National Centre of Excellence
CYBERSECURITY TECHNOLOGY AND ENTREPRENEURSHIP

इलेक्ट्रॉनिकी एवं सूचना प्रौद्योगिकी मंत्रालय
MINISTRY OF ELECTRONICS AND INFORMATION TECHNOLOGY
सत्यमेव जयते

DSCI
PROMOTING DATA PROTECTION
A nasscom Initiative

Securing Data in Distributed Systems

The SMPC Handbook

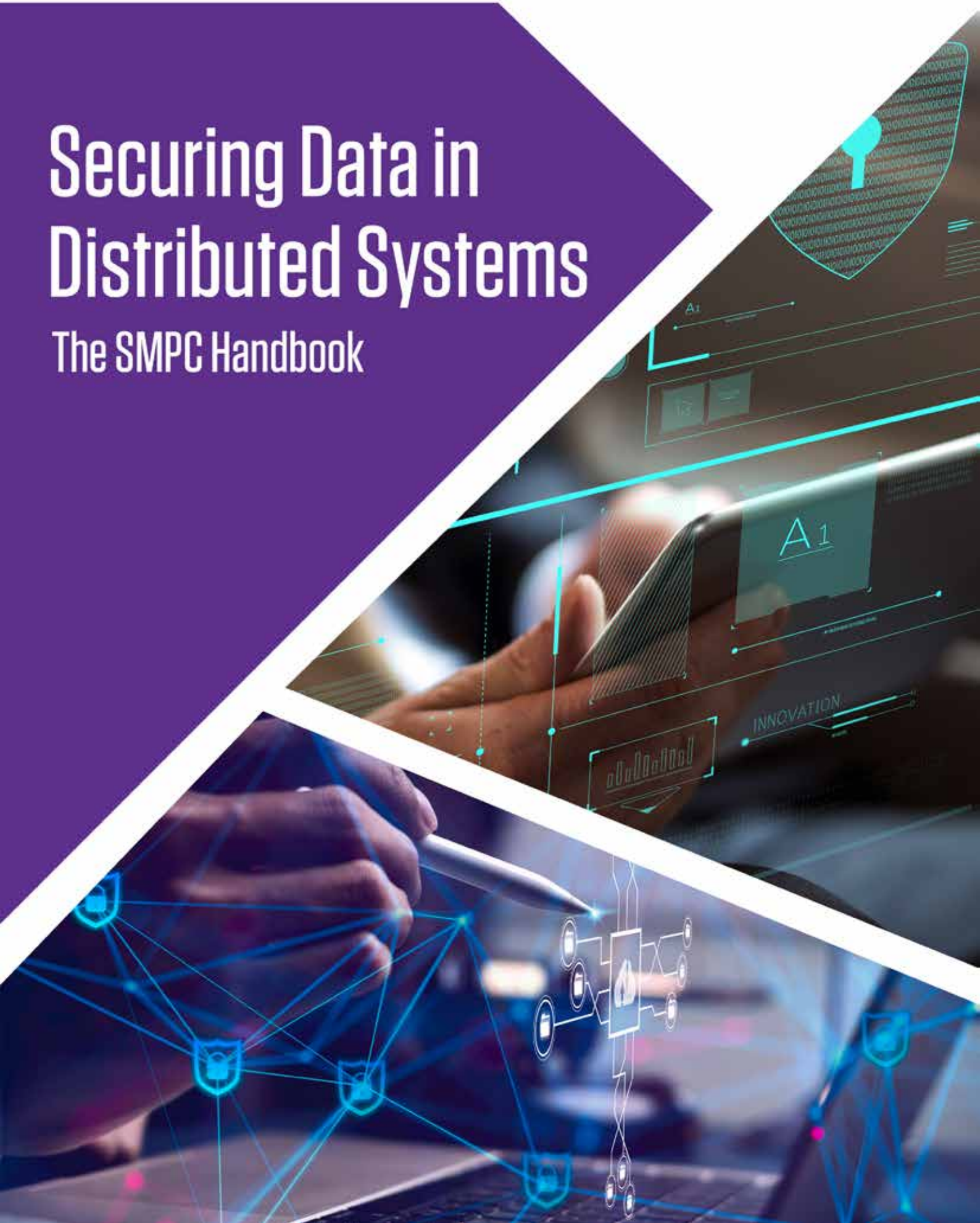


Table of CONTENTS

1

Executive Summary

The Secure Multi-Party Computation (SMPC) is one of the most important innovations in cryptography and privacy technologies in recent decades. Simply put, it enables multiple parties to execute a joint computation on their pooled datal like trend analysis, statistical computation, or running data-sim models—while each party's raw data remains concealed from the others. SMPC safeguards input privacy of all parties, even in the presence of malicious participants. This is executed by ensuring each party's trust as well as computation. Over the years, SMPC has transformed from just a theory to a set of practical tools and frameworks that are now being used by governments, organisations, and research institutes globally.

This handbook has two main audiences. The first are the people who are not familiar with SMPC but want an introduction to the basic ideas and potential real-world applications. And the second are the people who, although familiar with cryptography, want to dive deeply into the protocols, how to implement them, and current research trends. Processing data in a distributed way without a trusted party to facilitate the interactions, or having the parties directly share their data, is an important milestone for cybersecurity and privacy. Organizations can now gain insights that used to be accessible only through fully central data collection, which raised security and regulatory issues. SMPC go around these issues by

solving the problem of data protection during computation, leading to some groundbreaking use cases in finance, healthcare, advertising, machine learning, etc. This handbook covers the entire landscape of SMPC – its history, cryptographic building blocks, latest protocols, case-studies etc – in its chapters. The information provided also includes recommended practices, frequent mistakes, rules and regulations and new trends such as quantum-resistant SMPC and privacy-preserving AI. In the end, they will cover how SMPC can be integrated into various distributed systems so that secure, privacy-preserving collaboration can scale.

1.1 Purpose and Scope

The purpose of this handbook is to provide a structured, in-depth exploration of Secure Multi-Party Computation in the context of distributed systems. We address both conceptual underpinnings and hands-on practicalities. Readers can expect to learn:

- **Key Principles and Motivation**: Why SMPC exists and the essential problems it aims to solve in the realm of data privacy.
- **Technical Foundations**: The cryptographic primitives—secret sharing, oblivious transfer, homomorphic encryption, and more—that underpin SMPC.
- **Core Protocols**: Classic and advanced SMPC techniques, their design rationales, and performance trade-offs.
- **Implementation Insights**: Guidance on selecting frameworks, optimizing performance, and balancing security requirements with operational constraints.
- **Use Cases and Regulatory Aspects**: Real-world scenarios where SMPC has been successfully deployed, along with legal and ethical considerations.
- **Future Trends**: Emerging themes like quantum-safe SMPC, integration with blockchain, differential privacy combinations, and next-generation hardware support.

While the handbook provides substantial technical details, it is also designed to be accessible. Our aim is to help readers develop a robust mental model for how SMPC fits into modern distributed systems, what types of computational tasks it can handle, and how to plan for its integration and maintenance. We avoid limiting the discussion to purely theoretical aspects; instead, we include examples, diagrams, and guidelines that highlight how SMPC operates in practice.

Readers with a general background in security or distributed computing should find the discussion approachable, while cryptography specialists will appreciate the deeper dives into protocol mechanics and optimization strategies.

