IJARSCT
International Journal of Advanced Research in Science, Communication and Technology
International Open-Access, Double-Blind, Peer-Reviewed, Refereed, Multidisciplinary Online Journal
ISSN: 2581-9429
Volume 6, Issue 7, October 2026
Impact Factor: 8.2

Bitcoin Security: Enhancing Electronic Cash System Using Proof-of-Work (PoW)

Dr. Shilpa B. Sarvaiya[1], Saloni Gupta[2] & Maithilii Ingle[2]
[1]Head, Department of MCA
[2]Student, Department of MCA
Vidyabharti Mahavidyalaya, Amravati, Maharashtra, India
[1] sarvaiya.shilpa@gmail.com, [2]sg818664@gmail.com, [3]maithiliingle68@gmail.com

Abstract: *Electronic payments normally rely on a trusted institution to stop the same money from being spent twice. Bitcoin, proposed by Nakamoto in 2008, removes that institution by letting network participants agree on one transaction history through proof-of-work (PoW). This paper studies how PoW strengthens the security of an electronic cash system. It gives a layered view of a PoW-secured system covering digital signatures, hash-linked blocks, the mining puzzle, difficulty adjustment, and the rule that selects the chain with the most accumulated work. It then analyses double spending using the catch-up probability published by Nakamoto, maps the main attacks (majority hash power, selfish mining, eclipse attacks, and zero-confirmation fraud) to the countermeasures that resist them, and discusses the limits set by energy use, throughput, and privacy. The work is a design and literature study: it reports no new experimental results, every statement about earlier work is attributed to its authors, and the numerical values are computed from a published formula. Four figures and seven tables summarise the architecture, the block structure, the mining flow, the attack probabilities, the threats, and the limitations.*

Keywords: Bitcoin, proof-of-work, electronic cash, double spending, blockchain security, mining, consensus, cryptocurrency, cybersecurity

I. INTRODUCTION

Money that exists only as data has an obvious weakness: data can be copied. A digital coin that can be duplicated is worthless, so every electronic cash system must stop the same coin from being spent twice. The usual answer is to appoint a trusted party, such as a bank or a payment processor, that checks each payment against its own ledger. This works while the party is honest, available, and secure, but it gathers trust and risk in one place. A failure or a dishonest insider can damage every account at once, payments can be reversed, so merchants must guard against fraud, and the operator charges for the service.

In 2008 Nakamoto proposed an electronic cash system that needs no such party [1]. Payments pass directly between users, and the network agrees on their order by means of a proof-of-work (PoW) chain. The idea draws on earlier work: moderately hard puzzles proposed as a price for sending messages [2], the hashcash stamp that put such puzzles to use against abuse [3], and hash-linked timestamping of documents [4]. In Bitcoin the puzzle is attached to every block of transactions, and each block contains the hash of the block before it, so changing an old payment means redoing the work for that block and for every block after it.

The security of the system therefore rests on arithmetic and economics rather than on the reputation of an institution. Honest participants are safe as long as together they control most of the computing power, and attackers are discouraged because cheating costs more than honest mining pays. Many studies have examined parts of this argument [5]–[8], but a student or practitioner who wants to see how each ingredient of PoW contributes to security, and how long to wait before a payment of a given value can be trusted, must collect the answer from several sources. This paper brings these threads together.

Copyright to IJARSCT
www.ijarsct.co.in
DOI: 10.48175/IJARSCT-40457

IJARSCT
International Journal of Advanced Research in Science, Communication and Technology
International Open-Access, Double-Blind, Peer-Reviewed, Refereed, Multidisciplinary Online Journal
ISSN: 2581-9429
Volume 6, Issue 7, October 2026
Impact Factor: 8.2

This paper makes three contributions:
a layered description of a PoW-secured electronic cash system that shows which component provides which security property;
a worked double-spend analysis based on Nakamoto's published formula, with a table and a plot of the attacker's success probability against the number of confirmations; and
a threat analysis, a discussion of privacy and environmental concerns, and an evaluation plan, together with an honest list of limitations.
No new experimental results are claimed. Findings of earlier researchers are attributed to them, and the values in Table IV are computed from a published formula [1].

