Koneru Lakshmaiah Education Foundation

(Deemed to be University)

DEPARTMENT OF EL&GE

Report on

Linux,Encryption, Self Hosting&Open Source Contributions

SUBMITTED BY:

ID Number	Name
2400100047	H S Shajiya Begum

UNDER THE GUIDANCE OF

Mr.Sripath Roy Koganti

Assistant Professor

KL UNIVERSITY

Green fields, Vaddeswaram – 522 502

Guntur Dt., AP, India.

Contents

1 Linux Distribution Used **2**

2 Encryption and GPG **4**

3 Sending Encrypted Email **6**

4 Privacy Tools from PRISM-Break **8**

4.1 DuckDuckGo 8

4.2 Brave Browser 8

4.3 Tutanota 9

4.4 Jitsi Meet 9

4.5 uBlock Origin 10

5 Open Source License Used **11**

6 Self-Hosted Server **12**

7 Open Source Contributions **15**

8 LinkedIn Activity **17**

1 Linux Distribution Used

Overview of Ubuntu 24.04.1 LTS

Ubuntu 24.04.1 LTS is a reliable and widely adopted Linux distribution released in 2024. I installed this version on my system to explore its interface, features, and performance. Ubuntu is well-known for its clean design, smooth functioning, and strong community support, making it suitable for beginners as well as experienced users.

Installation Experience

While installing Ubuntu, the process was mostly simple and user-friendly. The installer detected hardware properly and guided me through partitioning and user setup. However, I accidentally skipped the wireless configuration step during installation. Because of that, the Wi-Fi option didn't appear afterward. I tried troubleshooting it multiple times, but the issue persisted. As an alternative, I connected the laptop to the internet using USB tethering, which allowed me to install necessary updates and software.

Performance

Ubuntu performed smoothly on my device, with good memory optimization and quick booting. Essential applications like LibreOffice and Firefox were preinstalled, allowing me to start working immediately. Installing new tools through APT, Snap, or Flatpak was also very convenient.

Security and Stability

Being an LTS version, Ubuntu provides long-term security updates and stable performance. Built-in tools like UFW firewall, user permission control, and regular updates helped maintain a secure computing environment.

Conclusion

Overall, my experience with Ubuntu 24.04.1 LTS was positive. Apart from the Wi-Fi issue, everything worked efficiently, and using USB tethering enabled smooth operation. This experiment strengthened my understanding of Linux-based systems and open-source environments.

