TECHNOLOGIC
Technology and Engineering for Intelligent and Computational Systems
Vol.1, No.1, June (2026) 14-20

ANALISIS KINERJA ALGORITMA KRIPTOGRAFI CRAMER–SHOUP DALAM PROSES ENKRIPSI DAN DEKRIPSI FILE TEKS MENGGUNAKAN PYTHON

I. [1]Angga Dinata
Teknik Informatika, Fakultas Ilmu Komputer, Universitas Nahdlatul Ulama Sumatera Utara
E-mail: [1]anggadinata10@gmail.com

II. [2]Demonius Sarumaha
Teknik Informatika, Fakultas Ilmu Komputer, Universitas Nahdlatul Ulama Sumatera Utara
E-mail: [2]demonius213@gmail.com

III. [3]Hery Yusuf Simbolon
Teknik Informatika, Fakultas Ilmu Komputer, Universitas Nahdlatul Ulama Sumatera Utara
E-mail: [3]simbolonyusufheri@gmail.com

IV. [4]Muhadi M.Ilyas Gultom
Teknik Informatika, Fakultas Ilmu Komputer, Universitas Nahdlatul Ulama Sumatera Utara
E-mail: [4]yass.gt.mail@gmail.com

ABSTRACT

Data security has become a critical issue in the digital era due to the increasing exchange of sensitive information through electronic media. Cryptography provides an effective solution to ensure the confidentiality and integrity of digital data. This study aims to implement and evaluate the performance of the Cramer–Shoup public-key cryptographic algorithm for encrypting and decrypting text files using Python 3.9.7. The research employed an experimental approach involving key generation, encryption, decryption, and performance testing on plaintext files of different sizes. System performance was evaluated by measuring encryption and decryption execution times while verifying data integrity after the decryption process. The experimental results demonstrate that the proposed implementation successfully generates valid public and private keys and accurately restores the original plaintext without any loss or modification of characters. Performance analysis indicates that execution time increases as the file size grows, with encryption requiring more computational time than decryption due to the generation of random parameters and modular exponentiation during the encryption stage. The highest recorded encryption time for a 15-character text file was 0.054927 seconds, while the corresponding decryption time was only 0.004264 seconds. Despite this difference, the total processing time remained below one second, indicating that the implemented system is computationally efficient for securing small- to medium-sized text files. These findings confirm that the Cramer–Shoup algorithm provides reliable data confidentiality and integrity while maintaining satisfactory execution performance, making it a suitable public-key cryptographic method for text-based data protection.

Kata Kunci: *Cramer–Shoup, public-key cryptography, encryption, decryption, Python, text file security.*

INTRODUCTION

Perkembangan teknologi informasi telah meningkatkan intensitas pertukaran data digital pada berbagai sektor, seperti pendidikan, pemerintahan, bisnis, dan layanan publik. File teks masih menjadi salah satu media utama dalam penyimpanan maupun pertukaran informasi penting karena ukurannya yang relatif kecil, mudah diproses, serta kompatibel dengan berbagai sistem. Namun, meningkatnya aktivitas pertukaran data juga diikuti oleh berbagai ancaman keamanan, seperti penyadapan, pencurian, manipulasi, dan penyalahgunaan informasi oleh pihak yang tidak berwenang. Kondisi tersebut menunjukkan bahwa aspek kerahasiaan, integritas, dan keaslian data menjadi kebutuhan yang sangat penting dalam sistem informasi modern (Talha et al., 2019; Edy Soesanto et al., 2023). Oleh karena itu, diperlukan mekanisme keamanan yang mampu melindungi data selama proses penyimpanan maupun transmisi.

Salah satu teknologi yang banyak digunakan untuk menjaga keamanan informasi adalah kriptografi. Kriptografi bekerja dengan mengubah data asli (plaintext) menjadi data tersandi (ciphertext) sehingga hanya pihak yang memiliki kunci yang sah yang dapat mengembalikannya ke bentuk semula melalui proses dekripsi (Chandra, 2016). Seiring berkembangnya kebutuhan keamanan, algoritma kriptografi kunci publik terus mengalami penyempurnaan. Algoritma Cramer–Shoup merupakan salah satu algoritma kriptografi asimetris yang dikembangkan sebagai penyempurnaan dari algoritma ElGamal dengan memberikan jaminan keamanan terhadap adaptive chosen ciphertext attack (CCA). Keunggulan tersebut menjadikan algoritma Cramer–Shoup sebagai salah satu metode enkripsi yang memiliki tingkat keamanan tinggi untuk melindungi informasi digital (NOURDINE & SOW, 2024).

