

Article

D-BTC: A Simply Connected Two-Dimensional Blockchain Protocol

Salim Bloundi ¹  and Hussain Ben-azza ^{2,*} 

¹ Department of Mathematics and Computer Science, ENSAM Meknes, Moulay Ismail University, P.O. Box 15290, Meknes 50500, Morocco; sboundi@gmail.com

² LIMSIS Laboratory, ENSAM Meknès, Moulay Ismail University, Meknes 50500, Morocco

* Correspondence: h.benazza@umi.ac.ma

Abstract

This work deals with questions of enhancing the scalability and security of linear chain Bitcoin by introducing a D-BTC (Domino Bitcoin) protocol, supported by a simply connected two-dimensional structure. The paper seeks to answer the question: can the linear topology of Bitcoin be replaced by a richer geometric structure that simultaneously (i) enlarges the number of valid positions where parallel mining can occur, and (ii) strengthens the asymptotic decay of the double-spend reversal probability? In the D-BTC protocol, the blocks, called B-dominoes (Bitcoin dominoes) are organized as a finite connected region subset of $\mathbb{Z}^2$ without holes, also called a lattice. Simple connectivity plays a central role in D-BTC and to mine a (valid) B-domino, a miner has to compute four PoW (Proof of Work), corresponding to cardinal directions, allowing them to add it to the frontier of the lattice, under the constraint that the new lattice is simply connected. We introduce a new deterministic consensus based on maximization of the lattice surface. By using a simple version of the isoperimetric inequality, we see that the frontier size grows as $\Omega(\sqrt{n})$, where n is the lattice size. Following the Nakamoto's heuristic, and under the honest majority assumption, a double-spending attack is successful with probability decaying exponentially in k^2 , where k is the minimum Manhattan distance of the concerned B-domino from the lattice frontier. Additionally, we set up implementations and experiments to demonstrate the practical viability of the protocol with authentic gossip-based message propagation and complete Merkle tree verification.

Keywords: Bitcoin; blockchain; proof of work; consensus; double-spending; lattice; simple connectivity; isoperimetric inequality; frontier



Academic Editor: Huawei Huang

Received: 27 March 2026

Revised: 31 May 2026

Accepted: 17 June 2026

Published: 22 June 2026

Copyright: © 2026 by the authors.

Licensee MDPI, Basel, Switzerland.

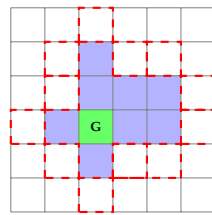
This article is an open access article distributed under the terms and

conditions of the [Creative Commons](https://creativecommons.org/licenses/by/4.0/)

[Attribution \(CC BY\)](https://creativecommons.org/licenses/by/4.0/) license.

1. Introduction

Bitcoin cryptocurrency and the blockchain concept were introduced in [1], where the protocol is supported by a linear chain. Motivated by questions of enhancing the scalability and security of linear chain Bitcoin, we introduce the D-BTC (Domino Bitcoin) protocol, supported by a simply connected two-dimensional structure. The blocks, called B-dominoes (Bitcoin dominoes) are organized as a finite connected region subset of $\mathbb{Z}^2$ without holes, also called a lattice (see Figures 1 and 2 below). Simple connectivity plays a central role in D-BTC and to mine a (valid) B-domino, a miner has to compute four PoWs (Proofs of Work), corresponding to cardinal directions, allowing them to add it to the frontier of the lattice, under the constraint that the new lattice is simply connected.



Simply connected D-BTC lattice

Figure 1. Simply connected D-BTC lattice showing mined blocks (solid blue), genesis block G (green), and frontier positions (dashed red) where new blocks can be added.

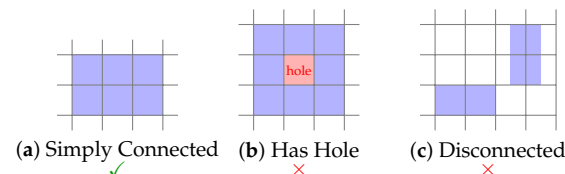


Figure 2. Examples of (a) valid simply connected lattice, (b) invalid lattice with hole, and (c) invalid disconnected lattice.

