Block-Hash Signature (BHS) for Transaction Validation in Smart Contracts for Security and Privacy using Blockchain

Sonika Bhatnagar[1], Mohit Dayal[2], Deepti Singh[3], Shitiz Upreti[4], Kamal Upreti[5,*] and Jitender Kumar[6]

[1]*Dept. of Computer Science & Engineering, Government Polytechnic Baheri, Bareilly, India*
[2]*Dept. of Applied Science, Bharati Vidyapeeth's College of Engineering, New Delhi, India*
[3]*Dept. of Information Technology, ABES Institute of Technology, Ghaziabad, India*
[4]*Dept.of Information Technology, Asian Education Group (AEG), Noida*
[5]*Dept. of Computer Science & Engineering, Dr. Akhilesh Das Gupta Institute of Technology and Management, New Delhi, India*
[6]*Dept. of Computer Science & Engineering, J.C Bose University of Science & Technology, YMCA, Faridabad, Haryana, India*
E-mail: sonika.22dec@gmail.com; mohitdayal.md@gmail.com; deeptisingh0691@gmail.com; upretiec@gmail.com; kamalupreti1989@gmail.com; jk85ymca@gmail.com
[*]*Corresponding Author*

Received 11 September 2022; Accepted 20 November 2022;
Publication 29 April 2023

Abstract

Some of the well-known signature techniques like Winternitz and Lamport are not considered to be very appropriate for the usage of hashing or smart contracts in Blockchains security because of their size $O(n^2)$, which is prominently too high. Although in Blockchain, the security concern is on the top priority because of its distributed P2P design still, the security enhancement is required to sign and verify the documents forwarded to the peers, especially in Hyperledger Fabric. Here, this paper presents a new signature technique "Block-Hash" to enhance Blockchain security by using it in smart contracts

Journal of Mobile Multimedia, Vol. 19_4, 935–962.
doi: 10.13052/jmm1550-4646.1941

as well as hashing with size 3Xn bits (n = 256, generally for SHA-256 Hashing) and which can score 112 bits security. The proposed signature can be used appropriately for signing a smart contract by the endorser or committer node. Also, it can be used with a hash algorithm in forming a Merkle tree. Apart from the description and implementation of Block-Hash Signature, this paper has covered the analysis of its security and correctness measures with a table for result comparison.

Keywords: Digital signature, blockchain security, hyperledger fabric, smart contract.

1 Introduction

The traditional centralized systems (banks mainly), have their shortcomings like, the time you need to initiate a fund transfer or any single transaction you have to take the assistance of a third party which may be a bank or app. And by doing that you are giving them the power to control your money, personal information, bank details, and trust. To get rid of this shortcoming of the current monetary system, Satoshi Nakamoto proposed a new concept of cryptocurrency which was formulated using Blockchain [1]. He published a paper in 2007, in which he proposed a monetary framework that is decentralized or distributed and also introduced digital money (currency) named "Bitcoin". The fundamental innovation of Bitcoin cryptocurrency is termed Blockchain. A series of linked records or blocks is all that makes up a blockchain. It follows a chronological order (chronological implies the blocks are entered into the chain in the order they occurred) and is immutable, meaning that once an entry is added to the chain, it cannot be changed. Due to the distributed nature of blockchain, there is no centralised governance or authority structure [2]. It follows the concept of a linked list as the next block is connected with the previous block by containing the hash of the previous block which is shown in Figure 1.

Blockchain can be defined precisely as "Blockchain is a persistently increasing distributed ledger (file) that holds a persistent record of all its transactions held, in an immutable, chronological, and secure manner." Secure fund transfers, correct notice of property to be bought or sold, hiring of legal contracts, etc. are only a few of the use cases for blockchains that don't require a third-party arbiter like banks or governmental entities. In other words, Blockchain is essentially a permanent record of "who owns what," which is also referred to as the largest filesystem (or spreadsheet) in the world and is continually expanding. A blockchain is a distributed ledger

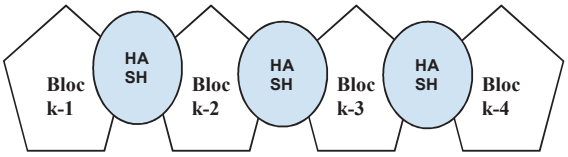

Figure 1 Common blockchain linkage.

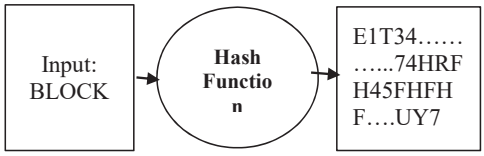

Figure 2 Process of hashing.

that uses a peer-to-peer architecture that permits the global storage of data across millions of servers [3]. Since it is distributed in nature, everyone in the network will have a track record of every single transaction that took place in the blockchain. This makes almost zero possibility of centralized authority or governance because every single person in the blockchain equally owns the regulations of Blockchain.

1.1 Key Features of Blockchain

- Distributed: There is no concept of a single or central governing body (authority).
- Immutable: Information once written into the Blocks, cannot be manipulated or altered.
- Security: Hashing is used as a cryptographic key function which is an irreversible one-way process. A hash function, shown in Figure 2 converts a variable-sized input into a unique encrypted fixed-length output and this makes it almost impossible for the hackers to guess the Hash size and value. SHA256 is the widely used Hash Function [4].
- Distributed Ledger: This is a public (digital) ledger that will keep a record of all transactions and respective participants. It's all transparent, nothing is hidden.
- Consensus: To decide whether to accept or reject a block from being added, all the active nodes in the Blockchain network must come to a consensus. There are three Consensus algorithms majorly used:
- Proof of Work: A node has to prove it has solved and submitted the right answer to a complex mathematical problem (golden nonce) first among all other nodes.

- Proof of Stake: Participants invest by holding the coins as a stake.
- Proof of Capacity: The validators invest their storage (hard drive space).

2 Architecture of Blockchain

Figure 3. Depicts the basic architectural view of Blockchain. The components of (Genesis) blockchain [5] are listed below:

- Node – A computer or user inside the blockchain network (each node holds a copy of the entire blockchain ledger).
- Consensus (Protocol) – Set of rules to reach a common agreement for blockchain operations to be performed.
- Block – It is a data structure that keeps track of a set of transactions in the Blockchain network.
- Chain – A series of blocks in chronological order.

