December 2020

Decentralized Decision Making for Limited Resource Allocation Using a Private Blockchain Network in an IoT (Internet of Things) Environment with Conflicting Agents

Vignesh Prabhu
Clemson University, vigneshclemson@gmail.com

DECENTRALIZED DECISION MAKING FOR LIMITED RESOURCE ALLOCATION USING A PRIVATE BLOCKCHAIN NETWORK IN AN IoT (INTERNET OF THINGS) ENVIRONMENT WITH CONFLICTING AGENTS

A Thesis
Presented to
the Graduate School of
Clemson University

In Partial Fulfillment
of the Requirements for the Degree
Master of Science
Mechanical Engineering

by
Vignesh Prabhu

December 2020

Accepted by:
Dr. Lonny Thompson, Committee Chair
Dr. Sandip Dutta
Dr. Ardalan Vahidi

ABSTRACT

Blockchains have gotten popular in recent times, owing to the security, anonymity, and lack of any third-party involvement. Blockchains essentially are record keeping tools that record any transactions between involved parties. One of the key aspects of handling and navigating of any autonomous traffic on the streets, is secured and simple means of communication. This thesis explores distribution of minimum resources between multiple autonomous agents, by settling conflicts using events of random nature. The thesis focusses on two specific events, tossing of a coin and the game of rock, paper, and scissors (RPS). An improvement on the traditional game of RPS is further suggested, called rock, paper, scissors, and hammer (RPSH). And then seamless communication interface to enable secure interaction is setup using blockchains with smart contracts. A new method of information exchange called Sealed Envelope Exchange is proposed to eliminate any involvement of third-party agents in the monitoring of conflict resolution. A scenario of assigning the sole remaining parking spot in a filled parking space, between two vehicles is simulated and then the conflict is resolved in a fair manner without involving a third-party agent. This is achieved by playing a fair game of RPSH by using blockchains and simulating cross chain interaction to ensure that any messages and transactions during the game are secured.

DEDICATION

I would like to dedicate this thesis to my family for their undying love and support. I would also like to thank my friends, especially Raghavendra Rao and Vinay Gawande for their constant support and encouragement. It kept me motivated to complete my thesis.

ACKNOWLEDGMENTS

I would like to express my deepest appreciation for Dr. Sandip Dutta for his guidance, patience, and persistence. Without his immensely invaluable advice and guidance, this would not have been possible.

I would like to thank my advisors Dr. Lonny Thompson and Dr. Ardalan Vahidi for their help and support. Their motivation was instrumental in the completion of this thesis.

Additionally, I would like to thank Levi Barnes, software engineer at NVIDIA, for introducing me to the topic of blockchains and helping me get a better understanding of its concepts and algorithm implementation in Python.

TABLE OF CONTENTS

Page

LIST OF FIGURES

Figure Page

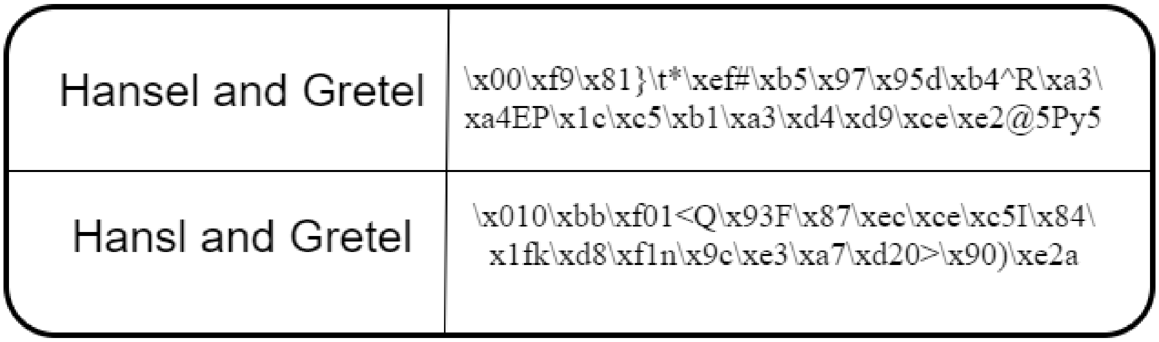
CHAPTER ONE

BLOCKCHAINS

Blockchains were first introduced, in 2008, by Satoshi Nakamato in his paper introducing bitcoins [1]. The advent of blockchain has been compared to the rise of internet and it is being referred to as the next big technological innovation. It is a secured digital ledger for economic book-keeping. It can securely record any information that can be expressed as code [2]. Blockchains enables decentralized exchange of information using secured keys, that allows authorization of transactions, by all involved parties. Blockchain has found its use in various applications from finance, real estate to healthcare. Operations that earlier needed a third party to regulate and secure transactions are not needed anymore.

The blocks serve as information while the chain serves as the database [3]. This database is usually a public ledger that is updated as new blocks are added to it and the chain grows. This database can also be a private ledger, thereby making the blockchain private. Such a blockchain has limited access and only few people or groups can make transactions (create blocks) on the chain.

Blocks in a blockchain serve as a form of database. These can be used to store any form of data. Ideally, they are used to save information about successful transactions. Once a transaction is successfully verified, the blocks record the information, and a new block is formed. The first block or the building block of the chain is called the genesis block. Along with the string stored, the blocks also store the hash of the previous block and their own hash.



Hansel and Gretel	\x00\xf9\x81}\t*\xef#\xb5\x97\x95d\xb4^R\xa3\xa4EP\x1c\xc5\xb1\xa3\xd4\xd9\xce\xe2@5Py5
Hansl and Gretel	\x010\xbb\xf01<Q\x93F\x87\xec\xce\xc5I\x84\x1fk\xd8\xf1n\x9c\xe3\xa7\xd20>\x90)\xe2a

Figure 1.1: Hash Encryption

“\x00\xf9\x81}\t*\xef#\xb5\x97\x95d\xb4^R\xa3\xa4EP\x1c\xc5\xb1\xa3\xd4\xd9\xce\xe2@5Py5” This string of various hexadecimal characters is just the encrypted version of the string “Hansel and Gretel”, while “\x010\xbb\xf01<Q\x93F\x87\xec\xce\xc5I\x84\x1fk\xd8\xf1n\x9c\xe3\xa7\xd20>\x90)\xe2a” is the encrypted string for “Hansl and Gretel”. A hash is the encrypted version of a string as shown in Figure 1.1. A hash function converts a given string into a series of hexadecimal characters. This helps make the chain tamper proof. There are certain requirements for the selection of a hash function.

1. The hash function should convert strings of any given length into an encrypted string of fixed length.
2. No two strings should ever result in the same output.

3. There should be no way of converting an encrypted string back into the original string.
4. A small change in the input string should result in a major difference in the output string [4].

Even though no two strings should ever result in the same output. There are infinite possibilities for the input string but only a finite number of possibilities for the output string, owing to the fixed size limitation. In rare cases, this can result in two strings having the same output. This is known as collision. The probability of this can be reduced by choosing a good hash function [4].

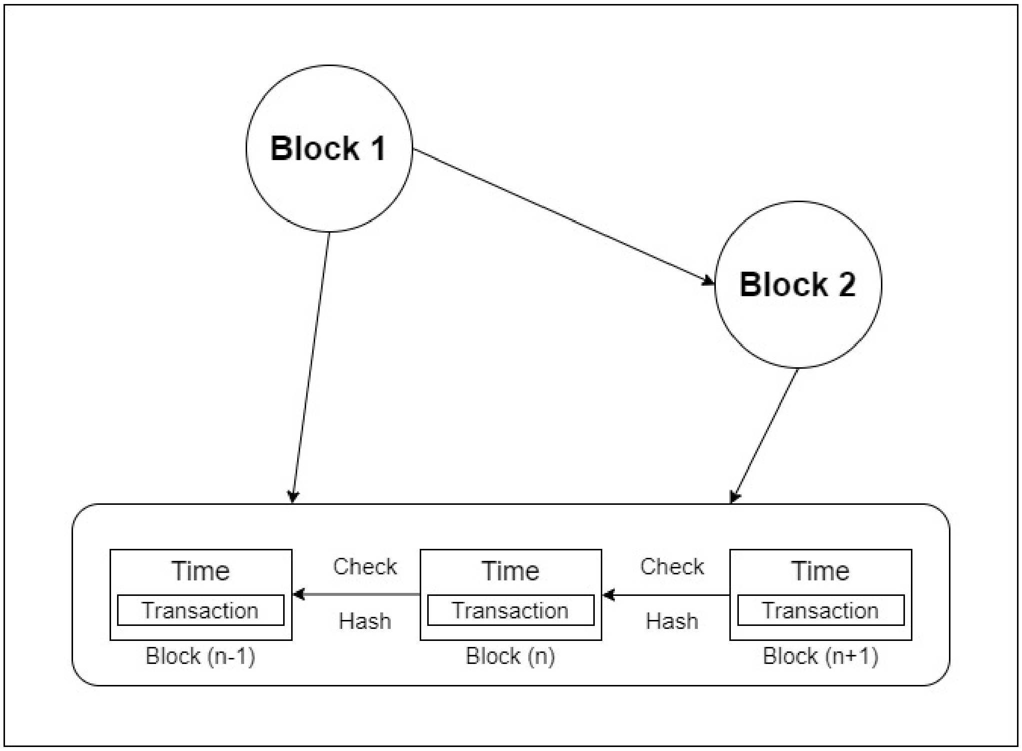


Figure 1.2: Hash [5]

The encryption ability of a hash function serves a very important application in securing block chains. When any transaction is carried out and the information is stored inside a block. The block generates a hash from this data. Since, the hash of a string cannot be the same as another string. Anytime an attacker attempts to modify data in a block, the hash of the block changes. As shown in figure 1.2, every block records the time, transaction, and the hash of the previous block along with its own. So, if the attacker were to make any changes to block n, it would change the hash of the block, so when the next block attempts to verify the authenticity of the transaction by comparing the hash of the previous block with the modified hash, it results in an error. This makes the chain tamper proof.

Hash functions are Trapdoor One-Way functions (TOWF),

$$f: X \rightarrow Y, with\ f(x) = y \text{ where,}$$

f is a unidirectional function. So, it is easy to compute $f(x) = y$, but nearly impossible to obtain $x = f^{-1}(y)$ for given $y \in Y$. So, a hash function, h can be described as [6],

$$h: M \rightarrow \{0,1\}^1, \text{ with } h(m) = \widehat{m}$$

Owing to the numerous advantages that blockchains have to offer, they have been used for several different applications. The most popular being cryptocurrencies. But apart from those, they are also used for record keeping, building smart contracts, monitoring supply chain, data backup, transportation networks and IOT.

1.1 Blockchain Network

A blockchain network is a set of agents/nodes working on the same blockchain, where each node holds copies of the chain. All these nodes form a peer to peer network. Each node is assigned a private key and a public key. The sender authenticates the transaction using their private key and their peers can access this using the public key. An invalid transaction is discarded. Valid transactions are authenticated using and added to the chain. Each block on the chain is verified by the hash of the previous block.

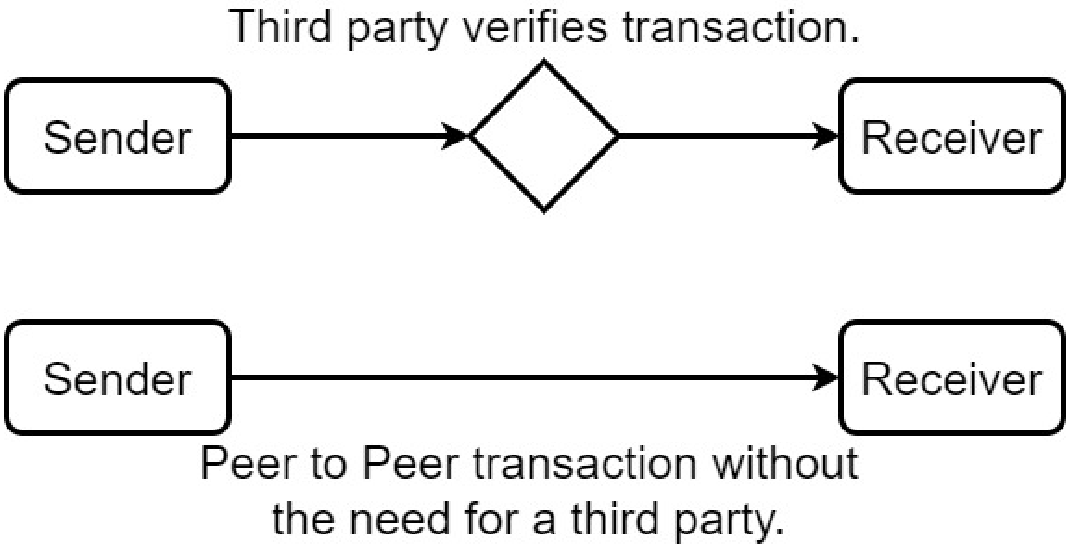

Third party verifies transaction.
Sender
Receiver
Sender
Receiver
Peer to Peer transaction without the need for a third party.

Figure 1.3: Peer to Peer Transaction

1.2 Proof of Work

Let us say Mack, went to a coffee shop and paid for his coffee with bitcoins. Mack returns home accesses the chain, removes the transaction, and later shares this modified chain with his peers. This would redeposit all the coins back in his wallet. This is called double spending, wherein one of the users modify previous transactions or delete them and add them back to the chain. This way a user can use the same coins multiple times. This makes it extremely important that all the nodes share the same copy of the chain [7]. Just as it is important for an orchestra to work together, to produce a beautiful melody, it is important for all the nodes to be in sync.

To avoid this all the nodes creating blocks are required to provide a cryptographic proof. There are several ways of doing this, but the most popular method is by adding a nonce. A nonce is a small integer value, that along with other hash parameters reduces the hash value than the current target value [7]. The resulting hash value starts with a certain number of zeroes and there is no way to predict what the correct value will be. The nodes try running different computations on their validated blocks. This method utilizes the computing power as a factor for each node, to assess validity of the blocks. This can increase the block interval for a chain. The block interval is the frequency with which content is written to a blockchain. A difficult computation will increase the block interval, which will consequently result in fewer blocks being written on the chain and vice versa [7].

1.3 Smart Contracts

Any business of conventional commerce involves signing contracts. Assume that there are two parties A and B. A signs the contract first and sends it to B for being signed.

A is at a disadvantage here since there is no guarantee that B might be fair. The only way to make the process fair is to either have them sign the contract face to face or do it remotely but in the presence of a fair third party [8]. Such a situation demands that smart contracts be fairer and more convenient.

Kim et al. [8] discuss the two different approaches on solving this issue. The first one is using procedural programming. The contract formed lays out all the tasks in order and the protocols. It becomes a unit of the organization that governs the contact and the all the tasks are performed. The second one is that the contract itself becomes the contractor. This can be enforced either by any of the parties involved in the transaction or a fairer third party. This party is assumed to be efficient and impartial. These contracts can lay out the obligations and how they are to be enforced without following up. They do not track the progress, rather they just ensure that all requirements are met.

1.4 Private Blockchains

Public blockchains are not suitable for some commercial applications, owing to the universal public access to the chain. Another type of blockchain, with a closed environment that only a few known nodes can access, can help solve the problem. This is known as a private blockchain. Consensus algorithms and smart contracts help make private blockchains more efficient than public blockchain [9]. Private blockchains are faster, safer, and more efficient as compared to public blockchains.

1.5 Cross Chain Communication

Over the years several different blockchains have been developed. All of these work independently, without any cooperative operation. There is a lack of communication among these chains, which hinders any development that could result with resource sharing within multiple different chains [10]. Within the current ecosystem, new blockchains are created to meet any emerging needs. This results in fragments of contained spaces which can hinder the growth of the industries deploying these chains. Another disadvantage is that with each chain, industries might need to make certain trade off, compromises with security to leverage higher efficiency or vice versa [11]. This can create certain limitations for individual chains. Essentially it is highly impractical for blockchain to serve as a 'one size fits all' solution, where a single chain supports all transactions and performs all desired operations globally [12].

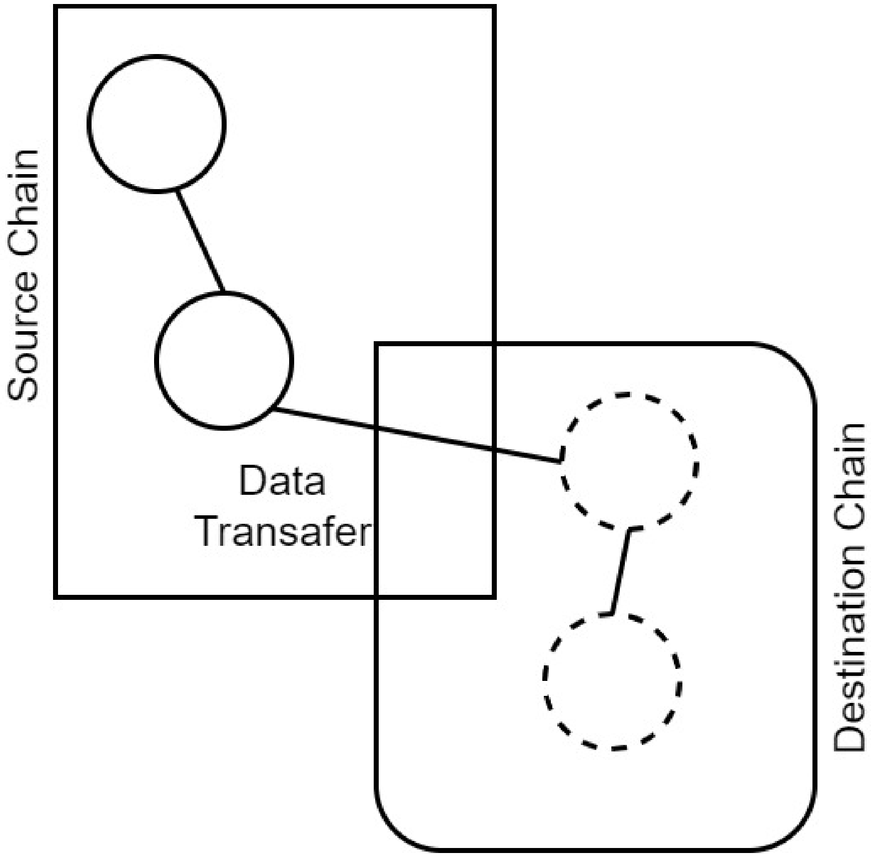


Figure 1.4 Cross Chain Communication [13]

As shown in Figure 1.4 Cross chain communication is the means for interoperability between independent chains. Chains can interact with one another by means of resource exchange between independent and unrelated blockchains. This can Cross chain communication facilitates direct transfer between different blockchains. This is especially useful when dealing with multiple private blockchains, as it provides a means of transfer between willing agents with different ledgers. It helps link independent chains [13].