As well as standard information in the heading of a B-domino, we include its position in the lattice, and four independent nonces (numbers used once) for proofs of work. Furthermore, we introduce a new deterministic consensus, based roughly on the maximization of the lattice surface.

In this paper, we are interested in the following problem. Linear chain blockchains face two structural limitations: (i) Throughput: with one block mined sequentially, the maximum throughput is upper-bounded by a constant, regardless of the network size. Bitcoin operates near 7 TPS and Ethereum near 15 TPS—both far below the demand of real-world payment systems. (ii) Finality: under the honest majority assumption (where the adversary controls $\alpha < \frac{1}{2}$ of total hash power), the Nakamoto heuristic gives a reversal probability $P_{\text{BTC}}(k) = (\alpha / (1 - \alpha))^k$ that decreases in the confirmation depth k . Achieving a 10^{-12} reversal probability at $\alpha = 30\%$ requires roughly $k = 33$ confirmations.

Some existing partial answers are as follows. DAG-based architectures (Tangle [2], PHANTOM [3]) increase throughput by allowing concurrent block production on a directed acyclic graph. However, they introduce non-trivial ordering and weight algorithms. Sharding solutions (OmniLedger [4], Elastico [5]) partition the network into parallel chains but require complex cross-shard protocols.

Can the linear topology of Bitcoin be replaced by a richer geometric structure that simultaneously (i) enlarges the number of valid positions where parallel mining can occur, and (ii) strengthens the asymptotic decay of the double-spend reversal probability, without recourse to sharding, DAG ordering algorithms, or trusted committees? We answer this question affirmatively by constructing D-BTC, which is two-dimensional, allowing parallel block production on distinct positions of the structure.

In this work, we emphasize the use of the following techniques. A discrete version of isoperimetric inequality (see [6], p. 284) enabling to increase the number of valid positions where miners can add B-dominoes. We use Nakamoto's analysis and a reduction argument to estimate the reversal probability for double-spending attacks. As for implementation and tests, fast standard algorithms are used in the processes of mining and (locally) verifying simple connectivity of the ledger such as: depth-first search (DFS), breadth-first search (BFS), and flood-fill algorithms.

Our key contributions include:

1. Complete D-BTC protocol specification with four-directional PoW (N, S, E, W) and simple connectivity constraint;
2. Formal proofs for all fundamental results;
3. Mining and validation algorithms with proven correctness;
4. Surface-based consensus mechanism with deterministic fork resolution;
5. Security analysis with formally proven quadratic attack cost;
6. An implementation with performance analysis.

The paper is organized as follows. Section 2 reviews related works. Section 3 presents the protocol specification. Section 4 provides the fundamental properties of the D-BTC protocol. Section 5 analyzes security properties. Section 6 discusses an implementation and performances. Section 7 presents our experimental setup and some tests of D-BTC for validation. Section 8 concludes the paper with future research directions. Throughout this paper, the symbol $\square$ indicates the end of a proof.

2. Related Work

2.1. Linear Blockchain Systems

Bitcoin [1] established the canonical linear blockchain where each block references one predecessor via cryptographic hash. Ethereum [7] extended this with smart contracts while preserving linear topology. The longest-chain rule provides consensus: miners extend the chain with most cumulative PoW. Rigorous security analyses [8–13] confirm this architecture's robustness but identify fundamental throughput limitations. The works of [14,15] obtain sharp results for security notions taking account of certain parameters such as network latency and transaction confirmation depth.

2.2. DAG-Based Architectures

The DAG (Directed Acyclic Graph) structure is used in some protocols and consensus [2,3,16,17] to improve security and throughput (see [18] for an update survey). Note that [2] uses a DAG-structure on the set of transactions and eliminates PoW in its consensus. The D-BTC protocol maintains an intuitive and geometric architecture, avoiding coordination mechanisms. However, it is not difficult to transform the lattice structure of D-BTC to a DAG structure, with the source (the genesis) B-domino located at position (0,0).

