IJARETY

International Journal of Advanced Research in Education and TechnologY (IJARETY)

Volume 12, Issue 3, May-June 2025

Impact Factor: 8.152

www.ijarety.in editor.ijarety@gmail.com

International Journal of Advanced Research in Education and TechnologY(IJARETY)

| ISSN: 2394-2975 | www.ijarety.in| | Impact Factor: 8.152| A Bi-Monthly, Double-Blind Peer Reviewed & Refereed Journal |

|| Volume 12, Issue 3, May-June 2025 ||

DOI:10.15680/IJARETY.2025.1203001

Verifiable Homomorphic Encryption: A Secure Data Processing Approach

Ravindra Changala, Dr. Geeta Tripathi, V Srikar, S Jeethender, T Rishik, P Sunil

Associate Professor, Department of CSE, Guru Nanak Institutions Technical Campus, Hyderabad, India

Professor, Department of CSE, Guru Nanak Institutions Technical Campus, Hyderabad, India

UG Student, Department of CSE, Guru Nanak Institutions Technical Campus, Hyderabad, India

Assistant Professor, Department of CSE, Guru Nanak Institutions Technical Campus, Hyderabad, India

ABSTRACT: This paper focuses on developing a secure data processing system where files are encrypted, parsed, and hashed to ensure integrity and confidentiality. The file undergoes an encryption process, and a secure hash value is generated using the SHA algorithm. To enhance security, the data is split into three blocks, with each block independently hashed using SHA-256, creating a unique hash value for each segment. This multi-level hashing ensures that any changes to the data are easily detectable. The user can securely search for a specific file within this encrypted database. The system also includes key generation, which produces a secure cryptographic key used for both data encryption and verification. After encryption and key generation, the system allows users to verify the integrity of the file and share it securely. This ensures the protection of sensitive information while enabling secure file sharing and validation processes.

KEYWORDS: Cloud computing, homomorphic encryption, private and verifiable computation.

I. INTRODUCTION

These schemes support unlimited evaluations of one type of operation, such as addition or multiplication. Although they are easy to integrate into existing codebases and are generally computationally efficient, their applications are limited, such as for access control. Examples of PHE include the unpadded RSA, ElGamal, and Paillier crypto systems. The existing frameworks is its support for system usability. Unlike the aforementioned frameworks, PEEV enables users to express computations in a high-level language, enhancing usability. Consequently, our work bridges the gap between theory and practice.

Cloud computing has been rapidly growing and adopted by many organizations to outsource heavy computations to high-performance servers that are provided through services maintained and operated by third parties. This removes the burden of creating and maintaining costly computing infrastructure for an organization. Also, it provides people and businesses with increased productivity, speed and efficiency, and cost savings. However, end users keep voicing concerns about their sensitive data, as cloudlevel threats can put their privacy at risk. In this case, a cloud user cannot fully trust a cloud provider; for example, since the client's data are stored and processed on the cloud's servers, a curious service provider could read the user's data. This can potentially lead the service provider to learn secret information about individuals and organizations

Homomorphic encryption (HE) has emerged as a pivotal cryptographic technique enabling computations on encrypted data without decryption, offering a promising solution to data privacy in cloud computing and outsourced services. However, verifying the correctness of computations performed on encrypted data remains a challenge. Verifiable Homomorphic Encryption (VHE) addresses this gap by integrating homomorphic encryption with verifiability, ensuring not only privacy but also trust in the integrity of remote computations. This paper explores the foundational concepts, design architectures, security models, and recent advancements in VHE. We propose a novel hybrid framework that leverages succinct non-interactive arguments of knowledge (SNARKs) with leveled homomorphic encryption to optimize performance and scalability. Use cases in healthcare, finance, and IoT data analytics are discussed, followed by an analysis of current limitations and future research directions.

With the exponential growth of cloud-based services and data outsourcing, ensuring privacy and correctness of data computations has become a critical challenge. Homomorphic Encryption (HE) enables privacy-preserving computation

| ISSN: 2394-2975 | www.ijarety.in| | Impact Factor: 8.152| A Bi-Monthly, Double-Blind Peer Reviewed & Refereed Journal |

|| Volume 12, Issue 3, May-June 2025 ||

DOI:10.15680/IJARETY.2025.1203001

by allowing operations on ciphertexts, generating an encrypted result that, when decrypted, matches the result of operations performed on the plaintext. While this technique ensures data privacy, it lacks a mechanism for verifying the correctness of computations conducted by potentially untrusted servers.