II. RELATED WORK

Earlier research touches this topic from three directions: the origins of proof-of-work and timestamping, formal and quantitative security analyses of Bitcoin, and attacks on mining and on the network. Table I lists the works discussed.

Table 1 : SUMMARY OF REVIEWED WORKS

Work	Year	Contribution relevant here
Haber & Stornetta [4]	1991	Hash-linked timestamping so that tampering with a document history is detectable
Dwork & Naor [2]	1992	Pricing through processing: a moderately hard computation as the cost of sending a message
Back [3]	2002	Hashcash: partial hash-collision stamps as an anti-abuse cost
Nakamoto [1]	2008	Peer-to-peer electronic cash secured by a proof-of-work chain
Karame et al. [9]	2012	Double-spending attacks on fast payments accepted without confirmation
Eyal & Sirer [8]	2014	Selfish mining: a pool below a majority can earn above its fair share
Garay et al. [6]	2015	Formal backbone model with common-prefix and chain-quality properties
Heilman et al. [11]	2015	Eclipse attacks on the Bitcoin peer-to-peer network, with countermeasures
Gervais et al. [7]	2016	Quantitative framework relating PoW security to protocol parameters

A. Origins of PoW and Timestamping

Dwork and Naor proposed that a sender pay a small computational price for each message, so that bulk junk mail becomes expensive [2]. Back's hashcash applied the same idea with partial hash collisions [3]. Haber and Stornetta described how documents can be timestamped by linking them with hashes, so that an altered history can be detected [4]. Nakamoto's design joins these ideas: the timestamp chain records payments, and the puzzle makes extending the chain costly [1]. The earlier works address spam and document integrity rather than payments.

B. Security Analyses of the Protocol

Garay et al. abstracted the core of Bitcoin as a backbone protocol and proved, under assumptions of a synchronous network and an honest majority of hash power, that honest nodes keep a common prefix of the chain and that honest miners contribute a fair share of its blocks [6]. Gervais et al. built a quantitative framework that links the security of a PoW blockchain to parameters such as block interval and block size, and used it to compare the cost of attacks such as double spending and selfish mining [7]. Bonneau et al. surveyed the research landscape for Bitcoin and related systems [5].

IJARSCT
International Journal of Advanced Research in Science, Communication and Technology
International Open-Access, Double-Blind, Peer-Reviewed, Refereed, Multidisciplinary Online Journal
ISSN: 2581-9429
Volume 6, Issue 7, October 2026
Impact Factor: 8.2

C. Attacks on Mining and the Network

Eyal and Sirer showed that a mining pool can earn more than its fair share by withholding blocks, even without holding a majority, a strategy known as selfish mining [8]. Karame et al. showed that merchants who accept payments without waiting for confirmation are open to double spending [9]. Decker and Wattenhofer studied how blocks and transactions spread through the network and linked propagation delay to temporary forks [10]. Heilman et al. demonstrated eclipse attacks, in which an attacker controls all the connections of a victim node, and proposed countermeasures [11].

D. Research Gap

The reviewed works are mostly written for specialists, and each usually examines one attack or one layer of the system. What remains useful is a compact view that ties every PoW component to the security property it provides and to the attacks it resists, together with numbers that a practitioner can use to decide how long to wait before treating a payment as final. That is the purpose of this paper.

III. PROPOSED METHODOLOGY

A. Overview

The methodology is analytical rather than experimental. We describe a PoW-secured electronic cash system layer by layer, derive its resistance to double spending from a published formula, and map the known attacks onto the components that resist them. Figure 1 shows the five layers. Wallets hold the private keys with which users sign payments. The transaction layer records each payment as a signed transfer of coins. In the network layer, nodes pass transactions and blocks to their peers, and every full node checks all rules on its own. In the consensus layer, miners compete to solve the PoW puzzle, and the chain with the most accumulated work is accepted. The ledger layer stores the accepted history as hash-linked blocks [12]. No layer depends on a trusted central party.