Cross chain communication has several advantages to offer. There is no need for a trusted third party or escrow. Verified nodes can transfer assets over secured channels. Cryptocurrencies can be exchanged freely. Chains can access common or other chain's database. This can help in developing smart contracts or trigger any other transfer [10]. Headers and events from multiple chains can be verified and consensus algorithms can be developed for multi-chain operations. Chains compromising on certain features can leverage them from other chains thereby supporting cross chain development [11].

There are several different methods of implementing cross chain communication. Even though such a type of interaction can be tricky with complex protocols and the technology is still under development, there are ways this can be implemented. One of the easiest ways is to have a trusted third party like an escrow to facilitate all asset transfers. The party can be one that both the users on different chains trust. The intermediary can verify all transactions and activities before the final transfer is initialized. This method is more centralized in nature and compromises on any advantages, decentralization might offer to its users. Using sidechains that support multiple blockchains but are not dependent on any of the main chain can help avoid centralization. Any assets to be transferred can be held in the side chain until the transaction is completed. Another method is using hash locking. Hash locking uses a particular hash value. Both parties need to verify the hash value in stipulated time, as defined by the protocols to complete a transfer successfully. This method increases security but compromises on transaction throughput by increasing required computation.

This has limited applications. A more feasible method that the thesis focusses on is called the distributed private key control. This method uses a distributed private key exchange. Assets are locked on transfer and can only be accessed by private keys specific to all involved parties. This method helps maintain decentralization along with reducing computations. This also does not require any kind of modifications to the chain and works irrespective of the nature of the blockchains involved [13].

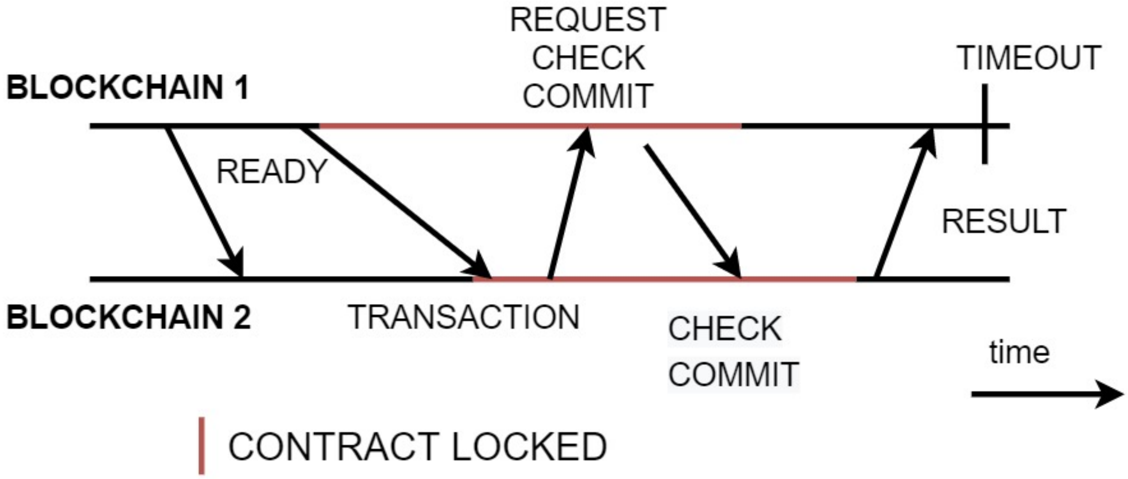


Figure 1.5 Smart Contracts in Cross Chain Communication

Smart Contracts are an essential part of any blockchain peer to peer network. They help ensure all transactions are regulated as per requirements. Figure 1.5 illustrates the implementation of smart contracts in blockchain communication. The first blockchain initiates the transaction, but the coins are not yet released to the client (Blockchain 2). As discussed earlier, smart contracts act as an overseer to make sure all tasks are accomplished between the involved parties. Once this is verified, only then are the funds released. These also make sure that the tasks are accomplished successfully in the stipulated time, or else the transaction fails.

To quote Aleks Larsen, a senior associate of the firm Blockchain capital, "*Strong interoperability could shield users from the trade-offs that blockchains make and allow them to interchangeably leverage the optimizations of different consensus mechanisms and virtual machines, in many cases off-loading tasks that are better accomplished on other chains and letting each focus on its core competencies. If we end up in a world of many blockchains, interoperability can make them more useful, user-friendly, efficient and scalable.*" [14].

CHAPTER TWO

IoT and Blockchain

IoT stands for Internet of Things, it is a rapidly emerging technology that plays a very important role in connecting multiple devices so they can interact, be queried, and controlled directly by its users. It connects physical devices so they can be operated remotely. It is a very innovative way for humans, organizations, and other entities to interact with each other seamlessly. IoT is dependent on various platforms and devices, that process a lot of information to keep the network running. A critical requirement of the IoT is that the devices must be connected to the internet and be interconnected to each other [15]. This can put a toll on cost, energy, and device lifetime. IoT applications are used in fields ranging from healthcare, education, and resource management, for example, using RFID tags for inventory management [16] or using a network connecting the alarm system to the coffee machine [17].

IoT networks generate a large amount of data. All the devices collect data from their surroundings and use it to plan their tasks. As a result, devices record every bit of information regarding person's habits, schedule, and preferences etc. Current IoT models work on a centralized client-server model, this leaves the network highly susceptible to data theft.   With growing IoT technology it is important to give users complete authority of their data [15].

Autonomous vehicles use networks to interact with each other. Internetwork communication between fleets of cars can help coordinate, while data collected during operation supports further AI development. These vehicles require extensive data to operate effectively. Sensors like cameras, GPS, wheel encoders and LiDar are used for 3D mapping of the environment [18]. Thus, IoT is also a big part of autonomous vehicles, that keep it connected for smooth operation.

IoT introduces a wide range of risks and security threats. An extended attack on even a single node can compromise the complete system. Many nodes are deployed for specific function such as cloud storage, computational capacity or saving energy. These nodes sometimes do not possess enough capabilities to protect themselves from an outside attack. Centralized service providers can further steal data from all the interconnected devices [19]. Current security protocols rely on brute force measures for cutting any access, an outside attacker might have once the system is infected. Such methods are hardly an option in mission critical systems such as manufacturing, delivery, health and especially V2V (Vehicle to Vehicle communication) [20].

Blockchains have the privacy feature built into their design. A decentralized approach can help users gain complete authority over their data. A peer to peer network is a more ideal choice in securing users data and providing them with more freedom. Furthermore, a peer to peer consensus algorithm can support securing any exchange between the interconnected devices on the network, with a transaction log monitoring each device’s activity. Ali et. al. (2017) [19] described this environment as a ‘trust-less’ one, which has more accountability built into it.

Blockchains can be used to maintain an immutable data access and operations log for all IoT operations. Each device can be assigned a private blockchain or a side chain to a root blockchain, with each chain being responsible for maintaining records for all sensor feeds and data exchange. The root blockchain forms a larger, decentralized private network. Only authorized devices are allowed access to this network and the root chain is responsible for logging any access requests for the user's data. This can further create an economy of personal data, where willing users can choose to sell their data to any interested parties [19].

Blockchains bring decentralized computing and storage, along with authentication, digital presence, and access control for users both in the public and business domain, that can help develop current IoT technology [21]. Smart contracts on blockchains can enable supporting more secure and flexible information exchange protocols. Smart contracts support interoperability of heterogeneous IoT smart contracts. Message exchange is treated similar to financial transactions and smart contracts can be leveraged to tracing any transaction between the interconnected devices and between the user and the devices. This can enable autonomous operation between all devices, while securing the user against data theft. Communication with devices or networks outside the network, i.e. with other chains or smart contracts can be blocked off. A smart contract is like a script with a unique address. The script is a reference to transactions that trigger data exchanges within the devices, that are part of the network. Since, all devices own the same copy of the contract, tampering is virtually impossible [22].

Various IoT security measures that can help secure the data channels and keep any unwanted visitors out, have been tested over time. Encrypting the data to avoid any tampering can help maintain confidentiality. Encryption is performed at the sender end while it is decrypted when received by the desired receiver. Only authorized users should be allowed access to the data. It is important that surrounding unauthorized nodes do not access any of the sensitive data. Receiver should authenticate correct receipt of package throughout the network. Certification helps confirm the identity of both entities that are communicating with each other. Access control can help block illegal access by any person, object, or machine. Certificate technology with identity identification can ensure no leak [17].

Current security measures for IoT make the model highly centralized. Using blockchain technology can help move the model towards a more decentralized approach. Using blockchain as a main archive helps instantly track and verify data. Since, the system is regularly updated with the latest block there is no scope for a point of failure. Once created, blockchains are immutable and incorruptible. Any attempted modification is flagged which nullifies any data tampering attacks [20]. Transactions are executed and stored via census algorithms. However, since public blockchains lack scalability that can lead to problems with an IoT infrastructure. Jiang et. al. (2019) [23] talk about a multi chain model that implements cross chain protocols with smart contracts and consensus algorithm for efficient and secure IoT data management. The model is a decentralized access model which optimizes device management and converts data into cash flow.

Access control protocols are put in place for privacy protection. They experiment proved that such a model was more suitable for managing IoT devices. It proved to be more efficient than a general blockchain structure.

Data gathered by IoT devices usually contain a lot of private information about the users and various interactions between the user and the system. Most IoT systems are highly centralized and have single points of failure that can hinder scalability. Existing methods of securing these systems and providing privacy protection are under third party entities. These entities also control all or most data storage servers and can adopt surveillance systems to misuse any of the personal data. Moreover, the network architecture faces a lot of threats. The entire network can be paralyzed, or a DOS attack can result in a massive failure. Users barely have any control over their data. Also, there is no accountability by third parties on network failures or data thefts. Lack of traceability follows, and users must place their blind faith in the system. As more and more devices are added centralized servers will lose their efficiency and it will become tougher to grow the IoT network, as these servers won't be able to handle the increasing amount of end to end communication between all these devices [24].

The shift towards integrating blockchain can improve the network making it stronger against any attacks. It will remove all single points of failures. New servers will not require third party entities to handle the network traffic anymore. Malicious software updates can be avoided and all the users in the network can verify transfer of data. All end to end communications can be authenticated and traceability can be maintained since all transactions are recorded onto a public ledger that is shared among all the users. This is further validated by the users, resulting in a tamper free environment. Users can monetize services provided and all transactions can be easily secured [24].

Cybersecurity is one of the most challenging barriers for IoT development. IoT devices are physical entities that are isolated and are subject to tampering. Also, these are connected to other devices and therefore it becomes difficult to protect individual devices and also secure all end to end communications. Also, IoT devices have limited computational power, so it is nearly impossible to deploy any sophisticated security protocols. Such a network with several interconnected devices, that are constantly communicating and sending messages throughout the entire network, over the internet, is highly susceptible to attacks. To secure such a network, two important steps that need to be taken are that, countermeasures need to be setup that are designed specific to each device and more computational power needs to be reserved for security purposes [24].

Data is collected from IoT users without any explicit consent. This deprives users of any control over when and where their data is shared or sold. The concern with IoT networks is not just data that but also data handling. Along with data collection another aspect is data storage, the volume of data collected every second from all these devices demands massive cloud storage. Without IoT platforms that can guarantee both scalability and privacy protection into their design, it is nearly impossible to develop these networks. Blockchain networks store a large amount of data over blockchain peers, which can offer data integrity to the users and a resilient network [24].

With growing networks and more complex security protocols and massive data storage support, there is also a growing concern of increasing maintenance costs. Cloud based infrastructure not only adds to the high maintenance costs but also communication costs. But blockchains use dedicated servers that are significantly cheaper. Moreover, since users can be monetized for using their computational and storage abilities, these prove to be a big step in handing control to the users themselves. Users can further gain cryptocurrency in exchange for their personal data [24].

Once a smart contract is executed, no intermediary can stop it from running or block it. Transactions play an important role in supporting these operations. If a code is sent, an input parameter can be set depending on the end point. For, a null account the transaction can trigger setting up a new smart contract, that will then be converted for interchain operability and executed. This can help when there are many devices or as newer devices are added to the network and thus blockchain can support smoother scalability and secure addition of new devices [25].

There are numerous advantages of such a model, and it can help grow several industries. Insurance companies for example can utilize this to calculate premiums and keep records of all their clients more secure. It can also help establish a network that can support payment from the clients directly to the insurance provider and aid in rendering services from the providers directly to the clients. The model can rid the agents of any third- party services that might be required. Smart contracts can help keep all processes compliant to legal standards and clients can be protected from any fraud. Companies can migrate to using cryptocurrencies [24]. Any industry requiring trading via third party service providers can rely on this model for ubiquitous service provision. Client confidentiality can be maintained along with assurance of secure payments and no data thefts. Companies can have an oversight over all trades via the ledger. A greater advantage would be to the supply chain industry. With unique identifiers, warehouses and distributors can have a wider control over all decisions regarding material distribution. Real time authentication of requests and subsequent provisions of service can be assured. Multi resource handling, of a variety of products with several clients can be easily handled.

Not all devices can engage in the network and keep records or support all of the operations on the network due to resource constraints. So, all devices need to be assigned specific duties. So, some devices can be assigned to keep contributing to the blockchain while others can support block validation and support decentralized consensus. Figure 3.1 highlights the different ways devices can assist in supporting the blockchain.

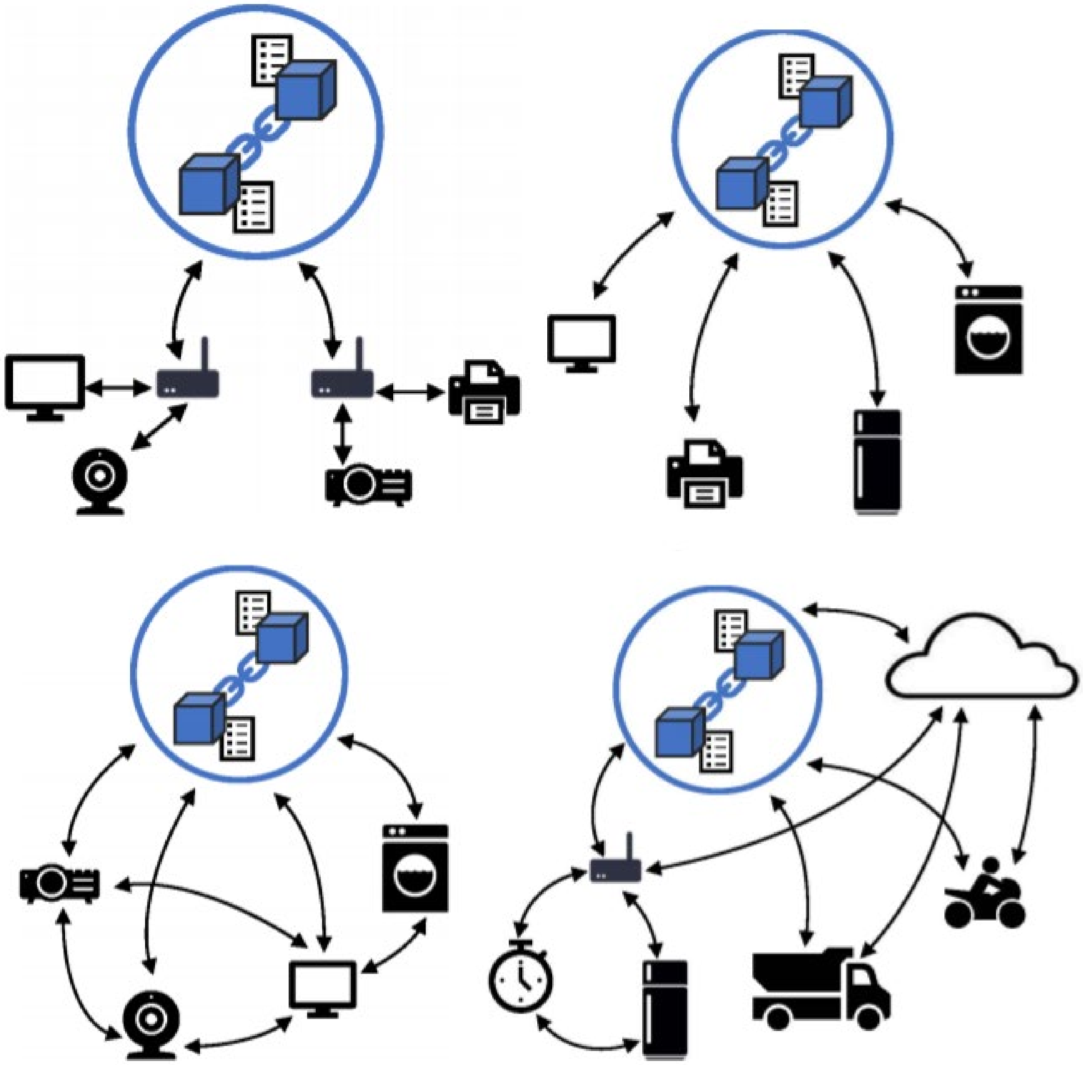


Figure 2.1: Integration of IoT devices and Blockchain [24]

Devices can act as gateways, while blockchain supports all the communication between them. Devices act as end points. This is shown in the top left diagram in figure 2.1. This makes the devices accountable for all the traffic and not all communication needs to be stored. Smart contracts in place can assist in supporting transactions and securing all communication. Such an infrastructure requires more computational power. But the network is not purely decentralized as all communication initiated by the device, first passes through the blockchain. In the second approach, as shown in the top right diagram in figure 2.1, the devices issue transactions to the chain. All the devices support encryption and the blockchain simply supports the devices in their operations. The architecture can only be supported by complex devices and will require increased computation for every device involved. For events where devices are already secure and only require the blockchain for record keeping, the system as illustrated in the bottom right diagram of figure 2.1 is more suited. All the devices can initiate transactions and can interact without using the chain. The devices choose what data needs to be recorded and accordingly communicate with the chain. Such an architecture is more desired with devices that communicate constantly and are reliable. The architecture illustrated in the bottom right diagram of figure 2.1 is more of an extension to the previous architecture. Apart from just controlling the communication between devices, here the devices also control any or all transactions initiated and only some of the transactions are passed through the blockchain [24].

Consider a case where devices fail during operation in a network. Blockchains via smart contract can support failure monitoring of devices. This ensures devices are performing their assigned tasks. This can also be used for data recovery from failed devices or to setup alerts. As more devices are added to the network, the number of nodes increases, and it becomes easier to find devices that can support any failed nodes. Such a node can support scaling and it can improve as more devices are added to the system [26].

Cryptography can ensure data security and privacy protection in an IoT environment. There are several methods of achieving this. Almost all of them not only protect the user's data but also their identities. This helps provide a completely secure framework. Data collection is based on the hash function used. Time taken for encryption is also determined by the size of the group. With a large number of devices encryption time increases [27].

CHAPTER THREE

Efficient Resource Allocation

Sometimes multiple operations are needed to be carried out using limited resources. These operations can be seen in day to day life. The problem can be described as Distribution of indivisible finite objects among a finite number of agents or processes. This can be seen in examples like distribution of labor, project management, budget allocation or even wireless network handling. In all such cases, enhancing the performance of the system is highly desired but the complexity involve with optimally allocating resources can get really challenging [28].

