Jurnal Riset Sains dan Teknologi
Volume 8 No. 1 Maret 2024, 1-6

e-ISSN: 2549-9750
p-ISSN: 2579-9118

Kriptosistem Hybrid Algoritme RSA dan El-Gamal Menggunakan *Socket* TCP pada *Instant Messaging*

Hybrid Cryptosystem of RSA and El-Gamal Algorithms Using TCP Socket in Instant Messaging

Aminudin[1*], Luqman Hakim[2], Ilyas Nuryasin[3], Handika Rama Santiyas[4]
[1,2,3]Program Studi Informatika, Fakultas Teknik,
Universitas Muhammadiyah Malang
Jl. Raya Tlogomas No. 246 Malang, Indonesia

*Corresponding author: aminudin2008@umm.ac.id

DOI: 10.30595/jrst.v8i1.17124

Histori Artikel:

Diajukan:
16/03/2023

Diterima:
16/01/2024

Diterbitkan:
30/03/2024

ABSTRAK

Kriptografi adalah ilmu yang digunakan untuk mengamankan pesan agar hanya dapat dibaca oleh pihak yang berwenang. Salah satu teknik kriptografi yang umum digunakan adalah RSA dan El-Gamal. RSA adalah teknik kriptografi asimetris yang menggunakan kunci publik dan kunci pribadi untuk enkripsi dan dekripsi pesan. Sementara El-Gamal adalah teknik kriptografi yang juga asimetris, tetapi menggunakan operasi eksponensial modular pada bilangan prima sebagai dasar dari Algoritmenya. Pada paper ini akan dibahas kombinasi Teknik Algoritme RSA dan El-Gamal untuk mempersulit memecahkan pesan bagi pihak yang tidak punya kepentingan. Dalam metode ini, pesan dienkripsi dengan El-Gamal menggunakan kunci sesi yang hanya diketahui oleh pengirim dan penerima. Kunci sesi ini kemudian dienkripsi dengan RSA untuk memastikan bahwa hanya penerima yang dapat membaca pesan tersebut. Berdasarkan hasil pengujian menunjukkan bahwa metode yang diusulkan mampu menyulitkan pembacaan pesan bagi pihak yang tidak berkepentingan dibandingkan dengan menggunakan RSA atau El-Gamal secara terpisah dengan menggunakan menggunakan metode penyerangan *baby step-giant step*.

Kata Kunci: RSA, *El-Gamal*, *Instant Messaging*, *Modular*

ABSTRACT

Cryptography is the science used to secure messages so they can only be read by authorized parties. One of the commonly used cryptographic techniques is RSA and ElGamal. RSA is an asymmetric cryptographic technique that uses a public key and a private key for encryption and decryption of messages. Meanwhile ElGamal is a cryptographic technique that is also asymmetric, but uses modular exponential operations on prime numbers as the basis of its algorithm. In this research, we propose a new cryptographic method that combines RSA and ElGamal techniques to improve message security. This method uses RSA encryption to secure the session key which is then used in encrypting messages with ElGamal. In this method, messages are encrypted with ElGamal using a session key that only the sender and receiver know. This session key is then encrypted with RSA to ensure that only the recipient can read the message. The test results show that the proposed method is able to improve message security compared to using RSA or ElGamal separately using a baby step-giant step. In addition, this method also has quite good performance and can be applied to various applications that require secure cryptography, one of which is instant messaging.

Keywords: *RSA, El.Gamal, Instant Messaging, Modular*

1. PENDAHULUAN

Instant messaging (IM) adalah salah satu aplikasi komunikasi yang sangat populer dan banyak digunakan oleh orang-orang di seluruh dunia. Namun, penggunaan IM sering kali membawa risiko keamanan, terutama ketika pesan yang dikirim berisi informasi rahasia. Oleh karena itu, kriptografi menjadi semakin penting dalam penggunaan aplikasi IM (Aminudin et al., 2018). Teknologi yang digunakan untuk mengirim pertukaran data tersebut menggunakan *socket TCP*. Proses untuk pengamanan data tersebut dapat menggunakan Algoritme kunci *public* diantaranya adalah RSA dan El-Gamal.

