International Journal of Science and Research (IJSR)
ISSN: 2319-7064
SJIF (2022): 7.942

Comparative Analysis of Open-Source Encryption Tools: GnuPG vs. VeraCrypt

Ravi Kumar Singh[1], Vikalp[2], Dushyant Sharma[3], Aditya Pandey[4]

Scholar, Department of Computer Science and Engineering Chandigarh University, Mohali, India
Email: *21BCS1896[at]cuchd.in*

Scholar, Department of Computer Science and Engineering Chandigarh University, Mohali, India
Email: *21BCS1983[at]cuchd.in*

Professor, Department of Computer Science and Engineering Chandigarh University, Mohali, India
Email: *dushyant.e1646[at]gmail.com*

Scholar, Department of Computer Science and Engineering Chandigarh University, Mohali, India
Email: *21BCS2259[at]cuchd.in*

Abstract: ***This paper presents a comparative analysis of open- source encryption tools, specifically GnuPG and VeraCrypt, focusing on their performance, usability, and security features. Our objective is to determine their effectiveness for different data types and use cases by testing various parameters including encryption/decryption time and resource utilization.***

Keywords: Cryptography, Encryption, Decryption, Security, GnuPG, VeraCrypt

1. Introduction

Cryptography involves the process of transforming plaintext (normal, readable text) into ciphertext, a method known as encryption, and subsequently converting it back to plaintext, known as decryption. Cryptographic algorithms can be categorized in various ways, with the most common types being Secret Key Cryptography also referred to as Symmetric Key Cryptography and Public Key Cryptography, also known as Asymmetric Key Cryptography [1]. It is a mathematical science focused on encoding and decoding data, allowing secure storage and transformation of critical information across net- works or channels, which remain unreadable to anyone except the intended recipient [2]. Currently, researchers across various security fields, particularly in authentication and key exchange, are developing diverse protocols to enhance and secure the Internet of Things (IoT) environment and implement this approach effectively [3]. This paper compares two widely used open-source encryption tools: Gnu Privacy Guard (GnuPG) and VeraCrypt. GnuPG is an encryption standard known for securing files and communications using public/private key cryptography, we will be comparing it with AES(VeraCrypt). The study will demonstrate that while both tools excel in different contexts, their strengths and weaknesses make them suitable for distinct use cases, highlighted through multiple test case studies. The performance, usability, and security features will be compared through hands-on testing. Both tools will be evaluated under different scenarios, including encrypting files of varying sizes, which include text files, and media like 'MP4' and JPG images. We will be measuring encryption/decryption speed, and system resource usage. The detailed test cases provide an in-depth view of how these tools operate in real- world environments.

2. Proposed Methodology

The experiment has been conducted on Google Colab notebooks comparing the performance of GnuPG and Fernet encryption algorithms and evaluating the effectiveness, efficiency, and usability across various file types and sizes. The analysis focuses on factors such as encryption time, file size changes, and ease of use.

3. Encryption Algorithms Used

a) GnuPG

It is an Asymmetric Key Encryption that offers secure communication using digital signatures, public keys, and private key pairs for authenticity and integrity of data. It also has 'key management' support for key generation, signing, revocation, and expiration for long-term security.

b) Fernet (VeraCrypt)

It is Symmetric key encryption, Symmetric encryption also known as conventional or single-key encryption was the primary form of encryption before the advent of public key encryption in the late 1970s. It has been employed historically by various individuals and organizations, including modern diplomatic, military, and commercial entities, for secure communication [4]. It encrypts volumes and entire disks, securing large datasets. Plausible Deniability: Features hidden volumes that allow users to deny the existence of encrypted data, protecting against coercion. Password-based encryption simplifies key management with user-friendly password-protected volumes and supports key files for additional security. Both tools provide robust security features; however, GnuPG's focus on secure communications contrasts with VeraCrypt's full- disk encryption capabilities. In terms of usability, VeraCrypt is easier for non-technical users, while GnuPG offers greater flexibility and control, especially for those familiar with

encryption workflows.

4. Experimental Setup

In the Google COLAB notebook, the algorithms are implemented using Python, with encryption methods integrated into a user-friendly web interface using the STREAMLIT webpage framework.

5. Evaluation Parameters

The analysis is based on four key parameters: File Size (KB): The input file size before encryption is recorded. Encryption Time (s): The time taken to encrypt the file using GnuPG and Fernet is measured. Encrypted File Size (KB): The size of the output file after encryption is recorded for comparison. Decryption Time (s): The time taken to decrypt the file using the respective decryption methods.