Deterministic algorithms, linear programming, dynamic programming and, heuristic algorithms have been used in the past to solve such problems. But in many cases, especially with non-deterministic problems or cases where not much problem specific information is available. Also, cases where given activities are indivisible. In all such events the computation becomes incredibly tough. A new approach to solving such problems as described by Ergin (2002) [29], is a priority model. Ergin talks about resource allocation based on priority. A finite number of objects are to be distributed and the agents have a fixed preference and every agent receives at least one object. A pareto distribution is used to develop the model and assignments are distributed based on their superiority.

Solving problems based on priority can make the computation simple but it only works if the structure is acyclical. Once objects become scarce, the model becomes restrictive and less efficient. Also, the model assumes that all the agents receive at least one object each. In scenarios with n+1 agents and only n objects, where nϵN, the pareto efficiency for the model drops. I have focused on solving this problem with a much simpler and realistic approach.

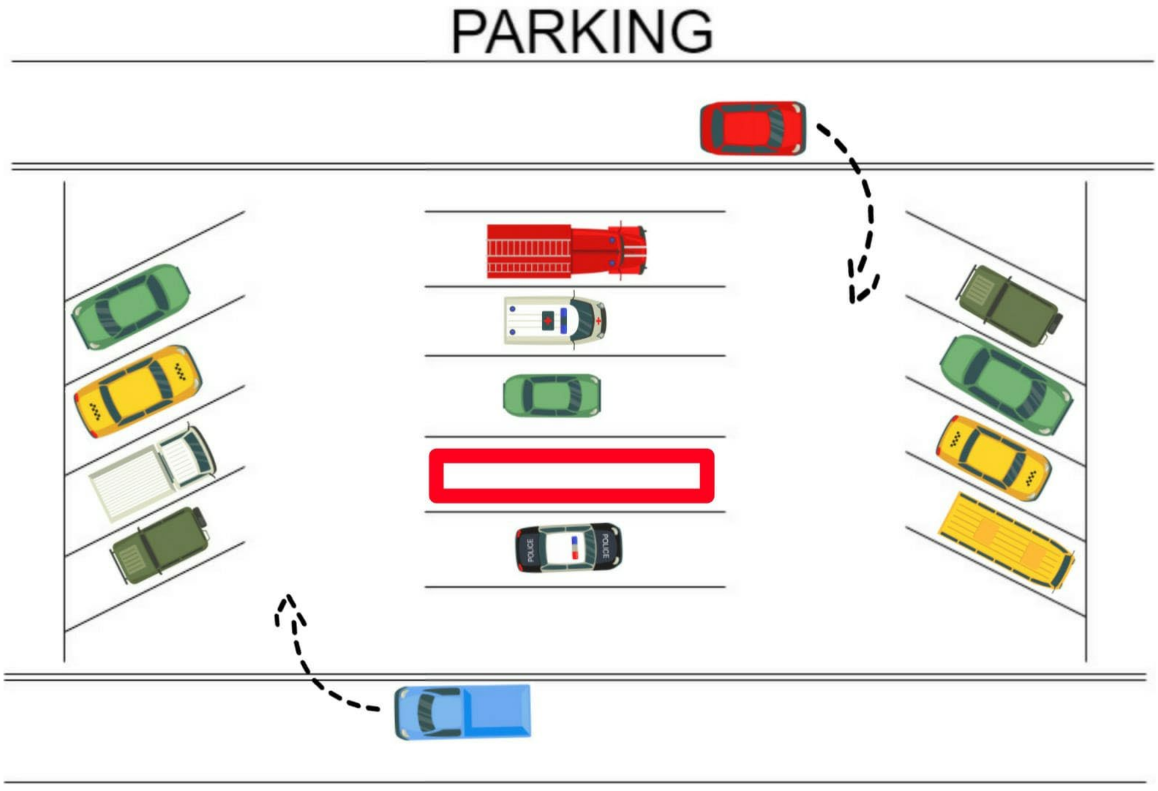


Figure 3.1: Parking Spot Allocation

Imagine the scenario illustrated in figure 3.1 with two autonomous agents. Both need access to a parking spot. Both agents are close to a parking lot with two entry and exit points. But there is only one spot remaining. The agents now must mutually come to a unanimous and fair agreement on who gets the spot. Both the agents have an equal right of way and are equidistant from the parking spot. There is no further information available about the two agents. We do not know their place of origin, their destination, or their fuel status. A unanimous decision is important to arrive at the required outcome. The two agents share no performance objectives other than the desire to park at the said spot. Assuming both agents are fair, deciding here is particularly tough because both agent's preferences are categorically opposed. There is no mutually advantageous outcome to this situation [30]. The error costs associated with the given situation are dependent on whether the agents come to a unanimous decision. If the agents base their own decisions on individual benefits, they can end up crashing into each other. This creates a conflict between their self and group interests [31]. Arriving at a decision can help them proceed as a cohesive entity in a dynamic environment.

3.1 Coin Toss

With time, there have been several optimization algorithms that have been developed for solving complex problems. All these algorithms consider numerous factors that may directly or indirectly affect the parties involved. Each of these have their own advantages and disadvantages. At the end, all of the algorithms, with their diverse approach share a common goal of achieving an optimized or fair decision. Decision making is a complex topic, even when concerned with everyday life. Even though there are optimal ways of choosing among alternatives, in carefully controlled well-structured settings. People do not rely on principles of optimal performance. It is easier to use heuristic methods over algorithmic strategies, even though these strategies generate deviations from optimal judgements [32].

Accumulating information to choose between several options is desirable, but it is not always feasible. The process itself can be tedious and time consuming. Sometimes, it can also complicate a situation and result in the inability to respond effectively to a situation due to an over analytical approach or an excess of available information. This is known as analysis paralysis. In such situations, a random decision-making alternative can help decide. Such an alternative does not require any arguments or information. Such methods help make quick, straight forward, and fair decisions.

A coin toss is an event of tossing a fair coin in the air to determine which side it lands on. A coin toss gives both sides an equal chance, it is efficient and quick. This method does not have any effect on the outcome as well. Moreover, no actions taken by the involved parties can influence the final decision as well. This method makes it easier to decide when none of the options to a situation have an advantage on the others and are weighed equally [33]. A random method of decision making can also ensure choosing between options with equal merit.

Even though events or practices that are random in nature can be of advantage in certain decision-making situations, not all of them are feasible or can be applied in every situation. For example, an event like a coin toss requires a third party to ensure that the whole process of flipping the coin is fair. The involved parties further need to agree on definitions of a fair coin and a fair method of flipping. For situations without a mediator or a third party, it can get difficult to keep the complete process fair and structured. So, such situations require tools that carry the same advantages of a random event, but also need to be simple enough to be carried out in the presence of only those parties that are affected by the outcome and deter such parties from trying to use unfair means to influence the outcome to their advantage.

3.2 Rock, Paper and Scissors

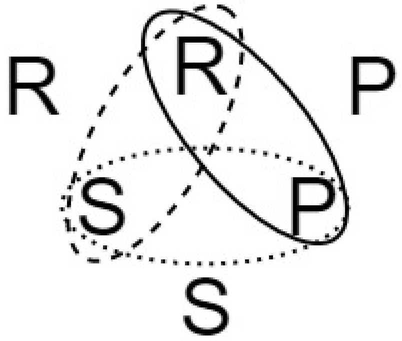


Figure 3.2: Rock, Paper and, Scissors

Rock Paper Scissors (RPS) is a fundamental non-cooperative game. It is a basic model that can help the decision-making process in non-deterministic and complex environments. The game has three candidates, Rock (R), Paper (P) and Scissors (S). Each player must make either an R, P or S gesture. The gestures are ideally to be made simultaneously, to avoid any unfair advantage that the player with a late response might gain from knowing the opponents move. The rules to the game, as illustrated in figure 3.2 are simple, R beats S, S beats P and P beats R. Choosing the same gesture i.e. R vs R, P vs P or S vs S results in a draw. Thus, the game can have a total of two outcomes, a player wins while the other loses or both draw [34]. Since, deviating from once strategy has little or no effect to the outcome, assuming the other player does not deviate, game theory predicts that the choices by either players will be completely randomized and there is no chance of exploitation, also referred to as the Nash equilibrium strategy [35]. Irrespective of what one player picks, there will always be some strategy that guarantees that the opponent wins.

There have been several instances in history where RPS has supported scenarios which required a fair decision-making model. It helped settle disputes and make fair and unanimous decisions. The game does not require the involvement of a third party and that also helps keep the process limited to the invested agents. In 2006, a federal judge ordered the defendant and the opposing counsel to settle a trivial debate over a deposition, by playing a game of RPS. In 2005, the CEO of a Japanese company decided to auction of his assets worth $12-16 million. The decision as to which firm would be responsible for carrying out the auction, was decided by a game of RPS between two firms, he believed to be good but was unable to pick one from the other. Several more such events through time have proved to be RPS a practical approach to settling disputes [36].

Several variations to RPS have been developed over the years. Each of these make the process more random, by adding in more candidates and thereby lowering the probability of selection for each. This makes the game more challenging. Two popular variations are, Rock, Paper, Scissors, Fire, Water (RPSFW) and Rock, Paper, Scissors, Lizard, Spock (RPSLS).

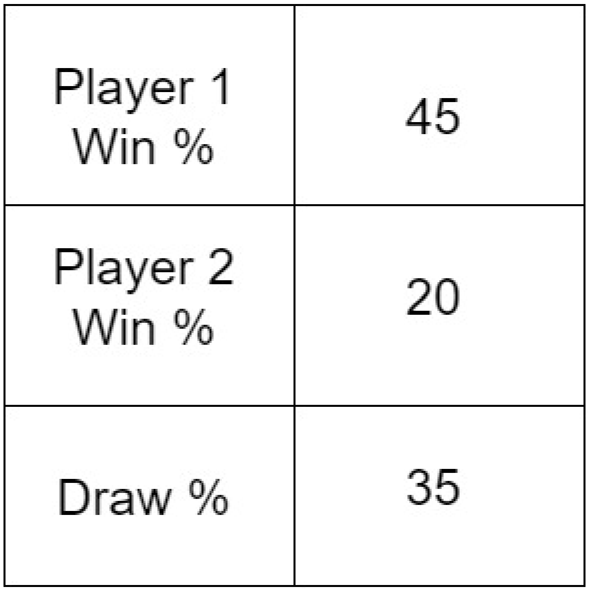
The rules for the RPSFW combination are,

- Paper beats Rock
- Scissors beats Paper
- Water beats Fire
- Water loses to everything other than fire and
- Fire wins against everything but water.

The rules for the RPSLS combination are,

- Spock beats Scissors and Rock
- Lizard and paper beat Spock
- Rock and Scissors beat lizard. [36]

Each of these renditions look simple and promising, one disadvantage that they all share is the possibility of drawing. The possibility of drawing between the agents, makes the situation complicated and time consuming. Thereby making all of these unreliable. Running a simulation where two players played a fair game of RPS 500 times. The results of the simulation, as shown in Figure 2.3, were that player 1 won 45% of the games, while player 2 won 20% of the games. Unfortunately, both of the players tied for about 35% of the games or 175 games.



Player 1 Win %	45
Player 2 Win %	20
Draw %	35

Figure 3.3: Simulation Results

3.3 Rock, Paper, Scissors and Hammer

Owing to the issue of the possibility of the game drawing and the transaction coming to a halt, a new rendition of the game Rock, Paper and Scissors, called Rock, Paper, Scissors and Hammer (RPSH), as shown in figure 3.4 is proposed [37]. The game has only two options, 1 and 0, to pick from which makes it easier for the players. The player that picks the first option is called the regular player and all the choices for the player are referred to as regular choices i.e. regular 1 and regular 0. While every choice that the opponent picks are labeled as alternate choices i.e. alternat 1 and alternate 0. The rules are simple,

- Alternate 0 beats regular 1
- Regular 0 beats alternate 0
- Alternate 1 beats regular 0

- Regular 1 beats Alternate 1

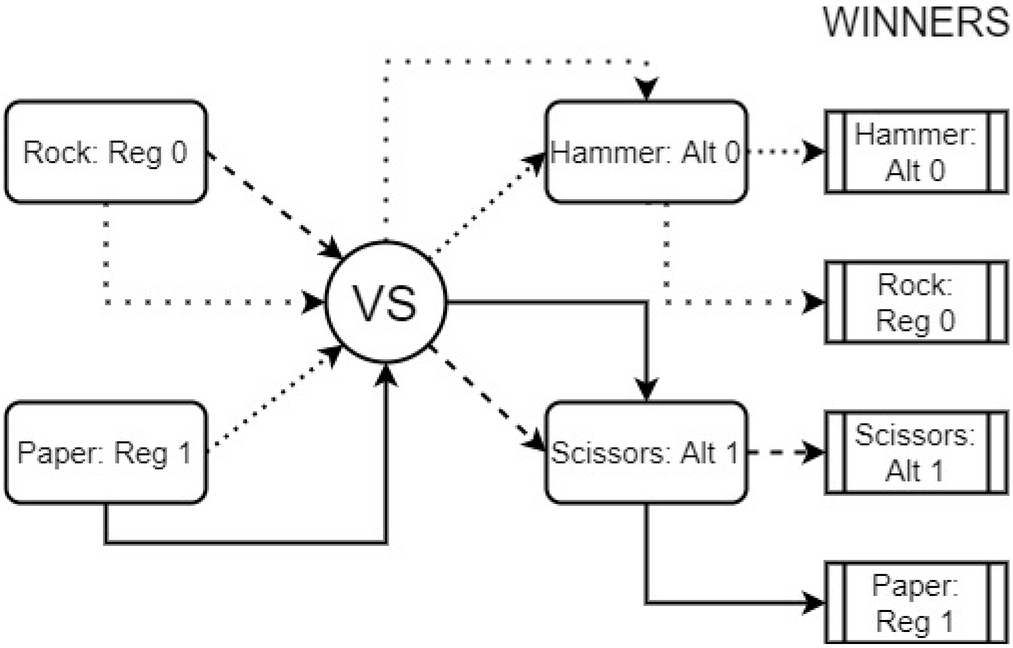
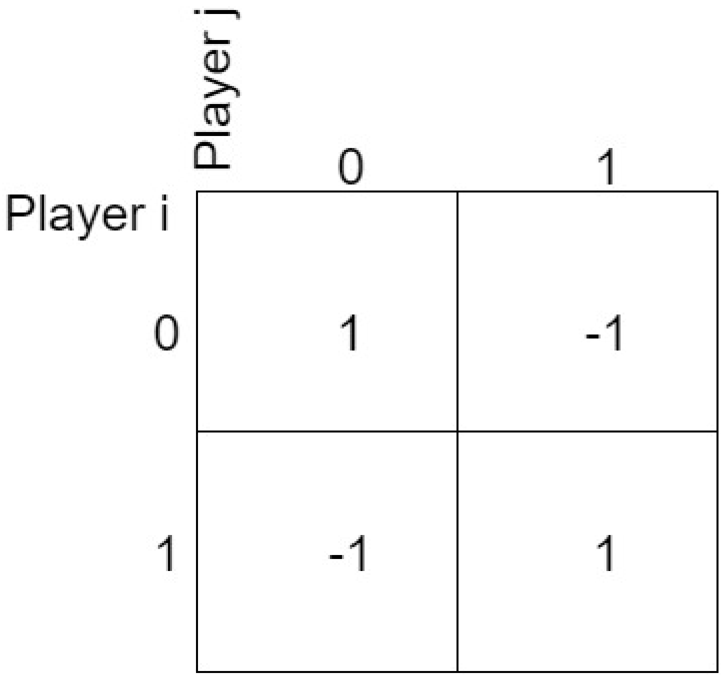


Figure 3.4: Rock, Paper, Scissors and Hammer

The game can only be played between two players, but it offers the advantage that the game never ties. This makes the process simple as each game ends in only two possibilities, either player 1 wins or player 2 wins.

Each player's set of possible options is denoted by, $A = \{0, 1\}$ and $\Delta A = \{(p(0), p(1) \in R^3 \mid p(0) > 0, p(1) > 0 \, and \, p(0) + p(1) = 1\}$ denotes the set of probability distributions on A. Assuming player i and player j are two players competing such that, $i \in \{1,2\}, j \in \{1,2\}$ and $i \neq j$. Figure 3.5 shows Player i's payoff matrix.



Player i \ Player j	0	1
0	1	-1
1	-1	1

Figure 3.5: Player i's payoff matrix

Player i's payoff is represented by,

$$EU_i(a_i, (p_i, p_j)) = (p_i, p_j) \in \Delta(A) \, x \, \Delta(A) =$$

$$\textstyle\sum_{(a_i,a_j) \in AxA} p_i(a_i) p_j(a_j) U_i(a_i, a_j) \qquad (1)$$

Now, it is known that a pair of mixed strategies (p_i, p_i) is a mixed Nash equilibrium, if either of the player's strategy is the best response to the opponent's strategy [58]. Consider the strategy profile $((\frac{1}{2}, \frac{1}{2}), (\frac{1}{2}, \frac{1}{2}))$. If $p_j = (\frac{1}{2}, \frac{1}{2})$, then,

$$EU_i(a_i, (\tfrac{1}{2}, \tfrac{1}{2})) = \tfrac{1}{2} \, x \, (-1) + \tfrac{1}{2} \, x \, (1) = 0 \; \forall \, a_i \in A \qquad (2)$$

Thus, the strategy $((\frac{1}{2}, \frac{1}{2}), (\frac{1}{2}, \frac{1}{2}))$ is a mixed Nash equilibrium.

Suppose player i plays any other strategy different from $((\frac{1}{2},\frac{1}{2}),(\frac{1}{2},\frac{1}{2}))$.

$$p_i \; = (p_i(0),\, p_i(1)) \; \neq \; (\tfrac{1}{2},\tfrac{1}{2}). \qquad (3)$$

Without loss of generality, let us assume, $p_i(0) \; \geq \; p_i(1)$, that implies, $p_i(0) > \frac{1}{2}$.

So,

$$EU_j(p_i, 1) = p_i(0) - p_i(1) \qquad (4)$$

$$p_i(0) + p_i(1) \; = \; 1 \qquad (5)$$

$$1 \; + EU_j(p_i, 1) = 2p_i(0) \; > \; 0 \qquad (6)$$

It is known that if $p_i \; \epsilon \; \Delta(A)$ is a best response to $p_j \; \epsilon \; \Delta(A)$ and player i plays $a_i \; \epsilon \; A_i$ such that $p_i(a_i) > 0$, then $EU_i\big(a_i, p_j\big) \geq \; EU_i\big(\acute{a}_\iota, p_j\big) \; \forall \; \acute{a}_\iota \in A$ [38]. This means that $p_j(1) = 0 \; \forall \; p_j$ is the best response to p_i.

Also,

$$EU_i\big(0, p_j\big) = p_j(0) - p_j(1) \text{ implying } p_j(0) > 0 \qquad (7)$$

So, we can conclude that any other strategy except $(\frac{1}{2},\frac{1}{2})$, is not a best response to any mixed strategy. This proves that there can be no mixed Nash equilibrium in any other strategy except for $(\frac{1}{2},\frac{1}{2})$.

