An Architecture That Enables Cross-Chain Interoperability for Next-Gen Blockchain Systems

M. Darshan, Matthieu Amet, Gautam Srivastava, *Senior Member, IEEE*,
and Jorge Crichigno, *Member, IEEE*

***Abstract*—Blockchain technology is crucial for cutting-edge demands and aligns with the trend toward decentralized architecture. Interoperability between private and public blockchain technology can revolutionize digital record-keeping and enable automation. Traditional database systems saw major developments when application programming interfaces (APIs) and data were used across centralized entities. For blockchain technology, the natural evolution would be to facilitate communication and data exchange between private and public blockchain technologies, potentially revolutionizing digital record-keeping for future automation. Our research tests this interoperability in real-world scenarios and explores smart city elements and use-cases for application in the next generation of blockchain systems.**

***Index Terms*—Blockchain, cross-chain, interoperability, networking, smart cities.**

I. Introduction

THE INCREASING importance of data exchange in our information society cannot be overstated [1], [2]. It has become a crucial aspect of our lives and enables many new possibilities that require complete trustworthiness and flawlessness. To achieve this, we need to ensure that data exchange is secure, private, immutable, and have clear records of identity and timing of data exchange. This is where blockchain technology can be beneficial. Blockchain technology can help achieve the essential principles of immutability, integrity, authentication, and nonrepudiation that are critical for data exchange. As [3] acknowledges, this technology is widely recognized for its potential to ensure secure and reliable data exchange.

Private enterprises are moving toward blockchain technology to handle their data. Private blockchains are designed to satisfy the specific needs of an enterprise while reducing the possibility of a single point of failure. Currently, many scholars are exploring the use of a unique blockchain for smart cities. However, cities are inherently heterogeneous environments, and smart cities should be diverse. This presents a significant challenge that must be resolved to make smart cities a reality [4]. Giffinger's model of a smart city illustrates this challenge well as shown in Fig. 1.

With the current shift from the Social Web 2.0 to a decentralized Web3, which is less reliant on states and private companies and more reliant on services, it is crucial to effectively link these services to meet users' needs. Service providers, such as Uber,[1] Uber Eats,[2] Kayak,[3] and Fiverr,[4] act as intermediaries between individuals, leading to the development of third-party trust. However, blockchain technology has the potential to replace these intermediaries and return to a peer-to-peer (P2P), trustless application that connects service providers with service seekers. This would result in a more secure and transparent system that removes the need for third-party intermediaries.

In today's era of smart cities, a single blockchain may not be suitable for meeting the diverse needs of various entities, such as government, healthcare, shops, and private enterprises. Customizing blockchain technology to adapt to each entity's requirements would be challenging, and having multiple intermediaries on a single blockchain can cause network congestion, making it impractical. Therefore, this article introduces a new concept of a *MainChain* that connects existing blockchains and users in a P2P relationship. The Main blockchain ensures the immutability and confidentiality of data exchanges, enabling multiple public or private entities to interact freely and with trust. By connecting different blockchains, this *MainChain* provides a practical solution that caters to the unique needs of each entity while maintaining a secure and transparent network.

By linking blockchains to the *MainChain* network, entities can freely exchange data with other entities, all without compromising the integrity of data. For instance, if a health

Manuscript received 8 May 2023; accepted 20 May 2023. Date of publication 24 May 2023; date of current version 9 October 2023. This work was supported in part by the Natural Sciences and Engineering Research Council of Canada (NSERC) under Grant RGPIN-2020-05363, and in part by the MITACS Globalink Research Internship (GRI) Program. *(Corresponding author: Gautam Srivastava.)*

M. Darshan is with the Department of Computer Science, Lakehead University, Thunder Bay, ON P7B 5E1, Canada, and also with the Department of Computer Science and Engineering, Amrita School of Engineering, Coimbatore 641112, India (e-mail: dmanoj1@lakeheadu.ca).

Matthieu Amet is with the Department of Computer Science, Lakehead University, Thunder Bay, ON P7B 5E1, Canada, and also with the Department of Computer Science, Université de Lorraine, 54000 Nancy, France (e-mail: mjamet@lakeheadu.ca).

Gautam Srivastava is with the Department of Computer Science, Lakehead University, Thunder Bay, ON P7B 5E1, Canada, also with the Department of Math and Computer Science, Brandon University, Brandon, MB R7A 6A9, Canada, also with the Research Centre for Interneural Computing, China Medical University, Taichung 404, Taiwan, and also with the Department of Computer Science and Math, Lebanese American University, Beirut 1102, Lebanon (e-mail: srivastavag@brandonu.ca).

Jorge Crichigno is with the College of Engineering and Computing, University of South Carolina, Columbia, SC 29208 USA (e-mail: jcrichigno@cec.sc.edu).

Digital Object Identifier 10.1109/JIOT.2023.3279693

[1] https://www.uber.com
[2] https://www.ubereats.com
[3] https://www.kayak.com
[4] https://www.fiverr.com/

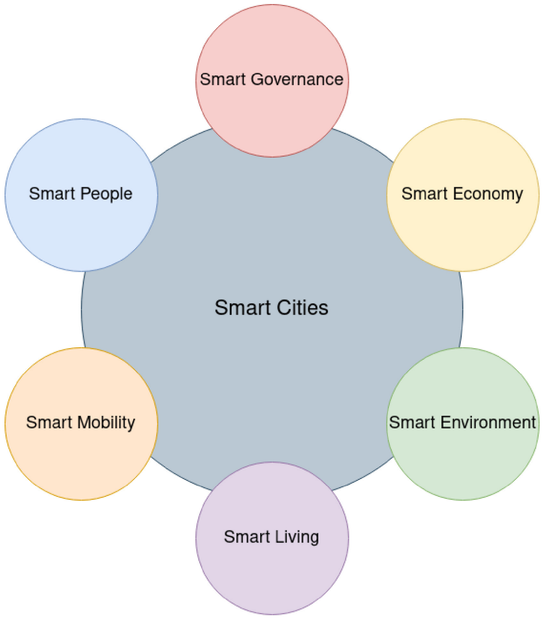


Fig. 1. Giffinger's smart city framework.

insurance company needs patient information from a hospital, it can request data through its blockchain, and the hospital can transmit the data in a secure and trustworthy way via its blockchain. The Main blockchain will also facilitate the connection between users and private entities, such as linking a potential customer with a hotel, or connecting people who require services, such as a ride or food delivery. Overall, the MainChain provides a reliable solution to meet the diverse needs of smart cities, allowing entities to interact with each other without compromising data security or causing network congestion.