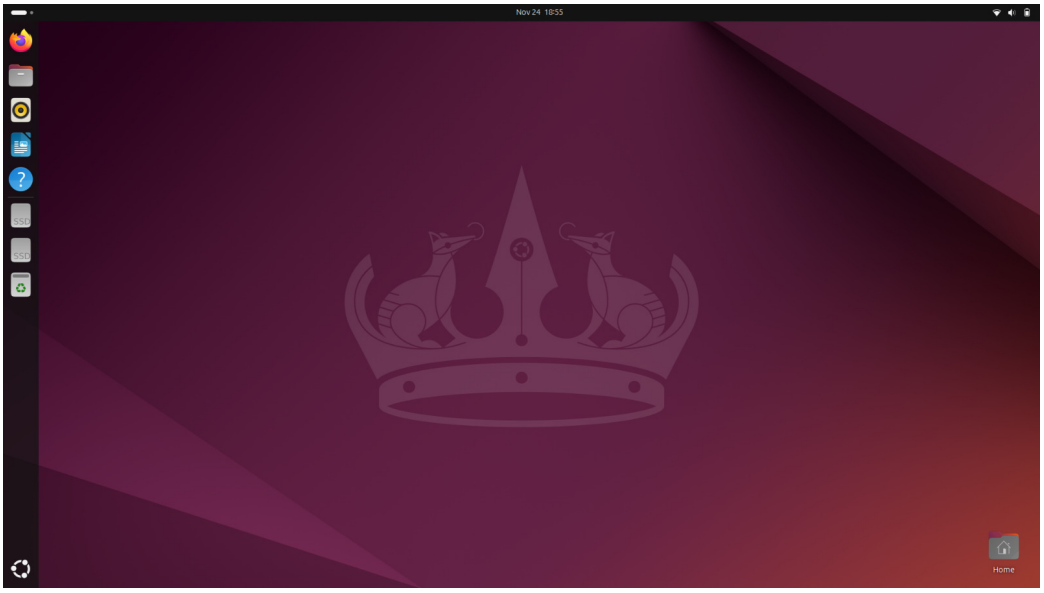

Figure 1.1: Successful Ubuntu Installation

2 Encryption and GPG

Encryption is a technique used to convert readable information into a protected format so that unauthorized users cannot access it. I learned how encryption helps in securing digital communication, personal data, and sensitive files.

GPG (GNU Privacy Guard) is an open-source tool based on the OpenPGP standard and uses public-key cryptography. Every user has two keys:

- **Public Key:** Shared with others so they can encrypt files for the user.
- **Private Key:** Kept confidential and used to decrypt files or sign documents.

Using GPG, I practiced encrypting/decrypting files, validating signatures, and understanding how integrity and authenticity are maintained in digital communication.