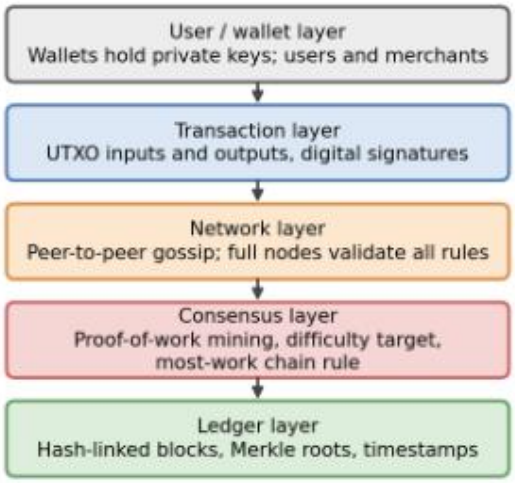


Figure 1. Layered architecture of a PoW-secured electronic cash system.

B. Block and Transaction Structure

Figure 2 shows the structure of a block. A transaction spends outputs of earlier transactions and creates new ones, and each spend is authorised by a digital signature made with the owner's private key [12]. A block bundles many transactions. Its 80-byte header holds a version number, the hash of the previous block, the Merkle root of the transactions, a timestamp, the difficulty target, and a nonce. The Merkle root is a single hash that commits the header to every transaction in the block, so a change to any one transaction changes the root, then the header hash, and then breaks the previous-hash link stored in the next block. Hashing uses SHA-256 [13], applied twice to the header. Table II summarises the components and the security role of each.

IJARSCT
International Journal of Advanced Research in Science, Communication and Technology
International Open-Access, Double-Blind, Peer-Reviewed, Refereed, Multidisciplinary Online Journal
ISSN: 2581-9429
Volume 6, Issue 7, October 2026
Impact Factor: 8.2

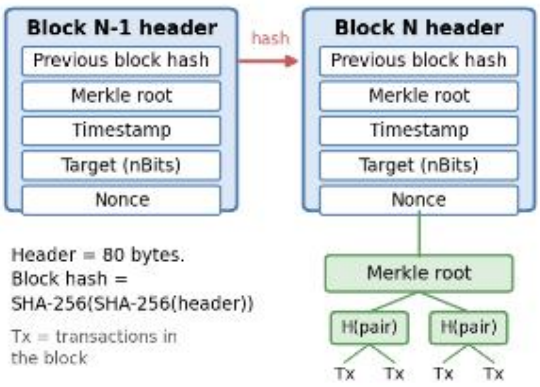


Figure 2. Block header fields and the hash link between blocks.

Table 2 : COMPONENTS AND SECURITY ROLES

Component	Security role
Digital signature	Proves the spender holds the key; stops forged payments
Hash function (SHA-256)	Links blocks and forms the puzzle; any change alters the output
Merkle tree	Commits a block to all its transactions with one 32-byte root
PoW puzzle	Makes block creation costly, so rewriting history needs real work
Difficulty adjustment	Keeps the block interval near ten minutes as hash rate changes
Most-work chain rule	Lets every node choose the same history without a coordinator
Block reward and fees	Pays honest miners, so following the rules is usually the profitable choice
Full nodes	Re-check every rule independently, so miners cannot bend the rules

C. Proof-of-Work and Difficulty Adjustment

To add a block, a miner must find a nonce for which the header hash, read as a 256-bit number, is not larger than the current target. The output of a good hash function cannot be predicted, so the only known way to find such a nonce is repeated trial. If the target is a fraction t of the 2^{256} possible outputs, each trial succeeds with probability t, and the expected number of trials is 1/t. Checking a solution takes a single hash, so every node can cheaply verify work that was costly to produce. Bitcoin expresses the target as a difficulty D relative to its easiest target, and the expected number of hashes per block is about $D \times 2^{32}$ [12].

