan (confidentiality)
   Kerahasiaan mencakup layanan untuk menjaga isi dari informasi agar tidak dapat diakses oleh siapapun selain yang memiliki otoritas untuk mengakses informasi, dalam hal ini adalah orang yang memiliki kunci rahasia.
b) Integritas data
   Integritas mencakup jaminan bahwa data yang diterima merupakan data yang masih asli dan tidak mengalami modifikasi oleh pihak yang tidak sah. integritas data dapat dijaga dengan menggunakan sistem yang mampu mendeteksi manipulasi data oleh pihak-pihak yang tidak berhak, meliputi penghapusan, penyisipan, dan pensubtitusian data lain kedalam data asli.
c) Autentikasi
   Autentikasi adalah aspek yang berhubungan dengan identifikasi/pengenalan. Dua pihak yang saling berinteraksi harus bisa memastikan identitas masing-masing serta keaslian sumber data.
d) Non-repudiasi atau nirpenyangkalan
   Non-repudiasi atau nirpenyangkalan adalah layanan yang mencegah terjadinya penyangkalan terhadap informasi yang dikirimkan/diciptakan oleh pengirim/pembuat informasi [6].

2.2. Kriptografi Asimetris

Kriptografi terbagi menjadi berbagai jenis algoritma. berdasarkan cara kerjanya, kriptografi terbagi menjadi dua jenis, yaitu kriptografi tradisional dan kriptografi modern. Algoritma kriptografi tradisional beroperasi dengan menggunakan mode karakter, sedangkan algoritma kriptografi modern bekerja dengan menggunakan mode bit.

Jurnal Riset Sistem Informasi Dan Teknik Informatika (JURASIK)
Volume 8, Nomor 1, Februari 2023, pp 142-154
Terakreditasi Nomor 204/E/KPT/2022 | ISSN: 2527-5771/EISSN: 2549-7839
https://tunasbangsa.ac.id/ejurnal/index.php/jurasik

Berdasarkan jenis kuncinya, algoritma kriptografi terbagi menjadi dua jenis, yaitu algoritma kriptografi simetris dan algoritma kriptografi asimetris. Pada algoritma kriptografi simetris, kunci yang digunakan untuk proses enkripsi maupun dekripsi adalah kunci yang sama. Beberapa contoh algoritma yang menggunakan konsep kunci simetris adalah DES (Data Encryption Standard), Blowfish, dan AES. Skema enkripsi dan dekripsi algoritma kriptografi simetris dijelaskan pada Gambar 3.

Gambar 3. Skema Enkripsi Kunci Asimetris

Algoritma kriptografi asimetris adalah algoritma yang menggunakan kunci yang berbeda untuk proses enkripsi dan dekripsi. algoritma kriptografi asimetris menggunakan kunci publik (public-key) untuk enkripsi dan kunci privat (private-key) untuk dekripsi. Algoritma ini dikenal juga dengan sebutan kriptografi kunci publik karena kunci yang digunakan untuk enkripsi bersifat umum dan dapat diketahui oleh publik. Kunci privat digunakan untuk dekripsi bersifat rahasia dan hanya dimiliki oleh pihak yang berwenang. Beberapa algoritma yang menerapkan konsep kunci asimetris adalah RSA, Diffie-Hellman, ElGamal, dan Rabin. Skema proses enkripsi dan dekripsi algoritma kriptografi asimetris divisualisasikan pada Gambar 4.

Gambar 4. Skema Dekripsi Kunci Asimetris

2.3. Algoritma Rivest-Shamir-Adleman (RSA)

Algoritma RSA merupakan algoritma yang menggunakan konsep kunci asimetris. RSA ditemukan pada tahun 1977 oleh Rivest, Adi Shamir, dan Len Adleman. RSA menggunakan 2 kunci yang berbeda untuk proses enkripsi dan dekripsi. enkripsi dan dekripsi didasarkan pada konsep bilangan prima dan aritmetika modulo. Kunci publik dan kunci privat yang digunakan merupakan bilangan bulat.