This article discusses the necessity for blockchain interoperability and how it could facilitate a shift from current closed blockchain systems to an open system where users and devices can interact across blockchain boundaries. This work extends our preliminary work in [5], adding more methodological as well as experimental analysis, and other details. We summarize our main contributions of this work as follows.

1) A deeper understanding of current blockchain operations and communication channels.
2) Enhanced knowledge of blockchain technology and its ecosystem.
3) Identification and comprehension of real-world use cases for blockchain-based interoperability.
4) Introduction of a mixed node architecture for blockchain-based interoperability.
5) Successful implementation of the proposed architecture using C++.
6) Rigorous testing of the system using unbiased and advanced methods that simulate a real-world scenario.
7) Comparison of different scenarios and presentation of detailed results through visualization.

We present this article in the following sections. Section II goes over relevant literature dealing with blockchains as well as cross-chains. In Section III, we give a detailed presentation of our architecture for an interoperable smart city. In Section IV, we openly discuss our solution and its implementation. In Section V, we focus on experimental evaluation. Finally, in Section VI, we give some concluding remarks and then Section VII talks about future directions.

II. Related Works

Alasbali et al. [6] proposed a realistic infinite loop smart network concept for building a standardized, cloud-based blockchain-based intermediate for Internet of Things (IoT) networking in smart cities. The research work provides detailed insights about integrating blockchain technology at various levels of IoT sensors in order to maximize utility of the technology in a smart city environment based on cloud computing. The authors provide brief detail about the conventional network architecture and draw a comparison to their proposed architecture which consists of decentralized blockchain-based solution to integrate real time data. This research work is related to the proposed research work as the smart city environment mimics this sort of an ecosystem while the research work does not provide any cross-chain interoperability. Hence, to extend cross-chain interoperability in a smart city environment this research work is proposed.

Ou et al. [7] presented work which emphasizes the importance of cross-chain technology in creating an Internet of Blockchains (IoB) and facilitating blockchain interoperability. As the blockchain ecosystem evolves and becomes more diverse, there is a growing need for cross-chain technology to adapt and ensure high efficiency and security in cross-chain operations. However, implementing cross-chain technology poses various challenges, such as security issues and connection robustness between cross-chain networks. The article also notes that cross-chain technology is currently undergoing significant research and development, and it will require extensive exploration to overcome technological bottlenecks and create a more dynamic and promising business model. The methodology which the authors have made use of consists of a smart contract which relays the information on to another blockchain. This happens to be different from the methodology that the proposed research work uses.

Huang et al. [8] proposed a blockchain-based framework for secure cross-domain authorization and authentication (AA) for smart cities. This framework allows application service providers (ASPs) to delegate their authentication capabilities to the blockchain, which can guarantee transparent cross-domain resource access while preserving user privacy. The framework uses several privacy-preserving techniques to hide users' sensitive information on the blockchain, such as threshold-based homomorphic encryption, zero-knowledge proof, and random permutation. A cryptographic accumulator and secure hash functions are integrated to improve user revocation efficiency. A proof-of-concept prototype has been developed to demonstrate the correctness and efficiency of the proposed framework. This research work provides the basis for the security aspect of the proposed research work.

Belchior et al. [9] proposed a methodology which highlighted the significance of blockchain interoperability and the fragmented knowledge on how to achieve it. The authors carried out a literature review of 404 documents, analyzing 102 of them, and classified the studies into three categories: 1) public connectors; 2) blockchain of blockchains; and 3) hybrid connectors. They utilized the blockchain interoperability framework to categorize 67 existing solutions. This article underscores that blockchain interoperability encompasses more than cross-chain asset transfers and cryptocurrencies. The authors further deliberate on auxiliary technologies, standards, use cases, open challenges, and future research directions.

Robinson [10] explained consensus in crosschain communications which refers to how participants on one blockchain are convinced of the state of a remote blockchain, so that the information can be trusted. This research paper surveys cross-chain communication protocols and categorizes them based on their usage scenarios: value swapping, crosschain messaging, and blockchain pinning. This article analyzes how each protocol achieves crosschain consensus, their trust assumptions, and their ability to operate in different blockchain contexts, including whether they can deliver atomic updates across blockchains.

Pillai et al. [11] dove into the significance of cross-chain interoperability for blockchain technology and points out the shortcomings of the current blockchain platforms that lack the ability to interact with each other. This article puts forward a proposal to tackle this issue by suggesting a transaction-based mechanism that would allow cross-chain interoperability without any intermediaries. The proposed approach involves initiating a transaction that contains information related to asset transfer from one blockchain system to another. This transaction would then be sent to both the sending and receiving blockchain systems for validation and authentication. The objective of this mechanism is to enable the exchange of assets, data, and services among different blockchain systems without compromising the security and credibility of individual blockchains.

Li et al. [12] investigated what they coin software-defined industrial IoT (SD-IIoT) and use both blockchain and federated learning to process all types of sensitive data at the edge of networks. Their work works to protect network data against poisoning attacks and uses tentacle data exchange as a mechanism for achieving this. While their work uses federated learning mostly, their is reference to using blockchain as a means to allow data silos to be able to share information in the federated learning systems.

Yan et al. [13] implemented semi-supervised learning (SSL) for use in data tagging in the Industrial IoT. The authors propose DeNeB, which is a Deep Neural Backdoor scheme that requires less counter data poisoning efforts while still producing stronger protection against backdoor intrusions. While the authors work improves security and also the privacy of data from potential attack, although the author mention adding blockchain to their schema to allow better data source communication across locations, it was left as future work only.

Lu et al. [14] discussed about the exchange of data between different blockchains, particularly consortium blockchains. Although there are current technologies available, they do not meet the requirements for efficient two-way data interaction. The main challenge is to increase the efficiency of cross-chain exchange while ensuring the safety of data transfer. To address this issue, this article suggests using a cross-chain architecture that is based on blockchain oracle technology and a bidirectional information cross-chain interaction approach (CCIO). The CCIO approach improves traditional blockchain oracle patterns and uses a combination of symmetric and asymmetric keys to encrypt data. Experimental results show that the CCIO approach is capable of achieving secure and efficient two-way cross-chain data interactions and is suitable for large-scale consortium blockchains [14].

