7
www.editada.org

International Journal of Combinatorial Optimization Problems and Informatics, 14(3), Sep-Dec 2023, 71-90. ISSN: 2007-1558. https://doi.org/10.61467/2007.1558.2023.v14i3.404

Analysis of the Feasibility of Smart Contracts in Mexico's Legal and Social Framework: A Study on the Future of Trade Agreements

Joel Silos Sánchez, Uriel Amado Ramírez Hernández, Yaneth Reyes Hernández, Luis Arturo Ortiz Suarez, Luis José Gómez Pérez, Daniel Cruz Rojano, Francisco Rafael Trejo Macotela, Daniel Robles Camarillo

Universidad Politécnica de Pachuca, México.
E-mails: ur763943@gmail.com, joel.siloss@gmail.com

Abstract. Smart contracts are a type of software program that facilitate, ensure, enforce, and execute agreements recorded between two or more parties. An example of this could be an agreement between individuals or organizations responsible for exporting and importing products. This study aims to determine whether smart contracts are a viable tool for use in Mexico, considering the legal and social framework.
Keywords: Smart Contract, Technology, Contracts, Traditional Contracts, Business.

Article Info
Received 11 Sep, 2023
Accepted 22 Dec, 2023

1 Introduction

Cryptographic algorithms are one of the most widely used algorithms today, their main function is to strengthen the security of a message or file using one or more authentication methods, called keys, usually of two types, private and public, cryptographic algorithms focus on the confidentiality of the message, the integrity of the message and the authenticity of the sender.

HASH algorithms start with input information of indeterminate length and output a code, which can be considered unique to some extent for each input. The function of these algorithms is deterministic, i.e. the same input always gives the same output. Simple, but very vulnerable examples are check digits and CRC (Cyclic Redundancy Code) (Palacios & Delgado, 2006).

Smart Contracts are a software programme that facilitates, secures, enforces and executes registered agreements between two or more parties, an example of this would be an agreement between a person or organization in charge of exporting and importing products.

In this paper aims to determine whether smart contracts are a viable tool for use in Mexico in terms of both the legal and social framework. As such, Smart Contracts would assist in the negotiation and definition of such agreements that will cause certain actions to happen as a result of a specific, pre-agreed set of conditions being met or breached (Ramírez, 2019).

2. Contracts

However, the modern economy together with capitalism, have originated the need to handle all contracts in writing and under certain legal certifications, where the rights, obligations and responsibilities of the parties involved are stipulated in a clear and orderly manner.

In Mexico and in the world, the need has arisen for a legal regulation for contracts that seeks fairness in transactions, as well as autonomy of will under the conception that those who intervene in the same enjoy the power to stipulate various clauses or modalities or simply adapt to the type of contracts regulated by law.

Technology has revolutionized financial services systems and even, to some extent, some legal systems, for example, the implementation of electronic signatures or e-signatures.

2.1 Blockchain

In order to understand how smart contracts work, it is essential to understand how the blockchain works, which "is a technology that allows the transfer of digital data with highly sophisticated encryption and in a completely secure manner", i.e. it is a text of digital events, this transfer or procedure does not require a centralized intermediary (Ramírez, 2019).

The blockchain is ideal as it provides immediate, shared and fully transparent data that is stored in an immutable or unalterable ledger that only authorised members can access.

The prosperity that technology creates is no longer greater than the intimacy it destroys. Yet in the digital age we live in, technology is at the centre of almost everything, for better and for worse. It allows us to value and violate the rights of others as never before (Tapscott & Tapscott, 2017).

2.2 Cryptography

It is responsible for studying the procedures or methods for modifying data and then securing it in a confidential manner, so that only authorised persons have access to the information, which ensures that only a legitimate document is opened and cannot be altered to any other modification once it has been established, thus guaranteeing the security of a contract or minutes of agreements.

A cryptographic algorithm is characterised by converting a clear text into another, so-called cipher text. The content of the information is the same as above but can only be understood by the authorised person (Fulgueira-Camilo, Hernández-Duany, & Henry-Fuenteseca, 2015).

In general, cryptographic algorithms can be classified and characterised as follows according to the article by (Castillo Rubí, Santana de la Cruz, Díaz Lobaton, & Almanza, 2011):

- Secret key cryptography or symmetric cryptography*:* This type of cryptography consists of having two keys, which are used to encrypt or decrypt, however by having one key, knowing the encryption key you can predict what the decryption key will be, and vice versa by knowing one you can decrypt the other. Two kinds of symmetric schemes are known which are block ciphers and byte ciphers, which are encrypted block by block and byte by byte or bit by bit.
- Public key cryptography or asymmetric cryptography. This cryptography arises as a solution to the key distribution problem of symmetric cryptography, which is to find an efficient method to agree on the exchange of secure keys. As a solution, a very complex mathematical formula was implemented to create the keys for data encryption and data decryption, allowing authentication and high data protection.
- HASH or summary algorithms. Hashing is an operation that is performed on a set of data, allows for easy searching, and is considered secure as it withstands attacks of all kinds as it has been impenetrable for the time being (Castillo Rubí, Santana de la Cruz, Díaz Lobaton, & Almanza, 2011).

Algorithm	Purpose	Key Range	Date of Creation	Remarks
AES	Encription	128, 192 y 256 bits	2001	Also known as Rijdael, it is a block cipher scheme, adopted as a standard by NITS as FIPS PUB 197. This algorithm encrypts blocks of length 128, 192 or 256 bits with the characteristic that defined the block size; it uses all keys of the same length.
MDS	Hashing	128 bits	1992	One-way hashing function, producing a 128-bit result. The result of the algorithm is 4 blocks of 32 bits that form a block of 128 bits. It has an algorithmic complexity of 228.
SHA-1	Hashing	160 - 264 bits	1994	This type of algorithm works with an algorithmic complexity of 228. This algorithm is based on principles similar to those used in MDS.
FAMILY SHA (SHA 224, SHA 256, SHA 384, SHA 512)	Hashing	256 - 312 bits	1994	The SHA family is a system of cryptographic hash functions published by NIST (FIPS 180-4), based on a somewhat modified design and increased output ranges. It differs from SHA-1 in that the algorithm includes some additional constants, as well as a different digest size and number of rounds.
HMAC	Hashing	128 - 160 bits		It is a function that uses the Hashes seen and an authentic secret key to two users by means of secret key systems. HMAC is a MAC defined in FIPS 198 and constructed using a cryptographic hash algorithm. The strength of HMAC depends on the strength of the Hash algorithm and the entropy length of the secret key.
RIPEMD	Hashing	128 - 320 bits	1996	It is an algorithm developed in Europe. It is based on the design principles of MD4 and is similar in security and performance to SHA-1. The 256-bit and 320-bit versions only decrease the possibility of collisions and do not have higher levels of security than RIPEMD-128 and RIPEMD-160.
Tiger	Hashing	192 bits	1996	It is an efficiency prediction hash function for 64-bit platforms. Tiger is designed using the Merklen-Damgard program, the compression function uses a combination of mixed operations with XQR and addition/subtraction, rotations and search in the S-Box. Optimized for 64-bit machines

The following table shows the family of Hash algorithms (Belmont, R. 2016).

