iJRASET
International Journal For Research in Applied Science and Engineering Technology



INTERNATIONAL JOURNAL FOR RESEARCH

IN APPLIED SCIENCE & ENGINEERING TECHNOLOGY

Volume: 13 **Issue:** IV **Month of publication:** April 2025

DOI: https://doi.org/10.22214/ijraset.2025.68370

www.ijraset.com

Call: 08813907089 | E-mail ID: ijraset@gmail.com

International Journal for Research in Applied Science & Engineering Technology (IJRASET)
ISSN: 2321-9653; IC Value: 45.98; SJ Impact Factor: 7.538
Volume 13 Issue IV Apr 2025- Available at www.ijraset.com

Privacy-Preserving Technologies: Homomorphic Encryption and Secure Multi-Party Computation

Abbas Abubaker Mohammed Ahessin[1], Ali Mohammed Omar Ali[2]
College of Technical Sciences - Sebha

Abstract: ***With the increasing concerns over data privacy in cloud computing, healthcare, finance, and IoT, privacy-preserving technologies have gained significant attention. Among these, homomorphic encryption (HE) and secure multi-party computation (SMPC) stand out as powerful cryptographic techniques that enable computations on encrypted data without exposing sensitive information. This paper explores the principles, advancements, and real-world applications of HE and SMPC, comparing their strengths and limitations. We also discuss challenges in scalability, performance, and adoption, along with emerging trends in privacy-preserving computation.***

Keywords: Privacy-preserving computation, homomorphic encryption, secure multi-party computation, cryptographic security, data privacy.

I. INTRODUCTION

In an era where data breaches and privacy violations are rampant, traditional encryption methods (e.g., AES, RSA) protect data at rest and in transit but fail to support secure computation on encrypted data[1]. Homomorphic encryption (HE) and secure multi-party computation (SMPC) address this gap by allowing computations on encrypted or distributed data without decryption [2].

1) Homomorphic Encryption (HE): Enables arithmetic operations on ciphertexts, producing encrypted results that match operations on plaintexts.
2) Secure Multi-Party Computation (SMPC): Allows multiple parties to jointly compute a function over their inputs while keeping them private [3].
3) This paper examines:
4) The mathematical foundations of HE and SMPC.
5) Key algorithms and implementations.
6) Real-world applications and performance trade-offs.
7) Challenges and future research directions.

II. HOMOMORPHIC ENCRYPTION (HE)

1) Principles and Types of HE

HE allows computations on encrypted data, classified into:

- Partially Homomorphic Encryption (PHE): Supports one operation (e.g., addition or multiplication) [4].
- Example: RSA (multiplicative), (additive).
- Somewhat Homomorphic Encryption (SHE): Supports limited operations before noise corrupts results.
- Fully Homomorphic Encryption (FHE): Supports arbitrary computations (addition and multiplication) indefinitely.
- Breakthrough: Craig Gentry's 2009 FHE scheme using lattice-based cryptography.

2) Key Algorithms and Advancements

- Gentry's FHE Scheme: Bootstrapping reduces noise but is computationally expensive.
- BGV, BFV, CKKS: Modern FHE schemes optimizing for efficiency.
- HE Libraries: Microsoft SEAL, PALISADE, HElib.

3) Applications

- Secure Cloud Computing: Process encrypted data without exposing it to the cloud provider.
- Private Medical Analysis: Hospitals share encrypted patient data for research without revealing identities.

©IJRASET: All Rights are Reserved

- Financial Privacy: Banks compute risk assessments on encrypted transaction data[5].

4) Challenges

- High Computational Overhead: FHE can be **1000x slower** than plaintext operations.
- Key Management: Secure distribution of encryption keys remains complex.

III. SECURE MULTI-PARTY COMPUTATION (SMPC)

Secure Multi-Party Computation (SMPC) is a cryptographic protocol that enables multiple parties to jointly compute a function over their private inputs without revealing those inputs to each other [6]. Introduced by Andrew Yao (1982) in the "Millionaires' Problem," SMPC ensures:

- Privacy: No party learns more than the final output.
- Correctness: The computation is accurate and verifiable.
- Fairness: All parties receive the output simultaneously (or not at all).

Table (1) Key Properties of SMPC

Property	Description
Input Privacy	No party sees others' raw data.
Computational Integrity	Output is correctly computed.
Adversary Resistance	Security against malicious/semi-honest adversaries.
Decentralization	No single trusted third party needed.