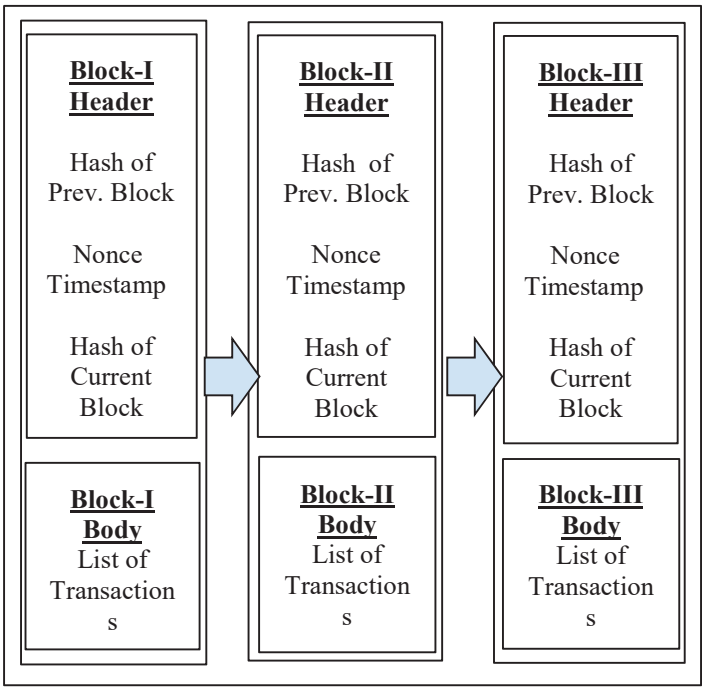

Figure 3 Basic architecture and components of genesis blockchain.

- Transaction – The smallest building block of the blockchain network (it may be information, records, etc.) which is served as the basis of a blockchain.
- Miners – (Any Person) Particular nodes that perform the verification process of the block before adding it into the blockchain.
- Nonce – A 32-bit nonce value is arbitrarily generated at the time of block creation, after which it produces a hash for the block header.

2.1 How Blockchain Works?

Figure 4 represents the overall working model of a Blockchain. It depicts the overall process which will be followed if a node wants to initiate a transaction within a blockchain [6].

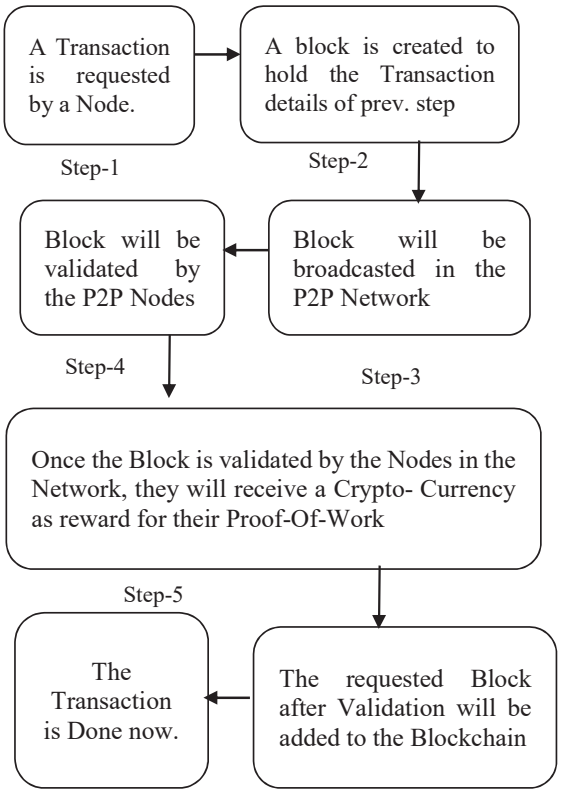

Figure 4 Workflow in blockchain.

2.2 Security Concerns In Blockchain

Figure 5 depicts an overview of the various threats to Blockchain security. At different levels, there are the different elements that are actually the matter of concern in the scenario of Blockchain security. The attackers target the vulnerabilities at different levels to bypass the security. As Blockchain has different types of interaction with the public, organizational and individual base, the attackers attack the defence system to hijack, steal and falsify the physical infrastructure between Blockchain and the users.

2.3 Smart Contacts (SC)

Simple agreement or consensus between two peers expressed as programme code is what smart contracts (SC) are. Because they are a component of Blockchain, smart contracts are stored in a single database and have an immutable nature. SC essentially consists of blockchain-based scripts that run when certain criteria are met [8]. They commonly are utilized to computerize the execution of a consensus (shown in Figure 6) so that all the members can be promptly sure of the result, with no go-between contribution or loss

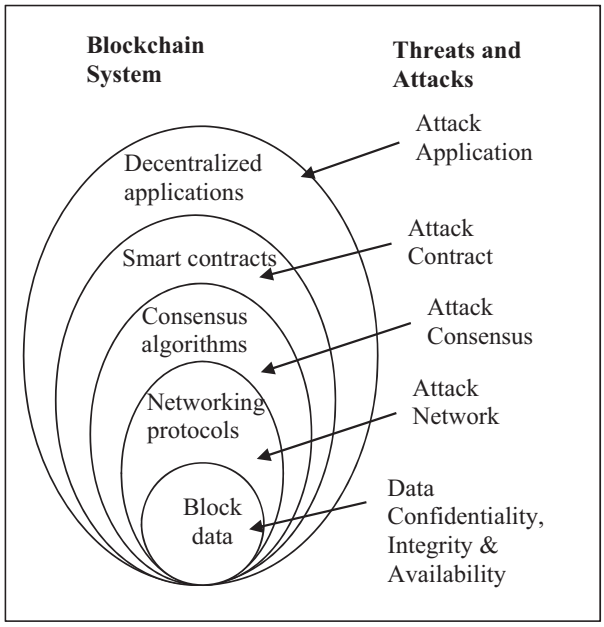

Figure 5 Common external threats to blockchain [7].

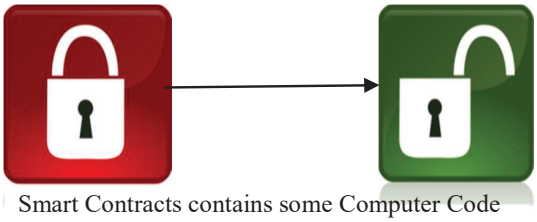

Figure 6 Common structure of smart contract.

of time. The exchanges that occur in a smart contract are handled by the blockchain, which implies they can be conveyed automatically without any of the third parties. An SC is a self-implementing consensus implanted in PC code oversaw by a blockchain. Inside an SC, there could be however many specifications on a case-by-case basis to convince the members that the undertaking will be finished agreeably. The title SC itself is somewhat unexpected since they are neither especially smart nor are they to be mistaken for a legitimate agreement: (1) An SC must be just about as smart as an individual code it, considering all accessible data at the hour of coding; (2) While SC may have possibly authorized lawful agreements if certain conditions are met, we first need to determine numerous techno-lawful inquiries which will require time and interdisciplinary talk among attorneys thus product engineers. Besides, SC security is as yet an issue that should be settled on a specialized level. We can likewise have to carry out more refined legally binding provisions, including decentralized conflict remuneration tools. Notwithstanding, we as of now actually need best practices, and will most likely need some an ideal opportunity to go through an aggregate learning stage.