Kekuatan algoritma RSA terdapat pada proses eksponensial dan dan pemfaktoran bilangan nonprima menjadi faktor primanya. Penemu algirtma RSA menyarankan dua bilangan faktor prima memiliki nilai dengan panjang lebih dari 100 digit, dengan demikian hasil kali kedua bilangan prima akan berukuran lebih dari 200 digit. Menurut Rivest dan kawan-kawan, usaha untuk menemukan faktor bilangan 200 digit membutuhkan waktu komputasi selama 4 milyar tahun dengan asumsi bahwa algoritma pemfaktoran yang digunakan merupakan algoritma yang tercepat saat ini dan komputer yang digunakan memiliki kecepatan 1 milidetik. Sampai saat ini belum ditemukan suatu algoritma yang efisien untuk melakukan pemfaktoran. Semakin besar bilangan yang difaktorkan maka semakin lama waktu yang dibutuhkan serta semakin sulit pemfaktorannya. Hal ini menjadi salah satu alasan algoritma RSA masih digunakan hingga saat ini dan masih direkomendasikan untuk digunakan dalam penyandian pesan.

2.4. Proses Pembangkitan Kunci dan Enkripsi Dekripsi Algoritma RSA

Algoritma RSA memiliki 3 langkah utama yang terdiri dari pembangkitan kunci, enkripsi, dan dekripsi. Langkah-langkah untuk melakukan pembangkitan kunci algoritma RSA adalah sebagai berikut:

a) Tentukan 2 buah bilangan prima p dan q.
b) Hitung nilai n dengan memasukkan rumus : n = p * q
c) Hitung nilai m dengan memasukkan rumus : m = (p-1) * (q-1)
d) Tentukan nilai e dengan syarat : e = e > 1 and gcd(m,e) = 1
e) Tentukan nilai d dengan syarat : d = (d*e) mod m = 1

Berdasarkan proses tersebut akan didapatkan kunci publik = (e,n) dan kunci privat = (d,n). Langkah selanjutnya adalah proses enkripsi dan dekripsi. Berikut contoh enkripsi dan dekripsi pesan algoritma RSA:

```
Plaintext = RSA,
R = 82, S = 83, A = 65
Tentukan p & q = 11 & 13
n = p * q = 143
m = (11-1) * (13-1)
m = (10) * (12)
m = 120
e = e > 1 and gcd(m,e)=1
e = gcd(120,17) = 1
d = (d*e) mod m = 1
d = (113*17) mod 120 = 1
Plaintext = M < n

Encryption:
Ciphertext = M^e(modn)
R Ciphertext  = 82^17(mod143) = 36
S Ciphertext  = 83^17(mod143)  = 96
A Ciphertext  = 65^17(mod143) = 65

Decryption:
Plaintext = C^d(modn)
R Plaintext  = 36^113(mod143) = 82
S Plaintext  = 96^113(mod143)  = 83
A Plaintext  = 65^113(mod143)  = 65
```

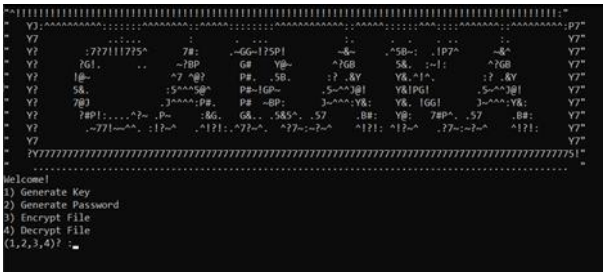
Jurnal Riset Sistem Informasi Dan Teknik Informatika (JURASIK)
Volume 8, Nomor 1, Februari 2023, pp 142-154
Terakreditasi Nomor 204/E/KPT/2022 | ISSN: 2527-5771/EISSN: 2549-7839
https://tunasbangsa.ac.id/ejurnal/index.php/jurasik

