

**SECURE DATA TRANSFER THROUGH INTER-PLANETARY FILE SYSTEM (IPFS)
AND BLOCKCHAIN-EMBEDDED SMART CONTRACT IN BUILDING
CONSTRUCTION**

BY

**ALENOGHENA, Ilunuamie Benjamin
MTECH/SICT/2018/9194**

**DEPARTMENT OF COMPUTER SCIENCE,
FEDERAL UNIVERSITY OF TECHNOLOGY MINNA.**

April, 2023

**SECURE DATA TRANSFER THROUGH INTER-PLANETARY FILE SYSTEM (IPFS)
AND BLOCKCHAIN-EMBEDDED SMART CONTRACT IN BUILDING
CONSTRUCTION**

BY

**ALENOGHENA, Ilunuamie Benjamin
MTECH/SICT/2018/9194**

**DEPARTMENT OF COMPUTER SCIENCE,
FEDERAL UNIVERSITY OF TECHNOLOGY MINNA.**

**A THESIS SUBMITTED TO THE POSTGRADUATE SCHOOL, FEDERAL
UNIVERSITY OF TECHNOLOGY, MINNA, NIGERIA IN PARTIAL FULFILLMENT
OF THE REQUIREMENTS FOR THE AWARD OF THE DEGREE OF MASTER OF
TECHNOLOGY IN DEPARTMENT OF COMPUTER SCIENCE, FEDERAL
UNIVERSITY OF TECHNOLOGY MINNA**

ABSTRACT

The introduction of Smart contract-embedded blockchain systems incorporated with Building Information Model (BIM) in construction projects brought about the transformation of public and corporate management. The system, although it has contributed to infrastructure development, still faces vulnerabilities such as eavesdropping, data tampering, and possible attacks, requiring a secure communication channel. In this research a secure system for transferring data through the Inter-Planetary File System (IPFS), BIM and a blockchain-embedded smart contract is developed. The system used a symmetric, multi-level authority and encryption to ensure the confidentiality, integrity, and availability of the transferred data. The use of IPFS and a smart contract allows for decentralized and distributed storage and transfer of data, while the use of multi-level authority and encryption provides additional security measures to protect against eavesdropping, data tampering and theft. The model developed was tested alongside a transposition cipher and RSA cipher against known cryptosystem attacks, dictionary attacks, and brute force attacks. The results showed that the proposed model after showing 0%-character placement accuracy for dictionary attack, known cryptosystem attack and brute force attack, shares the same 0% with RSA. The resulting system has the potential to revolutionize the way data is transferred and stored, with applications in various industries and contexts because it requires less technical know-how and capability to deploy when compared to modern cryptographic techniques.

TABLE OF CONTENTS

Content

Title Page	i
Declaration	ii
Certification	iii
Acknowledgement	iv
Abstract	v
Table of contents	vi
List of Tables	x
List of Figures	xi
Abbreviations	xii
CHAPTER ONE	1
1.0 INTRODUCTION	1
1.1 Background of the Study	1
1.2 Statement of The Research Problem	5
1.3 Aim and Objectives of The Study	6
1.4 Scope of the Study	7
1.5 Significance of The Study	7
CHAPTER TWO	8
2.0 LITERATURE REVIEW	8
2.1 Blockchain Technology	8

2.1.1	Structure of Blockchain	9
2.1.2	Types of Blockchain Technology	11
2.1.3	Advantages of Blockchain Technology	12
2.1.4	Drawbacks of Blockchain Technology	13
2.1.5	Blockchain Applications	14
2.2	Smart Contracts	17
2.2.1	Categories of Smart Contracts	19
2.2.2	Life Cycle of Smart Contracts	20
2.2.3	Application of Smart contract	22
2.3	Construction Industry	22
2.3.1	Smart Contract in Architecture, Engineering, and Construction (AEC) Industry	23
2.3.2	Smart contract in Construction Projects	25
2.4	Building Information Modelling (BIM)	26
2.5	Distributed File Systems (DFS)	30
2.5.1	Inter-Planetary File System (IPFS)	33
2.5.2	Identity layer in DFS	35
2.5.3	Network Layer	36
2.6	Data Encryption	36
2.6.1	Classical Ciphers	37

2.6.2	Modern Ciphers	38
2.6.3	Symmetric Key Encryption	39
2.6.4	Common Symmetric Encryption Algorithms	40
2.6.5	Asymmetric Key Encryption	43
2.6.6	Common Asymmetric Encryption Algorithms	44
2.6.7	Cryptographic Benchmarks	47
2.7	Related Studies	50
2.7.1	Summary of Related work	56
CHAPTER THREE		60
3.0	METHODOLOGY	60
3.1	System Design	60
3.1.1	Encryption Flow Diagram	61
3.1.2	Algorithm and Mathematical Model	63
3.1.2.1	Algorithm	63
3.1.2.2	Mathematical Model	63
3.4.3.3	Simulation	64
CHAPTER FOUR		67
4.0	RESULTS AND DISCUSSION	67
4.1	Evaluation Criteria	67

4.2	Implementation and Results	67
4.2.1	Discussion of Results	70
4.2.2	Integrity, Non-Repudiation and Authenticity	71
CHAPTER FIVE		72
5.0	CONCLUSION AND RECOMMENDATIONS	72
5.1	Conclusions	72
5.2	Recommendations	72
5.3	Contribution to Knowledge	73
REFERENCE		74
APPENDIX		81

LIST OF TABLES

Table	Page
2.1 Summary of Related work	54
3.1 Character Placement Accuracy	66
3.2 Word Placement Accuracy	67

LIST OF FIGURES

Figure	Page
2.1 Blockchain Structure (Lim et al., 2021)	10
2.2 Image of 3D Model of a BIM	29
2.3 IPFS Structural Stack (Alwis, 2020)	33
2.4 Merkle DAG in IPFS (Huang et al., 2020)	34
2.5 Basic Classification of Cryptography (Alenezi et al., 2020)	45
3.1 System flow diagram	58
3.2 Encryption Flow	60
3.3 Random Key Generation	62
3.4 Polymorphic Encryption	63
3.5 Transposition Encryption	63
3.6 Final Cipher	64
4.1 Graphical Representation of Result	68

CHAPTER ONE

1.0

INTRODUCTION

1.1 Background of the Study

The construction industry is going through constant changes and is managing the transition from the analog to the digital era. A prominent characteristic of large-scale structural engineering and construction projects in current years has been to design and build state-of-the-art structures and integrated complex systems never attempted before. The complicated nature of these designs has significantly led to the essential need for very high-performance capabilities not earlier considered (Hargaden *et al.*, 2019). With the rapid advancement in technology and construction projects becoming more and more complex, construction project managers must quickly learn to adapt to this ever-changing environment by approaching new practices and technologies. One such technology that has gained traction in recent years is blockchain and smart contracts.

The concepts of bitcoin and blockchain were first proposed in 2008 by pseudonym Satoshi Nakamoto, who described how cryptology and an open distributed ledger can be combined into a digital currency application (Xu *et al.*, 2019). Blockchain are tamper-evident and tamper-resistant digital ledgers implemented in a distributed fashion (i.e., without a central repository) and usually without a central authority (banks, companies, or government) (Yaga *et al.*, 2019). Blockchain is a finance-oriented extremely ingenious distributed shared ledger and peer-to-peer value transfer technology, that establishes trust between unknown stakeholders and automated payments. Blockchains also reformed the finance and supply chain industry by shortening the time needed to complete time-consuming processes and removing nearly all intermediaries (Guo *et al.*, 2021). At

first, the extremely high volatility of bitcoin and the attitudes of many countries toward its complexity restrained its development somewhat, but the advantages of blockchain which is bitcoin's underlying technology attracted increasing attention. Some of the advantages of blockchain include its distributed ledger, decentralization, information transparency, tamper-proof construction, and openness. The evolution of blockchain has been a progressive process. Blockchain is currently delimited to Blockchain 1.0, 2.0, and 3.0 based on their applications (Xu *et al.*, 2019).

Blockchain 1.0 is related to virtual currencies, such as bitcoin used commercially for small-value payments, foreign exchange, gambling, and money laundering. Blockchain 2.0 concentrated on applications in other areas of finance, where it is mainly used in secure trading, supply chain finance, banking instruments, payment clearing, anti-counterfeiting, establishing credit systems, and mutual insurance. The greatest contribution of Blockchain 2.0 which was the Ethereum project was the idea of using smart contracts to disrupt the traditional currency and payment systems (Xu *et al.*, 2019). Blockchain 3.0 is described as the application of blockchain in areas other than currency and finance, such as in government, health, science, culture, and the arts. Blockchain 3.0 aims to popularize the technology, and it focuses on the regulation and governance of its decentralization in society. Blockchain 3.0 envisions a more advanced form of “smart contracts” to establish a distributed organizational unit, be subject to its laws, and operate with a high degree of autonomy (Xu *et al.*, 2019).

A smart contract is an executable code that runs automatically on the blockchain by consensus nodes without any trusted third party. A smart contract can perform specified operations once pre-defined rules have been met (Wang *et al.*, 2020). For example, using a smart contract, Alice will only receive a certain amount of currency from Bob, if a particular set of conditions are met by

Bob. Ethereum is one of the most popular decentralized platforms for smart contract applications (Chang *et al.*, 2019). Users may design their contracts by defining data structures and functions in each contract and subsequently deploy the contract on the blockchain. Contracts can communicate with each other through their Ethereum addresses and application programming interfaces (APIs) (Ahmadisheykhsarmast & Sonmez, 2020).

Blockchain 3.0 is now used in the industries such as healthcare, education, government, charities, real estate, insurance, and banking (Guo *et al.*, 2021). Blockchain's characteristics of decentralization provide zero downtime, ensure tamper-proof data and non-repudiation with immutability, implement security with cryptography to establish trust between participating parties, and utilize consensus algorithms for data integrity, verification, and scalability on private and permissioned blockchains (Guo *et al.*, 2021).

A file system is a subsystem of an operating system whose purpose is to organize, retrieve, store and allow sharing of data files. Information technology evolves in multi-decade cycles of expansion, consolidation and decentralization. Periods of expansion follow the introduction of a new open platform that reduces the production costs of technology as it becomes a shared standard (Suralkar *et al.*, 2013).

A Distributed File System (DFS) is a distributed implementation of the classical time-sharing model of a file system, where multiple users who are geographically dispersed share files and storage resources. Accordingly, the file service activity in a distributed system has to be carried out across the network, and instead of a single centralized data repository, there are multiple and independent storage devices (Suralkar *et al.*, 2013).

The DFS server allows the client users to share files and store data just as if they are storing the information locally. However, the servers have full control over the data and give access control to the clients. Traditional Peer-to-Peer (P2P) distributed file systems have inevitable drawbacks such as instability, lacking auditing and incentive mechanisms. Inter-Planetary File System (IPFS), is one of the representative DFSs which integrate with blockchain technologies (Huang *et al.*, 2020).

IPFS is a peer-to-peer distributed file system for storing and accessing files, websites, applications, and data that adopt Filecoin which is a blockchain-based digital payment system as its incentive mechanism (Huang *et al.*, 2020).

Data encryption is the art of securing messages by converting them to hidden texts, whereas the inverse process of retrieving original texts from hidden texts is called decryption. Encryption/decryption is made possible with the help of some keys. Every encryption algorithm aims to make the decryption process as difficult as possible without the help of the key used in encryption. The types of cryptographic techniques we have include; Symmetric key encryption, Asymmetric key encryption and Hashing (Alenezi *et al.*, 2020).

The construction industry plays a significant role in the infrastructural and industrial development of Nigeria. However, this sector is too dependent on paper-based modes of communication, a process that wastes time, money and is prone to errors and omissions and often leads to cost overruns, delays and conflicts among the project team (Aina, 2015).

When contracts are signed between a contractor and a client, various resources required for the contract are provided by the procuring entity, covering the budgetary funds and relevant staff for the effective implementation of the contract (Zheng *et al.*, 2020).

Payments are the lifeblood of construction projects, in practice steady fund flows are rare, leading to objectionable consequences including delays, increased costs, reduced performance, disputes, and bankruptcies which could threaten the success of the projects. Payment problems are identified as one of the top dispute causes for construction projects (Ahmadisheykhsarmast & Sonmez, 2020).

1.2 Statement of The Research Problem

Various governments across the world economy have lost a significant amount due to weak contract management practices. Contract management covers the title of financial optimization, helping the business or government to protect itself from the renewal of some unwanted services. Furthermore, the operational efficiency of any business can only be achieved if a proper contract management system is applied for a better outcome (Z. Zheng et al., 2020).

When the contract is signed, various resources required for the contract are provided by the procuring entity, covering the budgetary funds and relevant staff for the effective implication of the contract. In contrast, conventional contracts need to be completed by a trusted third party in a centralized manner consequently resulting in long execution time and extra cost (Z. Zheng et al., 2020). furthermore, as a centralized system, their exit risk of content tampering, signature forger, and data loss. The introduction of Smart contract embedded blockchain systems brought about a solution to solving construction industry problems (Ahmadisheykhsarmast & Sonmez, 2020). Before this research, there exist a Blockchain-enabled smart contract that automates the conditioning of construction payments on the progress assessments making use of Inter-Planetary File System (IPFS) for off-chain storage of product flow (Hamledari and Fischer 2021). However, the IPFS cannot guarantee the confidentiality of the data shared through the system (Alwis, 2020).

Secure communication and data confidentiality (protecting data against unauthorized access or theft) are the most important factors when it comes to data protection. they can be achieved with the help of cryptography through data encryption and decryption.(Alenezi *et al.*, 2020). The need for a secure communication channel cannot be over-emphasized as the channel can be vulnerable to attacks such as eavesdropping and man-in-the-middle (MitM) attacks (Mallik *et al.*, 2019). A culprit positions himself between the supervisor and the IPFS; either to listen stealthily or to imitate one of the parties, alter the data either for or against the contractor or the client making it show up as though an ordinary trade of information is in progress. To that effect, there is a need for the encryption of the channel of transfer of data from the supervisor who inputs the progress of the construction to the IPFS.

1.3 Aim and Objectives of The Study

This research aims to develop a secure data transfer through inter-planetary file system (IPFS) and blockchain-embedded smart contract in building construction.

This will be achieved by the following objectives:

- i. To formulate an algorithm for a secured end-to-end transfer of data Using Multi-level Authority and Encryption
- ii. To implement the designed algorithm in i by leveraging the advanced features of the Python programming language.
- iii. To evaluate the proposed encryption system vis-à-vis state-of-the-art encryption systems, leveraging the powerful evaluation capabilities of CrypTool 2.1.

1.4 Scope of the Study

This study covers the area of construction contracts that studies the integration of Inter-Planetary File System (IPFS) and Building Information Modelling (BIM) for the automation of payments of construction contracts. This study is however limited to the encryption of the channel where data is being transferred in the system.

1.5 Significance of The Study

Non-repudiation of data sent from one end to another is very necessary as the sender of information is provided with proof of delivery and the recipient is provided with proof of the sender's identity. This puts the communication channel at risk of eavesdropping attacks. This study will bring about the encryption of information as they are transferred between Inter-Planetary File System (IPFS), Building Information Modelling (BIM) and the blockchain-embedded smart contract will provide additional security to protect against eavesdropping, data tampering and theft.

CHAPTER TWO

2.0 LITERATURE REVIEW

2.1 Blockchain Technology

In 2008, Satoshi Nakamoto published his study “Bitcoin: A Peer-to-Peer Electronic Cash System”, in which a peer-to-peer (P2P) electronic cash system was proposed (Nakamoto, 2008). This system was the answer to the financial crisis of 2008 that showed that banks and centralized financial institutions breached the trust of the people who deposit their money in them, by lending it while keeping very little in reserve. This technology called blockchain used Bitcoin as its first use case. Bitcoin has gained trust among cryptocurrency users as it has proven, indisputably, the security of its technology. This explains why it is the most used cryptocurrency in the world (Valdeolmillos & Mezquita, 2009). This system allowed payments to be directly initiated by one party and sent to the other without a third-party financial institution. The Bitcoin platform comprises a series of cryptographic protocols that transform how transactions are made. Thus, this platform has brought the financial system one step closer to a truly democratic economy constructed by the community (Valdeolmillos & Mezquita, 2009).

Blockchain is a technology underlying Bitcoin and other cryptocurrencies, maintained by a decentralized computer network. Blockchain technology is considered an open ledger where all online transactions are recorded and everyone is allowed to connect, send or verify transactions (Nguyen, 2016). In other words, Blockchain is a digitized system of accounting records that records in detail all transactions according to a mathematical set of rules to prevent illegal interference.

Research on the impacts of cryptocurrencies, decentralized ledger, and Blockchain has shown that they are potentially powerful tools to minimize costs and bring major changes to the financial field in the long run.

2.1.1 Structure of Blockchain

Blockchain is a distributed ledger composed of a series of data blocks that are linked through cryptographic methods (Zheng *et al.*, 2021). Each block records a batch of network transaction information; the structure of a block is shown in Figure. 2.1. A block consists of a block header and a block body. The block header holds information used to connect to the previous block and information used for verification, including the version number, the hash value of the previous block, the timestamp of the current block writing time, the nonce and difficulty target used to prove the difficulty of the workload, and a total hash of Merkle tree root for verifying the block body transaction. The block body contains transaction information and the Merkle trees of all the transaction information (Lim *et al.*, 2021). In Figure 2.1, “T” is used to represent “transaction”.

In a blockchain network, any two nodes can conduct transactions, and each transaction is broadcast by a single node to all nodes on the entire network (Frizzo-Barker *et al.*, 2020). Transaction information is linked in the blockchain when all nodes confirm that the records are correct, and this process relies on the consensus mechanism of the blockchain. The distributed structure enables each node to record all transaction information, and each node updates and stores all the information of the entire network in a real-time network (Feng *et al.*, 2020). Blockchain technology involves three technological innovations, namely, cryptography, consensus mechanisms, and smart contracts.