A smart contract (SC) is an understanding between at least two or more groups (as shown in Figure 6), which is implemented by the PC code without enabling either group to back out, therefore it guarantees the trust less accomplishment [9]. The SC is quite possibly the main highlights in blockchain usages, which carries out reliable transactions without any involvement of a third party. Though with the exponential development, the blockchain SC has likewise uncovered numerous security issues, and a few attacks brought by the vulnerable contract may lead to unfortunate losses. To more readily manage such difficulty, making a complete overview of the security check of blockchain SC from major logical databases is very irreplaceable. Although the importance of observing security validation of blockchain SC is clear, it

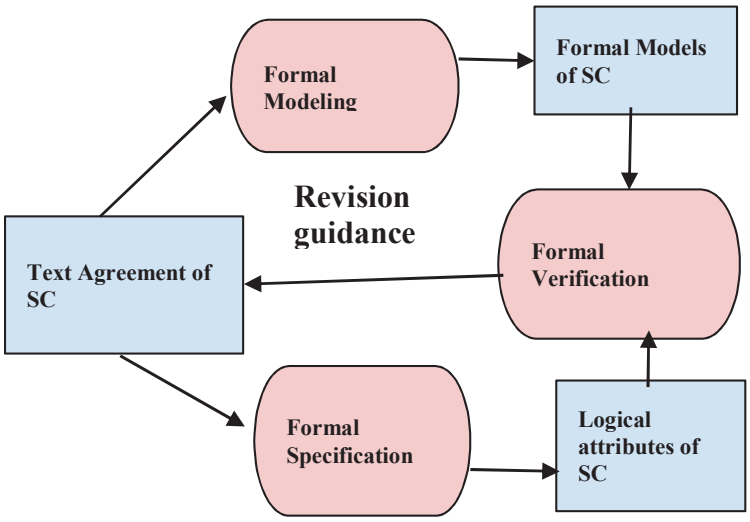

Figure 7 Common workflow diagram of smart contract.

is truly new yet. Figure 7 represents the workflow diagram of Smart Contract and how the agreement is stored and processed.

2.4 Major Security Concerns in Terms of Smart Contract

Irreversible, transparent, and traceable are the major pros of smart contracts (SC). Alongside the advantages of utilizing an SC, there are some security concerns innate to the cycle. SCs depend upon the blockchain technology that gives record keeping to the cryptocurrency network. Ethereum, the 2nd biggest cryptocurrency, has supposedly more than 32,000 SCs which are unsafe against hacking because of deficient coding [10]. A few of the vulnerabilities reported regarding SCs are discussed below:

- TOD (Transaction Ordering Dependence) – Exclusively a miner that locks the block concludes the order of transaction, and this vulnerability is considered as TOD. In TOD, the miners may unexpectedly behave maliciously.
- Dependency on Timestamp: Tragically, the miner has the ability to control the timestamp and the problem arises because the miner is generally not a friend of the user. For his personal profit, a miner can change the value of the timestamp anytime, which results in the change of output.

- Re-entrancy: The primary variant of re-entrancy included functions that could be often invoked before the previous function call was completed. Hence, various calls can get interconnected in the wrong way.

3 Literature Review

A good count of blockchain's security-related reviews has been done from alternate points of view, featuring some research breaks and future analytic bearings for scholastics and experts. [11] assessed various blockchain security problems, like, bilk attack, information scale, and cost issues. Though, the investigation isn't extensive enough for directing subsequent research. [12] led analysis on security commination to blockchain and studied the relating genuine attacks by analyzing mainstream blockchain frameworks like Ethereum. This review dismisses security concerns while using these frameworks to fabricate business applications. [13] reviewed agreement conventions from the information security and protection insurance viewpoint. Nonetheless, there exist security issues from different perspectives, like Smart Contracts. [14] outlined blockchain analysis from the viewpoint of computerized business change with its future advancements. Though, it shortfalls on a conversation on the security problems of blockchain. [15] received the blockchain reliability answer for distributed computing. Nonetheless, other application situations were avoided with regard to the review scope. [16] introduced a deliberate literature survey of blockchain-based implementations across various areas. still, the examination shortfalls on a conversation on blockchain security. Besides, numerous new blockchain security considerations have been led over the most recent two years and subsequently should be incorporated. Blockchain security is a major issue that should be reviewed as a component installed in the entire lifecycle of the system, from prerequisite examination to coding and support [17]. Still, researchers from the data frameworks region seem to have a narrow perception of the commitments accomplished by the analysts of different controls. It is consequently important to concentrate on how and how much blockchain security problems have been tended to and afterward sum up to significant analysis systems in past research. This investigation embraces the data systems point of view and presents a precise overview of blockchain security from various layers appeared in [18]. A security infrastructure proposed by [19] distinguished the requests for a coordinated design to accomplish the most extreme security. Also, the following two models presented by two IT organizations (i.e., Oracle & IBM) depicts ideas in security. IBM [20]

gave a perspective on security, comprising technology, business, administration, and a combination of the above domains that will in general share normal components. Oracle offered a reference design that incorporates three basic angles for accomplishing security, specifically, information security, extortion avoidance and detection, and consistency enablement. To finish the exchange of data between the members, SCs uncover the transaction record at the insecurity of losing delicate data. Otherwise, the challenges of management emerge [21]. Since SCs handover worth, its right execution, as well as secure execution against any attacks or altering, are very critical [22]. Vicious can use Criminal SCs (CSC), another latest digital weaponry, to deliver 0-day susceptible transactions or information exchanges. In [23], the security issues of SC are delivered in three aspects or levels, specifically, business, virtual machine, and code of agreement. Precisely, the first level of security issue which is business-level incorporates unapproved access, uncertain state, presence of malicious code, and TOD. The next level of security issue which is virtual machine incorporates the limit of the stack size, producing irregularity, and time imperatives. The last level of security issue is the code of agreement of SC, a basic issue to the blockchain [24]. The SCs are perhaps the most remarkable origin of safety issues on the cycle level of blockchain. Any SC which has been issued into the blockchain cannot be altered and hence, if any false or malicious data have been transacted it is almost impossible to delete because of strong consensus. The extent of transactions engaged in SC is huge, and more functional situations are expected to test the framework solidness to discover potential code flaws. The survey [25] delivered in 2018, "Finding The Greedy, Prodigal, and Suicidal Contracts at Scale" observed that one out of twenty SCs might be hacked because of security vulnerabilities. The article recognized three sorts of SCs which are especially unsafe mentioned in the title which additionally lock the funds endlessly, disclose them to random users, or be sensitive to be halted or removed by any client. As per [26], SC could be claimed in additional challenging situations, and the intricacy and the technical issues of the agreement code may likewise grow proportionately. Common security-based issues at the code-level incorporate calls to the obscure, and state of deadlock [27]. The scholars [28] discussed the option to recognize the susceptibility without getting to the source code, by utilizing "MAIAN", an open-source SC analysis tool in 10 seconds of analysis. While the research could not detect the particular SC which exactly had vulnerabilities. The research depicts that detection is probably possible, furthermore, the subsequent people that identify the point may have a malevolent plan. Ethereum

