Research Article

International Journal of Advanced Technology and Engineering Exploration, Vol 11(112)
ISSN (Print): 2394-5443 ISSN (Online): 2394-7454
http://dx.doi.org/10.19101/IJATEE.2023.10101898

Enhancing data security in cloud computing: a blockchain-based Feistel cipher encryption and multiclass vector side-channel attack detection approach

Ramakrishna Subbareddy* **and P. Tamil Selvan**
Department of Computer Science, Karpagam Academy of Higher Education, Coimbatore, India

Received: 27-June-2023; Revised: 27-March-2024; Accepted: 28-March-2024
©2024 Ramakrishna Subbareddy and P. Tamil Selvan. This is an open access article distributed under the Creative Commons Attribution (CC BY) License, which permits unrestricted use, distribution, and reproduction in any medium, provided the original work is properly cited.

Abstract

Cloud computing (CC) environments offer cost-efficient and flexible resources, appealing to users despite concerns about the reliability of cloud service providers (CSPs) and data privacy. To address these concerns, encrypting data before outsourcing to the CC environment is essential. However, encryption introduces challenges such as data leakage through side-channels in virtual machines (VMs). To address these issues, a Feistel cipher symmetric encryption with multiclass vector (FCSE-MV) side-channel attack detection method was developed, leveraging blockchain technology in CC. Initially, an aggregated Byzantine fault tolerance-based block generation model was employed for efficient block production. Subsequently, the presence or absence of side-channel attacks was determined using an FCSE-MV-based block validation model. Experiments conducted with the SCAAML database in JAVA demonstrated that FCSE-MV improved accuracy and throughput by 17.5% and 19%, respectively, and reduced communication complexity and attack detection time by 24% and 21%, compared to traditional attack detection methods. The proposed FCSE-MV method offers a secure and efficient solution suitable for CC environments.

Keywords

Virtual machine, Block generation, Block validation, Byzantine fault tolerance, Feistel deterministic cipher, Symmetric encryption, Multiclass vector.

1.Introduction

Cloud computing (CC) permits more users for storing as well as distributing their applications and data. CC continuously evolving technology that can improve agility, availability, collaboration, and scalability of data. It offers on-demand solution for different applications namely data storage, servers, databases, networking, and software. One of the most challenging issues in CC is detecting attacks. Parallel sponge-based authenticated encryptionwith side-channel protection and adversary-invisible nonces (PSASPIN) to sponge construction was designed [1]. Moreover, leveled implementation has also utilized in implementing key generation via pseudo random function (PRF). Next, data preprocessing has utilized to minimize the code size. Finally, security proof has ensured using game theory and therefore processing shorter messages significantly. The authenticated encryption (AE) schemes such as security, performance, and efficiency are enhanced. Despite improvement found in terms of communication complexity and throughput were not focused.

Side channel assault also utilizes physical data which leaks as of cryptographic tool. For avoiding from these attacks, divide-and-conquer approach is employed. Here, malicious user splits key into sub-keys, endeavors for recuperating sub-keys distinctly as well as integrates them to create key. Reinforcement-strategy-based side-channel attack has designed in [2] with divide-and-conquer. Initially, the power leakage models are modified, and accuracy has improved. Next, the recovered sub-keys are integrated during the attack phase through validation therefore mitigating side-channel attacks significantly and improving the successful recovery rate. Though successful recovery rate has ensured but the detection time and accuracy have not focused. The disclosure of the cloud storage in the recent years carries enormous east to person life. Moreover, business establishments as well as organizations transmitted huge data for storage also. However, the data privacy

*Author for correspondence

has to be protected. Nevertheless, the flexibility of reinforce file was not updated by using restrict methods. Blockchain-enabled public key encryption scheme with multi-keyword search (BPKEMS) was proposed in [3] for addressing on these aspects. Shared key without any interaction has obtained with Diffie–Hellman (DH) key exchange protocol. Blockchain mechanism has employed with guarantee the transaction. The smart contract utilized for achieving accuracy for verifying of ciphertext. With this type of design communication overhead has said to be reduced.

Yet another dynamic searchable symmetric encryption for multiuser (M-DSSE) mechanism to access them by issuing keyword search queries [4]. It comprises forward and backward security. Pseudorandom functions as well as homomorphic message authenticator (HMAC) have utilized for making reversed index as well as modernize search token. Efficiency of file has increased by symmetric encryption. To ensure secure and smooth communication symmetric encryption was employed that in turn improved throughput. However, the attack detection time has said to be higher.

Also, with the objective of addressing the issue which major of prevailing blockchain-basis of searchable encryption method provide precise keyword search as well as mistrust among cloud security (CS) as well as Cloud user (CU), in [5], searchable encryption method with blockchain framework has designed via fuzzy keyword that in turn ensured semantic security. Blockchain technology has integrated by using rivest–shamir–adleman (RSA) dynamic stasher that permits in) formation owner to modernize document. The smart contract has to confirm search outcomes. The fuzzy keyword search has performed to create fuzzy keywords set. But the throughput was not considered.

Side-channel attacks initiate grave menace to practical systems; they disclose pertaining to calculating internal process while executing the program. To circumvent these issues, research worker contains locate immense endeavor to safe cryptographic primordial methods as well as crafting efficient corrective actions. For minimizing latency, a novel and robust new blockchain-based distributed key management architecture (BDKMA) was designed [6]. The method proposed with the aid of blockchain mechanism not only satisfied decentralization, high scalability but also ensured privacy preservation for hierarchical access control. A new blockchain concept has presented for finding permit entrance queries with no trusted third party. Side blockchains preserved with security access managers (SAMs)multiblock chains has stored at cloud.

Hash-based authenticated nonce-misuse resistant encryption (HANMRE) has proposed in [7]. Benefit of this method has provisioning of security. Despite improvement found security aspects, the throughput rate along with the time and accuracy factors involved in the secured data transmission between CUs in the event of side-channel attack was not considered. To overcome the above problem feistel cipher symmetric encryption and multiclass vector (FCSE-MV) side-channel attack detection based on blockchain in CC environment is designed results in enhancement of throughput rate with minimum communication complexity and higher rate of accuracy.

Major aim of this manuscript is precisely recognizing side-channel attack using blockchain based method. A few researchers employed outmoded established events, or they made employ of sluggish, time utilizing. Along with literature review performed, work's incentive is to carry out attack detection depend on blockchain, which has not considered into description in history. Therefore, this problem has taken to consider in this paper. This work uses FCSE-MV-based block validation algorithm for block generation and validation.

The main contributions of this study are: The paper takes the network traces in the form of blocks (i.e., in blockchain), applied to the CC environment and uses, FCSE-MV method in an integrated manner ensuring minimum communication complexity and attack detection time. To design aggregated Byzantine fault tolerance-based block generation model with the objective of generating the blocks for the corresponding network traces acquired as input from side channel attacks assisted with machine learning dataset (SCAAML) database with minimum communication complexity and higher rate of throughput. The proposed FCSE-MV-based block validation algorithm is detecting different kinds of side-channels in efficient manner. Further, four experiments on the FCSE-MV method are conducted from distinct facets (sides). The results show that the side-channel attack detection with blockchain in CC environment has good with improved throughput rate and attack detection accuracy, also reducing the communication complexity and attack detection time.

The structure of manuscript is as follows: section 1 covers the introduction of cloud background, challenges of preceding literature, motivation, objectives as well as contribution of manuscript. Section 2 presented reviews related work concerning symmetric encryption, application of symmetric encryption with blockchain in CC framework and machine learning (ML) techniques applied for detecting side-channel attacks. Section 3 introduces the method, FCSE-MV side-channel attack detection with blockchain in CC framework. Section 4 provides experimental setup with detailed quantitative analysis through extensive simulations. Section 5 describes discussion and limitation. Section 6 summarizes the manuscript.

2.Related works

Information security has becoming crucial distress over the past few years as ingenious and smart security attacks continue on popping up. With the appearance of present-day fields like blockchain, CC and cyber-physical systems (CPS), the volume of digital data generated is said to be increasing. Malicious virtual machine (VM) that allocate hardware resources (co-reside) with other VMs acquire their information and does the act of side channel attacks (SCAs). A proportionate amount of research has been done in the area of cryptography resulting in the evolution of several crypto algorithms.

Deep learning (DL) algorithms have utilized in [8] with maximum security. Long short-term memory (LSTM) model comprises feature selection and extraction. LSTM has developed to process sequential data as well as capture temporal dependencies. Radial basis function (RBF) employed for discovering the benign attacks. But the time has higher. A privacy preserving keyword search employing symmetric encryption has performed in [9] with which, data security has said to be ensured. The encryption and decryption carried out by using asymmetric searchable encryption and cloud service provider (CSP). But the longer time utilized for encryption. Yet another multi-level security aspects have included in [10] by employing proxy re-encryption. The algorithm performance was enhanced. With this, not only the time involved in overall process was reduced but also resulted in the fine-grained delegation.

Several security researchers were discussed in [11] for CS. ML was major task in cloud to find diversity among legitimate and malicious attacks. DL is a branch of ML to offer amazing performance in CS issues. Yet another survey to ensure confidentiality and integrity in cloud was presented in [12]. National institute of standards and technology progresses within reach of producing regular decisions and focuses in accurately evaluating the security of the models, in addition to the SCA and their corrective actions. Hence, security beside SCA as well as recital of side-channel safeguarding the benchmark to choose corresponding standards. Defense mechanisms have classified in [13] for distributed denial-of-service (DDoS) attacks. Naive bayes algorithm (NBA) was employed in [14] to recognize the actual rate of transmissions. However, it failed to minimize the false positive rate (FPR).

Nevertheless, the decoding steps were however found as susceptible to SCA. A novel (unique) as well as robust key recovery SCA on hamming quasi-cyclic (HQC) was presented in [15]. Through feature of reuse of static secret key attack success rate has found to be advantageous.

Yet another systematic classification of side-channel attack types was made in [16]. Hybrid approach for low rate DDoS Detection (HA-LRDD) has examined in [17] for discovering low-rate DDoS attacks. But the accuracy has not enhanced. Stacked contractive autoencoder (SCAE) method has utilized in [18] to accurately find low-dimensional features. A novel cloud intrusion detection system (IDS) has introduced by support vector machine (SVM) with less overhead. Yet another post quantum signature mechanism has employed in [19] to combat against side-channel attack. A decentralized blockchain network employing lightweight finite field encryption has proposed in [20].

New attack detection framework has introduced in [21] with maximum detection accuracy. But the complexity has not minimized. DL algorithm has developed in [22] for discovering DDoS attacks. Blockchain technology was discussed in [23] for security threats. Honey badger algorithm with an optimal hybrid deep belief network (HBA-OHDBN) technique was utilized [24] to enhance network security. New IDS model based on DL algorithms were investigated in [25] to find intrusion. However, the computational efficiency has not decreased.