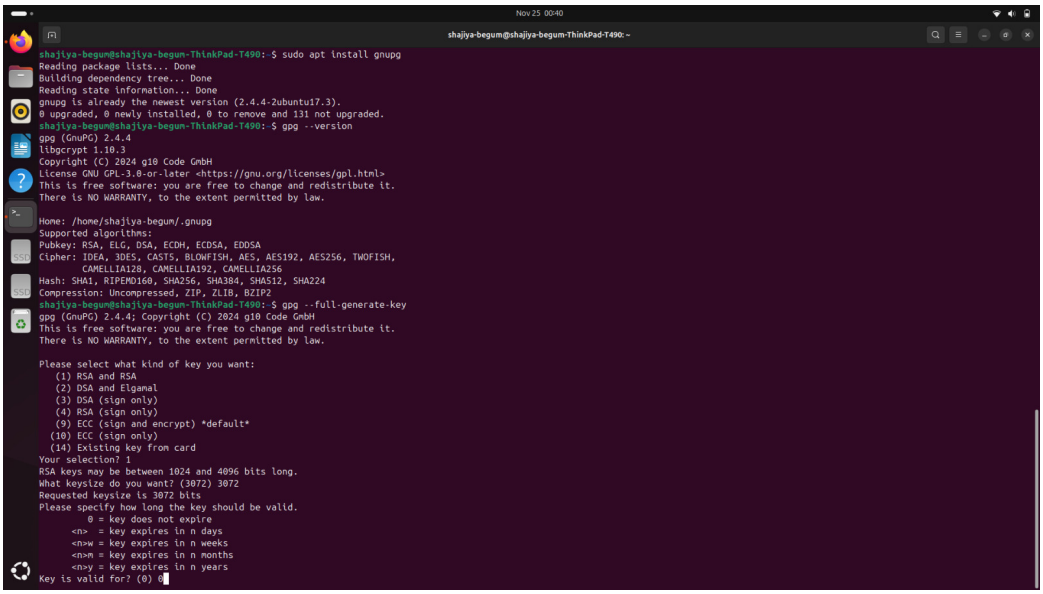

Figure 2.1: Encryption and GPG Illustration

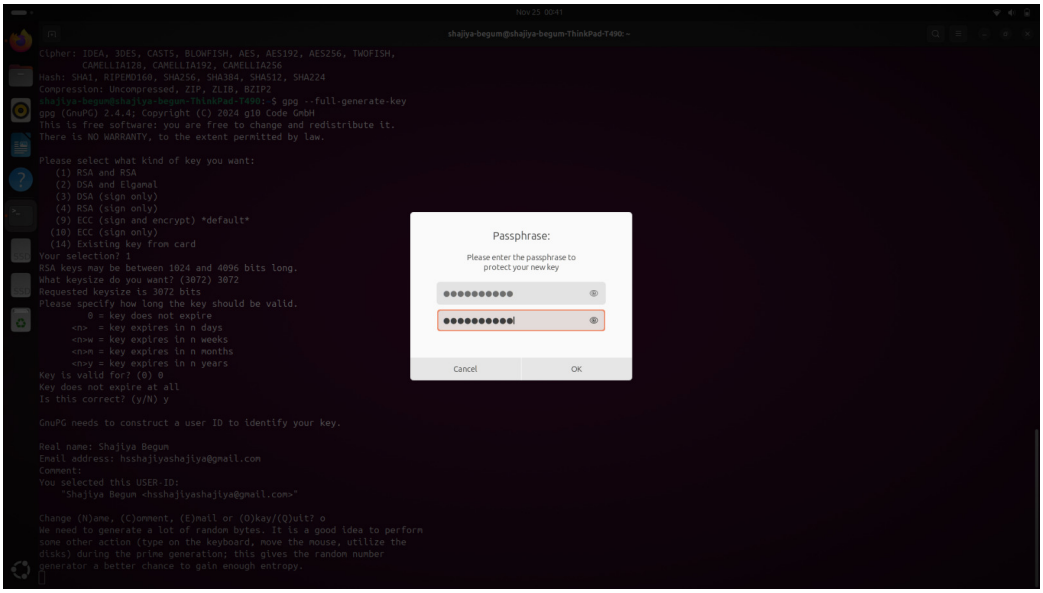

Figure 2.2: Encryption and GPG Illustration

3 Sending Encrypted Email

I learned how encrypted email ensures privacy during communication. Normally, email content can be intercepted easily, but by encrypting it with the recipient's public key, only the intended receiver can read it.

Using GPG, I imported the recipient's public key, encrypted my email message, and sent it securely. This process confirmed the importance of encryption in protecting sensitive information.

Benefits of encrypted email:

- Protects message confidentiality
- Ensures safe sharing of sensitive data
- Allows verification using digital signatures

