A Permissioned Blockchain System for Secure Multiparty Computation

A Permissioned Blockchain System for Secure Multiparty Computation

S Garg[1] and R Vashisht[2]
[1] Software Engineer, DLT Labs, Hyderabad, India
[2] Assistant Professor, Computer Science, and Engineering Department, Ajay Kumar Garg Engineering College, Ghaziabad, India

E-mail[1]: shobhitgarg9536@gmail.com
E-mail[2]: vashishtrohit@akgec.ac.in

ABSTRACT:
Permissioned blockchain is the blockchain network that requires access to be part of the network. Participant's actions are governed by the control layer that runs on top of the blockchain. This type of blockchains is preferred by individuals who need role description, identity, and security within the blockchain.
Secure multi-party computation (MPC) is a part of cryptography that involves the modeling of procedures for two or more participants who want to work together. These participants involve sharing of input and required computational data for a particular function without sharing their confidential data actively to each other and achieving a common goal which is beneficial to both as the outcome achieved is only revealed to the participants and is highly required for their functional purpose.
In this work, SPDZ (Speedz) implementation is explored leveraging additive secret sharing on the private blockchain (Hyperledger fabric). SPDZ protocol is chosen over any other computational protocol as it is highly secured from any active deceptive n-1 participant among the n participants. In this work, a backend is developed that uses a fabric SDK node.js library that interacts with the Hyperledger Fabric network. The proposed solution is shown through a demonstration. This paper concludes that for business-to-business scenarios, using SPDZ protocol on permissioned blockchain provides more security against adversaries as permissioned blockchain provides transparency over the participants of the network. As a result, permissioned blockchain is a more secure choice for enterprises to compute confidential data rather than permissionless blockchain.

Keyword: Private Blockchain, Additive secret sharing, Fabric Nodejs SDK, Secure Multiparty Computation, SPDZ, Hyperledger Fabric.

1. INTRODUCTION

Since the ’80s [1] [2] scholars are working on the secure multi-party computation which is a cryptographic technique that allows two or more parties to share their inputs which are secret and computed via a trusted party that is not handled by either of the participants. The cryptographic message which is the replacement of the virtual entrusted party has to guarantee the security of the data and come over with precise output.
MPC is primarily being conceptually [15,17,19,28,29,34] worked upon for decades, but recently its practical execution in the real world is endeavoring its way. Its practical implementation requires the execution of the theoretical protocol to improve notable performance and not abandoning the security levels when applying it in the practical world. In this trending blockchain, this research paper is one of its applications.
For the MPC protocol, this paper assumes that the identities of n parties are well-known and that each pair of channels has an authenticated and private communication channel. Normally, a Byzantine opponent has power over a certain number of parties, allowing them to deviate from protocol. They can exchange information with one another, avoid sending messages, send incorrect messages, and so on. These parties' primary aim is to deviate from the protocol by acquiring knowledge of private inputs or producing incorrect output from the function. It is presumed that the opponent is semi-honest if one party strictly follows protocol but is interested to learn secret knowledge about the other party. The adversary-controlled groups are dishonest or misleading,

while the others are honest or semi-honest. Static and adaptive adversaries are the two forms of an adversary. An adaptive adversary is not limited to choosing a collection of dishonest parties at the outset of the protocol; a static adversary is not.
This research paper operated on SPDZ (Speedz) [11], this protocol works on the execution of Multiparty protocol with additive secret shares and is secure from active static adversaries. This protocol is actively secure which detects the deception of n-1 among the n participants. The SPDZ protocol is linearly predicting in the online process for the n number of participants, and its expense is equivalent to the Shamir secret exchange protocol [14], which is categorized as passive security, making SPDZ more realistic with low procedure costs.
This research paper works on the implementation of Speedz a multiparty computation on Hyperledger Fabric [12] a private blockchain and to help drive the investigation, a working demonstration is also shown. Since most of the secure computation results were only stated as mathematical theorems and one had to read the relevant papers and implement them from scratch but this paper approach demonstrates how simple it is for different parties to coordinate and evaluate a function over their sensitive data without exposing it.

1.1 MEASURES OF EFFECTIVENESS

Typically, MPC protocol effectiveness is measured by the following metrics.
Resource Costs: This involves latency (total number of rounds of communication), computation cost, and communication cost (size of each message and the total number of messages sent).
Fault Tolerance: This criterion computes how much space a protocol allows has for an adversarial attack. They comprise how much bulk the protocol can tolerate, types of fault that are adversarial distributed, crash fault, Byzantine fault or random fault, how many nodes can adversary controls without offering correctness, and how many bits in messages can adversary corrupts.

1.2 BYZANTINE AGREEMENT (BA) AND MPC

MPC problem and Byzantine Agreement (BA) are closely coupled, in which there are a group of n participants and each participant has a different input value, and they all want to agree on one. Shostak, Pease, and Lamport[30] demonstrated that perfect-secure BA can be obtained if one-third of the parties are compromised. The following are some similarities between BA and MPC:
1. BA is much simpler than MPC as in BA privacy of inputs need not be maintained.
2. MPC protocols are closely coupled with broadcast channel usage. Unfortunately, where there are a larger number of parties, this broadcast channel provision is highly troublesome.

