Jurnal JOCOTIS - Journal Science Informatica and Robotics
Vol. 1 No. 1 September 2023 Hal. 10-21
https://jurnal.ittc.web.id/index.php/jct/

E-ISSN : xxxx-xxxx
P-ISSN : xxxx-xxxx

PENERAPAN KRIPTOSISTEM *HYBRID* UNTUK MENGENKRIPSI PESAN MENGGUNAKAN ALGORITMA RSA CIPHER

Muhamad Wahyu Saputra [1] , Anjeli Sapitri [2] , Mesy Aniza Putri [3]

[1]Teknik Informatika/STMIK Amik Riau, Jl.Purwodadi Indah , Wahyustaw@gmail.com
[2]Teknik Informatika/STMIK Amik Riau, Jl.Purwodadi Indah, Sapitrianjeli@gmail.com
[3]Teknik Informatika/STMIK Amik Riau, Jl.Purwodadi Indah, Mesyaniza@gmail.com

ABSTRACT

Perkembangan Teknologi Informasi membuat seseorang dengan mudah mendapatkan informasi, dan komunikasi juga dapat dilakukan dimanapun dan kapanpun. Dalam perkembangan untuk melakukan komunikasi bukan hanya dengan telepon tetapi juga lewat sms. Untuk mengantisipasi agar pesan tidak mudah di retas oleh orang yang tidak berhak, maka perlu dibuat sistem pengamanan pada pesan tersebut. Salah satu cara untuk melakukan pengamanan tersebut adalah dengan enkripsi. Pada penelitian ini, digunakan kriptosistem hybrid dengan kombinasi Algoritma RSA cipher. Untuk menjaga keamanan dari pesan yang dikirim perlu ditambahkan teknik error detection yang dalam hal ini menggunakan hash function. Adapun proses enkripsi nya dengan melakukan enkripsi menggunakan algoritma RSA cipher dengan bahasa pemograman python. Hasil dari penelitian ini agar pesan dapat lebih kuat dan aman sehingga apabila pesan yang dikirimkan tidak bisa diretas oleh orang yang tidak bertanggung jawab maka si peretas kesulitan untuk mengetahui isi pesannya. Sehingga kerahasiaan dan keaslian pesan terjaga sampai kepada penerima pesan.

***Keywords*:** *kriptografi, kriptosistem Hybrid, algoritma RSA cipher, python*

I. PENDAHULUAN

Pada umumnya layanan dalam mengirimkan pesan memerlukan jaringan internet untuk dapat saling terhubung. Kriptografi merupakan salah satu teknik untuk melakukan penyandian pesan sehingga pesan tidak mudah dibaca atau diretas. Penelitian ini memberikan solusi keamanan pesan agar isi dari pesan tersebut aman ketika dikirimkan kepada penerimanya, maka pesan yang dikirim pada layanan tersebut dilakukan pengamanan, sehingga jika dilakukan snifing oleh seseorang maka pesan tidak dapat dibaca. Jika tidak dilakukan pengamanan akan sangat merugikan bagi pemilik pesan, terlebih pesan tersebut bersifat rahasia. Teknik yang digunakan untuk melakukan pengamanan pesan dapat dilakukan dengan penerapan kriptosistem hybrid agar lebih aman dan efisiensi.

Keamanan informasi merupakan salah satu masalah penting, seiring dengan perkembangan software dan pengguna internet. Keamanan komputer berhubungan dengan pencegahan dari pencurian data atau informasi dari orang yang tidak bertanggung jawab, baik itu mengakses dan memodifikasi informasi. Pengamanan komputer berfungsi untuk melindungi informasi agar tidak dapat diakses bagi orang yang tidak berhak. Banyak cara yang dapat digunakan dalam pengamanan komputer, salah satunya dengan menggunakan kriptografi.

Pengamanan data dengan metode kriptografi merupakan salah satu teknik yang digunakan untuk menyembunyikan pesan menjadi suatu bentuk lain sehingga tidak dapat dipahami dan diterapkan untuk mengamankan file seperti dokumen, gambar, audio, dan video. Metode kriptografi dapat diklasifikasikan menjadi 3 jenis yaitu kriptografi simetris, asimetris dan hybrid.

Dengan adanya penelitian ini, diharapkan menambah wawasan kepada pembaca dalam memahami ilmu tentang kriptografi. Dan diharapkan dapat menjadi solusi terhadap serangan kepada hak yang tidak berwenang khususnya pada saat melakukan pengiriman pesan.