Adaptive multi-factor multi-layer authentication framework has designed in [26] for preserving the security of data. Ransomware attack detection has analyzed in [27] for improving CS. Novel network intrusion detector named the multi-branch model has utilized in [28] with lesser communication. The security threats have addressed in [29] by using secure and intelligent framework. New hybrid meta-heuristic adaptive particle swarm optimization (PSO) whale optimizer method has introduced in [30] with attaining security performance.

An improved intelligent intrusion detection method was investigated in [31] via DL. However, the efficient algorithms failed to improve the accuracy of the designed model. DL-based IDSs approaches have designed in [32] to different criteria. Collaborative intrusion recognition scheme-basis of deep blockchain framework (DBF) developed in [33] for the identification of cyber-attacks. ML based intrusion detection was executed in [34] for internet of things (IoT) botnet. IoT was a globally accepted technology designed in [35] to optimize security protocols.

A ML technique has developed in [36] to perform classification. Blockchain and ML approach has investigated in [37] for security attacks recognition. Destructive activities were determined in [38] by anomaly detection system (ADS). Novel optimized DL model [39] and secure framework [40] has presented to recognize threats. An advanced IDS has examined in [41] for industrial internet of things (IIoT). However, accuracy was not detected.

ADL based intrusion detection framework [42] developed to detect security threats in IoT environments. Unified IDS has developed in [43] for IoT. Deep random neural network (DRaNN) model [44] and regularized cross-layer ladder network [45] have designed for intrusion detection. Dissimilar ML techniques has introduced by [46] for cyber-vulnerability evaluation. Forensics- based DL method called deepfloyd IF (Deep-IF) has employed in [47] to discover attacks. Recent IDSs has examined in [48] for IoT. Botnet attacks were identified in [49] by high-performing DL models. Automated and robust IDS system has designed in [50] for IoT. However, it failed to minimize the detection time.

Safe as well as trustworthy medical documentation sharing system was designed in [51]. Adequate technique for identifying cyber intrusion is introduced in [52]. An efficient service constraint network condition and behavior quality (NCBQ) scheme is proposed in [53] to include the secure transmission by utilizing different factors of network, user behavior and quality of service. A block chain-based identity authentication system (IAS) protocol is performed in [54] to ensure the authenticity and security of data transmission in CC. In [55], secure-ring-verification-based authentication (SRVA) method is designed to guarantee safety. But detection accuracy was not enhanced.

A vehicle-based secure blockchain consensus (VBSBC) algorithm is carried in [56] to enhance the efficiency of data storage, processing, and sharing on the internet of vehicles (IoV). A blockchain and smart contract technology is designed in [57] to core elements and ensured the data integrity with build a decentralized verification method. Blockchain-based fair as well as reliable searchable encryption system is proposed in [58]. Technology as well as safety investigation of crypto currency depends on block chain is designed in [59]. Security method is designed in [60] to notice the malicious events, by data gathering as well as anomaly recognition. However, it failed to focus on the computational complexity.

Motivated through the discussed limitations and advantages, new and robust technique for circumventing against side-channel attacks using blockchain in CC environment called, FCSE-MV is introduced to secured data transmission with blockchain in CC environment is to enhance accuracy and throughput and minimize FPR and time.

3.Method

Blockchain technology is seen as a disruptive technology that can participate key part during securing against side-channel attacks that is most proliferating in CC environment. While decentralized, distributed pattern, blockchain utilizes block cipher symmetric cryptographically associated series of blocks for validating as well as store procedure information in CC environment. Consensus algorithm utilized through processing nodes (i.e., CUs) in generating the blocks. On the other hand, smart contracts, which are said to be programmable scripts executed in an automatic manner which is utilized in manipulating the data. Generated blocks contain header, body. The elements of block header are crucial in producing precise and reliable header. It comprises of current version number ‘V_No’, address of previous block (PBK) previous block header (PBH) that efficiently secures the chain by being associated with previous block, target hash value of current block.

Moreover, it also includes, a Merkle root '*M Root*' consist of every hashed transaction in hashed transaction, a nonce 'Nonce' that refers to a value utilized in creating distinct numbers of permutations for generating an accurate hash in the sequence, and finally timestamp 'T' that permits to look at the encoded record at a specific time. On the other hand, the block body usually comprises of the actual transactions. In *Figure 1* with the SCAAML database given as input, above 8000 samples instances acquired for experimental, every trace is envisaged in appearance of blocks. This is formulated as input vector matrix (IVM) as given below by Equation 1.

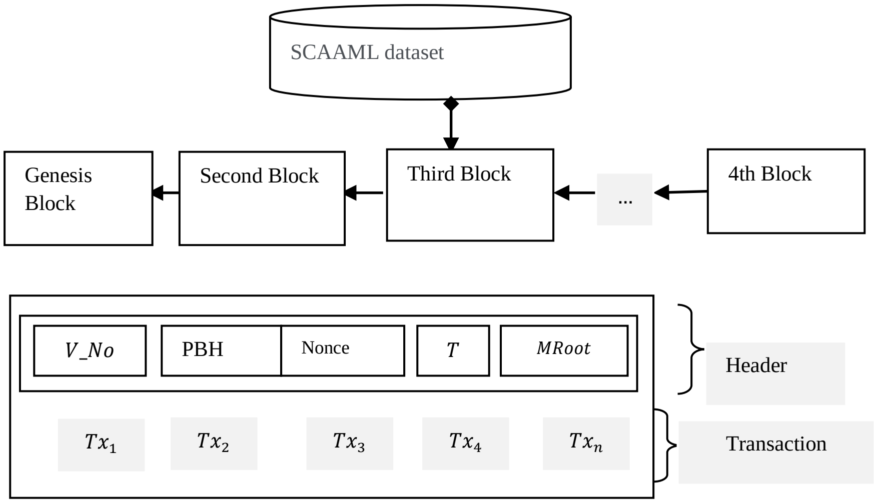

Figure 1 Block structure of FCSE-MV side-channel attack detection based on blockchain in CC environment

$$IVM = \begin{bmatrix} S_1TR_1 & S_1TR_2 & \dots & S_1TR_n \\ S_2TR_1 & S_2TR_2 & \dots & S_2TR_n \\ \dots & \dots & \dots & \dots \\ S_sTR_1 & S_sTR_2 & \dots & S_sTR_{tr} \end{bmatrix} \qquad (1)$$

From the above Equation 1, through SCAAML dataset given as input, 'IVM' is formulated depend on samples 'S_s' with their corresponding traces 'TR_tr' respectively. Amongst dissimilar types of SCA, our work focus is made on analyzing cache-based, timing attacks, data remanence and differential fault analysis. Cache basis of attack is types of side-channel attacks where shared cache exploited to acquire generated block information by analyzing the hit ratio also time required for accessing guaranteed cache lines. Timing attack is one more sort of SCA according to time several computations take to perform. Data remanence is kind of SCA where created block perceptive information is read and also removed. Differential fault analysis is sort of SCA wherein the created block is attained through developing faults. With the formulated 'IVM' block generation and block validation are performed in the following sections to combat against side-channel attacks via CC environment.

Aggregated Byzantine fault tolerance-based block generation model

Based on the block information as provided above, first block generation is performed in this work via Consensus algorithm. Blockchain referred as decentralized, immutable, transparent digital leger that records transactions across multiple computers or nodes. This eradicates the need to lessen the risk of side-channel attacks. Information is structured to blocks as well as every block includes transaction. Every novel block attach to every block previous to at cryptographic chain. All transactions within the blocks are validated with consensus mechanism. The decentralized nature of the technology also permits for enhanced transparency. In addition, the security and transparency is improved. Upon reception of a block, nodes obtain unavailable during process of voting prior reaching consensus. The consensus algorithm is a center part of the blockchain network because it attains agreement, trust, and security across a decentralized network. Two consensus methods namely proof of work and the proof of stake employed. A blockchain process is based on the consensus mechanism to verify a block valid transaction. Hence, it handles faults and errors. In this work, the aggregated Byzantine fault tolerance consensus algorithm verifies the correctness of a block via trusted 'CSP'. Each processing node can become a leader or header node owing to the reason that each has absolute access to the transaction.

Several blocks are associated via hash pointers therefore forming chain. Symmetric encryption is then utilized to sign the encrypted transactions with the objective of guaranteeing their authenticity. Followed by which signed transactions afterward symmetrically associated to construct tamper-proof block. *Figure 2* given below shows the structure of aggregated Byzantine fault tolerance-based block generation model.

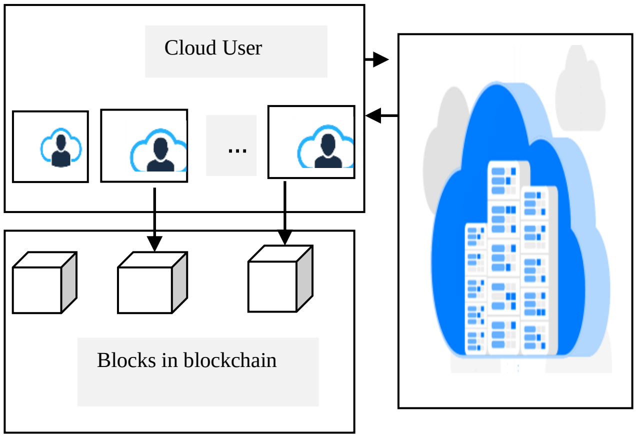

Figure 2 Structure of aggregated Byzantine fault tolerance-based block generation model

In above *Figure 2* through CUs blocks formulated in the form of input vector matrices, the ‘CSP’ performs the task of block generation with minimum communication complexity and maximum throughput. The CSP in the work is a trustable entity handled by the administrators of system. Information dwelled inside these trusted ‘CSP’ cannot be established through malicious user. It gives divide execution as well as memory security. Four distinct operations are performed during aggregated Byzantine fault tolerance-based block generation. They are, initialization, prepare, execute and validate/update, respectively. Initial, CU sends a request to CSP with request in form of a block ‘B’ by Equation 2.

$$CU: CSP \rightarrow (Req, CU_i, CSP, t, B_i) \qquad (2)$$

From the above formulation (2), the above request is sent by a ‘CU_i’ to the ‘CSP’ with a request block ‘B_i’. In addition, the time instance ‘t’ is utilized in ensuring single semantics of CU request execution. Second the ‘CSP’ upon reception of request from CU assigns the task of ‘$r+1$’ resources and is mathematically formulated as given below by Equation 3.

