GMR
The Original

SECURE CLOUD-BASED FRAMEWORK FOR GENOMIC DATA STORAGE AND ANALYSIS

Chethan Venkatesh[1*], Srikanta A. S [2], Shiva Murthy G [2]

[1]Associate Professor, Department: MCA College RV Institute of Technology and Management, Bangalore, Affiliated to VTU, Belagavi
Email: chethanvenkatesh.rvitm@rvei.edu.in, orcid id: 0000-0001-8009-6409
[2]Srikanta A.S, Senior Associate professor, Department of Chemistry and Biochemistry, Vijaya College, Bangalore, Affiliated to Bengaluru City University, Email: srikantaakumalla7@gmail.com
[3]Associate Professor, Department: Computer Science and Engineering, University: VTU Centre for PG Studies, Muddenahalli
Email: kgshivam@gmail.com
*Corresponding Author: Chethan Venkatesh, Email: chethanvenkatesh.rvitm@rvei.edu.in,

ABSTRACT
The genome files have sensitive biological, familial and disease related information and require a secure computational environment for data storage and analysis. This study proposed and tested a safe and secure cloud-based platform for storing and analyzing genomic data. The framework was intended to provide for encrypted storage of files, pseudonymization of metadata management, verification of integrity, approval of access control, secure execution of analysis, encrypted storage of results, controlled variant queries and audit logging. Using generated FASTQ and Variant Call Format datasets, a computational prototype was implemented in Python. They utilized de-identification procedures, integrity verification via checksums, authenticated encryption, role-based and attribute-based access control, consent-aware approval, and temporary workspaces for secure analysis in the framework. Both genomic files were successfully uploaded, encrypted, analyzed and logged by the prototype. All output from the FASTQ quality control analysis and the summary analysis of the Variant Call Format were generated in protected temporary workspaces and stored in encrypted format. The security evaluation found that there was no unauthorized upload by an auditor, no access for commercial purposes, and access for approved research was allowed. The results of the performance evaluation demonstrated low upload, encryption and analysis run-times in the simulated environment, which suggests that it is technically viable for small-scale genomic processing. Major workflow and security events were recorded in audit logs, aiding in traceability and accountability. The results demonstrate that two key aspects of a secure genomic cloud infrastructure should be encryption and consent governance, access control, workflow isolation, and controlled query disclosure. The suggested framework offers a repeatable model of storage and analysis for genomic data, while ensuring privacy. The framework should be validated with more genomic data, production cloud infrastructure, advanced key management, federated authentication, and more effective privacy-preserving analytical techniques in the future.

KEYWORDS: Genomic data security; Cloud-based genomic analysis; Encrypted genomic storage; Consent-aware access control; Privacy-preserving variant query.

1. INTRODUCTION
Genomic information is now at the heart of the molecular diagnosis, population genomics, pharmacogenomics, investigation of rare diseases and precision medicine. With the advent of high throughput sequencing, vast amounts of sensitive biological data are generated and need to be stored, manipulated, queried, and shared over widely distributed research settings. Cloud computing provides scalable storage and elastic computation for the analysis of genomes at scale, but also poses risks in terms of confidentiality, integrity, access control, and secondary use (Schatz et al., 2010). The associated risks are particularly severe because genomic information is also characterized by the presence of persistent biological identifiers, familial relationships, markers of ancestry and variants related to disease, which cannot be eliminated by traditional de-identification (Gymrek et al., 2013). Securing the genomic cloud systems should be coupled with storage security, privacy-preserving analysis and accountable access control. With increasing focus on responsible and federated sharing of data, rather than the free flow of data, international genomic initiatives have been increasingly focusing on this (Global Alliance for Genomics and Health, 2016). Standardized computational formats, and reproducible analysis tools, are also important for sharing. The Variant Call Format (VCF) provides a means to represent variants in a structured format, and Sequence Alignment/Map tools continue to be essential for processing sequences and subsequent interpretation of the genome (Danecek et al., 2011; Li et al., 2009). Technical reproducibility is not enough, when sensitive genomic files are transferred through cloud-based systems, new workflows of SAMtools and BCFtools further support reproducible genomic analysis (Danecek et al., 2021). Genomic files need to be protected during upload, storage, access, short term analysis, result generation and audit review, by using an appropriate framework. Discovery tools or query interfaces can also present privacy hazards, if the genomic data are made available. Beacon systems are used to determine the presence of a genomic variant in a sample efficiently, but previous deployments demonstrated that if these systems are