1.3 Paper Organization

The below is the format for the rest of the paper: Section 2 examines related work, with a focus on MPC on a blockchain. In Section 3, preliminaries are discussed which are publicly verifiable secret sharing, Speedz protocol overview, private blockchain over public blockchain, and Hyperledger fabrics (v2.0) basis. Section 4 contains the proposed SPDZ protocol on Hyperledger Fabric, backend architecture, and SPDZ protocol on Hyperledger Fabric demonstration. Finally, conclusion
in Section 5.

2. RELATED WORK:

Because of the large body of work, this paper does not seek a rigorous review of the MPC literature; instead, it focuses on scalable MPC on the blockchain.
Work on MPC in the recent past is primarily focused on protocol security inimical to passive adversaries used in Yao circuits [3] which is based upon two-party protocol and secret sharing techniques [4,5] which consists of multi-party contracts. Recently Active security [6, 7, 8] has gathered popularity and its real-time application gives more accurate outcomes but it increases the cost of operation when conducted between more than two-party transactions. An idea of covert security [9, 10] was introduced in 2007. In this security prototype, any corrupt party that tries to diverge from the contract is identified with higher accuracy of about 90% and this accuracy forces the adversary to act genuinely.
R Cleve's research [21] from 1986 demonstrated that it was difficult to achieve justice in the protocol when the majority of the malicious participants are involved. Rational multi-party computation (RMPC) [22] was introduced by Joseph Halpern and Vanessa Teague in 2004, which modifies the objective of conventional cryptography using the principle and approach of Game Theory. Under the conventional adversary model, the main point is to incorporate the idea of the logical adversary. The adversary is considered to be "self-interested"

when participating in the real-world protocol. It has a set of motivations and will always be driven by a set of interests. The adversary's tactics, actions, and abilities are all redefined by these functions. A mixed-behavior model was suggested by Anna Lysyanskaya et al. [23]. In this model, none of the parties are truthful. All of the participants are logical and self-centered, and they are only concerned with how to best serve their own interests. In their model, however, a trusted intermediary detects the presence of an attacker. Adam Groce et al. demonstrated that for arbitrary functions, equal justice is possible when both parties are logical, and cryptographic consequences can be avoided by using Game theory [24]. Mehrdad Nojoumian et al. created a new socio-rational secret sharing scheme by combining the rational and social secret sharing schemes [25]. In the reputation system, each user is assigned a reputation value, this system is kept in mind to build the public trusted network. The reputation value is updated in real-time based on the actions of the participants, and the reputation value is used to punish or reward the participants. The connection between stable two-party computation and Game Theory was investigated by Gilad Asharov et al. [26]. A number of two-party games was formed from the stable two-party computation protocol. In stable two-party computation, the analysis shows the connection between anonymity, Nash equilibrium, and correctness. MPC fairness is the primary concerned in the described above Game theory-based RMPC, but it is unable to address the issue of stability. Furthermore, alike MPC protocols are primarily designed to either they do not have a real context environment or have a centralized trustworthy third-party framework and to attain fairness.

With the advent of blockchain technology, embodied by Bitcoin [27], from the viewpoint of economics scholars started to incorporate economic rewards in Bitcoin into the MPC, owing to its features of trustability and decentralization, and due to its reward structure, it is entangled with Game Theory. It gives parties a sufficient and genuine reason to engage in MPC protocols. Based on the Bitcoin network, Marcin Andrychowicz et al. [16], [20] was the first to propose a secure two-party lottery protocol. The protocol's participants are compensated economically (such as Bitcoin) [32]. The two-party protocol was only specified in their work, not a multi-party protocol, and the initialization process requires several interactions with the Bitcoin network. Several ideal functionalities are described by Iddo Bentov and Ranjit Kumaresan [33]. e.g., F^{*}_{lot} (secure lottery with penalties functionality), F^{*}_{f} (secure computation with penalties functionality), F^{*}_{CR} (claim-or-refund functionality), They just need to call a constant round F^{*}_{CR} in the MPC protocol they build. Ranjit Kumaresan and Iddo Bentov investigated Bitcoin usage to inspire participants to complete accurate computations [36]. Secure computation with limited effusion, non-communal reward, fair secure computation, and confirmable computation are all aspects of their work. Moreover, Ranjit Kumaresan et al. spoke about how to make a decentralized poker game with Bitcoin [18]. Also, Ranjit Kumaresan et al. in 2017, proposed how to optimize the stable computation model using the penalty mechanism and strengthened the earlier suggested scheme [34]. En Zhang et al. in 2020, proposed a fair hierarchical threshold secret sharing mechanism that makes use of smart contracts, and the schema was implemented on Ethereum [39]. Juntao Gao et al. in 2021, proposed a fair and instant data trading schema based on bitcoin, it uses bitcoin script to requires the payee to provide the payer the data encryption key or he will not receive Bitcoin, and to guarantee fairness, the transaction's script will expose the account's private key if the same account was spent twice [37]. All of the previously mentioned work is constructed on the Bitcoin protocol, which focuses solely on justice in MPC and some modifications are done to the protocol to suit their needs [38]. The Bitcoin blockchain, as all know, has significant drawbacks in realistic implementation scenarios. Bitcoin's scripting language isn't Turing Complete, so it can't handle more finite and complex functions. Furthermore, owing to the transaction validation time of six blocks having a high transaction cost and lasting nearly an hour, the applications build on the Bitcoin protocol are still in the theoretical phase.

