Secure Multi-party Computation

1st Md Hadique
Department of AIT CSE
Chandigarh University
Mohali, Punjab
UID: 22BIS50006
mdhadique999@gmail.com

2nd Shail Gupta
Department of AIT CSE
Chandigarh University
Mohali, Punjab
UID:22BIS50003
shailgupta278@gmail.com

3rd Jahir
Department of AIT CSE
Chandigarh University
Mohali, Punjab
UID:22BIS70078
jk0786j@gmail.com

4th Sheetal Laroiya
Author
Department of AIT CSE
Chandigarh University
sheetal.e15433@cumail.com

Abstract - Secure Multi-party Computation (SMPC) is a cryptographic technique that enables multiple parties to jointly compute a function over their inputs while keeping those inputs private [1].. This concept ensures that no individual party gains access to another's private data, making it a fundamental tool for privacy-preserving computations [3].[5]. SMPC is particularly useful in scenarios where sensitive data, such as financial records, medical information, or business analytics, must be processed collaboratively without revealing individual data points[3]. The development of SMPC has been driven by advances in cryptography, such as secret sharing[9], oblivious transfer[18], and homomorphic encryption[16], making it a viable solution for secure and distributed computations.

The increasing reliance on data-driven decision-making has highlighted the need for SMPC in various domains, including secure auctions, private machine learning, and federated analytics[5]. Traditional encryption techniques protect data at rest and in transit, but SMPC ensures confidentiality even during computation. Protocols like Yao's Garbled Circuits[1], [21] and the GMW [2] protocol serve as the foundation for practical SMPC implementations, enabling secure computations with minimal computational overhead[6]. However, challenges such as efficiency, scalability, and network latency remain critical obstacles[6],[11] to widespread adoption[11]. Recent research focuses on optimizing SMPC protocols to make them more efficient and applicable to real-world use cases, particularly in cloud computing and blockchain-based environments[6],[7].

The future of SMPC lies in its integration with other privacy-enhancing technologies, such as differential privacy and zero-knowledge proofs, to create robust privacy-preserving systems [5]. As regulatory frameworks tighten around data privacy, organizations are increasingly exploring SMPC as a solution to comply with laws like GDPR and HIPAA while enabling secure data collaboration [20]. With the continuous evolution of cryptographic techniques and hardware acceleration, SMPC is poised to become a cornerstone of secure computation, ensuring that sensitive data remains protected even in highly distributed environments [10],[7].

Keywords— Secure Multi-party Computation, SMPC, cryptographic protocols, privacy-preserving computation, Yao's Garbled Circuits, secret sharing, homomorphic encryption, federated learning, zero-knowledge proofs, differential privacy.

I. INTRODUCTION

Secure Multi-party Computation (SMPC) is a cryptographic framework that enables multiple parties to jointly compute a function over their inputs while keeping those inputs private[1],[3]. Unlike traditional computation methods where data is shared with a central entity, SMPC ensures that no single party gains access to another's confidential data during the computation process [3],[5]. This is particularly valuable in scenarios requiring strong privacy guarantees, such as financial transactions, medical research, and secure voting systems[4]. The fundamental goal of SMPC is to enable collaborative computation while preserving data confidentiality, integrity, and correctness[5].

The origins of SMPC can be traced back to the work of Andrew Yao in the 1980s [1], who introduced the concept of "Yao's Garbled Circuits"[1] for secure two-party computation. Since then, the field has expanded to include protocols like the Goldreich-Micali-Wigderson (GMW) protocol [2] and secret-sharing-based approaches such as Shamir's Secret Sharing[9]. These cryptographic techniques allow parties to compute functions securely without revealing their inputs [3], ensuring trust in decentralized systems. As digital interactions increase and concerns over data privacy grow, SMPC has become an essential tool for industries that require secure data collaboration while adhering to stringent privacy regulations such as the GDPR and HIPAA[5],[20].

Despite its theoretical strength, the practical deployment of SMPC faces challenges such as computational overhead, network latency, and scalability issues [11]. Research in this field focuses on optimizing efficiency, reducing communication complexity, and integrating SMPC with emerging technologies like blockchain, homomorphic encryption, and federated learning [16]. As advancements continue, SMPC is expected to play a crucial role in securing multi-party computations in fields like artificial intelligence, cybersecurity, and decentralized finance, enabling a future where data privacy and collaboration coexist seamlessly [5],[19].