Every 2016 blocks, nodes compare the actual time taken with the intended two weeks and rescale the target, by at most a factor of four, so that blocks keep arriving about every ten minutes whatever the total hash rate [1], [12]. Miners are paid by a block subsidy that halves every 210,000 blocks and by transaction fees, which gives them a reason to build on the accepted chain instead of attacking it. Table III lists the main parameters.

Table 3 : KEY BITCOIN PROTOCOL PARAMETERS

Parameter	Value in Bitcoin
Block header size	80 bytes
Hash applied to header	SHA-256, applied twice
Target block interval	About 10 minutes
Difficulty retarget	Every 2016 blocks (about two weeks)
Limit on one retarget	Factor of four, up or down

IJARSCT
International Journal of Advanced Research in Science, Communication and Technology
International Open-Access, Double-Blind, Peer-Reviewed, Refereed, Multidisciplinary Online Journal
ISSN: 2581-9429
Volume 6, Issue 7, October 2026
Impact Factor: 8.2

Parameter	Value in Bitcoin
Initial block subsidy	50 BTC, halved every 210,000 blocks
Total supply cap	About 21 million BTC
Timestamp rule	Later than the median of the last 11 blocks; at most two hours ahead of network time
Coinbase maturity	100 blocks before the reward can be spent
Common practice for payments	Wait for 6 confirmations

Values follow the protocol as described in [1] and [12]; current software should be checked for any later change.

D. Transaction Validation and Mining Flow

Figure 3 follows a payment through the system. A node that receives a transaction checks its format and signature, confirms that every input refers to an output that exists and is unspent, confirms that the outputs do not exceed the inputs, and rejects any transaction that conflicts with one it already holds. Valid transactions are relayed to peers. A miner collects transactions into a candidate block, adds a special coinbase transaction that pays the reward, computes the Merkle root, and tries nonces until the hash meets the target. When the 32-bit nonce space is used up, the miner changes another field, such as extra data in the coinbase, and continues. A solved block is broadcast, and each peer checks the proof of work, the previous-hash link, and every transaction before accepting it. A block that fails any test is dropped, so a miner gains nothing by including invalid payments.

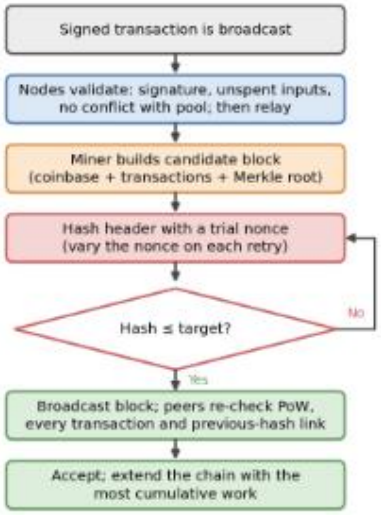


Figure 3. Validation and mining flow for one payment.

E. Chain Selection and Fork Resolution

Two miners may solve a block at nearly the same time, which creates a temporary fork. Nodes keep both branches and work on the one they saw first. When the next block arrives, the branch with more accumulated work becomes the accepted chain, the losing blocks become stale, and their transactions return to the pool. This is the chain with the most cumulative work, which equals the longest chain only while the difficulty is constant [1], [12]. Propagation delay between nodes raises the rate of such forks [10]. Because the reward is paid only in the accepted chain, honest miners have a direct reason to extend it.