This research paper is primarily focused on enterprises solution. Since enterprises have confidential data and they cannot work on a public blockchain so in this paper it has been shown how enterprises can implement multiparty computation on permissioned blockchain. In section 3.3, the advantages of private blockchain over public blockchain and the motive for choosing private blockchain over the public blockchain are discussed in detail.

3. PRELIMINARIES

3.1 PUBLICLY VERIFIABLE SECRET SHARING

Schoenmakers [31] suggested the first publicly verifiable secret sharing (PVSS). The most noticeable characteristic of a non-communal PVSS is the lack of a secret channel between each participant and the secret provider. Over a public channel, each and every share are encrypted and sent with the commitment confirmation. and can be confirmed publicly.

Init: A public algorithm is used to generate the necessary system parameters. The earliest participants must produce a set of keys (Pk_i; Sk_i) using an asymmetric encryption algorithm, retain the private key Sk_i secret, and should reveal public key Pk_i publicly.
Distribute: The Dealer selects a secret s selected by the distributor is to be distributed amidst n parties {P_1, …, P_n}. For each member, the dealer will create and encrypt the secret share s_i (i =1, …, n) with the respective participant's public key Pk_i. Then on the public channel, the ciphertext would be transmitted with the commitment c_i = COMM(s_i). Via a non-interactive verification algorithm PVSS ($EnPk_i$ (s_i), c_i), anybody can verify if they had received the correct share or not.
Reconstruct: The party P_i decrypts the ciphertext with his private key and collects his share s_i. Each party is responsible for disclosing its own share. Anyone may use the algorithm verify(s_i, c_i) to ensure that s_i is accurate, preventing dishonest participants from yielding invalid shares.
A reliable multi-party computation protocol could be created by PVSS. Participants only need to work locally on their own shares in an addition gate because of its additive homomorphism function. When using a multiplication gate, the parties are forced to use PVSS to distribute a secondary dispersion of a worthless intermediary value.

3.2 Speedz OVERVIEW
In this section, Speedz (SPDZ) [11] protocol is briefly described. In this paper, it is presumed for the procedure that is to be performed between n participants of characteristic p in a fixed finite field F_p.
Each player P_i has a secret value a, uniform share $\alpha i \in F_p$ of secret value $\alpha = \alpha_1 + \cdots + \alpha_n$, notion of a constant MAC key $\gamma(a) := \alpha \cdot a$. Let participant gives an input $a \in F_p$ is $\langle\cdot\rangle$-shared if P_i holds a triplet $(a_i, \alpha_i, \gamma(a)_i)$, where $\gamma(a)_i$ is an additive secret sharing of $\gamma(a) := \alpha \cdot a$, i.e. $\gamma(a) = \gamma(a)_1 + \cdots + \gamma(a)_n$ and a_i is an additive secret sharing of a, i.e. $a = a_1 + \cdots + a_n$,.
This is the lucid definition of MAC for scholars who are familiar with SPDZ. In this protocol diverse values that are $\langle\cdot\rangle$ shared are "partially opened", that is linked values α_i are disclosed and not the linked shares of MAC. Each party can perform linear operations (scalar multiplication and addition) on the $\langle\cdot\rangle$ sharing's without involving interaction with other parties and they can compute it by themselves. However computational multiplication is not direct, but that is not discussed in this paper as this paper only showing implementation of SPDZ addition on the private blockchain.

3.3 BLOCKCHAIN
A permissionless blockchain is referred to as a public blockchain. The public blockchain network is open for all, which allows them to read, write, and interact with public blockchains. Public blockchains are decentralized, meaning nobody has power over the network, and the data is protected in the sense that it cannot be altered until it has been validated. Since a user is unknown, it needs to provide an incentive for good behavior in the public blockchain network. To guarantee that everyone in the system acts ethically and respects the rules, economics, and game theory incentives are provided. Scenarios are created by group consensus in which ethical parties are awarded incentives, while deceptive participants are forced to work or pay a cost that they will never be able to recoup.
A Private Blockchain, on the other hand, is a permissioned blockchain. It authorizes the participates of the network and the transactions they can do. It is authorized that who can write and read data on a private blockchain. In this process, the initial step is to establish one's identity. It is the need to know who is connected to the blockchain. If the user is anonymous, defining rules for the type of data the user can commit and read from the ledger becomes complicated, if not impossible. This is in stark contrast to a public network such as Ethereum, which strives to protect and maximize privacy.
Permissioned Blockchain advantage is that it is known who a user is, what organization they're affiliated with, and what their job is. It is presumed that they'll behave equally because if they don't, an administrator would know who's misbehaving and they'll know they'll pay the price.
As a result, the public and private blockchains have somewhat different features. Many people believe that they compete with one another, but this is not the case. They only exist to have various forms of solutions.
The following are some of the advantages of private blockchains:

- *Faster Transactions*
  The output is faster when the nodes are distributed locally but there are fewer nodes participating in the ledger.