A. Core Techniques in SMPC

1) Garbled Circuits (Yao's Protocol)

- Used for: Two-party computations (e.g., secure auctions) [7].
- How it works:
  - One party ("garbler") encrypts a Boolean circuit.
  - The other party ("evaluator") computes on the encrypted circuit.
  - Only the final output is revealed.
- Limitations:
  - Only works for two parties.
  - High communication overhead for complex functions.

2) Secret Sharing (Shamir's/BGW Protocol)

- Used for: Multi-party computations (e.g., federated learning) [8].
- How it works:
  - Each party splits its input into shares distributed among others.
  - Computations occur on shares (e.g., addition is local; multiplication requires interaction).
  - Shares are combined to reconstruct the result.
- Example:
  - Shamir's Secret Sharing: Polynomial interpolation to split/reconstruct secrets.
  - BGW Protocol: Extends to multiplication with error correction.

3) Oblivious Transfer (OT)

- Used for: Secure data retrieval (e.g., private database queries).
- How it works:
  - A sender has multiple messages; a receiver selects one without the sender knowing which [9].
- Variants:

- 1-out-of-2 OT: Receiver gets one of two messages.
- k-out-of-N OT: Extends to larger datasets.

4) Homomorphic Encryption + SMPC Hybrids

- Example:
- HE for storage, SMPC for computation (e.g., hospitals storing encrypted records but using SMPC for joint research).

B. Adversarial Models in SMPC

1) Semi-Honest (Passive) Adversaries

- Behavior: Follows the protocol but tries to infer extra information.
- Security Goal: Privacy is preserved despite leakage.
- Example: A curious cloud provider analyzing encrypted computations.

2) Malicious (Active) Adversaries

- Behavior: May deviate from the protocol (e.g., submit fake inputs).
- Security Goal: Detect or prevent cheating.
- Mitigations:
- Zero-Knowledge Proofs (ZKPs): Prove inputs are valid without revealing them.
- Commitment Schemes: Parties lock in inputs before computation [10].

C. Real-World Applications

1) Privacy-Preserving Machine Learning (PPML)

- Federated Learning + SMPC:
- Hospitals train ML models on patient data without sharing raw records.
- Google's "Private Join and Compute" uses SMPC for aggregate analytics.

2) Financial Privacy

- Secure Auctions: Companies compute the highest bid without revealing bids.
- Fraud Detection: Banks detect money laundering without exposing transactions [11].

3) Healthcare & Genomics

- Cross-Institutional Research: Hospitals compute disease correlations without sharing patient data.
- 23andMe & SMPC: Analyzing genetic data while preserving user privacy.

4) Blockchain & DeFi

- ZK-Rollups: Use SMPC to validate transactions off-chain.
- Dark Pools: Private trading venues using SMPC for order matching [12].

IV. COMPARATIVE ANALYSIS: HE VS. SMPC

Table (2) Homomorphic Encryption Vs Secure Multi-Party Computation

Feature	Homomorphic Encryption (HE)	Secure Multi-Party Computation (SMPC)
Computation Model	Single-party processing	Multi-party interaction
Performance	Slow (especially FHE)	Faster than FHE but high communication
Use Case	Cloud computing, encrypted DBs	Joint computations among distrustful parties
Scalability	Limited by computational cost	Limited by network latency

V. FUTURE DIRECTIONS AND CHALLENGES

A. Hybrid Approaches: Combining HE and SMPC

1) Current Limitations

- HE is computationally expensive but works well for single-party encrypted processing.
- SMPC is more efficient for interactive computations but requires high communication overhead [13].

2) Proposed Solutions

- HE-SMPC Hybrid Models:
  - Partial HE for Data Storage: Use HE to store encrypted data in the cloud.
  - SMPC for Computations: When multiple parties need to compute on the data, switch to SMPC for efficiency.
  - Example: A healthcare system could encrypt patient records with HE but use SMPC when hospitals collaborate on research.
- Threshold Homomorphic Encryption:
  - Combines HE with secret sharing, allowing decryption only if a threshold of parties agrees.
  - Useful in decentralized finance (DeFi) and secure voting systems [14].

B. Performance Optimization & Hardware Acceleration

1) Challenges

- FHE is Still Too Slow: Even optimized schemes like CKKS take seconds to minutes for simple operations.
- SMPC Suffers from Network Latency: Multi-round protocols become impractical for large-scale computations [15].

2) Emerging Solutions

- Hardware Acceleration:
  - FPGAs & ASICs: Custom hardware (e.g., Intel's HE-accelerator) speeds up FHE by 10-100x.
  - GPU Parallelization: Frameworks like CuFHE leverage NVIDIA GPUs for faster HE operations.
- Algorithmic Improvements:
  - Lattice-Based Optimizations: New FHE schemes (e.g., TFHE, FHEW) reduce bootstrapping time.
  - Non-Interactive SMPC: Reducing communication rounds using advanced cryptographic primitives [16].