3 Smart Contracts

Smart Contracts are defined as contractual agreements between two or more parties that are self-executing. In Jet Raskin's words, they are contracts whose execution is automated, however, this is not a new or complex concept; it can be compared to a soda vending machine, the machine provides the requested products as long as the buyer complies with the requirements of the transaction, so smart contracts will only come into legal existence if the parties comply with all the enabling requirements (Ramírez, 2019).

The following;

- Speed and accuracy; being a digital application, the contract is executed immediately, can be automated and thus avoids the paperwork process.
- Transparency; because the smart contract has encryption methods, the fidelity of the information is guaranteed, i.e. it is known that it is not being altered.
- Reduced costs; smart contracts do not require the involvement of a third party so there are no extra costs related to additional transactions and paperwork.

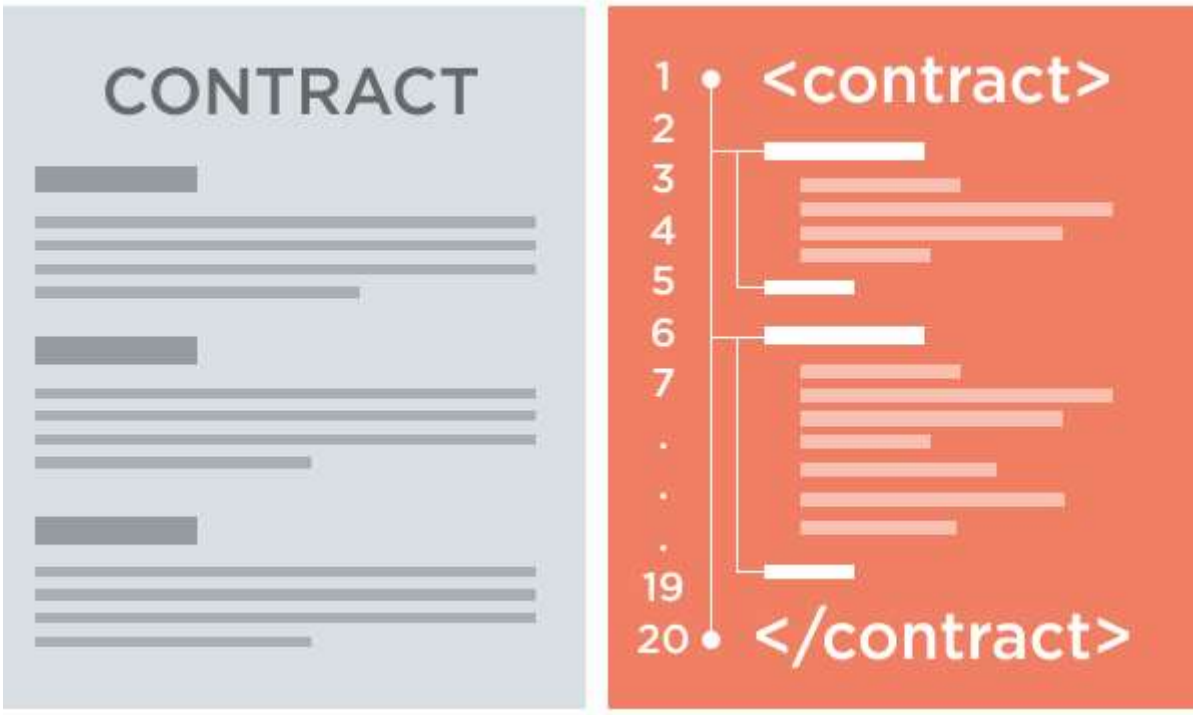
Although the main features of these contracts have already been mentioned above, their advantages are as follows according to Franco (2023).

- Self-executing: this means that, if a condition stipulated in the contract occurs, the agreed consequence will necessarily be activated automatically, without the need for human intervention. This characteristic is inherent or inherent to this type of contract since the programming with which they are designed is based on logical conditions.
- Decentralisation: refers to the fact that the network on which the Smart Contract is built is "managed, stored and guarded by multiple computers/person[s]"75 This characteristic allows Smart Contracts to have not only one or two copies of the contract, but all participants, i.e. all nodes, will have a copy of the contract, as the information is shared with everyone. This also means that the parties do not require other intermediaries for the conclusion of the contract, such as a notary, witnesses, judges, etc.
- Transparency: transactions are visible to all nodes at all times, so parties are certain that they could not have been hidden at some point to be modified. The use of blockchain allows transactions to be traceable, as there is a digital trace at all times.
- Immutability: this feature also gives certainty to the contracting parties that the contract cannot be modified over time, as the use of blockchain technology ensures that none of the agreements are modified, replaced or deleted.
- Self-verification: this feature could be considered as a product of the previous ones, since all of them contribute to the fact that the Smart Contract does not need to be verified or interpreted by any of the participating parties, since both its wording and compliance are reliably recorded in the blockchain.

3.1Smart Contracts in Mexico

It is worth mentioning that in Mexico, as in the rest of the world, the use of web applications or progressive web applications with microservices-based architecture, the popularity of API rest and similar services that can be easily consumed and used.

Mexico plays a leading role in blockchain development in Latin America. It has the highest percentage of blockchain-based companies and is the third country in Latin America with the highest use of cryptocurrencies (Hootsuite). The use of blockchain technology in Mexico is a promise of financial revolution that is becoming more and more valid, fulfilling its objectives of guaranteeing a simple, secure and reliable system (Andrade, 2022).



Based on a study carried out to find out which are the most popular platforms that help us to develop smart contracts, Ethereum, Hyper ledger Fabric, Nem, Stellar, NEO, NXT are mentioned (Rodriguez, 2018).

The following image shows a comparison table of the above-mentioned smart contract development platforms.