used repeatedly to make Boolean queries, they can expose membership information in a genomic dataset (Shringarpure and Bustamante, 2015). Federated Beacon infrastructures were used to enhance the controlled genomic discovery process between institutions, and Beacon version 2 was expanded to enable standardized biomedical data discovery (Fiume et al., 2019; Rambla et al., 2022). The advances are signs that a secure, cloud-based genomic system should not be just a file encryption system. It should also limit access rights of users, research questions, consent requirements, permissions to ask a question, and who should get access to the result of the query.

Technical and governance safeguards are advocated in the recent genomic privacy literature. The privacy issues in sharing the genome include re-identification, unauthorized inference, and misuse of sensitive genomic traits by institutions (Bonomi et al, 2020). Thus, sociotechnical safeguards are needed, as genomic privacy relies on the synchronization between cryptographic measures, data governance, consent management and user accountability (Wan et al., 2022). Restriction of access, re-identification and secondary use is also generally supported by individuals' concerns about their genetic information, as explained in Clayton et al. (2018). Responsible sharing frameworks also focus on the need to make genomic data useful for research while holding those who access it to account, fairly, and protecting the rights of the people involved (Knoppers, 2014). These goals can be achieved through secure cloud analysis when the cloud is integrated into the analytical environment, with the inclusion of a robust audit system, access control, workflow isolation and encryption (Langmead and Nellore, 2018).

In this work we thus propose and test a secure cloud platform to store and analyze genomic data. The goals are: (i) to create a robust and secure infrastructure for the storage of genomic files with de-identification, encryption, integrity checking and traceable metadata; (ii) to develop controlled access and analysis workflows based on user role, approved research questions and consent requirements; and (iii) to test the prototype for security, runtime indicators, generation of encrypted output, and controlled variant query functionality. The framework enables sharing of genomic data that can be reused, while ensuring governance and protection needs are in line with FAIR data principles through the use of secure storage, consent-aware authorization, audit logging, temporary workspaces, and controlled query response (Wilkinson et al., 2016; Raisaro et al., 2018).

2. MATERIAL AND METHODS

2.1 Study design and research objectives

In this study, computational prototype design method is applied to design and test a secure cloud-based solution for the storage and analysis of genomic data. The framework was achieved as a local cloud-based simulation in Python, which includes encrypted object storage, a SQLite metadata repository, role-based and consent-aware access control, audit logging, temporary analytical workspaces, and light-weight genomic analysis functions.

2.2 Computational environment and framework architecture

The prototype was written in Python with the help of the module’s pandas, matplotlib, sqlite3, hashlib, secrets, uuid, pathlib, AESGCM in the cryptography package. A structured project folder for sample data, encrypted genomic data, encrypted analysis result, temporary workspace, tables, figures, and for a local key vault was created. The unique tables were created using SQLite: Users, Genomic files, Encryption keys, policies, Consent requests, analysis jobs, analysis results, audit logs, and evaluation metrics. It was the architecture that involved raw data generation, encryption of data in storage, management of metadata, access approval, execution of data analysis, encryption of data results, and evaluation of the data analysis framework.

2.3 Genomic data generation and metadata handling

Two prototype genomes data sets were created for testing purposes. The FASTQ data had 1,500 75 base reads. The Variant Call Format dataset consisted of 300 variant records except for all records being utilized for framework computational validation only on the chromosomes of 1, 2, 3, 7, 11, 17, and X. Any direct identifiers of the patient such as patient's name, email, phone number, address, national identification, medical record numbers etc. were removed from the metadata prior to storage. The uploaded files were each given a pseudo-anonymous sample identifier using a hash derived from the SHA-256, an organism label Homo sapiens, and a genome build GRCh38.