TABLE I
RELATED WORKS FOR BLOCKCHAIN-BASED SMART CITIES

Reference Paper	**Summary**	**Limitations**	**Advancement Provided**
Zhao *et al.* [4]	Three layered blockchains with a rating scheme for the authentication of malicious vehicles using certification authority.	No direct communication between blockchain-powered entities.	Common platform to exchange data and information. Enhanced methods for authentication.
Liu *et al.* [16]	Sharing mechanism for smart city environment and multiple cross-link chains with secure data transfer.	Not widely tested with different combinations of consensus algorithms.	The main network based on the blockchain provides better security without sharing the data with third parties.
Chang *et al.* [17]	Private blockchains communicate using smart contracts	Smart Contracts take a lot of time and computational resources. Message passing isn't efficient using smart contracts.	Data transfer is provided using blockchain (main network) and low-level systems coded from scratch for increased throughput and reduced latency.
He *et al.* [18]	3 Different chains are used for different purposes such as storing transactions and authentication.	A certification authority is used for validation and smart contracts are used to relay the information on the chain.	Locks are provided for an entire blockchain-based ecosystem.
Biswas *et al.* [19]	Safekeeping of medical records on a blockchain with interoperable advantage.	Limited to medical use cases only.	Framework with broader use cases that can help in almost all enterprise-oriented solutions.

Khor et al. [15] addressed the problem of scalability in public blockchains for IoT applications, particularly in supply chain management. The existing solutions do not consider resource-constrained IoT devices, such as RFID tags. To address this issue, this article proposes an interoperable public blockchain-based protocol for the transfer of tagged goods that is scalable and suitable for resource-constrained IoT devices. The use of a public blockchain is necessary to ensure transparent ownership data transfer, guarantee data integrity, and provide on-chain data for the protocol. The protocol is validated using a decentralized Web application that is built on the Ethereum blockchain and the InterPlanetary file system. The protocol is proven to be secure and can support secure data transfer among RFID tags while being cost effective [15].

Presently, blockchains are not just utilized as the foundational technology for digital currencies, such as bitcoin, but they are also being recognized as a potentially disruptive technology in various fields, such as healthcare and supply chain tracking (See Table I). The surge in interest in blockchains has resulted in extensive research and development endeavors. Consequently, the current blockchain landscape is highly fragmented, with various incompatible technologies available to potential users. Since current protocols and standards typically do not allow interoperability between different blockchains, functionalities, such as executing smart contracts or transferring tokens from one user to another, can only be performed within a single blockchain.

III. Proposed Architecture

In this article, our proposed solution looks to link many blockchains that work independently making use of a *MainChain* that can assist with consensus even though the entities involved are diverse. In this manner, many entities may be able to share relevant data on a blockchain maintain high levels of data security and privacy.

A. Motivation

Taking into account the heterogeneity present in smart cities, many use cases may be able to be inspired making use of several blockchain entities. The main motivation of this work can be summarized looking at the use cases that are presented in depth in this section. All use cases focus on the need for digital record keeping as well as the need for automation.

1) Healthcare and Insurance Firms: Healthcare organizations and insurance companies must securely share data and finances. Only when both entities have been validated and certified on the network is this possible. By doing this, it becomes very convenient to automate insurance reimbursement over the network.

2) Private Enterprise and Its Customers/Employees: A private company that employs a blockchain can immediately track its clients and staff. Through various networks, messages can be analyzed, and the direct business value can be increased.

3) Gig Economy: A given user may want to ask another user for a paid service, like having groceries delivered. Then, by posing as a third party, this user can send a request to the *MainChain*. This request can be accepted by another user, who will then carry out the task and receive payment. The cost of delivery is decreased for the offer maker and the payment to the delivery person is increased since the blockchain serves as the third party.

B. Network Architecture

In Fig. 2, we see three networks that are based on blockchain. First. we see that $B1$ network that requires the ability to send data to Network $B2$. The data is relayed over the primary network in the proposed solution to this problem. Fig. 3 provides a sequence diagram to help understand this facet.

Table II can be used to understand the notations used in the sequence diagram.

TABLE II
Notations Used in the Sequence Diagram

$Character$	$Description$
K_{Bob}^{+}	Public key of Bob
K_{B2}^{+}	Public key of Bob in Blockchain 2
K_{Bob}^{-}	Private key of Bob
K_{B2}^{-}	Private key of Bob in Blockchain 2
H_{Bob}^{M}	Encryption of M with the public key of Bob
S_{Bob}^{M}	Signature of M using the private key of Bob

Alice intends to send a message M to Bob, from $B1$ to $B2$, by following a secure process. First, M is encrypted with Bob's public key K^{+}Bob. Then, the resulting ciphertext is further encrypted with the public key of the destination blockchain, $K^{+}B2$. Subsequently, $H_{B2}^{H^{M}\text{Bob}}$ is added to the ledger of $B1$, with $K^{+}B2$ specified as the target blockchain.

Next, a block containing the transaction is created, added, and finalized via $X1$ consensus to the $B1$ blockchain. The mixed nodes of $B1$ then send the new transaction to the Main blockchain, which records a pending transaction and forwards it to the destination blockchain, $B2$. The mixed nodes of $B2$ receive the transaction and forward it to $B2$ nodes.

Afterward, a block containing the transaction is created, added, and finalized via $X2$ consensus to the $B2$ blockchain. This information is signed by $B2$ and forwarded to the Main blockchain. If $B1$ wishes to check the status of the transaction, it can query the network, as the data on the Main blockchain is public. Finally, $B2$ decrypts $H_{B2}^{H^{M}\text{Bob}}$ using its private key $K^{-}B2$, and Bob decrypts the message M using his private key K_{Bob}^{-}.

C. Design Goals

We have used straightforward yet advanced techniques to ensure security goals and beyond with the solution. The public keys for every blockchain are kept in the dictionary data structure. A mixed node serves as a bridge between B_i $(i = 1, 2, \ldots)$ and the *MainChain*, facilitating transactions between them. To ensure confidentiality, the solution employs RSA encryption at the node level, while the transaction itself is secured using a hashing algorithm (SHA256). Additionally, a consensus algorithm is implemented to ensure that only verified users are able to add blocks. Taking this into account, in this article we design a secure solution so that both business and users alike may make use of this methodology that possesses both reliability and security.