Selain tingkat keamanan, aspek kinerja (performance) algoritma juga menjadi faktor penting dalam implementasi kriptografi. Algoritma dengan tingkat keamanan tinggi belum tentu memberikan efisiensi waktu yang baik ketika digunakan untuk mengenkripsi dan mendekripsi data. Penelitian sebelumnya lebih banyak berfokus pada pengembangan atau optimasi algoritma Cramer–Shoup, seperti optimasi proses komputasi menggunakan pustaka GMP maupun pengembangan skema Key Encapsulation Mechanism (KEM), sedangkan kajian mengenai implementasi praktis menggunakan Python dan analisis waktu eksekusi pada proses enkripsi serta dekripsi file teks masih relatif terbatas (Lafourcade et al., 2021; SOW & NOURDINE, 2023). Dengan demikian, diperlukan penelitian yang mengevaluasi performa algoritma tersebut dalam kondisi implementasi nyata menggunakan berbagai ukuran data.

Berdasarkan permasalahan tersebut, penelitian ini bertujuan mengimplementasikan algoritma kriptografi Cramer–Shoup menggunakan bahasa pemrograman Python serta menganalisis kinerjanya pada proses enkripsi dan dekripsi file teks. Evaluasi dilakukan dengan mengukur waktu eksekusi terhadap beberapa ukuran data untuk mengetahui hubungan antara ukuran file dan performa algoritma. Hasil penelitian diharapkan dapat memberikan gambaran mengenai efektivitas dan efisiensi algoritma Cramer–Shoup sebagai metode pengamanan data teks serta menjadi referensi bagi penelitian selanjutnya dalam pengembangan sistem keamanan informasi berbasis kriptografi kunci publik.

METHOD

Penelitian ini menggunakan metode eksperimen dengan pendekatan kuantitatif untuk menganalisis kinerja algoritma kriptografi Cramer–Shoup pada proses enkripsi dan dekripsi file teks. Tahapan penelitian dimulai dengan studi literatur mengenai kriptografi kunci publik, algoritma Cramer–Shoup, dan implementasinya menggunakan bahasa pemrograman Python. Selanjutnya dilakukan perancangan sistem, implementasi algoritma, pengujian terhadap beberapa ukuran file teks, serta analisis hasil berdasarkan waktu eksekusi proses enkripsi dan dekripsi. Pendekatan eksperimen dipilih karena mampu memberikan hasil pengukuran yang objektif terhadap performa algoritma dalam mengamankan data teks.

Implementasi sistem dilakukan menggunakan Python versi 3.9.7 dengan memanfaatkan pustaka hashlib sebagai fungsi hash, random untuk

pembangkitan bilangan acak, dan time untuk mengukur waktu eksekusi. Algoritma Cramer–Shoup diawali dengan proses pembangkitan pasangan kunci menggunakan bilangan prima (q), dua generator (g_1) dan (g_2), serta lima bilangan acak (x_1), (x_2), (y_1), (y_2), dan (z). Berdasarkan parameter tersebut dibentuk komponen kunci publik menggunakan Persamaan (1)–(3).

$$c = g_1^{x_1} \cdot g_2^{x_2} \bmod q \tag{1}$$

$$d = g_1^{y_1} \cdot g_2^{y_2} \bmod q \tag{2}$$

$$h = g_1^{z} \bmod q \tag{3}$$

Pasangan kunci yang dihasilkan terdiri atas kunci publik ((g_1,g_2,c,d,h,H)) dan kunci privat ((x_1,x_2,y_1,y_2,z)), yang selanjutnya digunakan dalam proses enkripsi dan dekripsi.