Category	Etereum	Tela Hyperiedger	NEM	Estelar	NEO	NXT
Contract Language	Solidez	Golang, JavaScript	Java, C++, XEM	C++, Go, Java, JavaScript, Python, Ruby, Shell	.Net, Java, Kotlin, C, C++, Go Python, JavaScript	Java
Transactions per Second	15	3.500	100	10000	10.000	100
Launch	2015	2015	2015	2014	2016	2013
Consensus Algorithms	Pow, Pos	PBFT, TAMIZ	PDI	SCP	dBFT	Point of Sale
Assets	ERC files	Sphere, LIVE coin	Available	e-pagaré	Governing Token, Utility	Available

					Token, tokens NEP-5	
Tokens	ERC-20, ERC-223, ERC-777	Tokend Kuma	XEM	XML	Tpkens NEP-5	NXT 1.0, NXT 2.0,
Anonymity	NO	Yes	NO	NO	NO	Mixed currencies
Cons	Network overload and security problems.	There are no active tokens yet.	Transportation rates	It cannot handle the development of complicated contracts.	Regulated by the Chinese government	There is no mining opportunity.
Advantages	No installation cost, solid support, wider usability.	Open source, better security, wide range of languages.	Multi-form Transactions Agentrus++ Encrypted Messaging	Targeted ICO, cheaper, faster.	Smart Contracts 2.0 Faster, multiple languages available.	Multiform transactions easy on-the-fly response a.k.a. transaction pruning.
Programmable	Yes	Yes	No	Yes	Yes	No
Voting	Third part	Third part	No	Yes	Introduced two new voting systems	Yes

3.2 Security

SecurityInformation security is the protection of information using methods to ensure the confidentiality, integrity and reliability of information. Cryptography is a mathematical and computer science discipline related to the encryption of information through the use of different technologies or algorithms that allow authentication and access to the content of encrypted material.

Smart contracts provide the function of localisation, i.e. a hash function is applied to associate a small, manageable number with a large number. Data encryption: a hash function is applied to the data and the result is stored in the database, so that if someone steals the database they cannot interpret its contents if they do not know the hash function.

Data verification: when data is sent from one module to another, e.g. in parallel processing, the data is sent together with the result of its hash, thus sending the data and its short version. The receiving module knows the hash function, so it can test whether the received data is the same as the one it received encrypted with the hash, thus determining whether the data reception is correct (Fuentes & Ojeda, 2014).

3.3 Why are smart contracts not widely implemented?

Blockchain is a new technology that still has a long way to go, so training materials on the subject are still scarce. Often companies in need of blockchain professionals are forced to turn to talent from outside the country (Andrade, 2022).

This technique is being used mainly in companies, however no user will be able to make it useful if he/she is not related to programming, since he/she must establish all the system editions and updates to the code every time something changes, so if he/she does not have fundamental programming concepts he/she will not be able to establish terms to be coded.

4.2 Is its use for legal purposes?

The legal system to which Mexico belongs is the Neo-Romanist system, i.e. it is characterised mainly because its rules and legislation are written. For this reason, in order to understand how Smart Contracts and Blockhcain are regulated in Mexico, it is essential to take a general overview of those rules that are related and that must be taken into consideration for their use, namely (Franco, 2023).

Although the use of smart contracts is not yet approved as a form of legal contract before the judge by means of an article implemented in the political constitution of the United Mexican States, it is not an impediment to be carried out between two or more parties involved, and its use is considered attractive, due to the fact that the clauses and agreements that are made within the Blockchain, allow the conditions to be executed, even in real time, once everyone approves in the way they determine, and can take as an example the electronic signature to confirm their consent.

4. State of the Art

Escalona & Inclán (2011) cover in a general way some of the most commonly used hash functions in cryptography such as MD5, SHA-1, RIPEMD and HAVAL, as well as the basic algorithms of operation of each of them, their application or use and their main security problems or weaknesses encountered.

López et al. (2009) define algorithm as a set of steps which, when executed in the correct way, leads to a result (within a given time).

Garcia Belmont (2016) raise awareness of the state of technology use and the alternatives for making a secure connection, as well as the importance of protecting your information.

Fulgueira-Camilo et al. (2015) refer to the parallelisation process of the GOST cryptographic algorithm.
Palacios & Delgado (2006) describe the cryptographic algorithms most commonly used today to protect information in electronic form.

Ramírez (2019) identifies and describes blockchain technology and its applicability in law, from the development of smart contracts or self-executing contracts and their difference with electronic contracts, paying special attention to the elements of will and consent, which are considered as the main pillars of civil and commercial laws in the world.

Tapscott & Tapscott (2017) explain how technology has revolutionised and transformed the modern economy.

Castillo et al. (2011) mention how mathematics plays an important role in modern cryptography and how it exploits the hard problems (in the computational sense) that exist in number theory to develop cryptographic protocols.

Rodriguez (2018) states that the Blockchain has so many services and the best thing is that it gets rid of intermediaries completely.

Franco (2023) analyses the relationship between technology and law when an obligation is contracted through the use of blockchain, i.e. through smart contracts or better known as: Smart Contracts.

Omar et al. (2021) propose a blockchain solution that uses smart contracts to automate the GPO contracting process. We propose a generic framework for the HCSC contracting process with detailed algorithms describing various interactions between HCSC stakeholders.

Myung & Lee (2020) introduce a transparent and safe power trading algorithm between users using blockchain. The proposed algorithm has been implemented with an executable distributed code (i.e., smart contract) in an Ethereum blockchain platform.

Al-Otaibi (2022) proposed a new secure authentication approach using machine learning. To identify the dynamic time attack detection and authentication in an IoMT environment, this work implements K-Nearest neighbour (KNN) and machine learning using smart contract (KNN-MLSC).

Ji & Zhu (2021) consider the channel error rate, detection probability, secondary user base station budget and remaining energy of the secondary users (SUs) and then establishes the SU's utility function as well as the game model.

Almasoud, Hussai, & Hussain (2020) find that the existing literature has not proposed a framework that facilitates the interchangeable use of smart contracts for blockchain-based reputation systems.

Idelberger et al. (2016) inspect what are the possible legal and technical (dis)advantages of logic-based smart contracts in the light of common activities presented by ordinary contracts, and then offer ideas on how to use such logic-based smart contracts in combination with blockchain systems.

Kirli et al. (2022) systematically reviewed 178 peer-reviewed publications and 13 innovation projects, providing a comprehensive analysis of the strengths and weaknesses of smart contracts used in the energy sector.

Badruddoja et al. (2021) proposes a Naive Bayes prediction algorithm to perform prediction with inside blockchain smart contracts that promises to open up more opportunities in the field of Blockchain-AI decentralized applications.