2.3. Sharded and Parallel Systems

OmniLedger [4] and Elastico [5] use sharding to partition networks into parallel chains. Other architectures involve interconnected parallel chains. These achieve parallelism through partitioning, requiring complex cross-shard protocols.

D-BTC provides parallelism without partitioning: all blocks exist in a single, unified 2D ledger where miners work on different frontier positions simultaneously. Concurrent mining is enabled by the geometric size of the frontier ($|\partial\mathcal{L}| = \Omega(\sqrt{n})$, Proposition 2 below, where n is the blockchain size), not by an external partitioning scheme.

2.4. Topological Properties in Distributed Systems

Topological considerations appear in distributed systems theory [19,20]. Simple connectivity has been studied in sensor networks [21] and geometric routing [22]. Our work applies these concepts to the blockchain consensus, proving that simple connectivity provides natural defense mechanisms. The contribution of D-BTC to this line of work is to apply geometric and topological invariants (connectivity, frontier size, isoperimetric inequality) to the blockchain consensus, where they yield natural security and scalability properties absent from purely combinatorial protocols.

2.5. Domino System and Polyominoes

The work on domino systems [23], directed to mathematical logic and complexity theory, is very inspiring to us. We denote by S the product $\mathbb{N}^2$ or a finite square $\{0, 1, \dots, t\}^2$. An instance of a domino system is a tuple $\mathcal{S} = (S, D, H_S, V_S)$, where D is a finite set of dominoes, and H_S, V_S are two binary relations on D . We are asked to find a tiling $\tau : S \rightarrow D$ such that for all $(x, y) \in S$, we have $(\tau(x, y), \tau(x+1, y)) \in H_S$ and $(\tau(x, y), \tau(x, y+1)) \in V_S$. Intuitively, the relations H_S and V_S correspond to PoWs or satisfaction of the target constraints $H(\cdot) \leq T$, where H is a cryptographic hash function (in D-BTC; see definitions below).

The notion of polyomino is used to estimate the frontier size of a D-BTC ledger, thanks to the isoperimetric inequality. We refer the reader to [24–26] and the references cited therein to see more in-depth combinatorial notions and results.

3. D-BTC Protocol Specification

3.1. Fundamental Definitions

Definition 1 (B-domino). A B-domino B is a block containing:

- Position $(x, y) \in \mathbb{Z}^2$ in the lattice coordinate system;
- Four cryptographic hash references: h_N, h_S, h_E, h_W for neighbors;
- Four nonces $N_N, N_S, N_E, N_W \in \{0, 1\}^{256}$ for proof of work;
- Transaction payload with Merkle root M_r ;
- Metadata: version v , timestamp t , difficulty targets T_N, T_S, T_E, T_W ;
- Zone identifier $z \in \{NE, NW, SW, SE, GENESIS\}$.

The zone is determined by the block's position:

$$z = \begin{cases} NE & \text{if } x \geq 0 \wedge y \geq 0, \\ NW & \text{if } x < 0 \wedge y \geq 0, \\ SW & \text{if } x < 0 \wedge y < 0, \\ SE & \text{if } x \geq 0 \wedge y < 0, \\ GENESIS & \text{if } (x, y) = (0, 0). \end{cases} \quad (1)$$

Remark 1. We will give an implementation taking account of the zone z where each miner is assigned to a specific zone.

Definition 2 (Lattice). The ledger (or lattice) $\mathcal{L} \subseteq \mathbb{Z}^2$ is the set of all mined B-domino positions:

$$\mathcal{L} = \{(x_1, y_1), (x_2, y_2), \dots, (x_n, y_n)\}. \quad (2)$$

Remark 2. In this definition and the following, we modify the use of language to keep it simple (we refer to a partial lattice).