$$CSP: ACU \rightarrow (Prepare, RNo, MD(B_i), B_i, CSP)(3)$$

From the above Equation 3, upon reception of the block ‘B_i’ from cloud user ‘〖CU〗 _i’, ‘CSP’ issue a random number ‘RNo’ where ‘MD(B_i)’ denotes the message digest for the corresponding block (i.e., active CU block) respectively. Now the active CUs ‘ACU’ block represented by ‘〖AB〗 _i’ acquires task from ‘CSP’, process as well as send outcome to ‘CSP’ itself. Third, upon acquiring the task from ‘CSP’, active CU block ‘AB_i’ verifies the message digest of its corresponding block for guarantying its integrity. In addition, random number ‘RNo’ should be ‘RNo_{g+1}’ where ‘$g+1$’ is the greatest random number. Finally, also verification regarding the active CU block ‘AB_i’ must not be with indistinguishable random number but necessitating different request. This is mathematically formulated as given below by Equation 4.

$$AB_i: CSP \rightarrow (Result, MD(\boldsymbol{AB}_i), B_i, CSP) \qquad (4)$$

Finally, upon reception of results acquired from all active cloud users ‘ACU’ block represented by ‘AB_i’, CU provider compare with the random number ‘RNo’ and ‘$Result$’ of every resource with valid signature. This is mathematically formulated as given below by Equation 5.

$$CSP: AB_i \rightarrow (Result, RNo, MD(AB_i), CSP) \quad (5)$$

From the above Equation 5, the ‘CSP’ sends the above result to ‘AB_i’ and also sends an update message ‘$Update$’ as given below by Equation 6.

$$CSP: AB_i \rightarrow (Update, Result, U, RNo, CSP)\ (6)$$

In this manner, as given in the above Equation 6, by means of update information ‘U’ several blocks are then linked therefore forming a chain. *Figure 3* Workflow diagram of Aggregated Byzantine Fault Tolerance Consensus algorithm is given below.

The pseudo code representation of Aggregated Byzantine Fault Tolerance Consensus is given in Algorithm 1.

Algorithm 1 Aggregated Byzantine fault tolerance consensus algorithm

```
Input: Dataset ‘$DS$’, Cloud User ‘$CU = CU_1, CU_2, \ldots, CU_m$’, Block ‘$B = B_1, B_2, \ldots, B_n$’, Cloud Service Provider ‘$CSP$’
Output: Throughput-improved Blocks generated
Initialize Samples ‘$S = S_1, S_2, \ldots, S_s$’, Traces ‘$TR = TR_1, TR_2, \ldots, TR_{tr}$’, ‘$s$’, ‘$tr$’
Begin
For each Dataset ‘$DS$’ with Cloud User ‘$CU$’, Block ‘$B$’, Samples ‘$S$’ and Traces ‘$TR$’
Formulate input vector matrix as given in (1)
//Initialization
Cloud user send request to Cloud Service Provider ‘$CSP$’ as given in (2)
//Prepare
```

```
6: Cloud service provider add a random number to the requested cloud users block and send it to all active cloud users block as given in (3)
**//Execute**
7: Active cloud users block execute the task sent from cloud service provider and send back to cloud service provider as given in (4)
**//Validate and Update**
8: Perform validation and accordingly update as given in (5) and (6)
9: **Return** blocks generated ‘$BG$’
10: **End for**
11: **End**
```

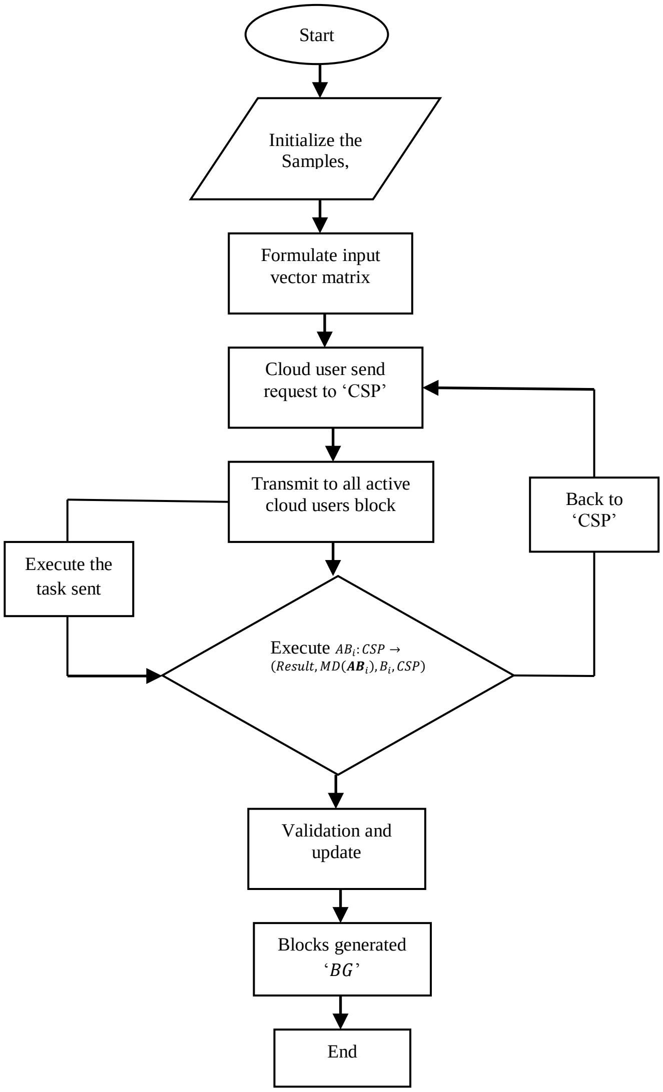
Start
Initialize the Samples,
Formulate input vector matrix
Cloud user send request to ‘CSP’
Transmit to all active cloud users block
Execute the task sent
Back to ‘CSP’
Execute $AB_i: CSP \rightarrow (Result, MD(\boldsymbol{AB}_i), B_i, CSP)$
Validation and update
Blocks generated ‘BG’
End

Figure 3 Workflow diagram of aggregated Byzantine fault tolerance consensus algorithm

In algorithm 1, with the traces (i.e., sample instances) acquired from SCAAML dataset in the form of blocks as input, objective enhancing throughput rate is with lesser communication complexity. With this objective, consensus method is designed that is split into four parts, i.e., initialization, prepare, execute and validate/update respectively. According to the design of four distinct parts, by means of aggregate function minimizes the resources by creating a subgroup of ‘r+1’operative and remaining ‘r’ non-operative resources and when the fault occurs during the generation of blocks, the non-operative resources become operative to generate blocks with the prevailing ‘r+1’ operative resources perform against fault occurred in the network. In this manner, blocks are said to be generated with minimum communication complexity and number of CU requested processed (i.e., throughput) is said to be improved.

FCSE-MV-based block validation

Smart contract a program which should run in container given through blockchain scheme in this work is executed in an arbitrary fashion with the purpose of validating the generated blocks by employing Feistel deterministic cipher and symmetric encryption via multiclass vector. Also, outcomes generated through smart contract saved in block, which ensures authenticity. However, a significant amount of risk of information leakage across VM is said to take place via side-channels. In this section, three types of side-channel attacks are focused via multiclass vector (i.e., SVM learning) and smart contract not only circumvent malicious tampering through rules, however, take benefit of minimizing falsification with minimum attack detection time. *Figure 4* shows the structure of FCSE-MV vector-based block validation.

As illustrated in the above *Figure 4*, the Feistel cipher-based symmetric key employed in our work utilizes the same cryptographic keys for both encryption of plaintext (i.e., generated block plaintext) and the decryption of ciphertext (i.e., generated block ciphertext). Feistel cipher is employed in edifice of block ciphers. The Feistel structure utilized to perform encryption and decryption operations. Key scheduling method obtains plaintext of 128-bit as well as key input of 128-bit, 192-bit, 256-bit. Encryption ‘Enc’ and decryption ‘Dec’ are carried out through CSP using smart contract. The smart contracts performance is measured in cloud. Automation decreases require for intermediaries and manual processes, leading to earlier transactions for enhanced efficiency. Blockchain technology guarantees that smart contracts are secure and tamper-proof with higher security.

Followed by which validated results are classified into three types of side-channel attacks upon detection of attack or normal, therefore ensuring smooth and safe communication between users in CC environment. Let ‘RF’ represents the round function and ‘$SK_0, SK_1, \dots, SK_u$’ represent the sub keys for the sequences ‘$0,1,\dots,u$’ respectively. Then, to start with the block generated plaintext ‘BGP’ is partitioned into two blocks ‘(BGL_0, BGR_0)’ by Equations 7 and 8.

$$BGL_{i+1} = BGR_i \qquad (7)$$

$$BGR_{i+1} = BGL_i \oplus Fun(BGR_i, SK_i) \qquad (8)$$

From the above Equations 7 and 8, for each sequence the partitioned blocks are measured via XOR function ‘$\oplus$’. Then, the block generated cipher text is represented as given below by Equation 9.

$$BGC = (BGR_{u+1}, BGL_{u+1}) \qquad (9)$$

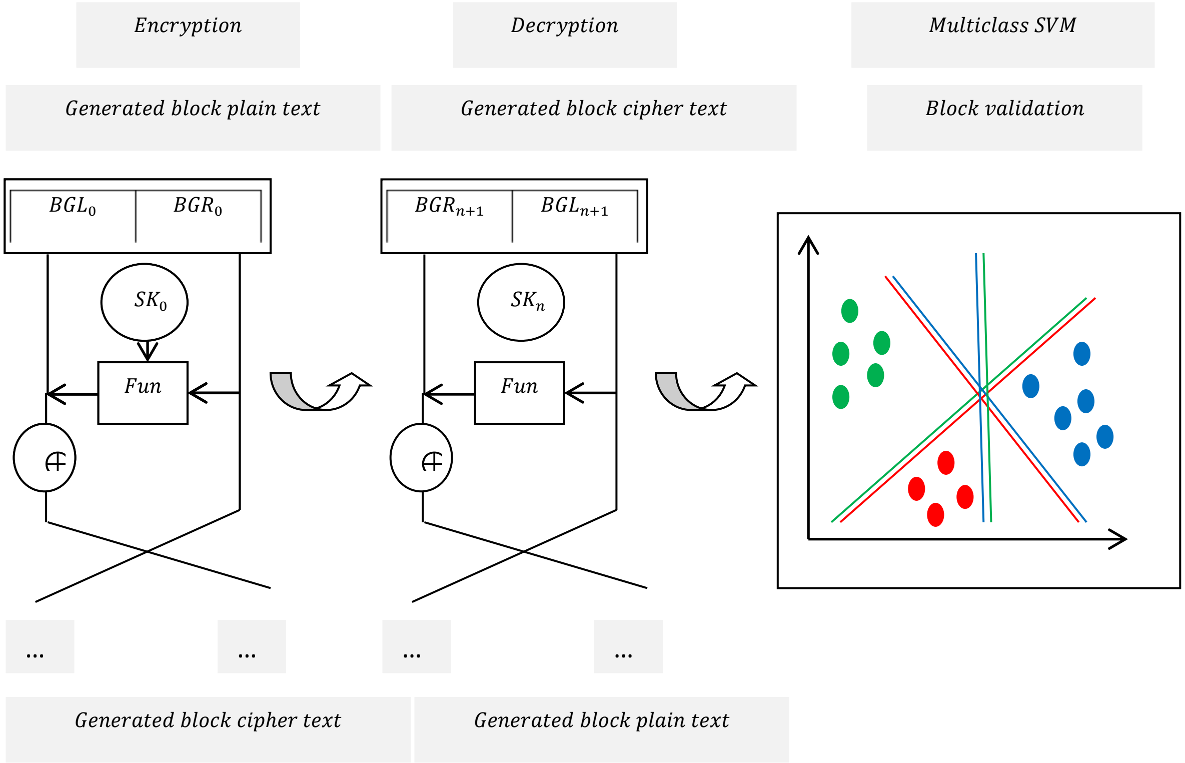

Figure 4 Structure of Feistel deterministic cipher symmetric encryption and multiclass vector-based block validation