RSA dan El-Gamal adalah dua teknik kriptografi yang umum digunakan untuk memproteksi pesan dalam aplikasi IM. RSA menggunakan faktorisasi bilangan bulat besar sebagai dasar Algoritme enkripsi dan dekripsi, sedangkan El-Gamal menggunakan operasi eksponensial modular pada bilangan prima. Kedua teknik ini memiliki keunggulan dan kelemahan masing-masing dalam hal keamanan dan efisiensi.

Algoritme RSA didasarkan pada masalah faktorisasi bilangan bulat (IFP), sedangkan Algoritme El-Gamal didasarkan pada masalah *logaritma diskrit* (DLP), keduanya memberikan kecepatan komputasi untuk *kriptosistem* asimetris dan bergantung pada kesulitan penyelesaian kedua masalah ini (Rutkowski & Houghten, 2020) (Rezal et al., 2018) (Meneses et al., 2016). IFP dari Algoritme RSA adalah kesulitan faktorisasi yang digunakan pada *generate key* Algoritme RSA (N=p x q) dan (e x d ≡1 mod Φ(N)). Sedangkan Algoritme El-Gamal didasarkan pada sulitnya menemukan masalah *logaritma diskrit* (DLP) pada Algoritme El-Gamal, yaitu sulitnya metode diferensial yang digunakan pada pembangkitan kunci El-Gamal (y= g^x mod pEL) (Munir, 2019) (Poulakis, 2020). Penelitian ini ini menyimpulkan bahwa keamanan kombinasi berdasarkan IFP dan DLP akan mencegah banyak kerentanan yang disebabkan oleh standar El-Gamal dan keamanan RSA. Penelitian yang mengacu pada kombinasi Algoritme RSA dan El-Gamal masih sedikit yang dipelajari karena kompleksitas Algoritme El-Gamal yang sangat terbatas untuk dikembangkan dibandingkan dengan Algoritme kriptografi lainnya. Berbeda dengan Algoritme RSA, yang merupakan subyek dari banyak penelitian dan pengembangan Algoritme RSA, Algoritme ini lebih fleksibel. Penelitian dan perancangan Algoritme pembangkitan kunci untuk Algoritme RSA dan El-Gamal yang dianggap dalam penelitian ini lebih aman dan lebih singkat daripada pembangkitan kunci menggunakan Algoritme RSA standar (Aminudin & Nuryasin, 2021).. Hanya kombinasi pembangkitan kunci dari Algoritme RSA dan El-Gamal yang dipelajari, bukan enkripsi atau dekripsi. Kajian lain yang mengangkat topik penggabungan RSA dan El-Gamal adalah tentang enkripsi ganda atau bisa disebut enkripsi, hasil dari Algoritme enkripsi RSA dienkripsi kembali menggunakan Algoritme El-Gamal (Iswari, 2017).

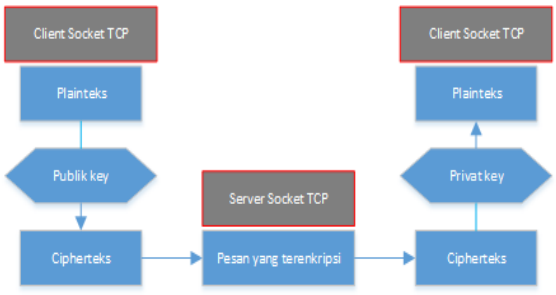
Metode ini menggunakan enkripsi RSA untuk mengamankan kunci sesi yang kemudian digunakan dalam enkripsi pesan dengan El-Gamal. Kunci sesi ini hanya diketahui oleh pengirim dan penerima pesan, sehingga pesan tidak dapat dibaca oleh orang lain yang tidak berwenang. Untuk menguji keamanan metode yang diusulkan, kami menggunakan pengujian keamanan *baby-step giant-step*. Pengujian ini dapat menentukan apakah kunci yang dihasilkan dari metode yang digunakan benar-benar tidak dapat diketahui oleh pihak yang tidak mempunyai kepentingan.