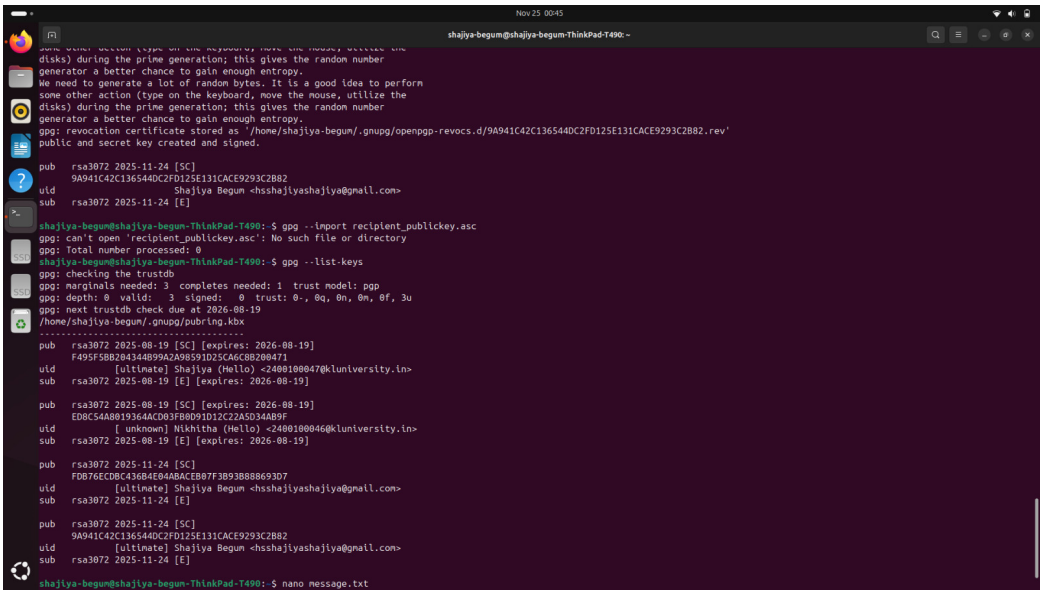

Figure 3.1: Sending Encrypted Email

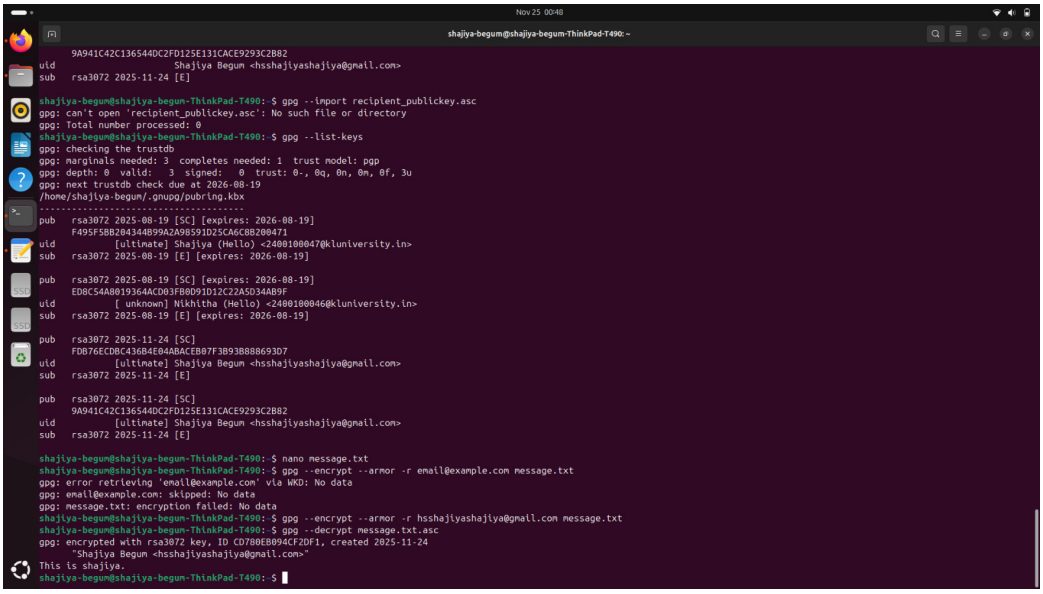

Figure 3.2: Sending Encrypted Email

4 Privacy Tools from PRISM-Break

PRISM-Break suggests privacy-focused and open-source alternatives to commonly used applications. I explored some widely recommended tools:

4.1 DuckDuckGo

A privacy-centric search engine that does not track user activity or store personal data.

Figure 4.1: DuckDuckGo

4.2 Brave Browser

A secure browser with built-in trackers and ads blocker, offering enhanced privacy protection.

Figure 4.2: Brave Browser

4.3 Tutanota

An encrypted email service that provides end-to-end security and open-source transparency.

Figure 4.3: Tutanota

4.4 Jitsi Meet

An open-source, privacy-friendly video conferencing tool with no account needed.

Figure 4.4: Jitsi Meet

4.5 uBlock Origin

A lightweight, open-source content blocker that prevents trackers, ads, and malicious scripts.

Figure 4.5: uBlock Origin

5 Open Source License Used

For my project **Daily Task Manager**, I selected the **MIT License**. It is simple, permissive, and allows others to freely use, modify, distribute, or integrate the software with minimal restrictions.

Why MIT License?

- Easy to understand and apply
- Allows modification and commercial usage
- Encourages learning and collaboration
- Widely adopted and developer-friendly