2

Introduction to Secure Multi-Party Computation (SMPC)

2.1 Historical Context and Evolution

The original paper that brings SMC into discussion was written in 1986 by Andrew C. Yao, "Protocols for Secure Computations". In his paper, he establishes the first two-party computation protocol which aptly works through a fundamental notion called "Oblivious Transfer". Andrew Yao, proposed this, whom we now refer to as Yao's Garbled Circuits. Subsequently, Yao's Garbled Circuits served as the foundation for a two-party protocol to compute a function f over their joint inputs while keeping their inputs secret.

The SMPC idea came From a beautiful puzzle called the "Millionaires' Problem." The problem asked how it is possible for two millionaires to find out who is richer without revealing their net worth.

After Yao, several researchers took Yao's work and extended it into the multiparty realm which resulted in some revolutionary protocols like BGW (Ben-Or, Goldwasser, and Wigderson) and GMW (Goldreich, Micali, and Wigderson). Through these early protocols, it was possible

to show that, in principle, any computable function could be securely executed as long as certain conditions, namely a bound on the number of colluding adversaries, holds.

Until recently, SMPC was primarily of interest to academics. This was because the protocols were either too computationally or communication-heavy to deploy in practice. But, as public-key cryptography got better, and computing power grew, researchers and experts began to tweak SMPC protocols. A combination of new techniques like Oblivious Transfer extensions, free-XOR in garbled circuits, better secret sharing and many other things were turning theory into practice. SMPC is becoming more important today because privacy has become very important due to regulatory pressures and increased public awareness.

Organizations in finance, tech, healthcare, and government are all looking at or already using SMPC-based solutions to analyze data without compromising privacy. This shift shows that SMPC has come a long way and is being recognized as a transformative tool for privacy.

2.2 Definition and Core Principles

SMPC, in essence, enables a group of parties, each possessing private inputs, to collectively compute a function on all those inputs. An extremely valuable characteristic is that no entity learns anything regarding the other entities' inputs, apart from what the final output reveals. To put it differently, five companies want to jointly compute a number using their own private data. Maybe they want to compute the sum, average or even a complex machine learning algorithm on all the data but without sharing any actual data with the other parties.

Four fundamental principles define SMPC's essence:

1. **Confidentiality**: Each party's input remains secret from all other parties.
2. **Correctness**: The protocol ensures that the computed result is accurate, matching the same result that would have emerged if a trusted third party had performed the computation.
3. **No Third-Party Requirement**: The computation is distributed. All trust is placed in the protocol's cryptographic guarantees rather than in any single external entity.
4. **Minimal Leakage**: Nothing is leaked except what can be inferred from the legitimate output. If the function's output is a single number, that number is all that any participant (or colluding group) learns, even if they exchange notes.

These principles collectively differentiate SMPC from simpler data-sharing or encryption schemes. Traditional encryption can protect data at rest or in transit, but once data needs to be processed, it is typically decrypted in a single location. SMPC extends security into the "in use" phase, ensuring data remains protected throughout the entire computation.

2.3 Importance in Modern Cybersecurity and Privacy

In an era of big data and regulatory scrutiny, the ability to collaborate on sensitive information without disclosing it outright is enormously valuable. Conventional models of

data collaboration require either trusting a central aggregator with complete visibility or distributing data with partial redaction—approaches that often fail security or compliance checks. SMPC addresses these pitfalls head-on: it removes the single point of vulnerability by distributing trust and ensures compliance by strictly limiting data exposure.

Several industries especially benefit:

- **Finance**: Institutions can do joint risk analysis or fraud detection without revealing confidential client records.
- **Healthcare**: Hospitals can use an aggregator to pool patient data for research into diseases or treatments without violating privacy laws or compromising patient confidentiality.
- **Advertising and Marketing**: Platforms and advertisers can determine campaign effectiveness or shared user bases without direct data leaks, critical in a privacy-conscious market.
- **Government and Public Sector:** overnment agencies can coordinate threat intelligence or carry out statistical analyses without breaching rules of jurisdictional data sharing.

As new technologies such as machine learning become integrated into decision making, SMPC is required more than ever. Machine learning systems require extensive and varied data

for effectiveness. SMPC-powered solutions allow parties to collaboratively train or evaluate machine learning models without revealing specific training or test datasets to one another.

2.4 Common Misconceptions and Limitations

Despite its promise, SMPC is sometimes misunderstood. People often think that "encryption for data in use" is simply another way of saying fully homomorphic encryption. Even though homomorphic encryption can do computations on ciphertexts, it can be thought of as a fully secret from a single party, SMPC involves more than one party communicating to keep a value secret. Another misconception is that SMPC is always slow. Through the years it had a significant overhead, today's protocols and hardware optimizations can make SMPC much quicker, though there is still some overhead to plain computation. It's also important to note.

It's also important to note:

- **Scalability**: There are many protocols that scale well for moderate numbers of parties, but large-scale multiparty scenarios present challenges both in terms of bandwidth usage and cryptographic complexity.
- **Output Leakage**: The final result itself might reveal partial information. The SMPC assures no other leakage besides the output. However, if the output is too revealing, then privacy is still compromised.
- **Complexity of Setup**: Some advanced protocols need a trusted setup phase or correlation generation. Handling these setup requirements can be tricky in large or open environments.
- **Regulatory Nuances**: While SMPC is very privacy-centric, different jurisdictions have varied interpretations of what "data sharing" means. SMPC mostly matches with privacy laws, but organizations should check for compliance with local regulations.

By understanding these nuances, organizations will use SMPC more responsibly. When carefully employed, there are tremendous benefits to its use, but success requires awareness of the prasctical trade-offs.

3

Cryptographic Foundations

3.1 Secret Sharing

Secret sharing is a fundamental building block in Secure Multi-Party Computation (SMPC) protocols, enabling distributed computations while preserving data privacy. The most basic approach is **Additive Secret Sharing,** where a secret value sss is divided into multiple shares $s\_{1}, s\_{2}, \ldots, s\_n$ such that:

$$s = s\_{1} + s\_{2} + \ldots + s\_n$$

This operation occurs within a finite field or integer ring. Each share is distributed to a different participant, ensuring that no individual share reveals any information about the secret. The original secret can only be reconstructed by combining all shares or a sufficient subset, depending on the scheme.

A more robust technique is **Shamir's Secret Sharing**, which provides threshold-based reconstruction by encoding the secret within a polynomial of degree ttt. This scheme follows a (t+1)-of-n

structure, where the secret can only be reconstructed if at least (t+1) shares are combined. Crucially, any subset of "t" or fewer shares provides absolutely no information about the

secret. This threshold mechanism is particularly valuable in scenarios where some participants may collude or become compromised.

Linear operations are efficiently supported by secret sharing. When participants hold shares of two values x and y , they can independently add their shares to produce shares of x+y with zero communication overhead. However, performing **multiplication** between shared values is more complex. It typically requires specialized protocols involving the generation of **random "triplets"** or other advanced cryptographic techniques to securely combine partial information without compromising the secret.

3.2 Homomorphic Encryption

Homomorphic Encryption (HE) allows computations on encrypted data. When you decrypt the result, it's as though the computation happened on the plaintext. Fully Homomorphic Encryption supports arbitrary computations, but even partial or somewhat homomorphic encryption can be powerful in SMPC contexts:

- **Additive Homomorphic Encryption**: Allows adding two ciphertexts to get a ciphertext of the sum of their plaintexts. Paillier encryption is a common example.
- **Multiplicative Homomorphic Encryption**: Allows multiplying ciphertexts to get an encryption of the product of plaintexts. RSA can be seen as multiplicative homomorphic for certain operations.