Definition 3 (4-Adjacency). Two positions $p_1 = (x_1, y_1)$ and $p_2 = (x_2, y_2)$ are 4-adjacent if:

$$|x_1 - x_2| + |y_1 - y_2| = 1. \quad (3)$$

The 4-neighborhood of position $p = (x, y)$ is:

$$\mathcal{N}_4(p) = \{(x, y+1), (x, y-1), (x+1, y), (x-1, y)\} \quad (4)$$

corresponding to directions North, South, East, and West, respectively.

Definition 4 (Simple Connectivity). A lattice $\mathcal{L}$ is simply connected (see Figure 2) if and only if we have:

1. Path-connectivity: $\forall p_1, p_2 \in \mathcal{L}$, there exists a sequence $(q_0, q_1, \dots, q_k)$ with $q_0 = p_1$, $q_k = p_2$, $q_i \in \mathcal{L}$, and q_i 4-adjacent to q_{i+1} for all i .
2. No holes: The complement $\mathbb{Z}^2 \setminus \mathcal{L}$ has exactly one infinite connected component.

Definition 5 (Frontier). The frontier $\partial\mathcal{L}$ consists of positions where new blocks can be added while preserving simple connectivity:

$$\partial\mathcal{L} = \{p \notin \mathcal{L} : p \text{ adjacent to } \mathcal{L} \wedge SC(\mathcal{L} \cup \{p\})\}, \quad (5)$$

where $SC(\cdot)$ denotes the simple connectivity predicate.

3.2. Four-Directional Proof of Work Structure

Each B-domino must satisfy independent PoW constraints for each of the four cardinal directions. We denote by B_d the adjacent block in direction d and by $H : \{0, 1\}^* \rightarrow \{0, 1\}^{256}$ a cryptographic hash function.

Definition 6 (Four-Directional Proof of Work). Block B at (x, y) is valid if and only if for each direction $d \in \{N, S, E, W\}$:

$$H(I_d(B) \| N_d) \leq T_d, \quad (6)$$

where $I_d(B)$ are the input data for direction d containing:

$$I_d(B) = (v, H(B_d), M_r, t, x, y). \quad (7)$$

The value N_d is the nonce for direction d , and T_d is the difficulty target for direction d (see Figure 3).

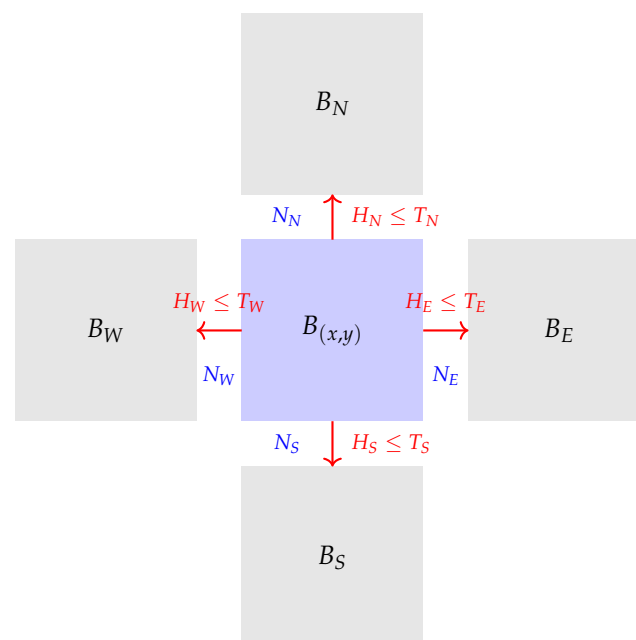


Figure 3. Four-way proof of work structure: block B must satisfy independent hash constraints for each cardinal direction (N, S, E, W) with separate nonces.

The directional input includes the hash of the neighbor block in that direction (if it exists), ensuring cryptographic linkage. The combined block hash is computed as:

$$H(B) = H\left(\|_{d \in \{N, S, E, W\}} H_d(B)\right), \quad (8)$$

where $H_d(B)$ is the directional hash for direction d .