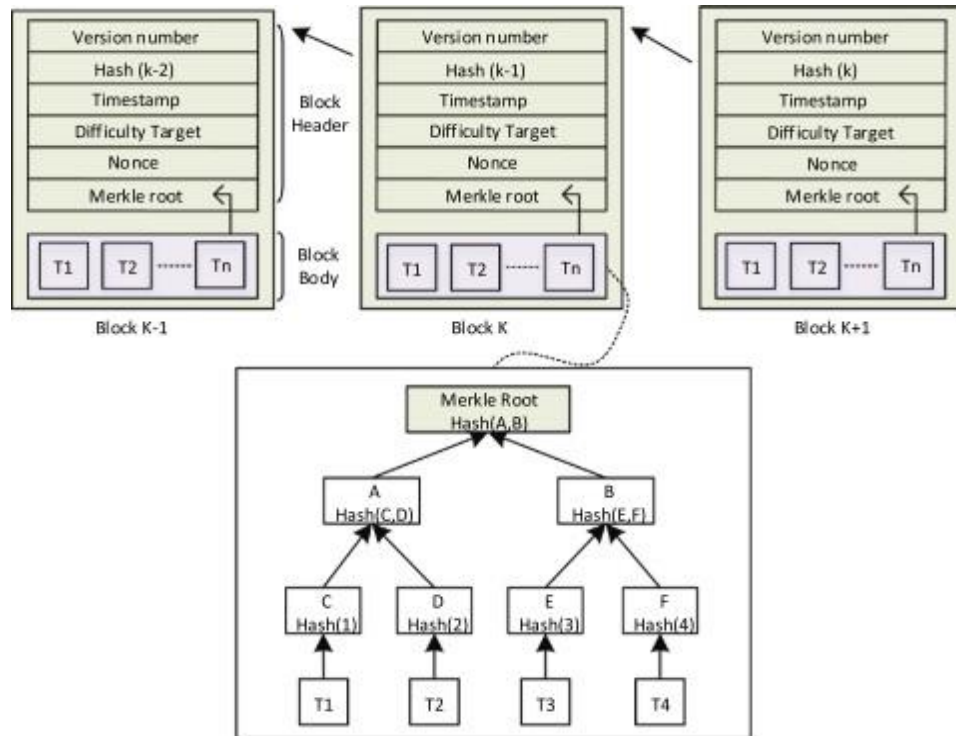


Figure 2.1: Blockchain Structure (Lim et al., 2021)

1. Cryptography technologies, including the hash function and public-key cryptography, are the basic technologies used to ensure the security of the blockchain system and are used in data structures, verification methods, communication protocols, and information storage. The hash function guarantees the integrity, authenticity, and immutability of the distributed ledger data through the hash value and hash pointer, and it can convert the input data into a fixed-length digest through a hash algorithm, whose process is irreversible (Lim et al. 2021). Public key cryptography is an asymmetric encryption algorithm that is used to provide identity verification to the blockchain network. Public key cryptography effectively solves the problem of key distribution and exchange in network communications, which ensures the security of information transmission (Ali *et al.*, 2019).

2. The consensus mechanism proves the ownership and accuracy of the bookkeeping nodes through a consensus algorithm that solves consistency problems. This mechanism establishes trust between different nodes in the blockchain system and guarantees that each transaction remains consistent for all nodes. With the development of blockchain technology, many consensus algorithms have been developed, such as proof of work (POW), proof of stake (POS), and practical byzantine fault tolerance (PBFT) (Choi *et al.*, 2020).
3. A smart contract is a set of promises defined in digital form; the contract content is fixed in the blockchain in the form of code, and eventually, an automatically executed script is generated. Each transaction is processed by a smart contract, and the corresponding contract terms can be executed automatically once a predefined condition is triggered. This process does not require a third party, which has an enormous impact on the design of business models. A smart contract can be used to model various types of businesses, organizational behaviours, and rules in the real world and affects interactions among various entities, including the transfer of asset ownership, payments of digital assets, and currency transactions (Lim *et al.*, 2021).

2.1.2 Types of Blockchain Technology

Public Blockchain (permissionless): This retains the idea that information can be accessed by anyone in the world. It requires a consensus mechanism to write or block information to the public blockchain. Bitcoin is an example of a decentralized public blockchain (Valdeolmillos & Mezquita, 2009).

Consortium Blockchain: It is partly private. This functions under a group's management rather than a single entity. Blockchain data reading may be allowed to be open or limited to a number of participants (Raghavendra & Kiran, 2020).

Private Blockchain: Only allows a predefined user group to write any blockchain data. Public or some restricted users can read the data (Raghavendra & Kiran, 2020).

2.1.3 Advantages of Blockchain Technology

The use of a decentralized ledger and the automation of Blockchain has formed a technological model for a payment infrastructure characterized by low costs and transparency, which will have major impacts on the global financial market. The advantage of blockchain as stated by (Nguyen, 2016) include:

1. Blockchain promotes smart contracts, which increases the efficiency of transactions and payments in the stock market. By providing faster and cheaper financial services, Blockchain technology can be a powerful tool that puts finance in harmony with the world's dynamic and changing landscape.
2. Fees for foreign exchange transactions, remittances, credit card transactions, and other products can be reduced substantially. Specifically, it is estimated that \$16 billion will be saved annually, equally one-third of transaction fees. Capital requirements for banks can be reduced by \$120 billion. Costs for remittances will fall by approximately 1% compared to the global average of the "traditional channels" of 7.7% (Ozturan *et al.*, 2019). Recently, Blockchain intermediaries have been providing excellent Bitcoin transaction services in those countries like Kenya and the Philippines.

3. The fact that information is automatically recorded and monitored during the transaction process makes the destination and purpose of the money transferred more transparent, which supports the fight against financial crimes such as money laundering.
4. The digitization and verification of records not only reduce necessary procedures and save paper but also ease the follow-up process of trade agreements. It also ensures that financial transactions are better protected while banks and regulatory bodies would be able to keep track of customers more easily.
5. Last but not least, Blockchain helps boost the speed and efficiency of execution, optimizing the time for transactions to be completed, which is currently up to 3 working days. Blockchain technology will help link networks of recordkeeping, reduce transaction costs and enhance access to the financial market. It can potentially impose a widespread impact on the financial market in terms of banking payments, security trading, web security, trade reporting, interest rate, etc. It is forecasted that 2 million bank jobs would be cut off once Blockchain technology is applied in the next decade.

2.1.4 Drawbacks of Blockchain Technology

Although Blockchain is a solution to improve the security of the data in a traditional system, there are several risks associated with blockchain technology they include:

1. It is being targeted by new and specific types of cyberattacks. Distributed Denial Of Services (DDOS), eclipse, and Sybil attacks are the most common attack faced by Blockchain Technology (Valdeolmillos & Mezquita, 2009).
2. It limits the competitiveness between banks to improve their system as blockchain networks will be shared among all banks that participated in the system (Nguyen, 2016).

3. Some of the blockchain platforms make use of Proof-of-Work (PoW) consensus algorithm, meaning that it wastes a lot of energy when adding new blocks to its blockchain (Nguyen, 2016).
4. The development of the new system would require project parties to learn the needed skill to be able to operate within the system. As a result, the construction industry may be slow to adopt due to its conventional nature (Owusu *et al.*, 2020).
5. For the banking industry, technological interruptions like blockchain require more time and effort in research and application. Besides, banks also face payment risks and effects on financial stability due to possible loss of balance in the financial system caused by high automation.

2.1.5 Blockchain Applications

The adoption of blockchain was pioneered through Financial Technology (FinTech) and has now expanded to other industrial sectors, particularly with building construction management applications. Some practical use cases in FinTech have been reported and widely discussed, e.g., Blockchain in Digital Currencies (Bitcoin, Ethereum, etc.). However, there are also many other applications, samples of which will be described as follows.

1. **Automated Paperwork Processing:** The problem of the long trail of paperwork associated with international container transportation has been explored (Hargaden *et al.*, 2019). For example, shipping refrigerated goods from East Africa to Europe requires stamps and approvals from around 30 people and organizations that must interact with each other on over 200 occasions. At each interaction, documents are potentially subjected to fraud, and overall, the cost of trade-related paperwork processing is estimated to be

between 15 and 50 per cent of the costs of the physical transport (Hargaden *et al.*, 2019). To tackle these inefficiencies and digitize paper records, IBM and Maersk jointly investigated the issue in 2015. They eventually settled for a permission blockchain solution called “TradeLens” as a means to connect the vast global network of shippers, carriers, ports, and customs (Hargaden *et al.*, 2019). A series of pilot implementations in 2017 was successful; in these pilots, every relevant document or approval was shadowed on the blockchain, meaning the legacy IT systems were not replaced but augmented. The problems associated with extensive paperwork in supply chains are not limited to this specific use case but hamper all kinds of trade flows, suggesting that this is a promising area for blockchain.

2. **Counterfeit Identification:** Blockchain promises to increase the security of high-value items that rely on paper certificates, which can get lost or tampered with. The authenticity of a diamond’s certificate (and whether the diamond was stolen) can sometimes be difficult to determine. De Beers deploys its blockchain to track its diamond supply chain and to fight against counterfeit diamonds and conflict/blood diamonds (Hargaden *et al.*, 2019). The start-up ‘Everledger’ takes an alternative approach and records 40 data points that uniquely identify a diamond. Using these publicly available records on the blockchain, a potential buyer can determine if the seller is the actual owner of the diamond and can also make sure they are not buying a diamond from a conflict region (i.e. “blood” diamond) (Underwood, 2016). Everledger plans to extend this fraud detection system into a provenance platform for many high-value items (Underwood, 2016). In the medical sector, blockchain could also improve patient safety by lowering the risk of counterfeit drugs through establishing supply chain transparency, from manufacturers through wholesale and pharmacies to individual patients. By using barcodes that link to a blockchain system,

patients could potentially be empowered to check whether they received the correct drugs (Turk & Klinc, 2017). Overall, blockchain is considered to make it much more difficult to tamper with products or to channel products of illegal origin.

3. **Traceability:** Using current enterprise resource planning systems, it is difficult for retailers and public health officials to identify the source of foodborne disease outbreaks, meaning it can take weeks to track down the origin of contaminated ingredients and to which stores they were delivered (Tian, 2016). To facilitate point-of-origin tracking for food items, Walmart partnered with IBM to augment the supply chain partners' existing IT systems through a superordinate blockchain ledger tracking the movements of food items. This shared forum is a substantial improvement over Walmart's earlier trials involving barcodes or auto-ID technology (Tian, 2016). In these new systems, data such as farm origin, batch numbers, factory data, expiration dates, and shipping details were written on the blockchain and instantly became available to all network members. Now, if a foodborne disease outbreak were to occur within one of the blockchain-augmented supply chains, these data enable tracking down the origin in seconds. Furthermore, Walmart believes blockchain could also reduce food waste as the newly available data are used as a parameter for order quantity optimization (Hargaden *et al.*, 2019). Traceability issues are also evident in other industries. The issue of conflict minerals that are mined and sold, under the control of armed groups, to finance war and violence was discussed by Intel in a white paper.
4. **Transaction Efficiency:** The nature of a blockchain system eliminates the central authority needed to validate transactions, thus enhancing many computational efficiencies. For example, transaction costs following a sale can occur; however, these can be eliminated on a blockchain system as the network validates the transaction (Subramanian, 2018). At the

same time, the payment between buyer and seller is recorded on a shared secure ledger. Considering a conventional e-commerce store, when a customer clicks the checkout button, systems such as payment and credit-card networks charge a fee. Even with this fee, fraudulent transactions are not always eliminated. In a decentralized online marketplace, users transact with each other securely and directly, and the network of nodes validates and records each transaction. Therefore, blockchain can also enable buyers and sellers to transact directly and without manipulation by intermediaries.

2.2 Smart Contracts

Smart contracts are computer protocols that verify, simplify, and enforce the performance or negotiation of a contract or eliminate the unforeseen clauses in the contract (Shojaei *et al.*, 2020). Smart contracts comprise several transactions taking place between verified parties; they usually vary widely in scale and complexity and are executed by computer codes (Zheng *et al.*, 2020). They are not only formed online but their very performance is enabled and guaranteed by a network of decentralized, co-operating computer nodes, known as blockchains (Mik, 2017). Originally, smart contracts were contemplated within a limited range of transactions, predominantly financial instruments. Progressively, however, the surrounding narrative has become broader, implying that all contracts can be made smart or that many different obligations can be enforced by code. What started as a niche phenomenon in such areas as financial derivatives and prediction markets, is now poised to change the entire legal landscape and “revolutionize” commerce. Nevertheless, transition from human-language contracts to technology-based system contracts creates new disorganizations. These issues arise from the following features of smart contracts:

1. Automation, which requires all agreements to be formed by fully-defined terms

2. Decentralization, which requires the verification of job performance by third parties; and
3. Anonymity reduces the dependency of the contract on the commercial context in which it is being used.

As a result, a semi-automated system is a likely outcome in the short and medium-term (Shojaei *et al.*, 2020). To encode the parameters of a contract by a programmer, smart contracts commonly omit the necessity of administrative staff and expenses. By automating the execution of the contract, it can be interpreted that smart contracts are legal self-help agreements outside the obligation of the law. Subsequently, computer codes are used to write them, which are not legal languages of contract law.

Smart contracts can streamline the contracting process, reduce transaction costs by eliminating intermediaries and, most importantly, simplify enforcement by obviating the need to seek protection from traditional legal institutions, such as courts (Mik, 2017). The theories underpinning smart contracts and blockchains combine multiple, interrelated threads all of which reflect an indiscriminate, if not irrational, fascination with certain technical characteristics of blockchains. They also reflect a surprising lack of trust in humans. As the latter is perceived as inherently biased and unreliable, things should be left to computers. Humans, especially bankers and judges, are fallible and not trustworthy (Mik, 2017). Computers, on the other hand, are objective, infallible and trustworthy. The very idea of smart contracts is thus inextricably tied to the elimination of human judgment, the reduction of dependence on financial intermediaries, and, in many instances, a detachment from the legal system (Mik, 2017). In other, more commercially-oriented contexts, smart contracts can simply be seen as part of the broader trend to use technology to ensure a consistent application of legal rules and agreements.

A smart contract can perform specified operations once pre-defined rules have been met (Wang *et al.*, 2020). For instance, using a smart contract, Bob cloud receives X currency units from Alice, if he sends the correct calculation results to Alice. In the smart contract system, each contract has a unique address and cannot be changed after being deployed into the blockchain. When users execute a contract, they only need to send the transaction to the address stated on that contract. Then, every active consensus node will execute this transaction in the smart contract system to get a consensual result. At present, researchers are trying to use smart contracts to solve various problems in a variety of areas, such as Insurance, Medical Care, e-Voting, Cloud Computing and, IoT (Wang *et al.*, 2020).

The determinism of a smart contract's actions is usually left to the developer. Thus, automated actions must be ensured and executed as planned and the results of these actions leave the data in a consistent state, regardless of the node(s) they are executed on. Smart Contract actions must achieve the same result each time the Smart contract is executed depending on the specific use case to be implemented (Sofia et al., 2019).

2.2.1 Categories of Smart Contracts

1. **Static standard output:** Static smart contracts do not call other smart contracts, do not reside on human interaction, take place in one step, and never change their predefined number of actions. Static smart contracts perform primitive math operations such as addition, subtraction, multiplication, and division. Other smart contracts can call, retrieve, and consume the results of their operation. All smart contracts receive parameters to perform actions and are somehow dynamic (Sofia et al., 2019).

2. **Dynamic non-standard output:** Dynamic smart contracts embed various rules that allow them to perform different actions. Examples of dynamic smart contracts include functions that monitor certain conditions and trigger intended actions. For example, when a dynamic smart contract monitors electricity consumption and temperatures logged on the BC of an energy-smart building. The dynamic smart contract includes thresholds for heating and consumption measurements to adjust temperatures in an eco-friendly way designed to avoid excessive electricity consumption and cost (Daniel & Guida, 2019).
3. **Artificial Intelligence (AI) Oracle driven:** AI oracle-driven smart contracts apply e-Government 3.0 to law applications. For example, laws for inheritance can change and notaries or other public servants in an oversight role must be formally informed regarding issues such as legacy transfer. An AI oracle accesses information from a government repository and writes to the blockchain when a specific law changes. After this, a notification is sent through a blockchain 3.0 application to prove the date and time sent, to inform interested parties, and to request and record confirmation of receipt on the blockchain (Sofia et al., 2019).

2.2.2 Life Cycle of Smart Contracts

(Zheng *et al.*, 2020) stated that the whole life cycle of smart contracts consists of four consecutive phases they include:

1. **Creation of smart contracts:** Several involved parties first negotiate the obligations, rights and prohibitions on contracts. After multiple rounds of discussions and negotiations, an agreement can reach. Lawyers or counsellors will help parties to draft an initial contractual agreement. Software engineers then convert this agreement written in natural

languages into a smart contract written in computer languages including declarative languages and logic-based rule languages (Idelberger *et al.*, 2016). Similar to the development of computer software, the procedure of the smart contract conversion is composed of design, implementation and validation (i.e., testing). It is worth mentioning that the creation of smart contracts is an iterative process involving multiple rounds of negotiations and iterations. Meanwhile, it is also involved with multiple parties, such as stakeholders, lawyers and software engineers.

2. **Deployment of smart contracts:** The validated smart contracts can then be deployed to platforms on top of blockchains. Contracts stored on the blockchains cannot be modified due to the immutability of block-chains. Any emendation requires the creation of a new contract. Once smart contracts are deployed on blockchains, all the parties can access the contracts through the blockchains. Moreover, the digital assets of both involved parties in the smart contract are locked via freezing the corresponding digital wallets (Sillaber & Walzl, 2017). For example, the coin transfers (either incoming or outgoing) on the wallets relevant to the contract are blocked. Meanwhile, the parties can be identified by their digital wallets.
3. **Execution of smart contracts:** After the deployment of smart contracts, the contractual clauses have been monitored and evaluated. Once the contractual conditions reach (e.g., product reception), the contractual procedures (or functions) will be automatically executed. It is worth noting that a smart contract consists of a number of declarative statements with logical connections. When a condition is triggered, the corresponding statement will be automatically executed, consequently, a transaction is executed and

validated by miners in the blockchains (Sillaber & Walth, 2017). The committed transactions and the updated states have been stored on the blockchains thereafter.

4. **Completion of smart contracts:** After a smart contract has been executed, new states of all involved parties are updated. Accordingly, the transactions during the execution of the smart contracts as well as the updated states are stored in blockchains. Meanwhile, the digital assets have been transferred from one party to another party (Zheng *et al.*, 2020).

2.2.3 Application of Smart contract

Smart contracts have a broad spectrum of applications ranging from the Internet of Things to sharing economy. In particular, (Zheng *et al.*, 2020) categorize major smart contract applications into six types namely:

1. Internet of Things.
2. Distributed system security.
3. Finance.
4. Data provenance.
5. Sharing economy.
6. Public sector Smart.