Although fully homomorphic schemes are often slower and more complex, partial homomorphic encryption can integrate into SMPC as a building block, especially for tasks heavy in additions or limited multiplications. Some modern protocols combine secret sharing with homomorphic encryption to reduce communication or offline overhead.

3.3 Oblivious Transfer and Garbled Circuits

Oblivious Transfer (OT) is a fundamental cryptographic primitive that underpins many SMPC protocols. In a typical 1-out-of-2 OT, a sender has two messages, and the receiver wants one of them—but the receiver must not learn the other message, and the sender must remain oblivious to which message was chosen. This seemingly paradoxical ability forms the basis for private input selection in two-party protocols, especially Yao's Garbled Circuits.

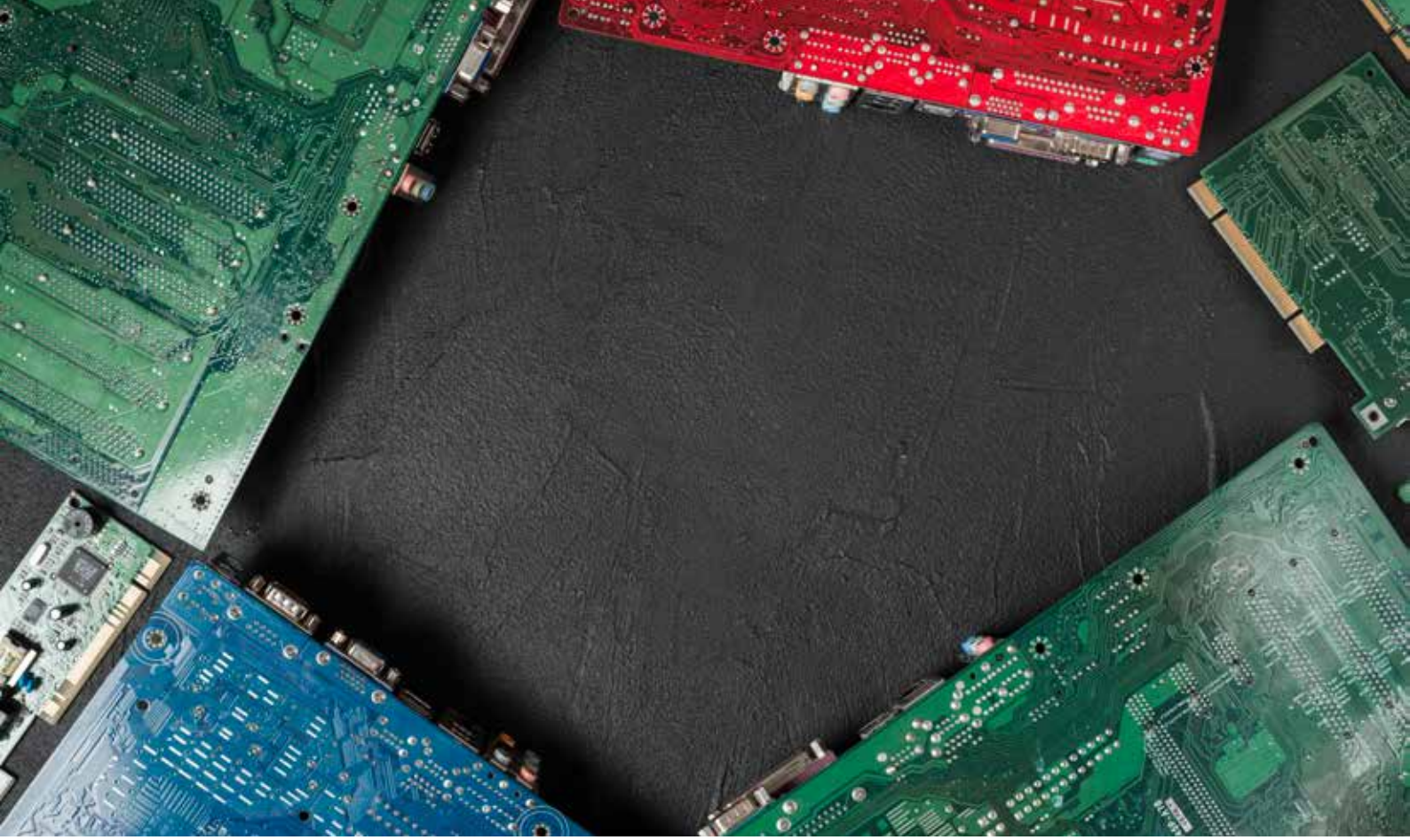
Example :

In a typical 1**-out-of-2 Oblivious Transfer,** there are two parties:

- The **Sender (Rahul)**, who has two confidential files: **A (Stock Market Report)** and **B (Economic Analysis)**.
- The **Receiver (Priya)**, who wants to access **only one of the files**, say **File A.**

The critical requirements of OT are:

- **Rahul should remain unaware** of which file Priya chose.
- **Priya should only receive the file she wants (File A)** without learning anything about the other file **(File B)**.



Yao's Garbled Circuits is a cornerstone technique for two-party computation. The "garbler" transforms a desired logical or arithmetic circuit into an encrypted form, giving "garbled tables" to the "evaluator." Each possible wire value (0 or 1 in a Boolean setting) corresponds to a random cryptographic label. During circuit evaluation, the evaluator uses OT to retrieve only the labels corresponding to its input bits. When the evaluator processes each gate, it can decrypt precisely one label for the output without learning anything about the other possibilities. This protocol ensures that neither party sees the other's raw inputs, and only the final result is revealed.

Example: Comparing AI Models (Company A and B)

Imagine two companies, A and B, want to compare the performance of their AI models to see which one is more accurate. However, neither party wants to reveal the inner workings of their models.

Here's how Yao's Garbled Circuits makes it possible:

1. **Garbler (A):**
    - **Builds a computational circuit** that represents the comparison process.
    - **Encrypts the circuit,** converting it into a series of encrypted tables known as **garbled tables.**
    - For every possible input value (0 or 1 in a Boolean setting), it generates **random cryptographic labels.**
    - Shares these encrypted tables with **Infosys (the Evaluator).**

2. **Evaluator (B):**
    - Uses **Oblivious Transfer (OT)** to receive only the labels that correspond to their own input.
    - Processes the encrypted circuit gate-by-gate, decrypting only the relevant labels at each step.
    - Computes the final result without learning TCS's inputs or revealing its own.

3.4 Zero-Knowledge Proofs

- Zero-knowledge proofs (ZKPs) allow one party to show knowledge of a fact (a secret key, the validity of a statement etc.) while not revealing anything else. In SMPC, ZKPs are used for.
- **Verifying Correct Behavior**: Making sure parties stick to the protocol in an honest way, especially in malicious security models.
- **Proving Data Integrity**: Demonstrating that certain inputs conform to rules (e.g., non-negative values, membership in a set) without disclosing the data itself.

ZKPs could be integrated into SMPC protocols to check for cheating or consistency. One example is a participant might be able to prove that the shares they provide correspond to a valid secret without exposing the secret. ZKPs may add computation but are very helpful to secure a hostile or untrusted environment robustly.

3.5 Threat Models

Understanding SMPC also means understanding who might behave dishonestly and how. Common threat models include:

- **Semi-Honest (Honest-But-Curious)**: situation where the parties follow the protocol's steps correctly but attempt to get extra information from the data they get.
- **Malicious**: Parties can deviate arbitrarily from the protocol, lie about inputs, or even refuse to send certain messages, aiming to breach secrecy or sabotage results.
- **Honest Majority vs. Dishonest Majority**: To ensure a solid level of security, protocols often require that less than half the parties are corrupted (an honest majority). Some protocols consider the possibility that everyone except at most one party might be corrupt.
- **Collusion**: A subset of parties might pool their information or shares to glean insights. Protocols specify the maximum colluding subset that can be tolerated without breaking security.