2. METODE PENELITIAN

Penelitian ini didasarkan atas kombinasi teori pemfaktoran dan logaritma diskrit dengan tujuan untuk melindungi kunci publik dan kunci *privat* yang tertanam di dalam Algoritme tersebut. adapun alur yang digunakan untuk menyusun penelitian ini adaah sebagai berikut:

2.1 Rancangan Arsitektur Sistem

Sistem yang dirancang berbasis dekstop dengan menggunakan bahasa pemrograman Java. Jaringan yang digunakan pada *socket TCP* adalah jaringan *localhost*. Sistem yang dibangun yaitu *server socket TCP* dan *client socket TCP*. *Server socket TCP* digunakan untuk menyediakan layanan percakapan (*chating*) antar *client* dan menyimpan daftar *user online* beserta pasangan kunci publiknya (N, e) dan (y, g, p). Sedangkan *client socket TCP* digunakan untuk menerima dan mengirim pesan (*chating*) pada *client socket TCP* lainnya setelah berhasil terkoneksi dengan *server socket TCP*. *Client socket TCP* dapat melakukan *chatting* setelah minimal dua *client socket TCP* yang terkoneksi dengan *server socket TCP*.



Gambar 1. Rancangan Sistem IM

2.2 Rancangan Pembangkitan Kunci Algoritme *Hybrid*

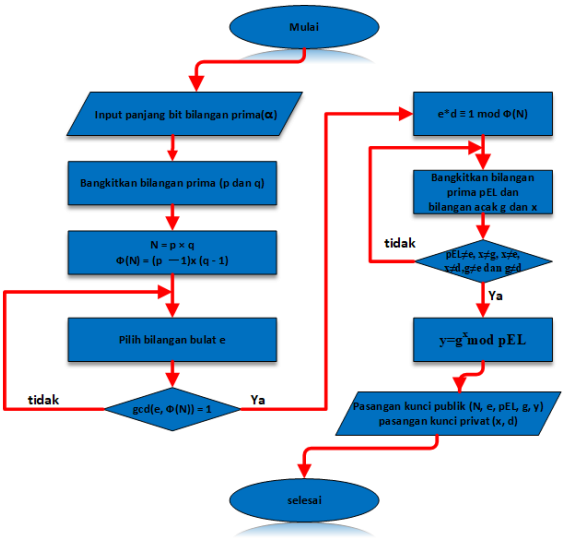
Skenario pemakaian IM yaitu *Client TCP* A melakukan koneksi ke *Server TCP* untuk melakukan *binding port* yang akan digunakan untuk melakukan komunikasi dengan *Client TCP B*. *Client TCP A* memilih bilangan prima acak dengan ukuran bit yang sama p dan q. Kemudian hitung $N = p\ x\ q$ pastikan nilai p dan q tidak sama dengan $N = p2$ untuk menghindari akar primitif pada nilai N. Adapun langkah-langkah kriptosistem hybrid:

Algoritme 1. Pembangkitan Kunci

Input : bilangan prima ($p\ dan\ q$)
Output : Kunci public (N, e, pEL, g, y) dan kunci privat (d, x)

1. $N = p\ x\ q$
   $\Phi(N) = (p-1)\ x\ (q-1)$
2. e dengan $\gcd(e, \Phi(N)) = 1$
3. Eksponen dekripsi d
   e $x\ d\ \equiv 1\ mod\ \Phi(N))$
4. pEL dengan $pEL \neq e > 0$
5. ≠g, x≠e dan x≠d, g≠e dan g≠d
6. Hitung $y = g^x\ mod\ pEL$

Pada Algoritme pembangkitan kunci poin 1 sampai dengan 3 merupakan hasil perhitungan Algoritme RSA dan poin 4 sampai dengan 6 merupakan hasil perhitungan perhitungan dari Algoritme El-Gamal. Adapun *flowchart* pembangkitan kunci pada Algoritme hybrid antara RSA dengan El-Gamal ditunjukan pada Gambar 2 sebagai berikut.