There are two pure strategies in the RPSH game. (x, y) denotes the generic social state of the population, with players using a strategy of either 0 or 1. At time t assuming, player $n_0(t)$ played 0, while player $n_1(t)$ played 1. Now, $n_0(t) + n_1(t) = N$. Thus, $x = \frac{n_0(t)}{N}$ and $y = \frac{n_1(t)}{N}$, where $x + y = 1 \; and \; x \geq 0, y \geq 0$. The total number of different social states for a population of size N is expressed as $N = \frac{(N+1)(N+2)}{2}$. So, for N=2, total number of social states are 6 [39]. The evolutionary trajectory for the state space system for RPSH, would form a triangle in 2-dimensional space with center point of the triangle being the Nash equilibrium point or $(\frac{1}{2}, \frac{1}{2})$.

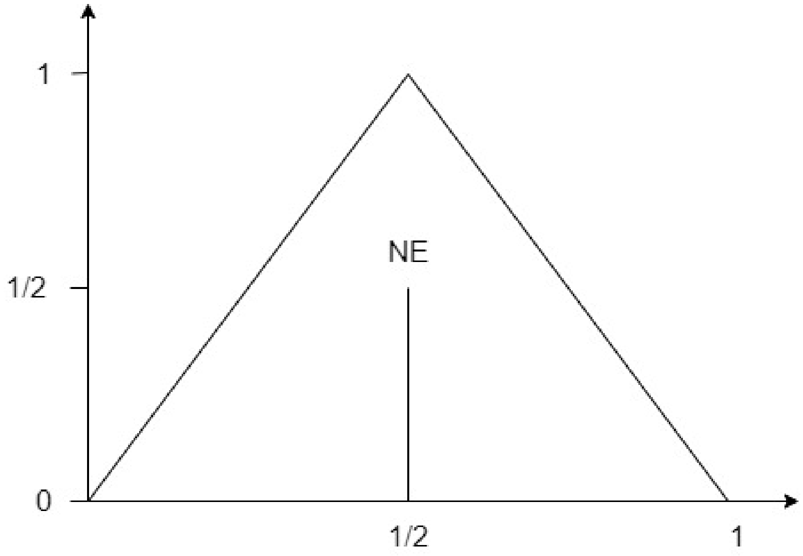


Figure 3.6: Social State Space of RPSH

3.4 IoT Devices and RPSH

Rock, Paper, Scissors and Hammer is a non-cooperative simple game, that helps arrive at a decision between options that hold equal merit, every time. One important aspect of playing the game is that the opponents need to reveal their gesture or their choice at the same time. This is an easy condition to fulfil when all involved parties are physically present at the same location or are visually connected, so that they may be able to present their choice without knowing any of the opponent's choices, thereby playing a fair match.

Users connected over an IoT network face a disadvantage when attempting to use such a process for making a decision. Such a situation requires a different approach when playing the game. Certain requirements to this are,

- The players should be able to share their selection without any interference from unrelated third parties
- None of the players should be able to change their selection once shared
- None of the players should have access to what the opponent has selected without having selected themselves
- The players should be able to stay anonymous if they choose so

This can be achieved by integrating a blockchain network using a smart contract to govern the transaction between the users.

Using a two key encryption method such as RSA, the user can encrypt their selection and share it with their opponent. Once the first player shares their encrypted selection and the public key with the opponent, a smart contract triggers a second transaction, allowing the second player to pick a selection in a stipulated time frame, encrypt it and further share it with the first player, along with the public key. This is stored in a block on the chain, the transaction is only approved each time, when the user shares their selection along with the key. The second player is only allotted a stipulated time which ensures that they do not attempt any unfair means of breaking the encryption or using any other unfair means to influence the outcome. Once the transactions are successfully completed, both the players share their private keys, this can help decrypt each of their selections. Such an exchange ensures that all the requirements are met, and the players arrive at a fair decision [40].

CHAPTER FOUR

Illustration of Practical Situations

Large virtual networks rely on the internet for providing several services. This puts all the users at a risk. Authorization and verification of service providers is uncertain. There is no surety of whether the person you are dealing with has any malicious intent or not [41]. Corporate companies have had a long history of cheating for their own gains. Over the years several car companies have been found to mess with the car's onboard computers to indicate wrongful compliance with laws. Since 1970, several car companies have been found to using defeat devices, that can manipulate a car's horsepower or mileage, by letting dirtier gas exhaust than set standards. A University of Denver professor, Donald Stedman, from the Chemistry department once said that the economics of the automobile industry make it profitable for car manufacturers to cheat. To maintain lower costs and high product standards, car manufacturers have been found to use methods that have gone against global standards for environment protection [42]. A study estimated that Volkswagen's use of defeat devices has led to an estimated 36.7 million kg of excess nitrous oxide gasses into the atmosphere, between 2009-2015 alone [43]. This makes it necessary to have a model which cannot be accessed by any third party. This also makes it necessary to have a decision-making model, involving autonomous cars to be independent of any interference by the agents themselves. As car manufacturers have been known to reprogram the automobile's ECU for their gains, any trust model that can be influenced by the agents themselves cannot be implemented and any computational trust has to be built free from any influence from any of the agents.

Computational trust is dependent on abstract notions of human trust. This means any models need to be built considering lack of reliable information and need to be uncontrolled by any involved agents. It is nearly impossible to predict how robust a system is with change in environment and how well it behaves under various assumptions. A model of trust between unknown agents only works when protocols are rigorously defined [44].

The model described above can be used for any situation that requires users to take quick optimized decisions. This can prove especially valuable to users connected via an IoT network. Several instances where at autonomous devices need to work together to achieve a common goal or make decisions without the hassle of complex optimization algorithms. Using blockchains to set the communication protocols for such devices can make them more secure, while reducing any delays in operation caused due to any interference from outside agents.

4.1 Parking Spot

Smart parking has become an issue over the years, with the increase in number of automobiles and urban population. It has both economic and strategic advantages. Parking has also become a big source of revenue for many cities. With smarter parking services, drivers can conserve resources spent, looking for a spot. A smart parking ecosystem can also solve driver conflicts. With many drivers looking for a space, a competition occurs, and this results in cumulative parking conflicts [45]. It is important to not just advertise parking availability to all nearby drivers but also let them attempt to solve any rising conflicts to streamline the process.

Considering the parking scenario as shown in figure 3.1, a private cross chain communication model is proposed. There are two agents (cars) that need to access a parking spot. Both enter a parking lot with only one spot available. The model uses a two key system where both are assigned their private and public keys. Data is converted into cash flow, the ledger updates as messages are sent across two chains with the cars acting as nodes. The two agents begin by forming the genesis block. The architecture for the model involves a threaded system. The exchange of information forms a block. The ledger forms the stack. The model uses RSA encryption with a 2048 bit key.

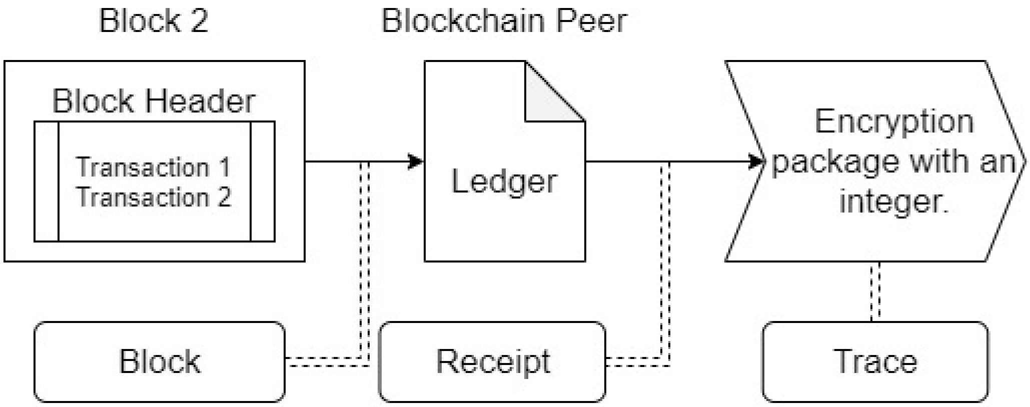


Figure 4.1: Message Passing

Threads are sets of commands that need to be executed in order, this ensures all necessary tasks are being performed by the system. Figure 4.2 shows the working of the process flow. All the transactions are recorded onto the ledger. The agents move onto the next transaction once all agents confirm receipt of response from the previous step. Thus, the smart contract stays enforceable without any governing third parties. This is made possible by integrating the use of message passing and thread management.

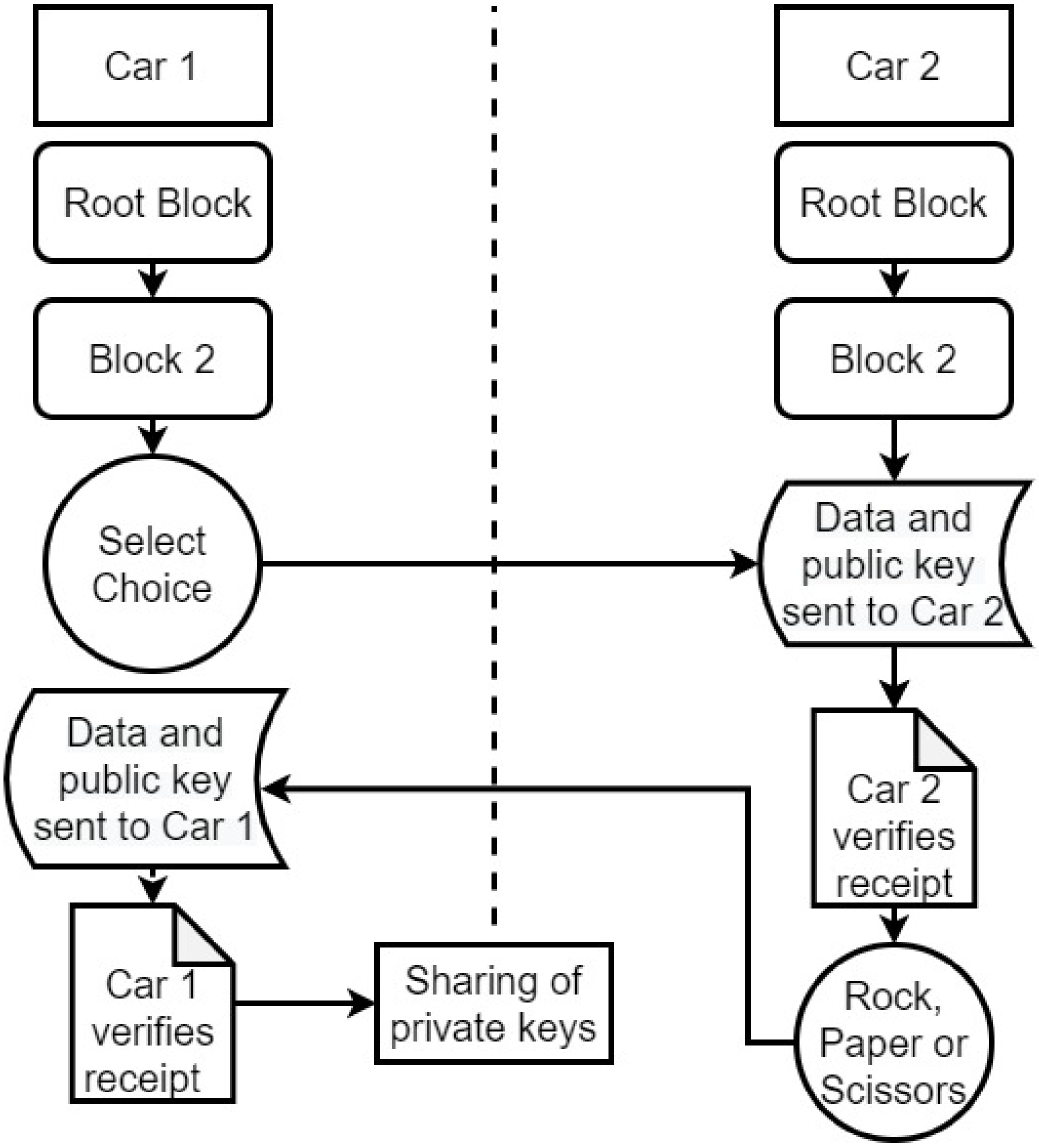


Figure 4.2: Communication Model

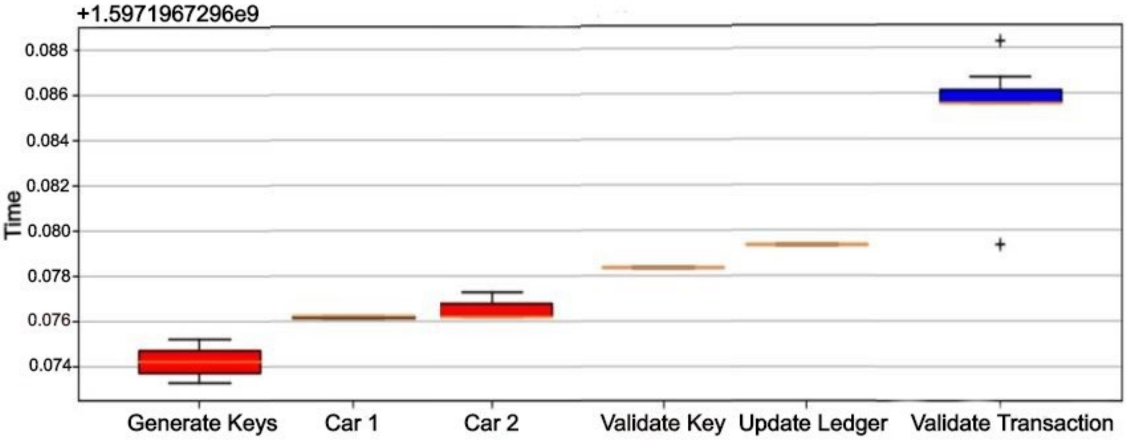


Figure 4.3: Timeline

The two agents play a fair game of RPSH. Appendix 1 highlights the algorithm of how the game is implemented and played by the two agents using blockchain. Each of them takes turns in making transactions. Figure 4.3 illustrates the complete process. This begins with the first agent choosing an option. The choice is encrypted using the keys and sent to the opponent. Now, the second agent knows that the first agent has decided, but there is no way of knowing what it is, unless he uses the private key to decrypt the data. The second agent now has a stipulated time to make his decision. Once the agent chooses, another transaction takes place where the agent shares his decision, and the key with the first agent. Both the agents share their private keys. This helps them know each other moves simultaneously. The transaction is verified along with the outcome of the game. The winner gets the spot. Figure 4.3 illustrates the timeline for the corresponding events. The timeline is divided into 6 events. The first is generate keys, where the private and public keys for the data encryption are generated. This is followed by the transactions by each agent. Finally, the ledger is updated, and the final decision is made, based on the choices.

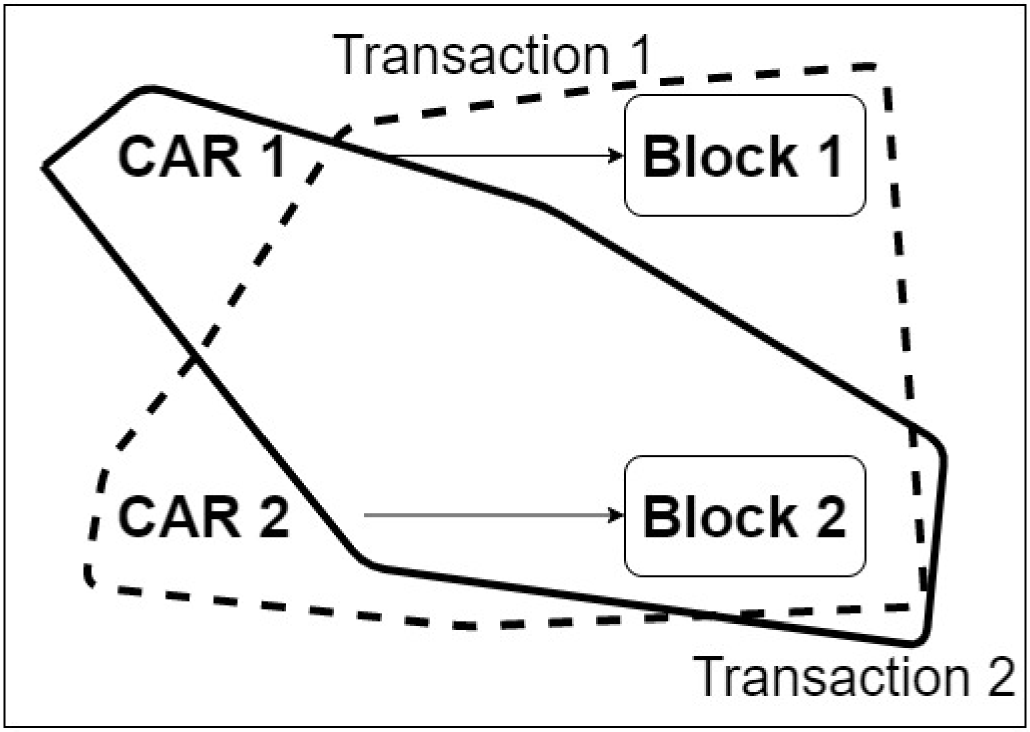


Figure 4.4: Transactions between the agents.

The complete process as shown in figure 4.5 is divided into two transactions. The first transaction is when the first agent sends the selection to the second agent. This leads to creation of the first block for both agents. The second transaction is when the second agent shares its selection with the first agent and the second block is created. This results in a small but private chain for each agent.

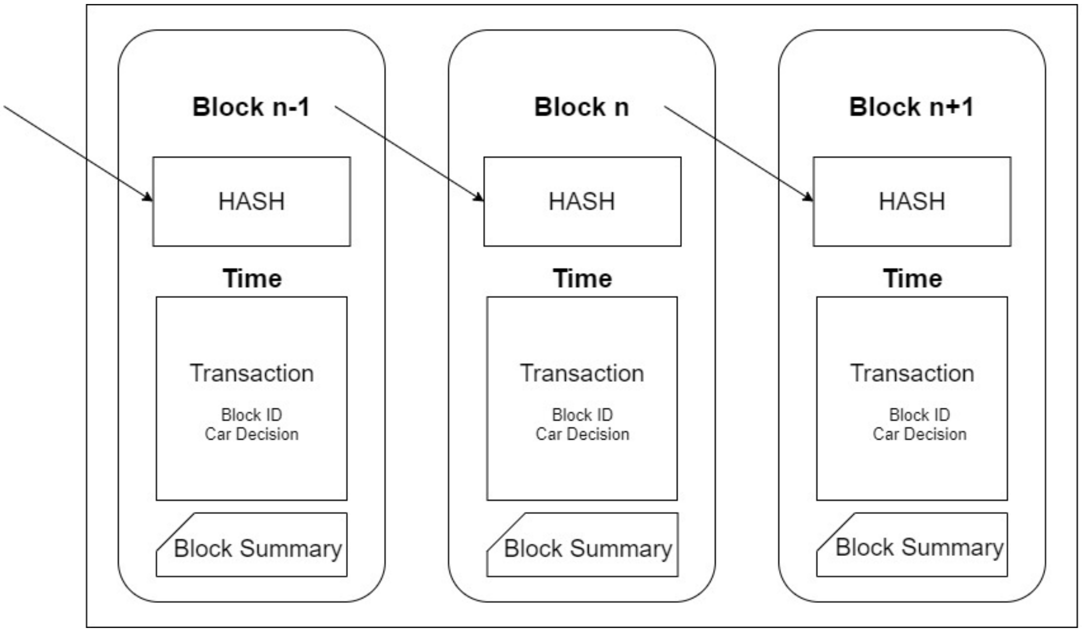


Figure 4.5: Block summary for the model

As discussed earlier the hash of a block is what makes the chain tamper proof. Each block is connected via the hash. As seen in figure 4.5, the blocks and the ledger ultimately, records the decision that each agent makes. The decisions are recorded as cash flow and each transaction is only valid when each agent makes their choice in the stipulated time.