IJARSCT
International Journal of Advanced Research in Science, Communication and Technology
International Open-Access, Double-Blind, Peer-Reviewed, Refereed, Multidisciplinary Online Journal
ISSN: 2581-9429
Volume 6, Issue 7, October 2026
Impact Factor: 8.2

F. Double-Spend Resistance

Consider an attacker who pays a merchant, waits until the merchant accepts the payment after z confirmations, and meanwhile mines a private chain in which the same coins are paid back to the attacker. If the private chain overtakes the public one, the attacker publishes it and the merchant's payment disappears. Let q be the share of total hash power held by the attacker and p = 1 − q the share held by honest miners. Following Nakamoto [1], the number of blocks the attacker finds while honest miners find z blocks is approximated by a Poisson distribution with mean λ, and an attacker who is m blocks behind catches up with probability $(q/p)^m$ when q < p. Combining the two gives

$$\lambda = z \cdot q/p \qquad (1)$$

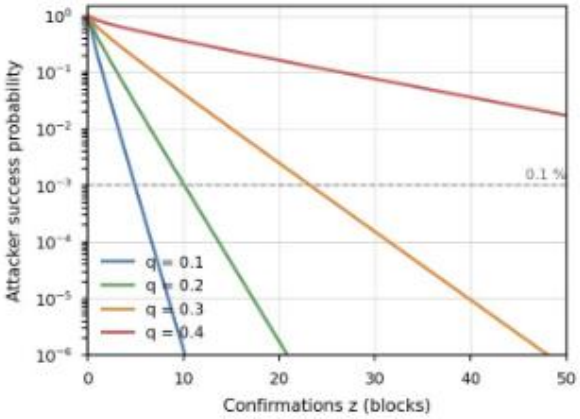
$$P(z) = 1 - \Sigma_{k=0\ldots z}\, \mathrm{Pois}(k) \cdot [1 - (q/p)^{z-k}] \qquad (2)$$

Here $\mathrm{Pois}(k) = e^{-\lambda}\lambda^k/k!$ is the Poisson probability that the attacker finds k blocks. If q ≥ p the attacker eventually succeeds with certainty, so P = 1. Rosenfeld [14] showed that the exact number of attacker blocks follows a negative binomial rather than a Poisson distribution, which makes Nakamoto's figures a close approximation rather than an exact result. Table IV gives values computed from (2) for q = 0.1 and q = 0.3, and Figure 4 plots them for four attacker sizes.

Table 4 : ATTACKER SUCCESS PROBABILITY P(Z)

z	P, q = 0.1	z	P, q = 0.3
0	1.0000000	0	1.0000000
1	0.2045873	5	0.1773523
2	0.0509779	10	0.0416605
3	0.0131722	15	0.0101008
4	0.0034552	20	0.0024804
5	0.0009137	25	0.0006132
6	0.0002428	30	0.0001522
7	0.0000647	35	0.0000379
8	0.0000173	40	0.0000095
9	0.0000046	45	0.0000024
10	0.0000012	50	0.0000006

Computed from Eq. (2) for this paper; z is the number of confirmations.



Graph 1 : Attacker success probability against confirmations (from Eq.(2)).

IJARSCT
International Journal of Advanced Research in Science, Communication and Technology
International Open-Access, Double-Blind, Peer-Reviewed, Refereed, Multidisciplinary Online Journal
ISSN: 2581-9429
Volume 6, Issue 7, October 2026
Impact Factor: 8.2

The results show the central idea of PoW: while honest miners hold the majority, the attacker's chance falls exponentially with each confirmation. With q = 0.1 the chance of success is about 20 % after one confirmation, about 1.3 % after three, and below one in a thousand after five. With q = 0.3 the same level is reached only after 24 confirmations, and with q = 0.4 after 89. With q of one half or more, no number of confirmations is enough. The common practice of six confirmations corresponds to about 0.024 % for q = 0.1 but about 13 % for q = 0.3, so a merchant should choose the number of confirmations from the value at stake and from the attacker size it wants to withstand.