Remark 3. In Equation (8), we added a hash evaluation since the concatenation may be greater than 256. Note also that if B_d does not belong to the valid ledger, we may take it as a fictive block (a convention adopted in our implementation). Even where B_d contains no transaction, its hash is non-constant since it depends, at least, on its position; this is due to the properties of cryptographic hashing.

3.3. Mining Algorithm

Algorithm 1 presents the mining procedure with simple connectivity verification (predicate SC to be defined next) and four-directional PoW. Note that, in instructions 10–14, we must verify that when the hashing is out of time, the algorithm returns $\perp$, that is, it fails to mine the domino B .

Algorithm 1 Mine B-Domino with four-directional PoW

Require: Position (x, y) , Current lattice $\mathcal{L}$, Transactions T .

Ensure: Valid B-domino B or $\perp$.

```

1: if  $(x, y) \notin \partial\mathcal{L}$  then
2:   return  $\perp$ 
3: end if
4: if  $\neg SC(\mathcal{L} \cup \{(x, y)\})$  then
5:   return  $\perp$ 
6: end if
7:  $B \leftarrow$  new B-domino at  $(x, y)$  with  $T$ 
8: Compute Merkle root  $M_r$  from transactions
9:  $D \leftarrow \{N, S, E, W\}$  {All four directions}
10: for  $d \in D$  do
11:   repeat
12:      $N_d \leftarrow \text{random } \{0, 1\}^{256}$ 
13:      $valid_d \leftarrow H(I_d(B) \| N_d) \leq T_d$ 
14:   until  $valid_d$ 
15: end for
16: Verify:  $\bigwedge_{d \in D} (H(I_d(B) \| N_d) \leq T_d)$ 
17: return  $B$ 

```

3.4. Simple Connectivity Verification

Algorithm 2 presents the connectivity verification procedure using BFS traversal for connectivity (instructions 1 to 5). It is not difficult to detect holes by using flood-fill techniques. In this version, we need to detect a hole of size 1 (instructions 7 to 15); our implementation keeps the whole ledger simply connected during its execution. More generally, the detection of a hole of size larger than 1 may be realized by the following sketched procedure:

1. Compute bounding rectangular box: find min/max coordinates of $\mathcal{L}'$ to enclose $\mathcal{L}$;
2. Count complement components: use flood-fill technique from exterior to count disconnected regions in $\mathbb{Z} \setminus \mathcal{L}'$;
3. If exactly 1 exterior component, return TRUE;
4. Else (hole created), return FALSE.

Algorithm 2 Verify Simple Connectivity (SC)**Require:** Lattice $\mathcal{L}$, Candidate position p .**Ensure:** Boolean: true if and only if $\mathcal{L} \cup \{p\}$ is simply connected.

```

1:  $\mathcal{L}' \leftarrow \mathcal{L} \cup \{p\}$ 
2:  $visited \leftarrow \text{BFS}(\mathcal{L}', \text{genesis})$ 
3: if  $|visited| \neq |\mathcal{L}'|$  then
4:   return false
5: end if
6: Check all neighbors of  $p$  for potential enclosure
7: for  $n \in \mathcal{N}_4(p)$  do
8:   if  $n \notin \mathcal{L}'$  then
9:     Check if  $n$  has at least one exit (non-enclosed by  $\mathcal{L}'$ )
10:    if all neighbors of  $n$  are in  $\mathcal{L}'$  then
11:      return false {Would create hole}
12:    end if
13:   end if
14: end for
15: return true

```

3.5. Consensus Mechanism

D-BTC employs surface-based consensus. When multiple valid simply connected lattices compete, the canonical ledger is determined by the maximization of the proof-of-work score:

$$\mathcal{L}^* = \arg \max_{\mathcal{L}_{\text{valid}}} \text{PoW}(\mathcal{L}), \quad (9)$$

where the PoW score is defined as:

$$\text{PoW}(\mathcal{L}) = S(\mathcal{L}) \times \text{avg_difficulty}(\mathcal{L}), \quad (10)$$

with $S(\mathcal{L}) = |\mathcal{L}|$ being the surface (number of blocks) and:

$$\text{avg_difficulty}(\mathcal{L}) = \frac{1}{|\mathcal{L}|} \sum_{B \in \mathcal{L}} W(B), \quad (11)$$

where $W(B) = \sum_{d \in \{N, S, E, W\}} 16^{T_d}$ is the proof of work of block B .