Berikut proses pembangkitan kunci serta enkripsi dan dekripsi algoritma RSA yang disajikan dalam pseudocode.

```
program RSA_Encryption and Decryption
deklarasi:
var      p,      q,      n,      phi_of_n,      public_key,      private_key,      message,
encrypted_message,decrypted_message : ZZ panjang_kunci: integer
Algoritma:
INPUT panjang_kunci
p ← GenPrime_ZZ(panjang_kunci, 80)
q ← GenPrime_ZZ(panjang_kunci, 80)
n ← p * q
phi_of_n ← (p-1)*(q-1)
i: ZZ
bits_of_n: long
begin
FOR ( ; i<=n; i*=10)
     bits_of_n++
END FOR
public_key ← GenPrime_ZZ(bits_of_n/2, 80) private_key ← InvMod(public_key, phi_of_n)
INPUT message
encrypted_message = PowerMod(message, public_key, n)
decrypted_message = PowerMod(encrypted_message, private_key, n);
OUTPUT (p, q, n, phi_of_n, public_key, private_key, encrypted_message)
```

3. HASIL DAN PEMBAHASAN

Berikut adalah tampilan antarmuka dari Caraka.bat. Pada tampilan terdapat nama dari program dan beberapa pilihan menu yang terdiri dari Generate Key, Generate Password, Encrypt Dokumen, Decrypt Dokumen.



Gambar 5. Tampilan Antarmuka Caraka.bat

Pengguna dapat menjalankan program dengan memilih pilihan yang akan dijalankan oleh program. Berikut adalah penjelasan dari masing-masing pilihan yang ada :

3.1. Generate Key

Proses pembangkitan pasangan kunci privat dan kunci publik. Program akan mengeksekusi perintah command line sebagai berikut :

OPENSSL genrsa -out private.pem 2048
OPENSSL rsa -in private.pem -outform PEM -pubout -out public.pem

Perintah pertama akan menginstruksikan program OpenSSL untuk membangkitkan kunci privat yang akan disimpan dengan nama private.pem

dengan panjang kunci 2048 bit. Perintah kedua akan menginstruksikan program OpenSSL untuk membangkitkan kunci publik dengan menggunakan kunci privat yang telah dibuat dan akan disimpan dengan nama public.pem.

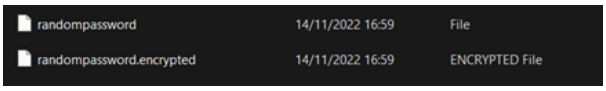
Gambar 6. Pasangan Kunci Publik dan Kunci Private yang Telah dibangkitkan.

3.2. Generate Password

Proses pembuatan password secara acak dengan memasukkan lokasi dari berkas kunci publik tersimpan. Program akan mengeksekusi perintah command line sebagai berikut :

set /p pub=[+]Path of Public Key:
OPENSSL rand -out randompassword -hex 32
OPENSSL rsautl -encrypt -inkey %pub% -pubin -in randompassword -out randompassword.encrypted

Perintah pertama akan menginstruksikan program untuk meminta input dari pengguna berupa lokasi berkas dari public.pem. Perintah kedua akan menginstruksikan program OpenSSL untuk membuat berkas random password hex 32. Perintah ketiga akan menginstruksikan program OpenSSL untuk melakukan proses enkripsi pada berkas randompassword menggunakan public.pem yang akan disimpan dengan nama randompassword.encrypted.



Gambar 7. Hasil Generate randompassword dan randompassword.encrypted

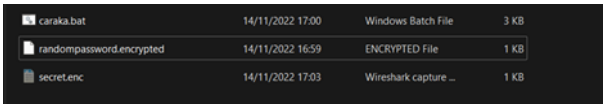
3.3. Encrypt Dokumen

Proses enkripsi berkas dengan memasukkan lokasi dari berkas yang ingin dilakukan enkripsi dan tempat tujuan untuk menyimpan berkas hasil enkripsi tersebut serta random password yang sebelumnya dibuat. Program akan mengeksekusi perintah command line sebagai berikut :

set /p dokumen=[+]Path of Dokumen:
set /p drive=[+]Destination:
set /p pw=[+]Path of password:
OpenSSL enc -p -aes-256-cbc -salt -in %dokumen% out secret.enc -pass dokumen:%pw%

Perintah pertama akan menginstruksikan program untuk meminta input dari pengguna berupa lokasi berkas yang akan dilakukan proses enkripsi. Perintah kedua akan menginstruksikan program untuk meminta input dari pengguna berupa lokasi tujuan berkas tersebut disimpan sesudah dilakukan proses enkripsi.