Gambar 2. Rancangan Pembangkitan Kunci Algoritme *Hybrid*

2.3 Rancangan Enkripsi Algoritme *Hybrid*

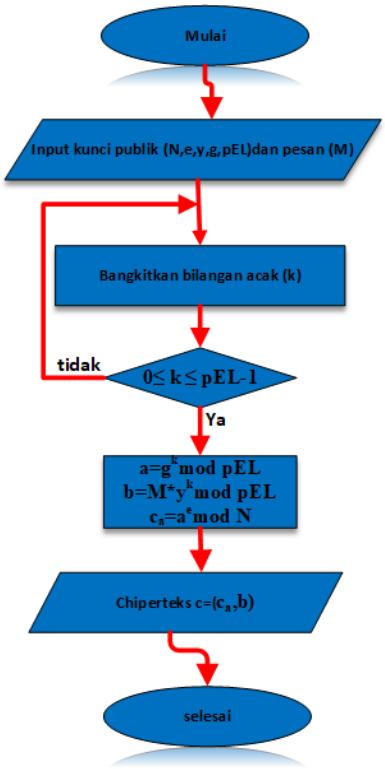
Simulasi proses enkripsi pada penelitian ini yaitu Client TCP A mengirim pesan M ke Client TCP B dimana pesan M direpresentasikan sebagai bilangan bulat dan dilakukan enkripsi untuk menghindari data dibaca oleh pihak yang tidak diinginkan. Metode enkripsi sebagai berikut:

Algoritme 2. Proses Enkripsi

Input : kunci publik (N, e, pEL, g, y)
Output : chipertext

1. blok $m_1, m_2 \dots \dots \dots m_n$ [0, pEL-1].
2. Bangkitkan nilai k 0≤ k ≤ pEl-1
3. $a = g^k\ mod\ pEL$
4. $b = m\ x\ y^k\ mod\ pEL$ dan $c_a = a^e\ mod\ n$

Pada Algoritme proses enkripsi ini perhitungan point 3 dan 4 merupakan hasil kombinasi antara Algoritme RSA dengan El-Gamal. Adapun *flowchart* untuk proses enkripsi pada Algoritme *hybrid* antara RSA dengan El-Gamal ditunjukan pada Gambar 3 sebagai berikut.



Gambar 3. Rancangan proses enkripsi Algoritme *Hybrid*

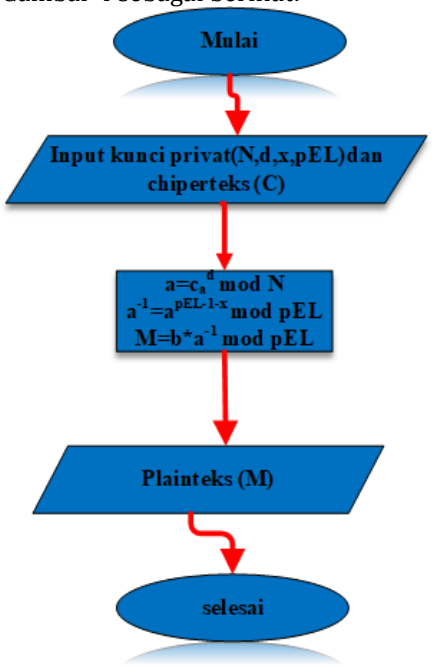
2.4 Rancangan Dekripsi Algoritme *Hybrid*

Skenario atau simulasi pada proses dekripsi ini yaitu client TCP B akan menerima pesan *chiperteks* dari Client TCP A, untuk membaca isi pesan tersebut *Client TCP B* melakukan penyusunan yang merepresentasikan nilai M dengan perhitungan sebagai berikut:

Algoritme 3. Proses Dekripsi

Input : kunci privat (d, x)
Output : plaintext
1. $a = c_a^d \ mod \ n$
2. Invers a sama dengan $a^{-1} = a^{pEL-1-x} \ mod \ pEL$
3. $m = b \ x \ a^{-1} \ mod \ pEL$