C. Standardization & Regulatory Compliance

1) Current Status

- NIST's Post-Quantum Cryptography (PQC) Project:
  - Standardizing quantum-resistant algorithms (e.g., CRYSTALS-Kyber, Falcon).
  - Future HE schemes may integrate PQC to resist quantum attacks.
- GDPR, HIPAA, CCPA:
  - These regulations encourage privacy-preserving computation but lack specific guidelines for HE/SMPC.

2) Future Needs

- Standardized HE/SMPC APIs:
  - Libraries like Microsoft SEAL and OpenMined are steps in this direction.
- Legal Frameworks for Encrypted Computation:
  - Governments must clarify whether encrypted computations comply with data residency laws [17].

D. Scalability for Large-Scale Applications

1) Challenges

- HE's Ciphertext Expansion: Encrypted data can be 1000x larger than plaintext, making storage costly.
- SMPC's Network Bottlenecks: Large datasets require significant bandwidth for secure computation.

2) Potential Solutions

- Compression Techniques for HE:
  - Research into sparse encoding and quantization-aware HE to reduce ciphertext size.

- Efficient SMPC Protocols:
  - Batching Techniques: Process multiple operations in a single round.
  - Offline Preprocessing: Reduce online computation time (e.g., SPDZ protocol).

E. Security Against Advanced Threats

1) Current Risks

- Side-Channel Attacks:
  - HE implementations may leak metadata (e.g., timing, power consumption).
- Malicious Adversaries in SMPC:
  - Some SMPC protocols assume "semi-honest" participants, but real-world adversaries may cheat [18,19].

2) Future Mitigation Strategies

- Formal Verification of HE/SMPC Implementations:
  - Tools like EasyCrypt can mathematically prove protocol security [20,21,23].
- Post-Quantum SMPC:
  - Integrating quantum-resistant signatures (e.g., Dilithium) into SMPC frameworks.

F. Industry Adoption & Usability

1) Barriers to Adoption

- Complexity: Most HE/SMPC solutions require deep cryptographic expertise.
- Lack of Developer Tools: Few high-level APIs exist for easy integration.

2) Future Trends

- Automated Privacy-Preserving Compilers:
  - Tools like E3 convert plaintext programs into HE/SMPC-compatible versions [24].
- Cloud-Based HE/SMPC Services:
  - AWS, Google Cloud, and Azure may offer "privacy-as-a-service" with built-in HE/SMPC [25].

VI. CONCLUSION

Homomorphic encryption and secure multi-party computation represent groundbreaking advances in privacy-preserving computation. While HE excels in single-party encrypted processing, SMPC is better suited for collaborative computations among multiple entities. Both face challenges in performance and scalability, but ongoing research in optimization, hardware acceleration, and hybrid models holds promise for wider adoption. As data privacy regulations tighten, these technologies will play a crucial role in secure data processing across industries.

VII. FUTURE DIRECTIONS

1) Optimized HE schemes (e.g., leveled HE) to reduce computational costs.
2) SMPC protocols with fewer trust assumptions for broader adoption.
3) Standardized frameworks integrating these technologies into federated learning pipelines for healthcare IoT.

In conclusion, both HE and SMPC are indispensable tools for privacy-preserving federated learning, and their strategic application will be crucial in advancing secure, decentralized healthcare analytics.