IV. Implementation

It has always been a challenge to be able to properly simulate an entirely new ecosystem that enables interoperability between blockchains. Despite using computing power as well as advanced machines, and tools, the exchange of data on the blockchain has never been an efficient task until recently. Our solution maintains security while achieving interoperability making use of various frameworks and tools. To alleviate concerns around some technical gaps in blockchain, we implement our solution in C++. Despite most developers using either Rust or Solidity for blockchain implementations, our solution uses C++ because of the many advantages it brings. For example, C++ can offer advanced multithreading, primitive control over memory, as well as other key object-oriented features, such as runtime polymorphism, function overloading, etc. These plus-points come in handy when implementing computing nodes because each node can be a single thread. Therefore, for the simulation

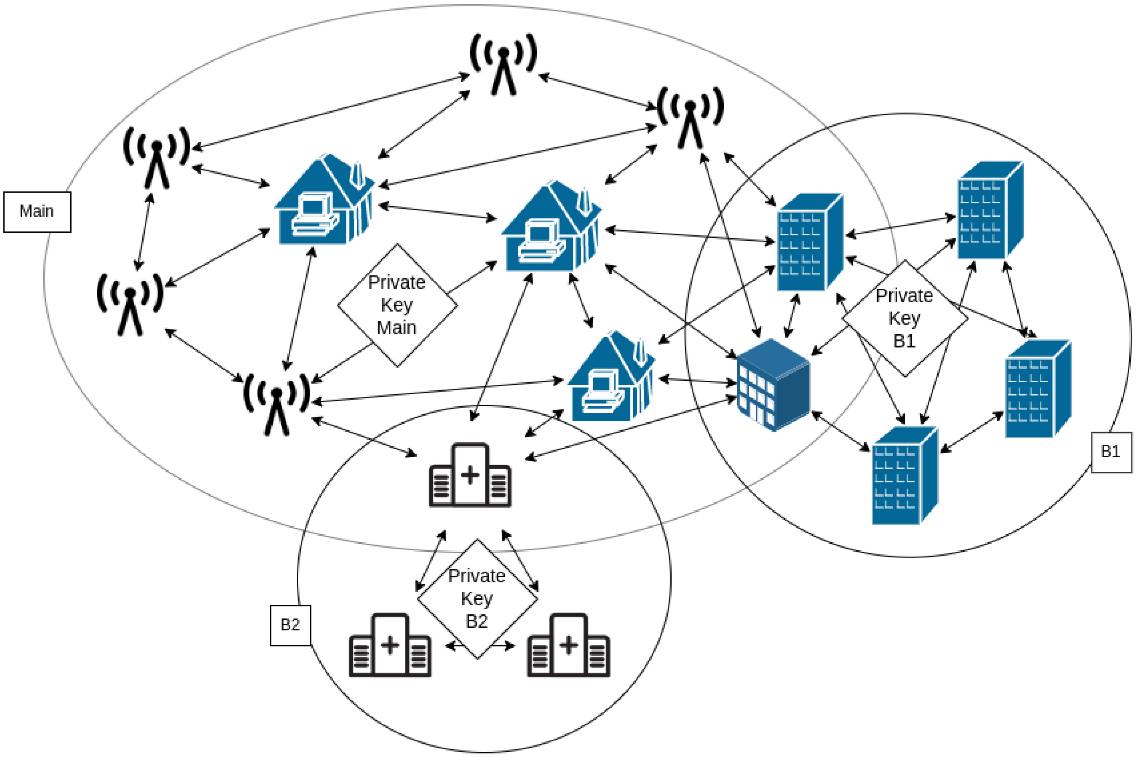


Fig. 2. Network interoperability architecture.

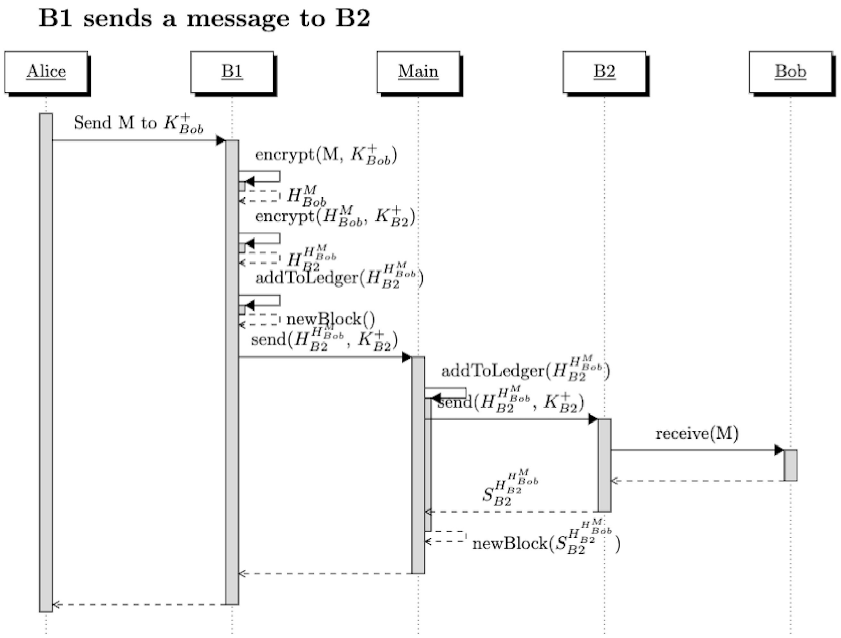

B1 sends a message to B2
Alice
B1
Main
B2
Bob
Send M to K^+_{Bob}
encrypt(M, K^+_{Bob})
H^M_{Bob}
encrypt(H^M_{Bob}, K^+_{B2})
$H_{B2}^{H^M_{Bob}}$
addToLedger($H_{B2}^{H^M_{Bob}}$)
newBlock()
send($H_{B2}^{H^M_{Bob}}$, K^+_{B2})
addToLedger($H_{B2}^{H^M_{Bob}}$)
send($H_{B2}^{H^M_{Bob}}$, K^+_{B2})
receive(M)
$S_{B2}^{H_{B2}^{H^M_{Bob}}}$
newBlock($S_{B2}^{H_{B2}^{H^M_{Bob}}}$)

Fig. 3. Sequence Diagram for interoperability.

of an entire test environment that can use multiple threads that need to be executed concurrently, C++ makes the most sense. It is also a programming language native to many Operating Systems in use today. Furthermore, for any RSA-style cryptography, the crypto++ library from C++ can be used. To ensure secure encryption, the system employs the SHA-256 function from Zedwood.[5] Critical transactions, such as message passing, are safeguarded by implementing mutex locks to prevent simultaneous access to shared resources. Semaphores are also utilized to ensure synchronization. To prevent invalid memory access, all transactions between multiple chains are recorded in a ledger. The following are the code snippets which are very important for this work. This section of this article are meant to assist any developers that wish to replicate the novel solutions proposed in this research.