From the above Equation 9, the block generated cipher text ‘BGC’ developing Ethereum smart contract for sender and receiver, each contract call is registered in blockchain. Hence, data transfers among CU (i.e., sender), CU (i.e., receiver) is tamper resistant as well as non-repudiation. ‘CSP’ appends sub keys ‘〖SK〗_0, 〖SK〗_1,…, 〖SK〗_u’ for the sequences ‘0,1,…,u’ which is then cached. Finally, the sub keys, sequences and the block generated cipher text uploaded to blockchain. As CU put request for block generated data access, authenticity is verified by checking the sub keys of sender and receiver. Moreover, a timestamp is also said to be appended upon successful authentication. The related information on the block generated cipher text is fetched from the cache. The CU (i.e., receiver) can currently decrypt ciphertext by his sub keys. Blockchain now confirm timestamp and request is stored on the blockchain. The decryption process is given below by Equations 10 and 11.

$$BGR_i = BGL_{i+1} \qquad (10)$$

$$BGL_i = BGR_{i+1} \oplus Fun(BGL_{i+1}, SK_i) \qquad (11)$$

From the above Equations (10) and (11) decryption of block generated cipher text ‘BGC’ is performed by computing the sequence for a fixed number of times. Finally, the block generated plaintext ‘BGP’ is mathematically stated as in Equation 12.

$$BGP = (BGL_0, BGR_0) \qquad (12)$$

Finally, according to the above results obtained, block validation is performed by means of multiclass SVM. Based on the hypothesis of binary classification performed using SVM. It is one of the most popular ML algorithms employed for classification. SVM method is mostly representation of dissimilar classes at hyperplane in multidimensional space. Aim of SVM is divide databases to classes to discover maximum marginal hyperplane. The implementation of the SVM algorithm is given below. Dataset was loaded. Database is separated to training as well as testing. Dataset includes multiclass categories of data denoted by blue, green and red. A SVM classifier works with making a straight line among two groups of data. In data points, one side of the line is denoted a particular category and other side of the line represented as dissimilar category. The classification is performed to classify the pair of coordinates in red, green or blue, which is plotted in *Figure 4*. Multi classification (i.e., multiclass) is performed in our work by splitting down into multiple binary classification by Equation 13.

$$w^T BGP - b = 0 \qquad (13)$$

From Equation 13, 'w' indicates normal vector to the hyperplane and '$\frac{b}{w}$' determines the hyperplane offset from origin for determining multiclass (i.e., identifying three types of side-channel attacks) by Equations 14, 15 and 16.

$$w^T BGP - b \geq 1, [MR = MR + 1],\ if Output_i = 1\ [cacheattack] \qquad (14)$$

$$w^T BGP - b \leq -1, [\mathrm{HR} = \mathrm{HR} + 1],\ if Output_i = -1 [timingattack] \qquad (15)$$

$$w^T BGP - b = 0, [BGP \neq BGC],\ if Output_i = 0\ [dataremanenceattack]] \qquad (16)$$

As given in the above Equations 14, 15 and 16, with the aid of multiclass SVM, addresses the issues of side-channel susceptibilities engrossing central processing unit (CPU) cache, timing as well as data remanence shortcoming of previous defenses during CC environment. *Figure 5* Workflow diagram of FCSE-MV -based block validation is given below.

The pseudo code representation of Feistel deterministic cipher symmetric encryption and multiclass vector-based block validation is shown in Algorithm 2.

In Algorithm 2 for minimizing FPR in minimum amount of time. First, by applying the Feistel deterministic block cipher for each block generated via encryption and decryption using the same keys, attack detection time is reduced. Also, by efficient differentiation between the data points from block generated cipher text and plain text, careful and cautious separation of hyperplane is made by maximizing the margin with minimum distance. As a result, the attack detection accuracy is enhanced.

Algorithm 2 Feistel Deterministic Cipher Symmetric Encryption and Multiclass Vector-based Block Validation

```
Input: Dataset '$DS$', Cloud User '$CU = CU_1, CU_2, \ldots, CU_m$', Block '$B = B_1, B_2, \ldots, B_n$', Cloud Service Provider '$CSP$'
Output: Computationally efficient attack detection
Initialize Samples '$S = S_1, S_2, \ldots, S_s$', Traces '$TR = TR_1, TR_2, \ldots, TR_{tr}$', '$s$', '$tr$'
Initialize blocks generated '$BG$', sub keys '$SK_0, SK_1, \ldots, SK_u$', '$u$', round function '$RF$'
Initialize hit ratio '$HR = 0$', miss ratio '$MR = 0$'
Begin
    Foreach Dataset '$DS$' with Cloud User '$CU$', Block '$B$', Samples '$S$',Traces '$TR$' and blocks generated '$BG$'
        Partition block generated plaintext '$BGP$' into two blocks '$(BGL_0, BGR_0)$'
        For each sequence
            //Encryption
            Formulate XOR function as given in (7) and (8)
            Evaluate block generated cipher text as given in (9) and stored in cache
            If '$BGC = Cache[BGC]$'
            Then '$HR = HR + 1$'
                No attacks detected
                Obtain timestamp '$TS$'
                If timestamp '$TS > T$'
                Then timing attack detected
            Else
                Evaluate '$MR = MR + 1$'
                Cache attack detected
            End if
            Obtain timestamp '$T$'
            Return block generated cipher text '$BGC$'
        End for
        //Decryption
        For each sequence
            Formulate reversal XOR function as given in (10) and (11)
            Evaluate block generated plain text as given in (12)
            Return block generated plain text '$BGP$'
        End for
        If '$BGP \neq BGC$'
        Then data remanence attack detected
        Else
            No attack detected and secured communication between cloud users
        End if
    End for
End
```

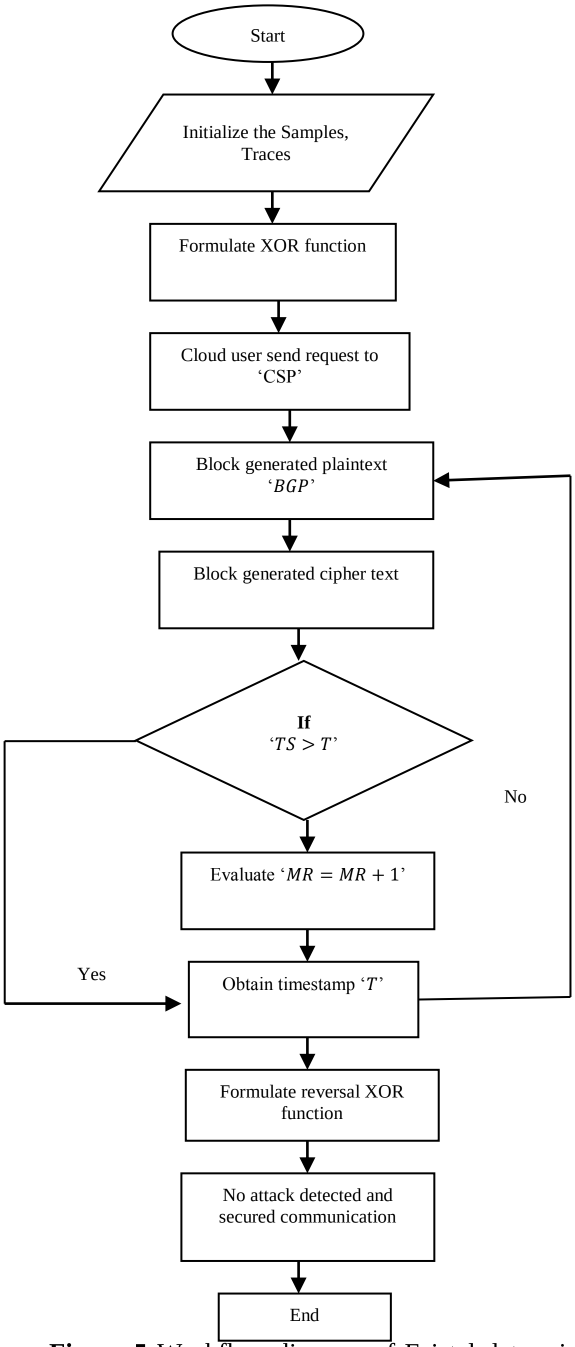
Start
Initialize the Samples, Traces
Formulate XOR function
Cloud user send request to 'CSP'
Block generated plaintext 'BGP'
Block generated cipher text
If '$TS > T$'
No
Yes
Evaluate '$MR = MR + 1$'
Obtain timestamp 'T'
Formulate reversal XOR function
No attack detected and secured communication
End

Figure 5 Workflow diagram of Feistel deterministic cipher symmetric encryption and multiclass vector-based block validation

4.Results

Experimental analysis

Experimental evaluation of FCSE-MV and conventional PSASPIN [1] and reinforcement-strategy-based side-channel attack [2] similar network traces are executed in JAVA language. SCAAML dataset was used for experimentation. For ensuring fair comparison among proposed FCSE-MV method and existing [1, 2] similar network traces are determined for evaluating various metrics like for ten dissimilar simulation runs. The experiment is conducted in hardware and software specification of Windows 10 operating system, Intel Core i5- 6200U CPU @ 2.30GHz 4 cores by 4 gigabytes of DDR4 RAM. Comparative performance analyses are done with the help of either table or graphical representation.