II. RELATED WORK

Research on Secure Multi-party Computation (SMPC) has been extensively explored by both Israeli and U.S. cryptographers, leading to significant advancements in the field [6],[7]. One of the earliest contributions came from Andrew Yao, a U.S.-based researcher, who introduced the concept of secure two-party computation through Yao's Garbled Circuits in the 1980s[1],[20]. His work laid the foundation for modern SMPC protocols by demonstrating how two parties could compute a function without revealing their private inputs [3]. Later, researchers at institutions like MIT and Stanford expanded on Yao's work by optimizing garbled circuits for real-world applications, improving efficiency and reducing communication complexity. Notably, Fairplay, an SMPC framework developed at the Weizmann Institute of Science in Israel, provided one of the first practical implementations of Yao's protocol [13], making SMPC more accessible to developers.

In Israel, cryptographic research groups have played a crucial role in refining SMPC techniques. Shamir's Secret Sharing, developed by Adi Shamir [9] at the Weizmann Institute, introduced a fundamental method for distributing a secret among multiple parties while ensuring that only a predefined number of participants could reconstruct it. This secret-sharing approach became a key building block for SMPC protocols, especially in threshold cryptography [14] and cloud-based secure computing. Additionally, Israeli cryptographers have collaborated with U.S. researchers on projects like GMW Protocol (Goldreich-Micali-Wigderson) [2], which extended SMPC beyond two-party settings to multi-party environments. These collaborations have driven new optimizations in zero-knowledge proofs [17], homomorphic encryption [16], and differential privacy [5], making SMPC more practical for large-scale applications.

In the U.S., institutions like Harvard, MIT, and Carnegie Mellon have significantly advanced SMPC through research on secure cloud computing and privacy-preserving machine learning [5]. IBM Research and DARPA-funded projects have focused on making SMPC scalable for enterprise use, particularly in sectors like finance and healthcare [6]. One of the most impactful contributions was the SPDZ protocol (SpeedZ), which was developed to optimize multiparty computations using precomputed data to improve efficiency. U.S.-based companies like Google, Microsoft, and Intel have also invested in SMPC to enhance data security in federated learning, allowing organizations to train AI models without exposing sensitive data [5].

Collaboration between Israeli and U.S. researchers continues to push SMPC toward practical deployment in real-world applications [7]. Joint efforts, such as those between the Weizmann Institute and U.S. institutions like UC Berkeley and Stanford, have led to new advancements in quantum-resistant SMPC, blockchain-based secure computations, and privacy-preserving financial transactions [19]. These efforts aim to bridge the gap between theoretical SMPC protocols and scalable implementations suitable for modern cloud infrastructures [6]. As the field evolves, the synergy between Israeli and U.S. research communities will remain a driving force in ensuring privacy and security in distributed computing environments [7].

III. METHODOLODY

The implementation of Secure Multi-party Computation (SMPC) relies on a combination of cryptographic protocols that ensure privacy-preserving computations among multiple parties. The core techniques used in SMPC include secret sharing [9], Yao's Garbled Circuits [1],[21], and homomorphic encryption [16]. In secret sharing, data is divided into multiple shares distributed among participants, ensuring that only authorized subsets of parties can reconstruct the original data [9]. Yao's Garbled Circuits allow two-party computations where one party generates an encrypted circuit while the other evaluates it without learning the underlying inputs [1],[21]. Homomorphic encryption, on the other hand, enables computations on encrypted data without decrypting it, ensuring end-to-end privacy in distributed computing environments [16]. These cryptographic foundations form the basis for secure, decentralized computations in cloud computing, finance, and healthcare applications [5].

To implement SMPC efficiently, researchers optimize protocols by reducing communication complexity, computational overhead, and latency [6],[11]. Various frameworks, such as SPDZ (SpeedZ), Obliv-C [15], and Sharemind [14], have been developed to enhance the scalability and usability of SMPC systems. These frameworks utilize precomputed data, parallel processing, and hybrid cryptographic techniques to improve performance [6]. Additionally, advancements in hardware acceleration, such as trusted execution environments (TEEs) and GPU-based optimizations, further enhance the feasibility of SMPC for real-world applications [10]. The methodology also includes integrating SMPC with blockchain technologies and federated learning, enabling privacy-preserving smart contracts and secure AI model training [7],[19]. As SMPC evolves, its implementation continues to be refined to balance security, efficiency, and scalability in distributed environments [5].