2.4 Security, storage, and access-control procedures

Extension has tested genomic files before upload. Accepted formats were FASTQ, FQ, VCF, BAM, SAM, BED, GFF, CSV, JSON and TXT. The files uploaded were encrypted in 256-bit Advanced Encryption Standard (AES) Galois Counter Mode (GCM), and key metadata was kept in the local key vault. The SHA-256 hash function was used for pre- and post-encryption. All users were given one of the five roles: administrator, data owner, researcher, clinician, or auditor. Upload, read, delete, approval, analysis, audit review, policy management and re-identification privileges were regulated by role permissions. Requests were checked for consent policies, whether the purpose was approved, whether the request was approved, the expiry date, and whether the request was read. In the controlled research configuration, only commercial use and re-identification were restricted.

2.5 Secure analysis and privacy-preserving query workflow

Only temporary workspaces specific to a certain job were used for approved files. The total reads, total bases, average read length and GC content were calculated by FASTQ analysis. VCF analysis performed on the results to compute total variants, mean quality, passing variants and number of unique chromosomes. Analysis outputs were stored, encrypted, and attached to job data and had an access status. Temporary workspaces were removed when they were finished or if

failed. A Beacon-style query layer was added to VCF files for the chromosome, position, reference allele and alternate allele queries to only return a controlled Boolean response.

2.6 Evaluation and output generation

Security evaluation was conducted on the following capabilities: denied unauthorized auditor uploads denied commercial use access, and approved research access. The results of the evaluation were captured as file size, upload time, encryption time and analysis time. Event data for user failure, user authentication, user creation, upload, access approvals, analysis, Beacon query and user access were captured in audit logs. The framework exported reproducible CSV tables of file size, upload time, encryption time, analysis time, security results, audit distribution, and components of the framework implemented. These outputs were kept for easy reproduction and assured manuscript level reporting.

3. RESULTS

The implemented framework generated a complete secure genomic storage and analysis workflow, including encrypted file upload, pseudonymized metadata registration, controlled access approval, secure analysis execution, encrypted result storage, security testing, performance measurement, audit logging, and final framework component reporting. The output files saved by the notebook confirm that the prototype successfully processed both FASTQ and Variant Call Format data while preserving controlled access and traceability. The uploaded genomic file metadata is summarized in **Table 1**.

Table 1. Uploaded genomic file metadata

Original filename	File type	File size (bytes)	Organism	Genome build	Consent group	Upload status
sample_reads.fastq	fastq	246390	Homo sapiens	GRCh38	controlled_research	Uploaded
sample_variants.vcf	vcf	11282	Homo sapiens	GRCh38	controlled_research	Uploaded

Table 1 shows that both genomic files were successfully uploaded, pseudonymized, assigned to the controlled research consent group, and stored with completed upload status.

The file size distribution shows that the FASTQ file represented the larger storage object, whereas the Variant Call Format file required substantially less storage. This pattern was expected because the FASTQ file contained 1,500 sequencing reads, while the Variant Call Format file contained summarized variant records. The size comparison is presented in

Figure 1.

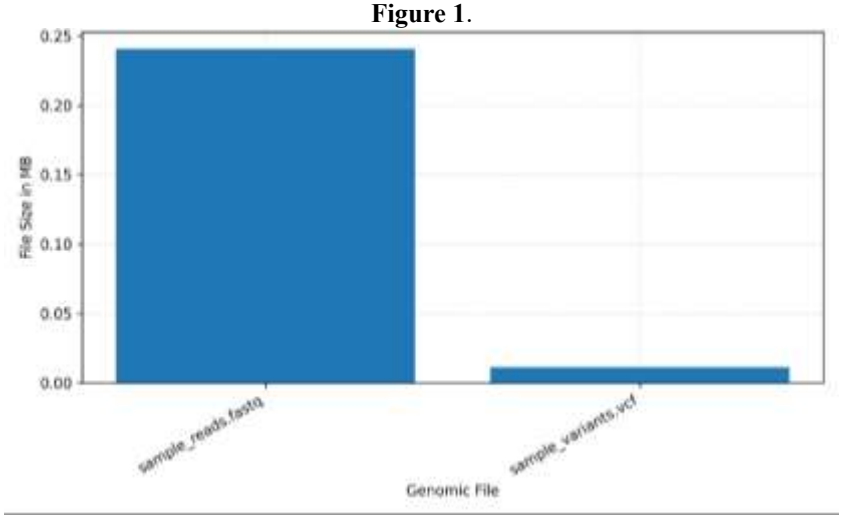

Figure 1. Uploaded file size

Figure 1 presents the uploaded genomic file size distribution and confirms the greater storage demand of sample_reads.fastq compared with sample_variants.vcf.

