A Risk Assessment And Analysis Framework For Identification And Analysis Of Possible Risks In Hyperledger Fabric

Deepti Singh[1], Mohit Dayal[2], Jitender Kumar[3]*, Sonika Bhatnagar[4], Priya Paliwal[5], Bhumika Pal[6]

Abstract:

The Hyperledger Fabric or smart contracts in blockchain are generally designed for the purpose of common coding languages that are notable by probable developers and coders, like Golang. Because of the loopholes in developing improvement particulars for smart Contracts utilizing all-purpose coding language, there exists frequently occurred risks related to hyperledger fabric. It will promote numerous problems and possible security risks to the clients after the smart cards are being circulated into hyperledger fabric. In spite of the fact that there already exists some possible risk identification methodologies for Hyperledger Fabric, the precision and inclusion of the methodologies are restricted. In reaction to such issues, this paper sums up three sorts of expected risks in the Hyperledger's smart contract: Logical Security, Non-deterministic, and Security of Personal Data. To identify such kinds of possible risks, a static strategy of analysis which is centered on FD (Functional Dependency) and AST (Abstract Syntax Tree) has been proposed in the paper. Simultaneously, an identification framework that can precisely find the area of possible risks in the Hyperledger Fabric's smart contract is designed to produce advancement ideas for the recommendation of the developers of smart contracts.

Index words: - Blockchain, Distributed Ledger, Hyperledger Fabric, Security, Smart Contracts.

DOI Number: 10.4704/nq.2022.20.14.NQ88009 **Neuro Quantology 2022; 20(14):58-68**

1. INTRODUCTION

1.1. About Blockchain :

1.1.1. Background and Motivation

"Blockchain is considered as a distributed and constantly growing ledger or file which keeps record of every single transaction, in a chronological, immutable or fixed, ordered, and secure way (Azaria, 2016)."

Few of the utilization of Blockchains includes secure transfer of funds, right information of a property to be bought or sold, employing of lawful agreements, and so forth without any intervention of a third-party or centralized authority such as banks or government organizations (Bettin, 2018). Blockchain is an enduring track of "who has what", which is also considered as the world's biggest file system or spreadsheet which is constantly expanding and called "Genesis Blockchain". A Blockchain is a Peer-to-Peer distributed ledger which permits the massive data to be recorded on a huge count of servers (Chohan UW, 2017). Because of its distributed nature, every single node within the network will get notification of even a single transaction occurring in the blockchain. This promotes zero chance of centralized administration (Fanning K, 2016) as everyone in the blockchain network possesses the Blockchain regulations at the same preference.

***Corresponding Author**: - Jitender Kumar
Address: - [1] ABES Institute of Technology, Gaziabad
[2] Dr. Akhilesh Das Gupta Institute of Technology & Management, New Delhi
*[3] J.C.Bose University of Science & Technology, YMCA, Faridabad, Haryana
[4] Government Polytechnic Baheri, Bareilly
[5] Dr. Akhilesh Das Gupta Institute of Technology & Management, New Delhi
[6] Dr. Akhilesh Das Gupta Institute of Technology & Management, New Delhi

Relevant conflicts of interest/financial disclosures: The authors declare that the research was conducted in the absence of any commercial or financial relationships that could be construed as a potential conflict of interest
Received: **Accepted:**

The shortcomings of centralised frameworks (say banks) such as, the involvement of third parties all the time, the enduring control of centralised systems on our data and assets, leads to the evolution of blockchain and cryptocurrencies (Gildfeer S, 2018).

To dispose of this inadequacy of the current financial framework, Satoshi Nakamoto suggested digital money over physical money which was formed utilizing Blockchain. In 2007, he tossed a decentralized financial system and furthermore presented a cryptocurrency "Bitcoin" through a research paper (Hofmann E, 2017) (J. Peng,). A blockchain is just a set of records kept together in blocks which are interconnected together. The blocks are added in chronological manner and once added cannot be changed or altered and this makes it immutable and thus, highly secure (Li H, 2019). Being distributed in nature the linkage between two blocks are followed by the concept of linked list as the current block is connected with the previous block by holding the hash of previous one which is displayed in Figure 1 (Li H, 2019).

Figure: 1: General linkage in Blockchain

1.1.2. Highlights of Blockchain

- Decentralized: Every single node in the Blockchain network is being treated as a peer without the existence of any central administration or ownership. All the peers are equally notified and requested for any transaction to initiate with the concept of common consensus (Androulaki E, 2018).
- Immutable: Data or transaction once composed into the Blocks or added into the network, can not be modified.
- Security: Hashing technique which is an enduring measure, is used to promote security in Blockchain. As shown in Fig.2, the hash function, inputs variable-sized data and converts it into an irreversible, fixed-length, encrypted and unique output data (Foschini L, 2020). Because of the above said features it is practically not possible for the extruders and hackers to figure out the size and value of Hash. The SHA256 hashing technique is the commonly used algorithm in Blockchain.