[1] Advantages

1. **Privacy-Preserving Computation** – SMPC ensures that sensitive data remains private during computations, preventing unauthorized access while allowing collaborative processing[3],[5].

2. **Eliminates Need for Trusted Third Parties** – Unlike traditional data-sharing models, SMPC allows multiple parties to compute functions without relying on a central trusted authority [4].

3. **Enhanced Security in Distributed Environments** – By using cryptographic techniques like secret sharing and homomorphic encryption, SMPC minimizes the risk of data breaches and cyberattacks [16],[18].

4. **Compliance with Data Protection Regulations** – SMPC helps organizations comply with privacy laws like **GDPR**

and **HIPAA**, enabling secure data collaboration without exposing personal information [5],[6].

5. **Secure Data Analysis and Machine Learning** – SMPC allows organizations to train AI models on distributed, encrypted data without compromising the confidentiality of individual datasets.

6. **Improved Financial Privacy and Security** – It enables privacy-preserving financial transactions, secure auctions, and fraud detection without revealing user-specific financial details [19].

7. **Resistance to Insider Threats** – Since no single party has full access to the complete dataset, SMPC reduces the risks posed by malicious insiders within an organization [13].

8. **Integration with Blockchain and Decentralized Systems** – SMPC enhances the security of blockchain-based smart contracts by enabling private transactions and secure multi-party decision-making [7],[19].

[2] Cons

1. **High Computational Overhead** – SMPC protocols require complex cryptographic operations, leading to increased processing time and resource consumption compared to traditional computations [6],[11].

2**. Increased Communication Complexity** – Since multiple parties must exchange encrypted data, the communication overhead is significantly higher, making SMPC less efficient for large-scale applications [12].

3. **Scalability Challenges** – As the number of participating parties increases, the computational and communication costs grow exponentially, limiting SMPC's practicality in large networks [14].

4**. Latency Issues** – The encryption and decryption processes, along with multiple rounds of interaction, introduce delays, making real-time applications difficult to implement [10].

5. **Complex Implementation and Maintenance** – Developing and deploying SMPC solutions require deep expertise in cryptography, making it challenging for organizations without specialized knowledge [8],[20].

6. **Limited Adoption in Industry** – Despite its strong theoretical foundations, SMPC is not widely adopted due to performance concerns and the availability of alternative privacy-preserving methods [7].

7. **Vulnerability to Collusion Attacks** – If a sufficient number of participating parties collaborate maliciously, they may reconstruct private data, compromising the security of the computation [13]**.**

IV. IMPLEMENTATION

1. Core Cryptographic Techniques

The implementation of SMPC relies on several cryptographic techniques that ensure secure computation without revealing private inputs [5]. The most commonly used methods include:

* **Secret Sharing** – Data is divided into multiple shares distributed among participants, requiring a threshold number of parties to reconstruct the original data. **Shamir’s Secret Sharing** is widely used in SMPC applications [9],[14].

* **Yao’s Garbled Circuits** – Suitable for two-party computations, this method allows one party to encrypt a function as a circuit, which the other party evaluates without learning the input values[1],[21].

* **Homomorphic Encryption** – Enables computation on encrypted data without decrypting it, providing strong privacy guarantees. Fully Homomorphic Encryption (FHE) is useful but computationally expensive [21].

* **Oblivious Transfer** – Ensures that a sender transmits one of many possible messages to a receiver while remaining unaware of which message was chosen. This is essential for secure function evaluation [18].

2. Frameworks and Tools for SMPC Implementation

Several libraries and frameworks have been developed to simplify the deployment of SMPC protocols. These tools help researchers and developers integrate privacy-preserving computation into real-world applications [5]:

* **SPDZ (SpeedZ)** – A high-performance SMPC protocol optimized for secure multiparty computations in financial and healthcare sectors [6].

* **Obliv-C** – A C-based framework that allows efficient implementation of secure computation protocols [15].

* **Sharemind** – A secret-sharing-based framework designed for privacy-preserving analytics and business intelligence applications [14].