The performance results showed that the processing and uploading of the genomic files could be encrypted within a short time, showing performance. It took 0.005420 s to upload the FASTQ and 0.003458 s to encrypt. Uploading Variant Call Format took 0.002759 s and encryption took 0.001286 s. Analytical runtime was also low, with quality control of FASTQs taking 0.023149 s, and with the Variant Call Format summary analysis taking 0.008642 s. These results are summarized in **Table 2**.

Table 2. Performance evaluation of upload, encryption, and analysis

File or output name	File size (MB)	Upload time (s)	Encryption or analysis time (s)
sample_reads.fastq	0.234976	0.005420	0.003458
sample_variants.vcf	0.010759	0.002759	0.001286
FASTQ_QC	0.000000	0.000000	0.023149

VCF_SUMMARY	0.000000	0.000000	0.008642

Table 2 reports the measured upload, encryption, and analysis runtimes. The prototype completed all file handling and analytical tasks rapidly under the simulated local cloud environment.
The relative timing pattern is further shown in **Figure 2**, where the FASTQ file had higher upload and encryption time than the Variant Call Format file. This reflected the larger file size and the additional data volume requiring encryption.

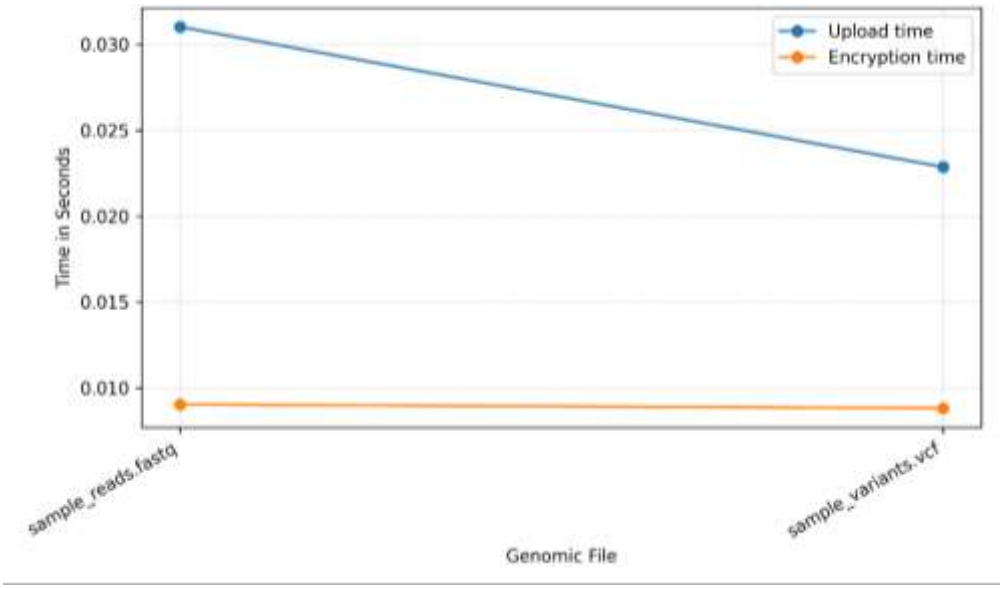

Figure 2. Upload Encryption time

Figure 2 compares upload and encryption times for the two uploaded genomic files and shows that processing time increased with file size.
The secure analysis stage generated two completed lightweight genomic analysis jobs. Both jobs used controlled encrypted input files and produced encrypted result outputs. The FASTQ job generated a quality-control summary, while the Variant Call Format job generated a variant summary. No job returned an error message, indicating that decryption into the temporary workspace, analysis execution, result generation, encryption of outputs, and workspace cleanup were successfully coordinated. The runtime comparison is shown in

Figure 3.