2.3 Construction Industry

Regardless of scale and complexity, construction projects are predominantly inter-firm collaborative tasks. For instance, lack of communication and collaboration have been reported as inhibitors of project success (Hargaden *et al.*, 2019). Industry reports point to a lack of communication as the primary roadblock to project success (Turk & Klinc, 2017). In addition, the

issue of managing virtual project teams and the geographical dispersal of project stakeholders suggest that enhancing collaboration is a key issue in need of improvement (Laan *et al.*, 2011).

Construction projects are highly regulated processes. In recent years, governments have imposed codes and regulations that provide minimum standards for various elements within the project. These standards require certification and approval from a combination of individuals and authorities. Coinciding with this is trust issues, which are common within construction, primarily due to the adversarial relationships between principal and contractor organizations. This trust conflict arises due to the high level of complexity, risk, and uncertainty associated with projects (Hargaden *et al.*, 2019). It is therefore evident that trust and transparency are becoming a huge challenge for construction managers.

2.3.1 Smart Contract in Architecture, Engineering, and Construction (AEC) Industry.

The construction and engineering management (CEM) domain constitutes one of the leading fields, with a very complex contractual system depending on the type at stake (Owusu *et al.*, 2020). As a result, any effort towards the improvements of contractual arrangements management or administration is worth exploring. Given that the AEC industry is one of the most contract-bound industries globally, it would not have been surprised to see extensive engagements and contributions toward smart contract research by CEM scholars. Due to Construction management applications being among the seven potential research areas identified in a review of distributed ledger technology and blockchain (Hamledari & Fischer, 2021), it is rather surprising that even with the upsurge of smart contract research in other domains, not much research has gone into examining the feasibility and applicability of this promising tool in AEC. To understand the benefits of smart contracts in the construction sector, focus groups were used to develop socio-

technical frameworks (Hamledari & Fischer, 2021). In the real estate industry, for example, a smart contract is estimated to reduce costs by 9% (Dakhli *et al.*, 2019). Token-based payment systems were identified as one of the six potential use case categories for smart contracts and blockchain (Hamledari & Fischer, 2021).

With the merits and findings of what smart contracts can do, it motivated a focus on the possibility of integrating blockchain and smart contracts with building information modelling. Blockchain can supplement current approaches to centralized modelling of building information by improving provenance tracking and increasing the trustworthiness of project records (Dakhli *et al.*, 2019).

One study sees smart contracts as an extension to BIM-based processes, while such integration is not considered an immediate research problem in another study while others discussed arguments both for and against such integration (Hamledari & Fischer, 2021). An industry survey further argues that smart contracts are not suitable for complex construction projects where changes are common (Gabert, 2018). Other works have listed the potential benefits of BIM and blockchain for post-disaster recovery (Nawari & Ravindran, 2019). Crypto-BIM uses blockchain technology and the InterPlanetary File System to create a content-addressable, immutable and distributed view of building information data (Hamledari *et al.*, 2018). Despite the potential applications found in the literature, successful adoption calls for a more careful analysis of the match between industry problems and smart contracts' key features (Hunhevicz & Hall, 2020). While blockchain and smart contracts can provide promise in the context of progress payment automation and concerning supply chain flows, it is not clear what distinguishes this technology from a computerized payment application or other potential means of achieving automation (Wang *et al.*, 2007). We lack an understanding of the why of blockchain in the context of payment automation; there is a need for

analyzing the underlying barriers to automation and their relationships to the defining characteristics of blockchain and smart contracts.

2.3.2 Smart Contract in Construction Projects

Smart contracts are self-executing programs that utilize blockchain technology to digitally enforce verifying or negotiation of a contract. Therefore, they offer credibility between contracting parties, without involving third parties (Leka & Selimi, 2021).

A notable characteristic of large-scale structural engineering and construction projects in recent years has been to design and build new structures and integrated systems with a level of complexity that has never before been attempted (Hargaden *et al.*, 2019). The intricate nature of these designs has led to the need for performance capabilities not previously considered. With technologies advancing at an exponential rate and construction projects becoming more and more complex, engineering project managers must learn to adapt to this ever-changing environment by adopting new practices and technologies. One such technology that has gained traction in recent years is blockchain.

Blockchain's potential has only recently gained attention in the construction industry. It has been suggested that blockchain technology will provide solutions to issues surrounding information management in construction (Turk & Klinc, 2017). In the past, the construction process has been somewhat fragmented. Companies either specialize in information processes or material processes. Information processes were held together by a chain of paper documents that were created collaboratively with people working closely together, before being exchanged by other firms as large and completed documents (Hargaden *et al.*, 2019). Authorship of these documents was clear and legitimate and never an issue as information was shared infrequently and signed on the paper

documents themselves. Alterations were therefore clearly visible and traceable (Turk & Klinc, 2017). Today, digital technology is enabling further fragmentation in the construction process. Digital communication has resulted in companies getting smaller and information exchanges across organizational and legal boundaries more frequent. The planning and design process is almost entirely digitized now and information is being shared in digital formats (Hargaden et al. 2019). Overall, blockchain utilization in construction is limited at present. Nevertheless, there are several potential use cases in the industry, such as maintaining records of digital property, timestamping acts or transactions, multi-signature transactions, smart contracts, and smart oracles (Erri *et al.*, 2019).

2.4 Building Information Modelling (BIM)

Although the construction industry has been evolving for centuries and researchers have been seeking innovative solutions for decades, diverse challenges still exist for making the construction process faster, safer, cheaper, and more accurate (Zhang *et al.*, 2013).

However, it is now believed that Building Information Modeling (BIM) can lead to greater efficiency through incremental collaboration. The data in the BIM system are extremely useful and can be generated to optimize the project delivery processes. Because BIM increases the design cost and requires a big learning curve, project participants are all concerned about the project cost, hindering the adoption of BIM for project delivery (Chang *et al.*, 2017).

Building Information Modelling (BIM) is a process of integrating and disseminating information around a network of project team members. It enables the project team to work simultaneously on a project in real-time and to construct a building directly from a digital model (Aina, 2015).

It is suggested that BIM will soon be the norm for construction work processes (Hargaden *et al.*, 2019). (Li et al., 2019) stated that the adoption of BIM has been commonly seen as a progression that involves levels of increasing capability maturity across technology, process, and policy fields. According to Autodesk, BIM is “an intelligent 3D model-based process that gives architecture, engineering, and construction professionals the insight and tools to more efficiently plan, design, construct and manage buildings and infrastructure” (Hargaden *et al.*, 2019). As seen in figure 2.2, the use of BIM has increased visualization to 3D visualization bettering information modeling and work process comprehension believed to significantly increase productivity as the digital documentation required by it will allow for more product data use (Pradeep *et al.*, 2019). Construction companies must be able to cope with and understand how to deal with this increase in data efficiently.

BIM technology is revolutionizing the construction process; the problem has shifted from modeling buildings to managing buildings (Turk & Kline, 2017). The technology is disrupting the traditional, hierarchical-based project which contains ‘silos of data. This is due to its collaborative features and the establishment of a methodology for data sharing (Mathews *et al.*, 2017). The substantial amount of information that is inputted into BIM suggests that a lot of the data is relevant from a legal perspective. For this reason, it can be said that BIM also refers to the administration of legally significant information that can be used in legal disputes with any of the numerous associated parties. Noting this, it is suggested that legal and security issues are the main obstacles to BIM adoption (Mathews *et al.*, 2017). Other concerns with BIM relate to organizational issues such as responsibility, ownership, traceability, and risk allocation. BIM provides an integrated system that can be used to simulate the behaviour of buildings in a real-world system, provide

information about quantities and properties of building elements and document design information in an integrated database (Aina, 2015).

It is proposed that blockchain technology could be significant in addressing these issues (Turk & Klinc, 2017);(Li *et al.*, 2019). The integration of BIM and blockchain can have a significant impact on construction activities and facilities management, especially where tracking of components proves useful and where there is duplication of work (Li *et al.*, 2019). Blockchain for BIM differs from traditional blockchain applications such as bitcoin from a transactional point of view. Bitcoin involves billions of 1-kilobyte transactions whereas blockchain for BIM comprises hundreds of gigabyte transactions (Hargaden *et al.*, 2019). Four potential architectural solutions for managing building information were proposed (Turk & Klinc, 2017), the most promising of which is ‘Blockchain for BIM Transactions’. This scenario fully integrates blockchain’s decentralized, immutable features within the BIM server.

It is the most beneficial solution proposed, as the amount of data stored within the blockchain solution is significantly greater than the current BIM solution. This allows for a substantially larger information share. Even though the solution requires far more data storage capacity, current technologies allow for this (Turk & Klinc, 2017). A very significant advantage of using BIM for cost is that it will provide accurate and reliable cost estimation based on the digital 3D model, and eliminate errors caused by manual measurements or estimations (Ye *et al.*, 2020).

The authors suggest that blockchain should be integrated within the transaction processing component of the BIM server as well as the storage functionality. It should also trace all information exchanges and communication between participants (Turk & Klinc, 2017). Similarly, the potential of blockchain for BIM was observed by highlighting the aspect of the ‘value network’

that the technology creates (Mathews *et al.*, 2017). This intrinsic value, and the underlying fundamental property that is data, lends itself to blockchain technology. By combining the technologies, a consensus system can be built for any amount of ‘value transactions’ within the construction supply chain network (Mathews *et al.*, 2017).

The start-up Bimchain is currently developing a blockchain-based solution that claims to revolutionize BIM into a collaborative, legally binding process. The ultimate aim of this platform is to create a decentralized version of BIM. The company is currently running proof-of-concept trials which are intended to integrate its distributed ledger technology into BIM processes, tools, and data. The company claims to have formed a link between 3D digital modelling and the formal and legally binding paper-based processes related to project administration, building control, insurance, and payment (Hargaden *et al.*, 2019).

BIM adoption has significantly increased in recent years after mandatory BIM requirements imposed by governments and large clients in different countries. For example, the UK government will require BIM on all public sector projects by 2016, which has pushed the UK construction industry to adopt BIM (Abdulrahman & Naim, 2018).



Figure 2.2: Image of 3D Model of a BIM

2.5 Distributed File Systems (DFS)

A file system is a subsystem of an operating system whose purpose is to organize, retrieve, store and allow sharing of data files. A Distributed File System (DFS) is a distributed implementation of the classical time-sharing model of a file system, where multiple users who are geographically dispersed share files and storage resources. Accordingly, the file service activity in a distributed system has to be carried out across the network, and instead of a single centralized data repository, there are multiple independent storage devices (Suralkar *et al.*, 2013).

A file system is a system that controls the way data is stored and retrieved. There are different types of file systems. A file system can be used just as simply as a local storage device, but there are also (distributed) network file systems. The latter uses network protocols in order to let connected devices communicate with each other within a local area network. A file system differs from a file synchronization protocol. Every storage device contains a file system. File

synchronization happens between multiple file systems. In the case of a distributed (network) file system, central storage servers (within the network) are used from which files are stored & retrieved. File synchronization can then e.g. provide solutions to have some sensitive documents synchronized between multiple (LAN) storage servers (Bruin, 2019).

In a traditional file system (TFS), data is organized by its physical location. Hard Drives, often called shared drives, are attached to file servers and shared amongst multiple clients. Clients then map the shared drives to their local network in order to see and access data on those drives. If a client needs to access a file on drive F, the address is \\host-a\F\<file-name>. There are issues with TFS that make it difficult to use in large-scale storage systems. One issue is that a TFS places the responsibility of being aware of shared drives on the clients. If a new file server or shared drive is added to the network, the client is unaware unless told about the new addition (Austria, 2020).

The DFS server allows the client users to share files and store data just as if they are storing the information locally. However, the servers have full control over the data and give access control to the clients. Traditional Peer-to-Peer (P2P) DFSs have inevitable drawbacks such as instability and lacking auditing and incentive mechanisms. Inter-Planetary File System (IPFS), is one of the representative DFSs which integrate with blockchain technologies (Huang *et al.*, 2020). A distributed file system is a physically distributed implementation of the classical time-sharing model of the traditional file system, allowing users to manipulate, organize and share data seamlessly, regardless of its actual location on the network. In this system, storage resources and system clients are dispersed in the network. Each user is both a creator and a consumer of data stored in the system. Thus, the challenge is to provide considerable incentives in an efficient, secure, and practical manner (Huang *et al.*, 2020). One of the most successful P2P distributed file systems, BitTorrent has supported over 100 million online users (Pouwelse *et al.*, 2004). The

biggest distributed file system is HyperText Transfer Protocol (HTTP), which is a web server used to upload data. Then, other peers can access particular data anywhere all over the world. To ensure data accessibility in web servers, a maintaining cost needs to pay. Such maintaining cost increases along with the growth of data popularity. Moreover, another problem is that there are very few ways to share the burden of information dissemination with the clients directly. This is because HTTP lacks upgrading design and thus fails to take advantage of the advanced file distribution techniques proposed in the past few years. Meanwhile, the P2P technique had been gathering at a great pace and soon dominated the majority of data packets on the Internet. Such P2P file systems, like BitTorrent, optimize resources brilliantly by giving different pieces of popular data to clients and enabling them to swap the missing parts with each other (Pouwelse *et al.*, 2004). In this way, the bandwidth consumption of hosts can be balanced and the overall cost of operational expenditure (OPEX) can be also degraded.

Recently, blockchain has become a buzzword in both industry and academia, and the combination of blockchain and the distributed file system is becoming a promising solution, where blockchain is expected to provide incentives and security for the stored files in systems. Currently, the popular blockchain-based distributed file systems include IPFS (IPFS, 2021), Swarm, Storj, and PPIO (Huang *et al.*, 2020).

IPFS is a peer-to-peer distributed file system for storing and accessing files, websites, applications, and data. Concerning the combination with blockchains, IPFS adopted Filecoin as its incentive mechanism; Swarm is a distributed storage platform and content distribution service based on Ethereum. Concerning the combination with blockchains, Swarm adopted Ethereum as its incentive mechanism; Storj is another peer-to-peer decentralized cloud storage platform that allows users to share data without relying on a third-party data provider. Concerning the

combination with blockchains, Storj adopted Metadisk as its incentive mechanism; PPIO is a decentralized programmable storage network that permits users to store and retrieve any data from anywhere on the web. PPIO exploits up to 4 proof algorithms (Huang *et al.*, 2020).

IPFS and Swarm, as the representative DFSs which easily integrate with blockchain technologies, are becoming a new generation of distributed file systems (Huang *et al.*, 2020).

2.5.1 Inter-Planetary File System (IPFS)

Interplanetary File System (IPFS) is the modern approach to solve the above problems in the client-server model and the HTTP web. IPFS is a peer-to-peer hypermedia protocol that stores the file in a distributed manner. It uses content- addressing rather than location-addressing like in Hypertext Transfer Protocol (HTTP). IPFS loads contents faster by using less bandwidth, and deduplication data. This protocol uses Distributed Hash Tables (DHTs) to coordinate and maintain metadata, BitSwap data exchange protocol to distribute pieces of files to each other, a Version Control System (Git) for representing immutable objects files as Merkle Directed Acyclic Graph (Merkle DAG) and Self-Certified File systems (SFS) for distributing the trust chains (Alwis, 2020).

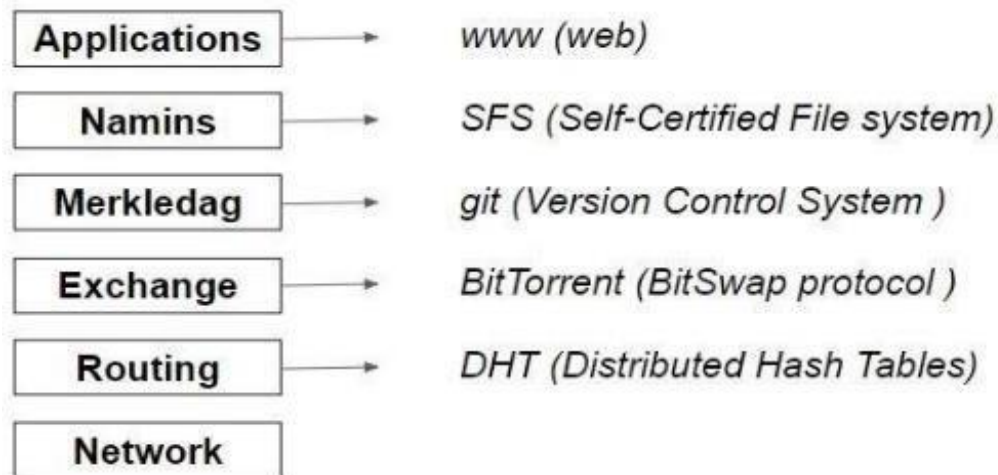


Figure 2.3: IPFS Structural Stack (Alwis, 2020).

The InterPlanetary File System (IPFS) uses a Merkle DAG (Directed Acyclic Graph) as its data object model (IPFS, 2021). An object in IPFS is a structure containing two attributes: Data and Links. Each Link structure includes three attributes: Name, Hash, and Size. Using this object structure, IPFS can compose objects and build a directed acyclic graph. The IPFS structure stack is shown in figure 2.3. In IPFS, Merkle DAG organizes the structure of a file or even a file directory as shown in Figure 2.4. In figure 2.4, there are two files (example.js and hello.txt) and one file path (dir) in the root path of this file directory, example.js is divided into three different data pieces and file path dir has two files: other.txt and example.txt (here file content of example.txt in dir and hello.txt in root path are the same therefore they are linked to the same object), each object derives its value through computing its children's value and the content of data.

In IPFS, data uploaded to the distributed file systems by users is divided into several pieces, which are then stored in different peers. The data content stored in the network is accessible by every peer (Huang *et al.*, 2020). Besides, according to the design of IPFS, transactions that record the developments of a peer can be easily collected. User information can be revealed through the graph

analysis of transactions. For example, according to (Yao *et al.*, 2019), a client can be identified through the peers it directly connects to. Thus, transactions stored in the blockchain behind distributed file systems are publicly visible.