Adapun flowchart untuk proses pengembalian pesan semula pada Algoritme *hybrid* antara RSA dengan El-Gamal ditunjukan pada Gambar 4 sebagai berikut:



Gambar 4. Rancangan proses dekripsi Algoritme *Hybrid*

3. HASIL DAN PEMBAHASAN

Pengujian dilakukan untuk menguji performa dan keamanan Algoritme RSA, EL-GAMAL dan *Hybrid*. Pengujian performa meliputi pembangkitan kunci, enkripsi dan dekripsi. Pengujian keamanan dilakukan metode fermat faktorisasi untuk algoritme RSA dan *baby step-giant* step untuk El-Gamal serta gabungan keduanya untuk *Algoritme Hybrid*.

3.1 Hasil Pengujian Performa Pembangkitan Kunci

Pembangkitan kunci salah satu faktor yang paling menentukan di dalam melakukan proses enkripsi dan dekripsi. Pembangkitan kunci juga yang menentukan prioritas keamanan di dalam Algoritme kriptografi karena di dalam proses inilah terjadi pembentukan kunci baik kunci publik maupun kunci privat. Adapun hasil pengujian performa algoritme RSA, El-Gamal dan *hybrid* adalah berikut:

Tabel 1. Perbandingan waktu pembangkitan kunci

Algoritme	64 bit (ms)	128 bit (ms)	256 bit (ms)
RSA	1.459	4.337	8.816
El-Gamal	0.148	1.534	3.643
Hybrid	0.575	3.342	8.421

Berdasarkan Tabel 1 dapat Tarik pernyataan bahwa waktu pembangkitan kunci Algoritme EL-GAMAL lebih cepat dibandingkan dengan Algoritme RSA dan Algoritme *Hybrid*. Pembangkitan kunci juga bergantung pada panjang bit bilangan prima, semakin besar bit bilangan tersebut maka semakin lama pembangkitan kuncinya.

3.2 Hasil Pengujian Performa Proses Enkripsi

Proses enkripsi digunakan untuk mengubah data yang terbaca menjadi data yang teracak. Pada hasil kali ini akan dibandingkan beberapa algoritme untuk mengukur performa ketiga Algoritme sesuai dengan bit yang digunakan.

Tabel 2. Perbandingan waktu proses enkripsi 64 bit

Panjang Karakter	Algoritme RSA (ms)	Algoritme El-Gamal (ms)	Algoritme *Hybrid* (ms)
50	1.918	0.750	1.536
100	2.737	1.263	2.578
160	4.129	2.790	4.968

Tabel 3. Perbandingan waktu proses enkripsi 128 bit

Panjang Karakter	Algoritme RSA (ms)	Algoritme El-Gamal (ms)	Algoritme *Hybrid* (ms)
50	5.634	1.719	4.056
100	9.710	3.146	8.664
160	13.116	5.784	14.672

Hasil perbandingan rata-rata waktu enkripsi berdasarkan Tabel 2 dan Tabel 3 bahwa waktu enkripsi algortima El-Gamal lebih cepat dibandingkan waktu enkripsi Algoritme RSA dan Algoritme *Hybrid*. Komputasi untuk melakukan enkripsi dari Algoritme El-Gamal lebih sedikit dibandingkan dengan Algoritme RSA dan Algoritme *Hybrid*. Waktu enkripsi Algoritme *Hybrid* lebih cepat dibandingkan waktu enkripsi Algoritme RSA. Komputasi Algoritme RSA lebih sedikit dari Algoritme *Hybrid* dalam enkripsi, hal ini dikarenakan Algoritme *Hybrid* adalah Algoritme penggabungan Algoritme RSA dengan Algoritme El-Gamal yang sudah bagian sebelumnya. Panjang bit bilangan prima

mempengaruhi waktu enkripsi. Waktu enkripsi Algoritme RSA, El-Gamal dan *Hybrid* lebih cepat menggunakan 64 bit dibandingkan dengan menggunakan bilangan prima 128 bit.