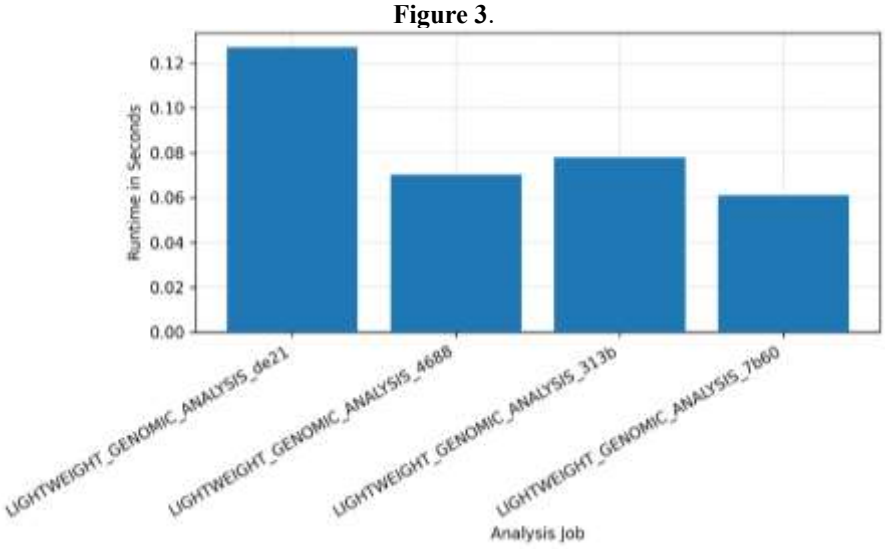

Figure 3. Analysis Run-time

Figure 3 shows the runtime of the two secure genomic analysis jobs. The FASTQ quality-control workflow required more time than the Variant Call Format summary workflow.
Security testing was done to establish that the access-control layer worked as expected. The auditor role did not have the permission to upload, commercial access was denied, and approved research access was permitted. All the three predefined security tests were successful, showing that role-based restrictions and access conditions based on consent have been applied in the prototype. These outcomes are summarized in **Table 3**.

Table 3. Security test outcomes

Security test	Passed
Auditor upload denied	1
Commercial purpose denied	1
Approved research access allowed	1

Table 3 confirms that all security tests were passed. A value of 1 indicates that the expected security behavior was achieved. The same security outcomes are visualized in **Figure 4**. The figure confirms that each access-control test returned a successful result, with no failed security-control outcome in the implemented test set.

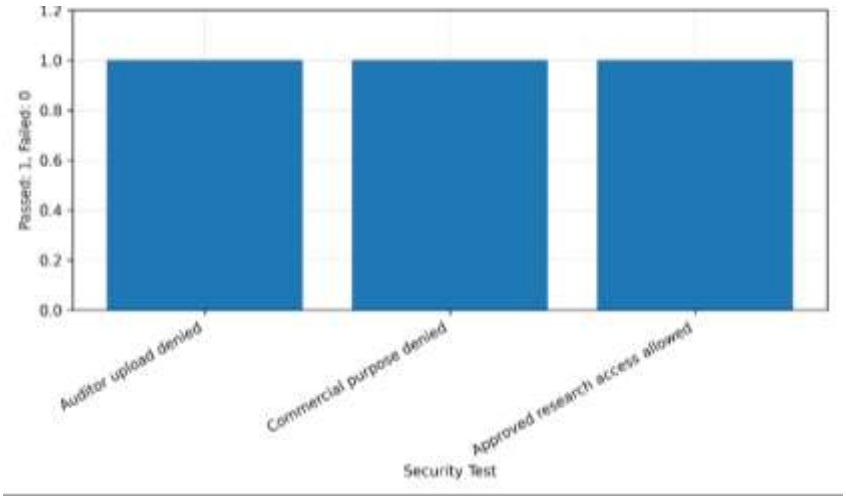

Figure 4. Security test outcomes

Figure 4 visualizes the pass status of the three security tests and confirms complete success across the tested control scenarios.

Audit logging produced 19 recorded events across user creation, login, file upload, access request creation, access approval, analysis completion, Beacon-style query execution, and failed unauthorized upload. The highest event count was for user creation, followed by user login and file upload. The audit distribution demonstrates that the framework captured both routine workflow events and security-relevant denial events. This audit pattern is presented in

Figure 5.