Tao et al. (2020) propose, a new distributed and dynamic fragmentation system is analysed and implemented to substantially improve the performance of smart contract-based blockchain systems, while requiring minimal communication between fragmentations.

Chatterjee et al. (2019) explore the idea of exploiting the treewidth of smart contracts for formal analysis and compiler optimisation.

Huh & Kim (2020) mention that Blockchain was conceived to be applied as a way to solve the problems of real estate transaction services. In the case of the previously applied Blockchain technology, it is characterised by its security against disclosure, manipulation or false information, and is suitable for application to an environment dealing with low frequency data.

Xiong & Hu (2022) propose a delegated contract signature solution to eliminate the potential risk of contractual fraud caused by asymmetry of information and interests.

Wei et al. (2020) propose an EVM-based taint analysis method to reduce invalid entries. a database of dangerous transactions is designed to identify the entries genetic algorithm to optimize the code coverage of the entry. for smart contracts.

Ahmed et al. (2022) examine the convergence of blockchain technology and artificial intelligence, a unique driver towards technological transformation in intelligent and sustainable IoT applications.

Sharma et al. (2021) propose a blockchain-based IoT framework with artificial intelligence that presents the integration of artificial intelligence and blockchain for IoT applications.

Singh et al. (2020) provide a comprehensive literature review of the security issues and concerns affecting the deployment of blockchain systems in smart cities. This paper presents a detailed discussion of several key factors for the convergence of Blockchain and AI technologies that will help form a sustainable smart society. Siddiqui et al. (2023) propose a service security architecture based on authentication and authorization for restricted environments during collaborative tasks for software-defined networking (SDN) and smart contract-enabled municipal smart cities.

Hewa, Ylianttila & Liyanage (2021) explored the significant applications which already benefited from the smart contracts. They also highlight the future potential of the blockchain based smart contracts in these applications perspective.

Alharby, Aldweesh, & Van Moorsel (2018) classified these papers into six categories, namely, security, privacy, software engineering, application, performance & scalability and other smart contract related topics. Khatoon (2020) review of applications available for the healthcare system using blockchain technology. Hewa, Liyanage, Kanhare & Ylianttila (2021) identified the significant technical aspects of blockchain-based smart contracts with the associated future research directions.

Pee, Kang, Song, & Jang (2019) propose a peer-to-peer (P2P) system that can freely trade the produced energy based on smart contracts.

Gans (2019) examines the capabilities of smart contracts from an economic perspective. It is demonstrated that by improving observability and reducing the costs of verification of contract obligation performance, the space of feasible contracts can be enlarged.

Inshakova, Goncharov, & Salikov (2020) substantiated that a smart contract can neutralize many civil law problems, including questions about applicable law, judicial jurisdiction, verification of counterparty powers. Vatiero (2022) propose several institutional expedients that may reduce these transaction costs of smart contracts.

Ante (2021) analyzes 468 articles on the topic of smart contracts and their 20,188 references, providing a summary and analysis of the current state of research on smart contracts and identifying intellectual structures and emerging trends.

Wang et al. (2018) describe the recent advances of smart contract and present its future development trends,aimed at providing helpful guidance and reference for future research efforts.

Dustdar et al. (2021) deal with questions related to blockchains in complex Internet of Things (IoT)-based ecosystems. Such ecosystems are typically composed of IoT devices, edge devices, cloud computing software services, as well as people, who are decision makers in scenarios such as smart cities.

Gourisetti et al. (2021) propose a reference framework for a transactive energy market based on distributed ledger technology such as blockchain.

Alharby & van Moorsel (2017) conduct a systematic mapping to collect all research that is relevant to smart contracts from a technical perspective.

Leka, Selimi, & Lamani (2019) identified where recent studies have been focused on and offers a broad perspective relating blockchain applications and smart contracts, their main problems and corresponding solutions and will help to specify gaps and future research.

Laarabi, Chegri, Mohammadia,& Lafriouni, (2022) examines literature focused on the use of smart contracts in real estate while providing a conceptual classification.

Awaji, Solaiman, & Albshri (2020) examine state of the art in blockchain-based applications that have been developed for educational purposes. Second, it summarises the challenges and research gaps that need to be addressed in future studies.

Alt, & Reitwiessner (2018) built an SMT-based formal verification module within the compiler of Solidity, a popular language for writing smart contracts. The tool is seamlessly integrated into the compiler, where during compilation, the user is automatically warned of and given counterexamples for potential arithmetic overflow/underflow, unreachable code, trivial conditions, and assertion fails.

Sen, Mukherjee, & Bhattacharya (2021) identifies the problems associated with the traditional waste management system, defines algorithms that underpin the proposed smart waste management system, and compares the two systems by discussing how the latter alleviates the problems of the traditional waste management system.

Wang, et al. (2019) focus is on the test coverage criteria for smart contracts, which are objective rules that measure test quality.

Yu et al. (2018) proposes a parallel smart contract model on blockchain which has a better performance in transaction processing.

Patel et al. (2021) propose a BC-envisioned IoT-enabled PSC scheme, SaNkhyA , which is executed in three phases. In the first phase, the scheme eliminates colluding dishonest miners through the proposed miner selection algorithm.

Skotnica, Klicpera, & Pergl (2020) proposes a model-driven approach to create blockchain smart contracts based on a visual domain-specific language called DasContract.

Wang et al. (2018) propose a smart-contract based algorithm for constructing service-based systems through the composition of existing services.

Kim (2022) analyses with the required Blockchain diploma. In addition, we use an automatic translation system, which incorporates natural language processing, to perform verification work that does not require an existing public certificate.

Pontiveros et al. (2018) propose a compression method for smart contracts deployed in the Ethereum blockchain. By taking advantage of the repetition of sections of bytecode among multiple smart contracts previously deployed in the Ethereum blockchain we propose a new pseudo opcode that acts as a pointer that will allow smart contracts to reuse previously deployed code.

Jiménez (2017) considers the introduction and expansion of smart contracts as contract enforcement devices in the marketplace.

Zou et al. (2019) performed an exploratory study to understand the current state and potential challenges developers are facing in developing smart contracts on blockchains, with a focus on Ethereum (the most popular public blockchain platform for smart contracts).

Kolvart, Poola, & Rull (2016) clarify that, usually, a smart contract is a programmed functionality which executes some part of the legal contract.

Raskin (2016) examines smart contracts from a legal perspective, explain smart contracts operation and place in existing contract law.

Rey (2018) explain the Smart Contracts phenomenon from a legal perspective and frame them in their particular ecosystem, and then address some main legal issues related to Smart Contracts.