Proses enkripsi dilakukan dengan memilih bilangan acak (r) untuk setiap plaintext sehingga dihasilkan ciphertext yang berbeda meskipun pesan yang dienkripsi sama. Selanjutnya dihitung komponen ciphertext menggunakan Persamaan (4)–(8).

$$u_1 = g_1^{r} \bmod q \tag{4}$$

$$u_2 = g_2^{r} \bmod q \tag{5}$$

$$e = h^{r} \cdot m \bmod q \tag{6}$$

$$a = H(u_1, u_2, e) \tag{7}$$

$$v = c^{r} \cdot d^{ra} \bmod q \tag{8}$$

Hasil proses enkripsi berupa ciphertext ((u_{1},u_{2},e,v)) yang dikirimkan kepada penerima. Adanya nilai acak (r) pada setiap proses enkripsi memberikan sifat probabilistik sehingga meningkatkan keamanan algoritma terhadap berbagai bentuk serangan kriptografi.

Pada proses dekripsi, sistem terlebih dahulu memverifikasi keaslian ciphertext menggunakan kunci privat. Jika hasil verifikasi valid, maka plaintext dikembalikan menggunakan Persamaan (9).

$$m = e \cdot (u_1^{z})^{-1} \bmod q] \tag{9}$$

Hasil dekripsi kemudian dibandingkan dengan plaintext awal untuk memastikan bahwa tidak terjadi perubahan informasi selama proses enkripsi maupun dekripsi sehingga integritas data tetap terjaga.
Pengujian dilakukan menggunakan file teks (.txt) dengan tiga variasi ukuran data, yaitu 5, 10, dan 15 karakter. Setiap data diuji melalui proses enkripsi dan dekripsi menggunakan pasangan kunci yang dibangkitkan secara otomatis oleh sistem. Waktu eksekusi masing-masing proses diukur menggunakan modul time pada Python, kemudian hasilnya dianalisis secara deskriptif dengan membandingkan waktu enkripsi dan dekripsi pada setiap ukuran data. Analisis ini bertujuan untuk mengetahui pengaruh ukuran file terhadap performa algoritma Cramer–Shoup serta mengevaluasi efisiensi implementasinya dalam mengamankan file teks.

RESULT AND DISCUSSION

Hasil Implementasi Sistem

Implementasi algoritma Cramer–Shoup berhasil direalisasikan menggunakan bahasa pemrograman Python 3.9.7. Sistem yang dikembangkan mampu menghasilkan pasangan public key dan private key, melakukan proses enkripsi terhadap file teks (.txt), serta mengembalikan ciphertext menjadi plaintext melalui proses dekripsi. Selain itu, sistem juga mampu menampilkan waktu eksekusi pada setiap proses sehingga kinerja algoritma dapat dianalisis secara kuantitatif.
Tahap pertama merupakan proses pembangkitan pasangan kunci (key generation). Sistem menghasilkan parameter kriptografi secara otomatis yang terdiri atas nilai generator, bilangan prima, serta komponen kunci publik dan kunci privat yang akan digunakan pada proses enkripsi maupun dekripsi.

```
=== PROSES PEMBANGKITAN KUNCI ===
Private key:
 x1=1613099383
 x2=115606462
 y1=872967924
 y2=315847868
 z=459927452
Public key:
 c=1570340424
 d=398683577
 h=1448711122
 q=1764302801
 g1=1447978010
 g2=434392889
```