6. Encryption and Decryption Procedures

a) GnuPG

Encryption: Files are encrypted using GnuPG's symmetric encryption with the AES-256 algorithm. The encryption is secured by a user-provided passphrase. Decryption: Encrypted files are decrypted using the same passphrase for authentication, ensuring confidentiality.

b) Fernet (VeraCrypt)

Encryption: Files are encrypted using Fernet, which generates a key stored in a local file. The key is loaded during each encryption operation to ensure consistency. Decryption: Files are decrypted using the same key, allowing users to recover the original content.

7. Data Collection and Analysis

File types tested include text, audio, video, and image files. Files are processed in batches to simulate real-world applications. For each file, the time taken for encryption and decryption is recorded. The file size before and after encryption is also noted to compare overhead between methods.

8. Visualization and Results Comparison Graphs

The performance data is visualized using line charts, plotting encryption time, decryption time, and file sizes. The charts provide a clear comparison between GnuPG and Fernet, showing the relative efficiency of each method. Downloadable Outputs: After encryption or decryption, the processed files are made available for download as a ZIP archive.

9. Test Case Studies

We conducted test cases with five file types, noting encryption and decryption speed for each.

	File Name	File Size (KB)	Encryption Time (s)	Encrypted File (KB)
0	test1.jpg	119.9971	.0013	160.0742
1	test2.mp4	23,111.1191	0.2106	30,814.9102
2	test3.txt	47.3574	0.0005	63.2227
3	test4.png	2,230.6143	0.0179	2,974.2422
4	test5.pdf	409.5938	0.0032	546.2227

Figure 1: Encryption using Fernet Algorithm

	File Name	File Size (KB)	Decryption Time (s)	Decrypted File Size (KB)
0	test1.jpg. encrypted	160.0742	.0014	119.9971
1	test2.mp4.encrypted	30,814.9102	0.2222	23,111.1191
2	test3.txt.encrypted	63.2227	0.0006	47.3574
3	test4.png.encrypted	2,974.2422	0.0211	2,230.6143
4	test5.pdf.encrypted	546.2227	0.0035	409.5938

Figure 2: Decryption using Fernet Algorithm

	File Name	File Size (KB)	Encryption Time (s)	Encrypted File (KB)
0	test1.jpg	119.9971	0.369	154.9229
1	test2.mp4	23,111.1191	1.7952	31,334.249
2	test3.txt	47.3574	0.3627	1.0518
3	test4.png	2,230.6143	0.4885	3,019.6279
4	test5.pdf	409.5938	0.3714	539.3213

Figure 3: Encryption using GnuPG

	File Name	File Size (KB)	Decryption Time (s)	Decrypted File Size (KB)
0	test1.jpg. encrypted	154.9229	0.6594	119.9971
1	test2.mp4.encrypted	31,334.249	1.7817	23,111.1191
2	test3.txt.encrypted	1.0518	0.6567	47.3574
3	test4.png.encrypted	3,019.6279	0.6072	2,230.6143
4	test5.pdf.encrypted	539.3213	0.3593	409.5938

Figure 4: Decryption using GnuPG

10. Results

Both encryption methods yielded distinct performance results across file types, showcasing unique strengths and optimal use cases.