[5] http://www.zedwood.com/

Algorithm 1: Algorithm to Launch a Node

```
Function launchNode is
    Generate keys for node;
    while true do
        Sign header with node's private key;
        add a block to blockchain;
    end
End
```

Algorithm 2: Algorithm to Launch a Mixed Node

```
Function launchMixedNode(RsaKeys nodeKeys) is
    bool mainNode = if a node is the main validator true else
      false;
    while true do
        Sign header with node's private key;
        add a block to the primary blockchain;
        Transactions[] transactions = retrieve transactions from
          the last block of primary blockchain;
        foreach Transaction $t_i \in$ transactions do
            if (mainNode AND $t_i$.receiver ==
              secondary.receiver) OR (!mainNode AND
              $t_i$.receiver != primary.receiver) then
                secondary.addTransaction($t_i$);
            end
        end
    end
End
```

A. Creation of a Single Node

In Algorithm 1, we create a key pair for the node. A node will try to add a block to the blockchain until there are no more transactions in the ledger. The node signs the header "name+node number" with its private RSA key created previously. Then, it will try to add a block to the blockchain using its id and the signed header.

B. Creation of Mixed Nodes

Algorithm 2 is used to launch a mixed node. We define the boolean "first" as the blockchain, the node will try to add blocks to it. We define the boolean "second" as the blockchain, the node will add "first blockchain's" transactions that are relevant to the ledger. This function is invoked twice; one where the first is the mainBlockchain and the second is the otherBlockchain, and another one where the first is the otherBlockchain and the second is the mainBlockchain.

1) As the function is called twice, we pass the same RSA keys as an argument to create the two subnodes (not created in the function for this reason).
2) The first few statements are trivial to block creation.
3) When a block of the first blockchain is added, we retrieve all of the transactions of it and we check the following.
    a) If it is the Main subnode, we check if the transaction has the same blockchain public key hash, if that is the case, we forward the transaction to the blockchain ledger as it might be the destination blockchain.
    b) If it is another blockchain subnode, we check if the transaction has a different blockchain public key hash, and if that is the case, we forward the transaction to the Main blockchain ledger for it to be forwarded to the destination blockchain (or not if the destination is the *MainChain*).

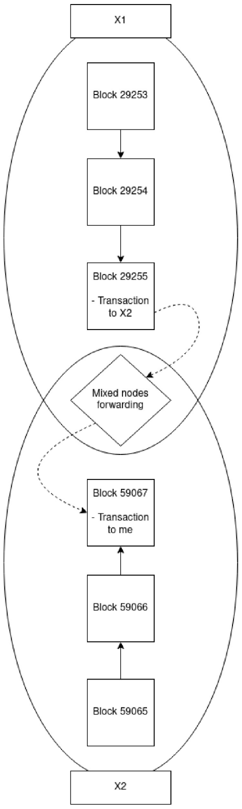


Fig. 4. Mixed nodes intersection visualization.

Algorithm 3: Algorithm to Mine a Block

```
Function mineBlock(int difficulty, int* bcSize) is
    while !_hash.beginWith('0'*difficulty) AND *bcSize ==
      _index do
        Increment _nonce
        Calculate new hash
    end
End
```

Fig. 4 explains how a transaction is forwarded to another blockchain. The mixed nodes forwarding is done by the *MainChain* to seek out mixed nodes of X2.

C. Consensus Algorithm

Algorithm 3 function implements Proof of Work (PoW) wherein it takes the difficulty (which is the number of bytes the hash must be equal to 0, for example, difficulty of three will give the hash $000f3a4b367c\ldots$ valid) and the blockchain size which is the current index of the block for the blockchain. In other words, this algorithm works like this.

1) While the hash does not begin by $\underbrace{0\ldots0}_{\times\text{difficulty}}$ and the size of the blockchain is equal to the index of the block we are looking to find a valid hash.

Algorithm 4: Algorithm to Validate a Block

```
Function stakeBlock(int difficulty, bool elected) is
    Sleep for difficulty seconds
    if elected then
        Calculate the hash of the block
    end
End
```

```
template<class T>
class MerkleTree: public Hashable {
public:
    MerkleTree();
    void buildTree(const T data[],
    const int length);
    void getLeaves(T* result);

private:
    T _data;
    MerkleTree* _left = nullptr;
    MerkleTree* _right = nullptr;

    void _calculateHash();
    void _getLeaves(T* result, int*
    counter);
    void _buildTree(const int depth,
    const T data[], const int length);
};
```

Listing 1. Merkle tree code.

2) We calculate a hash with another nonce.
3) If the blockchain size increased, it means that another node in the network found a valid hash so we stop our research (target block found).

D. Stake Block

The stake block function in Algorithm 4 shows that the threads are synchronized via semaphores, the nodes are sleeping to create a block approximately every difficulty seconds and the elected node passed as an argument, so if elected we calculate the hash of the block.

E. Merkle Tree

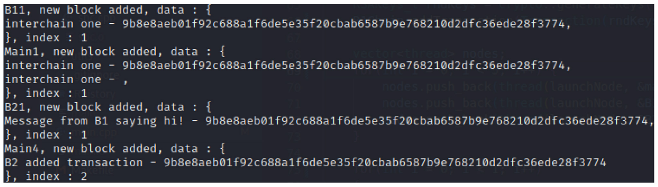
The C++ code in Listing 1 is an implementation of the header of the Merkle tree class that we used. This class inherits from the Hashable interface given in Listing 2.

With the help of the Merkle tree, we can visualize the message passing interface on the blockchain. This also ensures integrity which is one of the key pillars of security.

In Fig. 5, we see the result of an execution. The first node of $B1$ ($B1-1$) adds a new block to its chain with an interchain transaction hashed $9b8 \ldots 3774$. Then, mixed nodes of $B1$ handles the distribution of the transaction to the distributed ledger of the *MainChain*. So, the first node of the *MainChain* ($Main - 1$) receives a new information from $B1$ and adds the transaction to its next block. In the same way, mixed nodes of $B2$ see that a transaction was added in the *MainChain* so they send it to the distributed ledger of $B2$. The first node of $B2$ ($B2-1$) then receives the transaction of $B1$ and add it to the chain and. Then, the nodes of the chain can read the message that was sent saying "Message from $B1$ saying hi!" The last step is for the mixed nodes of $B2$ to tell the *MainChain* that the message was received so that $B1$ can ask it if its message was received by $B2$. We can infer that data can be exchanged between blockchains based on the proposed methodology since all steps in the process carried on without issues.

```
class Hashable{
public:
    string getHash() const;
protected:
    virtual void _calculateHash() = 0;
    string _hash;
};
```

Listing 2. Hashable interface.



```
B11, new block added, data : {
interchain one - 9b8e8aeb01f92c688a1f6de5e35f20cbab6587b9e768210d2dfc36ede28f3774,
}, index : 1
Main1, new block added, data : {
interchain one - 9b8e8aeb01f92c688a1f6de5e35f20cbab6587b9e768210d2dfc36ede28f3774,
interchain one - ,
}, index : 1
B21, new block added, data : {
Message from B1 saying hi! - 9b8e8aeb01f92c688a1f6de5e35f20cbab6587b9e768210d2dfc36ede28f3774,
}, index : 1
Main4, new block added, data : {
B2 added transaction - 9b8e8aeb01f92c688a1f6de5e35f20cbab6587b9e768210d2dfc36ede28f3774
}, index : 2
```

Fig. 5. Successful transaction of the message.