(cryptocurrency) is no more unknown to SC hacking. Parity Wallet, an Ethereum client, is considered a digital wallet that has been hacked in the year 2017, and with around 34 million dollars taken out. It is assessed that the hackers had taken out a sum of 2 billion dollars since 2017. Mike Orcutt, expresses that utilizing SCs for investment assets can be especially unsafe as they manage huge measures of cryptocurrencies. A survey by [29] has reported, the incidence of 2016 where the hackers stolen 60 million dollars of DAO (Decentralized Autonomous Organization) because of a flaw in SC. As per Orcutt's explanation, "A flaw in a live SC can make a rare kind of crisis". Research done in [30, 31], stated "In conventional software, an error can be corrected with a fix in programming, but not in the blockchain as it supports immutable transactions. A few methods [32, 33] being utilized to resist SC hacking incorporate the utilization of AI to observe the dubious activities or investigated issues. Auditing mechanisms are additionally being uncovered to distinguish flaws before the SC is delivered. As per [34] Uncertain states and Re-entrancy are likewise normal programming vulnerabilities [35]. Inability to encode a right state machine (fail to enquire the present status and discarding explicit changes) is the most regularly noticed issue. To confirm the security of SCs, it is most importance to catch their connotations and security belongings from the compiled bytecode during execution [36, 37]. The author in [35] offered "securify", a security analyser, which accordingly extricates exact semantic data of the security of SC from the code. To assess the rationale security of SC code, a few researchers have offered the computerized security measure technique [38, 39] and the emblematic execution framework [40] to distinguish security flaws. In the phase of the agreement plan, a semantic structure is required for supporting the plan of agreements and creating SCs by means of a collection of configuration series automatically [41, 43].

The integration of blockchain into a variety of industries, including healthcare, logistics, banking, energy, manufacturing, retail, and life science, is progressing steadily. The use of blockchain has also reached the field of higher education. It has been noted that blockchain technology has the potential to revolutionise higher education through document validity, transparency, immutability, and trust [45].

TAM has been shown to be an effective model for predicting new technology adoption intentions; yet, in some environmental circumstances, it must also be adjusted [46]. TAM was expanded by perceived security, privacy, and trust in keeping with the same. Studies have extensively covered the significance of security and privacy and shown that it acts as a barrier to the

adoption of new technology [47]. The threat of security was seen as impeding the adoption of new technology. Perceived security and privacy describe how someone thinks that access to, use of, and exposure of personal information will be kept private [48, 49].

4 Idea Proposed as Solution: Block-Hash Signature

With the coming of overpowering exploration towards designing quantum PCs and their consequences on presently distributed non-quantum counteractive cryptographic natives, there has been a rush in the plan of cryptographic natives which are quantum hostile [42]. Digital Signature is one of the crudes, which produces the electronic unique finger impression of a given note.

In this paper, we present a minuscule signature technique that is productive quantum-safe called "Block-Hash" signature blockchain applications specifically for smart contracts. In this way, our commitments are summed up as follows.

- For the signature, private key, and public key of 3n, 2n, and n bits, respectively, Block-Hash Signature (BHS) provides n-bit security. BHS is based on the concept of hampered signatures, which means that it will first encode the XOR data of two specific n-bit segments of the private key using an n-bit restricted value called "tr." Then, this encoded value will be used as part of the signature. This value, which is also known as a reserved key and is used to determine the value of "tr," is supplied after a predetermined period of time (determined by the network bandwidth) and is then used for confirmation.
- BHS is highly compact and effective since it frequently starts with a straightforward hash function, encrypts or decrypts data, and generates the digital signature and confirmation procedure using just two hash occurrences and a single call.

4.1 Notations Used

Below are the list of notations and values used in the proposed idea and its implementation:

- n is a any random positive number (integer).
- $X_2 = \{0, 1\}$, X_2^V of dimension $= \{d = (d_0,\ldots, d_v - 1) \mid d_i \in \{0, 1\}\}$, the vector space (binary) with dimension v.
- '+' is either of pairwise XOR or XOR, that too depending upon the context.

- $\{0, 1\}^*$ represents the set of all the bitstreams.
- i, j $\square$ X_2^n, i$\|$j provides two bitstreams concatenation of i and j.
- H represents the domain of hash functions (collision-resistant) from $\{0, 1\}^*$ to X_2^n.
- F represents the domain of encryption algorithms (symmetric) ct = Encr_k(msg) that is safe under the CPA (chosen-plaintext attack), where cipher-text (ct), message (msg), and key (k) are all of n-bit. The respective decryption is denoted by msg = Decr_k(ct).
- A represents the domain of consecutive memory-hard encryption procedure (symmetric) a(k, msg) where all the three 'k', 'msg', and 'ct' are vectors of n-bit, and the ciphertext is indicated as ct = MEncr_k(msg). and, similarly, msg = MDecr_k(ct).
- Here the significance of consecutive memory-hard is that the parallel algorithms can't asymptotically accomplish the efficiency benefit than the non-parallel ones.

4.2 Algorithms In BHS Scheme

BHS is overall comprised of three algorithms in which two are polynomial-time (probabilistic) named "Generation", "Signing", and one is deterministic that is "Verification".

1. **Generation Algorithm:** This algorithm is used to generate key. It takes security parameter 1^n as input and generates output (pk, sk) as public and private key pairs in the given way. Evenly select i,j $\square$ X_2^n, and adjust the private key value sk = (i, j). The respective pk will be, hash of (sk), similarly select h $\square$ HF and determine pk = h(i$\|$j).
2. **Signing Algorithm:** This algorithm creates a signature as output by taking sk and msg as inputs. The output signature is denoted by Sign_{sk}(msg) and the steps for its computation is given below:
   i. For message, msg $\square$ $\{0, 1\}^*$, the Signature algorithm evenly picks h $\square$ HF and computes st = h(msg) $\square$ X_2^n. We denote st = (st_0, st_1,…,st_{n-1}).
   ii. It computes ts = (ts_0, ts_1,…, $ts_{(n-1)}$ $\square$ X_2^n and ct by evenly choosing z $\square$ X_2^n:

$$\text{ts(l)} = \begin{cases} a(l) & if\ st(l) = 0 \\ b(l) & if\ st(l) = 1 \end{cases}$$

$$\text{ct} = \text{Encr}_r(\text{i} + \text{j}) \quad \text{where r} = \text{h}(\text{st}\|\text{z}). \tag{1}$$

iii. The signature contains two modules, ts and ct, and is expressed as Sigsk(msg)= (ts, ct). When (message, Sigsk(msg)) is released, the arbitrary integer z will be published after st > 0 instant. We refer to (ts, c) as a signature module and (u, st) as a reserve key with time-delayed delivery. It incorporates two modules which are ts and ct, i.e., $\mathrm{Sig_{sk}}$(msg) = (ts, ct). The arbitrary integer z will be published after δst > 0 moment after (msg, $\mathrm{Sig_{sk}}$(msg)) is released. We call (ts, c), a signature module, and u an reserve key with δst time-delayed conveyance.