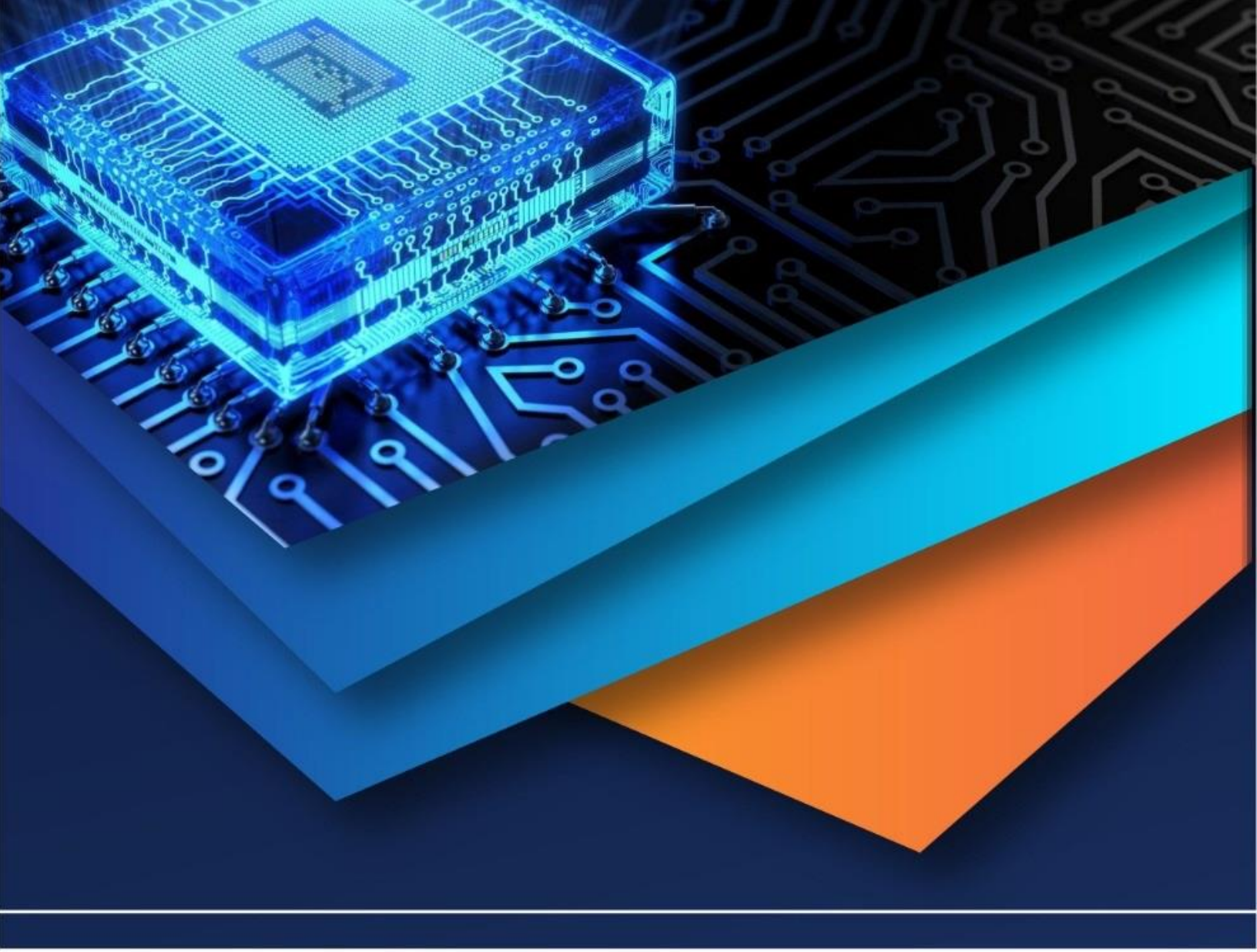
REFERENCES

[1] De Montesquieu, C. Montesquieu: The Spirit of the Laws; Cambridge University Press: Cambridge, UK, 1989.
[2] Nakamoto, S. Bitcoin: A Peer-to-Peer Electronic Cash System; Technical Report. 2019. Available online: https://git.dhimmel. com/bitcoin-whitepaper/ (accessed on 22 February 2021).
[3] Benhamouda, F.; DeCaro, A.; Halevi, S.; Halevi, T.; Jutla, C.; Manevich, Y.; Zhang, Q. Initial public offering (IPO) on permissioned blockchain using secure multiparty computation. In Proceedings of the 2019 IEEE International Conference on Blockchain (Blockchain), Atlanta, GA, USA, 14–17 July 2019.
[4] Vinod Vaikuntanathan. "Homomorphic Encryption References".
[5] R. L. Rivest, L. Adleman, and M. L. Dertouzos. On data banks and privacy homomorphisms. In Foundations of Secure Computation, 1978.