V. Experimental Evaluation

The experiments were conducted on a Linux 5.17.0 machine equipped with an Intel Core i5 9th gen and 8-GB RAM. The parameters tested included the number of transactions per block, consensus mechanism, number of nodes or mixed nodes, and the presence of cross-chain transactions. Although the difficulty was not adjusted for Proof of Stake (PoS) blockchains due to its lack of relevance, it was left unchanged for PoW blockchains because it was either too simple or too complex to find a block. The number of transactions per block varied from 512 to 8192 for single blockchains and from 512 to 2048 for cross-chain blockchains. However, the addition of more transactions resulted in a segfault of recursion for Merkle trees of that size. The consensus mechanisms tested were PoS and PoW. The number of mixed nodes was set to 10% of the total number of nodes in the blockchain, but it can be changed depending on the network's specific requirements. The difficulty was set to 5 for PoS, resulting in a 5-s delay before creating a new block. For PoW, the difficulty was set to 6, meaning that the first six bytes of the hash must be 0 to be considered valid.

A. Comprehensive Comparative Analysis

The research work also extends its solution's value by testing its capabilities in different test environments. For the

TABLE III
METRIC ANALYSIS

$Axis$	$Significance$
X	Number of Transactions per block
Y	Throughput (Transactions per second)

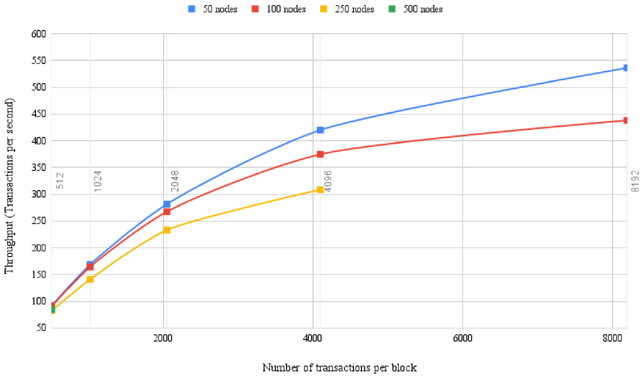


Fig. 6. Single blockchain PoS testing visualization.

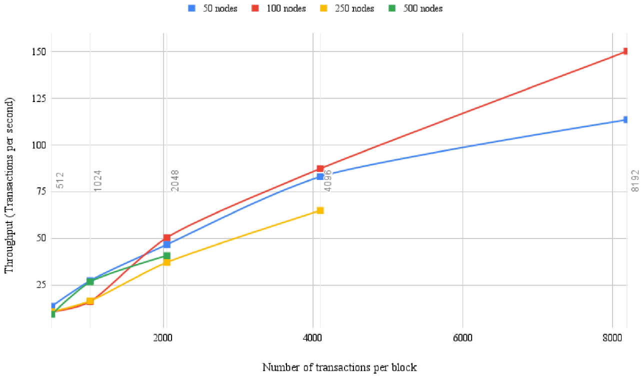


Fig. 7. Single blockchain PoW testing visualization.

following section, we have selected different combinations of consensus algorithms and based on the graphs we draw some astonishing conclusions.

1) Proof of Stake: In this test environment, a single blockchain network works on PoS as the consensus algorithm.

With the metrics of Table III in mind, the research work extends the testing to attain the following results.

The curve in Fig. 6 follows a logarithmic pattern suggesting that, as the number of nodes is increased the throughput falls considerably.

2) Proof of Work: In this test environment, a single blockchain network works on PoW as the consensus algorithm. The same metrics are used to visualize the data in Fig. 7 and the following conclusions are drawn. First, that the pattern observed can be considered as a combination of both linear and exponential curves. Second, that the pattern formed can be attributed to the ungovernable testing environment and processing capacities of the processors.

B. Metric Analysis for Combination of Consensus Algorithms

In the following section, the research work tests latency and throughput by analyzing multiple test case inputs. The testing metrics can be better understood using a sample data entry.

TABLE IV
SAMPLE OF DATA ENTRY FOR THE POS–POS MESSAGE PASSING

	5-B1	10-Main	5-B2
512 Transactions/block	5022.56	5027.76	5526.92
1024 Transactions/block	5042.16	5087.2	6129.76
2048 Transactions/block	5091.4	5202.84	7389.6

TABLE V
NUMBER OF NODES FOR B_i*Chain* AND *MainChain*

B_iChain	$MainChain$
5	10
25	50
50	100

TABLE VI
SUMMARY OF GRAPHICAL VISUALIZATIONS

Graph	**Inference**
Single Blockchain PoS	Throughput is directly proportional to the number of transactions per block until it attains a threshold after which it turns to a constant invariable of the number of nodes participating.
Single Blockchain PoW	The graph follows a linear pattern practically with just a few distortions which can be attributed to the computing hardware power of the test environment.
Cross-Chain PoS-PoS	The combination follows a linear approach with minimum variation till 1000 nodes and starts diverging after that providing the result of higher throughput with a low number of mixed nodes.
Cross-Chain PoW-PoW	The combination provides an interesting curve with FlowMin sometimes intersecting the resulting curve which depicts the non-uniform flow. The graph also provides evidence for optimizing the number of nodes to use the computing power efficiently.
Cross-Chain PoS-PoW	The combination provides fascinating results where the graph provides both PoS and PoW properties based on the sender and the receiver.
Result Visualization	The graph is obtained from the optimized results of all cross-chain combinations, and the results show that PoS-PoS cross-chain is the most efficient method and PoW-PoW cross-chain is the least efficient method. It is a data transfer method and indicates that the message is not the most efficient. blockchain. PoS-PoW cross-chains are exactly halfway between PoS-PoS and PoW-PoW cross-chains. It also represents an improved result of the mixed node algorithm.

In Table IV, we test on multiple numbers of nodes and on the number of nodes in that particular chain. For example, 5-$B1$ would signify that there are 5 computing nodes in the $B1$ chain. Tests were conducted for $512, 1024$, and 2048 transactions per block (wherever possible). The data entered and visualized are the average value from 25 test runs. The computing nodes for B_i and Main were split into the following numbers shown in Table V. A summary of graphical visualizations is given in Table VI.