2.5.2 Identity layer in DFS

To archive the content distribution between nodes in a P2P file system, each node has to be identified by a unique identifier, which needs to ensure collision-free. It means that two different data objects can never map to the same identifier. In IPFS, the encrypted hash (in multi-hash format) of a public key, i.e., NodeId, is used to identify each node. The format of multi-hash is {hash function code} {hash digest length} {hash digest bytes}. Nodes periodically check public keys and NodeId when connecting. In SWAM systems, the node hash-address is generated by Keccak 256 bit SHA3 using the public key of an Ethereum account (Huang *et al.*, 2020).

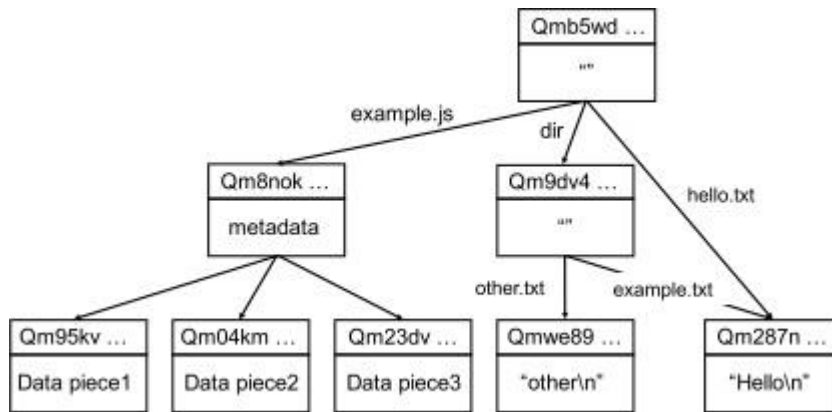


Figure 2.4: Merkle DAG in IPFS (Huang et al., 2020).

2.5.3 Network Layer

Under the framework of IPFS, an advanced generic P2P solution, named libP2P (Huang et al., 2020), is exploited as the network layer. libP2P is developed based on BitTorrent DHT implementation. Based on libP2P, IPFS can use any network protocol to transfer data. If the underlying network is not stable, IPFS can alter to choose UTP or SCTP. IPFS achieves this free shifting mainly by using a multi-add formatted technique (*IPFS*, 2021), which combines addresses and corresponding protocols. Swarm relies on the Ethereum P2P network, which is comprised of three different protocols:

1. RLPx (Recursive Length Prefix) for node discovery and secure data transmission.
2. DevP2P for node session establishment and message exchange.
3. Ethereum subprotocol.

DevP2P is inspired by libP2P and has security properties that are beneficial to Swarm. When discovering through RLPx, Swarm nodes establish TCP connections and send “HELLO” messages including NodeId, listening port, and other attributes based on DevP2P. Sessions start to transmit data packets. Due to the ecosystem of Ethereum, Swarm has a large number of long-term nodes, which support the robustness and stability of Swarm systems (Yao *et al.*, 2019).

2.6 Data Encryption

With the increased usage of data exchange and communication through the Internet, it becomes crucial to secure data from cyber-attacks (Alenezi *et al.*, 2020). Nowadays, providing data confidentiality and privacy has presented a significant challenge for researchers and professionals in the realm of cybersecurity. Data confidentiality means protecting data against unauthorized

access or theft. It can be achieved with the help of cryptography through data encryption and decryption. Cryptography aims to secure critical data or documents on a hard disk, or when it is transferred through an insecure communication channel (Alenezi *et al.*, 2020). Encryption regarding data storage is a security method that transforms data such that the data becomes unreadable. Only those in possession of the encryption key can decrypt and access the data (Austria, 2020).

Data encryption is the art of securing messages by converting them to hidden texts, whereas the inverse process of retrieving original texts from hidden texts is called decryption. Encryption/decryption is made possible with the help of some keys. Every encryption algorithm aims to make the decryption process as difficult as possible without the help of the key used in encryption. The basic classification of cryptography has been summarized in figure 2.5 (Alenezi *et al.*, 2020). The types of cryptographic techniques we have include; Symmetric key encryption, Asymmetric key encryption and Hashing (Alenezi *et al.*, 2020).

2.6.1 Classical Ciphers

Classical ciphers are used to encrypt plaintext messages written in a natural language into ciphertext based on a set of rules, i.e., the encryption algorithm, and a secret key only known to the sender and intended receiver of a message (Kopal, 2018). Many classical ciphers can be broken by brute-force search through the key-space. One of the pertinent problems arising in automated cryptanalysis is plaintext recognition. The first Classical Cipher, Caesar Cipher, an ancient Cipher deployed at the time of Julius Caesar that works with shift key 3 over modulo 26 where the plain text is over alphabets A to Z. By the Brute force attack the Caesar is vulnerable (Education, 2021). Classical ciphers, as well as ciphers in general, can be divided into two different main classes:

1. **Substitution Ciphers:** A substitution cipher replaces letters or groups of letters of the plaintext alphabet with letters based on a ciphertext alphabet (Kopal, 2018). Substitution ciphers can be furthermore divided into monoalphabetic and polyalphabetic ciphers (Forsyth & Safavi-Naini, 1993). With monoalphabetic ciphers, only one ciphertext alphabet exists. Thus, every plaintext letter is always replaced with the same letter of the ciphertext alphabet. If there are more possibilities to choose from the ciphertext alphabet the substitution cipher is a homophone substitution cipher (Forsyth & Safavi-Naini, 1993). If there is more than one ciphertext alphabet that is exchanged after each encrypted letter, the substitution is a polyalphabetic substitution, e.g., the Vigen`ere cipher. Substitution may also not only be based on single letters but on multiple letters (Schrödel, 2008).
2. **Transposition Ciphers:** Transposition ciphers do not change the letters themselves but their position in the text, i.e. plaintext alphabet and cipher text alphabet are equal (Kopal, 2018). Transposition ciphers change the positions of each letter in the plaintext based on a pattern that is based on a key. The most used transposition cipher is the columnar transposition cipher (Kopal, 2018). Here, the plaintext is written in a grid of columns. Then, the columns are reordered based on the lexicographical order of a keyword written above the columns. Finally, the ciphertext is read out of the transposed text column-wise. Decryption is done the same way but in the reverse order.

2.6.2 Modern Ciphers

Modern cryptography is a cornerstone of computer and communications security, with end products that are imminently practical (Bellare & Rogaway, 2005). Its study touches on branches of mathematics that may have been considered esoteric, and it brings together fields like number theory, computational-complexity theory, and probability theory. Modern cryptography addresses

a wide range of problems. But the most basic problem remains the classical one of ensuring the security of communication across an insecure medium (Bellare & Rogaway, 2005).

2.6.3 Symmetric Key Encryption

In symmetric key encryption, both encryption and decryption are done based on a single key called a private key. It is also referred to as a secret key. Hence the key is in the sense of mystery (Princy, 2015);(Alenezi *et al.*, 2020);(Hamza & Kumar, 2020). A secure channel is required for sharing this private key between the sender and receiver. Symmetric algorithms have the pros of not taking too much computing power and it works with a high level of speed in encryption. Symmetric key cryptographic algorithms are divided into two types based on the input data:

1. **Block Ciphers:** In block cipher-based systems, data is processed or encrypted on a fixed-length group of bits called a block (Alenezi *et al.*, 2020). Input is caught as a block of the plaintext of fixed size look upon the type of a symmetric encryption algorithm, a key of lasting size is applied to the block of plain text and then the output block of the same size as the block of plaintext is received. The key trick of a block cipher is the differing key length, block size and a number of rounds (Hamza & Kumar, 2020). Allowing the plaintext to be encrypted to ciphertext1 than ciphertext1 to encrypt again to ciphertext2 and so on. A basic number of round and fix plaintext block sizes allow the block cipher to be stronger than the historical counterpart's cipher techniques. Some used block cipher algorithms include Data Encryption Standard (DES) and Advanced Encryption Standard (AES) (Princy, 2015).
2. **Stream Ciphers:** In-stream cipher-based systems, data is processed on a stream of bits (Alenezi *et al.*, 2020). The bits used in sequence as keystream are generated & combined

with the plaintext using bitwise exclusive-OR (XOR) (Hamza & Kumar, 2020). Where this keystream could be generated independently from the plaintext (synchronous stream) or it could be linked and dependent on the plaintext (self-synchronous stream). Allowing higher performance in transforming the plaintext to ciphertext and via versa. Some most usable Symmetric-key algorithms include Data Encryption Standard, 3DES, and Advanced Encryption Standard (Princy, 2015).

The Key difference between the stream cipher and block cipher is Stream ciphers encrypt bits individually. This is achieved by adding a bit from a key stream to a plaintext bit and Block ciphers encrypting an entire block of plaintext bits at a time with the same key. This means that the encryption of any plaintext bit in a given block depends on every other plaintext bit in the same block (Hamza & Kumar, 2020).

2.6.4 Common Symmetric Encryption Algorithms

Many encryption methods are being used in cryptography. In this section, we detail a few common encryption algorithms based on both stream and block ciphers.

1. **Data Encryption Standard (DES):** DES is one of the basic symmetric key block cipher algorithms which takes plain texts as blocks each one carrying 64 bits and converts ciphertexts using keys of 64 bits. Out of these 64 bits, 8 bits of the key are used for odd parity which will not count in key length (Stallings, 2017). Therefore, there exist 256 possible ways to find the correct key. The DES algorithm performs two permutations (initial permutation and final permutation) and 16 processing steps, each of which is called a round, and for each round, a different key is used. DES is based on two cryptographic operations: substitution and transposition (Alenezi *et al.*, 2020). In each round of DES,

some substitutions and transpositions are performed. Before starting the first round, an initial permutation is applied to the plain text. For example, an initial permutation replaces the first bit of the plain text with the 58th bit, and the second with the 50th bit, and so on. The resultant permuted block is divided into two halves, both having 32 bits and each one going through 16 rounds of encryption processes. The final permutation is applied to the combined block to get the ciphertext. DES has been reported as vulnerable and as such was replaced with 3DES (Hamza & Kumar, 2020).

2. **Triple Data Encryption Standard (3DES):** Triple-DES is a block cipher encryption algorithm. As its name indicates, 3DES applies DES three times to each data block to enhance the security of the encrypted data. Since the security of 3DES is three times better than that of DES, it is now considered preferable to DES (Ratnadewi *et al.*, 2018). However, it does consume a considerable amount of time in comparison with its predecessor. 3DES works in the same way as DES, in a loop with length 3. Initially, the original plain text is encrypted with one key, the resulting ciphertext is again encrypted using another key, and finally, it is performed again with a third key (Alenezi *et al.*, 2020).
3. **Advanced Encryption Standard (AES):** Advanced Encryption Standard (AES), is a block cipher algorithm that came as a replacement for DES and Triple DES (Alenezi *et al.*, 2020). It encrypts and decrypts a 128-bit block of data. Based on the choice of key size, 128 bits, 196 bits, or 256 bits, AES can take 10, 12, or 14 rounds for encryption (Alenezi *et al.*, 2020). Each round consists of four operations: substitute bytes, shift keys, mix column and add round key. However, the mix column operation is not performed in the last round. Separate round keys generated from the given cipher key are used in each round of encryption. Data to be encrypted is divided into blocks. Each block is represented as an

array of data which is known as a state array. No cryptanalytic attack on AES has been discovered so far. AES has flexibility over key size, which allows it to protect to a certain point against the progress of the ability to run exhaustive searches for encryption keys by an attacker (Boicea, 2019).

4. **BlowFish:** BlowFish is a block cipher-based encryption algorithm whose key length varies from 32 bits to 448 bits. Each block handles 64 bits of data (Stallings, 2017). BlowFish encrypts the data through 16 rounds of operations. At each round, data undergoes a key-dependent permutation in the P-block and substitution in S-block. Each S-block carries 32 bits of data. BlowFish was made as a multi-purpose algorithm and to be an alternative to DES (Boicea, 2019).
5. **Twofish:** Twofish is also a block cipher-based symmetric encryption system that works in a similar manner to BlowFish. Unlike BlowFish, however, Twofish is considered to be flexible. Twofish allows users to customize encryption speed, key setup time, and code size and works fast in an 8-bit CPU as well as in smart cards, embedded chips, etc. It is freely available to use as it is unpatented, license-free software. Twofish encrypts the documents of 128-bit block size with key sizes of 128, 198, or 256 bits in 16 rounds of encryption (Alenezi *et al.*, 2020).
6. **Threefish:** Threefish is a tweak-able block cipher-based encryption standard that takes three inputs: a key, a tweak, and plain text, to be encrypted. Threefish uses the same length key as the data block size for encrypting a block of data. This encryption method is used for data blocks of size 256, 512, and 1024 bits. Threefish scheme produces encrypted data by repeating the same sequence of operations 72 times (or rounds) except for a 1024-bit block of data, which takes 80 rounds. A 128-bit tweak value is used for all of these data

block sizes. Operations of Threefish encryption standards are of three types: addition, XOR, and rotations (Alenezi *et al.*, 2020).

7. **Rivest Cipher 4 (RC4):** RC4 is a symmetric stream cipher algorithm in which each character is encrypted one at a time, commonly used in wireless routers. The key length of RC4 varies from 40 to 2048 bits. To get a more robust encrypted text, 16-byte keys are preferred. Data blocks are XORed with keystream bytes one by one to encrypt the data. The working of RC4 is mainly relayed on the creation of keystream bytes, which are entirely independent of plain text (Alenezi *et al.*, 2020). RC4 is very vulnerable to attacks (Boicea, 2019).

2.6.5 Asymmetric Key Encryption

Asymmetric key cryptographic systems require two pairs of keys; a public key and a private key. Encryption is accomplished with the use of a public key, whereas the secret key is used to decrypt the encrypted text. Both of these keys are mathematically related. Although asymmetric systems provide a higher level of security, they might not be well-suited for large-sized documents. This is because the speed is slow compared with symmetric key-based systems and they also record a huge overhead in terms of memory computation power and a higher rate of CPU utilization (Senthil *et al.*, 2016). Though the public key is broadcasted to every node, hackers will not be able to retrieve private keys only with the help of public keys. Assume that message ‘M’ was encrypted with the known public key. To decrypt the cipher text ‘C’ we need the private key. Occasionally we prefer to encrypt using private keys and decrypt using public keys.

2.6.6 Common Asymmetric Encryption Algorithms

There are few well-known algorithms in asymmetric encryptions. In this section, we detail a few common asymmetric algorithms.

1. **Rivest, Shamir, Adleman (RSA):** RSA cryptography, can encrypt a message using both the public and the private keys. The opposite key from the one used to encrypt a message is used to decrypt it. This attribute is one reason why RSA has become the most widely used asymmetric algorithm (Hamza & Kumar, 2020). It provides a method to assure the confidentiality, integrity, authenticity, and non-repudiation of electronic communications and data storage. RSA is used in software programs and browsers, as they need to establish a secure connection over an insecure network, like the internet, or validate a digital signature. RSA signature verification is one of the most commonly performed operations in network-connected systems The RSA can be used to securely transfer and exchange the symmetric encryption secret keys and validation of sender and receiver (digital signature, non-repudiation) and symmetric encryption is used to encrypt data bulks (Hamza & Kumar, 2020).
2. **Digital Signature Algorithm (DSA):** The RSA cryptosystem implements the Multiplicative Homomorphic encryption characteristics (Yang, 2022). For every 4096-byte block (30+ piconets/bit), it roughly cost user 1M piconets to interpret the verification and to protect the signature through a single hash-chain written in 1024-bit RSA, each one costs twenty 4096-byte blocks 180+ piconets/bit. In a computer, a digital signature is represented as a string of binary digits. By decrypting and using the client's key, a digital signature checks the integrity of the signed document as well as the identity of the signatory. A fresh hash file of the received picture is constructed using the same algorithm, and both hash

files are compared for signature validation; if they match, no harm has happened, and the sender is an authentic user. The capacity to generate and verify signatures is provided by an algorithm. There is a wide range of Digital signatures' that are used to secure the stability of common data or saved one, as well as to ensure recipients that the author is the right one. American National Institute of Standards and Technology (NIST) considered DSA as a general measurement of digital signatures (Yang, 2022)

3. **Elliptic Curve Cryptography (ECC):** ECC is a type of cryptography that works with dots on an elliptic curve. An essential mathematical procedure is Modular integer exponentiation. The action of scalar point multiplication, which measures $Q = kP$ (a dot P times k times ending in point Q on the graph), lies at the heart of elliptic curve arithmetic. Point additions (joint two different points) and point doublings (plus two transcripts of a point) are used to achieve scalar multiplication. Back 20 years before, Elliptic curves were treated as the foundation for discrete logarithm-based cryptosystems, separately by IBM's Victor Miller and the University of Washington's Neal Koblitz. Elliptic curves were already in use in many cryptographic applications at the time. Elliptic curves are complex numerical constructs that have demonstrated a wide range of applications (Yang, 2022).
4. **Diffie-Hellman Key Exchange algorithm:** Diffie-Hellman Key Exchange or Diffie-Hellman Protocol were the names given to the first public key algorithms published in Diffie and Hellman papers (Yusfrizal *et al.*, 2019). The Diffie-Hellman Key Exchange algorithm enables two users to exchange keys securely, and then encryption and decryption of subsequent messages. The Diffie-Hellman protocol provided the first practical solution to the key distribution problem, allowing two parties, never having met in advance or shared keying material, to establish a shared secret by exchanging messages over an open

channel. The key can then be used to encrypt subsequent communications using a symmetric key cipher. The security rests on the intractability of the Diffie-Hellman problem and computing discrete logarithms.

