The Journal of Open Source Software

EQUALencrypt: An R package for encrypting and decrypting research data

Kurinchi Gurusamy [1¶]

1 University College London ¶ Corresponding author

DOI: 10.xxxxxx/draft

Software

- Review
- Repository
- Archive

Editor:

Submitted: 31 August 2025
Published: unpublished

Summary

Two shiny applications *EQUALencrypt - Encrypt and decrypt whole files* and *EQUALencrypt - Encrypt and decrypt columns of data* allow people with no coding skills to encrypt and share research data. Without the correct digital signatures and private keys, data encrypted using the applications cannot be decrypted, ensuring data integrity and security during data transfer between researchers.

Statement of need

Sharing research data will enable testing the data for reproducibility of analysis and secondary analyses (Kelly et al., 2024; Lvovs et al., 2025). These will increase the quality of research and decrease the costs for performing research (Kelly et al., 2024; Lvovs et al., 2025). However, when the research involves human participants, it is important that personal identifiable data are not shared publicly to meet the legal requirements (Lvovs et al., 2025).

R is a free software (R Foundation, 2025) with advanced statistical algorithms, data encryption and decryption, and digital signature insertion verification. However, considerable coding skills are necessary. Shiny applications allow researchers without web development skills coding skills to develop web-based applications in which others can interact with the data processed or analysed by the researchers (Shiny, 2025).

State of the field

As far as we are aware, there are no shiny applications that support encryption and decryption for researchers without coding skills, particularly following the approach followed for these shiny applications (please see below).

Shiny applications

Two shiny applications *EQUALencrypt - Encrypt and decrypt whole files* (Gurusamy, 2025c) and *EQUALencrypt - Encrypt and decrypt columns of data* (Gurusamy, 2025b) were created, which allow people with no coding skills to encrypt data.

Approach for data encryption and insertion of digital signature

EQUALencrypt - Encrypt and decrypt whole files (Gurusamy, 2025c) accepts whole files of any extension as input and uses openssl package (Ooms & Keyes, 2025) to encrypt and digitally sign the file. To encrypt and digitally sign the file, it generates an unique pair of private and

public RSA keys (4096 bits), encrypts the file using symmetric AES256 algorithm (32 bits for the key and 16 bits for initialization vector), encrypts the AES key using the asymmetric RSA keys, and includes padding according to PKCS #1 v2.0 specifications. It then inserts a digital signature using the SHA384 algorithm for the hash function.

EQUALencrypt - Encrypt and decrypt columns of data (Gurusamy, 2025b) accepts only csv files (with only ASCII characters) and up to 7 levels of access of columns. The columns which have not been selected at any access level will be unencrypted and will be available to people with any level of access. People with higher access level will also be able to view the columns that people with lower access level can view. For example, a person with access level 4 will be able to view the unencrypted columns and the columns that people with access levels 1 to 3 can view in addition to access level 4 that they belong to.

Allowing multiple levels of access means that the same encrypted data can be shared with different private keys provided for people with different levels of access, without having to prepare multiple files of data depending upon the nature of the data and the role of the researcher in the research project. The columns in each level of access are encrypted using the same approach as for *Encrypt and decrypt whole files* (Gurusamy, 2025c), except that a set of unique pair of private and public RSA keys (4096 bits) are generated for each level of access which contain at least one column.

Sharing the information publicly and privately

The publicly shareable information include the encrypted files, digital signature, and the public key, while the private keys must be shared only with people who have permission to decrypt the data. It is recommended that sharing the publicly shareable information and the private keys are performed in two different ways, for example, by depositing the publicly shareable information in a shared drive or a repository as appropriate and sharing the private keys by email, to decrease the risk of undesirable decryption.

Verification of signature and decryption

The user uploads the encrypted data, digital signature, and the public and private keys to decrypt data or columns of data (into the corresponding applications). If the encrypted data, digital signature, and the keys match, the data is decrypted. This ensures data integrity and security during data transfer between researchers.

Testing of functionality

Testing of *EQUALencrypt - Encrypt and decrypt whole files* (Gurusamy, 2025c) was performed on simulated data, graphs generated from simulated data, and a set of CC0 images available from The *Cleveland Museum of Art* (The Cleveland Museum of Art, 2025).

Testing of *EQUALencrypt - Encrypt and decrypt columns of data* (Gurusamy, 2025b) was performed on simulated data.