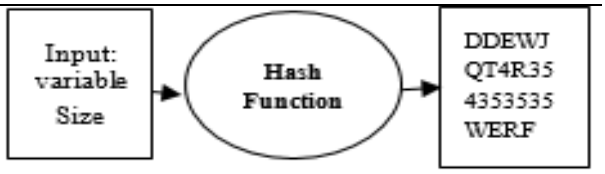

Figure: 2: Process of creating Hash Values through Hashing

- Distributed Ledger: Blockchain is a digital and public ledger and keeps track of transactions and peers which is all completely transparent to all the nodes, nothing is out of sight.
- Consensus: It is the agreement on which all the nodes give their accord for any transaction to take place. Below mentioned are the three major consensus techniques (Harz D, 2018) (S. Tang, 2022):

a) Proof of Work: To add a transaction or record, the node has to give prove of correctness as it has to solve and submit the correct solution of a complex mathematical puzzle which is to find out the unique combination of golden nonce at the earliest.

b) Proof of Stake: Nodes have to contribute by holding the coins for a longer period of time as a stake.

c) Proof of Capacity: The peers contribute by their maximum available free storage space i.e. hard drive.

1.1.2. Architecture of Blockchain

Fig. 3. Represents the common view of Blockchain architecture [12] . The elements of blockchain (Genesis) are given below:

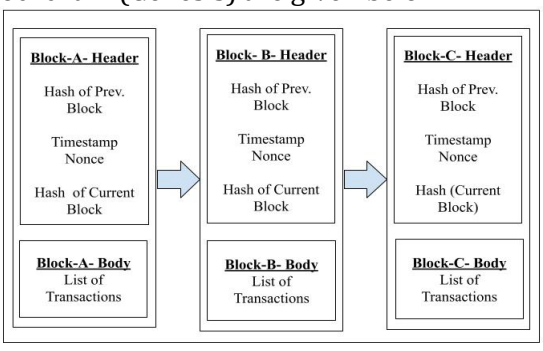

Figure: 3: Basic Architecture and components of Genesis Blockchain

- Block - Block is any structure of data which holds the record of the Blockchain transactions.
- Transaction - Any upgradation in records or information within blockchain.

- Chain - It is simply a linkage of blocks or nodes in consecutive manner.
- Node - A node may be a user or computer within the network of blockchain and every node keeps a replica or record of the complete ledger maintained in blockchain)
- Consensus - it is a protocol to agree on an agreement for the operations being performed in the.
- Nonce - It is a 32-bit value, randomly produced when the block is created, and then it causes a hash.
- Miners - A Person or node that is responsible for verification of blocks in the process of adding it to blockchain.

1.1.4. How blockchain works?

Figure 5 represents the overall working model of a Blockchain (Huang Y, 2019). It depicts the overall process which will be followed if a node wants to initiate a transaction within a blockchain.

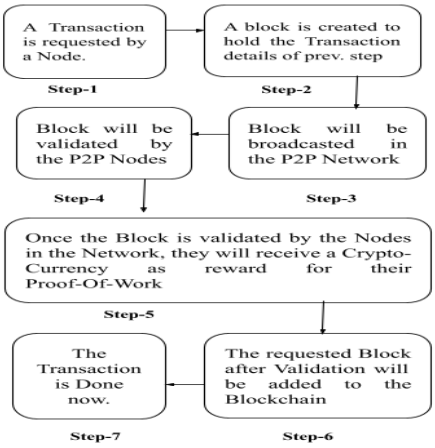

Figure 5: Workflow in Blockchain

1.1.5. Types of Blockchain

Figure 4 represents the classification of Blockchain through a block diagram (Yama-shita, 2019).

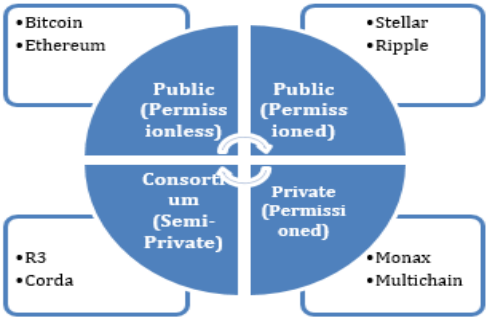

Figure 4: Blockchain Categories

There are two major categories of a Blockchain, on the basis of features, security, and access limitations, Closed and Open Blockchain. There are further two categories of each which are Permissioned and Permissionless. The overall categories (Harish, 2017) (A. Shahidinejad, 2022) and their respective comparison is mentioned in Table 1.

OPEN BLOCKCHAIN		CLOSE BLOCKCHAIN	
Public (Permissionless)	Public (Permissioned)	Consortium/ Federated	Private Blockchain
Open access to anyone. Everyone can perform both read and write	Read operation is open for everyone but Write operation is allowed to authorized ones only.	Both Reading and Writing are allowed only to authorized participants	Reading is for authorized participants and only the network operator is restricted to write and commit
Permissionless	Permissioned	Permissioned	Permissioned
Anonymous Participants	Authorized Participants for Write and Commit	Known Participants	Known Participants
Security: Consensus Algorithms	Security: Consensus Algorithms	Security: Priorly Approved Participants	Security: Priorly Approved Participants
Consensus : PoW / PoS	Consensus : PoW / PoS	Consensus: Voting/Multi-Party	Consensus: Voting/Multi-Party
Slow Transaction	Faster Transaction	Faster Transaction	FasterTransaction
Decentralized Network	Decentralized Network	Partially Decentralized Network	Decentralized (Hybrid of Both) Network
Large Energy Consumption	Large Energy Consumption	Slow Energy Consumption	Slow Energy Consumption

Table 1: Comparison of Open and Closed Blockchains

1.1.6. Security concerns in Blockchain

Figure 6 represents the summary of different types security and privacy threats to Blockchain (Ma C, 2019). At distinct categories, there exists different concerns for the of Blockchain scenario. The intruders target the loopholes at various levels to breakdown the security. Blockchain has distinct connection with the public or enterprise and individual unit, the intruders bypass the system to snatch, takeover and fudge the infrastructure between Blockchain and the clients.