3. **Verification Algorithm:** This algorithm generates conclusive verification by taking pk, msg, and signature as input and generates outputs in form of bit σ. The value of σ verifies whether it is valid ($\sigma = 1$) or invalid ($\sigma = 0$). σ is estimated as follows:

   i. Retrieving the summation of sk: An auditor or verifier node accepts (msg, ts, ct) first, and further evaluates the value of st = h(msg)=($\mathrm{st_0}$,..., $\mathrm{st_{n-1}}$) $\square$ $\mathrm{X_2^n}$. Post δst interval of time, it gets z. Upon receiving z, it then computes value of tr = h(st$\|$z), and then decrypts ct to get (i + j), i.e.,

$$\begin{aligned} &\mathrm{i + j = Decr_{tr}(ct)}, \\ &\text{where } \mathrm{tr = h(st\|z)}. \end{aligned} \tag{2}$$

   ii. sk retrieving process: With the signature module ts of the signature, st, and (i + j), the private-key i and j can be retrieved as below. For L = 0, 1,..., n − 1,

$$\begin{aligned} &\text{if} \mathrm{st_L = 0}, \text{ then } \mathrm{ts_L = i_L}, \\ &\text{computes } \mathrm{j_L = ts_L + (i_L + j_L)} \\ &\text{if } \mathrm{st_L = 1}, \text{ then } \mathrm{ts_L = j_L}, \\ &\text{computes } \mathrm{i_L = ts_L + (i_L + j_L)} \end{aligned} \tag{3}$$

   iii. The term of σ is calculated by

$$\sigma = \begin{cases} 0 & if\ pk \neq h(i\|j) \\ 1 & if\ pk = h(i\|j) \end{cases} \tag{4}$$

We illustrate this expression as $\sigma = \mathrm{Ver_{pk}}$(msg, Sign). i.e., if $\sigma = 1$, then it represents a successful verification, otherwise failure.

5 Implementation and Result Discussion

This section provides the abstract of the BHS signature strategy with the addition of non-delayed and delayed category using SPIX (a single-pass authenticated encryption algorithm) and Advanced Encryption Standard (AES provides a design principle termed as a replacement–permutation network, which is efficient in hardware as well as software), (state size of 256 bits) in the sponge framework. Here, firstly a brief overview is given about the significance of δst (time interval) and furthermore provides the execution results.

5.1 Time Interval δst

δst represents the sum of time for generating the signature (except the phase of key generation) and the procedure of verification. Thus, to get the need of protocol design for security with the reserve key z, we have to assure that δst meets the following provision,

$$\delta st > \frac{DT}{TR} \tag{5}$$

and

$$TR > \frac{DT}{\delta st} = TR' \tag{6}$$

where DT is the data transmitted (MB per transaction) and TR is the rate of transmission, also, $\frac{DT}{\delta st}$ as TR$'$. Fort the non-delayed scheme,n it represents the computation time as consequently memory-hard.

5.2 Time Interval δst Relates Blockchain Transactions

5G, 4G, 3G, and WiFi all transmit data at rates of 1 Gbps, 75 Mbps, 2.4 Mbps, and 50 to 320 Mbps, respectively. The Bitcoin (cryptocurrency) Core protocol limits the size of the block for the blockchain to 1 MB, and each block can only contain 4,000 transactions at most [40]. DT will therefore be 8/4000, or 0.002 MB, every transaction. Therefore, when the transmission rate TR $>$ 0.002/st, it will become secure and safer.

5.3 Instantiation of Block Hash using Sponge Function

To assess the performance of the current signature plot, the execution incorporates three stages i.e., key generation stage, signing stage and finally

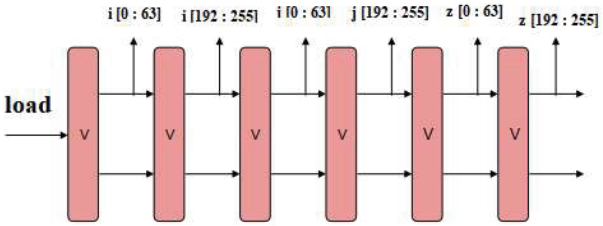

Figure 8 Sponge Structure to generate i, j, and z.

verification stage. Three of the stages utilize the sponge* construction of 256-bits of SPIX [41] to present hash, decryption and encryption. Additionally, we utilize precisely the same sponge function for correlation with AES aside from that the 256-bits of SPIX is supplanted by 256-bits AES transformation [42, 44]. The successive memory-hard scheme of encryption is carried out by mounting the rate for encryption to implement the transformation once to encode the one bit plain text, alluded to as non-postponed cases in the execution.

**Sponge Construction:* In terms of cryptography, the Sponge function is a technique that creates a mapping function from variable-size input to variable-size output based on a lining rule and a constant-length transformation [43]. Such operations are known as sponge functions because they accept binary strings of varying sizes as inputs and produce binary strings of the desired or goal size as outputs. The sponge function is a development of the ideas of stream cyphers and hash functions with a consistent output size. It works on a limited state by repeatedly applying the internal transformation to it, and interleaved with the passage of information or the recovery of output as shown in Figure 8.

To create the vectors i, j and z, we utilize the sponge representation shown in Figure 9 where V is either AES or SPIX. The expression is first stacked (loaded) with an arbitrary value and afterward 64-bits of expression is drawn out each single time to get i, j and z. For producing pk and ct, we utilize the hash function portrayed in Figure 10. The expression is first stacked (loaded) with a fixed input vector. Then, at that point the message (msg $\square$ $\{i\|j, st\|z\}$) is absorbed or ingested into the 64-bits expression at a time. After the retaining or absorbing stage, 256 bits digest message $HF_0\|HF_1\|HF_2\|HF_3$ is received as output. For the signing stage, we utilize the Encryption algo as demonstrated in Figure 10. The starting state is first stacked or loaded with the confidential value tr and afterward the transformation (permutation) is claimed.

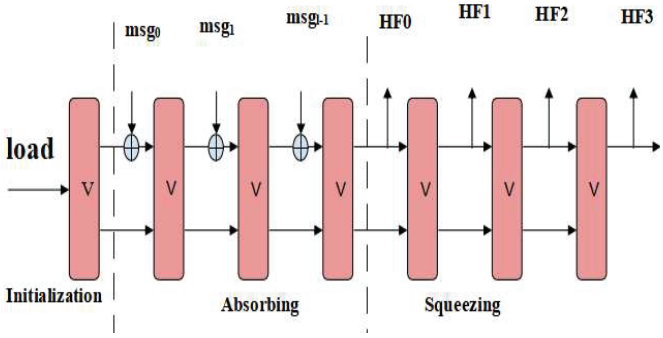

Figure 9 Hashing algorithm via a sponge mode.