Bhargavan et al. (2016) outline a framework for analysing and verifying both the runtime security and functional correctness of Ethereum contracts by translating them into F*, a functional programming language oriented towards program verification.

Bocek & Stiller (2017) describes smart contracts from multiple perspectives and identifies and clarifies some of the most common misconceptions regarding smart contracts. This study also provides some guidelines and insights on the proper management of smart contracts.

Garfatta, Gaaloul, & Graiet (2021) present a general overview of the different axes investigated by researchers towards the verification of smart contracts, while taking a special interest in studies that focus on formal verification, the different approaches they apply and vulnerabilities they target.

Ellul & Pace (2018) show how standard techniques from runtime verification can be used in the domain of smart contracts, including a novel stake-based instrumentation technique which ensures that the violating party provides insurance for correct behavior.

Qasse, Hamdaqa & Jónsson (2023) characterize smart contract upgrading patterns and analyze their prevalence based on the deployed contracts that exhibit these patterns. investigate the reasons why developers upgrade contracts.

Negara, Hidayanto, Andryani& Syaputra (2021) investigate technological developments and implementation of smart contracts in various domains.

Verheijke & Rocha (2022) collect a total of 26,799 verified Solidity smart contracts from Etherscan, to analyze the language constructs used in calling another contract or exchanging ether.

Bracamonte & Okada (2017) provide some evidence of the influence of the community in the implementation and improvement of security measures related to the smart contracts

Coita, Abrudan & Matei (2019) analyzed the existing literature, the experts' expectations regarding the blockchain and we conceptualized some implications for businesses, human resources management, and marketing.

Eggers, Hein, Weking, Böhm & Krcmar (2021) investigate the potentials for automation that organizations achieve through smart contracts and how smart contracts differ from established automation technologies, such as workflow management systems, enterprise resource planning systems, and robotic process automation.

Wang, Jin, Dai, Choo & Zou (2021) systematically review existing research efforts on Ethereum smart contract security, published between 2015 and 2019.

Rouhani & Deters (2019) reviews the key concepts and proposes the direction of recent studies and developments regarding the smart contract.

Kushwaha, Joshi, Singh, Kaur & Lee (2022) systematic review of the security vulnerabilities in the Ethereum blockchain is presented. The main objective is to discuss Ethereum smart contract security vulnerabilities, detection tools, real life attacks and preventive mechanisms.

Wan, Xia, Lo, Chen, Luo & Yang (2021) find that blockchain platforms have a statistically significant impact on practitioners' security perceptions and practices of smart contract development. Based on our findings, we highlight future research directions and provide recommendations for practitioners.

Zhou, Hua, Pi, Sun, Nomura, Yamashita & Kurihara (2018) proposed a security assurance method for smart contract source code to find potential security risks.

Wohrer & Zdun (2018) identify the creation process of writing well performing and secure contracts in Ethereum.

Momeni, Wang & Samavi (2019) introduce a machine learning predictive model that detects patterns of security vulnerabilities in smart contracts.

Parizi & others (2018) development a far-reaching experimental assessment of current static smart contracts security testing tools, for the most widely used blockchain, the Ethereum and its domain-specific programming language, Solidity.

Zhang & others (2021) propose EOSAFE, the first static analysis framework that can be used to automatically detect vulnerabilities in EOSIO smart contracts at the bytecode level.

Liu & Liu (2019) show how smart contracts in modern-day systems have changed the approach to money tracing.

Deng et al. (2020) present a survey of the Ethereum smart contract's various vulnerabilities and the corresponding defence mechanisms that have been applied to combat them.

Brent, Jurisevic, Kong, Liu, Gauthier, Gramoli & Scholz (2018) present Vandal: a security analysis framework for Ethereum smart contracts.

5. Mathematical model

1: Identify the Objective

The goal is to maximize the implementation of smart contracts by at least 10% to date. This will save time on procedures and the search for desired clauses, reducing additional expenses on legal advisors. It aims to involve only the interested parties in the contract, ensuring that once all parties agree, fraud or manipulation of the established terms is not possible, thus providing security.

2: Define Decision Variables

- X1 = Legal intermediaries (costs).
- X2 = Security certificates (costs).
- X3 = Stationery materials (costs).

3: Objective Function

The objective is to minimize the total consumption of costs for smart contract development based on the variables mentioned earlier. The goal is to minimize costs related to the process of smart contract development in terms of monetary expenses for third-party services and raw material purchases.

Objective Function: Minimize z = X1 + X2 + X3, where z represents the total use of resources and services measured in ($) seeking to minimize costs related to smart contract development.

4: Identify Constraints

- Y1 = Internet Provider Services (IPS) costs.
- Y2 = Hosting server costs for the web application serving as the smart contract development service.
- Y3 = Compliance with Mexican legislation (related taxes).

5: Formulate the Mathematical Model

- Objective Function: Minimize z = X1 + X2 + X3
- Subject to:
- X1 ≤ 50% of $8000
- X2 ≤ 30% of $3000
- X3 ≤ 80% of $500 (stationery material)
- Cost savings level X1 + X2 + X3 ≥ 65% of $12000 approximately

6: Resolution

Decision Variables:

- Legal intermediaries (35-50%)
- Security certificates (30-35%)
- Stationery materials (15-90%)

Subject to:

Minimize z = A1 * X1 + A2 * X2 + A3 * X3, where A1, A2, and A3 are weights reflecting the development cost of each contract.

Constraints:

- X1 ≥ 50% (legal intermediaries)
- 60% ≤ X2 ≤ 65% (security certificates)
- X3 ≥ 80% (stationery materials)

$$\sum_{i=1}^{n} = x_1 + x_2 + x_3$$

7. Pseudocode

```
# Function of the genetic algorithm
Def genetic_algorithm(initial_population,generations=100,
mutation_rate=0.01):
    current_population = initial_population.
    # print the initial population
    print("Initial population:")
    for individual in current_population:
        print(individual)
    for generation in range(1, generations + 1):
        # select parents
        parents            =            select_by_rule(current_population,
number_to_select=len(current_population))

        # crossover parents to create new population
        new_population = cross_parent(parents)
        # Apply mutation
        new_population = mutate_population(new_population, mutation_rate)
```

The code was executed using the Google Colab platform in Python, utilizing 0.9GB of RAM and 29.6GB of solid-state disk on the Huawei laptop, which is equipped with 8GB of RAM, 256GB of storage, 64 bits, and an AMD Ryzen 5th generation processor.