One of the interesting features of the model are message and key passing. The data is encrypted using the RSA algorithm. It uses a two-key method, a public and a private key, as illustrated in figure 4.6. The decision taken by each agent is encrypted and until the opposing agent makes its decision, the private key to decrypt the decisions is not shared. RSA is the first algorithm that can be used for encryption and to generate digital signatures. The algorithm is secure because it is nearly impossible to decompose large numbers [46].

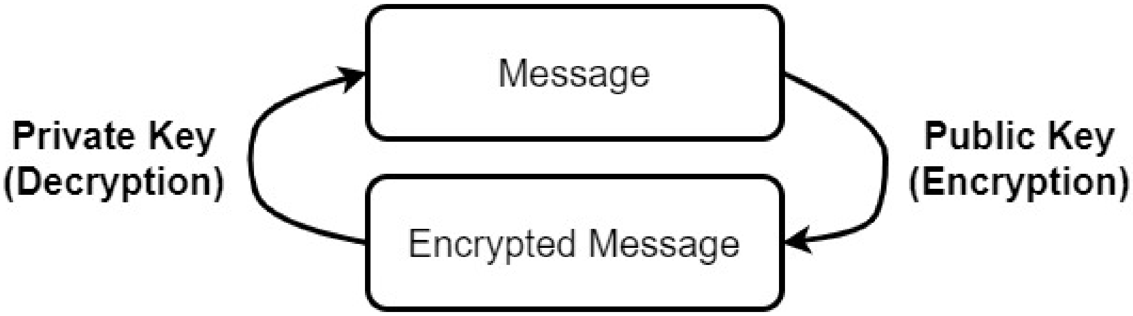


Figure 4.6: RSA Encryption

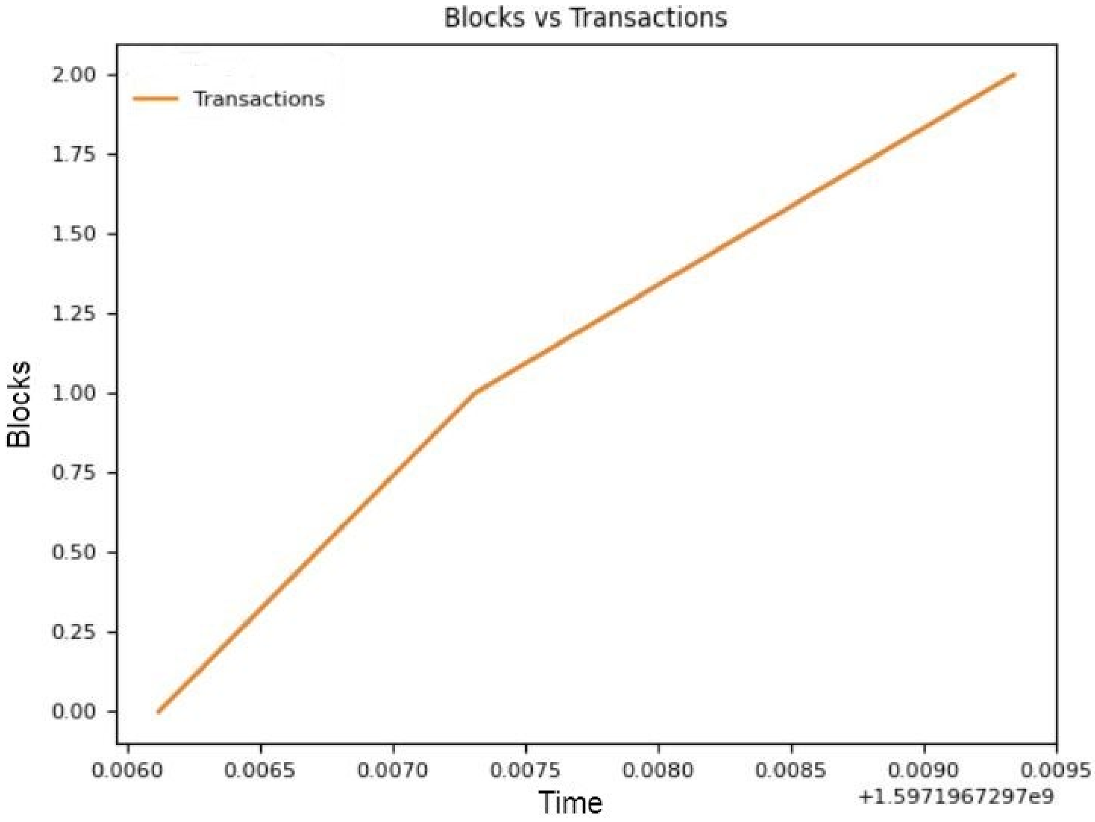


Figure 4.7: Transactions and Block Creation

Cross chain communication plays a very important role in the complete operation. Considering each event as a discrete one, figure 4.9 illustrates all the points where the two chains interact. Figure 4.8 is a time plot showing the formation of blocks as the transactions are successfully authorized. Each transaction is divided into discrete events and plotted versus time. The dots act as markers for each event. Comparing the two plots you will notice how the first block is created somewhere between 0.0070 and 0.0075s when the first transaction occurs between the two agents and there's a marker at the exact same time on the event plot. This is when the first agent shares its decision with the second agent. It takes agent 2 about 0.025 seconds to respond back. The plot represents a private chain making transactions seamlessly with limited number of blocks. Such a model works similar to a public blockchain but with lesser number of blocks, it requires lesser computational power and easier consensus algorithms, thereby showing the effectiveness of private blockchains.

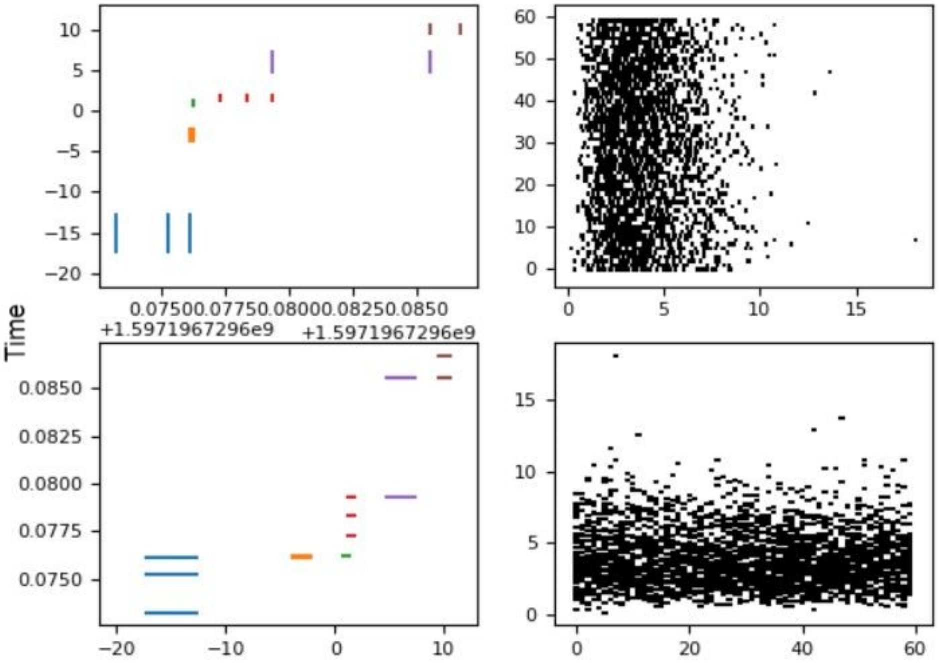


Figure 4.8: Event Plot

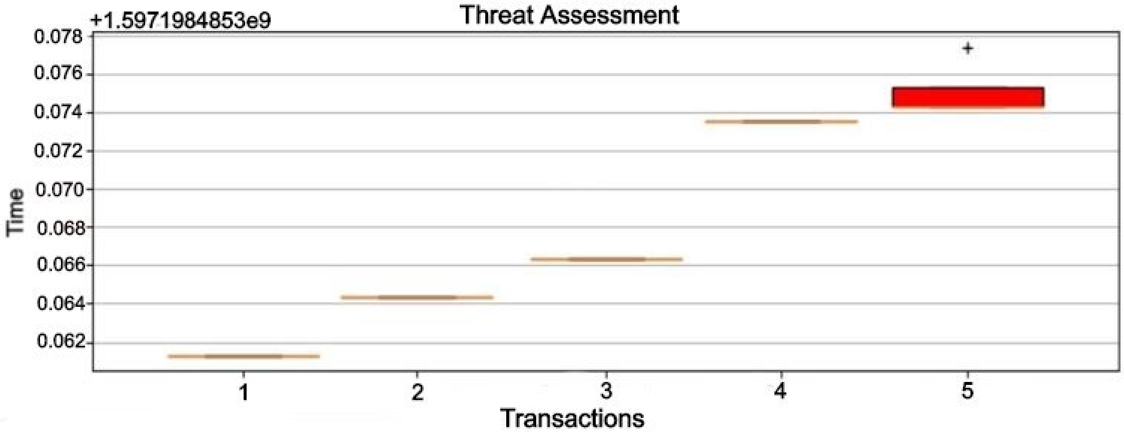


Figure 4.9: Threat Assessment

Since, there is always the possibility of having malicious agents, or agents that don't want to play fair. A threat assessment was done to highlight the security features of the model. Appendix 2 highlights the different vulnerabilities that were assessed and added into the simulation to ensure the communication was secure. Figure 4.9 is a box plot that illustrates the different attacks, that were tested. The second agent is given a stipulated amount of time to pick an option, if the agent fails to do so, the transaction fails. This discourages the agent from using any unfair means to break the encryption. It further ensures that the operation is completed quickly. As seen in the plot the first transaction where the agent takes longer than 0.6s to pick a choice, the transaction fails. Every blockchain transaction is only authenticated when the digital signatures for the participants are verified. In case the signatures are incorrect the transaction fails. Event 2 and Event 3 represent transaction failure due to the two agents failing to use the correct signatures to verify the final transaction. The forking in blockchain is the problem when a miner or a user intentionally introduces a long chain such that the current blockchain has two paths to take, in an order to access the blockchain's data. The fourth event is where a couple dubious blocks are introduced as an attempt to replace the good blocks on the chain with malicious blocks. These were flagged and the transaction fails. The final event is the model with no attacks on the chain and a fair consensus is achieved resulting in a successful transaction.

It is important to note that blockchain is an integral part of the model. Using the privacy features of blockchain, the need for a third party is eliminated, Figures 4.10 and 4.11 show the process flow of using a third party vs not involving a third party. The communication model without using blockchain can open up the situation to several vulnerabilities,

- Cars can attempt to sabotage and take control of the other car's onboard computer.
- Record keeping becomes tedious and must be integrated separately. Also, it becomes hard to keep a record of the interaction in real time.
- A centralized model would require higher computational power and a fair third party to make the decisions.
- Transparency can be achieved since all parties involved in the chain can keep a track of all the interactions.
- All data recorded is immutable and none of it can be altered, which is the most important advantage of using a blockchain.

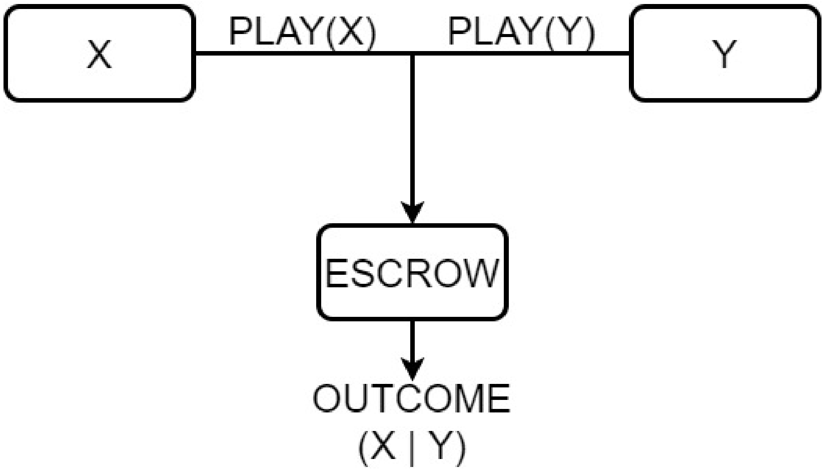


Figure 4.10: Process involving a third party

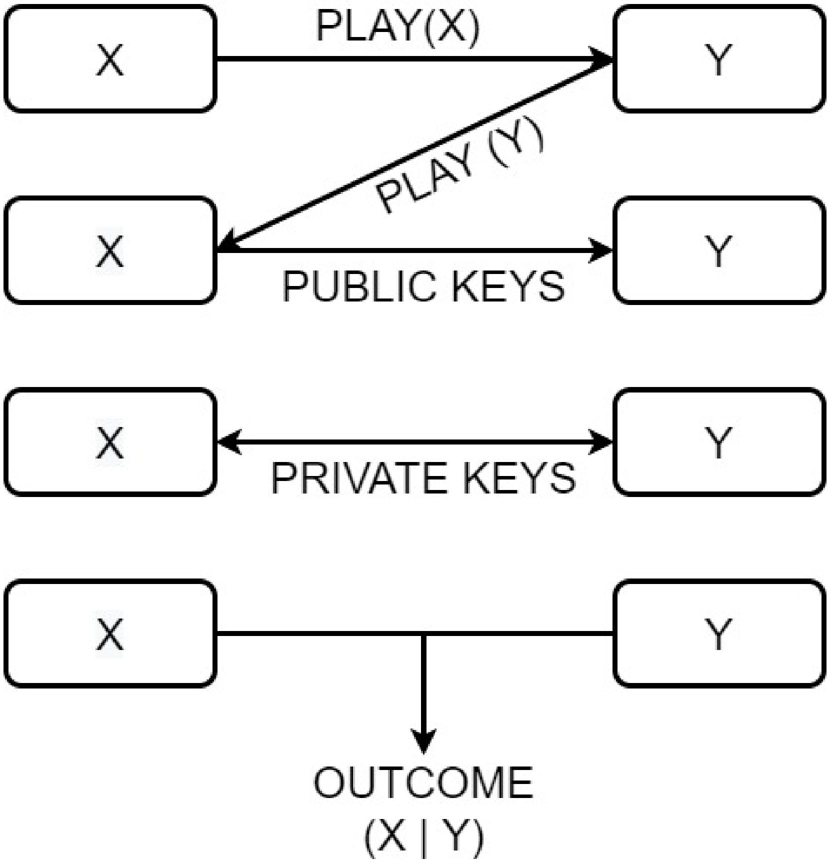


Figure 4.11: Process eliminating a third party

4.2 Forklifts Passing Through a Passage

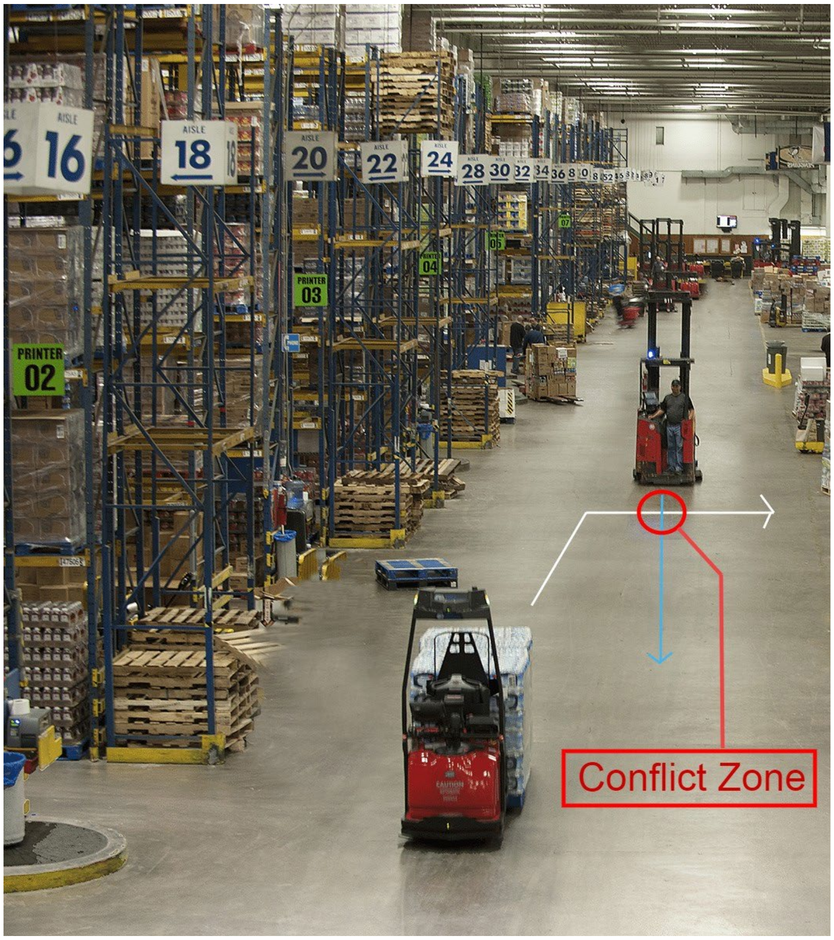


Figure 4.12: Forklifts Passing Through a Passage (adapted from [47])

In a factory setup, there are multiple forklifts running all day. These forklifts carry a lot of raw material and tools from the warehouse to the manufacturing lines. They are an indispensable part of all logistic operations. They can be used to carry up to 8000 lbs. of loads. These create increased efficiencies and play an important functional role in the production process [48]. There are about 540,000 operators working in 300 different industries in the United States [49].

With a huge number of forklifts working around the country, it becomes important to plan out optimal route strategies for all these forklifts working round the clock, for all scales of operation. It is also detrimental to establish secure communication lines so these lifts can interact with each other as necessary and carry out their daily assignments. With the advent of autonomous vehicles and robots, more of these industrial trucks and forklifts may be made autonomous as well.