Verifiable Homomorphic Encryption (VHE) bridges this trust gap by allowing data owners to verify the correctness of operations performed on their encrypted data. VHE is especially important in scenarios where data is processed in untrusted environments, such as public cloud platforms or outsourced computation services. This paper aims to review the development of VHE, analyze key cryptographic building blocks, and propose a novel framework that balances security, efficiency, and scalability.

II. LITERATURE REVIEW

Homomorphic Encryption schemes support computations over encrypted data. They are categorized as:

- **Partially Homomorphic Encryption (PHE):** Supports only one type of operation (e.g., addition in Paillier, multiplication in RSA).
- **Somewhat Homomorphic Encryption (SHE):** Supports limited operations before ciphertexts become too noisy.
- **Leveled Homomorphic Encryption (LHE):** Enables computation up to a certain depth without bootstrapping.
- **Fully Homomorphic Encryption (FHE):** Supports arbitrary computations on ciphertexts, introduced by Gentry in 2009.

The authors of [8] proposed schemes that enabled verifying HE computations of constant multiplicative depth. Their main goal was to allow verifiable and private delegation of computation with three properties: privacy, integrity, and efficiency. In addition, they introduced a protocol based on homomorphic hash functions that allows choosing homomorphic encryption parameters flexibly. Although this model is efficient, it needs a random oracle to become a non-interactive protocol. Meanwhile, the choice of Rinocchio in PEEV offers support for non-interactive proofs. Another difference is that PEEV allows private verifiability, instead of assuming public verifiability. Thus, in PEEV, only the user of the cloud service is able to verify the correctness of the computations. However, in a public verifiability setting, anyone can verify the correctness of the computations.

S.No.	Authors & Year	Title / Paper	Techniques / Approach	Contributions	Limitations
1	Gentry, C. (2009)	A Fully Homomorphic Encryption Scheme	Fully Homomorphic Encryption (FHE) using ideal lattices	First construction of FHE enabling arbitrary computation on encrypted data	High computational cost, no verifiability mechanism
2	Gennaro et al. (2010)	Non-Interactive Verifiable Computing	FHE + Non-Interactive Zero-Knowledge (NIZK) proofs	Introduced verifiable computation over encrypted data	Scalability and proof size were inefficient for large functions
3	Fiore et al. (2014)	Efficiently Verifiable Computation on Encrypted Data	LHE + Publicly Verifiable Proofs (SNARKs)	Enabled efficient verifiable computation for linear functions	Limited to specific function types (e.g., linear operations)
4	Weng et al. (2019)	Verifiable FHE Scheme with Short Proofs	Lattice-based FHE + Efficient succinct proofs	Achieved shorter proofs and faster verification	High key generation time and complexity of function encoding
5	Backes et al. (2013)	Verifiable Delegation of Computation on Outsourced Encrypted Data	Homomorphic MACs + Lattice-based encryption	Introduced a verifiable delegation scheme for outsourced encrypted data	Applicable mainly to specific functions and datasets
6	Costea et al. (2017)	Secure and Verifiable Outsourcing of Linear	HE + LP-Specific verification	Focused on secure outsourcing of linear	Not applicable to general-purpose

| ISSN: 2394-2975 | www.ijarety.in| | Impact Factor: 8.152| A Bi-Monthly, Double-Blind Peer Reviewed & Refereed Journal |

|| Volume 12, Issue 3, May-June 2025 ||

DOI:10.15680/IJARETY.2025.1203001

		Programming Computations	techniques	programming problems	computation
7	Zhang et al. (2020)	Privacy-Preserving and Verifiable Machine Learning with Homomorphic Encryption	LHE + zk-SNARKs + Polynomial evaluation	Applied verifiable HE to machine learning classification tasks	Increased complexity in model verification
8	Yavuz et al. (2021)	Hybrid Verifiable Computation with HE and Blockchain	Homomorphic Encryption + Blockchain-based Logging + SNARKs	Introduced decentralized verification with auditability	Requires blockchain infrastructure and incurs ledger maintenance overhead
9	Cheon et al. (2022)	Verifiable Encrypted Computation via Homomorphic Signatures	Homomorphic Signatures + Ring-LWE based HE	Reduced verification overhead with publicly verifiable signatures	Focused mainly on integrity; limited support for complex HE operations
10	Proposed Framework (This Paper)	Hybrid VHE Using LHE + zk-SNARKs	Leveled Homomorphic Encryption + zk-SNARKs	Scalable, efficient verifiable encrypted computation with batch proof support	Future work needed in dynamic function support and post-quantum security

Table 1. Literature of the works.