Selecting or designing an SMPC protocol typically involves balancing these threat models against performance constraints. Stronger security often entails higher computational or communication overhead. By specifying which model applies, implementers can choose or tailor protocols best suited for their environment.

4

Classical Protocols (BGW, GMW, Yao's Garbled Circuits)

4.1 Classical Protocols (BGW, GMW, Yao's Garbled Circuits)

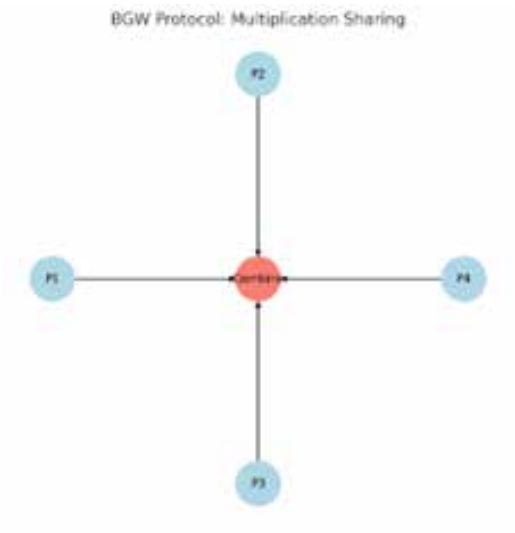
Early classical protocols remain highly influential:

- **BGW (Ben-Or, Goldwasser, Wigderson)**: This protocol relies on Shamir secret sharing in an honest-majority model. Each gate in a computation is securely computed by parties who combine shares to produce a new shared output. BGW's advantage lies in its information-theoretic security—if fewer than half the parties are corrupt, no computational assumption can break it.

Working Principle:

- **Input Sharing**: Inputs are split using simple XOR-based secret sharing, where each party receives a random bit, and their XOR gives the original input.
- **Addition & XOR Operations**: These operations are free (no communication required) because parties locally XOR their shares.

- **Multiplication (AND) Operations:**
    - For each AND gate, parties engage in **Oblivious Transfer (OT)** to securely compute the output without revealing their inputs.
    - This requires interaction between parties, making it **communication-intensive.**
- **Round Complexity**: The communication complexity is proportional to the **depth of the circuit,** making the protocol less efficient for deep circuits.
- **Security**: Can work under the dishonest-majority model but at the cost of increased communication.

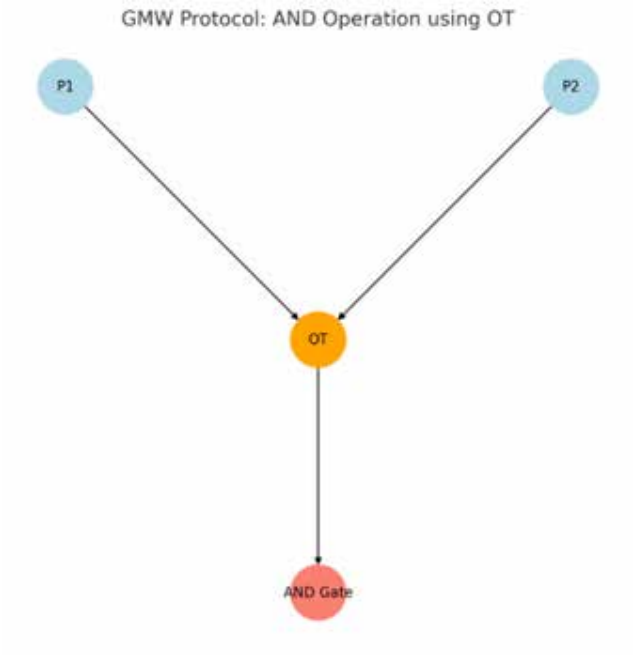


- **GMW (Goldreich, Micali, Wigderson)**: GMW uses bitwise secret sharing of inputs and employs Oblivious Transfer for AND gates in a Boolean circuit. It can work under dishonest-majority assumptions but typically requires multiple rounds of communication proportional to the circuit depth.

Working Principle:

1. **Input Sharing**: Inputs are split using simple XOR-based secret sharing, where each party receives a random bit, and their XOR gives the original input.
2. **Addition & XOR Operations**: These operations are free (no communication required) because parties locally XOR their shares.
3. **Multiplication (AND) Operations**:
    - For each AND gate, parties engage in **Oblivious Transfer (OT)** to securely compute the output without revealing their inputs.
    - This requires interaction between parties, making it **communication-intensive.**
4. **Round Complexity**: The communication complexity is proportional to the **depth of the circuit**, making the protocol less efficient for deep circuits.

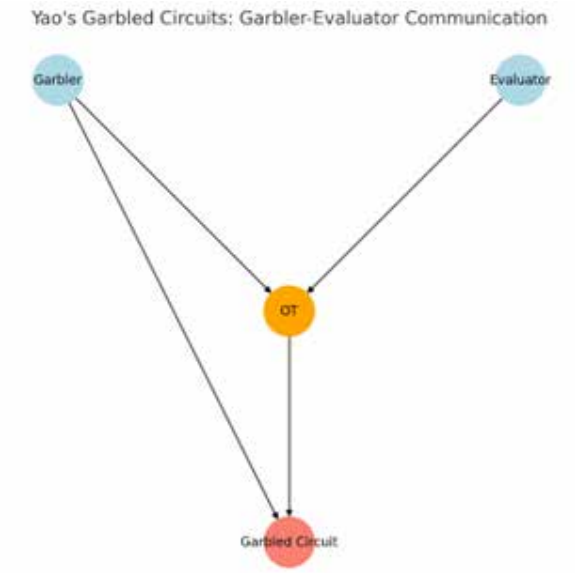
5. **Security**: Can work under the **dishonest-majority model** but at the cost of increased communication.



- **Yao's Garbled Circuits**: Ideal for two-party settings, it achieves constant-round protocols independent of circuit depth, which is beneficial in high-latency networks. One party "garbles" the circuit, and the other evaluates it, obtaining only the result.

Working Principle:

1. **Garbler (Party A)**:
    - Constructs the computation circuit as a set of logic gates (AND, OR, NOT, etc.).
    - **Encrypts (Garbles)** each gate by associating each wire value (0 or 1) with a random cryptographic label.
    - Creates a **garbled table** for each gate, where only the correct combination of labels can produce a valid output label.
2. **Evaluator (Party B)**:
    - Uses **Oblivious Transfer (OT)** to obtain the garbled labels corresponding to its own inputs without revealing them to the Garbler.
    - Evaluates the garbled circuit using the garbled tables, producing the final result without learning the inputs of the Garbler.
3. **Security**: Only the final result is revealed to the evaluator. Neither party learns the other's inputs, ensuring privacy.
4. **Efficiency**: The number of communication rounds is independent of the circuit depth, making it very efficient for complex computations.



Comparison

Protocol	Security Model	Efficiency	Communication Overhead	Suitable For
BGW	Honest-Majority	Efficient for shallow circuits	High (For multiplication)	Information-theoretic security
GMW	Dishonest-Majority	Requires multiple rounds	High (For AND operations)	Arbitrary circuits
Yao's Garbled Circuits	Two-Party Setting	Constant-round, independent of depth	Low (Efficient evaluation)	Two-party computations

Each classical protocol has different trade-offs in terms of round complexity, communication overhead, and assumptions about corruption. Even decades after their invention, these protocols serve as the foundation of many optimized or hybrid SMPC solutions.