- *Enterprise Permissioned*
  Blockchain resource access is controlled by the organization, making it private and/or permissioned.
- *Better Scalability*
  The ability to add resources and nodes by request can be a huge benefit.

3.4 HYPERLEDGER FABRIC BASICS

In Hyperledger fabric [12] the ledger is accessible by the nodes called peers that belong to an organization. In Hyperledger fabric the process of making transactions held in two phases. There has to be an entry of a client who requires any transaction to be executed. With a transaction proposal, this client has to approach endorsing peers. These peers have to validate the proposal by executing it on a smart contract which is referred to as chaincode in fabric. The endorsing peers in the fabric network have to determine whether the transaction has to be endorsed or not and return the endorsement response which has to be reflected in their state of the ledger. It has to be necessary that the endorses must look towards an identical transaction and if it is not then this proposal will be getting rejected automatically in the next phase of the transactions. Once the client receives a sufficient number of endorsements then the endorsed transaction is sent to the ordering service. This ordering service sends this transaction to all the participant peers which will update their ledger accordingly. Once the ledger gets initialized the endorsement policy can determine the required number of endorsements for the transaction to be processed. There has to be a single endorsement policy in the channel for all transactions for example there has to no less than one endorser peer from each organization or organization that can mutually construct their endorsement policy.

4. PROPOSED WORK

4.1 PROPOSED Speedz PROTOCOL ON HYPERLEDGER FABRIC

In this work Speedz (SPDZ) protocol on the private blockchain is proposed which uses encryption and decryption keys generated by the Hyperledger Fabric network. The proposed SPDZ protocol is described below.

1. The Hyperledger fabric functionality generates keys $(pk, sk)_i$, a set of public and private keys for each organization (which is used to authenticate admin and also to encrypt and decrypt private share).
2. **Pre-Processing Phase**: Each party P_j does the following
   a. Generate shared secrete key $[[\alpha]]$
   b. His secret data $[[x_j]]$ is defined as
      $<x_j> = (x^1_j, x^2_j, \ldots, x^n_j)$,
      $MAC_j = <x_j.\alpha_j> = (m^1_j, m^2_j, \ldots, m^n_j)$,
      $<\alpha_j> = (\alpha^1_j, \alpha^2_j, \ldots, \alpha^n_j)$, where n = number of participating organizations. Mac and secret share keys are randomly generated.
      Each party P_j will compute an additive sharing triplet $(x^i_j, m^i_j, \alpha^i_j)$ such that
      $[[x_j]] = \sum_{i=1}^{n} x_j^i$, $[[x_j.\alpha_j]] = \sum_{i=1}^{n} m_j^i$,
      $[[\alpha_j]] = \sum_{i=1}^{n} \alpha_j^i$
   c. Encrypt each ORG_j share, secret key, and MAC with its respective organization public key and commit P_j's additive sharing triplet ($Enc_{pk_j}(x^i_j)$, $Enc_{pk_j}(m^i_j)$, $Enc_{pk_j}(\alpha^i_j)$) to the ledger.
3. **Online Phase**: After all participating parties commit their secret shares to the ledger. Each organization will reveal its secret share key α^i_j.
   a. Each party P_i use his additive sharing triplet and decrypt its secret share x^i_j of ORG_j with its private key

      $x^i_j = Dec_{pk_i}(Enc_{pk_i}(x^i_j))$
      $\alpha^i_j = Dec_{pk_i}(Enc_{pk_i}(\alpha^i_j))$
      $m^i_j = Dec_{pk_i}(Enc_{pk_i}(m^i_j))$

   b. Compute
      $d^i_j = m^i_j - \alpha^i_j.x^i_j \ \forall\ j \in \{0, 1, \ldots, n\}$

c. Compute partial additive duplet which is defined as
$d_i = \sum_{j=1}^{n} d_j^i$
$\Delta_i = \sum_{j=1}^{n} x_j^i$
and each party P_i will commit its partial additive duplet share (d^i_j, Δ_i) to the ledger.

4. **Open phase**: When all parties commit their partial additive duplet share. Now, it's an open phase where all parties can see a partial additive duplet share of all parties
   a. Each party P_i will locally compute
   $d = d_1 + d_2 + \ldots + d_n$, which is defined as
   $d = \sum_{i=1}^{n} d_i$
   if $d \neq 0$, then abort otherwise continue
   b. Party P_i can locally compute additive secret which is defined as
   Additive secret = $\sum_{i=1}^{n} \Delta_i$

4.2 DETAILS OF HYPERLEDGER FABRIC IMPLEMENTATION

***DATA ENCRYPTION**:* In the data encryption module confidential data of organizations is divided into random pieces and keep it in the ledger in the encrypted form. The concern to be deal with is that how the encrypted data has to be placed in the ledger and how to make the transaction proposal of any client appear identical among all the peers of the organizations. The best way is that the client has to put his transaction proposal data in encrypted form in order to keep his data confidential from the peers of the other participant. This encrypted data has to be assisted by the encrypted keys which have to be given access to the client.