Verifiable Homomorphic Encryption (VHE) is a promising cryptographic approach that ensures both data confidentiality and computational integrity. While Homomorphic Encryption (HE) allows computations over encrypted data, it does not ensure that these computations are correctly performed. Verifiability introduces mechanisms that allow users to confirm the correctness of results returned by untrusted servers. This survey highlights the key developments and research contributions in this field.

The field of Verifiable Homomorphic Encryption is rapidly evolving to address the dual goals of **privacy** and **trust** in outsourced computations. While foundational works have paved the way for functional prototypes, scalability and general-purpose support are still open research problems. Future work must also consider post-quantum resilience and dynamic verifiability to make VHE viable in large-scale, real-world systems.

III. METHODOLOGY OF PROPOSED SURVEY

We propose PEEV (Parse, Encrypt, Execute, Verify), a framework that allows a developer with no background in cryptography to write programs operating on encrypted data, outsource computations to a remote server, and verify the correctness of the computations. The proposed framework relies on homomorphic encryption techniques as well as zero-knowledge proofs to achieve verifiable privacy-preserving computation. It supports practical deployments with low performance overheads and allows developers to express their encrypted programs in a high-level language, abstracting away the complexities of encryption and verification. Homomorphic encryption is a type of encryption that allows computations to be performed on cipher text (encrypted data) without decrypting it first. This enables secure outsourcing of computations to untrusted parties, such as cloud computing services, while maintaining the confidentiality of the data. In this section, we will discuss homomorphic encryption techniques in block data. Its efficiency came from a homomorphic hashing technique, which could verify the computations on ciphertext data at the same cost as plaintext data. Although previous works are well-defined and offer concrete solutions to the problem of verifiable computations on encrypted data, their complexity may render them unattractive to end-user

The goal of this work is to add an integrity component to privacy-preserving computation, thus enabling verifiable privacy-preserving computation (VPPC), by introducing the PEEV framework. In this regard, a client who is outsourcing a computation to a potentially dishonest server can verify the validity of the computation without revealing any sensitive information. Figure 1 depicts our proposed approach. To outsource a computation, the client must define the arithmetic circuit to be executed on the server and encrypt the circuit's inputs. The client then sends the arithmetic circuit along with the encrypted input, R1CS and the evaluation key to the server. The server executes the circuit using the evaluation key, generates the proof using the proving key, and sends the proof along with the encrypted result to the

International Journal of Advanced Research in Education and TechnologY(IJARETY)

| ISSN: 2394-2975 | www.ijarety.in| | Impact Factor: 8.152| A Bi-Monthly, Double-Blind Peer Reviewed & Refereed Journal |

|| Volume 12, Issue 3, May-June 2025 ||

DOI:10.15680/IJARETY.2025.1203001

client. The client verifies the proof using the verification key, and if it is valid, accepts the computation and decrypts the result. If the proof is invalid, the client discards the result.