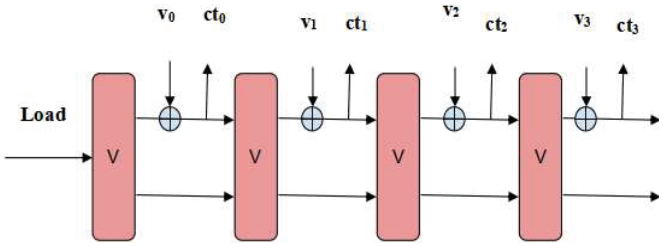

Figure 10 Encryption algorithm via a sponge mode.

Table 1 Conduct of block hash using SPIX and AES

Strategy (256-bit)	Generation (ms)	Signing (ms)	Verification(ms)
Delayed SPIX	2.36	0.75	1.63
Delayed AES	1.81	0.50	1.12
Non Delayed SPIX	2.31	9.61	10.47
Non Delayed AES	1.81	2.59	2.74

Afterwards, each 64 bit plain text v_l is taken absorbed into the mode and the respective 64 bits cipher text ct_l is received as output. The decryption procedure is closer to encryption one and therefore the details are taken off.

6 Implementation and Result Analysis

The results on the basis of performance of Block Hash Signature and the respective values of TR′ and δst has been represented through Tables 1 and 2.

A comparison of BHS with other existing methodologies has been shown in Table 3. As we can see the latency of BHS is smaller when compared with

Table 2 The parameters δst and TR$'$

Strategy (256-bit)	δst (ms)	TR$'$ (mbps)
Delayed SPIX	2.38	0.840
Delayed AES	1.62	1.235
Non Delayed SPIX	20.08	0.0996
Non Delayed	5.33	0.375

Table 3 Differentiation of Cost (Resource) and latency with existing ones

Strategy (256-bit)	Resources	Latency
ECC	Gates = 15 K	75 000
NIST HW-SW	Gates = 11.7 K	6 000 000
NIST SW	I. HW Multiplier II. Code = 34 KB	15 584 000
XMSS	I. Gates = 13.5 K II. Code = 5.22 KB	4 814 160
Proposed BHS	2742 (application-specific integrated circuit (ASIC) = 130 nm) or 2611 (ASIC = 65 nm)	4032

Table 4 The size of private key, public key, and signatures in bits

	Private Key	Public Key	Signature
Winternitz	256	$\frac{n}{w}$ n $= 32 \times 256$	$\frac{n}{w}$ n $= 32 \times 256$
XMSS	256	$\frac{n}{w}$ n $= 32 \times 256$	$\frac{n}{w}$ n $= 32 \times 256$
SPHINCS	256	$\frac{n}{w}$ n $= 32 \times 256$	$\frac{n}{w}$ n $= 32 \times 256$
BHS	256	$2 \times$ n $= 2 \times 256$	$3 \times$ n $= 3 \times 256$

others. ECC (elliptical curve cryptography) is seems to be second compact one is multiple times more than BHS. In Table 4 the comparison of key size has been represented.

By fixing a similar size of public key and accomplishing a similar level of security, BHS has much compact private key and size of signature when compared with SPHINCS, XMSS, and Winternitz. In Figure 4, the size of public-key for every one of those methods can make onto the size n utilizing a hash structure, yet the verification is considerably more costly contrasted to our plan, therefore, we just have one call require for the hash to produce the public-key.

7 Conclusion

This paper presents the "Block-Hash Signature," a lightweight signature approach that offers 112-bits of blockchain security. It has three 256-bit signatures, a two 256-bit public key, and a single 256-bit private key (SHA-256 Algorithm for Hashing). The security of smart contracts (SC), a part of Hyperledger Fabric, is the area of particular focus. According to the data, even if smart contracts are very safe and secure, smart card hacking has been documented numerous times, costing billions of dollars. Strict security requirements are provided by the Block Hash Signature, which must be signed by both the endorser and committer node. The agreements are rigorously validated before the transaction is recorded into the blockchain. The cryptographic hash will only receive a maximum of two calls. The Advanced Encryption Standard (AES) and single-pass authenticated encryption method performance analysis is also included in this research (SPIX). It has been discovered that using the signature system to sign smart contracts is quite effective.

References

[1] Ekblaw, A. Azaria, J.D. Halamka, A. Lippman, "A case study for blockchain in healthcare: medrec prototype for electronic health records and medical research data", 2016. URL: https://www.media.mit.edu/publications/medrecwhitepaper/.

[2] S. Huckle, R. Bhattacharya, M. White, N. Beloff, "Internet of things, blockchain and shared economy applications", Proc. Comput. Sci. 98 (2016) 461–466.

[3] A. Dorri, S.S. Kanhere, R. Jurdak, P. Gauravaram, "Blockchain for iot security and privacy: The case study of a smart home", in: IEEE Percom Workshop on Security Privacy and Trust in the Internet of Thing, 2017.

[4] Xiaoqi Li, Peng Jiang, Ting Chen, Xiapu Luo, Qiaoyan Wen, "A survey on the security of blockchain systems", https://doi.org/10.1016/j.future.2017.08.020, Volume 107, June 2020.

[5] P. Bailis, A. Narayanan, A. Miller, and S. Han, "Research for practice: Cryptocurrencies, blockchains, and smart contracts; hardware for deep learning," *Commun. ACM*, vol. 60, no. 5, pp. 48–51, 2017.

[6] BlockGeeks. (2017). 17 blockchain applications that are transforming society. [Online]. Available: https://blockgeeks.com/guides/blockchain-applications/