II. LITERATURE REVIEW

1.1 Kriptografi

Kriptografi merupakan sebuah perlindungan keamanan pesan rahasia dengan mengacaukan dan menyandikan pesan rahasia menjadi kode-kode rahasia atau ciphertext.

Kriptografi simetris menggunakan kunci yang sama disebut kunci privat, terdiri dari metode-metode diantaranya Data Encryption Standard (DES), Rivest

Cipher 4 (RC4), Advanced Encryption Standard (AES), One Time Pad (OTP), Blowfish, dan sebagainya, sedangkan kriptografi asimetris menggunakan kunci privat dan kunci publik dalam mengamankan data misalnya algoritma RSA (Rivest Shamir Adleman), El Gamal, Elliptic Curve, Hill Cipher, Diffie-Hellman dan sebagainya. Kriptografi hybrid memanfaatkan dua tingkatan kunci, yaitu kunci rahasia (simetri) – yang disebut juga session key, untuk enkripsi data dan pasangan kunci privat-kunci publik untuk melindungi kunci simetri.

Tujuan utama penggunaan teknik kriptografi dalam pengiriman pesan rahasia terbagi menjadi beberapa poin penting, yaitu:

a. Confidentialy (Kerahasiaan), merupakan hal paling penting. Dipastikan bahwa pesan yang dikirimkan hanya bisa dimengerti oleh pengirim maupun penerima yang telah memiliki kunci untuk membuka pesan rahasia. Dalam hal ini dipastikan selain pengirim dan penerima pesan tidak seorang pun dapat mengerti pesan rahasia tersebut.

b. Authentication (keaslian), merupakan proses pembuktian identitas yang menjamin keamanan komunikasi dalam pengiriman pesan rahasia. Pengguna dan sistem dapat membuktikan identitas yang mereka miliki berhak untuk membuka pesan rahasia tersebut.

c. Data integrity (integritas data), dipastikan bahwa data yang diterima adalah data yang sama dengan data yang dikirimkan dan tidak ada perubahan data.

d. Non-Repudiation (anti penyangkalan), mencegah pihak pengirim pesan tidak mengakui bahwa telah mengirimkan sebuah pesan rahasia.

e. Access Control (kendali akses), proses yang digunakan untuk pencegahan penggunaan yang tidak sah dari sumber daya.

1.2 Algoritma RSA (Rivest-Shamir-Adleman)

Algoritma Rivest Shamir Aldeman (RSA) adalah kunci publik dan kunci rahasia dalam mengenkripsi data, dimana kunci publik boleh diketahui oleh siapa saja sedangkan kunci rahasia hanya boleh diketahui oleh pihak tertentu guna mendekripsi data. Menggunakan variable ukuran enkripsi blok dan ukuran kunci variabel. Dalam penggunaannya untuk otentikasi, server mengimplementasikan public key dengan client dengan memberikan signature pada pesan dengan menggunakan private key. Kemudian signature tersebut

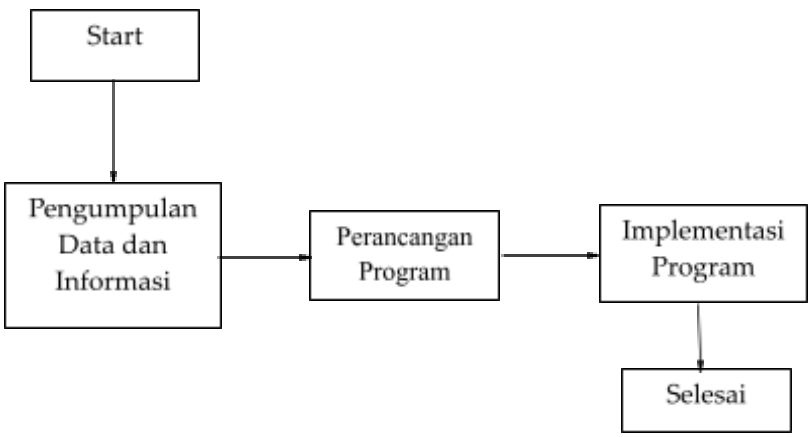
dikembalikan ke client selanjutnya diverifikasi dengan menggunakan public key yang diketahui server.