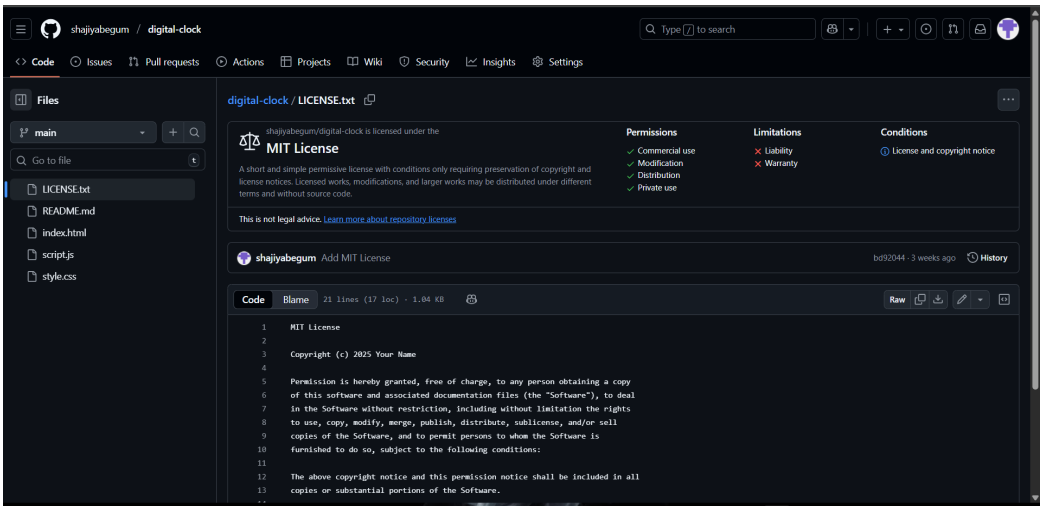

Figure 5.1: MIT License

6 Self-Hosted Server

About the Server

I self-hosted **Draw.io (diagrams.net)** on my system to use it offline. Draw.io is helpful for creating flowcharts, UML diagrams, network designs, and other visual content. Self-hosting ensures data privacy, offline access, and complete control over saved diagrams.

Installation Steps

The steps followed to self-host Draw.io are:

1. **Install Docker** If Docker is not installed:

   ```
   sudo apt update
   sudo apt install docker.io -y
   sudo systemctl enable --now docker
   ```

   Check installation:

   ```
   docker --version
   ```

2. **Pull the official Draw.io image**

   ```
   sudo docker pull jgraph/drawio
   ```

3. **Run Draw.io container** Run it on port 8080:

```
sudo docker run -d --name drawio \
    -p 8080:8080 \
    jgraph/drawio
```

4. **Run the Application Locally**

```
http://localhost:8080
```

After this setup, Draw.io is successfully self-hosted.

Localized (Hindi Translation)

इस परियोजना में, मैंने Draw.io (diagrams.net) को अपने सिस्टम पर ऑफलाइन उपयोग के लिए स्वयं होस्ट किया। Draw.io एक ओपन-सोर्स डायग्रामिंग टूल है जिसका उपयोग फ्लोचार्ट, नेटवर्क डायग्राम, और अन्य तकनीकी चित्र बनाने के लिए किया जाता है।

इसे स्वयं होस्ट करने का मुख्य लाभ यह है कि यह बिना इंटरनेट के पूरी तरह ऑफलाइन चलता है। इससे डेटा पूरी तरह सुरक्षित रहता है और किसी बाहरी सर्वर पर अपलोड नहीं होता। इस कारण यह शैक्षणिक कार्यों और गोपनीय दस्तावेज तैयार करने के लिए अत्यंत उपयोगी है।

Poster

Figure 6.1: Self-Hosted Draw.io Poster

7 Open Source Contributions

During my open-source journey, I created and submitted several pull requests that focused on implementing algorithms, building utility programs, and contributing meaningful code to different repositories. These contributions enhanced my understanding of coding standards, project structures, and collaborative development.

Pull Requests Submitted

- **IntroSort Implementation** — Added a complete implementation of the IntroSort algorithm, combining QuickSort, HeapSort, and Insertion Sort for optimized performance.
- **MaximumSubarray.c** — Submitted an efficient solution to compute the maximum subarray sum using standard algorithmic techniques.
- **IntegertoRoman.c** — Developed a clean and accurate program to convert integers into their Roman numeral representation.
- **GenerateParentheses.c** — Implemented a backtracking-based solution to generate all valid combinations of parentheses.
- **SimpleBankSystem.c** — Created a simple banking system program involving fundamental operations like deposit, withdrawal, and account balance handling.
- **RegularExpressionMatching.c** — Added an implementation for matching strings against patterns using dynamic programming logic.