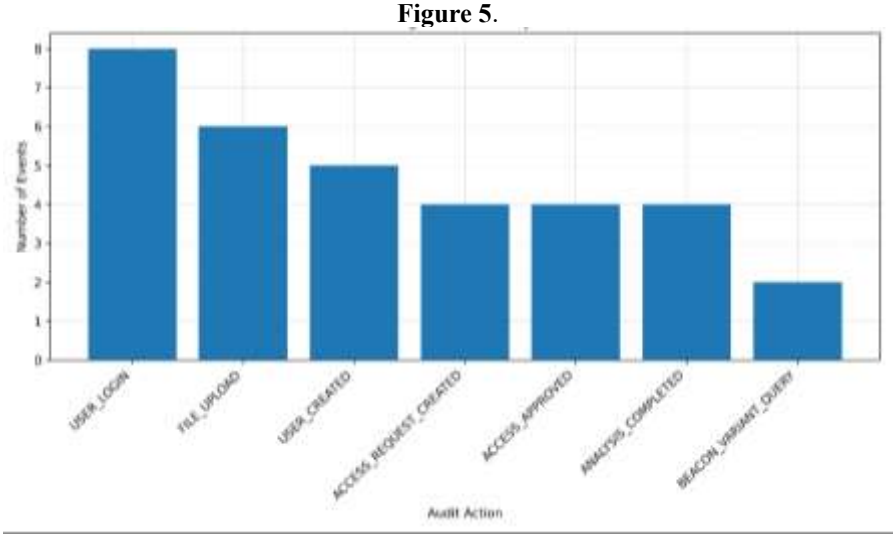

Figure 5. Audit Log Distribution

Figure 5 shows the distribution of audit log events by action type and confirms traceability across the major storage, access, analysis, and security events.

The actual output of the component-level activities showed that 12 framework components were put in place. These encompassed authentication, role-based access control, attribute-based access control, encrypted data storage, key management, data integrity checking, de-identification, secure analysis, genomic analysis, privacy-preserving queries, audit logging and assessment.

In conclusion, it is found that the proposed framework has added security and ensured secure storage of genetic data, pseudonymized handling of metadata, consent-aware access control, controlled genomic analysis, encrypted storage of

results, privacy-preserving query response, and traceability of execution. The performance results show that the small-scale files of the genomic data were feasible in the simulated cloud environment and the security tests confirmed that the upload of the data and unauthorized access purposes were properly and correctly limited. The objective was to demonstrate the suitability of the framework as a repeatable prototype for securing genomic data storage and analysis. The goal is to show that this is an appropriate framework that can be repeated to securely store and analyze genomic data.

4. DISCUSSION

The Proposed framework should be read in the context of the broader literature on genomic privacy, secure computation and governance through the cloud in the biomedical realm. The remaining references help to support the discussion: genomic query interfaces can be reconstructed and/or inferred (Ayoz et al., 2021; Froelicher et al., 2021); privacy preserving analytics can be used to support distributed precision medicine (Humbert et al., 2017); and privacy risks may extend to relatives and biologically related individuals (Ayoz et al., 2021; Froelicher et al., 2021; Humbert et al., 2017). They also advocate for the healthcare and genomic data systems' technical significance of authenticated encryption, homomorphic encryption, and the formal security-control standards (McGrew and Viega, 2004; Munjal and Bhatia, 2023; National Institute of Standards and Technology, 2007; National Institute of Standards and Technology, 2020). In this context, the present framework provides a small and reproducible prototype in which the security of storage, access control, secure temporary analysis, controlled variant query and auditing are integrated in one computational workflow.

The results show that the prototype was able to address the important functional requirements of a secure genomic storage and analysis system. The upload of both FASTQ and Variant Call Format files were encrypted; pseudonymous identifiers were assigned to them and traceable information about their origin was registered. However, when using quality-control and Variant Call Format summary analysis (FASTQ), the successful analysis of the genomic files revealed that they could only be decrypted in a temporary workspace specific to each analysis job and could only be re-encrypted after analysis. This workflow is key as it will minimize the amount of unnecessary exposure of raw genomic files whilst maintaining analytical usefulness. Additionally, by verifying and auditing file handling through a checksum, the trustworthiness of handling of files was further bolstered, as all of the major events within the workflow were documented, such as upload, access approval, analysis completion, execution of queries via the Beacon style and failure of a file to be uploaded due to unauthorized access.