1.3 Kriptosistem Hybrid

kriptosistem hybrid merupakan teknik kriptografi dengan menggunakan dua atau lebih cipher yang berbeda dalam waktu bersamaan. Kriptosistem hybrid ini dibangun dengan menggunakan dua atau lebih kriptografi yang terpisah. Pada kriptosistem hybrid tersebut melakukan enkripsi atau dekripsi pesan yang panjang akan efisien dengan menggunakan symmetric-key. Sedangkan kunci publik hanya digunakan untuk mengenkripsi/mendekripsi kunci simetris yang pendek. Kriptografi hybrid merupakan protokol yang memanfaatkan beberapa sandi dari algoritma berbeda secara bersamaan dengan keunggulan tiap algoritma tersebut. Salah satu cara yang sering diterapkan adalah membangkitkan kunci simetris dan mengenkripsi kunci ini dengan kunci asimetris dari kunci publik penerima.

III. METHODOLOGY

Berdasarkan metode yang digunakan dalam penelitian ini adalah metode penelitian eksperimen. Tahapan yang dilakukan dalam rangka melakukan penelitian pengembangan prototipe layanan pesan dengan pengamanan pesan ditunjukkan pada gambar berikut ini:



Gambar 1. Langkah-langkah penelitian

Langkah-langkah dalam penelitian ini dijabarkan lewat uraian dibawah ini:

1. Pengumpulan Data dan Informasi
   Menurut Sugiyono (2013), menyatakan bahwa teknik pengumpulan data adalah langkah paling strategis dalam penelitian karena tujuan utama dari sebuah penelitian adalah untuk memperoleh data.
2. Perancangan Program

   Setelah melakukan pengumpulan data dan informasi, maka akan dilakukan perancangan terhadap program. Dalam melakukan perancangan program tersebut, peneliti menggunakan algoritma RSA cipher. Hal ini terdiri dari pemilihan bahasa pemrograman, struktur data, dan fungsi-fungsi yang diperlukan dalam program.

3. Implementasi Program
   Tahap terakhir yaitu melakukan Implementasi Program berupa penerapan kriptosistem hybrid dengan menggunakan algoritma RSA cipher. Terdiri dari penulisan code, pengujian, dan debugging program untuk memastikan bahwa algoritma kriptografi berjalan dengan benar dan sesuai dengan tujuan penelitian.

IV. RESULTS AND ANALYSIS

Penerapan kriptosistem hybrid ini dimulai dengan membuat fungsi untuk mengenkripsi pesan menggunakan algoritma RSA cipher. Kemudian peneliti memasukkan pesan yang akan dienkripsi dan kunci yang akan digunakan. Berikut ini adalah penerapan kriptosistem hybrid pada program yang telah peneliti buat menggunakan metode algoritma RSA cipher dengan bahasa pemrograman python.