Communication complexity

Initial metric of importance was the communication complexity, or overhead or the complexity involved in performing three different actions to establish communication or simply generating blocks. This was due to the reason that while generating blocks i.e., associating one block with another and performing three distinct processes, proportionate memory was said to be consumed. This is defined as communication complexity and expressed as by Equation 17.

$$CC = \sum_{i=1}^{n} T_i \times Mem[Prepare + Execute + Validate] \quad (17)$$

In Equation 17, communication complexity 'CC' was calculated depend on network traces 'T_i' , memory consumed in performing three actions 'Prepare', 'Execute' and 'Validate' respectively. CC was measured in kilobytes (KB). Communication complexities obtained by various classifiers i.e., FCSE-MV, PSASPIN [1] and Reinforcement-strategy-based side-channel attack [2] depend on outcomes of the Equation 17 are given below *Figure 6*. *Figure 6* depicts communication complexity calculated of KB with respect to 8000 distinct network traces obtained at different timestamps. As depicted in the *Figure 6*, 'x' axis represents the traces and 'y' axis indicates the communication complexity. The different scalability of three FCSE-MV, [1, 2] are represented by three different colors of lines namely blue, orange and gray as shown in the above plot. From the above Figure, increasing the network traces causes a proportionate increase in the increase in the size of blocks and proportionately increasing the generated block size, therefore increasing the communication complexity also. However, simulations performed using 800 network traces involved communication overhead to be 240KB, 280KB using [1] and 400KB using [2] respectively. The simulation results of communication complexity were enhanced by using FCSE-MV method upon comparison to conventional methods. This is due to application of aggregated Byzantine fault tolerance consensus algorithm through aggregate function reduces the resources by creating subgroup and upon the occurrence of fault during block generation, with the non-operative resources becoming operative. The entire performance of ten dissimilar results denotes which CC of FCSE-MV is reduced by 14% and 34% than the [1, 2].

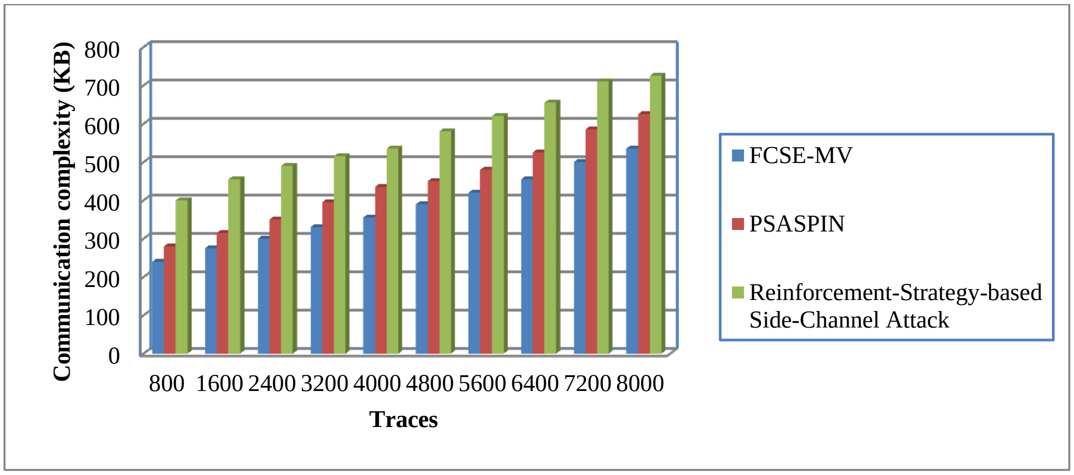

Figure 6 Communication complexity comparison across different numbers of traces for FCSE-MV, PSASPIN, and reinforcement-strategy-based side-channel attack methods

Throughput

It defined as data being broadcasted in receiver end. Higher the throughput rate, more efficient the method will be. Throughput is measured as shown in Equation 18.

$$Throughput = \sum \frac{T_{received}}{T_{sent}} \times 100 \qquad (18)$$

In Equation 18, throughput rate 'Throughput' was calculated depend on network traces sent 'T_sent' , network traces received 'T_received'. It was measured in percentage (%). *Figure 7* summarizes the throughput using three methods, FCSE-MV, [1, 2], respectively.

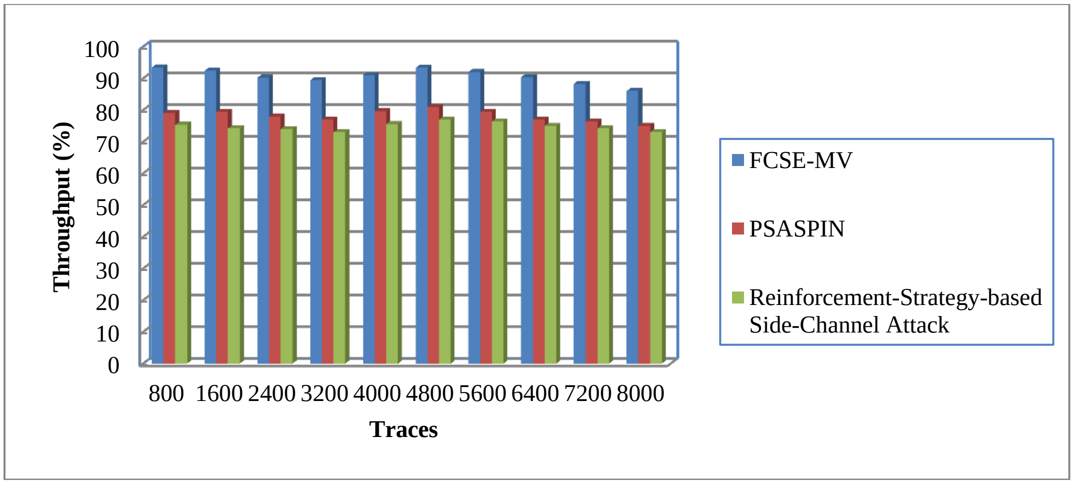

Figure 7 Throughput comparison across different numbers of traces for FCSE-MV, PSASPIN, and reinforcement-strategy-based side-channel attack methods

Figure 7 depict throughput rate in the y axis for different network traces obtained from SCAAML dataset in the x axis. The ten different results of throughput are reported with respect to different network traces considered of 800 to 8000 collected from the dataset. For distinct numbers of network traces, the throughput rate obtained is also different. However, simulations for 800 network traces found the throughput rate of 93.47% using FCSE-MV, 79% using [1] and 75.33% using [2]. With these results the throughput rate using FCSE-MV was better compared to [1, 2]. The comparative outcome is proof which FCSE-MV method achieves higher throughput when compared to conventional methods. Subsequently, nine runs are carried out and the results are compared. The average results indicate that the throughput rate is enhanced using FCSE-MV method by 16% and 21% when compared to existing [1, 2]. Enhancement of throughput rate using FCSE-MV method was owing to the application of Aggregated Byzantine Fault Tolerance Consensus algorithm. Through using this algorithm, first, blocks for corresponding network traces were generated. Also only depend on the four distinct operations, like, initialization, prepare, execute, validate/update, the corresponding blocks were generated. As a result, the block generated for the corresponding network traces were received at the other end via consensus algorithm for maximum throughput.

Attack detection accuracy

Next performance parameter of importance was attacking detection accuracy. During attack detection process not only the detection of attack has to be made but also the type of side-channel attack has to be estimated in an efficient manner. Higher the accuracy of side-channel attack, then the method is of higher efficiency. It is calculated as shown in Equation 19.

$$SCAD_{acc} = \sum_{i=1}^{n} \frac{T_{ad}}{T_i} \times 100 \qquad (19)$$

In Equation 19, side-channel attack detection accuracy 'SCAD〗_acc' is calculated based on network traces 'T_i' , network traces which has identified kind of attack precisely 'T_ad'. It is calculated in percentage (%). It taken by various classifiers i.e., FCSE-MV, PSASPIN [1] and Reinforcement-strategy-based side-channel attack [2] are given in *Figure 8.*

Figure 8 shows graphical representation of attack detection accuracy using FCSE-MV, PSASPIN [1], reinforcement-strategy-based side-channel attack [2]. In *Figure 8,* SCAD〗_acc' is neither enhancing nor reducing proportionate through network traces given as sample. The traces is taken as input to calculate the accuracy. In *Figure 8*, three different color lines designate the accuracy of three different techniques namely FCSE-MV, [1, 2]. This is due to reason which through differing network traces obtained at dissimilar timestamps sub bytes created. But, through simulations carried out by 800 network traces, 70 network traces being detected with three types of attacks, cache attack (35 network traces), timing attack (20 network traces) and data remanence (15 network traces), by applying FCSE-MV method 33 cache attacks were identified, 18 timing attacks were identified and 13 data remanences were identified. In a similar manner using [1, 2], identified cache attacks were 30, 28, timing attacks were 15, 14 and data remanence attacks were 11, 10.

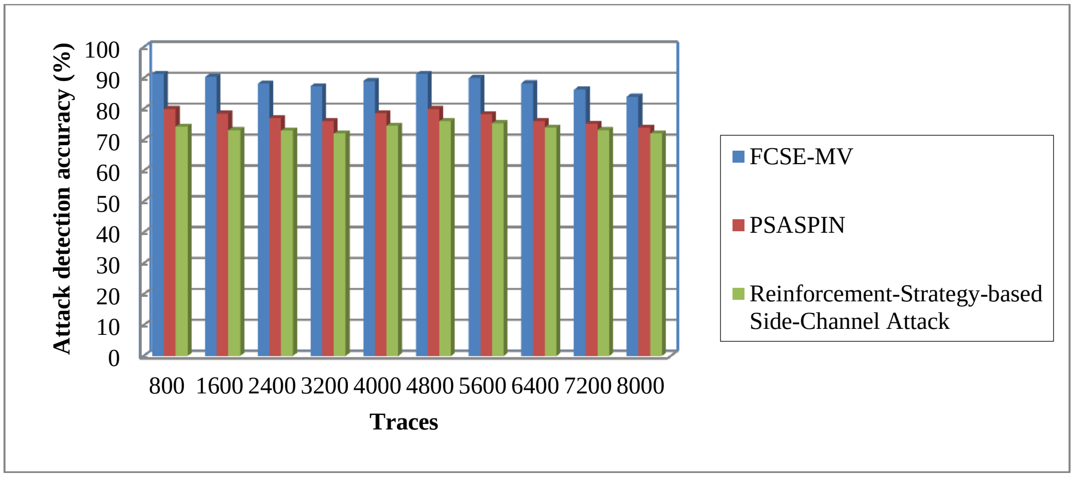

Figure 8 Attack detection accuracy comparison across different numbers of traces for FCSE-MV, PSASPIN, and reinforcement-strategy-based side-channel attack methods

Through this simulation outcomes, attack detection accuracy for 800 network traces by three techniques were 91.42%, 80% and 74.28%, so effective recognition was attained by FCSE-MV technique. This is due to application of FCSE-MV-based block validation algorithm. By using this algorithm, significant difference among data points from block generated cipher text and plain text were made by maximizing the margin with minimum distance via hyperplane separation in an accurate and robust manner. The average of ten result of side-channel attack detection accuracy using FCSE-MV method was enhanced by 15% and 20% when compared to existing [1, 2].