Hashing: In hashing, an input message is mapped into a compact fixed-size bit string called a hash. Hash functions are one-way functions which are mathematical algorithms that map the input message of arbitrary size into a fixed-size hash or message digest. Hash functions are mainly used for password storage and data integrity check. The most widely used hash functions include:

1. Secure Hashing Algorithm (SHA)
2. RACE Integrity Primitives Evaluation Message Digest (RIPEMD)
3. Message Digest Algorithm (MD)
4. Whirlpool

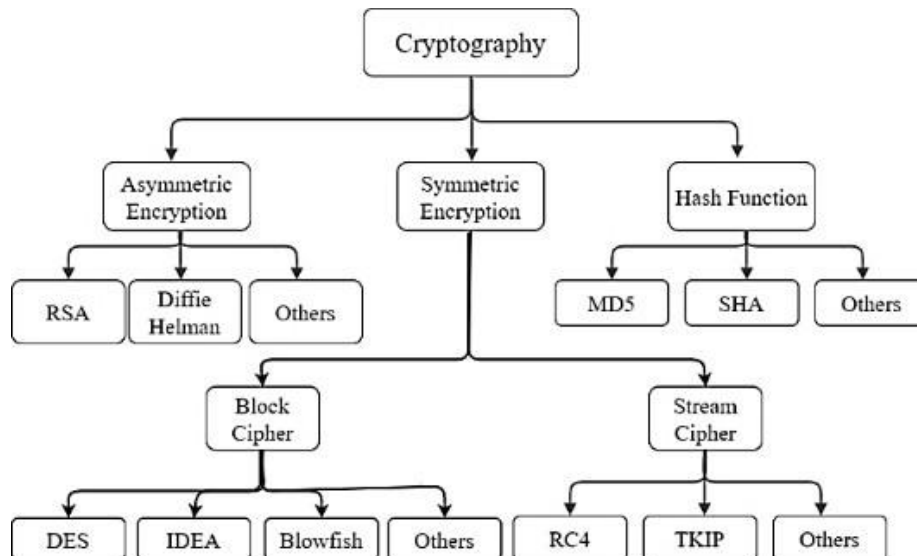


Figure 2.5: Basic Classification of Cryptography (Alenezi et al., 2020)

2.6.7 Cryptographic Benchmarks

Cryptographic benchmarks are a set of tests that measure the performance of cryptographic algorithms, such as encryption and decryption, on a specific hardware platform. These benchmarks can be used to evaluate the security and efficiency of cryptographic implementations, and to compare the performance of different cryptographic libraries or hardware devices (Braga *et al.*, 2017). Some common cryptographic benchmarks include;

1. **Advanced Encryption Standard (AES) benchmark:** The AES benchmark is created to replace the widely used Data Encryption Standard (DES). It was established by the National Institute of Standards and Technology (NIST) in 2001 to set in motion a chain of activities that promises to build a foundation for stronger and better cryptographic standards for the 21st century, which is vital in this era of e-commerce and e-government (Burr, 2003). It is widely used for secure data encryption and is the successor to the previously used Data Encryption Standard (DES). AES uses a fixed block size of 128 bits and supports key sizes of 128, 192, and 256 bits. It's widely considered a standard for symmetric encryption and is used in various systems as standards such as TLS/SSL, VPNs, disk encryption, etc. (Nechvatal *et al.*, 2001). AES encryption is also commonly used in benchmarks as a way to test and compare the performance of different processors and devices (Burr, 2003).
2. **Secure Hash Algorithm (SHA) benchmark:** Secure Hash Algorithm (SHA) is a family of cryptographic hash functions that are widely used for data integrity verification and digital signatures (Sklavos & Koufopavlou, 2003). The most commonly used versions of SHA are SHA-1, SHA-256, and SHA-3. SHA-1 is now considered a weak algorithm,

which has been replaced by the more secure SHA-2 and SHA-3. SHA-256 and SHA-3 are considered to be more secure and are currently recommended for use by NIST. SHA functions are primarily used for the integrity check of data, like in digital signature schemes, the most common usage of SHA-256 is in bitcoin mining (Dahal *et al.*, 2013). SHA can also be used as the benchmark, typically in performance tests of hashing algorithms and is used to test the speed and efficiency of various devices and systems.

3. **National Institute of Standards and Technology (NIST):** The National Institute of Standards and Technology (NIST) is a non-regulatory agency of the United States Department of Commerce. NIST's mission is to promote innovation and industrial competitiveness by advancing measurement science, standards, and technology (Barker *et al.*, 2012). The NIST provides guidelines and recommendations for cryptographic benchmarks in order to help organizations choose and use cryptographic systems that provide the level of security they require. NIST's guidelines for cryptographic benchmarks include the selection of cryptographic algorithms, key sizes, and other parameters, as well as the testing and evaluation of cryptographic systems (Sonmez *et al.*, 2021). (Barker *et al.*, 2012) is a NIST cryptographic guideline that provides recommendations for the selection, use, and management of cryptographic keys used in Federal information processing. It also recommends the key sizes, cryptographic algorithm and protocol for different levels of protection. (Barker *et al.*, 2011) is another NIST cryptographic guideline that provides guidelines for transitioning from cryptographic algorithms that are being phased out to newer, more secure algorithms. It also recommends the cryptographic algorithm and key size for use in the Federal government. These publications are subject

to change, review and revision as the field of cryptography evolves, but they have become widely referenced and considered a source of best practice in the industry.

4. **European Union Agency for Cybersecurity (ENISA):** ENISA is an agency of the European Union (EU) that was established in 2004 to improve the EU's cybersecurity capabilities. Its mission is to protect the EU's digital infrastructure and services against cyber-attacks by providing technical and operational assistance to EU member states, raising awareness of cybersecurity issues, and promoting cooperation between member states and other stakeholders on cybersecurity matters (Brun, 2017). One of ENISA's main areas of focus is cryptography, which it addresses through research, technical assistance, and guidance to EU member states and other stakeholders. This includes analyzing the latest cryptographic algorithms and technologies, providing recommendations for the secure use of cryptography in various domains, and helping to develop EU-wide standards and guidelines for cryptography. Additionally, ENISA collaborates with other international organizations and experts to stay up-to-date on the latest developments in the field (Kouroumbashev, 2019).

Cryptography brings about the provision of higher security levels for devices that are prone to multiple threats and security risks through the implementation of symmetric and asymmetric encryption and decryption algorithms (Boicea, 2019). However, these algorithms have strengths and weaknesses, therefore, the existing cryptographic algorithms should be well benchmarked in terms of efficiency and lack of vulnerabilities before use. Given the same file size or string size, key length (if applicable) and the number of tests to be performed by the algorithm, measure the time taken for the algorithm to encrypt and decrypt the file. Some key

terms in calculating benchmarks for testing the strength and weakness of a cryptographic algorithm include:

1. Execution Time for both Encrypting and Decrypting
2. Data W.R.T.
3. Key Length
4. File Size
5. File Format

2.7 Related Studies

The research and integration of blockchain smart contracts in other domains have been on the high side as the technology brings about a lot of possibilities, yet little research has gone into examining the feasibility and applicability of this promising tool in Architecture, Engineering, and Construction (AEC). Before the integration of blockchain into construction management, previous research proposed measures to Improve construction contract management. For example, standard construction contracts have been proposed by many countries and regions as references for contract formalization for specific types of construction projects, such as the FIDIC contract (Nael, G., 2005). However, standard construction contracts focus on the improvement of the contract structure and are still difficult to interpret by individuals who are not lawyers by profession. To simplify contract management, e-contracts were proposed. E-contracts are created by analyzing relationships between the contract participants and contractual information, followed by modelling traditional textual contracts in XML format (Cardoso & Oliveira, 2008). However, current applications of e-contracts are mainly found in the electronics trade, where the complexity of

relationships between parties, obligations, and activities is simpler compared to that in construction contracts.

Despite the availability of digitized progress data, payment automation in AEC is just beginning to receive attention as projects still rely on traditional payment applications that are time-consuming, information-intensive, and cannot support payment automation (Penzes *et al.*, 2018). Researchers over the years have tried to show how much impactful the integration of blockchain into the AEC. (Hamledari & Fischer, 2021) who researched the role of Blockchain-Enabled Smart Contracts could play in Automating Construction Progress Payments, due stated that progress payment automation is still far from reality, the writer also presented the theory of social reality to identify the underlying barriers that hinder the automation. The writer also argued that the reliance on centralized control, execution mechanisms, and lack of guaranteed execution, the current payment applications, and their supporting contract documents, even when computerized, cannot support progress payment automation. The paper concluded that the introduction of blockchain-enabled smart contracts could bring about the automation of payments by converting product flow (the observation of as-built conditions) to cash flow (progress payments) without reliance on the role of intermediaries.

Few research has tried to store and automate the payment of contracts using blockchain smart contracts with reviews on how much impact blockchain will bring to the automation of payment systems. (Nanayakkara *et al.*, 2021) carried out a literature review using questionnaires on an expert forum of 24 members including the upstream and downstream of the construction supply chain and university academics to identify the payment and related financial issues in the construction supply chain and construction industry. Opinions were carefully compared and the writers concluded that blockchain and smart contract technologies could assist in overcoming

payment-related issues, such as partial payments, payment delays, non-payments, cost of finance, long payment cycle, retention, and security of payment issues, to a great extent.

In the implementation of blockchain-based smart contracts, (Guo *et al.*, 2021) proposed a blockchain-based smart contract to manage contract documents, monitor the signing process, and provide automated contract execution and payment settlement. The system proposed will handle the signing, verification, and validation of certificates and saving contract files using blockchain smart contract technology. The system will ensure that the contracts are protected by digital signatures and certificates. The system proposed cuts the time it takes to sign a contract from 55 to 190 hours (for a conventional paper-based contract) to 16-46 hours. Also, the cost of signing a contract was reduced from RMB2363 to RMB229 per contract. (Ahmadisheykhsarmast & Sonmez, 2020) took a step further by not only storing the information of the contract but also automating the payment process. (Ahmadisheykhsarmast & Sonmez, 2020) proposed a system using smart contracts to automate the payment of construction contracts from employers to contractors. Using solidity as the smart contracts design language, the system's architecture consisted of an add-on software developed in Microsoft Project 2019 to transfer the necessary schedule and cost data to the smart contract via a project management software and a smart contract-based decentralized application designed to be deployed on the Ethereum blockchain. The system will ensure direct payments on fixed periods (weekly or monthly) from the employer's wallet to the wallets of subcontractors and suppliers to improve cash flow and reduce payment issues.

(Luo & Cheng, 2019) proposed a smart contract-based blockchain framework to facilitate the automation of contract payment. The model was proposed to automate payments in the supply chain of construction projects by formalizing construction contracts into smart contracts. The

contract logic formalization involved the contractor, inspector, quantity surveyor, engineer, and employer. The execution of the smart contract was done through a permissioned blockchain-based framework. Consists of an automated consensus process based on pre-defined conditions of the smart contract, storing information in 2 different locations; Ledger and a data model for tasks completed and payments, and a manual process that requires input from the authorized stakeholder.

The introduction of BIM in the construction sector brought about a huge turnaround. It is said to be the most flourishing technology in the construction sector (Martínez-aires *et al.*, 2018), and integrating the technology with blockchain and smart contracts will significantly make a huge impact. (Shojaei *et al.*, 2019) proposed a system that was to integrate the BIM model into a smart contract and create a cyber-physical space for administrating the project through the blockchain network. This study was carried out to test the feasibility of blockchain technology as the link between the BIM model and the physical world with the implementation of smart contracts as the business logic of the blockchain network. The system used a private, permission-based blockchain, using Hyperledger fabric due the cryptocurrency aspect of the blockchain was not used, monetary compensations were executed through traditional channels such as electronic deposits. The system used seven (7) participants (client, architect/engineer, General Contractor (GC), regulators, inspectors, suppliers, and sub-contractors). The writers concluded that due the research method used in the study is by no means optimal, and it is only adopted as a starting point to show the feasibility of the approach. (Ye *et al.*, 2020) proposed a framework for automated contract, invoice, and billing management from a BIM model mapped in a blockchain-enabled smart contract. The proposed system process payments automatically through the banks and use a Common Data Environment (CDE) as off-chain storage that handles all the payment-related files for which a BIM Contract Container (BCC) is used, which contains all payment-relevant data. The BIM model and

a BoQ with QTO were used to create the billing model which is then automatically processed via smart contracts for payments to be automated thereby completely simplifying the payment process (Liu *et al.*, 2019).

IPFS due to its limitations of not having content encryption has managed to be used by many sectors in relation to the blockchain. The fact that IPFS is decentralized in nature and uses the Distributed Sloppy Hash Table (DFSH), has brought about its popularity.

In the Identity management sector, (Liu *et al.*, 2019) proposed an identity management system on biometrics and blockchain/smart contracts to enable secure and privacy-preserving identity management. The proposed system used both the IPFS well-known way hashing algorithm and ground-truth information to verify an individual's identity (Access control).

In the medical sector, (Jabarulla *et al.*, 2017) proposed a secure sharing of medical image data using blockchain and a distributed file system known as InterPlanetary File System (IPFS). The system uses an image pre-processing layer that includes the steganography and encryption techniques, a network layer used to upload encrypted images in IPFS and store user information on the blockchain ledger, and finally, an authentication layer that performs decryption and verifies the authenticity of the image.

In the auto insurance sector, (Nizamuddin & Abugabah, 2021) proposed a Blockchain for automotive using IPFS. The system used an Ethereum-powered smart contract to control and regulate all the entities involved in the insurance claim process and IPFS was used to store the claim form submitted by the customer after reporting the accident.

In the data storage sector, (Kumar *et al.*, 2019) proposed a system for the storage and retrieval of data stored via IPFS through a locally created blockchain. The system uploads files on to Inter Planetary File System (IPFS) and the IPFS generates a unique IPFS hash value.

In the document sector, (Vashistha & Ferdous, 2020) proposed a blockchain-embedded smart contract document management system. The system used a truffle framework is used to implement client applications and the document are encrypted by a symmetric key using the advanced encryption standard (AES) cryptographic algorithm.

2.7.1 Summary of Related work

The related study as captured in this research is summarized in table 1 as shown below.

Table 2.1:Summary of Related work

S/N	Author/Year	Methodology	Contribution to Knowledge	Draw Backs of Research
1	Lingling Guo, Qingfu Liu, Ke Shi, Yao Gao, Jia Luo, And Jingjing Chen (2021)	Blockchain Smart Contract Technology.	A Blockchain smart contract was used to manage contract documents, monitor the signing process, and provide automated contract execution and payment settlement.	Data contents stored in the blockchain is visible to all (No data Encryption)
2	Salar Ahmadisheykhsarmast, Rifat Sonmez (2020)	A Project Management Software (Microsoft Project 2019) and a Smart Contract (Ethereum Blockchain)	Focus on integration of building information modelling to the smart contracts to enable automated smart contract progress payment systems.	Data contents stored in the system are venerable to confidentiality attacks. The payment method is only suitable for small and medium-sized construction firms.
3	H. Luo, M. Das, J. Wang and Cheng (2019)	A permissioned blockchain-enabled smart contract and asymmetric encryption.	Automate payments in construction projects by using smart contracts. The system also used an encryption	Permissioned blockchains are vulnerable to hacking. Asymmetric encryption

			mechanism for data confidentiality and integrity.	takes a lot of time and computational overhead
4	Xuling Ye, Katharina Sigalov and Markus König (2020)	Quantity Take-Off (QTO), Bill of Quantities (BoQ) Building Information Model (BIM), Common Data Environment (CDE) and a BIM Contract Container (BCC).	Framework automation of payments during the construction process by combining the BIM contract container (BCC) with smart contracts.	Data contents stored in the system are vulnerable to confidentiality attacks
5	Alireza Shojaei, Ian Flood, Hashem Izadi Moud, Mohsen Hatami, and Xun Zhang (2020)	A permissioned blockchain-enabled smart contract.	Test the feasibility of blockchain technology as the link between the BIM model and the physical world with the implementation of smart contracts.	Permissioned blockchains are vulnerable to hacking. Data contents passing through the system are venerable to confidentiality attacks
6	Liu, Yaoqing Sun, Guchuan Schuckers, Stephanie (2019)	InterPleneerary File System (IPFS) and blockchain-enabled smart contract (Ethereum)	Developed an identity management framework to integrate a user's transformed biometrical data into a smart contract.	Data contents stored passing through the system are venerable to confidentiality attacks
7	Mohamed Yaseen Jabarulla, Giljun Jung, and Heung-No Lee (2017)	Asymmetric encryption, InterPleneerary File System (IPFS), and steganography	Eliminated third-party intermediaries for image sharing by storing encrypted images on a ledger.	Asymmetric encryption takes a lot of time and computational overhead

8	Barbhuiya, Ferdous Ahmed (2020)	InterPlanetary file system (IPFS) and a symmetric key using an advanced encryption standard (AES) cryptographic algorithm	Transfer documents through a network in a safe, secure and immutable environment.	Data contents stored in the ledger are venerable to confidentiality attacks
9	Kumar Bhosale, Kadayak Akbarabbas, Jadhav Deepak, Awani Sankhe (2019)	Inter Planetary File System (IPFS)	Transfer data to a secure and immutable environment.	Data contents stored in the ledger are venerable to confidentiality attacks
10	Raghavendra, Marangappanavar Kiran, M (2020)	Inter Planetary File System (IPFS), symmetric key encryption and Smart contract	Develop a smart contract and an access control mechanism to effectively secure the data that can be shared with patients.	Once there is access to any doctor's system, the encrypted key can be gotten.
11	Nizamuddin, Nishara Abugabah, Ahed (2021)	Inter Planetary File System (IPFS), Ethereum-powered smart contract.	Securely store the claim form submitted by the customer via IPFS after reporting the accident and an Ethereum-powered smart contract used to control and regulate all the entities involved in the insurance claim process.	Data contents stored in the ledger are venerable to confidentiality attacks

After a critical study of related works on the automation of payments in relation to blockchain smart contracts, it was observed that many researchers have made efforts to automate the payment of contractors. However, most of the limitations is that data contents transferred in the existing systems are vulnerable to confidentiality attacks.

This work aims to design a system to encrypt the data transferred through the inter-planetary file system (IPFS) and blockchain-embedded smart contract using a multi-level authority and encryption for enhanced data security.

CHAPTER THREE

3.0

METHODOLOGY

3.1 System Design

The system design is a three-unit tunnelling system by the IPFS. The encryption key used to encrypt the data input by a supervisor is generated randomly by a key generator. The BIM, where the degree of completion is confirmed, receives the encrypted data next via IPFS. The data is further encrypted after verification, sent through IPFS, and then decrypted in the smart contract. After decryption, the data is analysed by the smart contract, and once payment authorization has been granted, the data is further encrypted before being sent to the payment portal via IPFS. Figure 3.1 displays the procedure.