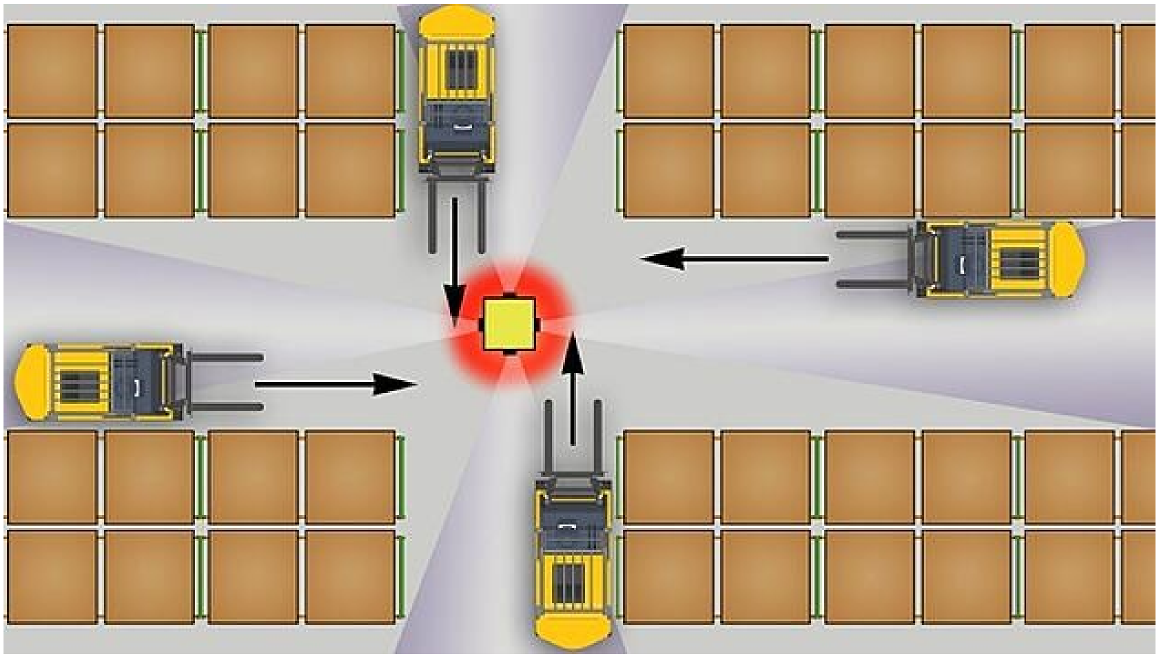


Figure 4.13: Conflicting Paths [50]

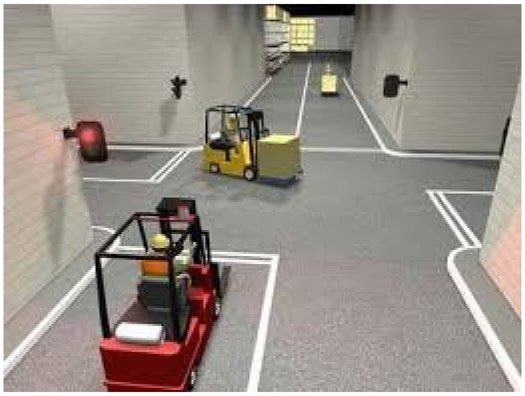


Figure 4.14: Shared Path [51]

Consider a scenario with two forklifts wanting to access parts in the same aisle as shown in figure 4.12. Both are headed towards the aisle and need to pick their respective parts up and get it at different locations, so the production lines can stay stocked and on schedule. None of them can afford any and both want to take the optimal path to conserve energy and time. This is also true when two forklifts need to make a decision of passing through a door or have conflicting routes as shown in figures 4.13 and 4.14. In such situations it becomes necessary to establish communication between them, so both are aware where they are heading and avoid any mishaps. Even if both get there and avoid any accidents, they might reach a standstill and then need to negotiate on the path to be taken, this will result in delays and reduced efficiencies.

Establishing a network to connect these forklifts over a blockchain, where they can communicate securely and make decisions on path planning can help save a lot of time and increase efficiency. The forklifts use the model described above and can decide on who takes the optimal path and who takes the alternative path, or they can decide on who enters in first and which one follows. This can help take this quick and simple decision without having to waste any resources.

Moreover, since all such transactions are recorded onto a ledger by the blocks, they can be later accessed to check for how many times were their conflicts between different forklifts on their path. This data can help plan future path planning operation or plan the time of order for such forklifts better. Thus, the model doesn't just help avoid conflicts in the present but the data can be used to plan future operations in a much safer and efficient possible manner, without having to dedicate a lot of resources to the cause.

4.3 Vehicles Switching Lanes

Figure 4.15: Switching Lanes [52]

A more everyday application for such a decision-making model would be autonomous vehicles achieving lane change maneuvers safely as illustrated in figure 4.15. For various lane change maneuvers, especially on an interstate when lanes merge, it becomes important to take quick decisions, that not only account for change in your path but others as well. In such a system coordination is extremely vital. Blockchain can facilitate secure communication between all these vehicles. The ledger can be used for insurance purposes in case of accidents, as all the interactions recorded are done in real time and are immutable. Good and quick communication can not only save from any damage, but it can also conserve resources and save energy by avoiding activities like sudden change in speed. In such situations the model could prove to be an effective tool.

4.4 Robotics and Commercial Autonomous Vehicles

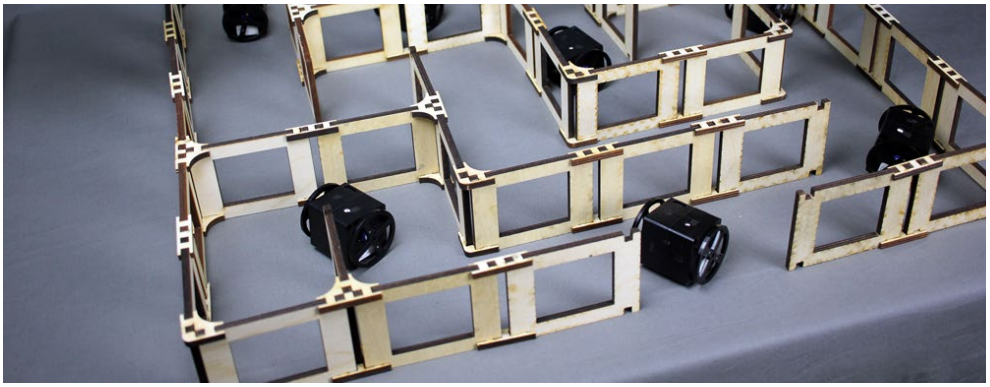


Figure 4.16: Swarm Robotics [53]

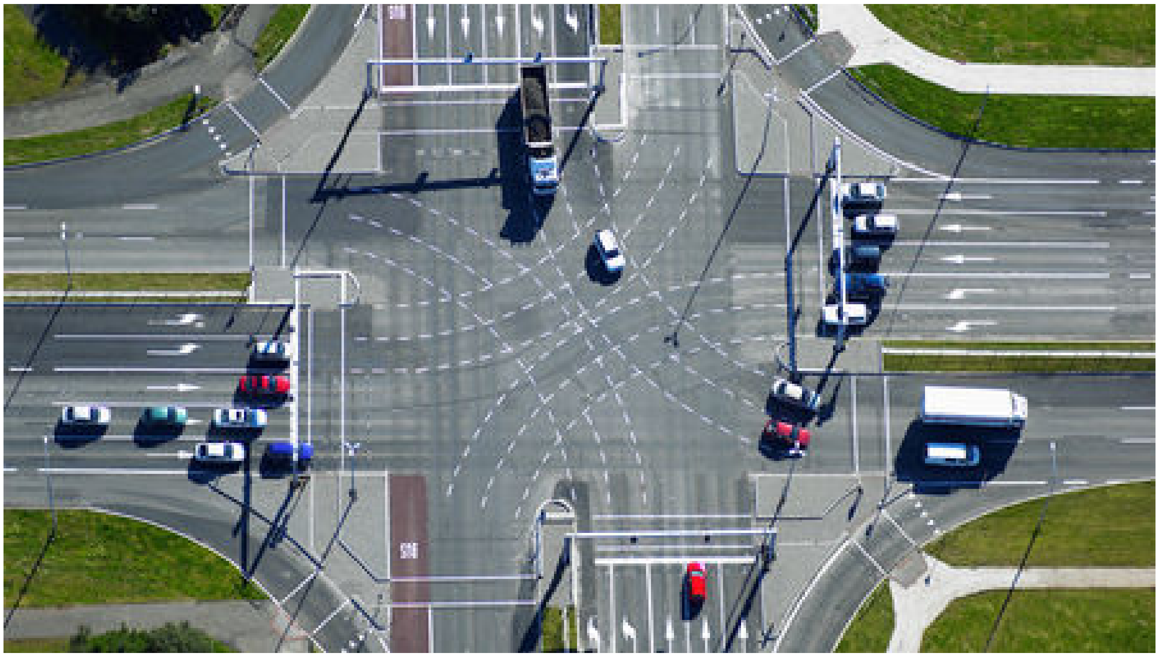


Figure 4.17: Connected Vehicles Environment at an Intersection [54]

Swarm robotics deals with multiple robots in a connected environment dedicated to achieving a common goal. Figure 4.16 illustrates a situation where the robots are trying to work together to realize a common goal of navigating the maze. Using IoT these can stay connected and coordinate, blockchain can protect them from deviating from their goal or any outside attacks. The ledger that records all these interactions can later be analyzed for any optimizations in path or in control. Figure 4.17 illustrates a similar situation with connected autonomous vehicles at an intersection. Communication is key in all such situations and can lead to optimized trajectory planning and resource distribution. The model can essentially help make communication more secure along with solving any conflicts.

Autonomous vehicles and robots use numerous sensors to gather data about their environment. These also have the ability of collecting personal data of users around, including the location and everyday movement data. Such a situation holds multiple threats. First, this can be used as a surveillance system by private companies and second ethical decision making. Autonomous agents can further increase any bounds set for their own development. Autonomous vehicles operate in the public domain and thus have a high chance of invading the privacy of others around. Moreover, such an open network is susceptible to attacks, for data thefts or other malicious intent. Companies might also use this to their own advantages. Insurance companies might track license plate number to check for speeding and increase insurance rates. They could log physical movements of every person in the environment and be used to stalk them. All these take place in a public setting and thus are at a higher risk of data loss. Autonomous cars can turn into new privacy invasion tools. There is no accountability to these companies of these devices. Nor is there any traceability of any of the data being collected. In such a situation, users lose any control over their data. Also, such an open network is highly susceptible to any attacks [18].

4.5 Competing Agents

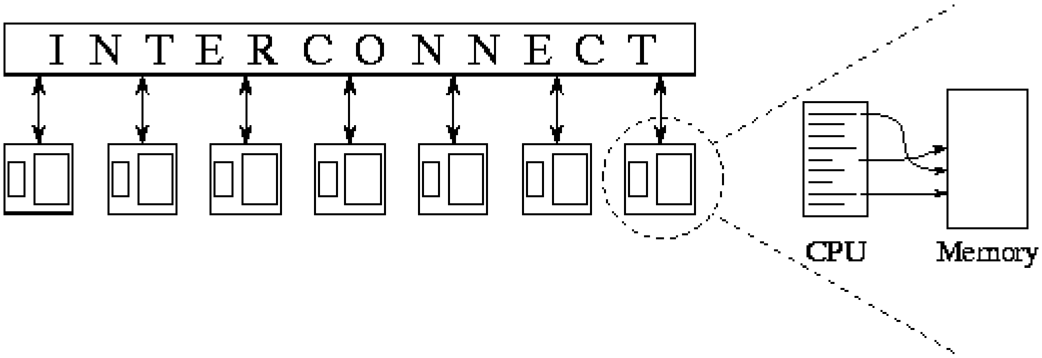


Figure 4.18: Parallel Machines [55]

Several situations demand agents to compete for common processing resources. For example, in a parallel machine environment as shown in figure 4.18. In such cases it is important, to achieve the best solution for one or all agents, while making sure the agents left do not have to accept solutions at high costs. Such a situation can arise in several application environments, where negotiations are needed. Machine Language (ML) has been used to solve a large number of such situations including but not limited to, devising schedules for agents with unacceptable task timings, supply chain problems, scheduling trains to share railway tracks and sharing of satellite communication resources. It is important to serve all the different needs or objects that different agents might have. Some agents might accept delays, while others might accept data loss, it is important to keep that in mind [56]. But not all agents have enough computational power to support ML algorithms and these require a simpler and computationally less invasive methods. Moreover, some situations cannot support alternative scheduling and there are limited resources that agents must pick from. In such cases it is simpler to have a non-cooperative game theory approach in settling all arguments.

4.6 Conflict Resolution in a Connected Vehicle Environment

Intersections are one of the major contributors to traffic congestion. A conflict zone is an intersection where different vehicles access the crossing at the same time. The model can be implemented to provide efficient ways for crossing vehicles and to cooperate with other vehicles approaching the conflict zones. Instead of a centralized controller, vehicles can themselves negotiate for time and space allotments. Output trajectories can be planned accordingly, and energy can be conserved based on optimal trajectory allocation. Vehicles can decide whether to yield or pass and thereby avoid collision [57].

Existing conflict resolution algorithms are mainly centralized, with emphasis on control of traffic using traffic lights. Other systems with extensive communication models still have conflicts between agents with same strategies. A ‘yield’ and ‘go’ model can still experience deadlocks in local decisions. Such methods are inefficient as they may require vehicles to stop even in no conflict zones. For, multiple vehicles at a 4/2 way intersection, it becomes a challenge to determine who must go first. A distributed conflict resolution mechanism should indeed be feasible, it should not be computationally intensive and should have a passing order for any set of vehicles passing through a conflict zone. Decisions should be made in real time and no two vehicles should either be allowed to go or yield at the same time. The priority should always be on finding a feasible solution [58].

The proposed algorithm is suitable for real time control. Only nearby agents need to participate and resolve the problem together, this makes it easier for them to communicate, instead of having a lot of agents competing for communication resources. Also, agents on the downstream can take charge and later decide with the passing vehicle on who slows down and relay this information to succeeding vehicles. Such an ecosystem is not computationally intensive and thus will have lesser delays and time lags. A fair outcome can be desired and none of the agents or car manufacturers can use any unfair means to influence the final decision.

4.7 Intelligent Docking System

It is important to extend any smart parking mechanisms to public transportation systems. This has several advantages in an urban setting. In rail networks different types of vehicles can share the same settings. But on road bus systems need to be more robust. Congestions imply longer wait times, decrease in quality of service and waste of resources. Driverless settings can employ various sensors ranging from Lidars to Magnetic sensors, for placement and detection [59]. The algorithm presented can be employed for quicker scheduling and route picking services. Docking of a fleet of such vehicles can be easier scheduled with internal conflict resolution between the vehicles. This can be extended to a fleet of commercial trucks as well.

4.8 Fog Computing and Sharing of Computational Resources

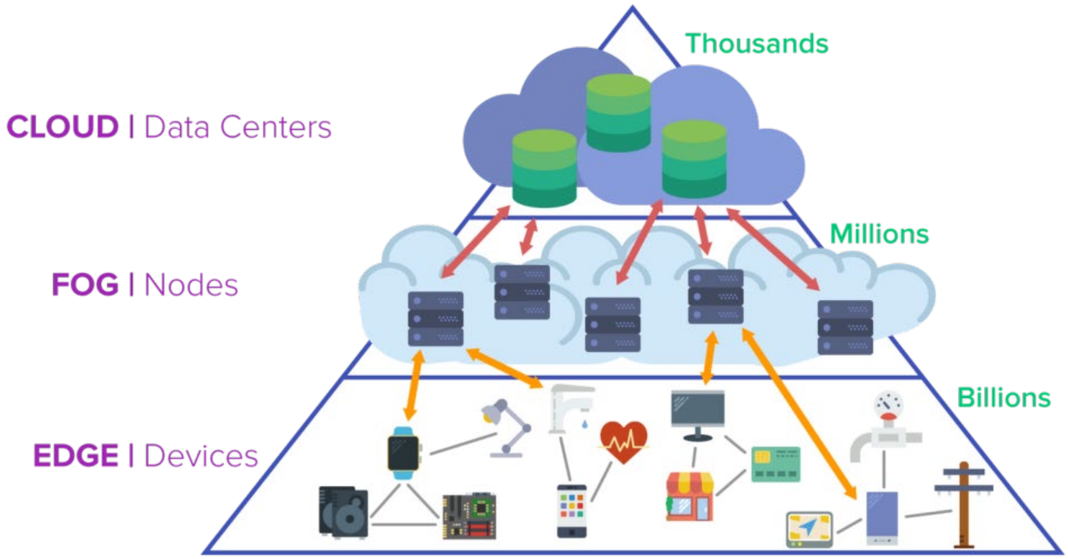


Figure 4.19: Fog Computing [60]

Fog computing is an extension of cloud computing with several advantages. It provides low latency and has a widespread distribution. It can support scalability and can support varied platforms. It can support a variety of critical IoT services. Fog computing provides computational and storage services to several end devices. This also requires cooperation models between users and providers that can provide services to clients. With more utilities and agencies switch to fog computing there will also be a need to assign and share resources within all providers. It can support real time analytics, for connected vehicle system or smart city applications. Fog computing can also support an IoT network with several devices and a bidirectional flow of information and resources. A blockchain supported IoT setting can be further integrated with fog computing, for real time interaction, data validation and support cloud data storage systems [61].

CHAPTER FIVE

Simulation

The following simulation has been developed for the parking scenario. It is an application developed in python to highlight and visualize all the interactions taking place between the two cars (Refer to Appendix 4). The chapter focusses on the algorithm for key passing and the visual simulation of RPSH. It is important to highlight the importance of IoT in the model. Various sensors can help establish trust between the two agents. IoT has several advantages to offer,

- Sensors ensure and authenticate the physical presence of a car and that it's not just a bot that is trying to hold the space for some other car as illustrated in figure 5.1.
- The internet offers a way to establish virtual communication.
- Bigger cars can attempt to bully any smaller cars. Cars can be used to block the other car and hold the space. A universal virtual presence with IoT can ensure no bullying or any agent with more resources does not take advantage of the situation.