4.2 Advanced Protocols (SPDZ, ABY, MASCOT)

Modern SMPC has seen the rise of sophisticated protocols that improve efficiency, especially in the dishonest-majority setting:

- **SPDZ**: A family of protocols designed for actively secure multiparty computation without assuming an honest majority. It uses secret sharing with precomputed "Beaver triples" for efficient multiplication. A unique feature is the use of information-theoretic Message Authentication Codes (MACs) to prevent cheating. SPDZ splits the computation into an offline phase (heavy cryptography) and an online phase (fast operations using precomputed data).
- **ABY (Arithmetic, Boolean, Yao)**: A framework that can dynamically switch between arithmetic secret sharing, Boolean GMW, and Yao's garbled circuits. This lets users exploit whichever representation is most efficient for particular operations—arithmetic for sums, Boolean for bit-level conditions, and so forth.

- **MASCOT**: Focuses on generating multiplication triples more efficiently using Oblivious Transfer extension. MASCOT is used within SPDZ-like protocols to speed up the offline phase and thereby improve overall throughput.

These advanced protocols push SMPC closer to practical deployment for a wide range of real-world tasks, offering better performance while preserving strong security guarantees.

4.3 Synchronous vs. Asynchronous Approaches

SMPC protocols typically assume at least partial synchrony in communication—i.e., messages arrive within a reasonable timeframe. However, network delays or failures can disrupt computations. Two broad categories emerge:

- **Synchronous Protocols**: Assume parties operate in lockstep or well-defined rounds. This simplifies protocol design but can stall if a party becomes slow.
- **Asynchronous Protocols**: Tolerate varying delays, continuing the computation as partial messages arrive. Asynchronous protocols are more complex but can remain robust against network hiccups or partial failures.

For many real-world deployments, partial synchrony is assumed—participants generally communicate in real-time, but the protocol can handle occasional slow delivery. Fully asynchronous solutions exist but tend to have higher overhead due to the complexities of handling unpredictable latencies and potential reorderings of messages.

4.4 Communication Overhead vs. Efficiency Trade-Offs

A central challenge in SMPC is the tension between communication and computational overhead:

1. **Circuit Depth vs. Rounds**: Protocols like GMW require one round of interaction per circuit layer, making them potentially expensive for deep circuits but with minimal per-gate overhead.
2. **Garbled Circuit Size**: In Yao's scheme, each gate typically has multiple encrypted values. This can lead to large data transfers but only a small, constant number of communication rounds.
3. **Preprocessing vs. Online Computation**: Many advanced protocols (e.g., SPDZ) split computation into heavy offline phases and lightweight online phases. This approach can drastically reduce online response time, but the offline phase needs careful planning and resource allocation.
4. **Security Level**: Malicious security requires extra checks, typically increasing both computation and communication compared to semi-honest scenarios.

When deciding on a protocol, practitioners must evaluate the problem size, type of computation (arithmetic vs. Boolean heavy), latency constraints, and adversarial threat model. Often, hybrid or mixed protocols can adapt to the structure of the function, minimizing overhead in practice.

5

Practical Implementation Considerations

5.1 Performance Metrics (Computation, Communication, Latency)

Before deploying SMPC in a production environment, it is crucial to define clear performance metrics. Three primary metrics dominate discussions:

- **Computation Time**: The CPU or GPU cycles required. This includes cryptographic operations like encryption, decryption, multiplication of secret-shared values, and any zero-knowledge proof generation or verification.
- **Communication Volume**: The total data exchanged across the network. Large garbled circuits or repeated Oblivious Transfers can generate significant bandwidth consumption.
- **Latency (Round Complexity)**: The number of interactive rounds required. Each round typically involves waiting for all parties to send and receive messages. High network latency can become the bottleneck, overshadowing raw computational speed.

Balancing these factors often leads to trade-offs. For instance, a protocol might minimize total communication at the expense of more interaction rounds, or vice versa. The desired outcome often depends on network conditions (local data center vs. global distribution) and system constraints (hardware availability, cost of bandwidth, etc.).

5.2 Scalability Challenges in Distributed Systems

SMPC typically scales well when the function's complexity grows in terms of data size, provided the number of participating parties remains fixed and small. But real-world distributed systems can involve numerous parties:

- **Linear or Quadratic Complexity in Number of Parties**: Some protocols require pairwise operations (like pairwise OT), leading to complexity that grows with n2n^2n2.
- **Honest Majority vs. Dishonest Majority**: In honest-majority protocols, more parties can sometimes enhance security and possibly reduce the complexity of certain checks. But dishonest-majority protocols often become costlier as the group size increases.
- **Data-Parallel vs. Party-Parallel**: If thousands of data records are distributed among a small set of compute nodes, SMPC is easier. If thousands of distinct parties each hold data, overhead can become substantial.

Engineers must often limit the number of active SMPC participants to remain within feasible computation and communication bounds. In practice, solutions might adopt an architecture with a few computing servers that internally represent data owners, each server using secret sharing or partial homomorphic encryption on behalf of multiple data contributors.

5.3 Security vs. Efficiency Balancing

SMPC always involves a balance between the desired level of security and acceptable overhead. If participants are known and trusted to follow protocol instructions (semi-honest model), overhead is markedly lower than in the fully malicious model. Malicious security adds:

- **Consistency Checks**: Techniques like cut-and-choose in Yao's Garbled Circuits or MAC-based verification in SPDZ.
- **Zero-Knowledge Proofs**: Additional computational steps to prove correctness of shares or gate evaluations.
- **Increased Communication**: More messages to broadcast or exchange partial proofs and commitments.

Organizations should carefully assess real-world adversarial risks and weigh them against performance demands. For instance, if participants are bound by strict legal contracts and face severe penalties for cheating, a semi-honest solution might suffice. If participants' incentives to cheat are high and detection alone is insufficient, malicious security is prudent.

5.4 Hardware Acceleration (GPUs, FPGAs, TEEs)

Because SMPC involves numerous cryptographic operations, hardware acceleration can dramatically speed up computations:

- **GPUs** excel at parallel tasks such as large matrix multiplications and hashing. Protocols that rely on repeated symmetrical operations can offload them to GPU kernels.
- **FPGAs** can be tailored to specific cryptographic primitives, achieving high throughput with lower power consumption once the hardware logic is designed.
- **Trusted Execution Environments (TEEs)** like Intel SGX or ARM TrustZone can run computations in an isolated enclave. In some hybrid approaches, TEEs reduce SMPC overhead by locally handling partial data, but complete trust in hardware remains a consideration.

Utilizing hardware acceleration effectively requires specialized libraries and sometimes low-level coding. Nonetheless, as SMPC frameworks mature, they increasingly offer GPU and FPGA modules to exploit parallel cryptographic routines.

5.5 Available Frameworks and Libraries (MP-SPDZ, Sharemind, EMP)

Several frameworks help developers integrate SMPC without having to re-implement core protocols:

- **MP-SPDZ**: Supports a wide range of protocols (SPDZ variants, semi-honest, malicious, honest-majority, etc.) and provides a high-level language for writing secure computations. It includes offline/online phases and can leverage hardware acceleration.
- **Sharemind**: Focuses on three-party honest-majority settings. It has been used in real deployments for privacy-preserving analytics, especially with its user-friendly SecreC language.
- **EMP-Toolkit**: A collection of C++ libraries offering optimized building blocks for Yao's garbled circuits, GMW, and Oblivious Transfer. It's well-suited for developers comfortable with low-level or mid-level cryptographic programming.