8. Instances

IDE	Initial table identifier
STRUCTURE	In legal terms of the contract, that is, 'clauses'
MODIFIER	Development modifier data
EVENTS	State of development
ENUMERATORS	Measurement of the state of progress of the contract
FUNCTIONS	It is based on the conditions that are going to be executed
INTERNAL	Embedded operation
SSL	Safety certificate
GET	Web request
POST	Web request
QUERY	Database query
API	Consumption of an APIrest
MATERIAL	Office stationery
COMPUTERS	Equipment on which the application is running
SOFTWARE	Web application development
INTERNET	Internet provider services
LAWS	Legal framework in mexico
PRIVATE_KEY	Public key encryption
PUBLIC_KEY	Private encryption key

9. Complexity analysis

fitness O(1), generate_random_data O(n), section_by_route $O(n^2)$, cross_parent_crossing O(n), mutate_population O(n), genetic_algorithm $O(n^2)$.

$O(1)+O(n)+O(n^2)+O(n)+O(n)+O(n^2) =O(1+ 2n +2n^2)$

10. Interpretation of Results

Implementing the use of the genetic algorithm has yielded a solution with X1 = 40%, X2 = 59%, and X3 = 82%. This implies that the optimal percentage of labor is 40% for third parties not involved in the contract, 60% for security certificates for secure data transit through the application, and 70% for optimal material usage, minimizing effort consumption without violating constraints.

Based on these considerations, it can be concluded that applying the use of smart contracts is efficient, saving monetary resources through a more agile process and, likewise, reducing environmental pollution by minimizing paper usage and digitizing all related information for storage on a server.

11. Validation and Testing

A genetic algorithm was developed based on the variables of primary interest to create and iterate various values related to them with the aim of obtaining an approximate prediction of our model.

6.- Conclusions

In general, the use of algorithms will always provide better results and solutions in different ways to different problems, however, it is certain that any company or association that knows the hash algorithms and wants to work in a fast and secure way to share their information and manage it without fear that it can be stolen, will not hesitate to use this algorithm.

The implementation of the same for the application of smart contracts is quite common because its encryption methods guarantee the fidelity of the data exchange processes, the use of smart contracts is still not very common, however, both in Mexico and in different parts of the world, have begun to implement different legal regulations that allow the use of these without incurring in any fault, it is important to mention that the application of Smart Contracts represents multiple competitive advantages compared to the use of traditional contracts.

References

Ahmadisheykhsarmast, S., & Sonmez, R. (2020). A smart contract system for security of payment of construction contracts. *Automation in construction*, 120, 103401.

Ahmed, I., Zhang, Y., Jeon, G., Lin, W., Khosravi, M. R., & Qi, L. (2022). A blockchain-and artificial intelligence-enabled smart IoT framework for sustainable city. *International Journal of Intelligent Systems*, 37(9), 6493-6507.

Alharby, M., & van Moorsel, A. (2017). A systematic mapping study on current research topics in smart contracts. Available at SSRN 3876872.

Alharby, M., Aldweesh, A., & Van Moorsel, A. (2018, November). Blockchain-based smart contracts: A systematic mapping study of academic research (2018). In 2018 International Conference on Cloud Computing, Big Data and Blockchain (ICCBB) (pp. 1-6). IEEE.

Almasoud, A. S., Hussain, F. K., & Hussain, O. K. (2020). Smart contracts for blockchain-based reputation systems: A systematic literature review. *Journal of Network and Computer Applications*, 170, 102814.

Al-Otaibi, Y. D. (2022). K-nearest neighbour-based smart contract for internet of medical things security using blockchain. *Computers and Electrical Engineering*, 101, 108129.

Alt, L., & Reitwiessner, C. (2018). SMT-based verification of solidity smart contracts. In Leveraging Applications of Formal Methods, Verification and Validation. Industrial Practice: 8th International Symposium, ISoLA 2018, Limassol, Cyprus, November 5-9, 2018, Proceedings, Part IV 8 (pp. 376-388). Springer International Publishing.

Andrade, J. (2022). Blockchain, the challenges for Mexico. *Forbes Network*, 15-17.

Ante, L. (2021). Smart contracts on the blockchain–A bibliometric analysis and review. *Telematics and Informatics*, 57, 101519.

Awaji, B., Solaiman, E., & Albshri, A. (2020, July). Blockchain-based applications in higher education: A systematic mapping study. In Proceedings of the 5th international conference on information and education innovations (pp. 96-104).

Badruddoja, S., Dantu, R., He, Y., Upadhayay, K., & Thompson, M. (2021, May). Making smart contracts smarter. In 2021 IEEE International Conference on Blockchain and Cryptocurrency (ICBC) (pp. 1-3). IEEE.

Bhargavan, K., Delignat-Lavaud, A., Fournet, C., Gollamudi, A., Gonthier, G., Kobeissi, N., ... & Zanella-Béguelin, S. (2016, October). Formal verification of smart contracts: Short paper. In Proceedings of the 2016 ACM workshop on programming languages and analysis for security (pp. 91-96).

Bocek, T., & Stiller, B. (2017). Smart contracts–blockchains in the wings. In Digital marketplaces unleashed (pp. 169-184). Berlin, Heidelberg: Springer Berlin Heidelberg.

Bracamonte, V., & Okada, H. (2017). An exploratory study on the influence of guidelines on crowdfunding projects in the ethereum blockchain platform. In Social Informatics: 9th International Conference, SocInfo 2017, Oxford, UK, September 13-15, 2017, Proceedings, Part II 9 (pp. 347-354). Springer International Publishing.

Brent, L., Jurisevic, A., Kong, M., Liu, E., Gauthier, F., Gramoli, V., & Scholz, B. (2018). Vandal: A scalable security analysis framework for smart contracts. arXiv preprint arXiv:1809.03981.

Castillo Rubí, M. A., Santana de la Cruz, N., Díaz Lobaton, A. M., & Almanza (2011). Number theory in cryptography and its weakness in the face of the possible era of quantum computers. *Multidisciplinary scientific journal of foresight*, 264-273.

Chatterjee, K., Goharshady, A. K., & Goharshady, E. K. (2019, April). The treewidth of smart contracts. In Proceedings of the 34th ACM/sigapp symposium on applied computing (pp. 400-408).

Coita, D. C., Abrudan, M. M., & Matei, M. C. (2019). Effects of the blockchain technology on human resources and marketing: an exploratory study. In Strategic Innovative Marketing and Tourism: 7th ICSIMAT, Athenian Riviera, Greece, 2018 (pp. 683-691). Springer International Publishing.