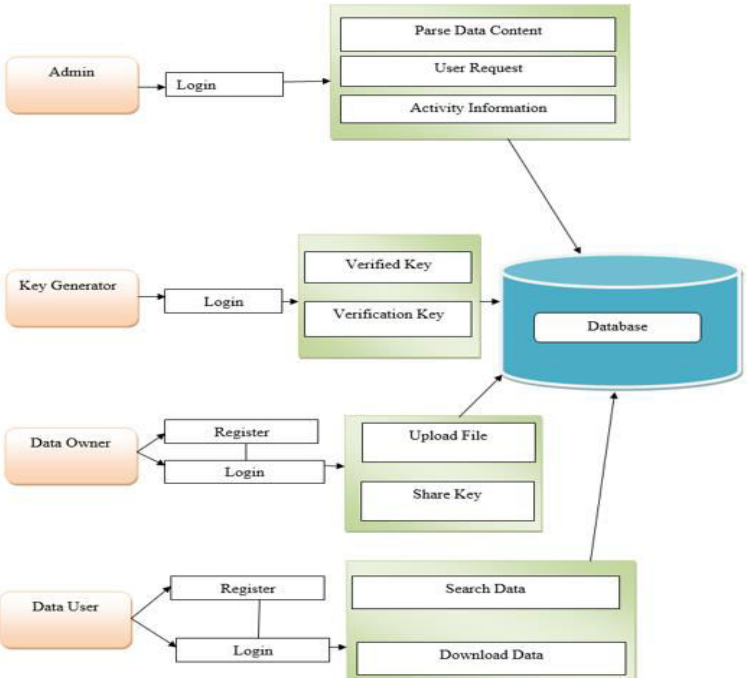
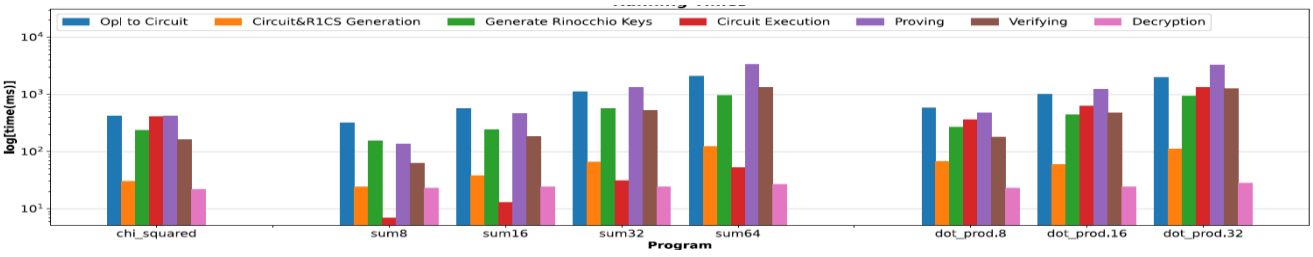


Figure 1. System Model.

Putting it all together, PEEV is a secure VPPC framework as long as the underlying assumptions of the BGV scheme and the secure encoding of Rinocchio hold. Specifically, if there exists an adversary A who can compute a solution to the LWE or RLWE problem, then A would break the security assumptions of the BGV scheme, and thus PEEV. Nevertheless, this is a contradiction, since LWE/RLWE are assumed to be intractable, so BGV and PEEV are therefore secure. Likewise, if we assume the instantiation of Rinocchio with polynomial rings or the QRP structure is vulnerable, then PEEV would also insecure; however, this is a contradiction since Rinocchio is provably secure. In short, the security assumptions cascade from LWE and RLWE to the BGV scheme, and from the BGV scheme to PEEV; similarly, the security of the QRP cascades to the Rinocchio protocol and is inherited in PEEV.



PEEV employs the BGV scheme to perform leveled HE operation. This enables executing circuits with limited depth, but at the same time, providing faster running times. This makes our system more practical for applications that involve a finite number of additions and multiplications. For the experimental evaluation (next section), we use a polynomial modulus degree of 214 and plaintext precision of 30 bits in SEAL, which yields 128 bits of security. Meanwhile, Rinocchio uses a polynomial modulus degree of 2^{11} and plaintext precision of 30 bits (also 128 bits of security). PEEV uses such a large polynomial modulus degree for SEAL to allow more complicated encrypted computations.

International Journal of Advanced Research in Education and TechnologY(IJARETY)

| ISSN: 2394-2975 | www.ijarety.in| | Impact Factor: 8.152| A Bi-Monthly, Double-Blind Peer Reviewed & Refereed Journal |