Other frameworks (like ABY, SCALE-MAMBA, or commercial SDKs) offer unique advantages. Choosing a library depends on trust assumptions, the number of parties, performance needs, and developer familiarity.

6

Real-World Use Cases and Case Studies

6.1 Finance (Secure Auctions, Confidential Data Exchange)

Financial institutions often need to compute joint risk metrics, detect fraud spanning multiple organizations, or perform auctions without disclosing sensitive bids. SMPC supports:

- **Secure Auctions**: Multiple bidders can submit secret bids in an SMPC protocol. The highest bid (and possibly second price) is revealed without exposing losing bids or the complete ordering. This is invaluable for procurement auctions or sensitive corporate auctions.
- **Cross-Bank Fraud Detection**: Banks can privately check if a single customer or account is conducting suspicious transactions across multiple institutions. Each bank retains confidentiality over its internal data while collaboratively identifying potential laundering or fraud.
- **Benchmarking**: Banks or hedge funds might collectively compute industry benchmarks (e.g., average portfolio risk) without revealing individual holdings.

Such applications highlight how SMPC eliminates a central data aggregator, preserving institutional confidentiality while benefiting from pooled intelligence.

6.2 Healthcare (Privacy-Preserving Patient Data Analysis)

Healthcare data is extremely sensitive and heavily regulated. SMPC empowers hospitals, clinics, or researchers to collaborate on:

- **Clinical Studies**: Multiple hospitals can combine patient records to investigate disease prevalence, treatment outcomes, or genetic correlations, all without exposing raw data.
- **Genetic Data Analysis**: Genomic databases can remain encrypted or secret-shared while researchers compute aggregate statistics or identify shared genetic markers.
- **Pandemic Response**: Public health agencies might securely aggregate infection or vaccination data across different jurisdictions. SMPC ensures no single agency ever sees individual-level data from another.

In these scenarios, SMPC satisfies both the scientific need for large datasets and the legal requirement to maintain strict confidentiality.

6.3 Advertising Technology (Attribution Without Data Leakage)

Modern digital advertising thrives on precise attribution—understanding which ads led to conversions. But privacy regulations, browser restrictions, and user expectations hamper direct data sharing between advertisers, publishers, and platforms. SMPC solutions include:

- **Conversion Lift Studies**: An advertiser and platform share user data in encrypted or secret-shared form to compute how many users who saw an ad eventually purchased, without revealing personal details.
- **Customer Overlap Analysis**: Two businesses can compute the intersection of their customer lists (for co-marketing or partnerships) securely. Only the intersection count or hashed identifiers for matching customers is revealed, preserving privacy for the rest.

This approach maintains confidentiality while preserving crucial marketing insights, offering an alternative to invasive tracking techniques.

6.4 Machine Learning and AI (Federated Learning with SMPC)

Machine learning typically requires large, centralized datasets. SMPC can distribute the learning process:

- **Privacy-Preserving Model Training**: Multiple data owners (hospitals, banks, or corporations) use SMPC to aggregate gradients during training. No participant sees any other participant's raw data or intermediate updates that might reveal data patterns.
- **Secure Prediction Serving**: A user's input and a model's parameters remain secret while producing a prediction. This is especially relevant if the model itself is proprietary or the input is sensitive (e.g., medical images or financial details).

Combining SMPC with federated learning frameworks reduces data movement and ensures compliance with privacy regulations, opening new avenues for collaborative AI development.

6.5 Government and Defense (Inter-Agency Data Sharing)

Government agencies frequently hold complementary but siloed data. SMPC allows them to cooperate without violating legal restrictions:

- **Threat Intelligence**: Intelligence or law enforcement agencies can jointly identify cross-border threats or criminals whose information is split among multiple databases.
- **Statistical Data Integration**: National statistical offices can combine census, tax, and welfare data for improved policy insights. SMPC ensures no raw data leaks while deriving aggregated results.
- **Secure Voting and Census**: Protocols can be used for e-voting schemes, providing verifiable tallies without exposing individual votes.

In defense, the secrecy demands are particularly high, making SMPC appealing for multi-nation alliances that must share threat data yet maintain national security constraints.

7

Security, Privacy, and Compliance

7.1 Common Attack Vectors (Side-Channel Attacks, Rogue Participants)

While SMPC secures data in the cryptographic sense, attackers might resort to other tactics:

- **Side-Channel Attacks**: Observing execution times, power consumption, or memory access patterns to glean hidden information. Implementations must minimize data-dependent branching or ensure constant-time operations.
- **Rogue Participants**: A participant might inject erroneous inputs, or deviate from the protocol steps, to cause incorrect results or glean extra data. Malicious-secure protocols employ verification checks or zero-knowledge proofs to mitigate such sabotage.
- **Collusion**: If the protocol tolerates up to ttt corrupt parties, more than ttt participants colluding can break security by combining their shares or insider knowledge.

Awareness of these threats helps developers and organizations adopt robust coding practices, thorough testing, and careful protocol selection.

7.2 Regulatory Landscape (GDPR, CCPA, HIPAA,DPDPA)

Regulations worldwide increasingly emphasize user privacy and data protection:

- **GDPR (EU)**: Mandates data minimization, purpose limitation, and strict consent rules. SMPC, by not revealing raw data to other parties, can help companies meet "privacy by design" obligations.
- **CCPA (California)**: Focuses on consumer rights to know, delete, and opt out of data sales. SMPC ensures that even if data is used for aggregated analytics, no personal detail is exposed externally.
- **HIPAA (US Healthcare)**: Enforces health data confidentiality. SMPC can be used for multi-hospital studies without creating unauthorized disclosures of patient information.
- **DPDPA (India)**: Enacted in 2023, India's Digital Personal Data Protection Act introduces stringent requirements for consent management, data minimization, and accountability. Key considerations for SMPC:
    - **Consent-Centric Processing**: DPDPA requires explicit consent for data processing unless under legitimate uses (e.g., public interest). SMPC's architecture ensures no raw data is shared without participants' agreement, aligning with the law's emphasis on user control.
    - **Data Localization**: While DPDPA allows cross-border data transfers, it mandates safeguards for sensitive data. SMPC's cryptographic processing—where data need not leave jurisdictional boundaries physically—can simplify compliance.
    - **Purpose Limitation**: DPDPA restricts data usage to predefined purposes. SMPC's function-specific computation (e.g., only calculating aggregate trends) inherently limits data usage to agreed-upon objectives.
    - **Breach Accountability**: Organizations must report breaches to the Data Protection Board of India. SMPC's audit trails (via cryptographic transcripts) can help demonstrate protocol adherence during investigations.

Although SMPC generally aligns well with privacy requirements, each use case must be analyzed to ensure compliance, especially regarding consent and the final outputs that might still reveal sensitive patterns.

7.3 Ensuring Compliance via Cryptographic Guarantees

SMPC protocols inherently provide strong cryptographic guarantees. They can reinforce compliance by:

- **Data Minimization**: Only outputs agreed explicitly upon are revealed. This aligns with regulatory principles of collecting and revealing the minimum necessary data and DPDPA's focus on collecting only "as much data as needed."
- **Auditability**: Many SMPC frameworks produce cryptographic transcripts. If a party claims a breach occurred, auditors can verify if the protocol was followed properly. (critical for DPDPA's accountability mandates)

- **Anonymization**: Because inputs remain hidden, SMPC effectively anonymizes data during computation. Combined with other techniques like differential privacy, SMPC can provide multi-layered protection. (DPDPA's anonymization standards for reduced re-identification risks)

However, compliance also hinges on how the final output is handled. If the result is too granular (e.g., revealing single-record statistics), re-identification risks remain. Designing the function to produce safe outputs is thus critical.