Figure 1. Hasil Pembangkitan Public Key dan Private Key

Setelah pasangan kunci berhasil dibangkitkan, sistem melakukan proses enkripsi terhadap file teks menggunakan public key. Hasil enkripsi menunjukkan bahwa seluruh karakter pada plaintext berhasil diubah menjadi ciphertext sehingga isi pesan tidak dapat dibaca secara langsung tanpa proses dekripsi.

```
r fix/Cramer-shoup cryptosystem.py"
o -> (452847596, 214078499, 698747840)
l -> (1704197235, 380934179, 1012839971)
o -> (1349158137, 892461964, 1087194880)
r -> (8530404, 439337197, 1175965785)
  -> (232061454, 643801723, 177370996)
s -> (683290434, 991292004, 1154417323)
i -> (1583097458, 906890804, 428065715)
t -> (34872459, 630144973, 1173725677)
  -> (329272999, 1245444772, 1757956405)
a -> (1061028794, 278651044, 1544990544)
m -> (1666443614, 276441921, 881031229)
e -> (1404796361, 1596224946, 1707813095)
t -> (772152196, 549029926, 149102483)
, -> (1426949051, 40645371, 854530460)
  -> (682377027, 1408593579, 1528698954)
c -> (366577654, 439585035, 1136511921)
o -> (905460143, 1079973039, 1725464503)
n -> (35824486, 1719199623, 1494725171)
s -> (1006026452, 242745709, 822988839)
e -> (1681586415, 1432592054, 754802047)
c -> (1283598223, 1157826671, 1734370078)
t -> (1033118337, 1464788893, 611756049)
e -> (1209278448, 1126548565, 931419574)
t -> (265727309, 632797861, 693227095)
u -> (563226905, 1197280007, 21392418)
e -> (837108068, 1567564795, 1749477687)
r -> (418188678, 940494715, 245233928)
  -> (1242374702, 1462484769, 1362883994)
a -> (470650153, 554760426, 1494817694)
d -> (244053279, 1741706057, 1353108302)
i -> (1510632885, 1577908585, 609899547)
p -> (485703798, 917682416, 228777307)
i -> (1712545798, 587934296, 7469643)
s -> (550689487, 449698667, 316674437)
c -> (634383823, 1621732292, 1148100468)
i -> (19196874, 101197004, 799105031)
n -> (1138200681, 700289250, 1203299464)
g -> (518405808, 1110025750, 1285997907)
  -> (816145607, 1026549336, 4295054)
e -> (1382755442, 39895977, 140592266)
l -> (908832798, 470813382, 834572293)
i -> (1760493194, 446725451, 1629621542)
t -> (352168916, 807890815, 615482884)
. -> (1369076501, 1125716370, 84263471)
  -> (1368755243, 1063349319, 746611748)
A -> (158531870, 500183, 145253977)
e -> (1654918357, 188715132, 1709177525)
n -> (1235939369, 199771127, 465160973)
e -> (224115019, 86636594, 971948887)
a -> (534396471, 1088610340, 1658661053)
n -> (1493580965, 1704565108, 740781120)
  -> (82004235, 593031457, 1328738872)
c -> (1443603831, 282852342, 1724481212)
o -> (155569456, 965735198, 617847270)
m -> (1543391103, 1364879883, 1704400830)
m -> (1569689625, 1359823553, 1336311725)
o -> (733433849, 859786070, 1691622622)
d -> (1482895233, 1328614184, 1067886816)
o -> (1203077310, 917678981, 1444659930)

 Hasil Enkripsi / Ciphertext = °¡ù9J±·3'¶W"
W½@ß¹¼æ¾ÞàÙ

Waktu Enkripsi: 0.009335 detik
```

Figure 2. Hasil Proses Enkripsi File Teks

Selanjutnya dilakukan proses dekripsi menggunakan private key. Hasil pengujian menunjukkan bahwa ciphertext berhasil dikembalikan menjadi plaintext tanpa perubahan karakter, sehingga integritas data tetap terjaga setelah melalui proses enkripsi dan dekripsi.