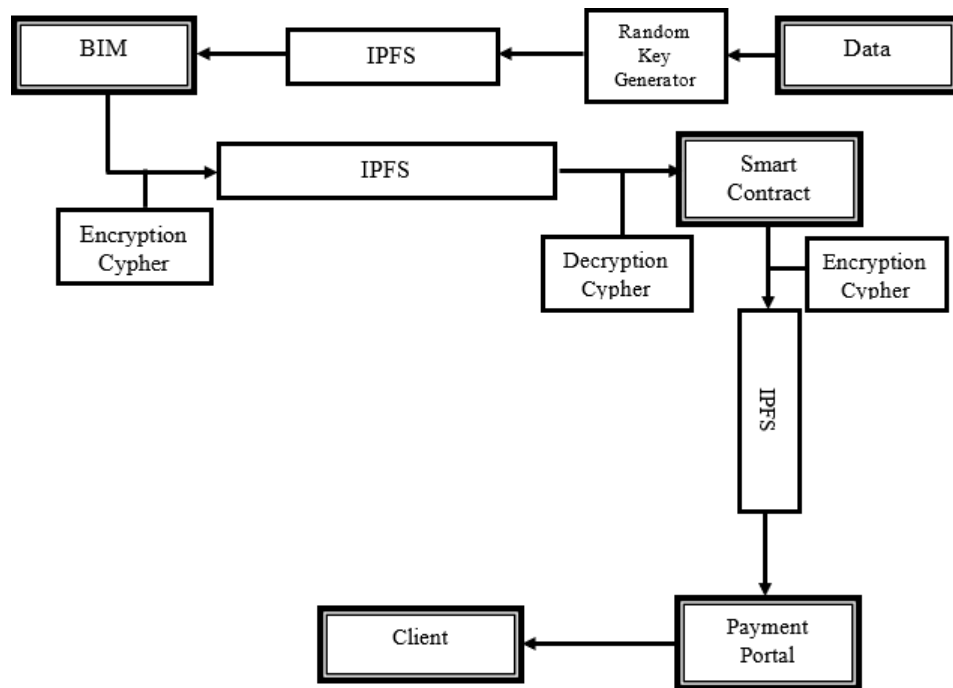


Figure 3.1: System flow diagram

3.1.1 Encryption Flow Diagram

The encryption flow is divided into seven (7) levels.

Level 1: At this level, the supervisor gets the access code from a random key generator.

Level 2: At this level, using the key generated for Level 1 the plan text is put into a polyalphabetic encryption system to generate a cipher text

Level 3: At this level, the cipher text is divided into several segments based on the level of the building, and then transposition encryption is performed.

Level 4: At this level, the cipher text gotten from level 3 is divided into 2; “A” and “B”. “B” is sent to the smart contract.

Level 5: At this level, the cipher text received is decrypted and the smart contract authorizes a payment amount.

Level 6: At this level, the payment order is then added to part “A” from level 4 and then a transposed encryption is performed.

Level 7: At this final level, the final encryption is sent to the payment portal to pay the client.

A pictorial explanation of the proposed algorithm's operation is presented in the encryption flow diagram as shown in figure 3.2.

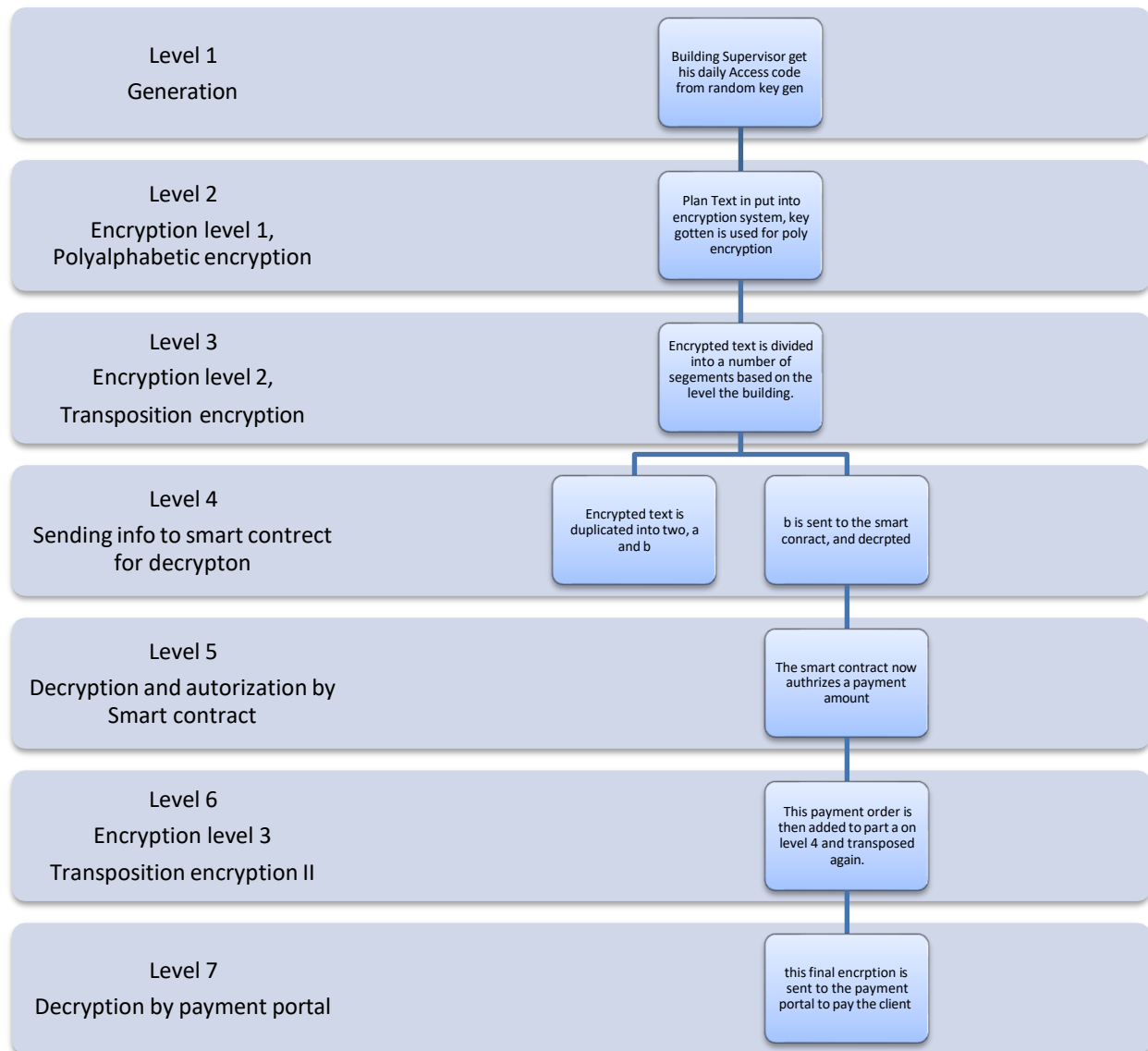


Figure 3.2: Encryption Flow

3.1.2 Algorithm and Mathematical Model

3.1.2.1 Algorithm

ALGORITHM1: DATA ENCRYPTION

Input a: Plaintext

Input b: Building Level

- 1 *P: convert a to its digital equivalent*
- 2 *R: Random string of 1 – 6 is generated without repetition*
- 3 *foreach*
- 4 $Z = \sum_{1-\infty}^P R$
- 5 *end foreach*
- 6 *Km: eliminate digits greater than b in R*
- 7 *using Km, transpose $Z \leftarrow Fx$*
- 8 *Send Fx to Smart contract*

3.1.2.2 Mathematical Model

A random code algorithm developed for the model is shown below;

Random code Algorithm

Random String (Count 1 – 6), Integer must be positive and no repetition allowed

Random Key = (R₁, R₂, R₃, R₄, R₅, R₆)

Encryption Level 1: Polyalphabetic Encryption

Plaintext = (P₁, P₂, P₃, P₄, P₅)

Ef(x) = (P₁+R₁, P₂+R₂, P₃+R₃, ..., P_∞+R_∞)

$$= \sum_{1-\infty}^P R = Y_1, Y_2, Y_3, \dots, Y_n$$

Buildings are divided into levels; Level logged in by supervisor from levels 2 – 5. The number is then used to divide cypher text into the count e.g.; If count is 3, Y₁, Y₂, Y₃ = Y₁¹, Y₂², Y₃³.

The key for this encryption is private and agreed on to the extent of levels 1 – 6, however, sequences are put in levels available. If the Key is 3, 5, 1, 4, 2 and the current count is 3, then the key will equal 3, 1, 2

Cipher Text 2 = Y_3, Y_1, Y_2

Cipher text 2 is sent to the smart contract. After decryption, the smart contract adds another string to the cipher text, P_n, Y_3, Y_1, Y_2, P_n . The same key is then applied again.

Encryption Level 2: Transposition Encryption

Transposition $Y_3^1, Y_1^2, Y_2^3, P_N^4$

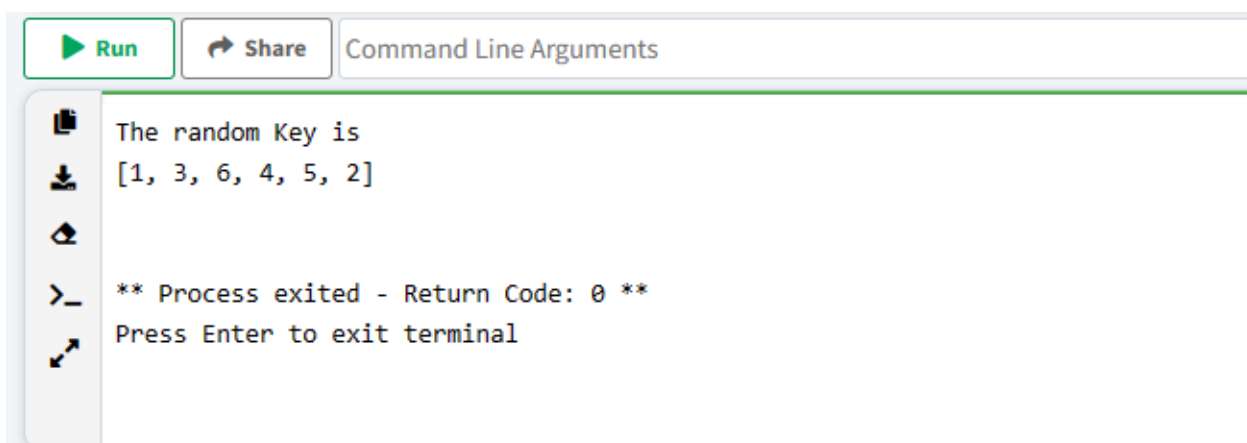
3, 5, 1, 4, 2, 6 \rightarrow 3, 1, 4, 2

$Y_2, Y_3, P_n, Y_1 \rightarrow$ Final cipher text sent to payment portal

3.1.2.3 Simulation

Plaintext = F I R S T F L O O R C O M P L E T E D = 6 9 18 19 20 6 12 15 15 18 3 15 13 16 12
5 20 5 4

Random Key = 1, 3, 6, 4, 5, 2

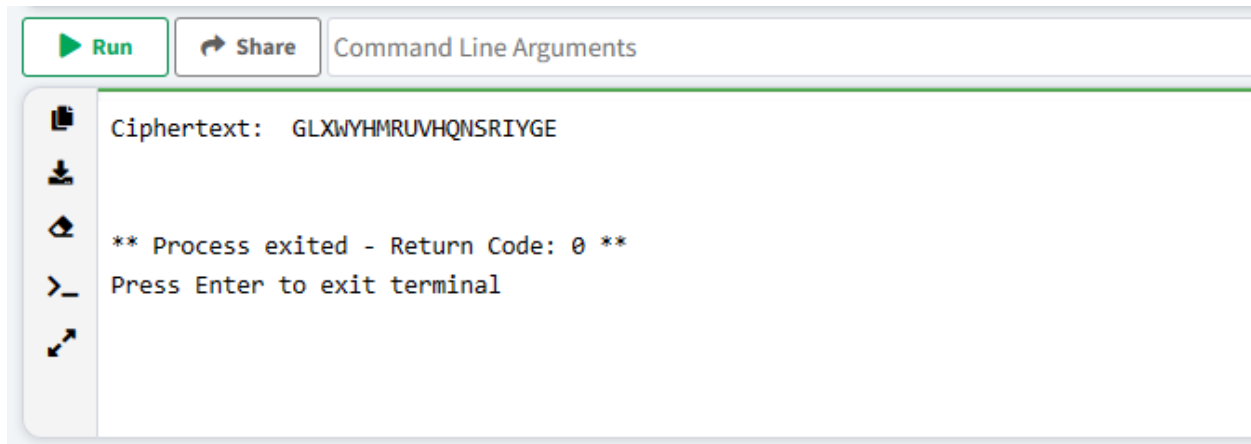


```
The random Key is
[1, 3, 6, 4, 5, 2]

** Process exited - Return Code: 0 **
Press Enter to exit terminal
```

Figure 3.3: Random Key Generation

$Ef(x) = GLXWYHMRUVHQNSRIYGE$



```
Run Share Command Line Arguments

Ciphertext: GLXWYHMRUVHQNSRIYGE

** Process exited - Return Code: 0 **

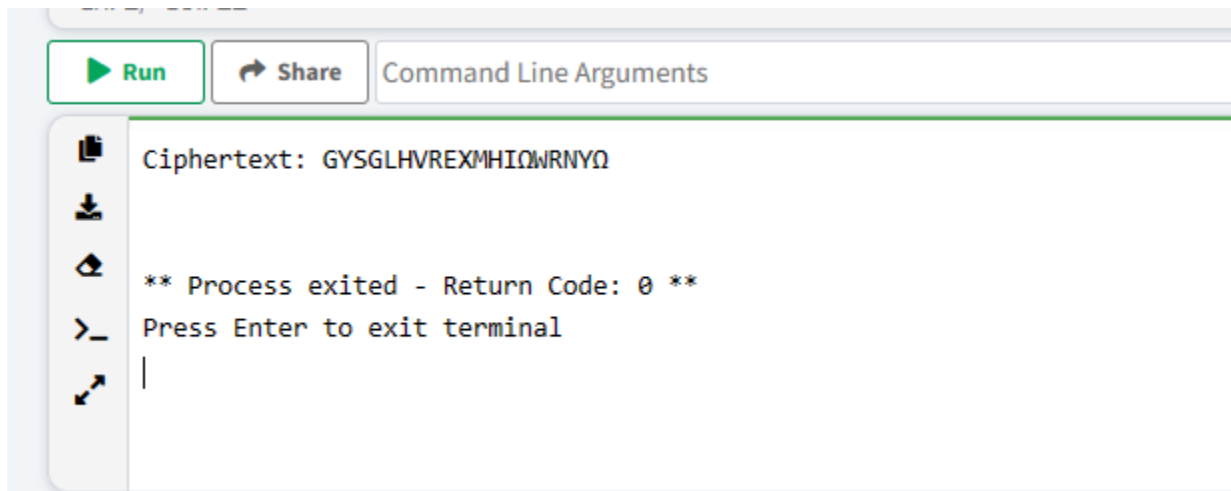
Press Enter to exit terminal
```

Figure 3.4: Polymorphic Encryption

Level logged in by supervisor = 4

From Key 136452, eliminate numbers greater than the Level logged in by supervisor = 1342

Cipher text 2 (Transposition Encryption) = G Y U S G L H V R E X M H I Q W R N Y Q



```
Run Share Command Line Arguments

Ciphertext: GYUSGLHVREXMHIQWRNYQ

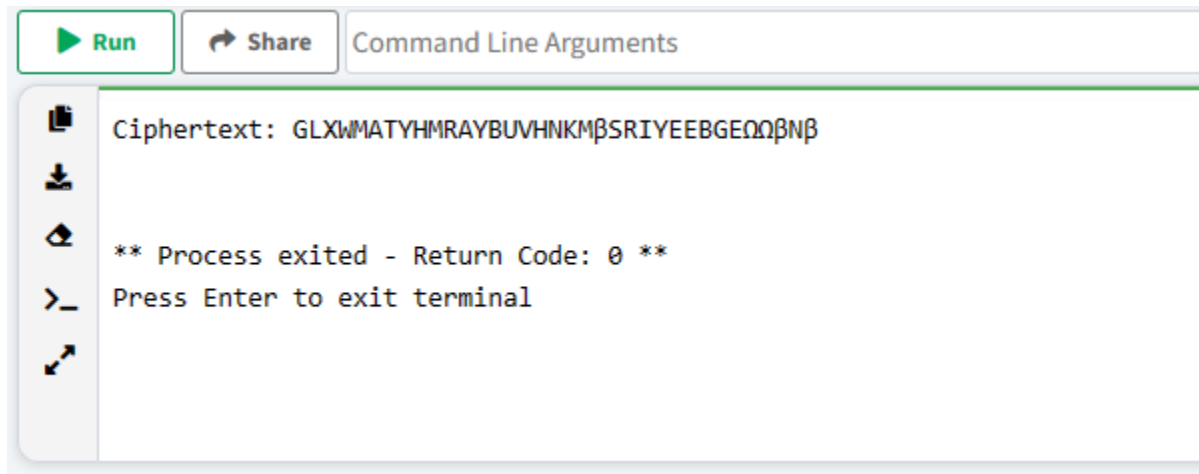
** Process exited - Return Code: 0 **

Press Enter to exit terminal
```

Figure 3.5: Transposition Encryption

The smart contract will now add “**make payment**” and add another level

Final Cipher text = G L X W M A T Y H M R A Y B U V H N K M β S R I Y E E B G E Ω Ω
β N β



```

Run Share Command Line Arguments
Ciphertext: GLXWMATYHMRAYBUVHNKMβSRIYEEBGEΩΩβNβ
** Process exited - Return Code: 0 **
Press Enter to exit terminal

```

Figure 3.6: Final Cipher

We now proceeded to encrypt the final cipher text “firstfloorcompletedmakepayment” using a classical encryption method (transposition cipher) and a modern encryption method as well.

Transposition Cipher Text= rolamflmdafoeptiopmysreketcten

The key used was ZEBRAS

RSA Cipher Text =bT?? ? ?it4ش? %?N?????} ?e ?gH, ?\TAIX,

CHAPTER FOUR

4.0 RESULTS AND DISCUSSION

4.1 Evaluation Criteria

In order to extend IPFS with an access control mechanism, the following information security requirements need to be considered. They are confidentiality, integrity, availability, non-repudiation and authenticity.

Performance and cost testing were done on an HP Laptop machine. The specs of the machine are listed below:

- Operating system: Windows 10
- Processor speed: 1.0 GHz, Core I5
- RAM: 8GB
- Hard drive: 500 GB solid-state drive (SSD)

4.2 Implementation and Results

The developed system was implemented using python 3 (codes in appendix). CrypTool 2.1 (Stable Build 9481.2) was used to test for its performance as multilevel encryption.