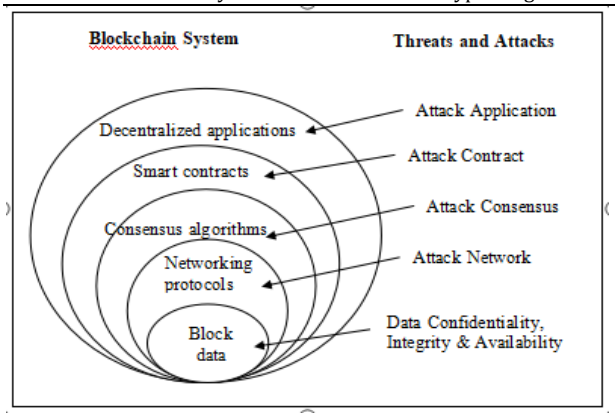

Figure 6: Common external threats to Blockchain

1.2. What is Hyperledger Fabric

Hyperledger Fabric (HLF) is a framework under the Hyperledger Modular Umbrella Approach which is a Permissioned (Private) blockchain and has an exceptionally configurable and modular architecture (Behamouda F, 2019) (S. Jain, 2022). HLF was proposed in 2015 by The Linux Foundation as an establishment for designing applications with a secluded framework, it permits segments, like membership and agreement services, to be ready for use. It is an Open Source and Governed by a different group of people from numerous organizations. Empowers security measures to promote trust and transparency in the network against each transaction. Below are few features of HLF which makes it convenient to use in Blockchain to get efficient transaction processing and security:

- Permissioned and Private
- Highly configurable and exponentially modular architecture
- Ready to use consensus protocols
- Low latency in confirmation of the transaction
- High throughput performance
- Best suitable for enterprise
- Open-source
- Provides access control and unique identity

1.2.1. Architecture of Hyperledger Fabric

HLF is a private Blockchain and is used in enterprises. Figure 7 represents the architecture and components of the HLF network (Liu X, 2018). The components are listed below as:

- Client Application- A person who will make transactions through it.
- SDK (HFC)- Software Development Kit (Hyperledger Fabric Client) can be in any language to add transaction into the Blockchain (Li Z, 2018).
- Membership Services - It provides certificates with the help of External/Internal Certificate Authority (CA) to the user that now you can send your transaction for further acceptance/rejection.
- Ordering Services- Set transactions into an order to form a final block to send to the committer.

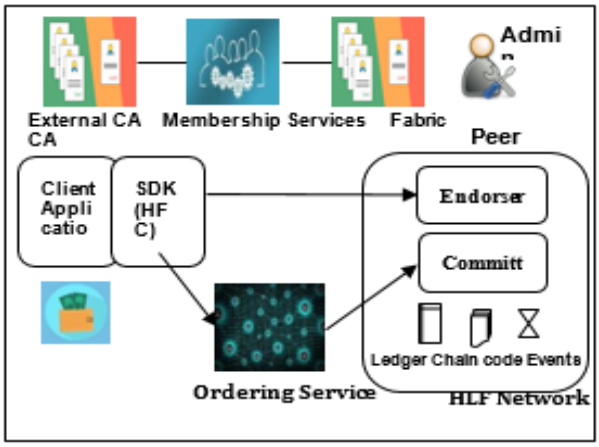

Figure 7: Architecture and Workflow of HLF

- Peer- Any node in the Blockchain. It includes the following members:
  - Committer - It will add the final block into BlockChain. Responsible for maintaining the ledger and may keep (chaincode) smart contract.
  - Endoresor - It receives and either Signs or simply rejects the client's transactions. Also, it essentially keeps smart contract.
  - ChainCode (CC) - Smart Contracts with all policies.
  - Events - Notifications regarding the accept/reject activity.

1.2.2. Working Process of HLF

In HLF network, the client-side application initiates the transactions by sending the peer a request on a medium (Zheng, 2016). This flow of transaction accepts that the application client has enrolled and registered with the system's CA (Certificate authority). Figure 7 portrays the flow of transaction which includes 7 stages (Vukolic M, 2015). The flow of transaction is mentioned below:

a) Propose Transaction: HLF Client (HFC), which can be written in any programming language is used by the Client to propose a transaction to the endorser to check for further acceptance (by signing the proposal) or to rejection.
b) Execute transaction: Endorser with the help of chaincode (CC) checks the transaction whether to accept it by signing it or to reject .
c) Proposal Response: Here, the endorser sends the response of the proposal to the client whether it has signed or rejected.
d) Order transaction: If the endorser's response is positive, then the client sends the signed transaction to the ordering service to arrange and add it according to the list of transactions to form a block.
e) Deliver Transaction: At this stage, the ordering service sends the block to the committer.
f) Validate Transaction: As the client sends the transaction proposal to multiple endorsers and all endorsers have access of chaincodes (CC), here the committer validates the transaction as per the terms and conditions mentioned in the CC against the received block.
g) Notification: The client will be notified if the transaction gets validated by the committer through the events.