G. Cryptographic Protection and Key Handling

Payments are authorised by digital signatures (ECDSA over the secp256k1 curve in Bitcoin, with Schnorr signatures added in later versions) [12]. In the common address types, addresses are derived from public keys by hashing, so a public key is revealed only when coins are spent. Whoever holds the private key controls the coins, and a lost key cannot be recovered because no authority exists to reset it. Keys should be generated with a trustworthy source of randomness, backed up offline, and kept in hardware wallets or split among several parties with multisignature schemes. Signing gives authenticity and prevents forged payments; it does not hide amounts or addresses, which are public.

IV. SECURITY AND PRIVACY ANALYSIS

A. Threat Analysis

The adversaries considered are an attacker with a large share of hash power, a selfish mining pool, a double-spender who targets a merchant, a network attacker who isolates a node, a miner who submits invalid blocks or false timestamps, a flooder who spams the network, and a thief who steals keys. Table V pairs each threat with the countermeasure in the design.

Table 5 : THREATS AND COUNTERMEASURES

Threat	Countermeasure
Double spending after few confirmations	Wait for more confirmations as the value rises (Table IV)
Majority hash-power attack	High cost of acquiring hash power; spread of mining power; alerts on deep reorganisations
Selfish mining [8]	Monitoring of pool shares; tie-breaking changes proposed in [8]
Zero-confirmation acceptance [9]	Wait for at least one confirmation on valuable goods; watch for conflicting transactions
Eclipse attack [11]	Diverse, randomly chosen peers; countermeasures proposed in [11]
Invalid blocks or false timestamps	Every full node re-validates; timestamp rules bound the error
Spam and flooding	Transaction fees; relay policy; independent validation by each node
Key theft or loss	Hardware wallets, multisignature, protected offline backups

Some risks remain. PoW guarantees the order and immutability of recorded payments, not that a deal made outside the chain was honest. If one party ever controls a majority of hash power, it can rewrite recent history, so the security is economic rather than absolute and depends on the cost of acquiring and running such power. Mining concentrated in a few pools weakens the assumption of many independent miners. Selfish mining shows that the honest-majority condition can be too optimistic under some network conditions [8]. Finally, because the block subsidy halves over time, the long-term security budget will depend more on fees, and how that will work out is not settled.

IJARSCT
International Journal of Advanced Research in Science, Communication and Technology
International Open-Access, Double-Blind, Peer-Reviewed, Refereed, Multidisciplinary Online Journal
ISSN: 2581-9429
Volume 6, Issue 7, October 2026
Impact Factor: 8.2

B. Privacy, Legal, and Environmental Considerations

Bitcoin is pseudonymous, not anonymous: payments are tied to addresses rather than names, but every transaction is public, and analysis of the transaction graph can link addresses together [5]. Using a fresh address for each payment and avoiding the mixing of coins from different sources reduce this linkage, and network-level privacy tools reduce the exposure of a user's IP address. The legal treatment of Bitcoin, including taxation and anti-money-laundering duties for exchanges, differs between countries and changes often, so legal review is advised before any deployment. PoW also converts electricity into security, and de Vries drew attention to the scale of this consumption [15].

V. PROPOSED EVALUATION PLAN

The analysis above can be tested with three small studies. First, a Monte Carlo simulation of the double-spend race can be run for several values of q and z and compared with Eq. (2) and with the refined calculation of Rosenfeld [14]. Second, a private test network of several Bitcoin Core nodes in regression-test mode can be used to measure block propagation delay and the rate of stale blocks while the block interval and block size are varied, to study the effects discussed in [10] and [7]. Third, a simulation of the selfish-mining strategy of [8] can show how the revenue of a pool changes with its share of hash power and with network propagation. The metrics are attack success rate, stale-block rate, confirmation latency, and the number of blocks needed to reach a chosen risk level. The analytical results of Section III-F are the baseline for the first study, and a trusted-third-party ledger is the baseline for latency and throughput.