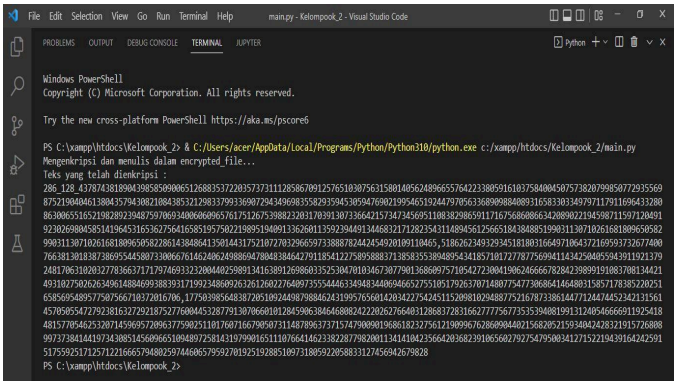
```
import sys

#ukuran blok harus lebih kecil daripada atau sama dengan ukuran kunci
#ukuran blok dalam bytes, ukuran kunci dalam bits
#ada 8 bits dalam 1 byte
UKURAN_BLOK_STANDAR = 128
#maksudnya terdapat 128 bytes
UKURAN_BYTE = 256
#dalam satu byte memiliki 256 nilai yang berbeda

def main():
    #jalankan sebuah test yang mengenkripsi sebuah pesan dalam sebuah file atau mendekrip sebuah pesan dari file
    namafile = 'encrypted_file.txt'
    #file untuk menulis atau untuk membaca
    mode = 'encrypt'
    #tentukan apakah enkripsi atau dekripsi

    if mode == 'encrypt':

        pesan = """Journalists belong in the gutter because that is where the ruling classes throw their guilty secrets." -Gerald Priestland "The Founding Fathers gave the free press the protection it must have to bare the secrets of government and inform the people." -Hugo Black"""
        namaKunciFilePublik = 'matius_celcius_sinaga_pubkey.txt'
        print('Mengenkripsi dan menulis dalam %s...' % (namafile))
        teksTerenkripsi = enkripsidantulisDalamFile(namafile, namaKunciFilePublik, pesan)

        print('Teks yang telah dienkripsi :')
        print(teksTerenkripsi)

    elif mode == 'decrypt':
        namaKunciFilePrivate = 'matius_celcius_sinaga_privkey.txt'
        print('Membaca dari %s dan mendekripsikan...' % (namafile))
        pesanTerdekripsi = membacadariFiledanDekripsi(namafile, namaKunciFilePrivate)

        print('Teks yang telah dekripsi:')
        print(pesanTerdekripsi)



def menetapkanBlokblokDariTeks (pesan, ukuranBlok=UKURAN_BLOK_STANDAR):
    #mengubah sebuah pesan menjadi sebuah daftar blok bilangan bulat
    #setiap bilangan bulat menghasilkan 128 (ukuran ukuranBlok) karakter

    pesanDalamByte = pesan.encode('ascii')
    #mengubah bilangan bulat menjadi bytes

    blockInts = []
    for mulaiBlok in range(0, len(pesanDalamByte), ukuranBlok):
        #menghitung blok bilangan bulat pada blok pada teks
        blockInt = 0
        for i in range(mulaiBlok, min(mulaiBlok + ukuranBlok, len(pesanDalamByte))):
            blockInt += pesanDalamByte[i] * (UKURAN_BYTE ** (i % ukuranBlok))
        blockInts.append(blockInt)
    return blockInts

def menetapkanTeksDariBlokblok (blockInts, panjangPesan, ukuranBlok=UKURAN_BLOK_STANDAR):
```

```
    #mengubah daftar blok bilangan bulat pada pesan asli
    #pesan asli panjang yang dibutuhkan secara khusus
    #untuk mengubah bilangan bulat blok terakhir
    pesan = []
    for blockInt in blockInts:
        pesanBlok = []
        for i in range (ukuranBlok - 1, -1, -1):
            if len(pesan) + i < panjangPesan:
                #mengubah pesan string menjadi 128
                #sesuai dengan ukuranBlok
                #karakter pada blok bilangan bulat
                angkaASCII = blockInt // (UKURAN_BYTE ** i)
                blockInt = blockInt % (UKURAN_BYTE ** i)
                pesanBlok.insert(0, chr(angkaASCII))
        pesan.extend(pesanBlok)
    return ''.join(pesan)


def enkripsiPesan(pesan, kunci, ukuranBlok=UKURAN_BLOK_STANDAR):
    #mengubah pesan menjadi daftar blok bilangan bulat
    #lalu enkripsi setiap blok bilangan bulat
    #melalui kunci PUBLIK untuk enkripsi
    blokTerenkripsi = []
    n, e = kunci

    for blok in menetapkanBlokblokDariTeks(pesan, ukuranBlok):
        # ciphertext = plaintext ^ e mod n
        blokTerenkripsi.append(pow(blok, e, n))
    return blokTerenkripsi

def dekripsiPesan(blokTerenkripsi, panjangPesan, kunci,
ukuranBlok=UKURAN_BLOK_STANDAR):