Dustdar, S., Fernández, P., García, J. M., & Ruiz-Cortés, A. (2021). Elastic smart contracts in blockchains. *IEEE/CAA Journal of Automatica Sinica*, 8(12), 1901-1912.

Eggers, J., Hein, A., Weking, J., Böhm, M., & Krcmar, H. (2021). Process automation on the blockchain: an exploratory case study on smart contracts.

Ellul, J., & Pace, G. J. (2018, September). Runtime verification of ethereum smart contracts. In 2018 14th European Dependable Computing Conference (EDCC) (pp. 158-163). IEEE.

Escalona, S. B., & Inclán, L. V. (2011). Funciones resúmenes o hash. *Telemática*, 10(1).

Franco, M. E. (2023). Smart Contracts: Perspectives in Current Mexican Legislation and Considerations for their Application. Infortec, 142-144.

Fuentes, M. d., & Ojeda, J. C. (2014). Introduction to algorithm analysis and design. Universidad autónoma metropolitana.

Fulgueira-Camilo, M., Hernández-Duany, O., & Henry-Fuenteseca, V. (2015). Parallelization of the GOST Cryptographic Algorithm Employing the Shared Memory Paradigm. Lámpsakos, (14), 18-24.

Gans, J. S. (2019). The fine print in smart contracts (No. w25443). National Bureau of Economic Research.

García Belmont, R. (2016). Digital signature based on HASH functions and a hybrid cryptographic algorithm.

Garfatta, I., Klai, K., Gaaloul, W., & Graiet, M. (2021, February). A survey on formal verification for solidity smart contracts. In Proceedings of the 2021 Australasian Computer Science Week Multiconference (pp. 1-10).

Gourisetti, S. N. G., Sebastian-Cardenas, D. J., Bhattarai, B., Wang, P., Widergren, S., Borkum, M., & Randall, A. (2021). Blockchain smart contract reference framework and program logic architecture for transactive energy systems. Applied Energy, 304, 117860.

He, D., Deng, Z., Zhang, Y., Chan, S., Cheng, Y., & Guizani, N. (2020). Smart contract vulnerability analysis and security audit. *IEEE Network*, 34(5), 276-282.

He, N., Zhang, R., Wang, H., Wu, L., Luo, X., Guo, Y., & Jiang, X. (2021). {EOSAFE}: security analysis of {EOSIO} smart contracts. In 30th USENIX Security Symposium (USENIX Security 21) (pp. 1271-1288).

Hewa, T. M., Hu, Y., Liyanage, M., Kanhare, S. S., & Ylianttila, M. (2021). Survey on blockchain-based smart contracts: Technical aspects and future research. IEEE Access, 9, 87643-87662.

Hewa, T., Ylianttila, M., & Liyanage, M. (2021). Survey on blockchain based smart contracts: Applications, opportunities and challenges. *Journal of network and computer applications*, 177, 102857.

Huh, J. H., & Kim, S. K. (2020). Verification plan using neural algorithm blockchain smart contract for secure P2P real estate transactions. *Electronics*, 9(6), 1052.

Idelberger, F., Governatori, G., Riveret, R., & Sartor, G. (2016). Evaluation of logic-based smart contracts for blockchain systems. In Rule Technologies. Research, Tools, and Applications: 10th International Symposium, RuleML 2016, Stony Brook, NY, USA, July 6-9, 2016. Proceedings 10 (pp. 167-183). Springer International Publishing.

Inshakova, A. O., Goncharov, A. I., & Salikov, D. A. (2020). Electronic-digital smart contracts: modernization of legal tools for foreign economic activity. In The 21st Century from the Positions of Modern Science: Intellectual, Digital and Innovative Aspects (pp. 3-13). Springer International Publishing.

Ji, C., & Zhu, Q. (2021). Smart contract-based secure cooperative spectrum sensing algorithm. International *Journal of Distributed Sensor Networks*, 17(12), 15501477211058673.

Jiménez, J. W. I. (2017). Legal issues surrounding blockchain ("blockchain") and smart contracts ("smart contracts"). icade. Law School Journal, (101).

Khatoon, A. (2020). A blockchain-based smart contract system for healthcare management. *Electronics*, 9(1), 94.

Kim, S. K. (2022). Blockchain smart contract to prevent forgery of degree certificates: Artificial intelligence consensus algorithm. *Electronics*, 11(14), 2112.

Kirli, D., Couraud, B., Robu, V., Salgado-Bravo, M., Norbu, S., Andoni, M., & Kiprakis, A. (2022). Smart contracts in energy systems: A systematic review of fundamental approaches and implementations. *Renewable and Sustainable Energy Reviews*, 158, 112013.

Kolvart, M., Poola, M., & Rull, A. (2016). Smart contracts. The Future of Law and etechnologies, 133-147.

Kushwaha, S. S., Joshi, S., Singh, D., Kaur, M., & Lee, H. N. (2022). Systematic review of security vulnerabilities in ethereum blockchain smart contract. *IEEE Access*, 10, 6605-6621.

Laarabi, M., Chegri, B., Mohammadia, A. M., & Lafriouni, K. (2022, March). Smart Contracts Applications in Real Estate: A Systematic Mapping Study. In 2022 2nd International Conference on Innovative Research in Applied Science, Engineering and Technology (IRASET) (pp. 1-8). IEEE.

Leka, E., Selimi, B., & Lamani, L. (2019, September). Systematic literature review of blockchain applications: Smart contracts. In 2019 International Conference on Information Technologies (InfoTech) (pp. 1-3). IEEE.

Liu, J., & Liu, Z. (2019). A survey on security verification of blockchain smart contracts. *IEEE Access*, 7, 77894-77904.

Momeni, P., Wang, Y., & Samavi, R. (2019, August). Machine learning model for smart contracts security analysis. In 2019 17th International Conference on Privacy, Security and Trust (PST) (pp. 1-6). IEEE.

Myung, S., & Lee, J. H. (2020). Ethereum smart contract-based automated power trading algorithm in a microgrid environment. *The Journal of Supercomputing*, 76(7), 4904-4914.

Negara, E. S., Hidayanto, A. N., Andryani, R., & Syaputra, R. (2021). Survey of smart contract framework and its application. *Information*, 12(7), 257.

Omar, I. A., Jayaraman, R., Debe, M. S., Salah, K., Yaqoob, I., & Omar, M. (2021). Automating procurement contracts in the healthcare supply chain using blockchain smart contracts. *IEEE Access*, 9, 37397-37409.

Palacios, R., & Delgado, V. (2006). Introduction to cryptography: types of algorithms. In Annals of mechanics and electricity.