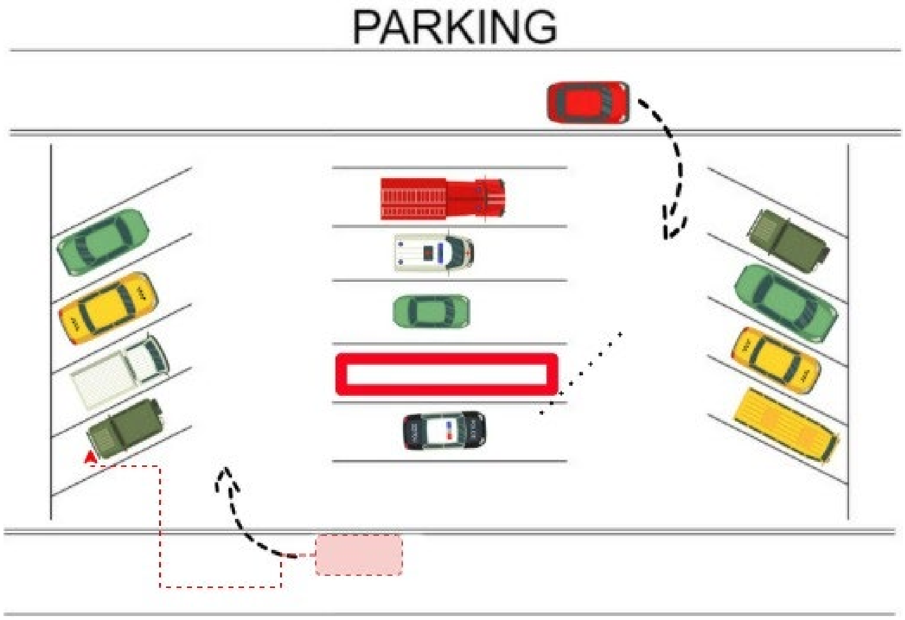


Figure 5.1: Human Generated Bot Mimicking a Car.

5.1 Key passing

One important aspect of the communication between the agents is message passing. An encrypted message along with the public key. As this key is not enough to decrypt the message, it assures the opponent that the decision has been taken.

```
function key_passing
    publicexponent = 65537, keysize=2048
    publickey = rsa.generatekey(publicexponent, keysize, backend)
    encoding and serialization to generate signature
    sign(message, privatekey) //see appendix 1 for sign function
    Transaction_input.append(publickey, message)
    if Transaction.verify() == True //see appendix 1 for verify function
        return True
    else
        return False
```

One of the applications of the described model is resolving a parking issue between two autonomous agents. The following sections comprise of a visualization of the model. The agents, ‘Car 1’ and ‘Car 2’ come to a unanimous decision on deciding who gets a parking spot using the Rock, Paper, Scissors and Hammer (RPSH) decision making game and later also the Rock, Paper, Scissors.

5.2 RPSH

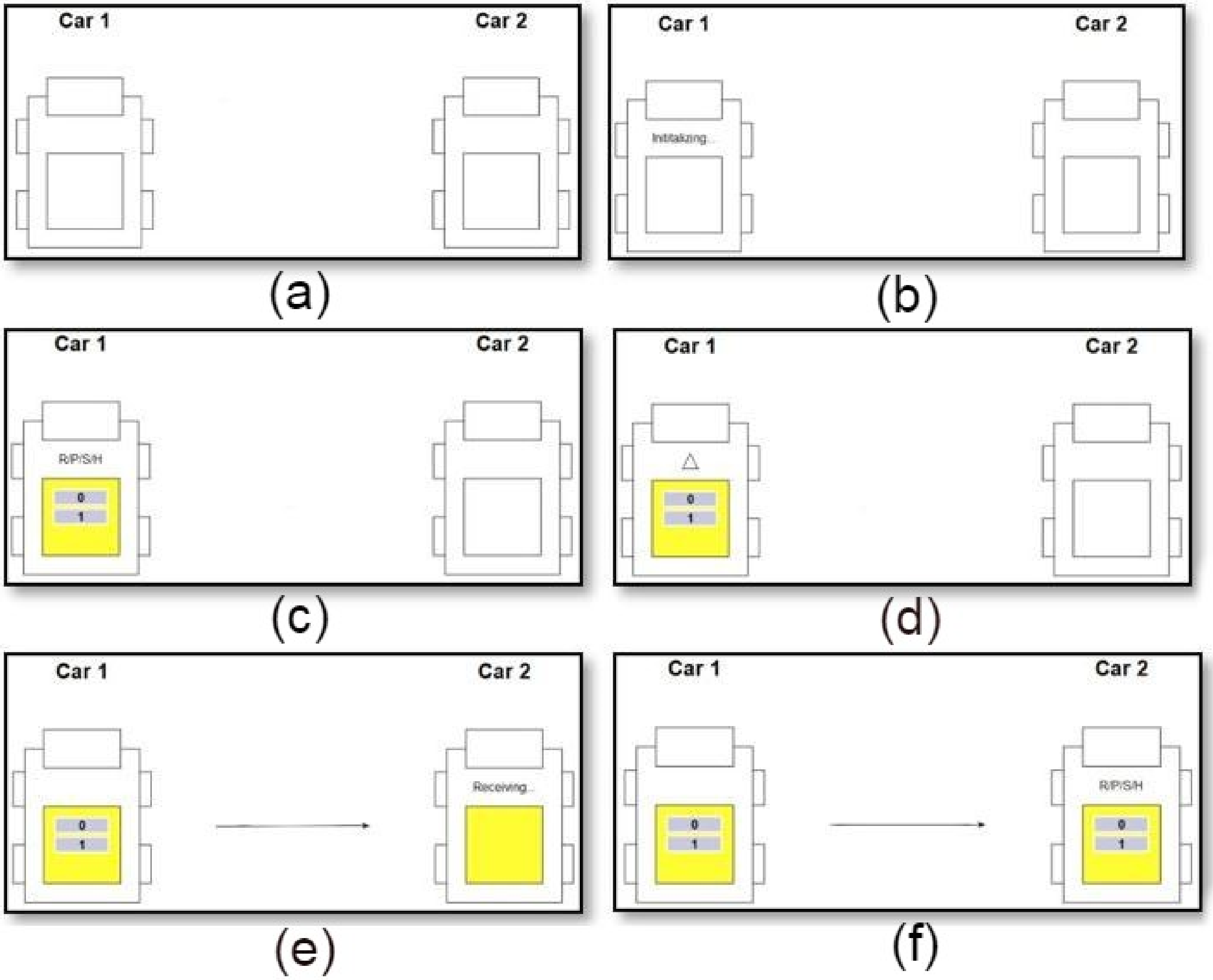


Figure 5.2: RPSH Simulation

Figure 5.2 (a) shows the beginning screen for the simulation applet. The applet was designed using python. Once you press the start button, the agents begin to communicate via a private blockchain network. Figure 5.2 (b) shows the agents begin initializing, a genesis block is created and each of them sends a message saying ‘hey”.

Next, The agent prompts the user to pick and option as shown in Figure 5.2 (c). Selecting it triggers the message passing algorithm. Appendix 3 discusses about the RSA algorithm that has been used to develop the required signatures and keys. RSA has also been used to encrypt the data and ensure that it is impossible to decrypt any data.

Figure 5.2 (d) shows a visual confirmation of the selection. This is extremely important in case of humans trying to interact with the machines. Vehicles must be capable of conveying information to humans interacting with the vehicle, regarding it's intention and it's performance [61]. Figure 5.2 (e) illustrates the completion of the first transaction. The selection is encrypted and exchanged. This prompts the second agent to begin making their selection as shown in figure 5.2 (f).

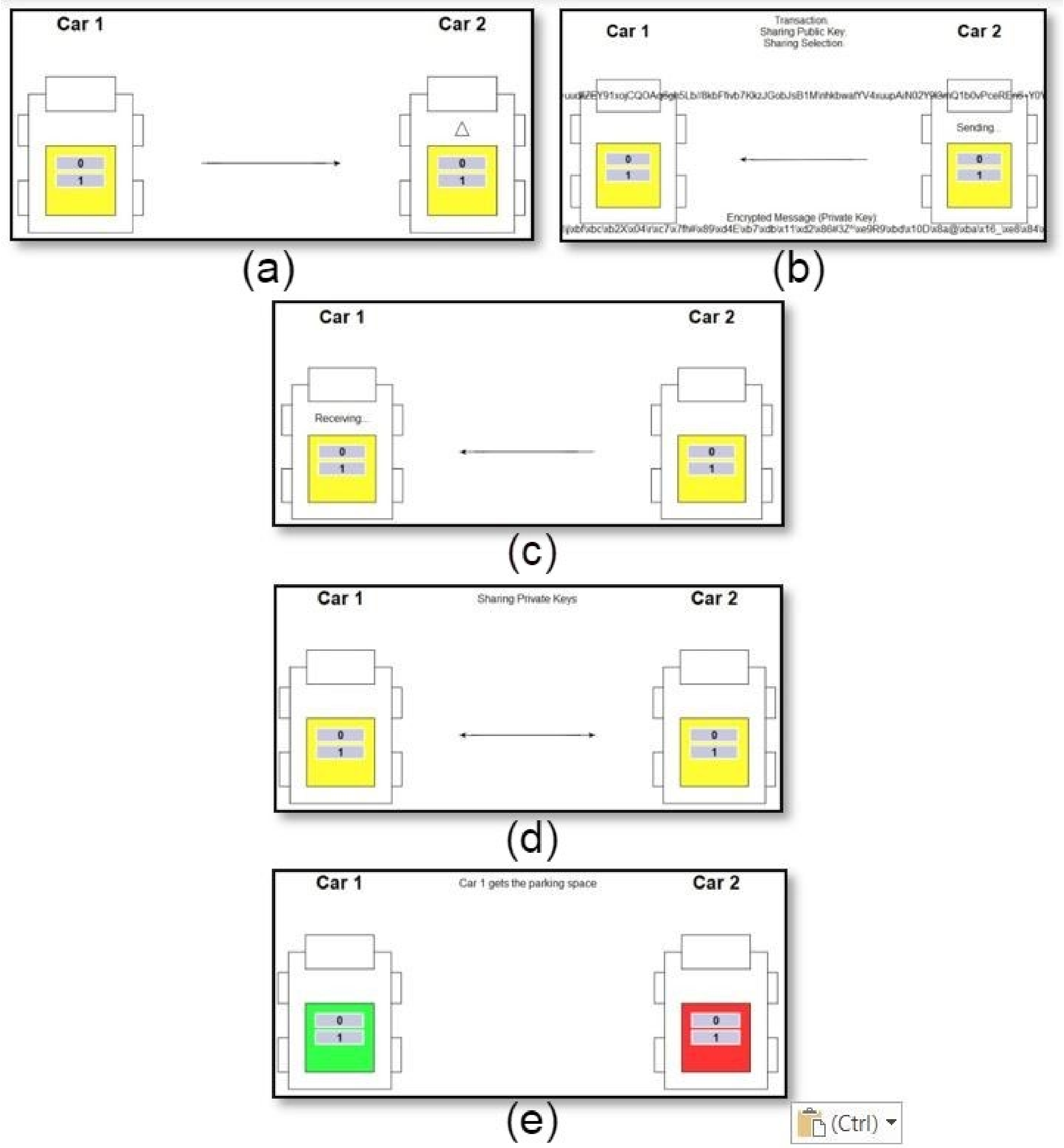


Figure 5.3: RPSH Simulation – 2

Figure 5.3 (b) shows the final transaction, where private keys are exchanged. The cars decrypt each other's selections and arrive on a mutual outcome as shown in figure 5.3 (e).

CHAPTER SIX

Conclusion and Future Work

The parking simulation shown demonstrates the effectiveness of the model. Using Rock, Paper, Scissors and Hammer, the drawbacks of the traditional game of Rock, Paper and Scissors have been countered. It highlights the versatility of the blockchain technology and how it can prove to be an important tool in governing autonomous agents, especially in a vulnerable environment. Companies have had a long-standing history of bending the rules. An environment with agents having the computational ability to take their own decisions, individuals with malicious intent can always try breaking into and sabotaging the environment for their own personal gains. An IoT environment with a network governed by blockchain can help build computational trust and bring accountability. Having an immutable blockchain ledger, with the ability to record all interactions in real time, helps determine faults in the system and develop solutions to required problems. The threat assessment shows several events where agents with malicious intent fail, highlighting the effectiveness of smart contracts. The method of Sealed Envelope Exchange gives such agents the opportunity to interact with unknown agents. Event plots highlight the activity of the agents in real time.

The future potential for this model is not just limited to the ability to setup communication between agents using blockchains, but also integrating sensors and making use of bio mimic logic to identify agents before they even begin interacting.

Exploring the ability to assign keys that act as digital identities, that can be used to interact over the blockchain. Such identifiers can create a safer and more trusting environment. Physical identification over digital keys can also help root out bots or computers mimicking actual vehicles. A fixed key assigned to vehicles can help develop strong virtual presences similar to a physical presence. It can help bring accountability to agents that would deliberately sabotage or attempt to delay communication in order to forcibly acquire the parking spot. For agents that attempt to intentionally ignore any competing agents and forcibly acquire spots, a digital identity can be recorded onto the ledger indicating proof of established communication or proof of no response. This can bring accountability to all agents that do not want to be fair. Digital identities can also indicate emergency vehicles and handicap vehicles and special permissions can be extended to such vehicles.

A digital presence can aid in building a universal presence for agents irrespective of the type of car, so conflicts between cars with varying horsepower or agents with varying resources can still be resolved in a fair manner, without any unfair advantages. Using more advanced sensor systems can help keep better records of any illegal or unfair ways that other agents might use to influence the decision. Records from the ledger indicating frequency of conflicts and traffic frequency can help plan better traffic control and optimal path strategies for autonomous vehicles.

APPENDICES

APPENDIX 1

RPSH model for blockchain

```
//Initialize RPSH state variables as [0,0,0,0]
//Variables are updated as rounds progress and the agents make their selection
Keys:
        Generate Private keys and Public Keys using RSA
        Public_exponent=65537
        Keysize=2048
        //Keys are serialized, and bytes are returned by function
Block:
        Initialize data and hash variables as 0
        Data = Transaction_input
        Previousblock = previous_block_data
        //If this is the first block, there is no previous block and the data will be updated
        //as NULL
        Previous_block_hash == Current_Block_Hash
                //If verified block is created, else ledger is updated with an error
        Compute_Hash = sha256() //python function generates hash
                //Hash for current block is updated. Hash for previous block is verified
                //If verified, hash is returned
//Function returns True after valid block creation, else false.

Sign(Message, Private Key):
        Message is taken from user and encoded in utf-8 as bytes.
        //Message encrypted using private keys withing a set maximum length
        //Hashes.SHA256() //hash object returned from SHA256 funtion

Transaction(Block):

        Data received from input is updated as Transaction_input
        //Valid function checks for the correct Hash and keys
        If Transaction.valid()
        return True
        else
        return False

Verification(Message, Sign, Public Key):

        Load serialized Public Key along with message.
        If Signature Invalid
                print(“Error”)
                return False
        Elseif
                PreviousBlockHash == Hash
                SelfPublicKey == LoadedPublicKey
                Return True

RPSH(Message, Private_Key):

        Load Serialized Private Key along with message
        If Transaction.verify() = False
                Return False
        else
                Decrypt Message with Private Key
                C1 = AgentSelfChoice //The AgentSelfChoice saves the choice selection
                //by the agent themselves
                C2 = Message

                if C1 > C2:
                    print("C1")
                    Win = 1
                     return Win
                 else:
                    Win = 2
                    print("Win")
                    return Win

//Win variable indicated which of the two agents have won

UpdateLedger(Win):
        If Win == 1
                Load txt.dat file in write mode
                Print to file(“Car 1 gets the spot”)
                Close txt.dat
        If Win == 2
                Load txt.dat file in write mode
                Print to file(“Car 2 gets the spot”)
                Close txt.dat

        Block.add(Win, Transaction)
         If Block.verify(Win, sign, public key) == True
                Block count updated.
                Valid Block Created Successfully.
        else
```

```
        return False
```

Appendix 2

Threat Assessment

Refer to figure 4.10. The threat assessment figure shows the following four threats that were blocked.

1. Delayed response by the second agent
2. Incorrect signature is used by the first agent
3. Incorrect signature is used by the second agent
4. Incorrect/Dubious blocks are added to the chain

Each agent is presented with two choices, 0 or 1. Once the first agent has decided, the selection is encrypted and sent to the second agent. Now, the second agent has 6s to make their selection and make the transaction. If the agent fails to do so, the transaction returns a Null message, and the block creation is failed.

```
function Threat1
    Message.sign(PrivateKey)
    Transaction1 = Transaction()
    Transaction2 = Transaction()
    Transaction1_input.append(PublicKey1, Message)
    Transaction1_output.append(PublicKey2, Message)
    Block.AddTransaction(Transaction2)
    t = time.time()
    while(t<6)
        Transaction2_output.append(PublicKey1, Message)
        Transaction2.verify(Message, Signature, PublicKey1)
        return True
    return False
```

If the signatures are not verified a block cannot be authenticated as the message is not from whom it was intended to be.

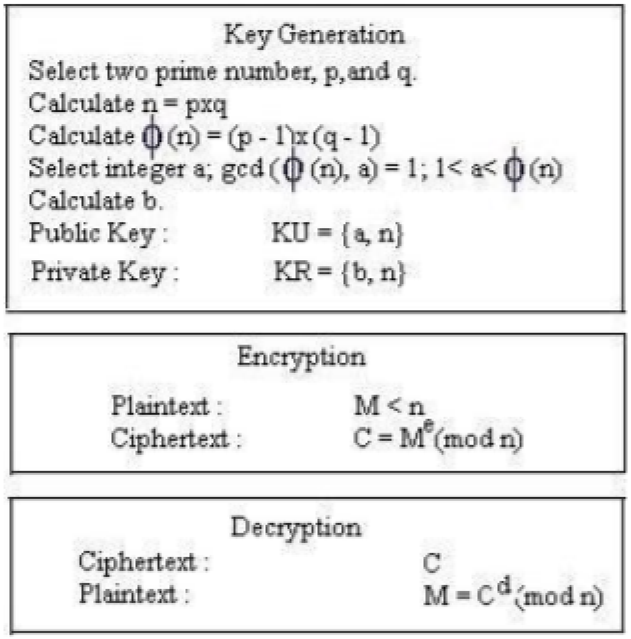
```
function verify
    load_public_key = serialization.load_pem_PublicKey(pem, Password, backend)
    if (load_public_key == PublicKey)
        return True
    else
        return False
function Threat2
    Transaction_output.append(PublicKey2, Message)
    Transaction.verify(Message, Signature, PublicKey2)
// The function returns false and the Transaction is not verified.
```

The final threat was the creation of dubious blocks. This was verified by checking the hash of the block with the previous block. Since the hashes did not match, the block was not added, and the transaction was declined. The SHA256 function is used to generate all hashes. All key generation and serialization functions were used from an open source python directory [63].

Appendix 3

Rivest-Shamir-Adleman (RSA)

RSA is one of the oldest public and private key generation algorithms, widely used in data encryption. The algorithm uses two large prime as exponents for encryption and decryption. The algorithm is shown in figure A3.1 [64].