Every participant organization has its privileged client who has all organization's public key access. These organizations' public keys will be used under the proposed SPDZ protocol which is discussed in detail under section 4.1 that will generate the encrypted triplet and forward it in the transaction proposal to the endorsing peers.

SOFTWARE MODULES: Fabric's chaincode is composed in Node.js, the backend is also written in Node.js using Fabric SDK library [13], organizations can also have their own backend written in any language under the condition that it should have the ability to interact with the fabric network. This backend will interact with the fabric network and provide all the API's used by the organizations for registering its client and providing crypto material, creating and joining the channel, installing chaincode on their peers, instantiate and invoke chaincode on the channel and perform the transaction, there are other services as well which are used by the organization. These are the generation of the secret key, MAC, dividing confidential data, secret key, MAC between the total number of organizations that have joined the channel and then encrypt each data with respective organization's public key and send a transaction proposal for endorsement, computation of additive secret duplet, check whether to abort or compute additive secret.

THE ENDORSING PEERS: The architecture of the fabric is so designed that it makes the duty for the client to choose among the target peers for the transaction proposal when invoking the chaincode from the fabric SDK, but in this work, the service discovery feature of the Hyperledger Fabric is used because there may be a scenario where maybe some peers of some organization are not online for endorsement. So, Service Discovery provides us with the endorsement policy of the chaincode, and the availability of the peers, and orderers for the endorsement at that time.

4.3 SECURITY CONCERNS

The security-related features discussed below are addressed in this work implementation, and in any production system as well they must be taken care of.

ENDORSEMENT POLICIES: The requirement of creating the endorsement policy occurs in order to assure that the confidential data that belongs to the organization does not get modified without the endorsement of that organization and in this paper demonstration, an endorsement policy is set in which at least one peer from each organization will endorse the transaction. In order to make the organization's data secure the only best option is that every organization should endorse every transaction. The endorsement policy will be validating for every transaction.

***CLIENT CONSENT**:* There has to be a policy that authorizes the client under the fabrication backend. In this paper demonstration, each organization will designate some clients as privileged clients that have access to all organization public keys and can insert organization confidential data. Unprivileged clients have a hold on to

perform get queries on the ledger's state. For example, invoking partial results of their organization, verifying that protocol aborts in case of wrong data are committed to the ledger, query the final outcome of the additive secret sharing.

VALIDATING TRANSACTION. In the process of endorsement, it is hard to validate the encrypted data from the client. In order to face this issue, it is implemented in this paper demonstration which involves the non-interactional zero-knowledge proof [1] of actual endorsement.

4.4 ARCHITECTURE

The implementation comprises two phases, firstly it consists of a Fabric network and the second is the backend that interacts with the fabric network. This is elucidated in fig. 1. Demonstration in section 4.5 involves three organizations holding two peers each. Let their identification details can be ca_n.org_n.demo.com, peer_n.org_n.demo.com, couch_n.org_n.demo.com, _n ∈ {0, 1, 2}). A raft consensus (five orderers) and CouchDB over goleveldb for each peer for storing data is used by us and that gives an advantage of performing complex queries in the database.

The below architecture has a layer of the backend, and it assumes that each organization can have its separate backend. This backend will interact with the fabric network. This layer was developed via Hyperledger Fabric Node.js SDK [13] and utilizes the express.js framework. In order to rationalize coding, a demonstration in section 4.5 is executed on a single backend web server. This server works for all the three participant organizations (but in this paper, it is suggested that every participant organization can have its own backend in the production system). Each organization is free to write its own backend in any language. It just has the capabilities to interact with the fabric network.

In this paper, it is presumed that each organization separate backend so each organization can separately maintain their organization's peers' asymmetric keys, and other organizations don't have any access to those.

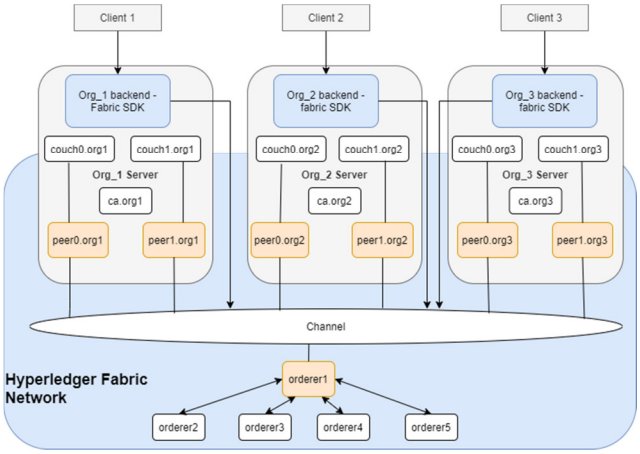

Fig. 1: High-level architectural diagram for the implementation of the SPDZ protocol on Hyperledger Fabric