[7] Y. Yuan and F. Wang, "Blockchain and Cryptocurrencies: Model, Techniques, and Applications," IEEE Transactions on Systems, Man, and Cybernetics: Systems, 48, pp. 1421–1428, (2018).
[8] M. Crosby, Nachiappan, P. Pattanayak, S. Verma, and V. Kalyanaraman, "BlockChain Technology: Beyond Bitcoin," Applied Innovation Review, 6, pp. 1–16, (2016).
[9] Y. Zhang, J. Wen, The IoT electric business model: Using blockchain technology for the internet of things, Peer-to-Peer Netw. Appl. (2016) 1–12.
[10] J. Sun, J. Yan, K.Z. Zhang, Blockchain-based sharing services: What blockchain technology can contribute to smart cities, Financ. Innov. (2016) 26.
[11] X. Xu, C. Pautasso, L. Zhu, V. Gramoli, A. Ponomarev, A.B. Tran, S. Chen, The blockchain as a software connector, in: The 13th Working IEEE/IFIP Conference on Software Architecture, WICSA, 2016.
[12] E. Nordström, Personal Clouds: Concedo (Master's thesis), Lulea University of Technology, 2015.
[13] J.S. Czepluch, N.Z. Lollike, S.O. Malone, The use of block chain technology in different application domains, in: The IT University of Copenhagen, 2015, Copenhagen.
[14] Ethereum, Etherscan: The ethereum block explorer, 2017. URL https://www.ethereum.org/.
[15] L. Luu, D.-H. Chu, H. Olickel, P. Saxena, A. Hobor, Making smart contracts smarter, in: The 2016 ACM SIGSAC Conference on Computer and Communications Security, 2016, pp. 254–269.
[16] V. Buterin, Critical update re: Dao vulnerability, 2016. URL https://blog.ethereum.org/2016/06/17/critical-update-re-dao-vulnerability/.
[17] J. Adelstein, Behind the biggest bitcoin heist in history: Inside the implosion of mt.gox, 2016. URL http://www.thedailybeast.com/articles/2016/05/19/behind-the-biggest-bitcoin-heist-in-history-inside-the-implosion-of-mt-gox.html.
[18] N. Atzei, M. Bartoletti, T. Cimoli, A survey of attacks on ethereum smart contracts (sok), in: international Conference on Principles of Security and Trust, 2017, pp. 164–186.
[19] Z. Zheng, S. Xie, H.-N. Dai, H. Wang, Blockchain challenges and opportunities: A survey, Internat. J. Web Grid Serv. (2016).
[20] Shuai Wang, Liwei Ouyang, Yong Yuan, *Senior Member, IEEE*, Xiaochun Ni, Xuan Han, and Fei-Yue Wang, "Blockchain-Enabled Smart Contracts: Architecture, Applications, and Future Trends", IEEE

Transactions on Systems, Man, and Cybernetics: Systems (Volume: 49, Issue: 11, Nov. 2019), DOI: 10.1109/TSMC.2019.2895123.

[21] Jing Liu and Zhentian Liu, "A Survey on Security Verification of Blockchain Smart Contracts" IEEE Access (Volume: 7), DOI: 10.1109/ACCESS.2019.2921624.

[22] Alboaie, S., Cosovan, D., Chiorean, L., Vaida, M.F.: Lamport n-time signature scheme. In: 2018 IEEE International Conference on Automation, Quality and Testing, Robotics (AQTR), pp. 1–6 (May 2018).

[23] AlTawy, R., Gong, G., He, M., Mandal, K., Rohit, R.: SPIX: an authenticated cipher round 2 candidate to the NIST LWC competition (2019).

[24] Altawy, R., Rohit, R., He, M., Mandal, K., Yang, G., Gong, G.: SLISCP-light: towards hardware optimized sponge-specifific cryptographic permutations. ACM Trans. Embed. Comput. Syst. 17(4), 81:1–81:26 (2018).

[25] Bernstein, D.J., et al.: SPHINCS: practical stateless hash-based signatures. In: Oswald, E., Fischlin, M. (eds.) EUROCRYPT 2015. LNCS, vol. 9056, pp. 368–397. Springer, Heidelberg (2015). https://doi.org/10.1007/978-3-662-46800-515

[26] Bertoni, G., Daemen, J., Peeters, M., Van Assche, G.: Sponge functions. In: ECRYPT Hash Workshop, vol. 2007. Citeseer (2007).

[27] Bindel, N., et al.: Submission to NIST's post-quantum project: lattice-based digital signature scheme qTESLA (2018).

[28] Bosmans, J., Roy, S.S., Jarvinen, K., Verbauwhede, I.: A tiny co-processor for elliptic curve cryptography over the 256-bit NIST prime field. In: 2016 29th International Conference on VLSI Design and 2016 15th International Conference on Embedded Systems (VLSID), pp. 523–528 (January 2016).

[29] Buchmann, J., García, L.C.C., Dahmen, E., D¨oring, M., Klintsevich, E.: CMSS – an improved merkle signature scheme. In: Barua, R., Lange, T. (eds.) INDOCRYPT 2006. LNCS, vol. 4329, pp. 349–363. Springer, Heidelberg (2006). https://doi.org/10.1007/1194137825.

[30] Butin, D.: Hash-based signatures: state of play. IEEE Secur. Priv. 15(4), 37–43 (2017).

[31] Chalkias, K., Brown, J., Hearn, M., Lillehagen, T., Nitto, I., Schroeter, T.: Blockchained post-quantum signatures. In: 2018 IEEE International Conference on Internet of Things (iThings) and IEEE Green Computing and Communications (GreenCom) and IEEE Cyber, Physical

and Social Computing (CPSCom) and IEEE Smart Data (SmartData), pp. 1196–1203 (July 2018).
[32] Chalkias, K., Brown, J., Hearn, M., Lillehagen, T., Nitto, I., Schroeter, T.: Blockchained post-quantum signatures. IACR Cryptol. ePrint Arch. 2018, 658 (2018).
[33] Cruz, J.P., Yatani, Y., Kaji, Y.: Constant-sum fifingerprinting for Winternitz one time signature. In: 2016 International Symposium on Information Theory and its Applications (ISITA), pp. 703–707 (October 2016).
[34] Daemen, J., Rijmen, V.: The Design of Rijndael: AES-The Advanced Encryption Standard. Springer, Berlin (2013). https://doi.org/10.1007/978-3-662-04722-4
[35] Dods, C., Smart, N.P., Stam, M.: Hash based digital signature schemes. In: Smart, N.P.(ed.) Cryptography and Coding 2005. LNCS, vol. 3796, pp. 96–115. Springer, Heidelberg (2005). https://doi.org/10.1007/115868218.
[36] Kumar, Pravin, Dayal, Mohit, Khari, Manju, Fenza, Giuseppe, Gallo, Mariacristina, "NSL-BP: A Meta Classifier Model Based Prediction of Amazon Product Reviews", International Journal of Interactive Multimedia & Artificial Intelligence. June 2021, Vol. 6 Issue 6, pp. 95–103. 9p.
[37] Radhika Saini, Manju Khari, "Defining Malicious Behavior of a Node and its Defensive Techniques in Ad Hoc Networks", International Journal of Smart Sensors and Ad Hoc Networks (IJSSAN) Volume-1, Issue-1, 2011.
[38] M. Khari, A. K. Garg, A. H. Gandomi, R. Gupta, R. Patan and B. Balusamy, "Securing Data in Internet of Things (IoT) Using Cryptography and Steganography Techniques," in IEEE Transactions on Systems, Man, and Cybernetics: Systems, vol. 50, no. 1, pp. 73–80, Jan. 2020, doi: 10.1109/TSMC.2019.2903785.
[39] N. Singh, M. Dayal, R. S. Raw and S. Kumar, "SQL injection: Types, methodology, attack queries and prevention," 2016 3rd International Conference on Computing for Sustainable Global Development (INDIACom), 2016, pp. 2872–2876.
[40] K. S. Sahoo et al., "An Evolutionary SVM Model for DDOS Attack Detection in Software Defined Networks," in IEEE Access, vol. 8, pp. 132502–132513, 2020, doi: 10.1109/ACCESS.2020.3009733.