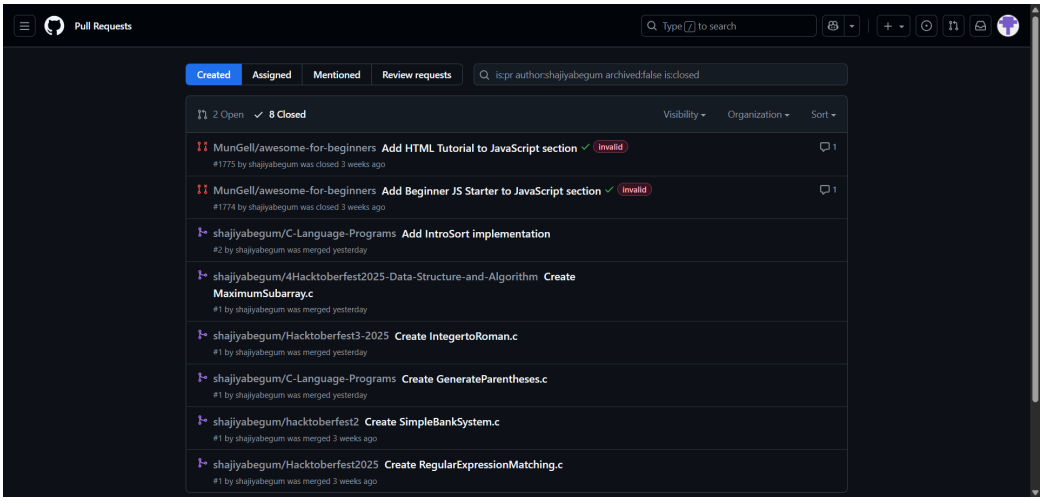

Figure 7.1: PRs Submitted and Merged

8 LinkedIn Activity

Throughout this project, I consistently documented my learning journey on LinkedIn. These posts covered key milestones such as self-hosting tools, contributing to open-source projects through pull requests, and writing technical blogs to share my understanding with a wider audience. This online documentation not only helped me track my progress but also allowed me to engage with the tech community and receive valuable feedback.

Post Links

Below are the three LinkedIn posts related to my work:

- **Self Hosting Post:**

  https://www.linkedin.com/posts/h-s-shajiya-begum-598653367_drawio-diagramsnet-ubuntu-activity-7390419755670654976-6bzR?utm_source=social_share_send&utm_medium=member_desktop_web&rcm=ACoAAFsK5IQBfMLSDGiCrTXVnuQvd9WrseHskFc

- **PR Merge Post:**

  https://www.linkedin.com/posts/h-s-shajiya-begum-598653367_opensource-github-coding-activity-7398793644540841985-sx-z?utm_source=social_share_send&utm_medium=member_desktop_web&rcm=ACoAAFsK5IQBfMLSDGiCrTXVnuQvd9WrseHskFc

- **Technical Blog Post:**

  https://www.linkedin.com/pulse/practical-journey-through-linux-open-source-h-s-shajiya-begum-itu3e

Conclusion

This project provided me with a strong foundation in Linux, security, and open-source technologies. Installing and exploring Ubuntu 24.04.1 LTS helped me understand how Linux systems function, how software is managed, and how common issues can be resolved practically. Even the Wi-Fi configuration issue gave me a chance to troubleshoot and learn alternative ways to connect and update the system.

Working with GPG encryption and sending encrypted emails allowed me to appreciate the importance of data privacy and secure communication. Through hands-on practice, I gained clarity on how public and private keys operate and why encryption is essential in protecting sensitive information.

Exploring privacy tools from PRISM-Break increased my awareness of how open-source applications can offer safer alternatives to mainstream software. These tools highlighted the role of privacy, anonymity, and user control in today's digital environment.

Self-hosting Draw.io using Docker was one of the most impactful parts of the project. It helped me understand containerization, local hosting, and maintaining tools without relying on cloud services. This experience also emphasized the value of data ownership and offline accessibility.

Finally, contributing to open-source repositories through multiple pull requests improved my coding confidence and problem-solving skills. It gave me real experience in writing clean programs, following standards, and collaborating through GitHub. Overall, this project strengthened both my technical abilities and my approach to open-source learning.