4.5 PROPOSED SPDZ PROTOCOL ON HYPERLEDGER FABRIC DEMONSTRATION

The normal flow of this research work is as follows:

1) Let's say there are three organizations named A, B, C. And they want to know how many total cabs wherein service in rush hour.
2) Each organization runs its own Peer, CA on their organization's server. By consortium, all organizations join the fabric network. A channel will be created and all organizations join that channel which has an endorsement policy such that each transaction should be endorsed by at least one peer from each organization.
3) All organizations will install chaincode on their peers and instantiate chaincode on the channel so clients can start invoking chaincode. Each organization if want can write its own chaincode in any language and install it on its peers.

4) An admin of the organization performs a transaction of its public key on the ledger so that other organizations can use it for the encryption of data.
5) In preprocessing phase, each organization will interact with its middleware and generate a secret key and MAC and divide its confidential data, secret key, and MAC between the total number of participating parties and encrypt each data with its respective organization's public key. Send additive secret triplet in transaction proposal to peers for endorsement. If the endorsement fulfills that is at least one peer from each organization will approve the transaction then this will be committed to the ledger. Since every secret is in the encrypted form and the other party does not have verifiability of the data so zero-knowledge proof [1] is used.
   A:
   $x_1 = 45$, $\alpha_1 = 26$, Mac = $x_1.\alpha = 1170$
   $x^1_1 = -90$, $\alpha^1_1 = 10$, $m^1_1 = 170$
   $x^2_1 = 56$, $\alpha^2_1 = -6$, $m^2_1 = 400$
   $x^3_1 = 79$, $\alpha^3_1 = 22$, $m^3_1 = 600$
   B:
   $x_2 = 35$, $\alpha_2 = 64$, Mac = $x_2.\alpha = 2240$
   $x^1_2 = -270$, $\alpha^1_2 = 13$, $m^1_2 = 1200$
   $x^2_2 = 0$, $\alpha^2_2 = 41$, $m^2_2 = 540$
   $x^3_2 = 305$, $\alpha^3_2 = 10$, $m^3_2 = 500$
   C:
   $x_3 = 70$, $\alpha_3 = 148$, Mac = $x_3.\alpha = 10360$
   $x^1_3 = -220$, $\alpha^1_3 = 90$, $m^1_3 = 450$
   $x^2_3 = -144$, $\alpha^2_3 = 50$, $m^2_3 = 9000$
   $x^3_3 = 434$, $\alpha^3_3 = 8$, $m^3_3 = 910$

6) In the online phase, all organization will reveal their secret key (α) but not the MAC. Each organization P_i will compute d^i_j ∀ j € {0, 1, …, n}. And compute additive secret duplet which is defined as $d_i = \sum_{j=1}^{n} d^i_j$
   $\Delta_i = \sum_{j=1}^{n} x^i_j$, where n = total number of participating parties. Each organization then sends a transaction proposal of additive secret duplet for endorsement.
   A:
   $d^1_1 = m^1_1 - \alpha_1.x^1_1 = 170 - 26(-90) = 2510$
   $d^1_2 = m^1_2 - \alpha_2.x^1_2 = 1200 - 64(-270) = 18480$
   $d^1_3 = m^1_3 - \alpha_3.x^1_3 = 450 - 148(-220) = 33010$
   $d_1 = d^1_1 + d^1_2 + d^1_3 = 54000$
   $\Delta_1 = x^1_1 + x^1_2 + x^1_3 = -90-270-220 = -580$
   B:
   $d^2_1 = m^2_1 - \alpha_1.x^2_1 = 400 - 26(56) = -1056$
   $d^2_2 = m^2_2 - \alpha_2.x^2_2 = 540 - 64(0) = 540$
   $d^2_3 = m^2_3 - \alpha_3.x^2_3 = 9000 - 148(-144) = 30312$
   $d_2 = d^2_1 + d^2_2 + d^2_3 = 29796$
   $\Delta_2 = x^2_1 + x^2_2 + x^2_3 = 56+0-144 = -88$
   C:
   $d^3_1 = m^3_1 - \alpha_1.x^3_1 = 600 - 26(79) = -1454$
   $d^3_2 = m^3_2 - \alpha_2.x^3_2 = 500 - 64(305) = -19020$
   $d^3_3 = m^3_3 - \alpha_3.x^3_3 = 910 - 148(434) = -63322$
   $d_3 = d^3_1 + d^3_2 + d^3_3 = -83796$
   $\Delta_3 = x^3_1 + x^3_2 + x^3_3 = 79+305+434 = 818$
7) In the open phase, each organization can individually compute
   $d = \sum_{i=1}^{n} d_i \quad \forall i \in \{0, 1, \ldots, n\}$.
   If $d \neq 0$ then one or many participating parties are corrupted and wrong data is committed so protocol abort and additive secret will not be calculated. And the probability of happening this is 1/p.
   $d = d_1 + d_2 + d_3 = 54000 + 29796 - 83796 = 0$
8) Otherwise, parties compute additive secret by adding all partial additive secret which is defined as