    #dekrip daftar blok enkripsi yang ada menjadi pesan asli
    #panjang pesan asli yang dibutuhkan harus sesuai dengan blok akhirnya
    #pastikan agar kunci privatnya untuk melakukan dekripsi
    blokTerdekripsi = []
    n, d = kunci
    for blok in blokTerenkripsi:
        #teks awal = sandi teks ^ d mod n
        blokTerdekripsi.append(pow(blok, d, n))
```

```
    return menetapkanTeksDariBlokblok(blokTerdekripsi, panjangPesan, ukuranBlok)

def membacaKunciFile(namaFileKunci):
    #berikan nama file untuk setiap file yang memiliki kunci publik atau private
    #menghasilkan kunci sebagai berikut (n,e) atau (n,d) nilai tuple
    fo = open(namaFileKunci)
    content = fo.read()
    fo.close()
    ukurankunci, n, EorD = content.split(',')
    return (int(ukurankunci), int(n), int(EorD))

def enkripsidantulisDalamFile(namaFilePesan, namaFileKunci, pesan, ukuranBlok=UKURAN_BLOK_STANDAR):
    #dengan menggunakan kunci dari file kunci, enkripsi seluruh pesan dan simpan menjadi file
    #menghasilkan pesan yang sudah dienkripsi
    ukurankunci, n, e = membacaKunciFile(namaFileKunci)


    #cek bahwa ukuran kunci lebih besar daripada ukuran blok
    if ukurankunci < ukuranBlok * 8:
    #* 8 ubah bytes menjadi bits
        sys.exit('ERROR: Ukuran blok adalah %s bits dan ukuran kunci adalah %s bits. RSA cipher membutuhkan ukuran blok yang sama atau lebih besar daripada ukuran kunci. Begitu juga dengan kenaikan ukuran blok atau menggunakan kunci yang berbeda.' % (ukuranBlok * 8, ukurankunci))

    #mengenkripsi pesan
    blokTerenkripsi = enkripsiPesan(pesan, (n, e), ukuranBlok)

    #menguah nilai lebih besar dari satu jenis nilai lainnya
    for i in range(len(blokTerenkripsi)):
        blokTerenkripsi[i] = str (blokTerenkripsi[i])
    kontenYangTerenkripsi = ','.join(blokTerenkripsi)

    #menuliskan pesan yang sudah terenkripsi pada hasil keluaran file
    kontenYangTerenkripsi = '%s,%s,%s' % (len(pesan), ukuranBlok, kontenYangTerenkripsi)
    fo = open(namaFilePesan, 'w')
    fo.write(kontenYangTerenkripsi)
```

```
    fo.close()
    #juga menghasilkan string yang terenkripsi
    return kontenYangTerenkripsi

def membacadariFiledanDekripsi(namaFilePesan, namaFileKunci):
    #dengan menggunakan kunci pada file kunci, membaca sebuah pesan terenkripsi
dari sebuah file lalu mendekripnya
    #menghasilkan pesan terdekripsi dalam bentuk string
    ukurankunci, n, d = membacaKunciFile(namaFileKunci)

    #membaca panjang pesan dan mengenkripsi pesan tersebut dari file
    fo = open(namaFilePesan)
    content = fo.read()
    panjangPesan, ukuranBlok, pesanTerenkripsi = content.split('_')
    panjangPesan = int(panjangPesan)
    ukuranBlok = int(ukuranBlok)


    #cek bahwa ukuran kunci lebih besar dari ukuran blok
    if ukurankunci < ukuranBlok * 8:
    # * 8 diubah dari bytes menjadi bits
        sys.exit('ERROR : Ukuran blok adalah %s bits dan ukuran kunci adalah %s bits.
RSA cipher membutuhkan ukuran blok yang sama atau lebih besar daripada ukuran
kunci. Apakah anda telah benar-benar memeriksa kunci file dan melakukan enkripsi
pada file ?' % (ukuranBlok * 8, ukurankunci))

    #mengubah pesan terenkripsi hingga menjadi lebih besar dari nilai integer
    blokTerenkripsi = []
    for blok in pesanTerenkripsi.split(','):
        blokTerenkripsi.append(int(blok))

    #mendekrip nilai int yang lebih besar
    return dekripsiPesan(blokTerenkripsi, panjangPesan, (n,d), ukuranBlok)

#jika rsaCipher.py sudah berjalan termasuk didalamnya seluruh modul maka panggil
fungsi main()
if __name__ == '__main__':
    main()
```