Attack detection time

One of important performances parameter for side-channel attack detection with blockchain in CC environment was time utilized in identifying side-channel attack. To be more precise, 〖SCAD〗_time' refers to time utilized in identifying three different types of SCA. 〖SCAD〗_time' was lower, method was more effective as previous time utilized in identifying side-channel during migration performed between CUs via VM earlier remedial actions can be taken. It is expressed as by Equation 20.

$$SCAD_{time} = \sum_{i=1}^{n} T_i \times Time[AD] \qquad (20)$$

In Equation (20), side-channel attack detection time ' 〖SCAD〗_time' was calculated on basis of network traces measured for simulation 'T_i' , time utilized in identifying side-channel attacks 'Time [AD]'. It is calculated in milliseconds (ms). Attack detection time was shown in *Figure 9*

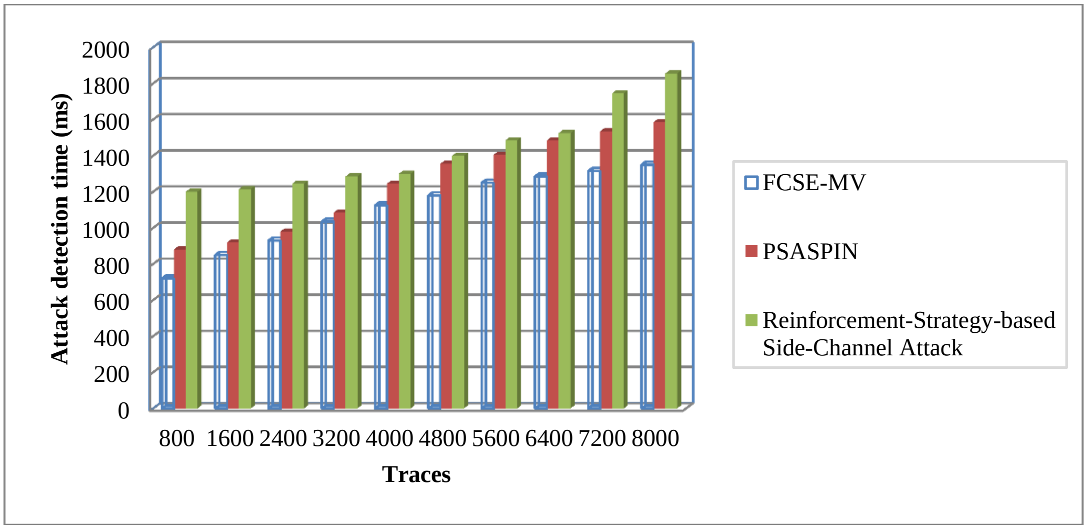

Figure 9 Attack detection time comparison across different numbers of traces for FCSE-MV, PSASPIN, and reinforcement-strategy-based side-channel attack methods

Figure 9 depicts graphical depiction of 〖SCAD〗_time by FCSE-MV, PSASPIN [1] and PSASPIN [2]. From *Figure 9*, it is deduced the 〖SCAD〗_time enhances throughput increase in network traces given during simulation process. This is due to cause which through larger number of network traces found in CC environment, increased in the amount of time is utilized through process of presence/absence of three types of attacks, increases side-channel attack detection time also. So, the side-channel attack detection time is straightly proportional to network traces. However, with simulations performed with 800 network traces, build a reliable SCA recognition scheme, the time consumed in detecting correct type of side-channel attack for a specific network trace being '0.09ms', the overall attack detection time using FCSE-MV was 720ms, the time consumed in detecting correct type of attack for a particular network trace being '0.11ms', the overall attack detection time using [1] was 880ms, time utilized in identifying accurate attack for specific network trace '0.15ms', entire attack detection time using [2] was 1200ms. From this outcome 〖SCAD〗_time in identifying three dissimilar types of attacks by FCSE-MV method is observed enhanced than the [1, 2]. Enhancement is owing to application of *FCSE-MV* -based block validation algorithm in FCSE-MV method. By applying this algorithm, with the block generated as input that in turn forms the plain text, Feistel deterministic block cipher function was applied to each block via encryption and decryption using the same keys. With the application of similar keys for both the encryption and decryption process via block cipher the overhead was said to be reduced. With the overhead being minimized, the attack involved are said to be detected at an early stage. The average of side-channel attack detection time using FCSE-MV method was reduced by 15% and 27% when compared to conventional [1, 2].

FPR

FPR measured as ratio of number of network traces which has wrongly identified dissimilar kinds of attack. It was expressed as follows.

$$FPR = \sum_{i=1}^{n} \frac{T_{ad}(wrongly)}{T_i} \times 100 \qquad (21)$$

In Equation 21, FPR denotes an FPR, $T_{ad}(wrongly)$ indicates network trace which identified kind of attack wrongly T_i represents total number of network traces. It was calculated in percentage (%). The FPR is shown in *Figure 10.*

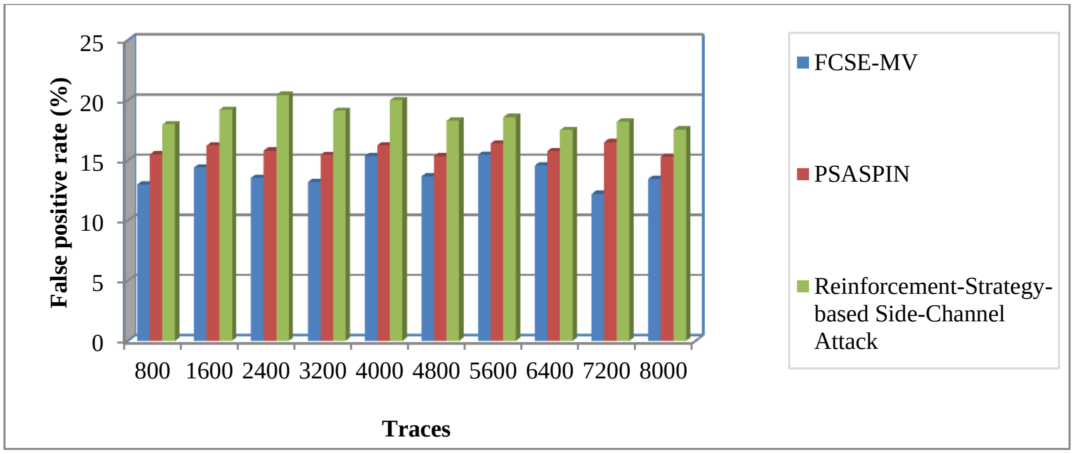

Figure 10 FPR comparison across different numbers of traces for FCSE-MV, PSASPIN, and reinforcement-strategy-based side-channel attack methods

Figure 10 given above illustrate the performance of FPR. From the graphical results, the FCSE-MV outperforms well to achieve minimum FPR than the existing results. Let us consider 800 numbers of traces, the result of FPR was found to be 13% using FCSE-MV. By applying [1, 2], performance of FPR was found to be 15.5% and 18% respectively. Likewise, dissimilar performance outcomes are examined for every technique. Entire observed results of FCSE-MV are compared with existing methods. The comparison results obviously demonstrate that performance of FPR using GPMELCB method is reduced by 12%,26% when compared to [1, 2] respectively. Cause behind enhancement was because of application of FCSE-MV-based block validation algorithm to find the minimum error of attack recognition. This helps to minimize wrong identification of SCA.

Scalability

It is capability of dissimilar methods to correctly detect diverse type of attacks for secured data transmission. It is calculated in percentage (%) (Equation 22).

$$S = \left(\frac{T_{CI}}{T_i}\right) \times 100 \qquad (22)$$

Where ‘S’ denotes scalability, T_{CI} denotes the number of traces that are correctly detected.

Figure 11 portray the experimental results of scalability. The above table depicts the scalability of the three techniques namely FCSE-MV, PSASPIN [1], and reinforcement-strategy-based side-channel attack [2]. For each method, the ten different results of scalability are obtained with respect to traces in the range of 800 to 8000. Examined outcome describes which FCSE-MV better well in terms of archiving the higher scalability. This is proved by statistical calculation by considering the 800 of software programs. With the 800 of the input traces, the scalability of the FCSE-MV is 88% and the scalability of wolf pack algorithm- particle swarm optimization (WPA-PSO) [1] and FPSONNM [2] are 87% and 80.28% respectively. Similarly, the nine various results are obtained for each method. This is cause for attaining enhanced S of FCSE-MV by using multiclass vector-based block validation algorithm. At last, the observed results of the FCSE-MV are compared to conventional methods. Average scalability outcomes of FCSE-MV are increased by 6% when compared to [1] and 13% when compared to [2].

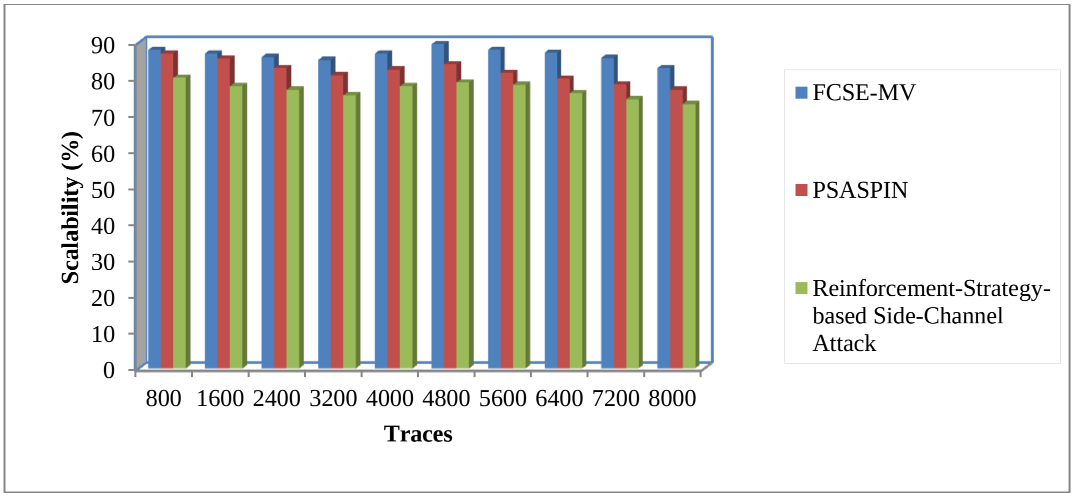

Figure 11 Scalability comparison across different numbers of traces for FCSE-MV, PSASPIN, and reinforcement-strategy-based side-channel attack methods

Ablation experiments

Ablation experiments are the procedure of removal section of input systematically which elements are inputs are similar to output. FCSE-MV includes of different types of attack results namely cache attack, timing attack, data remanence. *Figure 12* illustrates the Average accuracy metrics breakdown in terms of performance on the cache attack, timing attack, data remanence. In *Figure 12*, x axis denotes different types of attack as well as y axis gives the accuracy in percentage (%) for proposed FCSE-MV, PSASPIN [1] and reinforcement-strategy-based side-channel attack [2]. In addition, the proposed FCSE-MV is provided better performance than compared to existing methods.