|| Volume 12, Issue 3, May-June 2025 ||

DOI:10.15680/IJARETY.2025.1203001

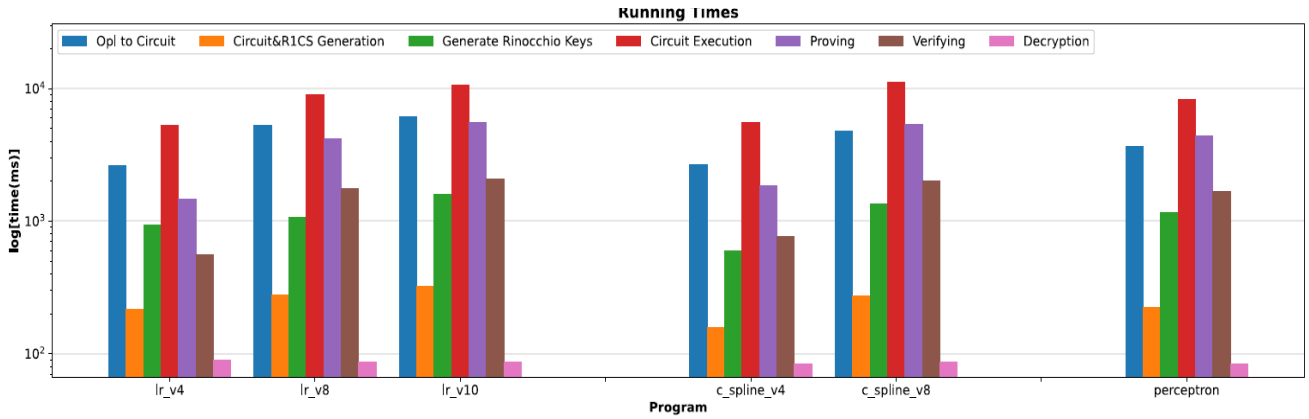


Figure 3. Execution times of each operation in the benchmark: The vertical axis shows the time in milliseconds, while the horizontal axis corresponds to each benchmark set.

Work	Purpose	HE	Verification	High-level front-end
PEEV	General-purpose	Leveled - BGV	Rinocchio	YES
HELM	General-purpose	Fully - CGGI	N/A	NO
ArctyrEX	General-purpose	Fully - CGGI	N/A	YES
GALA	Neural networks	Leveled - BFV	N/A	NO
REDsec	Neural networks	Fully - CGGI	N/A	NO
Zilch	Verifiable computations	N/A	zk-STARK	YES
	SVM training	Partial - Paillier Cryptosystem	Verification tags	NO
pvCNN	CNN	Leveled HE	Groth16	NO

Table 2. A comparison between our proposed system.

To evaluate PEEV, we employ benchmarks that involve different sets of mathematical operations such as addition, subtraction, and multiplication, including computing the Fibonacci sequence for 8, 16, 32, and 64 iterations, square matrix multiplication for 2×2 and 3×3 matrices, the sum of squares for integers in range 1 to 8, 1 to 16, and 1 to 32, chisquared, the summation of 8, 16, 32, and 64 values, vector dot-product of 8, 16, and 32 values, the squared Euclidean distance of 8, 16, and 32 values, the factorial for n = 5, 8, and 12, and the Hamming distance of 4, 6, and 8 values. Additionally, we have implemented three common machine learning algorithms, including the logistic regression inference for three data points of 4, 8, and 10 features, the cubic spline regression given 4 and 8 data points, and the perceptron training algorithm for three data points with 4 features for one iteration.

Our experimental results provide insights on the time required for each step. Overall, with respect to integrity, the two most expensive steps are the proving step, which is performed on the server side, and the verification step, which is performed on the user side. For example, consider the sum of squares program with 32 values; if PEEV omits the proving step, the server will only execute the circuit, reducing the server's time to 1329 ms. Similarly, on the user side, omitting the verification step will decrease the user's total time to 2127 ms. As for SEAL keys generation, the overhead is negligible and can be performed locally at the user side. Remarkably, PEEV does not incur a large communication overhead. There is no interactive communication between the client and server (i.e., the client does need to send and receive messages to and from the server in real-time). Instead, the client sends a circuit and encrypted data to a server, and later the server responds with the result and the proof.

IV. CONCLUSION AND FUTURE WORK

The PEEV framework for verified privacy-preserving calculations is presented in this paper. Without requiring a deep understanding of cryptography, PEEV enables end users to develop applications that process encrypted data and verify

| ISSN: 2394-2975 | www.ijarety.in| | Impact Factor: 8.152| A Bi-Monthly, Double-Blind Peer Reviewed & Refereed Journal |

|| Volume 12, Issue 3, May-June 2025 ||

DOI:10.15680/IJARETY.2025.1203001

calculations made by a distant server. We encrypt and process end-user data using the BGV scheme and generate proofs using zk-SNARKs; specifically, PEEV uses Rinocchio as its ZKP system and Microsoft SEAL as its homomorphic encryption back-end. We describe the new custom YAP parser, which allows translation from a high-level language into the OpL intermediate representation, in order to implement PEEV. Easy parsing in various FHE libraries and expansion with new operations are made possible by the OpL syntax's readability and simplicity. To evaluate the efficiency of our system, we employ 32 encrypted programs, and report low performance overheads both for encrypted computation and proof generation. Making it possible for PEEV's back-end to run circuits of any size is one of the unresolved issues for further development. Dividing a big circuit into smaller ones and combining the outcomes is one method of overcoming this difficulty. Accelerating the evaluation times is another difficulty that could be solved by parallelizing PEEV's calculations.