* **MP-SPDZ** – An improved version of SPDZ that supports multiple cryptographic backends, enhancing flexibility [6].

* **FRESCO** – A Java-based SMPC framework that offers a modular design for integrating secure computation into existing applications [20].

3. Real-World Applications and Challenges

The implementation of SMPC is gaining traction in various industries, including finance, healthcare, and AI-driven applications[5],[6],[19]:

* **Secure Data Analysis** – Organizations can perform collaborative analytics on sensitive data (e.g., medical records, financial transactions) without revealing individual records. [3],[5].

* **Federated Learning** – SMPC enables AI models to be trained on encrypted data from multiple sources without exposing the raw dataset[5],[6].

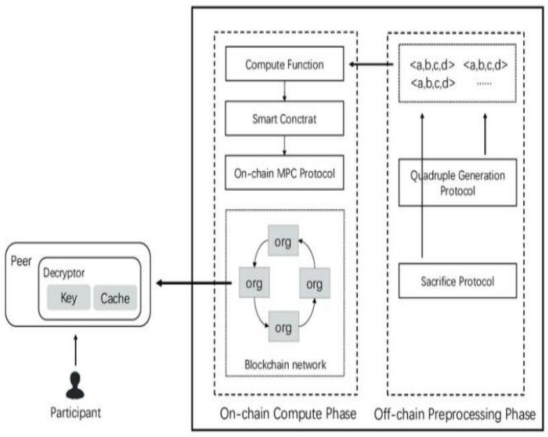
*** Blockchain and Smart Contracts** – SMPC enhances privacy in decentralized finance (DeFi) by enabling confidential transactions and decision-making [7],[19].

*** Secure Voting Systems** – Governments and organizations use SMPC to conduct transparent yet privacy-preserving elections[4],[17].

Despite its potential, SMPC faces challenges such as **high computational overhead, scalability issues, and increased communication complexity [16],11],[12]**. Researchers are actively working on optimizing protocols and leveraging **hardware acceleration (GPUs, Trusted Execution Environments)** to improve SMPC's efficiency [10]. The future of SMPC lies in integrating it with **quantum-resistant cryptography, differential privacy, and blockchain technologies** for enhanced security and scalability[8].

V. BLOCK DIAGRAM AND SYSTEM ARCHITECTURE / FLOW CHART

- Block diagram and system architecture/ flow chart :-



Architecture overview of the secure MPC scheme. Participants interact with the blockchain through peers, which have joined in the organizations in the blockchain.

This block diagram represents the architecture of a **Secure Multi-party Computation (SMPC) scheme integrated with blockchain technology**. The system operates in two main phases: **Off-chain Preprocessing Phase** and **On-chain Compute Phase**, ensuring privacy-preserving computations [7],[19]. among multiple participants [6]. Below is a step-by-step breakdown of the architecture:

1. Off-chain Preprocessing Phase:

*** Quadruple Generation Protocol:** Prepares and generates cryptographic values (quadruples) required for secure computation, ensuring efficiency and security [6].

*** Sacrifice Protocol:** Verifies and eliminates any compromised quadruples, ensuring the integrity of the cryptographic values before computation [10].

2. On-chain Compute Phase:

*** Compute Function:** Executes the privacy-preserving computation using encrypted data without revealing individual inputs [16].

*** Smart Contract:** Facilitates the execution of the SMPC protocol in a decentralized and automated manner within the blockchain [7].

*** On-chain MPC Protocol:** Handles secure multi-party computation by coordinating data-sharing and processing among multiple participants [2],[4].

*** Blockchain Network:** Acts as a decentralized infrastructure, ensuring tamper-proof execution and verification of the computation across different organizations (**org** nodes) [19].