The framework works out well with the small size of the genomic files, implying the viability of the framework in a simulated cloud environment. For both test files, the upload and encryption times were short, and the Analysis time spent was long, owing to the size of the file - the FASTQ file being larger than the Variant Call Format file. The results are then technically consistent with respect to the size of the file and duration of execution. Security tests were also conducted to verify whether the access control logic works as designed. The upload by auditor was not successful, commercial access was not allowed and approved research access was allowed. These findings also validate the core design principle of the framework, which is to implement secure genomic systems that incorporate the elements of encryption, purpose limitation, role permissions, consent status and auditable authorization.

The takeaway is that secure genomic cloud architecture should be engineered as a coordinated governance and computation system instead of being just an encrypted repository. While encryption keeps data safe while it is stored, it will not enforce who can access the data, how, why or under what consent, or how much disclosure is required. It proposes a solution to these problems by integrating role-based access control, attribute-based access control, consent-aware approval, pseudonymized metadata, result encryption and privacy preserving Boolean query response in the prototype. This type of integration is applicable to research institutions, biobanks, clinical genomics laboratories and joint genome-based initiatives that demand information utility and also privacy protection.

Several limitations to the study. The framework was tested with generated test data sets instead of a production grade multi-user cloud, in a simulated cloud setup. The FASTQ and Variant Call Format files were small files, and do not reflect the computational load of whole genome sequencing repositories and multi-terabyte clinical genomics repositories. The access-control and consent policies were not dynamic, meaning that there was no policy change for consent based on legal, institutional or jurisdiction changes. More sophisticated privacy protections (e.g., query throttling, noise addition, secure multiparty computation, federated policy enforcement) were not fully developed and were not applied to the Beacon-style query layer.

The framework should be tested in future studies with more extensive genomic data, real cloud infrastructure, multi-institutional access scenarios and distributed storage, and with more extended genomic pipelines, all of which are packaged in containers. Further development of future versions should also include more robust key-management, policy-aware consent dashboards, automated risk-scoring, federated authentication and scalable privacy-preserving computation. Along with researchers and clinicians, data stewards should perform additional evaluation, which also includes stress testing, adversarial query testing, formal threat modeling and usability assessment. Such extension would enhance the framework's applicability in secure genomic data sharing and analysis in the real-world biomedical research contexts.

5. CONCLUSION

This work is a development and testing of a secure cloud-based infrastructure to store and analyze genomic data. The framework included encrypted storage of genomic files, pseudonymized management of metadata, checking integrity, access control (both role-based and attribute-based), consent-aware approval, provision of secure temporary analysis spaces, generation of encrypted results, a control for variant queries following the Beacon model and audit logging. Both FASTQ and Variant Call Format (VCF) files were found to be successfully uploaded, encrypted, analyzed and documented using traceable metadata and audit events. The key findings show how essential the need for having secure management

of genomic data is not only at the storage level. The prototype demonstrated that access permissions and research purpose, consent status, and analysis controls can be integrated into a single reproducible workflow. The security test found that unauthorized auditor upload was not allowed, commercial purpose access was not allowed and approved research access was allowed. Technical feasibility was demonstrated in the simulated environment with low upload, encryption and analysis times to provide a picture of technical feasibility in the context of small-scale genomic files. The framework also allowed for controlled boolean variant queries, generation of variant summaries and analysis of the quality-control. The framework also did not expose raw files unnecessarily and maintained the analytical utility without losing out on features such as controlled boolean variant queries, generation of variant summaries and analysis of the quality-control. The study offers a practical basis to develop privacy-preserving genomic cloud infrastructure, particularly for environments where there is a need for the controlled reuse of genomic information for research. But validation is necessary at a wider level prior to being used on a large scale. Extension suggests the use of more extensive genomic data, real cloud services, more powerful mechanisms for key management, pipelines using containers, federated authentication, adversarial query evaluation, and more sophisticated privacy preserving computation in future work. This would enhance the scalability, security and institutional preparedness of the responsible sharing and analysis of genomic data.