Additive secret = $\sum_{i=1}^{n} \Delta_i$
Additive secret = $\Delta_1 + \Delta_2 + \Delta_3$
= -580- 88 + 818 = 150
$x = x_1 + x_2 + x_3 = 45 + 35 + 70 = 150$

5. CONCLUSION

At present, data trading faces several problems including fairness, latency, and legality Although many solutions are proposed but are limited to the permissionless blockchain. In this paper, it had been shown that how enterprises can use permissioned blockchain and perform data trading more securely and fairly. In SPDZ schema there are three phases, first phase is the pre-processing phase in which each participant will divide its confidential data, secret key, and MAC between the total number of participating parties and encrypt each confidential data with its respective organization's public key. Send additive secret triplet in transaction proposal to Hyperledger fabric peers for endorsement and only when endorsement will be fulfilled then only it will be committed to the ledger. The second phase is the online phase in which each participant will share its secret key with every organization of the network and each participant will compute additive secret duplet and partial additive secret and send transaction proposal for endorsement. If an endorsement is fulfilled protocol will move to the last phase otherwise protocol will abort. The last phase is the open phase in which all additive secret duplet shared by all the participants are added and if that is not equal to zero than one or more than one participant has corrupted in the protocol so the protocol will be aborted otherwise additive secret will be revealed by the protocol to all the participants in the permissioned network.
In this work, architecture is designed that supports SPDZ additive secret sharing on Hyperledger fabrics, a permissioned blockchain, and designed a demonstration structure using the fabric. This research work identifies that Hyperledger Fabrics has the capability with which one can easily implement SPDZ as an invulnerable multi-party computation protocol. This work can be extended by enhancing this paper's solution by supporting multiplication using the SPDZ protocol on secret data.

References

1. Goldreich O, Micali S, and Wigderson A 1991 Proofs that yield nothing but their validity or all languages in NP have zero-knowledge proof systems *Journal of the ACM* **38(3)** pp 691–729
2. Yao A C Nov. 1982 Protocols for secure computations (extended abstract) *23rd FOCS, IEEE Computer Society Press* pp 160–164
3. Malkhi D, Nisan N, Pinkas B, Sella Y 2004 Fairplay - Secure two-party computation system *USENIX Security Symposium* pp 287–302
4. Bogdanov D, Laur S, Willemson J 2008 Sharemind: a framework for fast privacy-preserving computations *Jajodia, S., Lopez, J. (eds.) ESORICS 2008* LNCS **vol 5283** Springer, Heidelberg pp 192–206
5. Damgård I, Geisler M, Krøigaard M, Nielsen J B 2009 Asynchronous multiparty computation: Theory and implementation *Jarecki, S., Tsudik, G. (eds.) PKC 2009* LNCS **vol. 5443** Springer, Heidelberg pp 160–179
6. Kreuter B, Shelat A, Shen C-H 2012 Towards billion-gate secure computation with malicious adversaries *USENIX Security Symposium 2012* pp 285–300
7. Lindell Y, Pinkas B, Smart, N P 2008 Implementing two-party computation efficiently with security against malicious adversaries *Ostrovsky, R., De Prisco, R., Visconti, I. (eds.) SCN 2008* LNCS **vol 5229** Springer, Heidelberg pp 2–20
8. Pinkas B, Schneider T, Smart N P, Williams S C 2009 Secure two-party computation is practical *Matsui, M. (ed.) ASIACRYPT 2009* LNCS **vol 5912** Springer, Heidelberg pp 250–267
9. Aumann Y, Lindell Y 2007 Security against covert adversaries: Efficient protocols for realistic adversaries *Vadhan, S.P. (ed.) TCC 2007* LNCS **vol 4392** Springer, Heidelberg pp 137–156
10. Aumann Y, Lindell Y 2010 Security against covert adversaries: Efficient protocols for realistic adversaries *Journal of Cryptology* **23(2)** pp 281–343
11. Damgård I, Pastro V, Smart N P, Zakarias S 2012 Multiparty computation from somewhat homomorphic encryption *Safavi-Naini, R., Canetti, R. (eds.) CRYPTO 2012* LNCS **vol 7417** Springer, Heidelberg pp 643–662