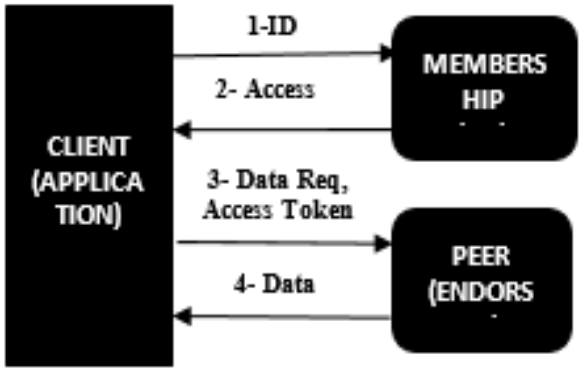

Figure 8: The process of Authentication in HLF

Figure 8 demonstrates the process of authentication in HLF transaction flow process (Androulic E, 2018). It represents the access of data to the client through the certificate authorities and endorsers.

1.2.3. Risk Factors in Hyperledger Fabric:

● Non-Deterministic Risk (NDR)

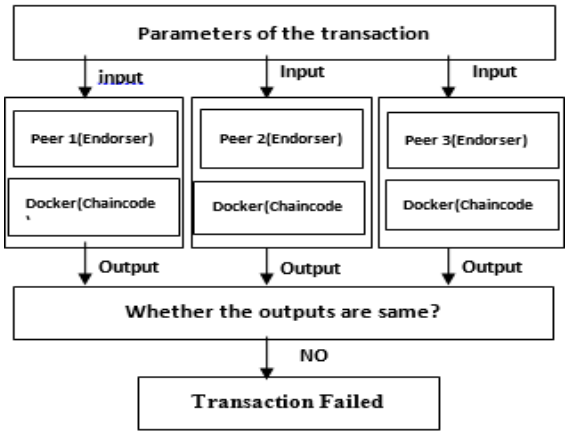

Figure 9: Failure of a transaction brought about by NDR

In HLF, the failure of a transaction brought about by the NDR is depicted in the endorsement phase, as displayed in Figure 9 (Buterin, 2016). The independent and distributed endorsers take the same specification regarding transaction, and afterward perform the CC execution with the NDR to reenact transactions. However, the results of transactions from various peers are diverse, this contravenes the agreement rules of HLF and results in failure of transaction. Thus, it is important to identify the NDR in the CC (Kumar P, 2021). The NDR mostly comes from Non-deterministic Process of Execution, or Non-deterministic Sources of Data or system timestamp. Non-deterministic Sources of Data which includes variables or objects with different values at different peers like random generation of numbers or diverse addresses of objects, and Non-deterministic Calls (Son L, 2017) (W. Liang,2022). Non-deterministic Process of Execution alludes to the action in which the sequence of execution is different for the internal logic of identical functions of the CC at distinct peers, or the same variables carry their different values, which prompts unpredictable results of transaction (Khari M, 2020). External Calls (Non-deterministic) allude to intervention from outerward the blockchain which promote the unpredictable results of transactions like External File Accessing, and Execution of external System Command.

- Data Privacy Risk: Risk of Data Privacy alludes to the possibility of failure of initiated transaction because of authentication issues, or leakage of delicate information because of

the absence of safety measures while reproducing the CC (J. Maeng, 2022). It incorporates Invocation of Cross Channel CC and insecure delicate Data.

- Logical Security: Mostly talks about the risk of emerging from the database activities of HLF, such as Read/Write or Query Risk. HLF gives a few range question strategies to get to the databases, for example, Get Private Data Query Result (). These strategies are implemented while the stage of endorsement, however are not executed again throughout the confirmation stage. Hence, these techniques can't be utilized to alter the CC ledger and must be utilized to question the transaction within the ledger. In HLF, writing transaction-related data into a blockchain ledger is implemented once the transaction is finished and confirmed. Accordingly, read-write operation can't be performed on the data of the same operation in the CC, i.e., Read the Write aren't upheld in HLF (Singh N, 2016).

2. ARCHITECTURE & IMPLEMENTATION OF PROPOSED MODEL

2.1 ARCHITECTURE

To discover possible risks and deformities in the CC of HLF, a static analysis based identification strategy has been proposed for CC, created by Golang, as displayed in Figure 4. The framework mainly incorporates three sections:

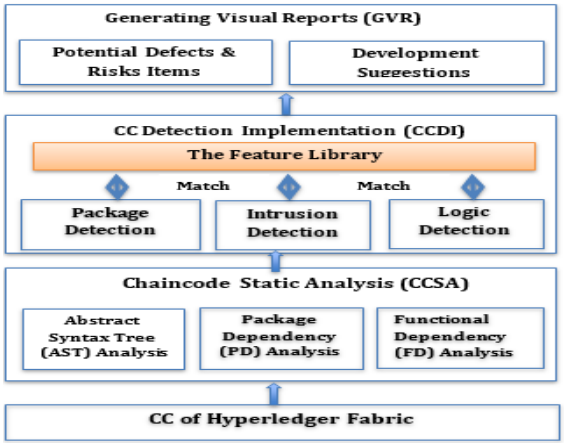

Figure10: Architecture of proposed model

1. In CCSA section, the chaincodes of HLF are broken down and reviewed to get information about the static structure, for example, Abstract Syntax Tree (AST), Functional Dependency (FD) etc.
2. In CCDI section, it is intended to find out the variety and areas of expected risks and deformities by coordinating with an attribute library made out of the collection of static attributes for the risks in CC.