7.4 Ethical Considerations in Privacy-Preserving Tech

Beyond legal obligations, ethical questions arise when using SMPC for large-scale data analysis:

- **Participant Consent**: Are all individuals whose data is included aware and in agreement with how results will be used?
- **Unintended Inferences**: Even if raw data is never revealed, certain aggregate outputs might enable inference attacks. Balancing SMPC with additional privacy safeguards like k-anonymity or differential privacy might be needed.
- **Equity and Fairness**: Machine learning computations on shared datasets can inadvertently perpetuate biases if the data or algorithms are skewed. SMPC does not inherently solve bias; it only hides the raw inputs.
- **Right to Explanation**: Some regulations require explaining decisions made by algorithms. With SMPC, the logic remains transparent (the function is known), but ensuring interpretability can still be a challenge.

Emphasizing ethical guidelines fosters public trust and sets a high standard for privacy-preserving solutions, going beyond mere technical compliance.

8

Best Practices and Implementation Roadmap

8.1 Assessing Organizational Readiness for SMPC

Before adopting SMPC, organizations should evaluate:

1. **Use Case Suitability**: Is there a genuine need for multi-party collaboration that cannot be handled by a centralized approach or simpler solutions?
2. **Data Sensitivity**: The higher the sensitivity and regulatory risk, the stronger the motivation for SMPC.
3. **Available Infrastructure**: SMPC can demand significant computing resources and reliable network setups. Organizations need to ensure they have or can provision suitable environments.
4. **Stakeholder Commitment**: Multiple parties must align on protocol choice, cryptographic parameters, and usage policies. A single party cannot unilaterally deploy SMPC if others are not on board.

Often, a pilot or proof-of-concept helps demonstrate feasibility and clarifies performance constraints, paving the way for more extensive production rollouts.

8.2 Deployment Blueprint (From Pilot to Production)

A typical SMPC deployment roadmap could involve:

1. **Proof-of-Concept**
    - Identify a contained, high-value use case.
    - Deploy a small-scale or open-source SMPC framework (e.g., MP-SPDZ) with a test dataset.
    - Measure performance, correctness, and ease of integration.
2. **Pilot Phase**
    - Expand the number of parties or volume of data.
    - Conduct thorough security audits.
    - Develop a user-friendly interface or an API so domain experts can supply computations or queries without cryptographic expertise.
3. **Production Deployment**
    - Move to optimized hardware solutions (GPU clusters or specialized servers).
    - Implement robust logging, monitoring, and fallback mechanisms for partial outages.
    - Formalize an operational model that covers data pre-processing, secret sharing procedures, and post-computation output handling.
4. **Scaling & Optimization**
    - Fine-tune protocol parameters and switch between protocols (e.g., ABY) to optimize for different function segments.
    - Possibly incorporate advanced techniques like offline/online splitting for heavily used computations.

Throughout these stages, organizations should maintain clear governance: define who can propose computations, who receives outputs, and how frequently computations occur.

8.3 Maintenance, Monitoring, and Incident Response

Once deployed, SMPC solutions require ongoing care:

- **Continuous Monitoring**: Track throughput, latency, and error rates. Anomalies might indicate protocol misuse or network issues.
- **Key Management**: Securely rotate keys used in OT or secret sharing. In multi-year deployments, cryptographic parameters may need periodic updates.

- **Incident Response**: Predefine an incident response plan in case of suspected collusion or data breach. Although SMPC protocols are robust, real-world systems can fail at integration points (like compromised endpoints).
- **Regular Audits**: Confirm that any changes or software patches do not introduce vulnerabilities. Assess compliance with evolving regulations and policies.

Maintaining an SMPC system can be more complex than a standard central database due to distributed responsibilities and cryptographic intricacies. A well-documented policy and automated checks help in sustaining reliability.

8.4 Lessons Learned from Industry Implementations

Organizations that successfully implement SMPC often share certain insights:

- **Pick the Right Tool**: Not all protocols are equally efficient for every workload. Some tasks thrive with garbled circuits, others with arithmetic sharing.
- **Plan for Overhead**: Even the most advanced SMPC solutions run slower than plaintext processing. Over-provisioning or parallelization can mitigate this.
- **User Experience Matters**: Domain experts (e.g., data scientists in healthcare) should not be forced to dive into cryptographic details. Abstraction layers and APIs are key.
- **Handle Edge Cases**: If the final computed output is extremely small or unique, it may inadvertently reveal details about a single party's input. Designing the function and output format to remain privacy-preserving is a nuanced task.
- **Trust Framework**: In some industries, a consortium approach may be needed, where third-party audits or neutral computing providers (e.g., cloud services) help guarantee no single entity dominates the protocol.

Through methodical planning, careful protocol selection, and a robust operational framework, many organizations have demonstrated that SMPC is both feasible and beneficial, especially for high-stakes, privacy-sensitive collaborations.

9

Future Directions and Emerging Trends

9.1 Post-Quantum SMPC

Cryptographic systems face a looming threat from quantum computing. Many standard techniques, including certain OT instantiations, rely on number-theoretic assumptions that quantum computers could break:

9.1.1 Quantum-Resistant Cryptographic Approaches

Researchers are exploring lattice-based, code-based, and other post-quantum approaches to oblivious transfer and homomorphic encryption. These solutions can be integrated into SMPC protocols, providing resilience against future quantum attackers. While often less efficient than classical methods, ongoing work aims to improve performance.

9.1.2 Migration and Performance Considerations

Organizations like healthcare might take on post-quantum cryptography so that long-term data confidentiality requirements are already met. However, the practical overhead is non-

trivial. Solutions that use a combination of methods or layered approaches will incur fewer costs. Here, for example, a part of the protocol remains classical, whereas only the most critical steps apply post-quantum technology.

9.2 Federated Learning & Generative AI

9.2.1 Privacy-Preserving Training on Large Models

As neural networks and other model architectures grow in complexity, the need to train them on broader datasets intensifies. SMPC-based federated learning can bring together data from disparate owners without compromising individual privacy. This approach can reduce the risk of data breaches and meet strict compliance rules.

9.2.2 Confidential Inference and Prompt Engineering

Beyond training, SMPC can safeguard inference services—where a user feeds private data to a proprietary model. Even generative AI models (like large language models) might leverage SMPC to handle private user prompts and produce outputs without storing or revealing sensitive content. This scenario requires specialized protocols that handle large parameter sizes efficiently.

9.3 Differential Privacy & SMPC

9.3.1 Adding Noise for Output Privacy

SMPC guarantees that the input values stay hidden when the computing parties perform the computation; however, in the end, sometimes final output may still expose sensitive values. We can add noise to the output, which can make sensitive information more difficult to learn through the output. Combining SMPC with differential privacy provides data confidentiality & anonymization for the inputs and outputs ensuring E2E protection.

9.3.2 Balancing Accuracy with Privacy Budgets

Incorporating differential privacy into an SMPC workflow requires careful budgeting of the noise parameter. If you add less noise, it could lead to re-identification. If you add more noise, it would compromise utility. Research is ongoing to build automation tools that can analyze the trade-off between a particular SMPC circuit's privacy and utility, ensuring robust privacy while giving useful output

9.4 Blockchain Integration & Confidential Computing

9.4.1 TEEs for Off-Chain MPC Acceleration

Blockchain ecosystems often require that the computations are verifiable publicly. However, storing or processing large amounts of data within the chain is not realistic. MPC solutions that take place outside blockchain networks (Off Chain -MPC)effectively provide verified outputs to execute smart contracts. They are sometimes combined with trusted execution environments to speed up heavy computations.