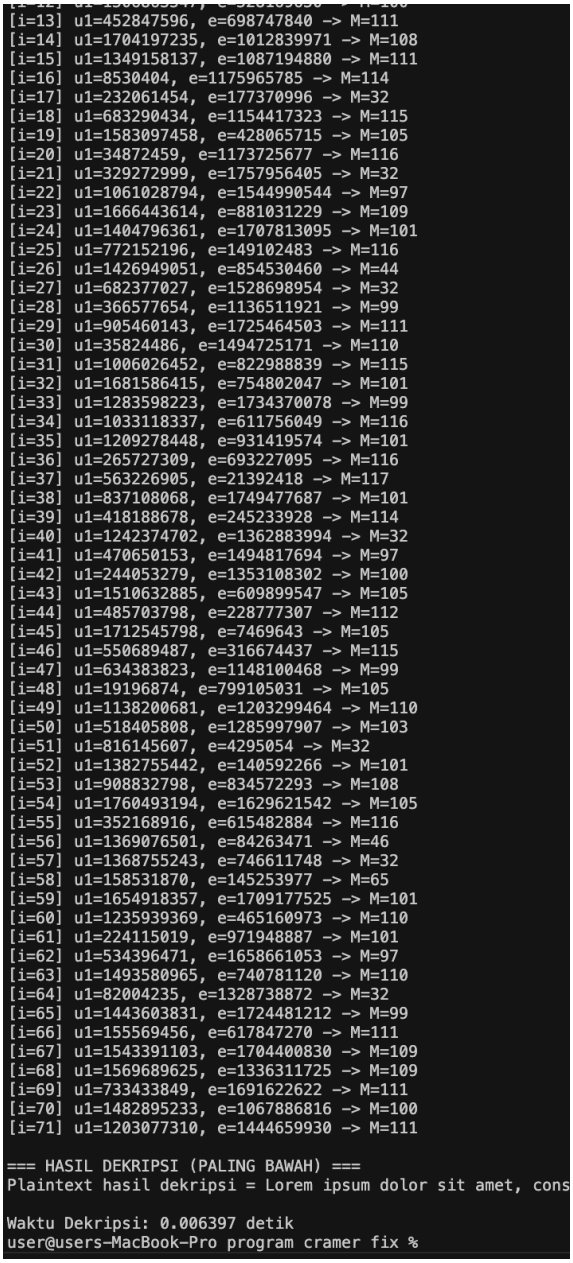

Figure 3. Hasil Proses Dekripsi File Teks

Keberhasilan proses dekripsi membuktikan bahwa implementasi algoritma Cramer–Shoup telah berjalan sesuai dengan mekanisme kriptografi kunci publik, yaitu hanya pihak yang memiliki private key yang dapat mengembalikan ciphertext menjadi informasi asli.

Hasil Pengujian Kinerja

Pengujian dilakukan menggunakan file teks (.txt) dengan tiga variasi ukuran data, yaitu 5, 10, dan 15 karakter. Parameter yang diamati meliputi waktu eksekusi proses enkripsi dan dekripsi. Rata-rata hasil pengujian disajikan pada Table 1.

Table 1. Result

No	Ukuran File	Waktu Enkripsi
1	5	0.002498
2	10	0.009335
3	15	0.054927

Berdasarkan Tabel 1, terlihat bahwa waktu eksekusi algoritma meningkat seiring bertambahnya ukuran data yang diproses. Pada file dengan panjang 5 karakter, proses enkripsi membutuhkan waktu sebesar 0,002498 detik, sedangkan proses dekripsi hanya 0,000467 detik. Ketika ukuran file meningkat menjadi 10 karakter, waktu enkripsi bertambah menjadi 0,009335 detik, sedangkan waktu dekripsi tercatat 0,006397 detik. Pengujian pada file berukuran 15 karakter menunjukkan waktu enkripsi tertinggi, yaitu 0,054927 detik, sedangkan proses dekripsi hanya memerlukan 0,004264 detik. Hasil tersebut menunjukkan bahwa peningkatan ukuran data berpengaruh terhadap waktu komputasi algoritma, terutama pada proses enkripsi.

Pembahasan

Hasil implementasi menunjukkan bahwa algoritma Cramer–Shoup berhasil diimplementasikan menggunakan Python 3.9.7 dan mampu menjalankan seluruh tahapan utama, yaitu pembangkitan pasangan kunci, enkripsi, serta dekripsi. Seluruh plaintext yang diuji dapat dikembalikan menjadi bentuk semula tanpa kehilangan maupun perubahan karakter setelah proses dekripsi. Hal ini membuktikan bahwa algoritma mampu menjaga kerahasiaan sekaligus integritas data selama proses pengamanan.

Dari sisi performa, hasil pengujian memperlihatkan bahwa waktu eksekusi meningkat seiring bertambahnya ukuran data yang diproses. Kondisi tersebut terjadi karena setiap karakter pada plaintext

harus melalui serangkaian operasi aritmetika modular dan perpangkatan modular (modular exponentiation) yang menjadi inti algoritma Cramer–Shoup. Semakin banyak karakter yang diproses, semakin besar pula jumlah operasi matematika yang harus dilakukan sehingga waktu komputasi meningkat.