CrypTool 2.1 was used because it is specially designed for both beginners and advanced users, comparing encryption algorithms, and it provides a wide range of features for learning and experimenting with encryption, decryption techniques and historical algorithms alongside its support for a wide range of programming languages, including C#, Java, Python, and Visual Basics (VB).

In order to extend IPFS with an access control mechanism, the following information security requirements need to be considered. They are confidentiality, integrity, availability, non-repudiation and authenticity. The model proposed by this research; Multilevel and Multi authority systems (MLMA) alongside two other cryptosystems; transposition cipher and RSA cipher performances were tested against known cryptosystems, Dictionary and Brute force attacks.

Transposition cipher was used because of its simplicity and low computational requirements while the RSA cipher was used because it is a modern cipher, with high security and Wide usage and acceptance. A character placement accuracy and a word placement accuracy showing the results of the tests are shown in table 2 and table 3 respectively.

Table 3.1: Character Placement Accuracy

Cryptosystem	Crypto Analysis Used	The text gotten from cryptanalysis	Percentage Character Placement Accuracy
MLMA	Known Cryptosystem	W X L G T A M R M H Y B Y A N H V U 2 Î M K I R S B E E Y © Î E G 2 Î © Î 2 Î N	0%
Transposition cipher	Known Cryptosystem attack	amemtrcffkyiotolesoleemtrpanpd	13.3%
RSA Cipher	Dictionary Attack	gX?4?}??U? TeAnTI?????? byTHiNt??\	0%
Transposition cipher	Brute Force	rpomlyasmrfelkmedtacftoenptio	6.6 %
Transposition cipher	Dictionary Attack	lfmoypareomtinteerkecomaldpstf	7%

MLMA	Brute Force	GMKB LRMG XAßE WYSΩ MBRΩ AUIß TVYN YHEß HNE	0%
MLMA	Dictionary Attack	❖ G❖ B❖ST YH WMAΩBK AEX E❖V HI ❖ L❖ GNRy ❖ EN MRY ❖UM	0%
RSA Cipher	Brute Force	btT❖,¾Tش❖\❖y❖4❖eTIU❖❖❖❖A❖?%❖gl ❖❖}HX?iN❖, n	0%

Table 3.2: Word Placement Accuracy

Cryptosystem	Crypto Analysis Used	The text gotten from cryptanalysis	Percentage word Placement Accuracy
MLMA	Known Cryptosystem attack	W X L G T A M R M H Y B Y A N H V U²Î M K I R S B E E Y©Î E G ²Î ©Î ²Î N	0%
Transposition cipher	Known Cryptosystem attack	amemtrcffkyiotolesoleemtrpanpd	3%
RSA Cipher	Dictionary Attack	❖,❖❖gX❖❖?❖❖4❖❖}❖❖U❖ ❖❖❖ ❖❖TeAnTl❖❖❖❖❖❖ ❖?❖ 1,%λ❖❖λ ❖byTHiNt❖❖\	0%
Transposition cipher	Brute Force	rpomlyasmrfelkmedtacftoenptio	0%
Transposition cipher	Dictionary Attack	lfmoypareomtinteerkecomaldpstf	0%
MLMA	Brute Force	GMKB LRMG XAßE WYSΩ MBRΩ AUIß TVYN YHEß HNE	0%
MLMA	Dictionary Attack	❖ G❖ B❖ST YH WMAΩBK AEX E❖V HI ❖ L❖ GNRy ❖ EN MRY ❖UM	0%
RSA Cipher	Brute Force	btT❖,¾Tش❖\❖y❖4❖eTIU❖❖❖❖A❖?%❖gl ❖❖}HX?iN❖, n	0%

4.2.1 Discussion of Results

The proposed system of this research ensures confidentiality through the cryptographic mechanism. The system would have additional multi-level Authority and encryption. Brute force was used to test the confidentiality of the system. The results of both the character placement accuracy and the word placement accuracy shown in the table 2 and table 3, and graph in figure 4.1 revealed that the transposition cypher performed poorly, whereas RSA, Multi-Level and Multi Authority both produced the same result and showed no similarities in deciphered text from cryptanalysis or plain text.

The developed model “Multi-Level and Multi Authority” outperformed the transposition cipher in known crypto, dictionary and brute force attacks. However, its performance matches that of RSA in data encryption utilizing fewer resources.

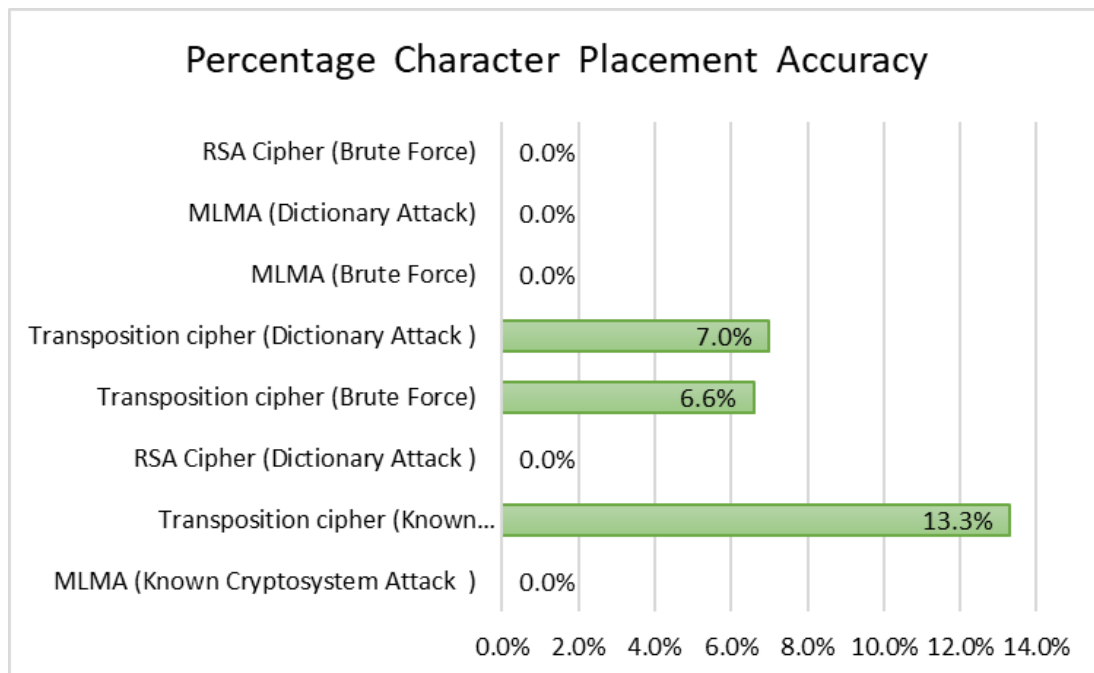


Figure 4.1: Graphical Representation of Result

4.2.2 Integrity, Non-Repudiation and Authenticity

IPFS is designed for the permanent web. It accesses the shared contents by the content hashes. Once the content is changed; the hash of the content also has to be changed. IPFS assign a unique node id for every node in the network and it verifies the sender of the data. Therefore integrity, non-repudiation and authenticity are already implemented within the IPFS protocol. The E-IPFS system also ensures these properties because it works on top of the IPFS (Alwis, 2020).

CHAPTER FIVE

5.0 CONCLUSION AND RECOMMENDATIONS

5.1 Conclusions

The use of BIM and IPFS in a smart contract allows for decentralized and distributed storage and transfer of data, while the use of multi-level authority and encryption will provide additional security measures to protect against eavesdropping, data tampering and theft. This research proposed a cryptographic mechanism, which can be implemented within the IPFS distributed file system. By the proposed mechanism it achieves the information security constraints which are described in the evaluation. The algorithm developed from this research has the potential to revolutionize the way data is transferred and stored, with applications in various industries and contexts. For construction projects that require security but do not require the hassles of a complex system, we would advise using our model rather than the more complicated and resource-intensive RSA scheme.

5.2 Recommendations

some areas of research that can be further explore include:

1. Using the proposed model in this research to encrypt data in the BIM for more security on the BIM.
2. Using the proposed model in this research in other sectors such as Healthcare, finance, document management, etc.

5.3 Contribution to Knowledge

In the context of secure data transfer in the Architecture, Engineering, and Construction (AEC) Industry, this research added a multi-level authority and encryption model to IPFS, blockchain, smart contracts to secure the end-to-end transfer of data between the system. This will provide a robust and decentralized solution for storing, accessing, and transferring sensitive data ensuring integrity, non-repudiation and authenticity. By leveraging the strengths of each of these technologies, it may be possible to create a system that is resistant to tampering, censorship, or unauthorized access.

REFERENCE

- Abdulrahman, A., & Naim, A. L. (2018). *an Investigation of Building Information Modelling Implementation in Ksa*. 2018-06-01. <http://hdl.handle.net/2436/621890>
- Ahmadisheykhsarmast, S., & Sonmez, R. (2020). A smart contract system for security of payment of construction contracts. *Automation in Construction*, 120, 103401. <https://doi.org/10.1016/j.autcon.2020.103401>
- Aina, O. O. (2015). *Evaluation of Building Information Modelling Usage in Construction Industry in Lagos State, Nigeria*. March, 1–60. <https://doi.org/10.13140/RG.2.2.31850.82883>
- Alenezi, M. N., Alabdulrazzaq, H., & Mohammad, N. Q. (2020). Symmetric encryption algorithms: Review and evaluation study. *International Journal of Communication Networks and Information Security*, 12(2), 256–272.
- Ali, I., Gervais, M., Ahene, E., & Li, F. (2019). A blockchain-based certificateless public key signature scheme for vehicle-to-infrastructure communication in VANETs. *Journal of Systems Architecture*, 99(August), 101636. <https://doi.org/10.1016/j.sysarc.2019.101636>
- Alwis, R. A. H. A. De. (2020). *Access level control for shared content in Inter Planetary File System (IPFS)*. University of Colombo School of Computing.
- Austria, P. S. (2020). *Analysis of Blockchain-Based Storage Systems*. <https://www.proquest.com/openview/cb1594c6c3966b0a2c7272c728d807f1/1?pq-origsite=gscholar&cbl=18750&diss=y>
- Barker, E., Barker, W., Burr, W., Polk, W., Smid, M., & Division, C. S. (2012). NIST 800-57: Computer security. *NIST Special Publication 800-57, Revision 3*(July), 1–147.
- Barker, E., Roginsky, A., Locke, G., & Gallagher, P. (2011). Transitions: Recommendation for Transitioning the Use of Cryptographic Algorithms and Key Lengths. *NIST Special Publication, January*, 800–131.
- Bellare, M., & Rogaway, P. (2005). *Introduction to Modern Cryptography Bellare.pdf*. 1–283.
- Boicea, A. (2019). *Cryptographic Algorithms Benchmarking : A Case Study Cryptographic Algorithms Benchmarking : A Case Study*. October 2020.
- Braga, A., Dahab, R., Antunes, N., Laranjeiro, N., & Vieira, M. (2017). Practical Evaluation of Static Analysis Tools for Cryptography: Benchmarking Method and Case Study. *Proceedings - International Symposium on Software Reliability Engineering, ISSRE, 2017-Octob*, 170–181. <https://doi.org/10.1109/ISSRE.2017.27>
- Bruin, L. De. (2019). *Analyzing the Tahoe-LAFS filesystem for privacy friendly replication and file sharing*. August.
- Brun, L. (2017). The role of the European Union Agency for Network and Information Security (ENISA) in the governance strategies of European cybersecurity. *DIAL.Mem*, 67. https://dial.uclouvain.be/memoire/ucl/fr/object/thesis%3A16234/datastream/PDF_01/view
- Burr, W. E. (2003). Selecting the advanced encryption standard. *IEEE Security and Privacy*, 1(2),

43–52. <https://doi.org/10.1109/MSECP.2003.1193210>

- Cardoso, H. L., & Oliveira, E. (2008). A contract model for electronic institutions. *Lecture Notes in Computer Science (Including Subseries Lecture Notes in Artificial Intelligence and Lecture Notes in Bioinformatics)*, 4870 LNAI, 27–40. <https://doi.org/10.1007/978-3-540-79003-7-3>
- Chang, C.-Y., Pan, W., & Howard, R. (2017). Impact of Building Information Modeling Implementation on the Acceptance of Integrated Delivery Systems: Structural Equation Modeling Analysis. *Journal of Construction Engineering and Management*, 143(8). [https://doi.org/10.1061/\(asce\)co.1943-7862.0001335](https://doi.org/10.1061/(asce)co.1943-7862.0001335)
- Chang, S. E., Chen, Y. C., & Lu, M. F. (2019). Supply chain re-engineering using blockchain technology: A case of smart contract based tracking process. *Technological Forecasting and Social Change*, 144, 1–11. <https://doi.org/10.1016/j.techfore.2019.03.015>
- Choi, T. M., Feng, L., & Li, R. (2020). Information disclosure structure in supply chains with rental service platforms in the blockchain technology era. *International Journal of Production Economics*, 221(August), 107473. <https://doi.org/10.1016/j.ijpe.2019.08.008>
- Dahal, R. K., Bhatta, J., & Dhamala, T. N. (2013). Performance Analysis of Sha-2 and Sha-3 Finalists. *International Journal on Cryptography and Information Security*, 3(3), 1–10. <https://doi.org/10.5121/ijcis.2013.3301>
- Dakhli, Z., Lafhaj, Z., & Mossman, A. (2019). *The Potential of Blockchain in Building Construction*.
- Daniel, F., & Guida, L. (2019). A Service-Oriented Perspective on Blockchain Smart Contracts. *IEEE Internet Computing*, 23(1), 46–53. <https://doi.org/10.1109/MIC.2018.2890624>
- Education, M. (2021). *A Study on some modified Classical Ciphers for Secure Crypto-System*. 12(6), 5316–5319.
- Erri Pradeep, A. S., Yiu, T. W., & Amor, R. (2019). Leveraging blockchain technology in a bim workflow: A literature review. *International Conference on Smart Infrastructure and Construction 2019, ICSIC 2019: Driving Data-Informed Decision-Making, August*, 371–380. <https://doi.org/10.1680/icsic.64669.371>
- Feng, J., Zhao, X., Chen, K., Zhao, F., & Zhang, G. (2020). Towards random-honest miners selection and multi-blocks creation: Proof-of-negotiation consensus mechanism in blockchain networks. *Future Generation Computer Systems*, 105, 248–258. <https://doi.org/10.1016/j.future.2019.11.026>
- Forsyth, W. S., & Safavi-Naini, R. (1993). Automated cryptanalysis of substitution ciphers. *Cryptologia*, 17(4), 407–418. <https://doi.org/10.1080/0161-119391868033>
- Frizzo-Barker, J., Chow-White, P. A., Adams, P. R., Mentanko, J., Ha, D., & Green, S. (2020). Blockchain as a disruptive technology for business: A systematic review. *International Journal of Information Management*, 51(April), 0–1. <https://doi.org/10.1016/j.ijinfomgt.2019.10.014>
- Gabert, H. (2018). *Blockchain and smart contracts in the Swedish construction industry - An interview study on smart contracts and services*. <http://kth.diva-portal.org/smash/record.jsf?pid=diva2%3A1229491&dswid=6654>

- Guo, L., Liu, Q., Shi, K., Gao, Y., Luo, J., & Chen, J. (2021). A blockchain-driven electronic contract management system for commodity procurement in electronic power industry. *IEEE Access*, 9, 9473–9480. <https://doi.org/10.1109/ACCESS.2021.3049562>
- Hamledari, H., Davari, S., Azar, E. R., McCabe, B., Flager, F., & Fischer, M. (2018). UAV-enabled Site-to-BIM automation: Aerial robotic- and computer vision-based development of As-Built/As-Is BIMs and quality control. *Construction Research Congress 2018: Construction Information Technology - Selected Papers from the Construction Research Congress 2018, 2018-April(March)*, 336–346. <https://doi.org/10.1061/9780784481264.033>
- Hamledari, H., & Fischer, M. (2021a). Construction Payment Automation Using Blockchain-Enabled Smart Contracts and Reality Capture Technologies By. *Automation in Construction*, 132, 103926.
- Hamledari, H., & Fischer, M. (2021b). Role of Blockchain-Enabled Smart Contracts in Automating Construction Progress Payments. *Journal of Legal Affairs and Dispute Resolution in Engineering and Construction*, 13(1), 04520038. [https://doi.org/10.1061/\(asce\)la.1943-4170.0000442](https://doi.org/10.1061/(asce)la.1943-4170.0000442)
- Hamza, A., & Kumar, B. (2020, December). A review paper on DES, AES, RSA encryption standards. In 2020 9th International Conference System Modeling and Advancement in Research Trends (SMART) (pp. 333-338). IEEE.
- Hargaden, V., Newell, A., Khavia, A., & Scanlon, A. (2019). *The Role of Blockchain Technologies in Construction Engineering Project Management*.
- Huang, H., Lin, J., & Zheng, B. (2020). When Blockchain Meets Distributed File Systems : An Overview , Challenges , and Open Issues. *IEEE Access*, 8, 50574–50586. <https://doi.org/10.1109/ACCESS.2020.2979881>
- Hunhevicz, J. J., & Hall, D. M. (2020). Do you need a blockchain in construction? Use case categories and decision framework for DLT design options. *Advanced Engineering Informatics*, 45(November 2019), 101094. <https://doi.org/10.1016/j.aei.2020.101094>
- Idelberger, F., Governatori, G., Riveret, R., & Sartor, G. (2016). Evaluation of logic-based smart contracts for blockchain systems. *Lecture Notes in Computer Science (Including Subseries Lecture Notes in Artificial Intelligence and Lecture Notes in Bioinformatics)*, 9718, 167–183. https://doi.org/10.1007/978-3-319-42019-6_11
- IPFS. (2021, March 10). *IPFS Documentation*. [Online documentation]. IPFS. <https://docs.ipfs.io/>.
- Jabarulla, M. Y., Jung, G., Lee, H., & Member, S. (2017). *Decentralized Framework for Medical Images Based on Blockchain and Inter Planetary File System*. 24(6).
- Kopal, N. (2018, June). Solving classical ciphers with CrypTool 2. In Proceedings of the 1st International Conference on Historical Cryptology HistoCrypt 2018 (No. 149, pp. 29-38). Linköping University Electronic Press.
- Kouroumbashev, P. (2019). EU Legislation in Progress ENISA and a new cybersecurity act. *European Parliamentary Research Service*.
- Kumar, B., Kaday, A., Jadhav, D., & Awani, S. (2019). Blockchain based Secure Data Storage.