12. Welcome to Hyperledger Fabric. Available: https://hyperledger-fabric.readthedocs.io/, accessed. (Jan 2018) [Online].
13. Fabric Node SDK Github. Retrieved (2020) [Online]. Available: https://github.com/hyperledger/fabric-sdk-node.
14. Shamir Adi 1917 How to share a secret *Communications of the ACM* **22 (11)** pp 612–613
15. Asharov G, Jain A, Lopez-Alt A, Tromer E, Vaikuntanathan V and Wichs D 2012 Multiparty computation with low communication, computation, and interaction via threshold FHE *Pointcheval, D., Johansson, T. (eds.) EUROCRYPT 2012* LNCS, **vol 7237** Springer, Heidelberg pp 483‑501
16. Andrychowicz M, Dziembowski S, Malinowski D, Mazurek L 2014 Fair two-party computations via Bitcoin deposits *Financial Cryptography and Data Security* Berlin, Germany: Springer, pp. 105-121
17. Boyle E, Chung K-M, Pass R 2014 Large-scale secure computation *Cryptology ePrint Archive*, Report 2014/404
18. Kumaresan R, Moran T, Bentov I 2015 How to use Bitcoin to play decentralized poker *Proc. 22nd ACM SIGSAC Conf. Comput. Commun. Secur. (CCS)* pp 195-206
19. Damgård I, Ishai Y, Krøigaard M 2010 Perfectly secure multiparty computation and the computational overhead of cryptography *Gilbert H (ed.) EUROCRYPT 2010* LNCS, **vol 6110** Springer, Heidelberg pp 445‑465
20. Andrychowicz M, Dziembowski S, Malinowski D, Mazurek L May 2014 Secure multiparty computations on Bitcoin *Proc. IEEE Symp. Secur. Privacy* pp 443-458
21. Cleve R 1986 Limits on the security of coin flips when half the processors are faulty *Proc. 18th Annu. ACM Symp. Theory Comput. (STOC)* pp 364-369
22. Halpern J, Teague V 2004 Rational secret sharing and multiparty computation: Extended abstract *Proc. 36th Annu. ACM Symp. Theory Comput. (STOC)* pp 623-632
23. Lysyanskaya A, Triandopoulos N 2006 Rationality and adversarial behavior in multi-party computation *Advances in Cryptology-CRYPTO* Berlin, Germany: Springer, pp 180-197
24. Groce A, Katz J 2012 Fair computation with rational players *Advances in Cryptology-EUROCRYPT* Berlin, Germany: Springer, pp 81-98
25. Nojoumian M, Stinson R D 2012 Socio-rational secret sharing as a new direction in rational cryptography *Decision and Game Theory for Security* Berlin, Germany: Springer, pp 18-37
26. Asharov G, Canetti R, Hazay C 2011 Towards a Game Theoretic View of Secure Computation *Advances in Cryptology-EUROCRYPT* Berlin, Germany: Springer, pp 426-445
27. Nakamoto S 2008 Bitcoin: A Peer-to-Peer Electronic Cash System [Online]. Available: https://bitcoin.org/bitcoin.pdf
28. Boyle E, Goldwasser S, Tessaro S 2013 Communication locality in secure multiparty computation *Sahai, A. (ed.) TCC 2013* LNCS, **vol 7785**, Springer, Heidelberg pp 356‑376
29. Dani V, King V, Movahedi M, Saia J 2014 Quorums quicken queries: Efficient asynchronous secure multiparty computation *Chatterjee, M., Cao, J.-n., Kothapalli, K., Rajsbaum, S. (eds.) ICDCN 2014*. LNCS, **vol 8314**, Springer, Heidelberg pp 242‑256
30. Pease M, Shostak R, Lamport L 1980 Reaching agreements in the presence of faults *Journal of the ACM* **27(2)** pp 228‑234
31. Schoenmakers B 1999 A simple publicly verifiable secret sharing scheme and its application to electronic voting *Advances in Cryptology-CRYPTO* **vol 99**. Berlin, Germany: Springer, pp 148-164
32. Hemant Gupta and Mayank Singh 2019 Cyber threat analysis of consumer device, *Communications in Computer and Information Science*, vol. 1046, pp. 32-45.
33. Bentov I, Kumaresan R 2014 How to Use Bitcoin to design fair protocols *Advances in Cryptology-CRYPTO* Berlin, Germany: Springer, pp 421-439
34. Kumaresan R, Vaikuntanathan V, Vasudevan N P 2016 Improvements to secure computation with penalties *Proc. ACM SIGSAC Conf. Comput. Commun. Secur. (CCS)*, pp 406-417
35. Damgård I, Ishai Y, Krøigaard M., Nielsen J B, Smith A 2008 Scalable multiparty computation with nearly optimal work and resilience *Wagner, D. (ed.) CRYPTO 2008*. LNCS, **vol 5157** Springer, Heidelberg, pp 241‑261.
36. Kumaresan R, Bentov I 2014 How to Use Bitcoin to incentivize correct computations *Proc. ACM SIGSAC Conf. Comput. Commun. Secur. (CCS)* pp 30-41
37. Gao J, Wu T and Li X 2021 Secure, fair and instant data trading scheme based on bitcoin *Journal of Information Security and Applications* **vol 53**

38. Manjula Shanbhog, Krista Chaudhary, Mayank Singh and Shailendra Mishra 2017 Development of Secured Trust SLA Model from SLA Life Cycle Phases, *Communications in Computer and Information Science*, vol. 721, pp. 102-111.
39. Zhang E, Li M, Yiu S-M, Du J, Zhu J-Z and Jin G-G 2020 Fair hierarchical secret sharing scheme based on smart contract *Information Sciences* **vol 546**, pp 166-176