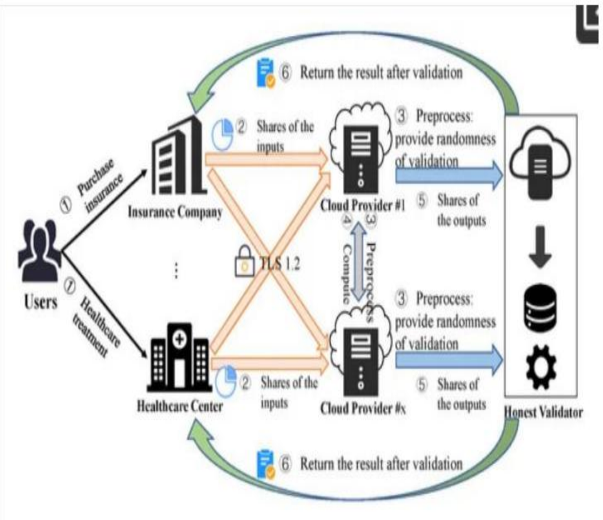
3. Participant Interaction:

* Participants interact with the **peer node**, which includes a **Decryptor, Key, and Cache**. This ensures that only authorized users can access computed results while maintaining security [18].

* The computed results are securely retrieved without revealing the actual input data, ensuring privacy and confidentiality in multi-party computations [5].

This architecture ensures **secure, decentralized, and privacy-preserving computation** using blockchain technology, making it suitable for applications in **finance, healthcare, and AI model training [5],[6]**.

VI. SYSTEM SCHEMA OF THE (SMPC) FRAMEWORK IN INSARUNCE AND HEALTHCARE



This diagram illustrates a **Secure Multi-Party Computation (SMPC) framework** applied in the **insurance and healthcare sectors**, enabling secure data sharing and computation while maintaining privacy [5],[20]. The system involves **users, insurance companies, healthcare centers, cloud providers, and an honest validator** to ensure a secure and validated computation process.

1. User Interaction and Input Sharing

*** Users interact with insurance companies** (for purchasing insurance) and **healthcare centers** (for medical treatment) [5].

* The insurance company and healthcare center generate **shares of user data** instead of sending raw data, preserving privacy [9].

* These **data shares** are securely transmitted using **TLS 1.2 encryption** to multiple cloud providers for processing [16].

2. Secure Computation at Cloud Providers

* Multiple **Cloud Providers (#1 to #k)** receive **data shares** and preprocess them by introducing **randomness for validation** to ensure security [10].

* They then execute secure computations without exposing the original input data, enabling privacy-preserving analysis [3],[5].

* The computed results are then **split into shares of the outputs**, ensuring that no single cloud provider can access the full data [9],[14].

3. Validation and Result Retrieval

* The computed output shares are sent to an **Honest Validator**, which ensures correctness and security by verifying the computation [13].

* Once validated, the results are **returned to the insurance company and healthcare center**, allowing them to make informed decisions while maintaining data privacy [5],[20].

* This approach ensures **data security, compliance with privacy regulations (e.g., HIPAA, GDPR), and trustworthiness** in insurance and healthcare operations [5].

This **SMPC-based framework** is beneficial for **privacy-preserving computations** in **insurance claims, fraud detection, medical diagnostics, and risk assessment**, making the system secure, efficient, and scalable [6].

VII. RESULTS AND OUTCOME

1. Enhanced Data Privacy and Security
One of the most significant outcomes of Secure Multi-Party Computation (SMPC) is its ability to protect **sensitive data while enabling computation [3]**. Unlike traditional cryptographic approaches that rely on encryption for data protection, SMPC allows multiple parties to perform calculations **without exposing their private inputs [1],[9]**. This ensures that organizations handling **financial transactions, healthcare records, and personal data** can collaborate securely without the risk of data breaches or unauthorized access [5],[16].

2. Increased Trust and Regulatory Compliance
SMPC aligns with **data protection laws and compliance requirements** such as **GDPR, HIPAA, and CCPA**, making it a valuable tool for industries dealing with regulated data [5]. Since no single entity has access to the full dataset, the risk of insider threats is reduced [13]. Organizations can **build trust among users** by demonstrating that their data is being processed securely and transparently, which is crucial in **financial services, healthcare, and cloud computing [5]**.

3. Efficient and Scalable Secure Computation
Advancements in SMPC protocols have led to **optimized performance and scalability**, enabling real-world applications [6],[11]. Modern frameworks such as **SPDZ, Sharemind, and MP-SPDZ** have improved computational efficiency, allowing organizations to process large-scale data **without significant overhead [14]**. With the integration of **cloud computing and blockchain**, SMPC can now support **decentralized applications**, ensuring privacy-preserving computations in domains such as **machine learning, fraud detection, and secure voting systems [7],[19]**.