Hasil penelitian juga menunjukkan bahwa proses enkripsi membutuhkan waktu lebih lama dibandingkan proses dekripsi. Hal ini disebabkan pada tahap enkripsi sistem harus membangkitkan bilangan acak (random number) untuk setiap proses, kemudian menghitung komponen ciphertext berupa (u_1), (u_2), (e), dan (v). Sebaliknya, proses dekripsi hanya memanfaatkan pasangan kunci privat yang telah tersedia sehingga jumlah komputasi yang dilakukan relatif lebih sedikit. Temuan ini sejalan dengan karakteristik algoritma Cramer–Shoup yang memiliki beban komputasi lebih besar pada tahap enkripsi dibandingkan tahap dekripsi.

Meskipun waktu enkripsi meningkat pada ukuran data yang lebih besar, seluruh proses enkripsi dan dekripsi masih dapat diselesaikan dalam waktu kurang dari satu detik. Hal tersebut menunjukkan bahwa implementasi algoritma Cramer–Shoup menggunakan Python cukup efisien untuk mengamankan file teks berukuran kecil hingga menengah. Selain memberikan tingkat keamanan yang tinggi melalui mekanisme kriptografi kunci publik, algoritma ini juga mampu mempertahankan performa yang baik sehingga layak diterapkan pada aplikasi pengamanan dokumen teks.

ACKNOWLEDGMENT

Penulis mengucapkan puji dan syukur kepada Tuhan Yang Maha Esa atas rahmat, karunia, dan penyertaan-Nya sehingga penelitian ini dapat diselesaikan dengan baik. Penulis juga menyampaikan terima kasih kepada kedua orang tua dan seluruh keluarga atas doa, dukungan, motivasi, serta semangat yang diberikan selama proses penyusunan penelitian ini.

Ucapan terima kasih juga disampaikan kepada dosen pembimbing, dosen penguji, serta seluruh sivitas akademika Program Studi Teknik Informatika Fakultas Teknik Universitas Islam Sumatera Utara yang telah memberikan bimbingan, masukan, dan fasilitas selama pelaksanaan penelitian. Selain itu, penulis mengapresiasi semua pihak yang telah memberikan bantuan, baik secara langsung maupun tidak langsung, sehingga penelitian mengenai implementasi algoritma kriptografi Cramer–Shoup untuk enkripsi dan dekripsi file teks ini dapat diselesaikan dengan baik.

REFERENCES

[1] Adawiyah Ritonga, & Yahfizham Yahfizham. (2023). Studi Literatur Perbandingan Bahasa Pemrograman C++ dan Bahasa Pemrograman Python pada Algoritma Pemrograman. Jurnal Teknik Informatika Dan Teknologi Informasi, 3(3), 56–63. https://doi.org/10.55606/jutiti.v3i3.2863

[2] AGUNG WIDYANTO. (n.d.). IMPLEMENTASI KRIPTOGRAFI TEKS MENGGUNAKAN RSA AGUNG WIDYANTO. In Community Service Article (COMERS) e-ISSN.

[3] Bahri, R., Budiman, M., & Nasution, B. (2024). Sign-Then-Encrypt Scheme with Cramer-Shoup Cryptosystem and Dissanayake Digital Signature. 131–138. https://doi.org/10.5220/0012444900003848

[4] Buyung Solihin Hasugian. (n.d.). 18.269-522-1-SM.

[5] Chandra. (2016). KEAMANAN DATA DENGAN METODE KRIPTOGRAFI KUNCI PUBLIK. In Jurnal TIMES (Issue 2).

[6] Chowdhury, A. (2013). A NEW TECHNIQUE TO ENCRYPT A CODE WORD. In International Journal of Students Research in Technology & Management (Vol. 1, Issue 05). www.giapjournals.com/ijsrtm/

[7] Da, N. T., & Thanh, H. T. (2021). Enhancing the Time Performance of Encrypting and Decrypting Large Tabular Data. Applied Artificial Intelligence, 35(15), 1746–1754. https://doi.org/10.1080/08839514.2021.1991661

[8] Dwi Nurcahya, S. (2022). Implementasi Aplikasi Kriptografi Metode Kode Geser Berbasis Java. Jurnal Nasional Komputasi Dan Teknologi Informasi, 5(4).