E-ISSN : xxxx-xxxx
P-ISSN : xxxx-xxxx



Gambar 2. Hasil Pengujian Program

Peneliti melakukan pengujian terhadap penerapan kriptosistem hybrid menggunakan algoritma RSA cipher yang telah di buat. Pengujian ini melibatkan beberapa kasus uji dengan variasi pesan dan kunci. Hasil pengujian menunjukkan bahwa penerapan kriptosistem hybrid ini berhasil mengenkripsi dan mendekripsi pesan dengan benar. Keberhasilan ini menunjukkan bahwa algoritma RSA cipher dapat digunakan sebagai metode enkripsi yang sederhana dan efektif.

V. KESIMPULAN DAN SARAN

Pada hasil penelitian ini dapat ditarik kesimpulan bahwa Algoritma Rivest Shamir Aldeman (RSA) adalah algoritma untuk enkripsi kunci publik (public-key encryption). Dengan menggunakan algoritma RSA cipher ini dapat dipakai dalam membuat kata sandi pesan maupun informasi dengan cara mengubahnya menjadi kata sandi yang tidak dapat dikenal oleh orang yang tidak berhak mendapatkannya atau tidak berwenang. Selain itu tidak ada bab "Meretas Cipher RSA" karena tidak ada serangan langsung terhadap matematika di balik sandi RSA. Dan serangan brute-force apa pun akan gagal, karena ada terlalu banyak kunci yang mungkin untuk dicoba. Secara umum algoritma RSA adalah cipher enkripsi nyata yang digunakan dalam perangkat lunak enkripsi profesional. Peretasan terhadap program enkripsi seperti rsa Cipher.py cukup canggih, tetapi memang ada. Sebuah sandi hanya aman jika semuanya kecuali kuncinya dapat diungkapkan tetapi tetap merahasiakan pesannya. Untuk

peneliti lebih lanjut dari pemaparan di atas mengenai perancangan aplikasi Kriptografi dengan menggunakan algoritma RSA cipher hingga pada tahapan Penerapan, tentunya masih butuh dicoba pengembangan sistem supaya bisa jadi aplikasi yang agar bisa meningkatkan tata cara kriptografi yang lain sehingga lebih variatif dan memperindah tampilan supaya lebih interaktif serta menarik.

DAFTAR PUSTAKA

Ahmad Pudoli dan Dewi Kusumaningsih. (2017). *Penggunaan hybrid cryptosystem untuk enkripsi dan dekripsi pesan messanger menggunakan algoritma rivest shamir adleman (rsa) dan advanced encryption standard (aes) dengan firebase pada android*. Jurnal Telematika Mkom Vol.9 No.3.

Gupta, Ravindra Kumar andParvinder Singh. (2013). *A New Way to Design and Implementation of Hybrid Cryptosystem for Security of The Information in Public Network*. International Journal of Emerging Technology and Advanced Engineering, Vol. 3.

Gutub, Adnan Abdul-Aziz. (2010). *Pixel Indicator Technique for RGB Image Steganography*. Journal of Emerging Technologies in Web Intelligence, Vol.2, No.1.

Abutaha, Mohammed, et.al. (2011). *Cryptography is The Science of Information Security*,Communication Theory of Secrecy Systems, Vol.5, No.3.

Seth, Shashi Mehrotra and Rajan Mishra. (2011). *Comparative Analysis Encryption Algorithms For Data Communication*. International Journal of Computer Science and Technology (IJCST), Vol. 2, No. 2.

Lukas, Samuel and Ni Putu Sri Artati. (2007). *Analisis Waktu Enkripsi-Dekripsi File Text Menggunakan Metoda One-Time Pada (OTP) dan Rivest, Shamir,Adleman (RSA)*. Seminar Nasional Sistem dan Informatika.

Sugiyono (2023). "Teknik pengumpulan data: pengertian, proses, dan jenis nya".https://blog.rumahweb.com/teknik-pengumpulan-data-adalah/

D. Ariyus. (2008). *Pengantar Ilmu Kriptografi: Teori, Analisis & Implementasi*, Yogyakarta: C.V Andi Offset, 2008.

R. Y. Rifai, Y. Chirstyono and I. Santoso. (2016). *Implementasi Algoritma Kriptografi Rivest Code 4, Rivest Shamir Adleman Dan Metode Steganografi Untuk*

Pengamanan Pesan Rahasia Pada Berkas Teks Digital," Transient Vol. 5 No. 1, pp. 86-91.

Sebastian Suhandinata, ,Reyhan Achmad Rizal , Dedy OngkyWijaya , Prabhu Warren , Srinjiwi. (2019). *Analisis performa kriptografi hybrid algoritma blowfish dan algoritma RSA*. JURTEKSI (Jurnal Teknologi dan Sistem Informasi) ISSN 2407-1811 (Print) Vol. VI No. 1.

R. P. S and V. Paul. (2011)."*A Hybrid Crypto System Based On A New Cicle-Symmetric Key Algorithm And RSA With CRT Asymmetric Key Algorithm For E-commerce Application*," International Journal of Computer Applications, pp. 14-18, 2011