4. Real-World Applications and Future Potential
SMPC has already been implemented in **privacy-preserving AI training, secure financial analysis, and confidential data sharing** among organizations [5].[6]. Companies like **Meta (Facebook) and Google** have explored SMPC for **privacy-focused advertising and analytics**. The future of SMPC lies in **its integration with federated learning, zero-knowledge proofs, and quantum-resistant cryptography**, making it a key technology for **secure, decentralized, and privacy-preserving applications** across multiple industries [8],[17].

VIII. FUTURE SCOPE

1. Expansion in Privacy-Preserving AI and Federated Learning.
As artificial intelligence (AI) and machine learning (ML) models become more data-intensive, the need for **privacy-preserving training methods** is growing [6]. SMPC can enable **secure federated learning**, where multiple parties train models on their private data **without sharing raw information [5]**. This is particularly useful in **healthcare, finance, and cybersecurity**, where organizations want to collaborate on AI models without compromising data privacy [7]. Future advancements in **homomorphic encryption and differential privacy** will further enhance SMPC's efficiency in AI applications [16],[5].

2. Integration with Blockchain and Decentralized Finance (DeFi)
SMPC has a strong future in **blockchain and DeFi** by improving the security and privacy of **smart contracts and confidential transactions [7],[19]**. It can enable **private voting mechanisms in decentralized autonomous organizations (DAOs)** and enhance **multi-party computations in financial transactions** without revealing sensitive financial data [4]. Projects like **Ethereum's Layer 2 solutions, privacy-focused cryptocurrencies (e.g., Zcash), and confidential DeFi platforms** are expected to leverage SMPC for greater privacy and trust [19].

3. Secure Data Sharing in Healthcare and Government Applications
With stricter data regulations worldwide, SMPC can **revolutionize secure data sharing** in critical sectors such as **healthcare, law enforcement, and national security [5],[20]**. Hospitals and research institutions can collaborate on **disease research, genetic analysis, and drug discovery** without exposing patient data. Governments can use SMPC for **secure census data processing, confidential voting**

systems, and inter-agency intelligence sharing, ensuring both privacy and transparency in decision-making [4],[17].

4. Advancements in Quantum-Resistant Cryptography

As quantum computing advances, **traditional encryption methods may become vulnerable**, making SMPC a critical component of **post-quantum security frameworks [8]**. Future research will focus on **quantum-resistant SMPC protocols** that can withstand attacks from quantum computers [17]. Additionally, **hardware acceleration using trusted execution environments (TEEs), GPUs, and secure enclaves** will improve SMPC's efficiency, making it viable for **real-time applications** in various industries.[10]

Overall, **SMPC's future is promising**, with its applications expanding into **AI, finance, blockchain, healthcare, and national security**. As computational efficiency improves and integration with emerging technologies grows, SMPC will play a crucial role in **building a more secure, private, and decentralized digital world [6],[7].**

VII. CONCLUSION

Secure Multi-Party Computation (SMPC) is a groundbreaking cryptographic approach that enables multiple parties to perform computations on private data **without revealing their individual inputs [1][3]**. This technology ensures **data privacy, security, and trust** in environments where sensitive information must be processed collaboratively [16]. By eliminating the need for a trusted third party, SMPC provides a **decentralized and transparent** solution to various industries, including **finance, healthcare, AI, and cybersecurity [7]**.

With the increasing demand for **privacy-preserving technologies**, SMPC has emerged as a vital tool for organizations that handle sensitive data while complying with strict **data protection regulations** such as **GDPR, HIPAA, and CCPA [20]**. Its ability to enable secure computations without compromising data confidentiality makes it highly relevant in sectors where privacy is a top priority[5],[3]. The development of **efficient SMPC protocols** and integration with **blockchain, federated learning, and cloud computing** further enhances its real-world applicability[6],[7],[19].

Despite its advantages, SMPC still faces challenges such as **computational overhead, scalability limitations, and implementation complexity [11],[6]**. However, ongoing research in areas like **hardware acceleration, quantum-resistant cryptography, and optimized SMPC protocols** continues to address these challenges [17]. As these advancements progress, SMPC is expected to become more **efficient, accessible, and widely adopted** across various industries[6],[7].