[41] Mohit Dayal and Bharti Nagpal, "A compendious investigation of Android malware family", International Journal of Information Privacy, Security and Integrity-2016, doi: 10.1504/IJIPSI.2016.082127.

[42] Chitrangada Chaubey and Swapnil Raj and Suresh Kaswan, "Security and Privacy Issues in Location Dependent Services for Mobile Communication: A Synergistic Review", IOP Conf. Ser.: Mater. Sci. Eng. 1149 012007, doi: 10.1088/1757-899x/1149/1/012007.

[43] Faisal Jamil, Muhammad Ibrahim, Israr Ullah, Suyeon Kim, Hyun Kook Kahng, Do-Hyeun Kim, Optimal smart contract for autonomous greenhouse environment based on IoT blockchain network in agriculture, Computers and Electronics in Agriculture, Volume 192, 2022, 106573, ISSN 0168-1699, https://doi.org/10.1016/j.compag.2021.106573.

[44] John, A., Reji, A., Manoj, A.P., Premachandran, A., Zachariah, B., Jose, J. (2022), "A Novel Hash Function Based on Hybrid Cellular Automata and Sponge Functions", In: Das, S., Martinez, G.J. (eds) Proceedings of First Asian Symposium on Cellular Automata Technology. ASCAT 2022. Advances in Intelligent Systems and Computing, vol. 1425. Springer, Singapore. https://doi.org/10.1007/978-981-19-0542-1_16.

[45] Grech, A., Camilleri, A.F.: Blockchain in Education. JRC Science for Policy Report, European Commission (2017).

[46] Mezhuyev, V., Al-Emran, M., Fatehah, M., Hong, N.C.: Factors affecting the meta modelling acceptance: a case study from software development companies in Malaysia. IEEE Access. 10(6), 49476–49485 (2018).

[47] Kumar, Nishant, et al. "Blockchain Adoption for Provenance and Traceability in the Retail Food Supply Chain: A Consumer Perspective." IJEBR vol. 18, no. 2, 2022: pp. 1–17. http://doi.org/10.4018/IJEBR.294110.

[48] Kumar, N., Singh, M., Upreti, K., Mohan, D. (2022). Blockchain Adoption Intention in Higher Education: Role of Trust, Perceived Security and Privacy in Technology Adoption Model. In: Al-Emran, M., Al-Sharafi, M.A., Al-Kabi, M.N., Shaalan, K. (eds) Proceedings of International Conference on Emerging Technologies and Intelligent Systems. ICETIS 2021. Lecture Notes in Networks and Systems, vol. 299. Springer, Cham. https://doi.org/10.1007/978-3-030-82616-1_27.

[49] Kumar, N., Upreti, K., Upreti, S., Shabbir Alam, M., & Agrawal, M. (2021). Blockchain integrated flexible vaccine supply chain architecture: Excavate the determinants of adoption. *Human Behavior and Emerging Technologies*, 3(5), 1106– 1117. https://doi.org/10.1002/hbe2.302.

Biographies

Sonika Bhatnagar, Lecturer Computer in Department of Computer Science And Engineering at Government Polytechnic Baheri Bareilly. She Completed her M.Tech in Computer Science and Engineering from HR Institute of Technology Ghaziabad affiliated from Dr. A.P.J Kalam Technical University Lucknow.

Mohit Dayal is Technical committee Member of IEEE INDIACom international conference, Delhi and Editorial member of International journal of Recent Advances in Science and Technology. He is currently working in Bharati Vidyapeeth's College of Engineering, Paschim Vihar, New Delhi as an Assistant Professor in Department of Applied Science. He received his master's degree in Information Security from Ambedkar Institute of Advanced Communication Technologies & Research of Guru Gobind Singh Indraprastha University, Delhi. He holds a bachelor's degree in Computer Science & Engineering from Guru Gobind Singh Indraprastha University, Delhi. His research interests include machine learning, Internet of Things, Big Data, Web Application attacks and information security.

Deepti Singh, Assistant Professor in Department of Information Technology at ABES Institute of Technology, Ghaziabad. She Completed her M.Tech in Information Security from Guru Gobind Singh Indraprastha University, Delhi.

Shitiz Upreti is currently working as an Assistant Professor in the Department of Information Technology, Asian Education Group(AEG), Noida, U.P. He completed his B.Tech and M.Tech in the field of Electronics & Communication Engineering. Also, completed my MBA in the field of Production and Operation Research. Currently, he is pursuing Ph.D. in the field of IT & Wireless Communication Engineering. He has published 7 patents, 4+ books, 10+ research journals in various national and international conferences. His area of interest includes Wireless communication, Machine Learning, Cloud Computing and Data Analytics. He also attended various FDPs and workshops regarding Machine Learning, SPSS & Blockchain Technology.

Kamal Upreti is currently working as an Associate Professor in Department of Computer Science & Engineering, Dr. Akhilesh Das Gupta Institute of Technology & Management (Formerly NIEC) affiliated to Guru Govind Singh Indraprastha University, Delhi, India. He is a Corporate Trainer in HCL company in the field of Cyber Security and Data Science. He completed is B. Tech (Hons) Degree from UPTU, M.Tech (Gold Medalist) from Galgotias University, PGDM(Executive) from IMT Ghaziabad and PhD in Computer Science & Engineering. Now, he is doing Postdoc in Project: Study on Smart healthcare monitoring system based on Internet of Things (IoT) from National Taipei University of Business, Taiwan.

He has published 50+ Patents, 35+ Books, 32+ Magazine issues and 60+ Research papers in various international Conferences and reputed Journals. His areas of Interest are Cyber Security, Machine Learning, Health Care, Wireless Networking, Embedded System and Cloud Computing. He is having enriched years' experience in corporate and teaching experience in Engineering Colleges.

He has attended as a Session Chairperson in National, International conference and key note speaker in various platforms such as Skill based training, Corporate Trainer, Guest faculty and faculty development Programme. He awarded as best teacher, best researcher, extra academic performer and Gold Medalist in M. Tech programme.

Jitender Kumar is working in SDEC Ghaziabad (UP). He has completed 10th and 12th from Yaduvanshi Shiksha Niketan Mahender Garh (HR.). He has cleared CEET in 2004 and got admission in YMCAIE & Technology Faridabad in B.Tech(IT) Branch. He has cleared GATE-2008 and got admission in YMCAIE & Technology Faridabad in M.Tech(CE). He has cleared NET-21. He has 11 years of Teaching Experience. He was the coordinator of different societies in college and Universities. He has worked in RIET Faridabad, GBPIT Okhla New Delhi, AIT Shakarpur New Delhi, JCBOSE YMCAUST Faridabad.