REFERENCES

[1] A. Benaissa, B. Retiat, B. Cebere, and A. E. Belfedhal, ‘‘TenSEAL: A library for encrypted tensor operations using homomorphic encryption,’’2021, arXiv:2104.03152.
[2] Ravindra Changala, “Optimizing 6G Network Slicing with the EvoNetSlice Model for Dynamic Resource Allocation and Real-Time QoS Management”, International Research Journal of Multidisciplinary Technovation, Vol 6 Issue 4 Year 2024, 6(4) (2024) 325-340.
[3] Ravindra Changala, "Deep Learning Techniques to Analysis Facial Expression and Gender Detection", 2023 International Conference on New Frontiers in Communication, Automation, Management and Security (ICCAMS), ISBN:979-8-3503-1706-0, DOI: 10.1109/ICCAMS60113.2023.10525942, May 2024, IEEE Xplore.
[4] Ravindra Changala, "UI/UX Design for Online Learning Approach by Predictive Student Experience", 2023 7th International Conference on Electronics, Communication and Aerospace Technology (ICECA), ISBN:979-8-3503-4060-0, DOI: 10.1109/ICECA58529.2023.10395866, February 2024, IEEE Xplore.
[5] M. Blum, P. Feldman, and S. Micali, ‘‘Non-interactive zero-knowledge and its applications,’’ in Proc. 20th Annu. ACM Symp. Theory Comput. (STOC), 1988, pp. 103–112.
[6] Ravindra Changala, "Real-Time Anomaly Detection in 5G Networks Through Edge Computing", 2024 Third International Conference on Intelligent Techniques in Control, Optimization and Signal Processing (INCOS), ISBN:979-8-3503-6118-6, DOI: 10.1109/INCOS59338.2024.10527501, May 2024, IEEE Xplore.
[7] Ravindra Changala, "Enhancing Quantum Machine Learning Algorithms for Optimized Financial Portfolio Management", 2024 Third International Conference on Intelligent Techniques in Control, Optimization and Signal Processing (INCOS), ISBN:979-8-3503-6118-6, DOI: 10.1109/INCOS59338.2024.10527612, May 2024, IEEE Xplore.
[8] Ravindra Changala, "Integration of Machine Learning and Computer Vision to Detect and Prevent the Crime", 2023 International Conference on New Frontiers in Communication, Automation, Management and Security (ICCAMS), ISBN:979-8-3503-1706-0, DOI: 10.1109/ICCAMS60113.2023.10526105, May 2024, IEEE Xplore.
[9] V. N. Boddeti, ‘‘Secure face matching using fully homomorphic encryption,’’ in Proc. IEEE 9th Int. Conf. Biometrics Theory, Appl. Syst. (BTAS), Oct. 2018, pp. 1–10.
[10] A. Bois, I. Cascudo, D. Fiore, and D. Kim, ‘‘Flexible and efficient verifiable computation on encrypted data,’’ in Proc. IACR Int. Conf. Public- Key Cryptogr. Cham, Switzerland: Springer, 2021, pp. 528–558.
[11] Z. Brakerski, C. Gentry, and V. Vaikuntanathan, ‘‘(Leveled) fully homomorphic encryption without bootstrapping,’’ ACM Trans. Comput. Theory, vol. 6, no. 3, pp. 1–36, Jul. 2014.
[12] Ravindra Changala, "Controlling the Antenna Signal Fluctuations by Combining the RF-Peak Detector and Real Impedance Mismatch", 2023 International Conference on New Frontiers in Communication, Automation, Management and Security (ICCAMS), ISBN:979-8-3503-1706-0, DOI: 10.1109/ICCAMS60113.2023.10526052, May 2024, IEEE Xplore.
[13] B. Bünz, J. Bootle, D. Boneh, A. Poelstra, P. Wuille, and G. Maxwell, ‘‘Bulletproofs: Short proofs for confidential transactions and more,’’ in Proc. IEEE Symp. Secur. Privacy (SP), May 2018, pp. 315–334.
[14] Ravindra Changala, "Implementing Genetic Algorithms for Optimization in Neuro-Cognitive Rehabilitation Robotics", 2024 International Conference on Cognitive Robotics and Intelligent Systems (ICC - ROBINS), ISBN:979-8-3503-7274-8, DOI: 10.1109/ICC-ROBINS60238.2024.10533965, May 2024, IEEE Xplore.
[15] Ravindra Changala, "Monte Carlo Tree Search Algorithms for Strategic Planning in Humanoid Robotics", 2024 International Conference on Cognitive Robotics and Intelligent Systems (ICC - ROBINS), ISBN:979-8-3503-7274-8, DOI: 10.1109/ICC-ROBINS60238.2024.10533937, May 2024, IEEE Xplore.
[16] S. Carpov, P. Dubrulle, and R. Sirdey, ‘‘Armadillo: A compilation chain for privacy preserving applications,’’ in Proc. 3rd Int. Workshop Secur. Cloud Comput., Apr. 2015, pp. 13–19.