Perintah ketiga akan menginstruksikan program untuk meminta input dari pengguna berupa lokasi berkas dari randompassword.encrypted. Perintah keempat akan menginstruksikan program OpenSSL untuk melakukan proses enkripsi pada berkas yang telah dipilih pada perintah pertama menggunakan randompassword.encrypted yang akan disimpan dengan nama secret.enc.



Gambar 8. Hasil enkripsi

3.4. Decrypt Dokumen

Proses dekripsi berkas dengan memasukkan lokasi berkas kunci privat dan nama yang ingin digunakan untuk berkas hasil dekripsi tersebut. Program akan mengeksekusi perintah command line sebagai berikut :

```
set /p private=[+]Path of Private Key:
set /p dokumen=[+]Result dokumenname:
OpenSSL rsautl -decrypt -inkey %private% -in randompassword.encrypted -out
randompassword
attrib -h secret.enc
OpenSSL enc -d -p -aes-256-cbc -salt -in secret.enc -out %dokumen% -pass
dokumen:./randompassword
rm secret.enc
```

Perintah pertama akan menginstruksikan program untuk meminta input dari pengguna berupa lokasi berkas dari private.pem. Perintah kedua akan menginstruksikan program untuk meminta input dari pengguna berupa nama yang ingin digunakan untuk menyimpan berkas setelah dilakukan proses dekripsi. Perintah ketiga akan menginstruksikan program OpenSSL untuk melakukan proses dekripsi pada berkas randompassword.encrypted terlebih dahulu menggunakan private.pem untuk mendapatkan randompassword yang akan digunakan untuk melakukan proses dekripsi pada berkas secret.enc. Perintah keempat akan menginstruksikan program untuk mengatur atribut yang dimiliki berkas secret.enc. Perintah kelima akan menginstruksikan program OpenSSL untuk melakukan proses dekripsi pada berkas secret.enc menggunakan randompassword.

Gambar 9. Hasil dekripsi

3.5. Performa Proses Enkripsi

Berikut disajikan tabel performa proses enkripsi menggunakan program Caraka.bat yang terdiri dari nama berkas, jenis berkas, ekstensi berkas, ukuran awal berkas sebelum dilakukan proses enkripsi dalam satuan byte, ukuran akhir

Jurnal Riset Sistem Informasi Dan Teknik Informatika (JURASIK)
Volume 8, Nomor 1, Februari 2023, pp 142-154
Terakreditasi Nomor 204/E/KPT/2022 | ISSN: 2527-5771/EISSN: 2549-7839
https://tunasbangsa.ac.id/ejurnal/index.php/jurasik

berkas setelah dilakukan proses enkripsi dalam satuan byte, ekspansi ukuran berkas dalam persentase, dan waktu proses enkripsi dalam satuan detik.

Tabel 1. Performa Proses Enkripsi

No	Nama Dokumen	Type	Ekstensi	Ukuran awal (Byte)	Ukuran akhir (Byte)	Ekspansi ukuran (%)	Waktu Proses Enkripsi (s)
1	dokumen-sample_1MB.doc	Dokumen	.doc	1.027.072	1.027.104	0,00311565304	0.45
2	dokumen-example_PDF_1MB.pdf	Dokumen	.pdf	1.042.157	1.042.176	0,00182314181	0.36
3	dokumen_example_PNG_1MB.png	Gambar	.png	1.006.708	1.006.736	0,00278134275	0.09
4	dokumen_example_JPG_1MB.jpg	Gambar	.jpg	1.042.592	1.042.624	0,00306927350	0.08
5	dokumen_example_AVI_1280.avi	Video	.avi	1.480.958	1.480.976	0,00121542947	0.10
6	dokumen_example_MP4_480.mp4	Video	.mp4	1.570.024	1.570.048	0,00152863905	0.10
7	dokumen_example_MP3_1MG.mp3	Audio	mp3	1.087.849	1.087.872	0,00211426402	0.10
8	dokumen_example_WAV_1MG.wav	Audio	.wav	1.073.218	1.073.248	0,00279533142	0.09
9	leakage.exe	Other	.exe	731.136	731.168	0,0043767507	0.15
10	testdpc_8.0.1.apk	Other	.apk	5.312.283	5.312.304	0,00039531026	0.09