In GVR, the report covers location and description of the possible risk units in the CC, and improvement ideas to wipe out these units.

CCSA and CCDI sections are the centre and most important components of the proposed system for the identification of risk. The next point mainly focuses on presenting the plan and execution of these modules.

2.2 STATIC ANALYSIS OF CC

This is to carry out the Package Dependency (PD) Analysis, Abstract Syntax Tree (AST) Analysis, and Function Dependency (FD) Analysis on the CC to get the information regarding their static framework: Package Dependency (PD) Relationship, Abstract Syntax Tree (AST), Function Call (FC) Relationship, which is displayed in Figure 11.

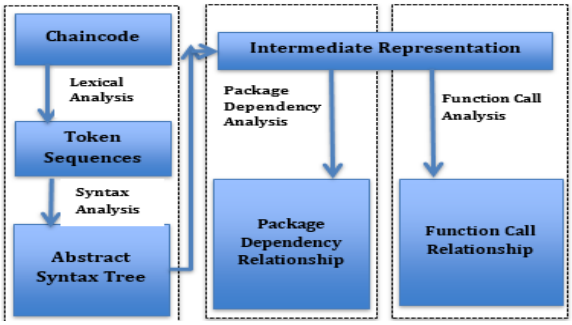

Figure 11: Static Analysis of Chaincode (CC)

2.3 Abstract Syntax Tree (AST) Analysis

It is the way toward acquiring the AST of the CC by utilizing the two passes of the compiler : Syntax Analysis and Lexical Analysis (P. Gaba, 2022). By Lexical Analyser the CC are utilized as data (input) to create identifiable token series. Then, at that point the output of Lexical Analysis i.e. tokens are utilized as input by the Syntax Analyser, to build the Abstract Syntax Tree of the CC. It uses the bottom-up approach in such a way that it steadily merges the sub-trees upwards to get the final AST. As displayed in Figure 12, the Abstract Syntax Tree of the CC takes the complete (ast.File) file as the first node which is root, and rest of the nodes

portray the syntactic structure of various levels in the record (file) considering top-down approach, and every node consists of detailed declaration as well as definition of its structure, which addresses its exact location in the CC and also its correlation with the other nodes (files).

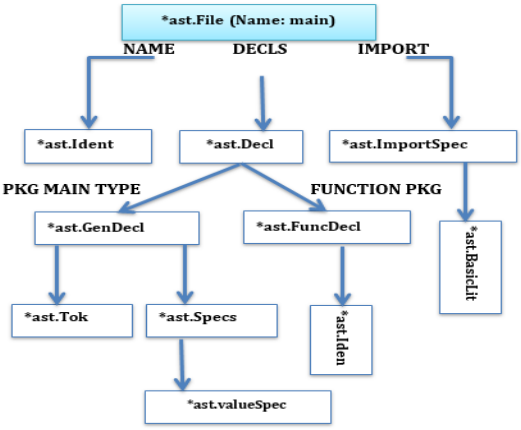

Figure 12: Abstract Syntax Tree (AST) for CC

2.3 Package Dependency (PD) Analysis

PD Analysis is a task of evaluating the Abstract Syntax Tree of the CC to get the correlations among the packages labeled by the CC. Algorithm 1 represents the entire steps of Package Dependency (PD) Analysis. Semantic (grammatical) analysis is executed first to generate the SSA (Static Single Assignment) based Intermediate Representation from the Abstract Syntax Tree of the CC (received from Syntax analysis phase).

Algorithm 1: Package Dependency (PD) Analysis

Step 1. Start
Step 2. */* Transform the AST of the CC into corresponding IR */*
IR = Semantic Analysis (AST)
Step 3. */* Get the specifications of the high- level package */*
Root_Node, Pckg_Name, level, import = get_Top_Pckg (IR)
Step 4. Process_Pckg (Root_Name, Pckg_Name, level, import) {
Step 5. /* Level > MaximumLevel */
If (……..), level++;
Level > Maximum Level
{Return NULL; }
Step 6. */* obtain the series of correlated packages of this class*/*
pckg = build Context. Import (pckg_Name, root)
Step 7. */* Obtain the path of the correlated package*/*
Import_Path = Normalize_Vendor (pckg_Name)
Step 8. Pckg [import Path] = pckg
Step 9. */* Recursive call*
For (……..),
Imp = range get Import (pckg) {
Step 10. if (…….),
Okay = pckg [imp];
!okay {
Step 11. Process_Pckg (pckg_Dir, imp, level, pckg_Name)}}}
Step 12. Return pckgs;
Step 21 Exit

After that, the path, name, and rest specifications of the high level package are acquired through the Intermediate Representation as the key-in of the Process_Pckg function. This is later utilized to read the keyword import in the predefined package to obtain the record of the reliant packages. Next, the algo visits the series of the correlated packages to call the function Process_Pckg recursively to obtain the series of the correlated packages at every layer. Among all packages, the root represents the highest level of dependency with name Pckg_Name, level represents the stage of the present package.

At the point the algo recursively visits the series of correlated packages from the Intermediate Representation, the library given by Golang can be perused at most upto 3rd layer, so the Maximum Level is decided to 3 [30]. At last, the PD relationship of the CC is achieved.

2.4 Functional Dependency (PD) Analysis

FD Analysis utilizes the inclusion based pointer analysis to explore the IR of the CC, having the properties of Static Single Assignment and builds Function Call (FC) Relationship inside the CC. Algorithm 2 represents the process of FD Analysis.