Key Generation

Select two prime number, p,and q.
Calculate $n = pxq$
Calculate $\phi(n) = (p - 1)x(q - 1)$
Select integer a; $\gcd(\phi(n), a) = 1; 1 < a < \phi(n)$
Calculate b.
Public Key : $KU = \{a, n\}$
Private Key : $KR = \{b, n\}$

Encryption

Plaintext : $M < n$
Ciphertext : $C = M^e (\bmod n)$

Decryption

Ciphertext : C
Plaintext : $M = C^d (\bmod n)$

Figure A3.1: RSA Algorithm [64]

Appendix 4

```
from graphic import*
import pygame, time, random
from cryptography import serialization, default_backend, hashes
win = graphicwin(“simulation”, 1100, 700)
win.setbackground(color_rgb(255,255,255))

//privatekeys ex1 and ex2 are generated for encryption of selection

Generate():
    private_key_ex_1 = rsa.generate_private_key(
     public_exponent=65537,
     key_size=2048,
     backend=default_backend()
     )
    private_key_ex_2 = rsa.generate_private_key(
     public_exponent=65537,
     key_size=2048,
     backend=default_backend()
     )

try:
    public_key.verify(
      sig,
      message,
      padding.PSS(
        mgf=padding.MGF1(hashes.SHA256()),
        salt_length=padding.PSS.MAX_LENGTH
      ),
      hashes.SHA256()
    )
    return True
  except InvalidSignature:
    return False
  except:
    print("Error")
```

```
      return False

Sign(Private_Key):
        sign = private_key.sign(
        message,
        padding.PSS(
                mgf=padding.MGF1(hashes.SHA256()),
                salt_length=padding.PSS.MAX_LENGTH
        ),
        hashes.SHA256()
        )
        return sign

//Define points for each car location

//button def
p1 = Point(100, 50)
btn = Button(win, p1, 100, 30, "Start")
btn.setFill(color_rgb(200,200,225))
btn.setBorderColor(color_rgb(255,255,255))
btn.setTextColor(color_rgb(0,0,0))
btn.activate()
btn.draw(win)

//Automated control for RPSH
RPSH1 = random.randrange(0,1)
RPSH2 = random.randrange(0,1)
Transaction.input(public_key_1, 1)
Transaction.output(public_key_2, 1)

//Encrypting message with RSA
sign(message, private_key1)
sign(message, private_key2)

//Exchange of options
Transaction.input(public_key_1, message)
Transaction.output(public_key_2, message)
Transaction.sign(private_key_1)
Transaction.input(public_key_2, message)
Transaction.output(public_key_1, message)
Transaction.sign(private_key_2)

//Exchange of keys
Transaction.input(public_key_1, private_key_ex_1)
Transaction.output(public_key_2, private_key_ex_2)
Transaction.sign(private_key1)

//Decryption_RSA_algorithm

//Compare the two options

*Refer to Appendix one for RSPH algorithm
if Win == 1:
                t = Text(Point(550,150),"Car 1 gets the parking space")
                t.draw(win)
else:
                t = Text(Point(550,150),"Car 2 gets the parking space")
                t.draw(win)

#EXIT
            time.sleep(10)
            t = Text(Point(550,350),"Please Close The Window To Exit")
            window = Rectangle(Point(50,50),Point(1050,650))
            window.setFill(color_rgb(255,255,255))
            window.draw(win)
            t.draw(win)
```

REFRENCES

1. S. Nakamoto, “Bitcoin: A Peer-to-Peer Electronic Cash System,”, 2008 p. 9.

2. L. Mearian, “What is blockchain? The complete guide,” *Computerworld*, Jan. 29, 2019. https://www.computerworld.com/article/3191077/what-is-blockchain-the-complete-guide.html (accessed Sep. 15, 2020).

3. Anonymous "Blockchain? What blockchain?" *Euromoney,* 2018. Available: https://www-proquest-com./docview/2137332857.

4. L. Barnes, “Blockchain Programming.” Accessed June 19, 2020. https://Udemy.com

5. R. Huijbers, “Threading models in a messaging system,” Apr. 2017, https://rix0r.nl/essays/2013/04/17/messaging-threading-model/ (accessed Aug. 11, 2020).

6. V. G. Martínez, L. Hernández-Álvarez, and L. H. Encinas, “Analysis of the Cryptographic Tools for Blockchain and Bitcoin,” *Mathematics*, vol. 8, no. 1, Art. no. 1, Jan. 2020, doi: 10.3390/math8010131.

7. M. E. Peck, “Blockchains: How they work and why they’ll change the world,” IEEE Spectrum, vol. 54, no. 10, pp. 26–35, Oct. 2017, doi: 10.1109/MSPEC.2017.8048836.

8. H. Kim and M. Laskowski, “A Perspective on Blockchain Smart Contracts:,”, 2018 p. 6.

9. Y. Hao, Y. Li, X. Dong, L. Fang, and P. Chen, “Performance Analysis of Consensus Algorithm in Private Blockchain,” in *2018 IEEE Intelligent Vehicles Symposium (IV)*, Jun. 2018, pp. 280–285, doi: 10.1109/IVS.2018.8500557.

10. S. Yang, H. Wang, W. Li, W. Liu, and X. Fu, “CVEM: A Cross-chain Value Exchange Mechanism,” in *Proceedings of the 2018 International Conference on Cloud Computing and Internet of Things - CCIOT 2018*, Singapore, Singapore, 2018, pp. 80–85, doi: 10.1145/3291064.3291073.

11. M. Hammond, “Blockchain Interoperability & Cross-Chain Communication Series,” *Medium*, Aug. 09, 2019. https://medium.com/@mchammond/blockchain-interoperability-319bce3f9105 (accessed Sep. 04, 2020).

12. PPIO, “Cross-Chains: How Blockchains Communicate With Each Other,” *Medium*, Oct. 12, 2019. https://medium.com/@ppio/understanding-cross-chain-technology-e36b9c0cfaf3 (accessed Oct. 02, 2020).

13. L. Deng, H. Chen, J. Zeng, and L.-J. Zhang, “Research on Cross-Chain Technology Based on Sidechain and Hash-Locking,” in *Edge Computing – EDGE 2018*, Cham, 2018, pp. 144–151, doi: 10.1007/978-3-319-94340-4.

14. A. Larsen, “A Primer on Blockchain Interoperability – BlockchainCapital.” https://blockchain.capital/top-highlight-a-primer-on-blockchain-interoperability/ (accessed Oct. 06, 2020).

15. D. Puthal and S. P. Mohanty, “Proof of Authentication: IoT-Friendly Blockchains,” *IEEE Potentials*, vol. 38, no. 1, pp. 26–29, Jan. 2019, doi: 10.1109/MPOT.2018.2850541.

16. T. Xu, J. B. Wendt, and M. Potkonjak, “Security of IoT systems: Design challenges and opportunities,” in *2014 IEEE/ACM International Conference on Computer-Aided Design (ICCAD)*, Nov. 2014, pp. 417–423, doi: 10.1109/ICCAD.2014.7001385.

17. S. Vashi, J. Ram, J. Modi, S. Verma, and C. Prakash, "Internet of Things (IoT): A vision, architectural elements, and security issues," in *2017 International Conference on I-SMAC (IoT in Social, Mobile, Analytics and Cloud) (I-SMAC)*, Feb. 2017, pp. 492–496, doi: 10.1109/I-SMAC.2017.8058399.

18. C. Bloom, J. Tan, J. Ramjohn, and L. Bauer, "Self-Driving Cars and Data Collection: Privacy Perceptions of Networked Autonomous Vehicles," p. 21, In *Thirteenth Symposium on Usable Privacy and Security (SOUPS 2017), Santa Clara, CA, USA.* https://www.usenix.org/conference/soups2017/techinical-sessions/presentation/bloom.

19. M. S. Ali, K. Dolui, and F. Antonelli, "IoT data privacy via blockchains and IPFS," in *Proceedings of the Seventh International Conference on the Internet of Things - IoT '17*, Linz, Austria, 2017, pp. 1–7, doi: 10.1145/3131542.3131563.

20. C. Tselios, I. Politis, and S. Kotsopoulos, "Enhancing SDN security for IoT-related deployments through blockchain," in *2017 IEEE Conference on Network Function Virtualization and Software Defined Networks (NFV-SDN)*, Nov. 2017, pp. 303–308, doi: 10.1109/NFV-SDN.2017.8169860.

21. C. Altun, B. Tavli, and H. Yanikomeroglu, "Liberalization of Digital Twins of IoT-Enabled Home Appliances via Blockchains and Absolute Ownership Rights,"

IEEE Communications Magazine, vol. 57, no. 12, pp. 65–71, Dec. 2019, doi: 10.1109/MCOM.001.1900072.

22. A. Taherkordi and P. Herrmann, “Microservice-based Design of Smart Contracts for Blockchains in IoT Systems,” p. 5, 2018.

23. Y. Jiang, C. Wang, Y. Wang, and L. Gao, “A Cross-Chain Solution to Integrating Multiple Blockchains for IoT Data Management,” *Sensors*, vol. 19, no. 9, Art. no. 9, Jan. 2019, doi: 10.3390/s19092042.

24. M. S. Ali, M. Vecchio, M. Pincheira, K. Dolui, F. Antonelli, and M. H. Rehmani, “Applications of Blockchains in the Internet of Things: A Comprehensive Survey,” *IEEE Communications Surveys Tutorials*, vol. 21, no. 2, pp. 1676–1717, Secondquarter 2019, doi: 10.1109/COMST.2018.2886932.

25. P. Lv, L. Wang, H. Zhu, W. Deng, and L. Gu, “An IOT-Oriented Privacy-Preserving Publish/Subscribe Model Over Blockchains,” *IEEE Access*, vol. 7, pp. 41309–41314, 2019, doi: 10.1109/ACCESS.2019.2907599.

26. B. Hamdaoui, M. Alkalbani, T. Znati, and A. Rayes, “Unleashing the Power of Participatory IoT with Blockchains for Increased Safety and Situation Awareness

of Smart Cities," *IEEE Network*, vol. 34, no. 2, pp. 202–209, Mar. 2020, doi: 10.1109/MNET.001.1900253.

27. Y. Liu, Y.-P. Wang, X. Wang, Z. Xia, and J.-F. Xu, "Privacy-preserving raw data collection without a trusted authority for IoT," *Computer Networks*, vol. 148, Nov. 2018, doi: 10.1016/j.comnet.2018.11.028.

28. Y.-J. Gong *et al.*, "An Efficient Resource Allocation Scheme Using Particle Swarm Optimization," *IEEE Transactions on Evolutionary Computation*, vol. 16, no. 6, pp. 801–816, Dec. 2012, doi: 10.1109/TEVC.2012.2185052.

29. H. I. Ergin, "Efficient Resource Allocation on the Basis of Priorities," *Econometrica*, vol. 70, no. 6, pp. 2489–2497, 2002, doi: https://doi.org/10.1111/j.1468-0262.2002.00447.x.

30. A. G. L. Romme, "Unanimity Rule and Organizational Decision Making: A Simulation Model," *Organization Science*, vol. 15, no. 6, pp. 704–718, Dec. 2004, doi: 10.1287/orsc.1040.0090.

31. J. R. Nahum, B. N. Harding, and B. Kerr, "Evolution of restraint in a structured rock-paper-scissors community," *Proceedings of the National Academy of*

Sciences, vol. 108, no. Supplement_2, pp. 10831–10838, Jun. 2011, doi: 10.1073/pnas.1100296108.

32. G. Klein, “Naturalistic Decision Making,” *Hum Factors*, vol. 50, no. 3, pp. 456–460, Jun. 2008, doi: 10.1518/001872008X288385.

33. G. Keren and K. H. Teigen, “Decisions by coin toss: Inappropriate but fair,” *Judgment and Decision Making*, vol. 5, no. 2, p. 19, 2010.

34. R. Cook, G. Bird, G. Lunser, S. Huck, C. Heyes, “Automatic imitation in a strategic context: players of rock–paper–scissors imitate opponents’ gestures† | Proceedings of the Royal Society B: Biological Sciences.” https://royalsocietypublishing.org/doi/full/10.1098/rspb.2011.1024 (accessed Oct. 06, 2020).

35. Z. Wang, B. Xu, and H.-J. Zhou, “Social cycling and conditional responses in the Rock-Paper-Scissors game,” *Scientific Reports*, vol. 4, no. 1, Art. no. 1, Jul. 2014, doi: 10.1038/srep05830.

36. “Rock paper scissors,” *Wikipedia*. Aug. 03, 2020, Accessed: Aug. 18, 2020. [Online]. Available:

https://en.wikipedia.org/w/index.php?title=Rock_paper_scissors&oldid=970948263.

37. S. Dutta (2020), *Rock-Paper-Scissors-Hammer: Tie-Less Protocol in Autonomous Decision Making.* Clemson Tech ID 2021-017

38. A. Nouweland, “Rock-paper-scissors; a new and elegant proof.”, 2007, https://minerva-access.unimelb.edu.au/handle/11343/34714 (accessed Oct. 30, 2020).

39. B. Xu, H. Zhou, Z. Wang, “Cycle frequency in standard Rock-Paper-Scissors games: Evidence from experimental economics | Elsevier Enhanced Reader.”, 2013, https://reader.elsevier.com/reader/sd/pii/S0378437113005578?token=EE2C79D81F58F2B38C5387555E8F10D22DCB90343CF8A6C1B51E01C432C2A0C1BB228062DBC99BA16DF276E29A72667D (accessed Oct. 30, 2020).

40. S. Dutta, (2020), *Sealed Envelope Exchange in Autonomous Decision Making.* Clemson Tech ID 2021-016

41. A. Rahman, “A framework for decentralised trust reasoning,” Ph.D. dissertation, Department of Computer Science, Univ. College London, London https://discovery.ucl.ac.uk/id/eprint/1444477/ (accessed Oct. 29, 2020).

42. Associated Press, “Not Just VW: A Long History of Cheating Car Companies,” *CBS News*, Sep. 29, 2015.

43. S. R. H. Barrett *et al.*, “Impact of the Volkswagen emissions control defeat device on US public health,” *Environ. Res. Lett.*, vol. 10, no. 11, p. 114005, Nov. 2015, doi: 10.1088/1748-9326/10/11/114005.

44. K. Krukow, M. Nielsen, and V. Sassone, “Trust models in ubiquitous computing,” *Phil. Trans. R. Soc. A.*, vol. 366, no. 1881, pp. 3781–3793, Oct. 2008, doi: 10.1098/rsta.2008.0134.

45. Lin, T., H. Rivano, and F. Le Mouël. 2017. “A Survey of Smart Parking Solutions.” *IEEE Transactions on Intelligent Transportation Systems* 18 (12): 3229–53. https://doi.org/10.1109/TITS.2017.2685143.

46. X. Zhou and X. Tang, “Research and implementation of RSA algorithm for encryption and decryption,” in *Proceedings of 2011 6th International Forum on*

Strategic Technology, Aug. 2011, vol. 2, pp. 1118–1121, doi: 10.1109/IFOST.2011.6021216.

47. "What are automated forklifts | Washington and California," Oct. 30, 2014. https://raymondhandling.com/learn/faqs/what-are-automated-forklifts/ (accessed Nov. 06, 2020).

48. 24/7 Staff, "The Economic Impact of the Industrial Lift Truck - Supply Chain 24/7." https://www.supplychain247.com/article/the_economic_impact_of_the_industrial_lift_truck/lift_trucks (accessed Oct. 27, 2020).

49. "Supply Chain Shout Out: Forklift Power - Inbound Logistics." https://www.inboundlogistics.com/cms/article/supply-chain-shout-out-forklift-power/ (accessed Oct. 27, 2020).

50. J. Lilford, "8 Things Your Competitors Could Teach You About Site and Forklift Safety." https://www.remaxdoors.com/warehouse-management-blog/8-things-your-competitors-could-teach-you-about-site-and-forklift-safety (accessed Nov. 06, 2020).

51. T. Larsson, T. Horberry, T. Brennan, J. Lambert, and I. Johnston, “A Guidebook of Industrial Traffic Management & Forklift Safety,”, 2003, p. 72.

52. V. L. Group, “Ten Tips for Avoiding California Lane Change Accidents | Vititoe Law Group | Westlake Village,” *Vititoe Law Group*, Aug. 30, 2017. https://www.vititoelawgroup.com/2017/08/30/ten_tips_for_avoiding_california_lane_change_accidents/ (accessed Nov. 06, 2020).

53. D. Jeong and K. Lee, “Distributed Communication and Localization Algorithms for Homogeneous Robotic Swarm,” in *Distributed Autonomous Robotic Systems*, vol. 112, N.-Y. Chong and Y.-J. Cho, Eds. Tokyo: Springer Japan, 2016, pp. 405–418.

54. “How artificial intelligence could negotiate better deals for humans | Science | AAAS.” https://www.sciencemag.org/news/2017/09/how-artificial-intelligence-could-negotiate-better-deals-humans (accessed Nov. 06, 2020).

55. “1.2 A Parallel Machine Model.” https://www.mcs.anl.gov/~itf/dbpp/text/node8.html (accessed Nov. 06, 2020).

56. A. Agnetis, P. B. Mirchandani, D. Pacciarelli, and A. Pacifici, “Scheduling Problems with Two Competing Agents,” *Operations Research*, p. 15, 2004.

57. M. Elhenawy, A. Elbery, A. Hassan, and H. Rakha, “An Intersection Game-Theory-Based Traffic Control Algorithm in a Connected Vehicle Environment,” Sep. 2015, pp. 343–347, doi: 10.1109/ITSC.2015.65.

58. C. Liu, C. Lin, S. Shiraishi, and M. Tomizuka, “Distributed Conflict Resolution for Connected Autonomous Vehicles,” *IEEE Transactions on Intelligent Vehicles*, vol. 3, no. 1, pp. 18–29, Mar. 2018, doi: 10.1109/TIV.2017.2788209.

59. Yoshioka, L. R. *et al.*, “Bus Corridor Operational Improvement with Intelligent Transportation System based on Autonomous Guidance and Precision Docking,” *INTERNATIONAL JOURNAL OF SYSTEMS APPLICATIONS, ENGINEERING & DEVELOPMENT*, vol. 8, pp. 116–123, 2014.

60. “What is Fog Computing? Definition and FAQs | OmniSci.” https://www.omnisci.com/technical-glossary/fog-computing (accessed Nov. 06, 2020).

61. F. Bonomi, R. Milito, J. Zhu, and S. Addepalli, “Fog computing and its role in the internet of things,” in *Proceedings of the first edition of the MCC workshop on Mobile cloud computing*, New York, NY, USA, Aug. 2012, pp. 13–16, doi: 10.1145/2342509.2342513.

62. “Automated Driving Systems: A Vision for Safety,”, National Highway Traffic Safety Administration, 2016, p. 36. Accessed on Nov 06, 2020. [Online]. Available: https://www.nhtsa.gov/sites/nhtsa.dot.gov/files/documents/13069a-ads2.0_090617_v9a_tag.pdf.

63. “Welcome to pyca/cryptography — Cryptography 3.3.dev1 documentation.” https://cryptography.io/en/latest/ (accessed Oct. 30, 2020).

64. M. Jhuria, S. Singh, and R. Nigoti, “A Survey of Cryptographic Algorithms for Cloud Computing,” *International Journal of Emerging Technologies in Computational and Applied Sciences*, May 2013.

65. "Car Vectors by Vecteezy" https://www.vecteezy.com/free-vector/car (accessed Oct. 30, 2020).