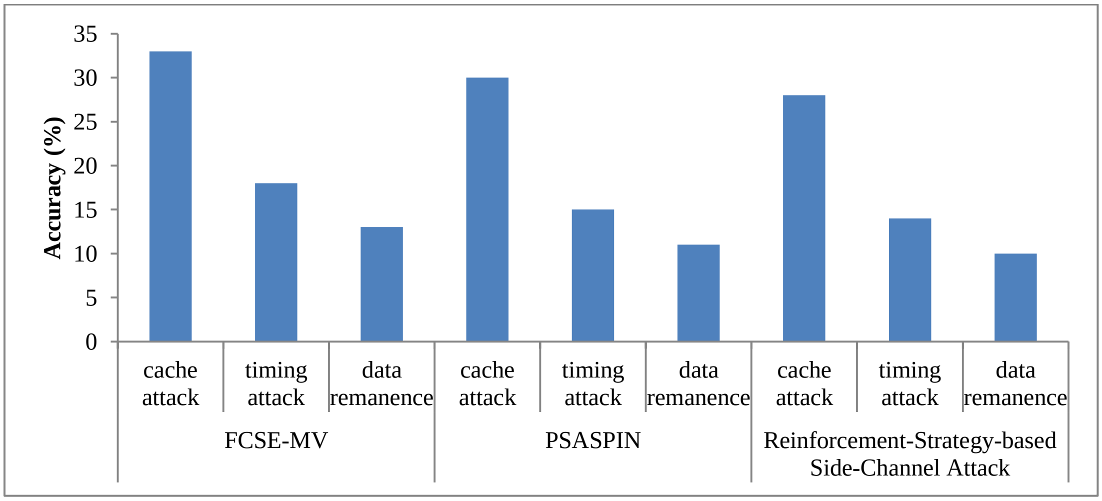

Figure 12 Average accuracy metrics breakdown in terms of performance on the cache attack, timing attack, data remanence

5.Discussion

The aim of proposed FCSE-MV side-channel attack detection method was designed to secured data transmission with blockchain in CC environment is to enhance accuracy and throughput and minimize FPR and time. The proposed FCSE-MV method implement to recovered sub-keys in JAVA programming language with Cloudsim simulator are compared by using SCAAML dataset. The limitation of existing methods was higher time, failed to consider communication complexity, throughput and accuracy. To address this issue, the proposed FCSE-MV method was implemented. The reduction in time can be attributed to the application of FCSE-MV, which was utilized for the encryption and decryption processes of every block via the block cipher. The computational complexity was minimized by proposed FCSE-MV method ensuring lesser attack detection time than conventional methods. The reason for higher accuracy and throughput is to apply multiclass vector-based block validation algorithm. It is vital difference among data points from block generated cipher text and plain text were done by maximizing the margin with fewer distance through hyperplane separation for accurate manner. From the experimental results, the following summary key finds are achieved: Proposed FCSE-MV method achieved higher throughput by 19% and accuracy by 18% when compared PSASPIN [1] and reinforcement-strategy-based side-channel attack [2]. FCSE-MV method also minimizes the communication complexity by 24%and attack detection time 21% when compared to existing methods. In comparative analysis, FCSE-MV method provides the better performance for attack detection and attains overall objectives in terms of accuracy, throughput, FPR, communication complexity and attack detection time. The limitation of FCSE-MV method is failed to analyze the multiclass attack such as differential fault analysis and allocation-based SCA. In addition, the failed to focus various parameters such as precision, recall for achieving secure data transmission. In the future, a cryptography method is recommended for classifying multiclass attacks by considering precision and recall parameters.

A complete list of abbreviations is summarized in *Appendix I.*

6.Conclusion

FCSE-MV side-channel attack detection method was proposed for secured data transmission with blockchain in CC environment. During this process, first block generation for each network trace was made by employing aggregated Byzantine fault tolerance-based block generation model. Next, only with the generated blocks, block validation was made by means of Feistel deterministic cipher symmetric encryption and multiclass vector-based block validation model. Finally, effectiveness of method was estimated through using various performance measures and the obtained results were compared with conventional methods. Therefore, on whole, our Feistel deterministic cipher symmetric encryption and multiclass vector-based block validation model technique was more beneficial than the conventional techniques in identifying SCA. As well, performance study is concerned, our SCA detection technique had shortest communication complexity and maximum throughput compared by two conventional data transmission method. In addition, FCSE-MV was identifying SCA in an accurate and timely manner. Hence, it was more suitable for secured data transmission with blockchain in CC environment.

In the future, the proposed method will be further extended to identify SCAs, including differential fault analysis and allocation-based SCAs, with enhanced security in cloud computing through the use of advanced deep learning techniques. Furthermore, the cryptography method is employed to carry out encryption and decryption processes, aiming to reduce communication overhead.

Acknowledgment

None.

Conflicts of interest

The authors have no conflicts of interest to declare.

Data availability

The SCAAML dataset used in this paper is publicly accessible at the following GitHub repository: https://github.com/google/scaaml/tree/master/scaaml_intro.

Author’s contribution statement

Ramakrishna Subbareddy: Literature review, design, data collection, implementation, analysis and interpretation of results and manuscript writing. **P. Tamil Selvan:** Literature review, design, analysis and interpretation of results and manuscript writing.

References

[1] Jimale MA, Z’aba MR, Kiah ML, Idris MY, Jamil N, Mohamad MS, et al. Parallel sponge-based authenticated encryption with side-channel protection and adversary-invisible nonces. IEEE Access. 2022; 10:50819-38.

[2] Jin S, Bettati R. Efficient side-channel attacks beyond divide-and-conquer strategy. Computer Networks. 2021; 198:108409.

[3] Chen Z, Wu A, Li Y, Xing Q, Geng S. Blockchain-enabled public key encryption with multi-keyword search in cloud computing. Security and Communication Networks. 2021; 2021:1-11.

[4] Zhang X, Su Y, Qin J. A dynamic searchable symmetric encryption scheme for multiuser with forward and backward security. Security and Communication Networks. 2020; 2020:1-13.

[5] Yan X, Yuan X, Ye Q, Tang Y. Blockchain-based searchable encryption scheme with fair payment. IEEE Access. 2020; 8:109687-706.

[6] Ma M, Shi G, Li F. Privacy-oriented blockchain-based distributed key management architecture for hierarchical access control in the IoT scenario. IEEE Access. 2019; 7:34045-59.

[7] Tran SD, Seok B, Lee C. HANMRE-an authenticated encryption secure against side-channel attacks for nonce-misuse and lightweight approaches. Applied Soft Computing. 2020; 97:106663.

[8] Amitha M, Srivenkatesh M. DDoS attack detection in cloud computing using deep learning algorithms. International Journal of Intelligent Systems and Applications in Engineering. 2023; 11(4):82-90.

[9] Salam MI, Yau WC, Chin JJ, Heng SH, Ling HC, Phan RC, et al. Implementation of searchable symmetric encryption for privacy-preserving keyword search on cloud storage. Human-centric Computing and Information Sciences. 2015; 5:1-6.

[10] Shen J, Deng X, Xu Z. Multi-security-level cloud storage system based on improved proxy re-encryption. EURASIP Journal on Wireless Communications and Networking. 2019; 2019(1):1-12.

[11] Sreelatha G, Babu AV, Midhunchakkarvarthy D. A survey on cloud attack detection using machine learning techniques. International Journal of Computer Applications. 2020; 975:21-7.

[12] Tyagi M, Manoria M, Mishra B. Survey and analysis for achieving the security of data in cloud. International Journal of Applied Engineering Research. 2019; 14(20):3954-9.

[13] Agrawal N, Tapaswi S. Defense mechanisms against DDoS attacks in a cloud computing environment: state-of-the-art and research challenges. IEEE Communications Surveys & Tutorials. 2019; 21(4):3769-95.

[14] Shang Y. Prevention and detection of DDOS attack in virtual cloud computing environment using naive bayes algorithm of machine learning. Measurement: Sensors. 2024; 31:100991.

[15] Goy G, Loiseau A, Gaborit P. A new key recovery side-channel attack on HQC with chosen ciphertext. In international conference on post-quantum cryptography 2022 (pp. 353-71). Cham: Springer International Publishing.

[16] Spreitzer R, Moonsamy V, Korak T, Mangard S. Systematic classification of side-channel attacks: a case study for mobile devices. IEEE Communications Surveys & Tutorials. 2017; 20(1):465-88.

[17] Pasha MJ, Rao KP, Mallareddy A, Bande V. LRDADF: an AI enabled framework for detecting low-rate DDoS attacks in cloud computing environments. Measurement: Sensors. 2023; 28:100828.

[18] Wang W, Du X, Shan D, Qin R, Wang N. Cloud intrusion detection method based on stacked contractive auto-encoder and support vector machine. IEEE Transactions on Cloud Computing. 2020; 10(3):1634-46.

[19] Karabulut E, Aysu A. Falcon down: breaking falcon post-quantum signature scheme through side-channel attacks. In 58th ACM/IEEE design automation conference 2021 (pp. 691-6). IEEE.

[20] Olanrewaju RF, Khan BU, Kiah ML, Abdullah NA, Goh KW. Decentralized blockchain network for resisting side-channel attacks in mobility-based IoT. Electronics. 2022; 11(23):1-22.

[21] Ramachandran D, Albathan M, Hussain A, Abbas Q. Enhancing cloud-based security: a novel approach for efficient cyber-threat detection using GSCSO-IHNN model. Systems. 2023; 11(10):1-30.

[22] Ramzan M, Shoaib M, Altaf A, Arshad S, Iqbal F, Castilla ÁK, et al. Distributed denial of service attack detection in network traffic using deep learning algorithm. Sensors. 2023; 23(20):1-24.

[23] Aliyu AA, Liu J. Blockchain-based smart farm security framework for the internet of things. Sensors. 2023; 23(18):1-13.

[24] Assiri FY, Ragab M. Optimal deep-learning-based cyberattack detection in a blockchain-assisted IoT environment. Mathematics. 2023; 11(19):1-16.

[25] Attou H, Mohy-eddine M, Guezzaz A, Benkirane S, Azrour M, Alabdultif A, et al. Towards an intelligent intrusion detection system to detect malicious activities in cloud computing. Applied Sciences. 2023; 13(17):1-19.