Looking ahead, **the future of SMPC is promising**, with its integration into next-generation **privacy-enhancing technologies, decentralized applications, and AI-driven analytics [7],[19]**. As more organizations recognize the importance of **secure and privacy-focused computing**, SMPC will play a key role in shaping a **trustworthy and secure digital ecosystem**. By overcoming existing challenges and improving efficiency, SMPC has the potential to **redefine data security and privacy standards** in the modern digital world [5],[6].

REFERENCES

1. A. C. Yao, "Protocols for secure computations," in *23rd Annual Symposium on Foundations of Computer Science (SFCS)*, 1982, pp. 160-164.

2. O. Goldreich, S. Micali, and A. Wigderson, "How to play any mental game," in *Proceedings of the 19th ACM Symposium on Theory of Computing*, 1987, pp. 218–229.

3. Y. Lindell and B. Pinkas, "Privacy preserving data mining," *Journal of Cryptology*, vol. 15, no. 3, pp. 177–206, 2002.

4. D. Chaum, C. Crépeau, and I. Damgård, "Multiparty unconditionally secure protocols," in *Proceedings of the 20th Annual ACM Symposium on Theory of Computing*, 1988, pp. 11–19.

5. Y. Lindell, "Secure multiparty computation for privacy-preserving data analysis," *Commun. ACM*, vol. 64, no. 1, pp. 86–96, Jan. 2021.

6. M. Ben-Or, S. Goldwasser, and A. Wigderson, "Completeness theorems for non-cryptographic fault-tolerant distributed computation," in *Proceedings of the 20th Annual ACM Symposium on Theory of Computing*, 1988.

7. E. Kushilevitz and Y. Lindell, *Introduction to Secure Multiparty Computation*. Springer, 2020.

8. R. Canetti, "Universally composable security: A new paradigm for cryptographic protocols," in *Proceedings of the 42nd IEEE Symposium on Foundations of Computer Science*, 2001, pp. 136-145.

9. A. Shamir, "How to share a secret," *Commun. ACM*, vol. 22, no. 11, pp. 612–613, Nov. 1979.

10 I. Damgård, M. Fitzi, E. Kiltz, J. B. Nielsen, and T. Toft, "Unconditionally secure constant-round multi-party computation for equality, comparison, bits and exponentiation," in *Theory of Cryptography*, 2006, pp. 285–304.

11. D. Beaver, S. Micali, and P. Rogaway, "The round complexity of secure protocols," in *Proceedings of the 22nd ACM Symposium on Theory of Computing*, 1990, pp. 503–513.

12. A. C. Yao, "How to generate and exchange secrets," in *27th Annual Symposium on Foundations of Computer Science*, 1986, pp. 162-167.

13. T. Rabin and M. Ben-Or, "Verifiable secret sharing and multiparty protocols with honest majority," in *Proceedings of the Twenty-First Annual ACM Symposium on Theory of Computing*, 1989, pp. 73–85.

14. A. Beimel, "Secret-sharing schemes: A survey," in *International Conference on Coding and Cryptology*, Springer, 2011, pp. 11–46.

15. M. Freedman, K. Nissim, and B. Pinkas, "Efficient private matching and set intersection," in *Advances in Cryptology–EUROCRYPT*, 2004, pp. 1–19.

16. J. Katz and Y. Lindell, *Introduction to Modern Cryptography*, 2nd ed., CRC Press, 2014.

17 . Y. Ishai, E. Kushilevitz, R. Ostrovsky, and A. Sahai, "Zero-knowledge from secure multiparty computation," in *Proceedings of the Thirty-Ninth Annual ACM Symposium on Theory of Computing*, 2007.

18. S. Micali and P. Rogaway, "Secure computation," in *Advances in Cryptology–CRYPTO*, 1991.

19. V. Goyal, P. Mohassel, and A. Smith, "Efficient two party and multi-party computation against covert adversaries," in *Advances in Cryptology–EUROCRYPT*, 2008.

20. C. Hazay and Y. Lindell, *Efficient Secure Two-Party Protocols: Techniques and Constructions*. Springer, 2010.

21. A. C. Yao, "Garbled circuits: An overview," *Foundations and Trends in Theoretical Computer Science*, vol. 13, no. 2-3, pp. 111–170, 2018.