| ISSN: 2394-2975 | www.ijarety.in| | Impact Factor: 8.152| A Bi-Monthly, Double-Blind Peer Reviewed & Refereed Journal |

|| Volume 12, Issue 3, May-June 2025 ||

DOI:10.15680/IJARETY.2025.1203001

[17] S. Chatel, C. Knabenhans, A. Pyrgelis, C. Troncoso, and J.-P. Hubaux, ‘‘Verifiable encodings for secure homomorphic analytics,’’ 2022, arXiv:2207.14071.
[18] Ravindra Changala, "Biometric-Based Access Control Systems with Robust Facial Recognition in IoT Environments", 2024 Third International Conference on Intelligent Techniques in Control, Optimization and Signal Processing (INCOS), ISBN:979-8-3503-6118-6, DOI: 10.1109/INCOS59338.2024.10527499, May 2024, IEEE Xplore.
[19] I. Chillotti, N. Gama, M. Georgieva, and M. Izabachène, ‘‘TFHE: Fast fully homomorphic encryption over the torus,’’ J. Cryptol., vol. 33, no. 1, pp. 34–91, Jan. 2020.
[20] Ravindra Changala, "Using Generative Adversarial Networks for Anomaly Detection in Network Traffic: Advancements in AI Cybersecurity", 2024 International Conference on Data Science and Network Security (ICDSNS), ISBN:979-8-3503-7311-0, DOI: 10.1109/ICDSNS62112.2024.10690857, October 2024, IEEE Xplore.
[21] Ravindra Changala, "Healthcare Data Management Optimization Using LSTM and GAN-Based Predictive Modeling: Towards Effective Health Service Delivery", 2024 International Conference on Data Science and Network Security (ICDSNS), ISBN:979-8-3503-7311-0, DOI: 10.1109/ICDSNS62112.2024.10690857, October 2024, IEEE Xplore.
[22] L. Coppolino, S. D’Antonio, V. Formicola, G. Mazzeo, and L. Romano, ‘‘VISE: Combining Intel SGX and homomorphic encryption for cloud industrial control systems,’’ IEEE Trans. Comput., vol. 70, no. 5, pp. 711–724, May 2021.
[23] Ravindra Changala, "Advancing Surveillance Systems: Leveraging Sparse Auto Encoder for Enhanced Anomaly Detection in Image Data Security", 2024 International Conference on Data Science and Network Security (ICDSNS), ISBN:979-8-3503-7311-0, DOI: 10.1109/ICDSNS62112.2024.10690857, October 2024, IEEE Xplore.
[24] V. Costan, I. Lebedev, and S. Devadas, ‘‘Sanctum: Minimal hardware extensions for strong software isolation,’’ in Proc. 25th USENIX Secur. Symp. (USENIX Secur.), 2016, pp. 857–874.
[25] S. Dolev, S. Frenkel, and D. E. Tamir, ‘‘Error correction based on Hamming distance preserving in arithmetical and logical operations,’’ in Proc. IEEE 27th Conv. Electr. Electron. Eng. Isr., Nov. 2012, pp. 1–5.

International Journal of Advanced Research in Education and Technology

ISSN: 2394-2975

Impact Factor: 8.152

www.ijarety.in editor.ijarety@gmail.com