Parizi, R. M., Dehghantanha, A., Choo, K. K. R., & Singh, A. (2018). Empirical vulnerability analysis of automated smart contracts security testing on blockchains. arXiv preprint arXiv:1809.02702.

Patel, N. S., Bhattacharya, P., Patel, S. B., Tanwar, S., Kumar, N., & Song, H. (2021). Blockchain-envisioned trusted random oracles for IoT-enabled probabilistic smart contracts. *IEEE Internet of Things Journal*, 8(19), 14797-14809.

Pee, S. J., Kang, E. S., Song, J. G., & Jang, J. W. (2019). Blockchain based smart energy trading platform using smart contract. In 2019 International Conference on Artificial Intelligence in Information and Communication (ICAIIC) (pp. 322-325). IEEE.

Pontiveros, B. B. F., Norvill, R., & State, R. (2018, February). Recycling smart contracts: Compression of the ethereum blockchain. In 2018 9th IFIP International Conference on New Technologies, Mobility and Security (NTMS) (pp. 1-5). IEEE.

Qasse, I., Hamdaqa, M., & Jónsson, B. Þ. (2023). Smart contract upgradeability on the Ethereum blockchain platform: An exploratory study. arXiv preprint arXiv:2304.06568.

Ramirez, J. P. (2019). Smart contracts. *Journal of research in information technology*, 7(14), 1-10.

Raskin, M. (2016). The law and legality of smart contracts. *Geo. L. Tech. Rev.*, 1, 305.

Rey, J. F. (2018). Smart Contract: Concept, ecosystem and main private law issues. La Ley Mercantil, (47), 1.

Rodriguez, N. (2018). Smart Contracts: Definitive Guide For Beginners. 101 Blockchains, 45-51.

Rouhani, S., & Deters, R. (2019). Security, performance, and applications of smart contracts: A systematic survey. *IEEE Access*, 7, 50759-50779.

Sen Gupta, Y., Mukherjee, S., Dutta, R., & Bhattacharya, S. (2021). A blockchain-based approach using smart contracts to develop a smart waste management system. *International Journal of Environmental Science and Technology*, 1-24.

Sharma, A., Podoplelova, E., Shapovalov, G., Tselykh, A., & Tselykh, A. (2021). Sustainable smart cities: convergence of artificial intelligence and blockchain. *Sustainability*, 13(23), 13076.

Siddiqui, S., Hameed, S., Shah, S. A., Khan, A. K., & Aneiba, A. (2023). Smart contract-based security architecture for collaborative services in municipal smart cities. *Journal of Systems Architecture*, 135, 102802.

Singh, S., Sharma, P. K., Yoon, B., Shojafar, M., Cho, G. H., & Ra, I. H. (2020). Convergence of blockchain and artificial intelligence in IoT network for the sustainable smart city. *Sustainable cities and society*, 63, 102364.

Skotnica, M., Klicpera, J., & Pergl, R. (2020). Towards model-driven smart contract systems–code generation and improving expressivity of smart contract modeling. Proc. EEWC, 20.

Tao, Y., Li, B., Jiang, J., Ng, H. C., Wang, C., & Li, B. (2020, April). On sharding open blockchains with smart contracts. In 2020 IEEE 36th international conference on data engineering (ICDE) (pp. 1357-1368). IEEE.

Tapscott, D., & Tapscott, A. (2017). The blockchain revolution. Discover how this new technology will transform the global economy. Ediciones Deusto.

UNAM. (n.d.). Study guides Plan 1471. PLAN 1475 School of Law - UNAM.

Vatiero, M. (2022). Smart contracts vs incomplete contracts: A transaction cost economics viewpoint. *Computer Law & Security Review*, 46, 105710.

Verheijke, D., & Rocha, H. (2022, May). An exploratory study on solidity guards and ether exchange constructs. In Proceedings of the 5th International Workshop on Emerging Trends in Software Engineering for Blockchain (pp. 1-8).

Wan, Z., Xia, X., Lo, D., Chen, J., Luo, X., & Yang, X. (2021, May). Smart contract security: A practitioners' perspective. In 2021 IEEE/ACM 43rd International Conference on Software Engineering (ICSE) (pp. 1410-1422). IEEE.

Wang, P., Liu, X., Chen, J., Zhan, Y., & Jin, Z. (2018, May). QoS-aware service composition using blockchain-based smart contracts. In Proceedings of the 40th International Conference on Software Engineering: Companion Proceeedings (pp. 296-297).

Wang, S., Yuan, Y., Wang, X., Li, J., Qin, R., & Wang, F. Y. (2018, June). An overview of smart contract: architecture, applications, and future trends. In 2018 IEEE Intelligent Vehicles Symposium (IV) (pp. 108-113). IEEE.

Wang, X., Xie, Z., He, J., Zhao, G., & Nie, R. (2019, October). Basis path coverage criteria for smart contract application testing. In 2019 International Conference on Cyber-Enabled Distributed Computing and Knowledge Discovery (CyberC) (pp. 34-41). IEEE.

Wang, Z., Jin, H., Dai, W., Choo, K. K. R., & Zou, D. (2021). Ethereum smart contract security research: survey and future research opportunities. *Frontiers of Computer Science*, 15, 1-18.

Wei, Z., Wang, J., Shen, X., & Luo, Q. (2020). Smart contract fuzzing based on taint analysis and genetic algorithms. *Journal of Quantum Computing*, 2(1), 11.

Wohrer, M., & Zdun, U. (2018, March). Smart contracts: security patterns in the ethereum ecosystem and solidity. In 2018 International Workshop on Blockchain Oriented Software Engineering (IWBOSE) (pp. 2-8). IEEE.

Xiong, W., & Hu, Y. (2022). Delegate contract signing mechanism based on smart contract. *Plos one*, 17(8), e0273424.

Yu, W., Luo, K., Ding, Y., You, G., & Hu, K. (2018, September). A parallel smart contract model. In Proceedings of the 2018 International Conference on Machine Learning and Machine Intelligence (pp. 72-77).

Zhou, E., Hua, S., Pi, B., Sun, J., Nomura, Y., Yamashita, K., & Kurihara, H. (2018, February). Security assurance for smart contract. In 2018 9th IFIP International Conference on New Technologies, Mobility and Security (NTMS) (pp. 1-5). IEEE.

Zou, W., Lo, D., Kochhar, P. S., Le, X. B. D., Xia, X., Feng, Y., & Xu, B. (2019). Smart contract development: Challenges and opportunities. *IEEE Transactions on Software Engineering*, 47(10), 2084-2106.