REFERENCES

1. Ayoz K, Ayday E and Cicek AE (2021). Genome reconstruction attacks against genomic data-sharing beacons. Proc. Priv. Enhancing Technol. 2021: 28-48.
2. Bonomi L, Huang Y and Ohno-Machado L (2020). Privacy challenges and research opportunities for genomic data sharing. Nat. Genet. 52: 646-654. doi: 10.1038/s41588-020-0651-0.
3. Clayton EW, Halverson CM, Sathe NA and Malin BA (2018). A systematic literature review of individuals' perspectives on privacy and genetic information in the United States. PLoS One 13: e0204417. doi: 10.1371/journal.pone.0204417.
4. Danecek P, Auton A, Abecasis G, Albers CA, et al. (2011). The variant call format and VCFtools. Bioinformatics 27: 2156-2158. doi: 10.1093/bioinformatics/btr330.
5. Danecek P, Bonfield JK, Liddle J, Marshall J, et al. (2021). Twelve years of SAMtools and BCFtools. GigaScience 10: giab008. doi: 10.1093/gigascience/giab008.
6. Fiume M, Cupak M, Keenan S, Rambla J, et al. (2019). Federated discovery and sharing of genomic data using Beacons. Nat. Biotechnol. 37: 220-224. doi: 10.1038/s41587-019-0046-x.
7. Froelicher D, Troncoso-Pastoriza JR, Raisaro JL, Cuendet MA, et al. (2021). Truly privacy-preserving federated analytics for precision medicine with multiparty homomorphic encryption. Nat. Commun. 12: 5910. doi: 10.1038/s41467-021-25972-y.
8. Global Alliance for Genomics and Health (2016). A federated ecosystem for sharing genomic, clinical data. Science 352: 1278-1280. doi: 10.1126/science.aaf6162.
9. Gymrek M, McGuire AL, Golan D, Halperin E, et al. (2013). Identifying personal genomes by surname inference. Science 339: 321-324. doi: 10.1126/science.1229566.
10. Humbert M, Ayday E, Hubaux JP and Telenti A (2017). Quantifying interdependent risks in genomic privacy. ACM Trans. Priv. Secur. 20: 3.
11. Knoppers BM (2014). Framework for responsible sharing of genomic and health-related data. HUGO J. 8: 3. doi: 10.1186/s11568-014-0003-1.
12. Langmead B and Nellore A (2018). Cloud computing for genomic data analysis and collaboration. Nat. Rev. Genet. 19: 208-219. doi: 10.1038/nrg.2017.113.
13. Li H, Handsaker B, Wysoker A, Fennell T, et al. (2009). The Sequence Alignment/Map format and SAMtools. Bioinformatics 25: 2078-2079. doi: 10.1093/bioinformatics/btp352.
14. McGrew DA and Viega J (2004). The security and performance of the Galois/Counter Mode of operation. In: Progress in Cryptology, INDOCRYPT 2004. Springer, Berlin, 343-355.
15. Munjal K and Bhatia R (2023). A systematic review of homomorphic encryption and its contributions in healthcare industry. Complex Intell. Syst. 9: 3759-3786.
16. National Institute of Standards and Technology (2007). Recommendation for block cipher modes of operation: Galois/Counter Mode and GMAC. NIST Special Publication 800-38D. National Institute of Standards and Technology, Gaithersburg. doi: 10.6028/NIST.SP.800-38D.
17. National Institute of Standards and Technology (2020). Security and privacy controls for information systems and organizations. NIST Special Publication 800-53 Revision 5. National Institute of Standards and Technology, Gaithersburg. doi: 10.6028/NIST.SP.800-53r5.
18. Raisaro JL, Choi G, Pradervand S, Colsenet R, et al. (2018). Protecting privacy and security of genomic data in i2b2 with homomorphic encryption and differential privacy. IEEE/ACM Trans. Comput. Biol. Bioinform. 15: 1413-1426.
19. Rambla J, Baudis M, Ariosa R, Beck T, et al. (2022). Beacon v2 and Beacon networks: a "lingua franca" for federated data discovery in biomedical genomics, and beyond. Hum. Mutat. 43: 791-799. doi: 10.1002/humu.24369.
20. Schatz MC, Langmead B and Salzberg SL (2010). Cloud computing and the DNA data race. Nat. Biotechnol. 28: 691-693. doi: 10.1038/nbt0710-691.
21. Shringarpure SS and Bustamante CD (2015). Privacy risks from genomic data-sharing beacons. Am. J. Hum. Genet. 97: 631-646. doi: 10.1016/j.ajhg.2015.09.010.
22. Wan Z, Hazel JW, Clayton EW, Vorobeychik Y, et al. (2022). Sociotechnical safeguards for genomic data privacy. Nat. Rev. Genet. 23: 429-445. doi: 10.1038/s41576-022-00455-y.