The tests revealed that the decrypted data was identical to the encrypted data when the encrypted data, digital signature, and the keys matched. Decryption was not performed in any instance in which there was a mismatch in any of encrypted data, digital signature, and the keys, or if there was an alteration to the data.The detailed test results are available in the *GitHub* and *Zenodo* repositories (Gurusamy, 2025a, 2025b, 2025c).

The Journal of Open Source Software

Deployment testing

Deployment testing was performed in *Docker Desktop v4.45.0* (Docker, 2025a). *EQUALencrypt - Encrypt and decrypt whole files* (Gurusamy, 2025c) can be pulled from *Docker Hub* (Docker, 2025c) and run in *Docker Desktop*. *EQUALencrypt - Encrypt and decrypt columns of data* (Gurusamy, 2025b) can be pulled from *Docker hub* (Docker, 2025b) and run in *Docker Desktop*. The user interfaces for the two applications are available from the port mapped to port *3838*, the port used by both the applications.

Availability to others

The software including the source code is available from the *GitHub* and *Zenodo* repositories (Gurusamy, 2025a, 2025b, 2025c). The functions used in *EQUALencrypt - Encrypt and decrypt whole files* (Gurusamy, 2025c) and *Encrypt and decrypt columns of data* (Gurusamy, 2025b) are available as "EQUALencrypt" package from *The Comprehensive R Archive Network* (CRAN) (Gurusamy, 2025d). The license for the software and the functions are *GNU General Public License version 3* (GNU Operating System, 2025).

Acknowledgements

No external source of funding.

Conflicts of Interest

My salary and promotions are linked to performing and reporting high-quality research.

References

Docker. (2025a). *Docker desktop* (No. 31/08/2025; Vol. 2025). https://docs.docker.com/desktop/

Docker. (2025b). *kurinchi2k/equalencrypt_columns* (No. 31/08/2025; Vol. 2025). https://hub.docker.com/r/kurinchi2k/equalencrypt_columns

Docker. (2025c). *kurinchi2k/equalencrypt_whole_file* (No. 31/08/2025; Vol. 2025). https://hub.docker.com/r/kurinchi2k/equalencrypt_whole_file

GNU Operating System. (2025). *GNU general public license* (No. 21/08/2025; Vol. 2025). https://doi.org/10.5281/zenodo.16744058

Gurusamy, K. (2025a). *EQUALencrypt* (No. 31/08/2025; Vol. 2025). https://github.com/kurinchi2k/EQUALencrypt

Gurusamy, K. (2025b). *EQUALencrypt - encrypt and decrypt columns of data* (No. 21/08/2025; Vol. 2025). https://doi.org/10.5281/zenodo.16744058

Gurusamy, K. (2025c). *EQUALencrypt - encrypt and decrypt whole files* (No. 21/08/2025; Vol. 2025). https://doi.org/10.5281/zenodo.16743676

Gurusamy, K. (2025d). *EQUALencrypt: Encryption and decryption of files and data for researchers without coding skills* (No. 26/08/2025; Vol. 2025). https://CRAN.R-project.org/package=EQUALencrypt

Kelly, M. M., Martin-Peters, T., & Farber, J. S. (2024). Secondary data analysis: Using existing data to answer new questions. *Journal of Pediatric Health Care*, *38*(4), 615–618. https://doi.org/https://doi.org/10.1016/j.pedhc.2024.03.005

Lvovs, D., Creason, A. L., Levine, S. S., Noble, M., Mahurkar, A., White, O., & Fertig, E. J. (2025). Balancing ethical data sharing and open science for reproducible research in biomedical data science. *Cell Rep Med*, *6*(4), 102080. https://doi.org/10.1016/j.xcrm.2025.102080

Ooms, J., & Keyes, O. (2025). *Openssl: Toolkit for encryption, signatures and certificates based on OpenSSL* (No. 21/08/2025; Vol. 2025). https://CRAN.R-project.org/package=openssl

R Foundation. (2025). *The r project for statistical computing* (No. 31/08/2025; Vol. 2025). https://www.r-project.org/

Shiny. (2025). *Easy web apps for data science without the compromises* (No. 31/08/2025; Vol. 2025). https://shiny.posit.co/

The Cleveland Museum of Art. (2025). *Open access* (No. 21/08/2025; Vol. 2025). https://www.clevelandart.org/open-access