Algorithm 2: Functional Dependency (FD) Analysis

Step 1. Start
Step 2. */*pick the package with main */*
Main = main Package (IR)
Step 3. */*Get the initial function callgraph */*
Step 4. Call Graph = Pointer Analysis (mains)
Step 5. /* Visit all links of the Call Graph */
Call Graph -> Graph_Visit_Links ( ){
Step 6. /* Discard the unwanted callgraphs */
Call Graph -> Discard Unwanted Link ( ) }
Step 7. /* Return the required function call (FC) Relationship */
return call Graph->Get Call Graph ( )
Step 8.Exit

The algo initially scans the IR of the CC to choose the packages along the main function. Afterwards, an Analysis pointer function by Golang is utilized to make analysis on the chosen package to develop the original call Graph function which consists of numerous call relationships considering nodes and links. After that, DFS traversal is used to visit each connection of the original call Graph function. Simultaneously, the algo eliminates the

unwanted links, like call packages related to peer packages, shim etc and keeps the call connections helpful for the next identification. At last, an explicit function call (FC) relationship in the CC is developed.

2.5 CCDI - Chaincode Detection Implementation

CCDI undertakes the job of recognizing possible risks of the CC. The static structural characteristics of the acknowledged risks of CC are extracted to develop a library of features. Next, the static structural characteristics (figured out by CCSA module) are matched with the library to get the final results. It understands the discovery of various risk components via the three detection modules i.e. instruction detection (ID), package detection (PD) and logic detection (LD).

a) The PD module utilizes the DFS algorithm to find out the PD Relationship of the CC. If any unsafe package is found during the exploration of PD Relationship, there exists a parallel risk in the CC. Else, if there exists a suggested package, then there will be no risk. The risk factors which are suggested packages and unsafe packages, recognized by this component, are displayed in Table 2.

TABLE 2: The various Package related risks

Risk Factors	Risky Packages
Random No. Generation	math/rand, ...
Web Services	net/http, ...
System Timestamp	time.Now, ...
Execution of System Command	os/exec, ...
External File Accessing	ioutil, ...
Risk Factors	**The Suggested Packages**
Un-encrypted Delicate Data	crypto/des, ...

b) The ID module utilizes the node properties of the Abstract Syntax Tree of the CC to discover the possible risks. The risk factors that this component can recognize are displayed in Table 3.

c) The LD module determines possible risks as indicated by the path attributes of the FC relationship inside the CC. The risk factors determined by this component incorporate Read Your Write and Range Query Risk (RQR). The RQR mostly validates whether there exists a call path between invoke function and the query function on data range. If there exists a path, that indicates the availability of risk also.

TABLE 2: Risk factors identified by instruction detection

Risk Factors
Global Variable
Iterating over Map Structure
Declarations of Field
Program Concurrency
Reification of Object Addresses
Cross Channel CC Calling
Mechanism for Unutilized Privacy Data

3. MODEL EVALUATION

3.1. Implementation Setup

After the identification of possible risks in CC, the Implementation Setup is planned to assess the correctness and effectiveness of the proposed framework. A crawler has been used to fetch 300 samples of CC by Golang and on Github. The collected CC samples are derived from various business sources and scenarios, like quality execution affirmation, vehicle exchanges. The proposed framework runs on the Linux installed PC, with Intel Core i7 and RAM of 8.00 GB. The model initially identifies the fetched samples of CC and calculates the occurrence count of sixteen risk objects, the count of false negatives and false positives. Then, at that point, the Accuracy, FNR (false negative rate), and FPR (false positive rate) of every identification factor is calculated on the basis of the statistical record of risk factor in CC samples.

3.2. Result Discussion

After testing, the observation was made that 212 out of 300 CC had possible risks, enclosing all of the 16 risk factors. This depicts that the CC created by Golang should be tested to track the possible risks before its deployment. Simultaneously, the FNR, FPR, and accuracy of every detection component of the proposed framework is calculated as displayed in Table 4.

Table 4: FNR, FPR and, Accuracy of every Detection module (DM)

DM of proposed model	FNR	FPR	Accuracy
Package DM	2.5%	4.8 %	96.1%
Instruction DM	4.5%	1.8 %	97.3%
Logic DM	4.4%	3.4 %	95.4%

As per the outcomes listed in Table 4, it can be seen that the precision of every detection element of the proposed framework is more than 95% also, the rate of false alarm of every detection element is below 5%, hence the detection consequence possesses explicit reference value. Hence, the proposed framework can help the developers to design reliable and secure chaincodes.

Table 5. Risk factors which occur frequently

Global-Variable	Generation of Arbitrary Number	Unutilized Data Privacy Mechanism	Unencrypted Delicate Data
35.7%	28.3%	40.4%	26.9%

Table 5 contains the frequently popped up risk factors out of the 16 risk factors in the CC samples. These risk factors are simple to pass over in the development task of CC, so the developers need to focus on these risk factors which occurs regularly, and think about the utilization of the mechanism of encryption and private data more as per explicit situations. During the task of identifying CC samples, the chain codes can be examined to get the analysis outcome within a couple of moments. The client experience is preferable.