The expectation is that simulated frequencies agree with Eq. (2) within sampling error, and that shorter block intervals or larger blocks raise the stale-block rate, as discussed in [7] and [10]. These expectations are hypotheses and must be confirmed by measurement.

VI. DISCUSSION

A. Comparison with Alternatives

Table VI contrasts a conventional trusted-third-party system with a PoW network. PoW removes the single point of trust and failure, but it replaces immediate finality with probabilistic finality and pays for security in electricity and throughput. Proof-of-stake, an early form of which was proposed by King and Nadal [16], uses locked-up coins instead of computation as the scarce resource, and Byzantine fault tolerant protocols such as PBFT [17] reach immediate agreement among a known and limited group of participants. Each rests on different assumptions and is outside the scope of this paper.

Table 6 : TRUSTED THIRD PARTY VERSUS POW NETWORK

Aspect	Trusted third party	PoW network
Double-spend check	Issuer checks each payment	Network orders payments; most-work chain wins
Trust needed	Honest, available issuer	Honest majority of hash power
Failure point	Issuer is a single point	Replicated across many nodes
Finality	Immediate once approved	Probabilistic; grows with confirmations
Throughput	High	Limited by block size and interval
Running cost	Low	High; mining uses electricity
Reversal	Possible through the issuer	Practically none after deep confirmation

IJARSCT
International Journal of Advanced Research in Science, Communication and Technology
International Open-Access, Double-Blind, Peer-Reviewed, Refereed, Multidisciplinary Online Journal
ISSN: 2581-9429
Volume 6, Issue 7, October 2026
Impact Factor: 8.2

B. Limitations and Mitigations

Table VII lists the main limitations of the approach and practical steps that reduce each one.

Table 7 : LIMITATIONS AND MITIGATIONS

Limitation	Mitigation
Probabilistic finality	Choose confirmations by value and attacker model (Table IV)
Low throughput [18]	Keep small payments off-chain in channels; batch transactions
High energy use [15]	Efficiency gains; cleaner energy sources; study of alternative consensus
Concentration of mining	Encourage spread across pools; monitor pool shares
Irreversible mistakes	Address checks; escrow or multisignature for large payments
Pseudonymity only	Fresh address per payment; privacy-aware wallets
Rule or software defects	Code review, testing, several client implementations, timely patches

C. Practical Deployment

A merchant or institution that accepts Bitcoin can apply the analysis in steps. First, run its own full node so that it checks the rules itself instead of trusting a third party. Second, choose a confirmation count from the value of each payment and the attacker size it wants to resist, using Table IV, and accept small payments with fewer confirmations than large ones. Third, watch for chain reorganisations and for unusual concentration of hash power. Fourth, protect keys with hardware wallets and multisignature. These steps should work alongside, not replace, normal controls such as network protection, accounting, and staff training.

D. Future Scope

The study can be extended in several directions: a fuller treatment of the exact double-spend probability of [14] that includes network delay; empirical measurement of reorganisation depth on a live network; comparison with proof-of-stake and hybrid designs; payment channels and other second-layer techniques that move small payments off the main chain; and signature schemes that resist quantum computers. A usability study of how non-experts choose confirmation counts would also be valuable.

VII. CONCLUSION

An electronic cash system needs a way to stop double spending without a trusted party. This paper explained how proof-of-work provides one: signed transactions, hash-linked blocks, a costly puzzle, an adaptive difficulty, and the most-work chain rule together make rewriting history expensive, and the attacker's chance of reversing a payment falls exponentially with each confirmation while honest miners hold the majority. Four figures and seven tables set out the layers, the block structure, the mining flow, the double-spend probabilities, the threats, and the limitations.

The approach has costs. Finality is probabilistic, energy use is high, throughput is limited, and security rests on an honest majority of hash power that selfish mining and pool concentration may weaken. How serious these costs are in practice must be shown by independent measurement. The main strengths of PoW are that it needs no central authority, that every node can verify the rules, and that its security can be quantified and tuned by choosing the number of confirmations.