- Laan, A., Noorderhaven, N., Voordijk, H., & Dewulf, G. (2011). Building trust in construction partnering projects: An exploratory case-study. *Journal of Purchasing and Supply Management*, 17(2), 98–108. <https://doi.org/10.1016/j.pursup.2010.11.001>
- Leka, E., & Selimi, B. (2021). Development and evaluation of blockchain based secure application for verification and validation of academic certificates. *Annals of Emerging Technologies in Computing*, 5(2), 22–36. <https://doi.org/10.33166/AETiC.2021.02.003>
- Li, J., Greenwood, D., & Kassem, M. (2019). Automation in Construction Blockchain in the built environment and construction industry : A systematic review , conceptual models and practical use cases. *Automation in Construction*, 102(January), 288–307. <https://doi.org/10.1016/j.autcon.2019.02.005>
- Lim, M. K., Li, Y., Wang, C., & Tseng, M. (2021). Computers & Industrial Engineering A literature review of blockchain technology applications in supply chains : A comprehensive analysis of themes , methodologies and industries. *Computers & Industrial Engineering*, 154(July 2020), 107133. <https://doi.org/10.1016/j.cie.2021.107133>
- Liu, Y., Sun, G., & Schuckers, S. (2019). Enabling Secure and Privacy Preserving Identity Management via Smart Contract. *2019 IEEE Conference on Communications and Network Security, CNS 2019*. <https://doi.org/10.1109/CNS.2019.8802771>
- Luo, H., & Cheng, J. C. P. (2019). Construction Payment Automation through Smart Contract-based Blockchain Construction Payment Automation through Smart Contract-based Blockchain Framework. *IAARC Publications*, 36, 1254–1260. <https://doi.org/10.22260/ISARC2019/0168>
- Mallik, A., Ahsan, A., Shahadat, M. M. Z., & Tsou, J. C. (2019). Man-in-the-middle-attack: Understanding in simple words. *International Journal of Data and Network Science*, 3(2), 77–92. <https://doi.org/10.5267/j.ijdns.2019.1.001>
- Martínez-aires, M. D., López-alonso, M., & Martínez-rojas, M. (2018). Building information modeling and safety management : A systematic review. *Safety Science*, 101(February 2017), 11–18. <https://doi.org/10.1016/j.ssci.2017.08.015>
- Mathews, M., Bowe, B., & Robles, D. (2017). BIM+Blockchain: A Solution to the Trust Problem in Collaboration? | Enhanced Reader. *CITA BIM Gathering 2017*. <https://arrow.tudublin.ie/bescharcon%0Amoz-extension://db27a32d-d4bf-c540-b67f-85ad2caf9dba/enhanced-reader.html?openApp&pdf=https%3A%2F%2Farrow.tudublin.ie%2Fcgi%2Fviewcontent.cgi%3Farticle%3D1032%26context%3Dbescharcon>
- Mik, E. (2017). Smart contracts: terminology, technical limitations and real world complexity. *Law, innovation and technology*, 9(2), 269-300.
- Nael, G., B. (2005). *The FIDIC forms of contract* (Third Edit). Blackwell Publishing Ltd. https://books.google.com.ng/books?hl=en&lr=&id=hk6gDodpTtsC&oi=fnd&pg=PT16&dq=Bunni+N.+G.+The+FIDIC+forms+of+contract.+Blackwell+Pub,+2005.&ots=gtOctNhWjr&sig=VEIYHyCIpd141zuMRFRWoSEPByM&redir_esc=y#v=onepage&q&f=false

- Nakamoto, S. (2008). Bitcoin : A Peer-to-Peer Electronic Cash System. *Decentralized Business Review*, 21260.
- Nanayakkara, S., Perera, S., Senaratne, S., & Weerasuriya, G. T. (2021). Blockchain and Smart Contracts : A Solution for Payment Issues. *Multidisciplinary Digital Publishing Institute*, 8(2), 36.
- Nawari, N. O., & Ravindran, S. (2019). Blockchain and Building Information Modeling (BIM): Review and applications in post-disaster recovery. In *Buildings* (Vol. 9, Issue 6). <https://doi.org/10.3390/BUILDINGS9060149>
- Nechvatal, J., Barker, E., Bassham, L., Burr, W., Dworkin, M., Foti, J., & Roback, E. (2001). Report on the development of the Advanced Encryption Standard (AES). *Journal of Research of the National Institute of Standards and Technology*, 106(3), 511–577. <https://doi.org/10.6028/jres.106.023>
- Nguyen, Q. K. (2016). *Blockchain – A Financial Technology For Future Sustainable Development*. <https://doi.org/10.1109/GTSD.2016.22>
- Nizamuddin, N., & Abugabah, A. (2021). Blockchain for automotive: An insight towards the IPFS blockchain-based auto insurance sector. *International Journal of Electrical and Computer Engineering*, 11(3), 2443–2456. <https://doi.org/10.11591/ijece.v11i3.pp2443-2456>
- Owusu, E. K., Chan, A. P. C., & Nani, G. (2020). *A Turn to Smart Contracts and Future Applications towards Construction Innovation : A Hybrid-Metric Review*. November. <https://doi.org/10.1061/9780784482865.027>
- Ozturan, M., Atasu, I., & Soydan, H. (2019). *Assessment of Blockchain Technology Readiness Level of Banking Industry : Case of Turkey International Journal of Business Marketing and Management (IJBMM)*. 4(12), 1–13.
- Penzes, B., KirNup, A., Gage, C., Dravai, T., & Colmer, M. (2018). Blockchain technology in the construction industry: Digital transformation for high productivity. *Institution of Civil Engineers (ICE)*.
- Pouwelse, J. A., Garbacki, P., Epema, D. H. J., & Sips, H. J. (2004). THE BITTORRENT P2P FILE-SHARING SYSTEM : MEASUREMENTS AND ANALYSIS J . A . Pouwelse , P . Garbacki , D . H . J . Epema , H . J . Sips Department of Computer Science , Delft University of Technology , the Netherlands. *Computer*, Oct, 205–216.
- Princy, P. (2015). a Comparison of Symmetric Key Algorithms Des , Aes , Blowfish ., *International Journal of Computer Science & Engineering Technology (IJCSET)*, 6(05), 328–331. <http://www.ijcset.com/docs/IJCSET15-06-05-055.pdf>
- Raghavendra, M., & Kiran, M. (2020). *Inter-Planetary File System Enabled Blockchain Solution For Securing Healthcare Records*. 171–178. <https://doi.org/10.1109/ISEA-ISAP49340.2020.235016>
- Ratnadewi, Adhie, R. P., Hutama, Y., Saleh Ahmar, A., & Setiawan, M. I. (2018). Implementation Cryptography Data Encryption Standard (DES) and Triple Data Encryption Standard (3DES) Method in Communication System Based Near Field Communication (NFC). *Journal of Physics: Conference Series*, 954(1). <https://doi.org/10.1088/1742-6596/954/1/012009>

- Schrödel, T. (2008). Breaking short Vigenere ciphers. *Cryptologia*, 32(4), 334–347. <https://doi.org/10.1080/0161190802336097>
- Senthil Kumaran, U., Nallakaruppan, M. K., & Senthil Kumar, M. (2016). Review of asymmetric key cryptography in wireless sensor networks. *International Journal of Engineering and Technology*, 8(2), 859–862.
- Shojaei, A., Flood, I., & Moud, H. I. (2019). An Implementation of Smart Contracts by Integrating BIM and Blockchain. *Proceedings of the Future Technologies Conference, October*, 519–527. <https://doi.org/10.1007/978-3-030-32523-7>
- Shojaei, A., Flood, I., & Moud, H. I. (2020). *An Implementation of Smart Contracts by Integrating BIM and Blockchain. 1*, 519–527.
- Sillaber, C., & Wlatl, B. (2017). Life Cycle of Smart Contracts in Blockchain Ecosystems. *Datenschutz Und Datensicherheit - DuD*, 41(8), 497–500. <https://doi.org/10.1007/s11623-017-0819-7>
- Sklavos, N., & Koufopavlou, O. (2003). On the Hardware Implementations of the SHA-2(256, 384, 512) Hash Algorithms. *Circuits and Systems International Symposium*, 5(256), 153–156.
- Sofia, T., Dimitrios, T., Konstantinos, V., & Kelly, C. (2019). *Blockchain 3.0 Smart Contracts in E-Government 3.0 Applications*. https://www.researchgate.net/publication/336551126_Blockchain_30_Smart_Contracts_in_E-Government_30_Applications
- Sonmez Turan, M., McKay, K., Chang, D., Calik, C., Bassham, L., Kang, J., Kelsey, J., Alagic, G., Alperin-Sheriff, J., Apon, D., Cooper, D., Dang, Q., Liu, Y.-K., Miller, C., Moody, D., Peralta, R., Perlner, R., Robinson, A., & Smith-Tone, D. (2021). NISTIR 8268 Status: Status Report on the First Round of the NIST Lightweight Cryptography Standardization Process. *Nistir* 8309, 1–27. <https://nvlpubs.nist.gov/nistpubs/ir/2019/NIST.IR.8268.pdf%0Ahttps://nvlpubs.nist.gov/nistpubs/ir/2021/NIST.IR.8369.pdf%0Ahttps://nvlpubs.nist.gov/nistpubs/ir/2020/NIST.IR.8309.pdf>
- Stallings, W. (2017). The principles and practice of cryptography and network security 7th edition, isbn-10: 0134444280. *Pearson Education*, 20, 7.
- Subramanian, H. (2018). Decentralized Blockchain-based electronic marketplaces. *Communications of the ACM*, 61(1), 78–84. <https://doi.org/10.1145/3158333>
- Suralkar, S., Mujumdar, A., Masiwal, G., & Kulkarni, M. (2013). *Review of Distributed File Systems : Case Studies*. 3(1), 1293–1298.
- Tian, F. (2016). An agri-food supply chain traceability system for China based on RFID & blockchain technology. *2016 13th International Conference on Service Systems and Service Management, ICSSSM 2016*. <https://doi.org/10.1109/ICSSSM.2016.7538424>
- Turk, Ž., & Klinc, R. (2017). Potentials of Blockchain Technology for Construction Management. *Procedia Engineering*, 196(June), 638–645. <https://doi.org/10.1016/j.proeng.2017.08.052>
- Underwood, S. (2016). Blockchain beyond bitcoin. *Communications of the ACM*, 59(11), 15–17. <https://doi.org/10.1145/2994581>

- Valdeolmillos, D., & Mezquita, Y. (2009). *Blockchain Technology : A Review of the Current Challenges of Cryptocurrency. 1*, 153–160. <https://doi.org/10.1007/978-3-030-23813-1>
- Vashistha, M., & Ferdous, Ahmed, B. (2020). Document Management System using Blockchain and Inter Planetary File System. *BSCI Poster Session*, 212–213. <http://arxiv.org/abs/1709.10000>
- Wang, H., Qin, H., Zhao, M., Wei, X., Shen, H., & Susilo, W. (2020). Blockchain-based fair payment smart contract for public cloud storage auditing. *Information Sciences*, 519(88), 348–362. <https://doi.org/10.1016/j.ins.2020.01.051>
- Wang, Y., Yang, J., & Shen, Q. (2007). The application of electronic commerce and information integration in the construction industry. *International Journal of Project Management*, 25(2), 158–163. <https://doi.org/10.1016/j.ijproman.2006.09.008>
- Xu, M., Chen, X., & Kou, G. (2019). A systematic review of blockchain. In *Financial Innovation* (Vol. 5, Issue 1). Financial Innovation. <https://doi.org/10.1186/s40854-019-0147-z>
- Yaga, D., Mell, P., Roby, N., & Scarfone, K. (2019). Blockchain Technology Overview. In *arXiv*. <https://doi.org/10.6028/NIST.IR.8202>
- Yang, W. (2022, April). ECC, RSA, and DSA analogies in applied mathematics. In *International Conference on Statistics, Applied Mathematics, and Computing Science (CSAMCS 2021)* (Vol. 12163, pp. 699-706). SPIE.
- Yao, Y., Zeng, X., Cao, T., Fu, L., & Wang, X. (2019). APRP : An Anonymous Propagation Method in Bitcoin Network. *Proceedings of the AAAI Conference on Artificial Intelligence*, 33(01), 10073–10074.
- Ye, X., Sigalov, K., & König, M. (2020). Integrating BIM- and cost-included information container with Blockchain for construction automated payment using billing model and smart contracts. *ISARC. Proceedings of the International Symposium on Automation and Robotics in Construction (ISARC 2020)*, 37, 1388–1395.
- Yusfrizal, Y., Meizar, A., Kurniawan, H., & Agustin, F. (2019). Key Management Using Combination of Diffie-Hellman Key Exchange with AES Encryption. *2018 6th International Conference on Cyber and IT Service Management, CITSM 2018, Citsm*, 1–6. <https://doi.org/10.1109/CITSM.2018.8674278>
- Zhang, Gao, D. and, & Zhili. (2013). Project time and cost control using building information modeling (BIM). *ICCREM 2013: Construction and Operation in the Context of Sustainability, November*, 545--554.
- Zheng, K., Zhang, Z., Chen, Y., & Wu, J. (2021). Blockchain adoption for information sharing: risk decision-making in spacecraft supply chain. *Enterprise Information Systems*, 15(8), 1070–1091. <https://doi.org/10.1080/17517575.2019.1669831>
- Zheng, Z., Xie, S., Dai, H. N., Chen, W., Chen, X., Weng, J., & Imran, M. (2020). An overview on smart contracts: Challenges, advances and platforms. *Future Generation Computer Systems*, 105, 475–491. <https://doi.org/10.1016/j.future.2019.12.019>

APPENDIX

Random Key Generator

```
import random

numbers = []
while len(numbers) < 6:
    new_number = random.randint(1, 6)

    if
        new_number not in numbers:
            numbers.append(new_number)

print(numbers)
```

Polymorphic Encryption

```
import itertools

def encrypt(plaintext):
    key = "136452"
    key_cycle = itertools.cycle(key)
    ciphertext = ""
    for c, k in zip(plaintext, key_cycle):

        ciphertext += chr((ord(c) + int(k)) % 256)

    return ciphertext

def decrypt(ciphertext):
    key = "136452"
    key_cycle = itertools.cycle(key)
    plaintext = ""
    for c, k in zip(ciphertext, key_cycle):
        plaintext += chr((ord(c) - int(k) + 256) % 256)

    return plaintext

plaintext = " FIRSTFLOORCOMPLETED"
ciphertext = encrypt(plaintext)

print("Ciphertext: ", ciphertext)
```

Transposition Encryption

```
import itertools
```

```

def encrypt(plaintext, key):
    key = [int(d) for d in str(key)]

    ciphertext = [""] * len(key)
    for col in key:
        pointer = col - 1
        while pointer < len(plaintext):
            ciphertext[col - 1] += plaintext[pointer]
            pointer += len(key)

    return "".join(ciphertext)

def decrypt(ciphertext, key):
    key = [int(d) for d in str(key)]
    n = len(ciphertext)
    r = len(key)
    c = n // r
    plaintext = [""] * c
    col = 0
    row = 0
    for i in range(1, r+1):
        for j in range(i-1, n, r):
            if key.index(i) < len(plaintext):
                plaintext[key.index(i)] += ciphertext[j]

    return "".join(plaintext)

key = 3142
plaintext = "GLXWYHMRUVHQNRSRIYGE"
ciphertext = encrypt(plaintext, key)

print("Ciphertext: ", ciphertext)

```

Final Encryption

```

import math

def transpose_cipher(plaintext, key):
    # Determine the number of rows and columns needed for the grid
    columns = len(key)
    rows = math.ceil(len(plaintext) / columns)

    # Create an empty grid with the appropriate number of rows and columns
    grid = [['_' for i in range(columns)] for j in range(rows)]

    # Fill the grid with the plaintext

```

```

current_char = 0
for row in range(rows):
    for col in range(columns):
        if current_char < len(plaintext):
            grid[row][col] =
                plaintext[current_char]
            current_char
                += 1

# Reorder the columns based on the
key new_order = [int(i) for i in key]
grid = [list(col) for col in zip(*grid) if new_order.index(i) in new_order]

# Read the message from the grid to create the ciphertext
ciphertext = ".join([".join(row) for row in grid])

```

IPFS Nodes

→ IPFS ipfs refs local

```

QmdL9t1YP99v4a2wyXFYAQJtbD9zKnPrugFLQWXBxb82s
n
QmZTR5bcpQD7cFgTorqxZDYaew1Wqgfb2ud9QqGPAkK2
V
QmXgqKTbzdh83pQtKFb19SpMCpDDcKR2ujqk3pKph9aCN
F
QmS4ustL54uo8FzR9455qaxZwuMiUhyvMcX9Ba8nUH4uVv
Qma4NNR8dUSDt2BvLYYtgDMLF8J3usKrT9kDFhHzfpB7oq
QmNcNo8TXi92Da91fDfzCMbYF5ScaHEJmQG1jqCEbkS7K
t
QmYCvbfNbCwFR45HiNP45rwJgvatpiW38D961L5qAhUM5
Y
QmejvEPop4D7YUadeGqYWmZxHhLc4JBUCzJJHWMzdcM
e2y
QmPhk6cJkRcFfZCdYam4c9MKYjFG9V29LswUnbrFNhtk2S
QmY5heUM5qgRubMDD1og9fhCPA6QdkMp3QCwd4s7gJsy
E7
QmSKboVigcD3AY4kLsob117KJcMHvMUu6vNFqk1PQzYU
pp
QmQ5vhrL7uv6tuoN9KeVBwd4PwfQkXdVVmDLUZuTNxqg
vm
QmZZRTyhDpL5Jgift1cHbAhexeE1m2Hw8x8g7rTcPahDvo
Qme7RW9zfGgYujt6CJ5yKiMkvV9zPSSkbB4hgipee3j6S
QmYQoke9bEqzBLWPGqyjhUYc3TwBEkn4wed2kUmAbxvL
Fu
QmdfTbBqBPQ7VNxZEYEj14VmRuZBkqFbiwReogJgS1zR1
n

```

```
import ipfsapi
if __name__ == '__main__':
    # Connect to local
    node try:
        api = ipfsapi.connect('127.0.0.1', 5001)
        print(api)
    except ipfsapi.exceptions.ConnectionError as ce:
        print(str(ce))
```