1) Combination of PoS–PoS: In this test environment, both private blockchain networks work on PoS as the consensus algorithm. Our results are shown in Fig. 8 and the following conclusions are drawn. First, that FlowMin is the minimum of the throughput obtained after testing. Second, that Fig. 8 follows a similar pattern to the single PoS blockchain network since the communication is the same.

2) Combination of PoW–PoW: In this test environment, both the private blockchain networks work on PoW as the consensus algorithm. Our results are shown in Fig. 9 and the following conclusions are drawn. First, that Fig. 9 follows a similar pattern to the single PoW blockchain network since the communication is the same. Second, that the result is obtained when FlowMin is divided by 3.

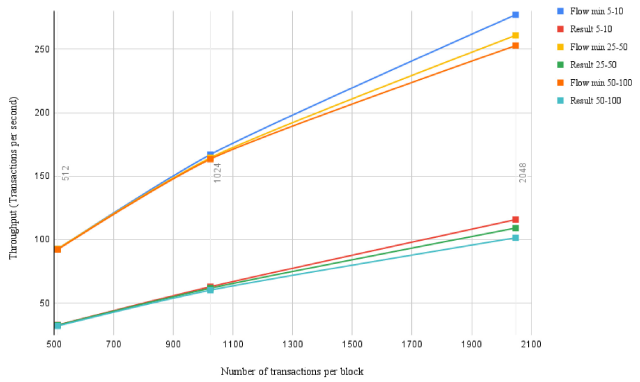


Fig. 8. Cross-chain PoS–PoS visualization.

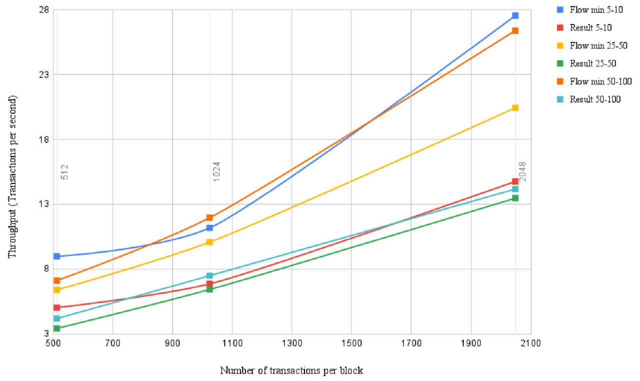


Fig. 9. Cross-chain PoW–PoW visualization.

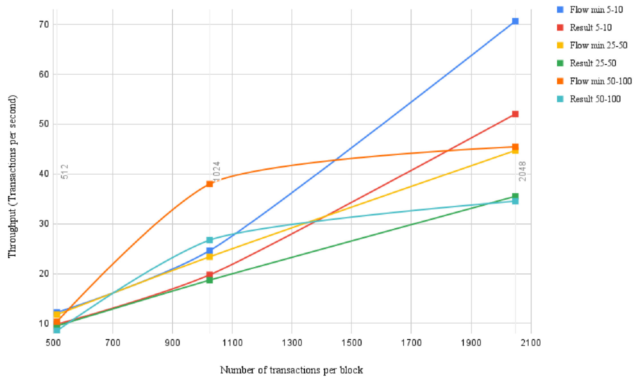


Fig. 10. Cross-chain PoS–PoW visualization.

3) Combination of PoS–PoW Visualization: In this test environment a private blockchain network works on PoW and the other on PoS as their respective consensus algorithms. Our results are shown in Fig. 10 and the following conclusions are drawn. First, that Fig. 10 follows both PoS and PoW curves depending on the consensus algorithm followed by the Main network. Second, that the curve is nothing but the result of optimized PoS and PoW together. This is shown in Fig. 11.

In Figs. 8–10, we see that the more nodes you have on the network, the more the throughput will reduce with the same number of transactions in a block. This is due to the testing environment as we are using threads acting as nodes. As there are more threads running on the testing computer than the number of cores we own, some threads will not actually work concurrently and there will be a slight delay. Though, this might simulate a real environment execution, as the more nodes on the network you have, the more you will have to wait for all the nodes to get an information.

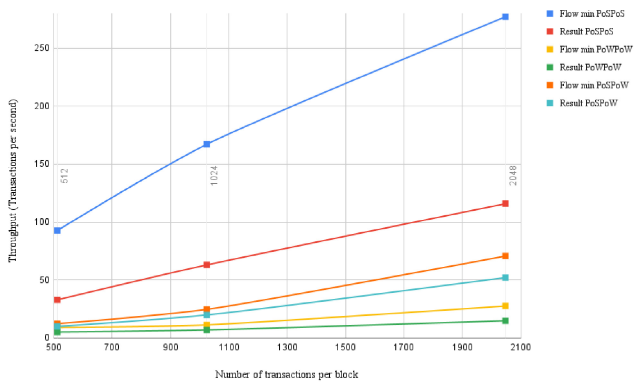


Fig. 11. Result visualization.

VI. Conclusion

Research into blockchain interoperability is still in its infancy. This study aims to deepen the understanding of interoperability and establish a foundation for any type of data exchange between different blockchain networks. Our work considers real-world scenarios where interoperability can be effectively applied and shows that interoperability is feasible without compromising data or computational power. By introducing interoperability, private entities can communicate and exchange data securely, and the research confirms that compute power is not lost in the process. The study concludes that a combination of blockchains using different consensus algorithms can perform well in practical applications, as demonstrated by extensive testing and analysis. In essence, this research provides the basis for application programming interfaces (APIs) and data exchange platforms in a decentralized and P2P blockchain environment.

VII. Future Work

The future of blockchain interoperability is very exciting. This will allow blockchain owners and maintainers to work together. The possibilities that they can potentially unlock are endless. All the methods used in this research can be further explored and extended with newer smart techniques. We hope to extend this research to be powerful enough to deploy smart contracts on one blockchain using trigger arguments from another blockchain. This allows for endless automation while data is securely stored and used.

References

[1] R. Singh, A. D. Dwivedi, G. Srivastava, P. Chatterjee, and J. C.-W. Lin, “A privacy preserving Internet of Things smart healthcare financial system,” *IEEE Internet Things J.*, early access, Jan. 2, 2023, doi: 10.1109/JIOT.2022.3233783.

[2] A. D. Dwivedi and G. Srivastava, “Security analysis of lightweight IoT encryption algorithms: SIMON and SIMECK,” *Internet Things*, vol. 21, Apr. 2023, Art. no. 100677.