Dari tabel performa enkripsi tersebut dapat dilihat bahwa program akan bekerja dengan ekspansi dokumen terbaik pada dokumen dengan ekstensi .apk yaitu dengan ekspansi ukuran terkecil sebesar 0,0003953102649%. Selain itu, program juga bekerja dengan waktu proses enkripsi tercepat pada tipe dokumen gambar dengan waktu proses enkripsi sebesar 0,8 s dan 0,9 s.

4. SIMPULAN

Dalam penelitian ini, disimpulkan bahwa kriptografi asimetris dapat dimanfaatkan untuk melakukan pengamanan dokumen pada media flashdisk. Penggunaan kunci publik dan kunci privat pada sistem kriptografi asimetris berperan penting dalam membuat pengamanan dokumen menjadi lebih kuat. Penelitian ini memanfaatkan library OpenSSL untuk mengimplementasikan algoritma RSA guna mengenkripsi suatu dokumen. Perintah-perintah OpenSSL disusun dan menghasilkan sebuah program shell bernama Caraka.bat yang dapat dijalankan di dalam sebuah flashdisk sebagai media pertukaran informasi. Caraka.bat dapat melakukan proses enkripsi dan dekripsi dokumen dengan ekstensi apa saja. Program Caraka.bat memastikan dokumen yang dienkripsi aman dari serangan karena menggunakan double encryption dimana dokumen akan dienkripsi menggunakan randompassword yang dibangkitkan lalu randompassword tersebut akan dienkripsi menggunakan kunci publik. Hasil enkripsi akan menghasilkan dokumen terenkripsi yang memiliki ekspansi ukuran sedikit lebih besar dibandingkan ukuran dokumen sebelum dilakukan enkripsi. Dokumen akan bekerja dengan performa terbaik dalam hal ekspansi ukuran untuk

tipe dokumen aplikasi, dan dalam hal kecepatan proses enkripsi untuk tipe dokumen gambar.

DAFTAR PUSTAKA

[1] E. Ndruru and T. S. Alasi, "Algoritma Tripple Des Dalam Pengamanan File Dengan Usb Flashdisk," *Jurnal Informasi Komputer Logika*, vol. 2, no. 4, 2022.

[2] M. Suryawinata, Buku Ajar Arsitektur Dan Organisasi Komputer. Umsida Press, 2018. doi: 10.21070/2018/978-602-5914-11-9.

[3] H. Prilandi and D. Kusumaningsih, "Penerapan Aplikasi Kriptografi Dengan Algoritma Advanced Encryption Standard Pada Perusahaan Pt Cahaya Televisi Indonesia," 2022.

[4] J. Walden, "The Impact of a Major Security Event on an Open Source Project," in Proceedings of the 17th International Conference on Mining Software Repositories, Jun. 2020, pp. 409–419. doi: 10.1145/3379597.3387465.

[5] Ananda Sekar Putri, Saepul Lukman, and Irfan, "Analisa Metode Kriptografi Modern Advance Encryption Standard (AES) 128 Bit dalam Mengenkripsi dan Mendekripsi File Dokumen Digital," Jurnal Ilmiah Komputasi, vol. 21, no. 3, Sep. 2022, doi: 10.32409/jikstik.21.3.2973.

[6] R. Firmansyah and A. A. Permana, "Implementasi Keamanan Pesan Teks Menggunakan Kriptografi Algoritma Rsa Dengan Metode Waterfall Berbasis Java," 2019.

[7] C. H. C. Leung and Q. H. Choo, "On the Execution of Large Batch Programs in Unreliable Computing Systems," IEEE Transactions on Software Engineering, vol. SE-10, no. 4, pp. 444–450, Jul. 1984, doi: 10.1109/TSE.1984.5010258.