3.3 Hasil Pengujian Performa Proses Dekripsi

Proses dekripsi berfungsi untuk mengubah data yang teracak menjadi data kembali seperti semula seperti sebelum data tersebut diproses enkripsi. Pada hasil kali ini akan dibandingkan beberapa algoritme untuk mengukur performa ketiga Algoritme sesuai dengan bit yang digunakan.

Tabel 4. Perbandingan waktu proses dekripsi 64 bit

Panjang Karakter	Algoritme RSA (ms)	Algoritme El-Gamal (ms)	Algoritme *Hybrid* (ms)
50	9.088	0.299	2.300
100	9.253	0.427	4.608
160	12.318	0.792	7.328

Tabel 5. Perbandingan waktu proses dekripsi 128 bit

Panjang Karakter	Algoritme RSA (ms)	Algoritme El-Gamal (ms)	Algoritme *Hybrid* (ms)
50	14.534	0.482	6.873
100	20.970	0.636	13.233
160	22.983	0.837	20.953

Hasil perbandingan rata-rata waktu dekripsi pada Tabel 4 dan Tabel 5 didapatkan bahwa waktu dekripsi algortima El-Gamal lebih cepat dibandingkan waktu dekripsi Algoritme RSA dan Algoritme *Hybrid*. Komputasi untuk melakukan dekripsi dari Algoritme El-Gamal lebih sedikit dibandingkan dengan Algoritme RSA dan Algoritme *Hybrid*. Waktu dekripsi Algoritme *Hybrid* lebih cepat dibandingkan waktu dekripsi Algoritme RSA. Komputasi Algoritme RSA lebih sedikit dari Algoritme *Hybrid* dalam dekripsi. Panjang bit bilangan prima mempengaruhi waktu dekripsi. Waktu dekripsi Algoritme RSA, El-Gamal dan *Hybrid* lebih cepat menggunakan 64 bit dibandingkan dengan menggunakan bilangan prima 128 bit.

3.4 Hasil Pengujian Keamanan menggunakan Fermat Faktorisasi

Algoritme RSA di dalam beberapa literature sangat rentan terhadap serangan fermat faktorisasi. Oleh karena itu di dalam penelitian ini akan diuji tingkat keamanan Algoritme RSA berdasarkan dari besaran bit ketika pembangkitan kunci beserta lamanya waktu eksekusi.

Tabel 6. Hasil Pengujian Keamanan Algoritme Hybrid menggunakan Fermat Faktorisasi

Panjang bit bilangan prima	Waktu eksekusi (ms)	Status kunci privat
16	8	ditemukan
32	5426284	ditemukan
64	32136219	tidak ditemukan

Dari hasil pengujian Fermat Faktorisasi pada Tabel 8 didapatkan bahwa Algoritme RSA dapat diserang menggunakan *fermat faktorisasi* ini berhenti di panjang bit bilangan prima 64 bit, hal ini dikarenakan *resource* penelitian kurang mencukupi. Resource untuk pengujian Fermat Faktorisasi ini membutuhkan *storage* yang besar untuk menampung data *dummy* dari program yang dijalankan.

3.5 Hasil Pengujian Keamanan menggunakan *Baby step-Giant Step*

Keamanan Algoritme El-Gamal bergantung pada pemecahan masalah logaritma diskrit dengan modulo yang sangat besar. Oleh karena itu, di dalam penelitian akan diuji cobakan keamanan Algoritme *hybrid* ini dengan menggunakan *baby step-giant step*.

Tabel 7. Hasil Pengujian Keamanan Algoritme *Hybrid* menggunakan *Baby Step – Giant Step*

Panjang bit bilangan prima	Waktu eksekusi (ms)	Status kunci privat
16	213	ditemukan
32	61744	ditemukan
64	5606126	tidak ditemukan

Berdasarkan Tabel 7 waktu eksekusi program tergolong cepat untuk panjang bit bilangan prima kecil, semakin besar panjang bit bilangan prima maka semakin lama waktu yang diperlukan. Pengujian *Baby step-Giant step* ini berhenti di panjang bit bilangan prima 64 bit, hal ini dikarenakan resource penelitian kurang mencukupi. Resource untuk pengujian *Baby step-Giant step* ini membutuhkan memori yang besar untuk menampung data *dummy* dari program yang dijalankan.