[9] Edy Soesanto, Achmad Romadhon, Bima Dwi Mardika, & Moch Fahmi Setiawan. (2023). Analisis dan Peningkatan Keamanan Cyber: Studi Kasus Ancaman dan Solusi dalam Lingkungan Digital Untuk Mengamankan Objek Vital dan File. Sammajiva: Jurnal Penelitian Bisnis Dan Manajemen, 1(2), 172–191. https://doi.org/10.47861/sammajiva.v1i2.226

[10] Ghaiyur Hayat, F., & B.N., M. (2022). Applying Encryption and Decryption Algorithm for Data Security in Cloud. INTERNATIONAL JOURNAL OF ENGINEERING AND MODERN TECHNOLOGY, 8(2), 16–23. https://doi.org/10.56201/ijemt.v8.no2.2022.pg16.23

[11] He, J., Ye, Q., Yang, Z., Wang, S., & Wang, J. (2025). A compact public key encryption with equality test for lattice in cloud computing. Scientific Reports, 15(1). https://doi.org/10.1038/s41598-025-12018-2

[12] Khan, M., & Munir, N. (2025). Cryptanalysis of image confidentiality scheme based on hybrid chaotic maps. Journal of King Saud University - Computer and Information Sciences, 37(6). https://doi.org/10.1007/s44443-025-00114-2

[13] Lafourcade, P., Robert, L., & Sow, D. (2021). Fast cramer-shoup cryptosystem. Proceedings of the 18th International Conference on Security and Cryptography, SECRYPT 2021, 766–771. https://doi.org/10.5220/0010580607660771

[14] Maulana, F., Wardhana, I. G. A. W., & Switrayni, N. W. (2019). Ekivalensi Ideal Hampir Prima dan Ideal Prima pada Bilangan Bulat Gauss. EIGEN MATHEMATICS JOURNAL, 1–5. https://doi.org/10.29303/emj.v1i1.29

[15] Mohammad. (n.d.). 5.Makalah2_Kripto_IF4020_2015_003.

[16] Nanda Amalya. (n.d.). IMPLEMENTASI KRIPTOGRAFI TEKS MENGGUNAKAN RSA AGUNG WIDYANTO. In Community Service Article (COMERS) e-ISSN.

[17] Naser, S. M. (2021). CRYPTOGRAPHY: FROM THE ANCIENT HISTORY TO NOW, IT'S APPLICATIONS AND A NEW COMPLETE NUMERICAL MODEL. In International Journal of Mathematics and Statistics Studies (Vol. 9, Issue 3). www.ea-journals.org

[18] NOURDINE, B., & SOW, D. (2024). DESIGN OF FAST CRAMER-SHOUP SCHEME INTO ELLIPTIC CURVE CRYPTOSYSTEM. JP Journal of Algebra, Number Theory and Applications, 63(2), 169–184. https://doi.org/10.17654/0972555524010

[19] Sarumaha, D., Mohammad Andri Budiman, & Muhammad Zarlis. (2023). Performance Analysis of Hybrid Cryptographic Algorithms

Rabbit Stream and Enhanced Dual RSA. Data Science: Journal of Computing and Applied Informatics, 7(1), 35–43. https://doi.org/10.32734/jocai.v7.i1-10483

[20] Shacham, H. (n.d.-a). A Cramer-Shoup Encryption Scheme from the Linear Assumption and from Progressively Weaker Linear Variants.

[21] Shacham, H. (n.d.-b). A Cramer-Shoup Encryption Scheme from the Linear Assumption and from Progressively Weaker Linear Variants.

[22] SOW, D., & NOURDINE, B. (2023). A NEW FAST CRAMER-SHOUP KEM. Universal Journal of Mathematics and Mathematical Sciences, 18, 85–97. https://doi.org/10.17654/2277141723006

[23] Talha, M., El Kalam, A. A., & Elmarzouqi, N. (2019). Big data: Trade-off between data quality and data security. Procedia Computer Science, 151, 916–922. https://doi.org/10.1016/j.procs.2019.04.127

[24] Ulrick, S. (n.d.). Implementation of Cramer-Shoup Cryptosystem.

[25]