ACKNOWLEDGMENT

The authors thank their institution for academic guidance and resources, and acknowledge the researchers whose published work formed the basis of the literature reviewed here.

IJARSCT
International Journal of Advanced Research in Science, Communication and Technology
International Open-Access, Double-Blind, Peer-Reviewed, Refereed, Multidisciplinary Online Journal
ISSN: 2581-9429
Volume 6, Issue 7, October 2026
Impact Factor: 8.2

REFERENCES

[1] S. Nakamoto, "Bitcoin: A Peer-to-Peer Electronic Cash System," 2008.
[2] C. Dwork and M. Naor, "Pricing via Processing or Combatting Junk Mail," in Advances in Cryptology – CRYPTO '92, LNCS 740, Springer, 1993, pp. 139–147.
[3] A. Back, "Hashcash – A Denial of Service Counter-Measure," technical report, 2002.
[4] S. Haber and W. S. Stornetta, "How to Time-Stamp a Digital Document," J. Cryptology, vol. 3, no. 2, pp. 99–111, 1991.
[5] J. Bonneau, A. Miller, J. Clark, A. Narayanan, J. A. Kroll, and E. W. Felten, "SoK: Research Perspectives and Challenges for Bitcoin and Cryptocurrencies," in Proc. IEEE Symp. Security and Privacy (SP), 2015, pp. 104–121.
[6] J. Garay, A. Kiayias, and N. Leonardos, "The Bitcoin Backbone Protocol: Analysis and Applications," in Advances in Cryptology – EUROCRYPT 2015, LNCS 9057, Springer, 2015, pp. 281–310.
[7] A. Gervais, G. O. Karame, K. Wüst, V. Glykantzis, H. Ritzdorf, and S. Capkun, "On the Security and Performance of Proof of Work Blockchains," in Proc. ACM SIGSAC Conf. Computer and Communications Security (CCS), 2016, pp. 3–16.
[8] I. Eyal and E. G. Sirer, "Majority Is Not Enough: Bitcoin Mining Is Vulnerable," in Financial Cryptography and Data Security (FC 2014), LNCS 8437, Springer, 2014, pp. 436–454.
[9] G. O. Karame, E. Androulaki, and S. Capkun, "Double-Spending Fast Payments in Bitcoin," in Proc. ACM Conf. Computer and Communications Security (CCS), 2012, pp. 906–917.
[10] C. Decker and R. Wattenhofer, "Information Propagation in the Bitcoin Network," in Proc. IEEE Int. Conf. Peer-to-Peer Computing (P2P), 2013.
[11] E. Heilman, A. Kendler, A. Zohar, and S. Goldberg, "Eclipse Attacks on Bitcoin's Peer-to-Peer Network," in Proc. 24th USENIX Security Symp., 2015, pp. 129–144.
[12] A. Narayanan, J. Bonneau, E. Felten, A. Miller, and S. Goldfeder, Bitcoin and Cryptocurrency Technologies. Princeton, NJ, USA: Princeton Univ. Press, 2016.
[13] National Institute of Standards and Technology, "Secure Hash Standard (SHS)," FIPS PUB 180-4, 2015.
[14] M. Rosenfeld, "Analysis of Hashrate-Based Double Spending," arXiv:1402.2009, 2014.
[15] A. de Vries, "Bitcoin's Growing Energy Problem," Joule, vol. 2, no. 5, pp. 801–805, 2018.
[16] S. King and S. Nadal, "PPCoin: Peer-to-Peer Crypto-Currency with Proof-of-Stake," 2012.
[17] M. Castro and B. Liskov, "Practical Byzantine Fault Tolerance," in Proc. 3rd USENIX Symp. Operating Systems Design and Implementation (OSDI), 1999, pp. 173–186.
[18] K. Croman et al., "On Scaling Decentralized Blockchains," in Financial Cryptography and Data Security Workshops, Springer, 2016, pp. 106–125.