As DeFi and other blockchain applications rise, the need to conceal transactions and computations is growing. SMPC can act as a privacy layer to facilitate confidential trading, bidding or voting in DAOs (Decentralized Autonomous Organizations) without revealing all details to the entire blockchain.

9.5 Data Mesh & Data Clean Rooms

9.5.1 Secure Collaborative Analytics Across Organizations

A data mesh architecture treats data as a domain-oriented product. With SMPC, each domain can remain fully autonomous and private. Data owners can publish secure "analytics endpoints" rather than raw data, enabling cross-domain insights with minimal friction.

9.5.2 Governance and Regulatory Implications

"Clean rooms" are closed environments where data is combined under strict usage policies. SMPC-based data clean rooms provide even stronger guarantees in that data owners are assured that no raw data leaves their domain. Policymakers typically favor such strong privacy mechanisms though they still need to be able to check that only permitted computations occur.

10

Conclusion

Secure Multi-Party Computation has revolutionized the process through which organizations handle data-driven tasks, particularly in highly regulated or sensitive situations. SMPC eliminates the need for a single trusted third party or central data owner by distributing trust among multiple parties and employing cryptographic techniques to protect data during the entire computation process. Due to this shift, new kinds of cooperation become possible: banks can cooperate on risk models, healthcare providers can share insights on patients and advertisers can check their campaign performance without jeopardizing the privacy of individuals or institutions.

In distributed systems, SMPC stands apart as it eliminates the fundamental trade-off between the usefulness of data and its confidentiality. Often, sharing data requires suffering a painful consequence; that of exposing oneself to possible breaches. Alternatively, keeping the data to oneself entails losing out on potentially valuable insights. SMPC has a cryptographic solution that allows to analyse together without centralising or exposing data. It results in a system model that ensures data security at each step, allowing the parties involved to trust each other.

Effective SMPC implementation involves dealing with the difficulties of selecting a protocol based on the targeted threat model, optimizing the protocol for performance and

communication overhead, compliance, and more. Various classical protocols like BGW, GMW and Yao's Garbled Circuits are the foundation as this handbook has shown. Contemporary structures such as SPDZ, ABY, and MASCOT leverage these protocols to enhance efficiency and security for practical usage in the world. SMPC's scope can be further enhanced with the cooperation of hardware acceleration, improved cryptographic primitives, differential privacy, etc.From a practical standpoint, success with SMPC depends on methodical preparation:

- **Use Case Identification**: Focus on scenarios where multi-party data collaboration is valuable yet risky if data is revealed.
- **Pilot and Phased Rollout**: Start with controlled pilots to measure feasibility, then scale as organizational confidence grows.
- **Regulatory and Ethical Caution**: Ensure the final computed outcomes do not inadvertently breach privacy or legal guidelines.
- **Continuous Improvement**: Anticipate updates in cryptographic techniques, post-quantum considerations, and software frameworks.

In the future, SMPC might be a reliable source of data processing solution across the sectors. As federated learning and advanced AI models emerge, we see further innovation as SMPC protocols learn to efficiently compute larger and more complex functions. At the same time, SMPC in blockchain and privacy-enhanced data marketplace opens up a whole new world of decentralized computation with privacy. In the end, SMPC shows that cryptography can enable new paradighms . Rather than viewing data privacy as an obstacle, SMPC transforms it into an enabler of collaborative intelligence. By embedding security guarantees in distributed systems, SMPC can provide a scalable and global solution for the responsible and ethical use of data. SMPC's future is set to expand as organizations and research groups allocate resources toward improvements like faster algorithms, solid hardware integrations, standardized libraries, and more. SMPCs power will keep growing, ensuring that data-driven decision-making and privacy will not become a zero-sum game.

References

- Andrew Yao, “Protocols for Secure Computations,” Proceedings of the 23rd Annual Symposium on Foundations of Computer Science.
- Ben-Or, Goldwasser, Wigderson, “Completeness Theorems for Non-Cryptographic Fault-Tolerant Distributed Computation,” STOC 1988.
- Goldreich, Micali, Wigderson, “How to Play Any Mental Game,” STOC 1987.
- Beaver, “Efficient Multiparty Protocols Using Circuit Randomization,” CRYPTO 1991.
- Paillier, “Public-Key Cryptosystems Based on Composite Degree Residuosity Classes,” EUROCRYPT 1999.
- Damgård et al., “Practical Covertly Secure MPC for Dishonest Majority – Or: Breaking the SPDZ Limits,” ESORICS 2013.
- Sharemind Documentation, Cybernetica, available in the official developer portal.
- MP-SPDZ GitHub Repository, “Multi-Protocol SPDZ,” open-source collection of MPC protocols.
- EMP-Toolkit, “Efficient MultiParty Computation Toolkit,” open-source library from Yale.
- Lindell, “How to Simulate It – A Tutorial on the Simulation Proof Technique,” Tutorials on the Foundations of Cryptography.
- Schneider et al., “GMW vs. Yao? Efficient Secure Two-Party Computation with Low Depth Circuits,” FC 2015.
- Aly et al., “Scale-Mamba v1.14: Documentation and Protocol Description,” open-source library for SPDZ-like MPC.
- Burkhart et al., “SEPIA: Privacy-Preserving Aggregation of Multi-Domain Network Events and Statistics,” USENIX Security Symposium.
- Partisia and Unbound Tech official websites for commercial MPC offerings.
- Intel SGX Developer Guide, discussing Trusted Execution Environments for secure enclaves.

National Centre of Excellence
CYBERSECURITY TECHNOLOGY AND ENTREPRENEURSHIP

The National Centre of Excellence (NCoE) for Cybersecurity Technology Development has been conceptualized by the Ministry of Electronics & Information Technology (MeitY), Government of India, in collaboration with the Data Security Council of India (DSCI). Its primary objective is to catalyze and accelerate cybersecurity technology development and entrepreneurship within the country. NCoE plays a crucial role in scaling and advancing the cybersecurity ecosystem, with a focus on critical and emerging areas of security.

Equipped with state-of-the-art facilities, including advanced lab infrastructure and test beds, NCoE enables research, technology development, and solution validation for adoption across government and industrial sectors. By adopting a concerted strategy, NCoE aims to translate innovations and research into market-ready, deployable solutions—contributing to the evolution of an integrated technology stack comprising cutting-edge, homegrown security products and solutions.

DSCI
PROMOTING DATA PROTECTION
A nasscom Initiative

Data Security Council of India (DSCI) is a premier industry body on data protection in India, setup by nasscom, committed to making the cyberspace safe, secure and trusted by establishing best practices, standards and initiatives in cybersecurity and privacy. DSCI brings together governments and their agencies, industry sectors including ITBPM, BFSI, telecom, industry associations, data protection authorities and think-tanks for policy advocacy, thought leadership, capacity building and outreach initiatives. For more info, please visit www.dsci.in

DATA SECURITY COUNCIL OF INDIA

+91-120-4990253 | ncoe@dsci.in

https://www.n-coe.in/

4 Floor, NASSCOM Campus, Plot No. 7-10, Sector 126, Noida, UP -201303

Follow us on

@CoeNational
nationalcoe
nationalcoe
NationalCoE