4. KESIMPULAN

Metode Hybrid Algoritme RSA dan El-Gamal memberi keamanan berganda dari penggabungan masalah *faktorisasi integer* (IFP) dan *logaritma diskrit* (DLP) sehingga akan jauh lebih sulit untuk mendapatkan kunci *privat* dari Algoritme *Hybrid* RSA dan El-Gamal. Kompleksitas Algoritme *Hybrid* RSA dan El-Gamal dari segi eksekusi waktu cenderung lebih lama dikarenakan penggabungan yang dilakukan

pada proses enkripsi dan dekrips. Pada Algoritme *Hybrid* ini juga terhitung lebih susah ditembus menggunakan metode *fermat factorization* dan *baby step – giant step*.

DAFTAR PUSTAKA

Ahmed, J. M., & Ali, Z. M. (2011). The enhancement of computation technique by combining RSA and El-Gamal Cryptosystems. *Proceedings of the 2011 International Conference on Electrical Engineering and Informatics, ICEEI 2011*, *July*. https://doi.org/10.1109/ICEEI.2011.6021779

Aminudin, A., Helmi, A. F., & Arifianto, S. (2018). Analisa Kombinasi Algoritme Merkle-Hellman Knapscak dan Logaritma Diskrit pada Aplikasi Chat. *Jurnal Teknologi Informasi Dan Ilmu Komputer*, *5*(3), 325–334. https://doi.org/http://dx.doi.org/10.25126/jtiik.201853844

Aminudin, A., & Nuryasin, I. (2021). Analisis dan Implementasi Algoritme Asimetris Dual Modulus RSA (DM-RSA) pada Aplikasi Chat. *RESTI (Rekayasa Sistem Dan Teknologi Informasi)*, *5*(10), 768–773. https://doi.org/https://doi.org/10.29207/resti.v5i4.3297

Arief, A., & Saputra, R. (2016). *Implementasi Kriptografi Kunci Publik dengan Algoritme RSA-CRT pada Aplikasi Instant Messaging*. *3*(1), 46–54.

Iswari, N. M. S. (2017). Key generation algorithm design combination of RSA and ElGamal algorithm. *Proceedings of 2016 8th International Conference on Information Technology and Electrical Engineering: Empowering Technology for Better Future, ICITEE 2016*. https://doi.org/10.1109/ICITEED.2016.7863255

Meneses, F., Fuertes, W., Salvador, S., Flores, D., Aules, H., Castro, F., Torres, J., Miranda, A., & Nuela, D. (2016). RSA Encryption Algorithm Optimization to Improve Performance and Security Level of Network Messages. *IJCSNS International Journal of Computer Science and Network Security*, *16*(8), 55–62.

Munir, R. (2019). *Kriptografi*. Informatika Bandung.

Permatasari, S., Aminudin, A., & Arifianto, S. (2020). Modifikasi Enkripsi dan Dekripsi AES dengan Polybius Chiper dalam Pengamanan Data. *JRST (Jurnal Riset Sains Dan Teknologi)*, *4*(1), 41. https://doi.org/10.30595/jrst.v4i1.6208

Poulakis, D. (2020). An application of Euclidean algorithm in cryptanalysis of RSA. *Elemente Der Mathematik*, *75*(3), 114–120. https://doi.org/10.4171/em/411

Rezal, M., Ariffin, K., Abubakar, S. I., & Yunos, F. (2018). *New Cryptanalytic Attack on RSA Modulus N = pq Using Small Prime Difference New Cryptanalytic Attack on RSA Modulus N = pq Using Small Prime Difference Method*. *January 2019*. https://doi.org/10.3390/cryptography3010002

Rutkowski, E., & Houghten, S. (2020). Cryptanalysis of RSA : Integer Prime Factorization Using Genetic Algorithms. *2020 IEEE Congress on Evolutionary Computation (CEC)*. https://doi.org/10.1109/CEC48606.2020.9185728