[3] M. A. Cheema, M. K. Shehzad, H. K. Qureshi, S. A. Hassan, and H. Jung, "A drone-aided blockchain-based smart vehicular network," *IEEE Trans. Intell. Transp. Syst.*, vol. 22, no. 7, pp. 4160–4170, Jul. 2021.

[4] N. Zhao, H. Wu, and X. Zhao, "Consortium blockchain-based secure software defined vehicular network," *Mobile Netw. Appl.*, vol. 25, no. 1, pp. 314–327, 2020.

[5] M. Amet, D. M, G. Srivastava, and J. Crichigno, "A cross-chain interoperability architecture for smart city environments," in *Proc. IEEE Globecom Workshops (GC Wkshps)*, 2022, pp. 263–268.

[6] N. Alasbali et al., "Rules of smart IoT networks within smart cities towards blockchain standardization," *Mobile Inf. Syst.*, vol. 2022, Feb. 2022, Art. no. 9109300. [Online]. Available: https://doi.org/10.1155/2022/9109300

[7] W. Ou, S. Huang, J. Zheng, Q. Zhang, G. Zeng, and W. Han, "An overview on cross-chain: Mechanism, platforms, challenges and advances," *Comput. Netw.*, vol. 218, Dec. 2022, Art. no. 109378. [Online]. Available: https://www.sciencedirect.com/science/article/pii/S1389128622004121

[8] C. Huang et al., "Blockchain-assisted transparent cross-domain authorization and authentication for smart city," *IEEE Internet Things J.*, vol. 9, no. 18, pp. 17194–17209, Sep. 2022.

[9] R. Belchior, A. Vasconcelos, S. Guerreiro, and M. Correia, "A survey on blockchain interoperability: Past, present, and future trends," *ACM Comput. Surv.*, vol. 54, no. 8, pp. 1–41, Oct. 2021. [Online]. Available: https://doi.org/10.1145/3471140

[10] P. Robinson, "Survey of crosschain communications protocols," *Comput. Netw.*, vol. 200, Dec. 2021, Art. no. 108488. [Online]. Available: https://www.sciencedirect.com/science/article/pii/S1389128621004321

[11] B. Pillai, K. Biswas, and V. Muthukkumarasamy, "Cross-chain interoperability among blockchain-based systems using transactions," *Knowl. Eng. Rev.*, vol. 35, p. e23, Jun. 2020.

[12] G. Li, J. Wu, S. Li, W. Yang, and C. Li, "Multitentacle federated learning over software-defined industrial Internet of Things against adaptive poisoning attacks," *IEEE Trans. Ind. Informat.*, vol. 19, no. 2, pp. 1260–1269, Feb. 2023. [Online]. Available: https://doi.org/10.1109/TII.2022.3173996

[13] Z. Yan, J. Wu, G. Li, S. Li, and M. Guizani, "Deep neural backdoor in semi-supervised learning: Threats and countermeasures," *IEEE Trans. Inf. Forensics Security*, vol. 16, pp. 4827–4842, 2021.

[14] S. Lu et al., "CCIO: A cross-chain interoperability approach for consortium blockchains based on oracle," *Sensors*, vol. 23, no. 4, p. 1864, 2023. [Online]. Available: https://www.mdpi.com/1424-8220/23/4/1864

[15] J. H. Khor, M. Sidorov, and S. A. B. Zulqarnain, "Scalable lightweight protocol for interoperable public blockchain-based supply chain ownership management," *Sensors*, vol. 23, no. 7, p. 3433, 2023. [Online]. Available: https://www.mdpi.com/1424-8220/23/7/3433

[16] C. Liu, S. Guo, S. Guo, Y. Yan, X. Qiu, and S. Zhang, "LTSM: Lightweight and trusted sharing mechanism of IoT data in smart city," *IEEE Internet Things J.*, vol. 9, no. 7, pp. 5080–5093, Apr. 2022.

[17] J. Chang, J. Ni, J. Xiao, X. Dai, and H. Jin, "SynergyChain: A multichain-based data-sharing framework with hierarchical access control," *IEEE Internet Things J.*, vol. 9, no. 16, pp. 14767–14778, Aug. 2022.

[18] Y. He, C. Zhang, B. Wu, Y. Yang, K. Xiao, and H. Li, "A cross-chain trusted reputation scheme for a shared charging platform based on blockchain," *IEEE Internet Things J.*, vol. 9, no. 11, pp. 7989–8000, Jun. 2022.

[19] S. Biswas, K. Sharif, F. Li, A. K. Bairagi, Z. Latif, and S. P. Mohanty, "GlobeChain: An interoperable blockchain for global sharing of healthcare data—A COVID-19 perspective," *IEEE Consum. Electron. Mag.*, vol. 10, no. 5, pp. 64–69, Sep. 2021.

M. Darshan is curently pursuing the bachelor's degree in computer science and engineering with Amrita Vishwa Vidyapeetham, Coimbatore, India.

In his academic career, he has published a total of eight papers in high-impact conferences in the field of blockchain technology and data science. His current research interests include blockchain technology and machine learning.

Matthieu Amet is curently pursuing the master's degree in computer science with a specialization in security, networks and virtual architectures with the Université de Lorraine, Metz, France.

His current research interests include blockchain technology and cybersecurity. This research paper marks a significant milestone in his academic journey, showcasing his dedication to advancing knowledge in emerging technologies.

Gautam Srivastava (Senior Member, IEEE) received the M.Sc. and Ph.D. degrees in computer science from the University of Victoria, Victoria, BC, Canada, in 2006 and 2012, respectively.

He is a Professor of Computer Science with Brandon University, Brandon, MB, Canada. In his 10-year academic career, he has published a total of 400 papers in high-impact conferences in many countries and high-status journals (*Science Citation Index* and *Science Citation Index Expanded*).

Prof. Srivastava is an Editor of several international scientific research journals, including IEEE TRANSACTIONS ON INDUSTRIAL INFORMATICS, IEEE TRANSACTIONS ON COMPUTATIONAL SOCIAL SYSTEMS, and IEEE INTERNET OF THINGS JOURNAL.

Jorge Crichigno (Member, IEEE) received the Ph.D. degree in computer engineering from The University of New Mexico, Albuquerque, NM, USA, in 2009.

He is a Professor with the College of Engineering and Computing, University of South Carolina (USC), Columbia, SC, USA, and the Director of the Cyberinfrastructure Lab, USC. He has over 15 years of experience in the academic and industry sectors. He has authored more than 90 refereed publications. His research focuses on offloading functionality to P4 programmable data plane switches, network security, and IoT devices. His work has been funded by private industry and U.S. agencies, such as the National Science Foundation, the Department of Energy, and the Office of Naval Research.

Dr. Crichigno received two best paper awards.