4. CONCLUSION

This article is centered around the possible security and privacy risks in the Hyperledger Fabric. It highlights the risks around CC created by the all-purpose coding language, and outlines the 16 counts of possible risks, which are further categorized into 3 groups: Non-deterministic, Security of Personal Data, and Logical Security. To distinguish various risks related to CC, another static examination strategy is proposed. Compared to the existing static investigation strategy, this technique plays out two AST (abstract syntax tree) based analysis models which are FDA (Functional Dependency Analysis) and PDA (Package Dependency Analysis) to get static qualities that can more readily communicate distinctive risks. Simultaneously, utilizing the proposed static examination technique, a risk identification arrangement for the CC has been planned and created by the language, Golang, after applying numerous investigations on the proposed framework.

The outcome represents that the framework can find the risk area with high precision. It likewise compensates for the deficiencies of the standard recognition frameworks of the CC in the security of personal data risk discovery. With an ever increasing number of utilizations of HLF, possible risky instances of the CC created by the all-purpose coding language will keep on being found, so the element library of the proposed framework should be constantly refreshed. As our framework can just recognize the CC created by Golang, this isn't sufficient. Our framework likewise needs to refresh to recognize the CC created by other all-purpose coding language, like Java and Nodejs. As the count of unsafe CC tests keeps on expanding, the utilization of artificial intelligence techniques can be considered to accomplish the recognition of expected risks related to CC.

References

Azaria, A, A Ekblaw, T Vieira and A Lippman (2016). "Medrec: Using blockchain for medical data access and permission management", 2016 2nd International Conference on Open and Big Data (OBD), IEEE, pp. 25–30.

Bettín-Díaz, R, AE Rojas and C Mejía-Moncayo (2018), "Methodological approach to the definition of a blockchain system for the food industry supply chain traceability", in International Conference on Computational Science and Its Applications, Springer, pp. 19–33.

Chohan, UW (2017), "The decentralized autonomous organization and governance issues", Available at SSRN 3082055.

Fanning, K and DP Centers (2016), "Blockchain and its coming impact on financial services", Journal of Corporate Accounting & Finance, 27(5), 53–57.

Goldfeder, S, H Kalodner, D Reisman and A Narayanan (2018), "When the cookie meets the blockchain: Privacy risks of web payments via cryptocurrencies", Proceedings on Privacy Enhancing Technologies, 2018(4), 179–199.

Hofmann, E, UM Strewe and N Bosia (2017), "Supply Chain Finance and Blockchain Technology: The Case of Reverse Securitisation", Springer, New York.

Li, H, L Pei, D Liao, G Sun and D Xu (2019), "Blockchain meets vanet: An architecture for identity and location privacy protection in vanet, Peer-to-Peer Networking and Applications", 12(5), 1178–1193.

Li, J, N Li, J Peng, Z Wu and H Cui (2019). Privacy protection of occupant behavior data and using blockchain for securely transferring temperature records in hvac systems", arXiv: 1904.04715.

Androulaki E , Manevich Y , Muralidharan S (2018), "Hyperledger fabric: a distributed operating system for permissioned blockchains" The Thirteenth EuroSys Conference.

Foschini L , Gavagna A , Martuscelli G (2020), "Hyperledger Fabric Blockchain: Chaincode Performance Analysis", ICC 2020 - 2020 IEEE International Conference on Communications (ICC). IEEE.

D. Harz and W. J. Knottenbelt (2018), "Towards safer smart contracts: A survey of languages and verification methods", CoRR, pages 1–20.

Zhang S, Zhou E, Pi B (2019), "A Solution for the Risk of Nondeterministic Transactions in Hyperledger Fabric", 2019 IEEE International Conference on Blockchain and Crypto currency (ICBC). IEEE, 2019: 253-261.

Huang Y, Bian Y, Li R (2019), "Smart contract security: A software lifecycle perspective", IEEE Access, 7:150184-150202.

Yamashita K, Nomura Y, Zhou E (2019), "Potential risks of hyper ledger fabric smart contracts", 2019 IEEE International Workshop on Blockchain Oriented Software Engineering (IWBOSE). IEEE, 2019: 1-10.

Harish Sukhwani, Jose M. Martinez, Xiaolin Chang, Kishor S. Trivedi, Andy Rindos (2017), "Performance Modeling of PBFT Consensus Process for Permissioned Blockchain Network (Hyperledger Fabric), 2017 IEEE 36th Symposium on Reliable Distributed Systems (SRDS). IEEE.

Ma C, Kong X, Lan Q (2019), "The privacy protection mechanism of Hyperledger Fabric and its application in supply chain finance, Cybersecurity, 2(1): 1-9.

Benhamouda F, Halevi S, Halevi T(2019), "Supporting private data on hyperledger fabric with secure multiparty computation", IBM Journal of Research and Development, 63(2/3): 3: 1-3: 8.

Liu X , Yin W , Yin Q (2010), "A SSA-based intermediate representation technique", 2010 International Conference on Computer, Mechatronics, Control and Electronic Engineering. IEEE.

Z. Li, J. Kang, R. Yu, D. Ye, Q. Deng and Y. Zhang (2018), "Consortium Blockchain for Secure Energy Trading in Industrial Internet of Things,"IEEE Transactions on Industrial Informatics, vol.14, no.8, pp.3690- 3700.

Z. Zheng, S. Xie, H. Dai (2016), "Blockchain Challenges and Opportunities: A Survey", Web and Grid Services. 2016, pp.1-25.

M. Vukolic (2015), "The quest for scalable blockchain fabric: Proof-of-work vs. BFT replication," in Open Problems in Network Security - IFIP WG 11.4 International Workshop, Zurich, Switzerland.