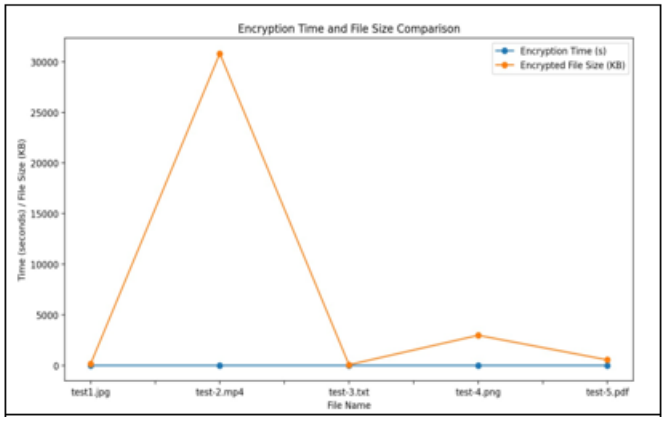

Figure 5: Encryption using Fernet Algorithm

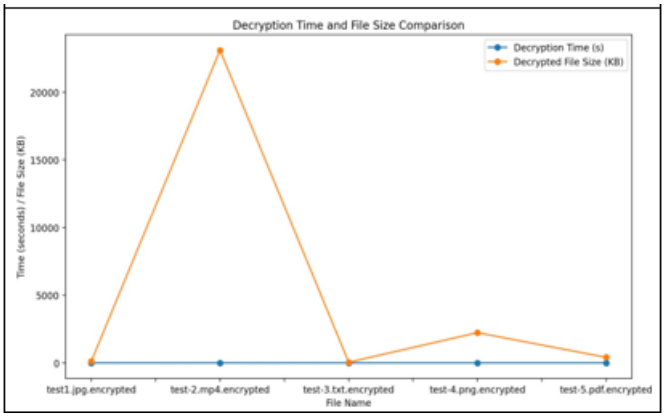

Figure 6: Decryption using Fernet Algorithm

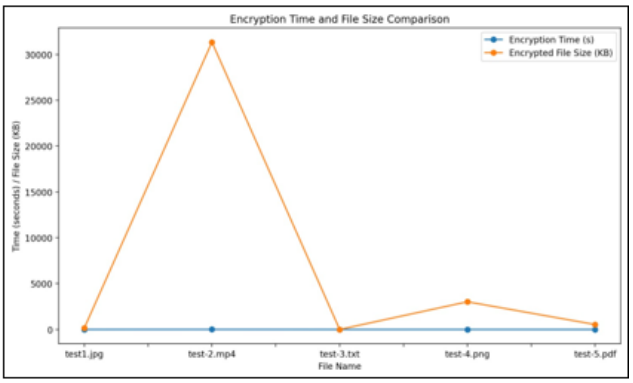

Figure 7: Encryption using GnuPG

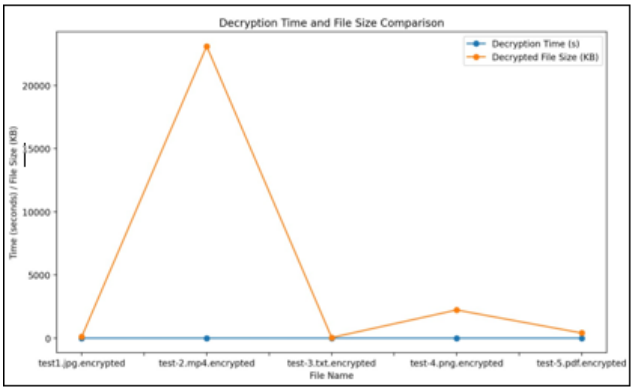

Figure 8: Decryption using GnuPG

11. Performance Analysis

We compared both encryption algorithms using a small dataset to evaluate performance [5]. Our main focus was on a comparative analysis of open-source encryption tools, GnuPG and VeraCrypt, and utilized a codebase designed for batch encryption and decryption. Tests on smaller data samples highlighted several performance metrics, including encryption/decryption speed, resource utilization, and data handling efficiency.

12. Conclusion

In the above graphs and tables, we can see that for different Encryption Algorithms and file types and sizes, we are getting different file sizes after encryption and decryption the speed of encryption and decryption is also different for various algorithms and file types. The GnuPG and VeraCrypt (AES/Fernet) provide robust encryption solutions but cater to other security needs. For most use cases, the choice between these tools depends on the nature of the data being protected GnuPG for secure file exchanges and signatures, and VeraCrypt for safeguarding entire systems or external storage. The performance tests indicate that while both the algorithm tools have given different results when introduced to other file types for encryption and decryption, both have unique security features and complexity. Analysis is based on performance (time and file size), and recommendations are made for which method might suit different use cases (e.g., small vs. large files, text vs. media files).

References

[1] N. Bhaskar, “Symmetric Key Cryptography Algorithm Using Comple- ment For Small Data Security,” International Journal of Engineering Research & Technology, vol. 2, no. 5, 2013.
[2] M. Tarik, S. Jabbehdari, S. J. F. Darvandi and A. shojaei,” Studying security protocol architecture based on cryptography algorithms,” IJISET- International Journal of Innovative Science, Engineering & Technology, vol. 2, no. 4, 2015.
[3] Y. Salami, V. Khajehvand and E. Zeinali,” arxiv.org,” 6 Dec 2022. [Online]. Available: https://arxiv.org/abs/2212.03308
[4] S. Sindhu and D. Sindhu, “Cryptographic Algorithms: Applications in Network Security,” International Journal of New Innovations in Engi- neering and Technology, vol. 7, no. 1, 2017.
[5] A. W. Soomro, N. A. I. Umar and N. Amin, “Secured Symmetric Key Cryptographic Algorithm for Small Amount of Data,” in 3rd International Conference on Computer & Emerging Technologies, 2013.