©IJRASET: All Rights are Reserved

[6] Sander, Tomas; Young, Adam L.; Yung, Moti (1999). "Non-interactive cryptocomputing for NC/Sup 1/". 40th Annual Symposium on Foundations of Computer Science (Cat. No.99CB37039). pp. 554–566. doi:10.1109/SFFCS.1999.814630. ISBN 978-0-7695-0409-4. S2CID 1976588.
[7] D. Boneh, E. Goh, and K. Nissim. Evaluating 2-DNF Formulas on Ciphertexts. In Theory of Cryptography Conference, 2005.
[8] Y. Ishai and A. Paskin. Evaluating branching programs on encrypted data. In Theory of Cryptography Conference, 2007.
[9] Gentry, Craig (2009). "Fully homomorphic encryption using ideal lattices". Proceedings of the forty-first annual ACM symposium on Theory of computing. pp. 169–178. doi:10.1145/1536414.1536440. ISBN 978-1-60558-506-2.
[10] Blakley, G.R. Safeguarding cryptographic keys. In Proceedings of the 1979 International Workshop on Managing Requirements Knowledge (MARK), New York, NY, USA, 4–7 June 1979; pp. 313–318.
[11] Zhong, H.; Sang, Y.; Zhang, Y.; Xi, Z. Secure multi-party computation on blockchain: An overview. In International Symposium on Parallel Architectures, Algorithms and Programming; Springer: Berlin, Germany, 2019, pp. 452–460.
[12] Ghadamyari, M.; Samet, S. Privacy-Preserving Statistical Analysis of Health Data Using Paillier Homomorphic Encryption and Permissioned Blockchain. In Proceedings of the 2019 IEEE International Conference on Big Data (Big Data), Los Angeles, CA, USA, 9–12 December 2019; pp. 5474–5479.
[13] Zaghloul, E.; Li, T.; Ren, J. Anonymous and Coercion-Resistant Distributed Electronic Voting. In Proceedings of the 2020 International Conference on Computing, Networking and Communications (ICNC), Big Island, HI, USA, 17–20 February 2020; pp. 389–393.
[14] Yan, X.; Wu, Q.; Sun, Y. A Homomorphic Encryption and Privacy Protection Method Based on Blockchain and Edge Computing. Wirel. Commun. Mob. Comput. 2020, 2020, 8832341. [CrossRef]
[15] Hyperledger Fabric. Available online: https://hyperledger-fabric.readthedocs.io/en/latest/whatis.html (accessed on 22 February 2021).
[16] Pedersen, T.P. Non-interactive and information-theoretic secure verifiable secret sharing. In Annual International Cryptology Conference; Springer: Berlin, Germany, 1991; pp. 129–140.
[17] Beaver, D. Efficient multiparty protocols using circuit randomization. In Annual International Cryptology Conference; Springer: Berlin, Germany, 1991; pp. 420–432.
[18] Damgård, I.; Pastro, V.; Smart, N.; Zakarias, S. Multiparty computation from somewhat homomorphic encryption. In Annual Cryptology Conference; Springer: Berlin, Germany, 2012; pp. 643–662.
[19] Feng, Q.; He, D.; Zeadally, S.; Khan, M.K.; Kumar, N. A survey on privacy protection in blockchain system. J. Netw. Comput. Appl. 2019, 126, 45–58. [CrossRef]
[20] Miers, I.; Garman, C.; Green, M.; Rubin, A.D. Zerocoin: Anonymous distributed e-cash from bitcoin. In Proceedings of the 2013 IEEE Symposium on Security and Privacy, Berkeley, CA, USA, 19–22 May 2013; pp. 397–411.
[21] Bonneau, J.; Narayanan, A.; Miller, A.; Clark, J.; Kroll, J.A.; Felten, E.W. Mixcoin: Anonymity for bitcoin with accountable mixes. In International Conference on Financial Cryptography and Data Security; Springer: Berlin, Germany, 2014; pp. 486–504.
[22] Heilman, E.; Baldimtsi, F.; Goldberg, S. Blindly signed contracts: Anonymous on-blockchain and off-blockchain bitcoin transactions. In International Conference on Financial Cryptography and Data Security; Springer: Berlin, Germany, 2016; pp. 43–60.
[23] Sun, S.F.; Au, M.H.; Liu, J.K.; Yuen, T.H. Ringct 2.0: A compact accumulator-based (linkable ring signature) protocol for blockchain cryptocurrency monero. In European Symposium on Research in Computer Security; Springer: Berlin, Germany, 2017; pp. 456–474.
[24] Andrychowicz, M.; Dziembowski, S.; Malinowski, D.; Mazurek, L. Secure multiparty computations on bitcoin. In Proceedings of the 2014 IEEE Symposium on Security and Privacy, Berkeley, CA, USA, 18–21 May 2014; pp. 443–458.
[25] Bentov, I.; Kumaresan, R. How to use bitcoin to design fair protocols. In Annual Cryptology Conference; Springer: Berlin, Germany, 2014; pp. 421–439.

©IJRASET: All Rights are Reserved | SJ Impact Factor 7.538 | ISRA Journal Impact Factor 7.894 |



10.22214/IJRASET

45.98

IMPACT FACTOR: 7.129

IMPACT FACTOR: 7.429

INTERNATIONAL JOURNAL FOR RESEARCH

IN APPLIED SCIENCE & ENGINEERING TECHNOLOGY

Call : 08813907089 (24*7 Support on Whatsapp)