E. Androulaki, A. Barger, V. Bortnikov, C. Cachin, K. Christidis, A. D. Caro, D. Enyeart, C. Ferris, G. Laventman, Y. Manevich, S. Muralidharan, C. Murthy, B. Nguyen, M. Sethi, G. Singh, K. Smith, A. Sorniotti, C. Stathakopoulou, M. Vukolic, S. W. Cocco, and J. Yellick (2018), "Hyperledger fabric: A distributed operating system for permissioned blockchains," in Proceedings of the 13th ACM SIGOPS European Conference on Computer Systems, Porto, Portugal.

V. Buterin (2016), "Ethereum platform review: Opportunities and challenges for private and consortium blockchains," 2016. [Online]. Available: http://r3cev.com.

Kumar, Pravin; Dayal, Mohit; Khari, Manju; Fenza, Giuseppe; Gallo, Mariacristina, "NSL-BP: A Meta Classifier Model Based Prediction of Amazon Product Reviews", International Journal of Interactive Multimedia & Artificial Intelligence. June 2021, Vol. 6 Issue 6, p95-103. 9p.

Son, L.H., Jha, S., Kumar, R. et al. Collaborative handshaking approaches between internet of computing and internet of things towards a smart world: a review from 2009–2017. Telecommun Syst 70, 617–634 (2019). https://doi.org/10.1007/s11235-018-0481-x

M. Khari, A. K. Garg, A. H. Gandomi, R. Gupta, R. Patan and B. Balusamy, "Securing Data in Internet of Things (IoT) Using Cryptography and Steganography Techniques," in IEEE Transactions on Systems, Man, and Cybernetics: Systems, vol. 50, no. 1, pp. 73-80, Jan. 2020, doi: 10.1109/TSMC.2019.2903785.

N. Singh, M. Dayal, R. S. Raw and S. Kumar, "SQL injection: Types, methodology, attack queries and prevention," 2016 3rd International Conference on Computing for Sustainable Global Development (INDIACom), 2016, pp. 2872-2876.

K. S. Sahoo et al., "An Evolutionary SVM Model for DDOS Attack Detection in Software Defined Networks," in IEEE Access, vol. 8, pp. 132502-132513, 2020, doi: 10.1109/ACCESS.2020.3009733.

Mohit Dayal and Bharti Nagpal, "A compendious investigation of Android malware family", International Journal of Information Privacy, Security and Integrity-2016, doi: 10.1504/IJIPSI.2016.082127.

Chitrangada Chaubey, Manas Kumar Nanda, "Cloud Database Management System Architecture", International Journal of Electrical Engineering and Technology (IJEET) Volume 11, Issue 10, December 2020, pp 260-266., Article ID: IJEET_11_10_036 ISSN Online: 0976-6553
DOI: 10.34218/IJEET.11.10.2020.036.

M. M. Oliveira and R. S. Cruz, "Distributed Ledger Technology to Enable Secure Management of IT Infrastructures: Development and evaluation of a Proof-of-concept tool using Hyperledger Fabric," 2022 17th Iberian Conference on Information Systems and Technologies (CISTI), 2022, pp. 1-7, doi: 10.23919/CISTI54924.2022.9820240.

S. Jain , "Smart Contract - Security Assessment Integrated Framework (SC-SIF) for Hyperledger Fabric," 2022 IEEE 7th International conference for Convergence in Technology (I2CT), 2022, pp. 1-11, doi: 10.1109/I2CT54291.2022.9824439.

P. Gaba, R. S. Raw, M. A. Mohammed, J. Nedoma and R. Martinek, "Impact of Block Data Components on the Performance of Blockchain-Based VANET Implemented on Hyperledger Fabric," in IEEE Access, vol. 10, pp. 71003-71018, 2022, doi: 10.1109/ACCESS.2022.3188296.

A. Shahidinejad and D. Abbasinezhad-Mood, "Ultra-Lightweight and Secure Blockchain-Assisted Charging Scheduling Scheme for Vehicular Edge Networks by Utilization of NanoPi NEO," in IEEE Transactions on Vehicular Technology, vol. 71, no. 8, pp. 8116-8123, Aug. 2022, doi: 10.1109/TVT.2022.3173076.

W. Liang ., "PDPChain: A Consortium Blockchain-Based Privacy Protection Scheme for Personal Data," in IEEE Transactions on Reliability, 2022, doi: 10.1109/TR.2022.3190932.

J. Peng, C. Li, B. Yuan, H. Meng, F. Yu and X. Li, "SCPS:A Secure and Copyright-Preserving System for IIoT Based on Hyperledger Fabric," 2021 Ninth International Conference on Advanced Cloud and Big

*e*ISSN 1303-5150

Data (CBD), 2022, pp. 224-229, doi: 10.1109/CBD54617.2021.00046.

S. Tang, Z. Wang, J. Dong and Y. Ma, "Blockchain-Enabled Social Security Services Using Smart Contracts," in IEEE Access, vol. 10, pp. 73857-73870, 2022, doi: 10.1109/ACCESS.2022.3190963.

J. Maeng, Y. Heo and I. Joe, "Hyperledger Fabric-Based Lightweight Group Management (H-LGM) for IoT Devices," in IEEE Access, vol. 10, pp. 56401-56409, 2022, doi: 10.1109/ACCESS.2022.3177270.