[26] Mostafa AM, Ezz M, Elbashir MK, Alruily M, Hamouda E, Alsarhani M, et al. Strengthening cloud security: an innovative multi-factor multi-layer authentication framework for cloud user authentication. Applied Sciences. 2023; 13(19):1-24.

[27] Singh A, Mushtaq Z, Abosaq HA, Mursal SN, Irfan M, Nowakowski G. Enhancing ransomware attack detection using transfer learning and deep learning ensemble models on cloud-encrypted data. Electronics. 2023; 12(18):1-31.

[28] Wang Y, Zheng W, Liu Z, Wang J, Shi H, Gu M, et al. A federated network intrusion detection system with multi-branch network and vertical blocking aggregation. Electronics. 2023; 12(19):1-14.

[29] Shah K, Jadav NK, Tanwar S, Singh A, Pleşcan C, Alqahtani F, et al. AI and blockchain-assisted secure data-exchange framework for smart home systems. Mathematics. 2023; 11(19):1-23.

[30] Bahaa A, Sayed A, Elfangary L, Fahmy H. A novel hybrid optimization enabled robust CNN algorithm for an IoT network intrusion detection approach. Plos one. 2022; 17(12):e0278493.

[31] Li A, Yi S. Intelligent intrusion detection method of industrial internet of things based on CNN-BiLSTM. Security and Communication Networks. 2024; 2024:1-9.

[32] Aldweesh A, Derhab A, Emam AZ. Deep learning approaches for anomaly-based intrusion detection systems: a survey, taxonomy, and open issues. Knowledge-Based Systems. 2020; 189:105124.

[33] Alkadi O, Moustafa N, Turnbull B, Choo KK. A deep blockchain framework-enabled collaborative intrusion detection for protecting IoT and cloud networks. IEEE Internet of Things Journal. 2020; 8(12):9463-72.

[34] Bagui S, Wang X, Bagui S. Machine learning based intrusion detection for IoT botnet. International Journal of Machine Learning and Computing. 2021; 11(6):399-406.

[35] Da CKA, Papa JP, Lisboa CO, Munoz R, De AVH. Internet of things: a survey on machine learning-based intrusion detection approaches. Computer Networks. 2019; 151:147-57.

[36] Atul DJ, Kamalraj R, Ramesh G, Sankaran KS, Sharma S, Khasim S. A machine learning based IoT for providing an intrusion detection system for security. Microprocess. Microsystems. 2021; 82:103741.

[37] Vargas H, Lozano-garzon C, Montoya GA, Donoso Y. Detection of security attacks in industrial IoT networks: a blockchain and machine learning approach. Electronics. 2021; 10(21):1-18.

[38] Awotunde JB, Chakraborty C, Adeniyi AE. Intrusion detection in industrial internet of things network-based on deep learning model with rule-based feature selection. Wireless Communications and Mobile Computing. 2021; 2021:1-7.

[39] Jothi B, Pushpalatha M. WILS-TRS—a novel optimized deep learning based intrusion detection framework for IoT networks. Personal and Ubiquitous Computing. 2023; 27(3):1285-301.

[40] Qureshi KN, Rana SS, Ahmed A, Jeon G. A novel and secure attacks detection framework for smart cities industrial internet of things. Sustainable Cities and Society. 2020; 61:102343.

[41] Kasongo SM, Sun Y. Performance analysis of intrusion detection systems using a feature selection method on the UNSW-NB15 dataset. Journal of Big Data. 2020; 7(1):105.

[42] Otoum Y, Liu D, Nayak A. DL-IDS: a deep learning–based intrusion detection framework for securing IoT. Transactions on Emerging Telecommunications Technologies. 2022; 33(3):e3803.

[43] Kumar V, Das AK, Sinha D. UIDS: a unified intrusion detection system for IoT environment. Evolutionary Intelligence. 2021; 14(1):47-59.

[44] Latif S, Idrees Z, Zou Z, Ahmad J. DRaNN: a deep random neural network model for intrusion detection in industrial IoT. In international conference on UK-China emerging technologies 2020 (pp. 1-4). IEEE.

[45] Long J, Liang W, Li KC, Wei Y, Marino MD. A regularized cross-layer ladder network for intrusion detection in industrial internet of things. IEEE Transactions on Industrial Informatics. 2022; 19(2):1747-55.

[46] Zolanvari M, Teixeira MA, Gupta L, Khan KM, Jain R. Machine learning-based network vulnerability analysis of industrial internet of things. IEEE Internet of Things Journal. 2019; 6(4):6822-34.

[47] Abdel-basset M, Chang V, Hawash H, Chakrabortty RK, Ryan M. Deep-IFS: intrusion detection approach for industrial internet of things traffic in fog environment. IEEE Transactions on Industrial Informatics. 2020; 17(11):7704-15.

[48] Elrawy MF, Awad AI, Hamed HF. Intrusion detection systems for IoT-based smart environments: a survey. Journal of Cloud Computing. 2018; 7(1):1-20.

[49] Mudassir M, Unal D, Hammoudeh M, Azzedin F. Detection of botnet attacks against industrial IoT systems by multilayer deep learning approaches. Wireless Communications and Mobile Computing. 2022; 2022:1-12.

[50] Vishwakarma M, Kesswani N. DIDS: a deep neural network based real-time intrusion detection system for IoT. Decision Analytics Journal. 2022; 5:100142.

[51] Tang X, Guo C, Choo KK, Liu Y, Li L. A secure and trustworthy medical record sharing scheme based on searchable encryption and blockchain. Computer Networks. 2021; 200:108540.

[52] Shukla D, Chakrabarti S, Sharma A. Blockchain-based cyber-security enhancement of cyber–physical power system through symmetric encryption mechanism. International Journal of Electrical Power & Energy Systems. 2024; 155:109631.

[53] Premkumar R, Priya SS. Service constraint NCBQ trust orient secure transmission with IoT devices for improved data security in cloud using blockchain. Measurement: Sensors. 2022; 24:100486.

[54] Prasad SN, Rekha C. Block chain based IAS protocol to enhance security and privacy in cloud computing. Measurement: Sensors. 2023; 28:100813.

[55] Ragu G, Ramamoorthy S. A blockchain-based cloud forensics architecture for privacy leakage prediction with cloud. Healthcare Analytics. 2023; 4:100220.

[56] Tu S, Yu H, Badshah A, Waqas M, Halim Z, Ahmad I. Secure internet of vehicles (IoV) with decentralized consensus blockchain mechanism. IEEE Transactions on Vehicular Technology. 2023; 72(9):11227-36.

[57] Yang X, Chen A, Wang Z, Li S. Cloud storage data access control scheme based on blockchain and attribute-based encryption. Security and Communication Networks. 2022; 1-12.

[58] Gao H, Luo S, Ma Z, Yan X, Xu Y. BFR-SE: a blockchain-based fair and reliable searchable encryption scheme for IoT with fine-grained access control in cloud environment. Wireless Communications and Mobile Computing. 2021; 2021:1-21.

[59] Yu C, Yang W, Xie F, He J. Technology and security analysis of cryptocurrency based on blockchain. Complexity. 2022:1-15.

[60] Kim J, Nakashima M, Fan W, Wuthier S, Zhou X, Kim I, et al. A machine learning approach to anomaly detection based on traffic monitoring for secure blockchain networking. IEEE Transactions on Network and Service Management. 2022; 19(3):3619-32.

Ramakrishna Subbareddy completed his masters in computer applications from Madurai Kamaraj University. He is working as a Solution Architect in DXC technology. He is also a Research Scholar in Department of Computer Science, Karpagam Academy of Higher Education, Coimbatore. His experience is 20 years and he has presented various papers in National and International Conference. His research interests lie in Cloud Computing Security.
Email: Ramki.blr@gmail.com

Dr. P. Tamil Selvan completed his Ph.D in Computer Science from Karpagam Academy of Higher Education in 2017. He is working as Associate Professor in Department of Computer Science, Karpagam Academy of Higher Education, Coimbatore. His experience is 13.5 yrs. He has presented more than 25 papers in International and national Journals and Conference. His research interests are Data mining and Networking.

Email: tamilselvancs@kahedu.edu.in

Appendix I

S. No.	Abbreviation	Description
1	ADS	Anomaly Detection System
2	AE	Authenticated Encryption
3	BDKMA	Blockchain-based Distributed Key Management Architecture
4	BPKEMS	Blockchain-Enabled Public Key Encryption Scheme With Multi-Keyword Search
5	CC	Cloud Computing
6	CPS	Cyber-Physical Systems
7	CS	Cloud Security
8	CSP	Cloud Service Provider
9	CU	Cloud User
10	CPU	Central Processing Unit
11	DBF	Deep Blockchain Framework
12	DDoS	Distributed Denial-of-Service
13	Deep-IF	DeepFloyd IF
14	DH	Diffie–Hellman
15	DL	Deep Learning
16	DRaNN	Deep Random Neural Network
17	FPR	False Positive Rate
18	FCSE-MV	Feistel Cipher Symmetric Encryption and Multiclass Vector
19	HANMRE	Hash-based Authenticated Nonce-Misuse Resistant Encryption
20	HBA-OHDBN	Honey Badger Algorithm with an Optimal Hybrid Deep Belief Network
21	HQC	Hamming Quasi-Cyclic
22	HMAC	Homomorphic Message Authenticator
23	HA-LRDD	Hybrid Approach for Low rate DDoS Detection
24	IAS	Identity Authentication System
25	IoT	Internet of Things
26	IIoT	Industrial Internet of Things
27	IoV	Internet of Vehicles
28	IVM	Input Vector Matrix
29	KB	Kilobytes
30	LSTM	Long Short-Term Memory
31	ML	Machine Learning
32	M-DSSE	Dynamic Searchable Symmetric Encryption for Multiuser
33	NBA	Naive Bayes Algorithm
34	NCBQ	Network Condition and Behavior Quality
35	PBH	Previous Block Header
36	PBK	Address of Previous Block
37	PRF	Pseudo Random Function
38	PSASPIN	Parallel Sponge-Based Authenticated Encryptionwith Side-Channel Protection and Adversary-Invisible Nonces
39	PSO	Particle Swarm Optimization
40	RBF	Radial Basis Function
41	RSA	Rivest–Shamir–Adleman
42	SCAAML	Side Channel Attacks Assisted With Machine Learning Dataset
43	SCAs	Side Channel Attacks
44	SAMs	Security Access Managers
45	SCAE	Stacked Contractive Autoencoder
46	SRVA	Secure-Ring-Verification-Based Authentication
47	SVM	Support Vector Machine
48	WPA-PSO	Wolf Pack Algorithm- Particle Swarm Optimization
49	VBSBC	Vehicle-Based Secure Blockchain Consensus
50	VM	Virtual Machine