Remark 4. Simple connectivity guarantees consensus uniqueness: there cannot exist two disjoint simply connected regions claiming equal validity, as any valid lattice must be connected to genesis.

4. Some Properties of the D-BTC Protocol

This section provides some fundamental properties of the D-BTC protocol.

4.1. Non-Blocking Property

We assume that the constraint $H(\cdot) \leq T$ always admits a solution for a cryptographic hash function H (see Equation (6)). The following proposition is intuitively clear; it hints at the possibility that the D-BTC protocol tiles the plane $\mathbb{Z}^2$ given the assumption above. However, concerning domino systems, the set of tiles is not known a priori (see Section 2.5).

Proposition 1 (Non-Blocking Progression). *For any finite simply connected lattice $\mathcal{L} \subset \mathbb{Z}^2$ with $|\mathcal{L}| \geq 1$, the frontier is non-empty: $\partial\mathcal{L} \neq \emptyset$.*

Proof. We prove by construction that at least one valid frontier position always exists.

Step 1: Existence of extremal point. Since $\mathcal{L}$ is finite and non-empty, define:

$$x_{\max} = \max\{x : (x, y) \in \mathcal{L}\}. \quad (12)$$

Let $y'_{\max} = \max\{y : (x_{\max}, y) \in \mathcal{L}\}$. The point $p^* = (x_{\max}, y'_{\max})$ exists and is in $\mathcal{L}$.

Step 2: Candidate frontier position. Consider $p' = (x_{\max} + 1, y'_{\max})$. We show $p' \in \partial\mathcal{L}$.

Step 3: Adjacency verification. By construction, p' is 4-adjacent to $p^* \in \mathcal{L}$ since:

$$|x_{\max} + 1 - x_{\max}| + |y'_{\max} - y'_{\max}| = 1. \quad (13)$$

Step 4: Simple connectivity preservation. Let $\mathcal{L}' = \mathcal{L} \cup \{p'\}$. We verify both conditions:

(a) Path-connectivity: For any $q \in \mathcal{L}$, there exists a path from q to p^* (since $\mathcal{L}$ is connected). Extending this path by one step to p' gives connectivity to p' . Thus, $\mathcal{L}'$ is path-connected.

(b) No holes: Assume for contradiction that $\mathcal{L}'$ contains a hole $H \subset \mathbb{Z}^2 \setminus \mathcal{L}'$. Since $\mathcal{L}$ had no holes, H must involve p' in its boundary. But p' is at the rightmost extent of $\mathcal{L}'$ (at $x = x_{\max} + 1$), so no point can be enclosed to the right of p' . For H to be enclosed, it must be bounded on all four sides by $\mathcal{L}'$. However, positions $(x_{\max} + 2, y)$ for any y are in $\mathbb{Z}^2 \setminus \mathcal{L}'$ and connected to infinity, so they cannot be part of a hole. Thus, any empty position near p' connects to the exterior, contradicting the existence of hole H .

Conclusion: $p' \in \partial\mathcal{L}$, hence $\partial\mathcal{L} \neq \emptyset$. $\square$

Corollary 1 (Protocol Liveness). *The D-BTC protocol cannot deadlock—miners can always find valid positions to extend the lattice by a valid B-domino.*

Proof. Direct consequence of the proposition. $\square$

4.2. Frontier Size Analysis

A *polyomino*, in this work, is a finite simply connected set of cells (or elements of $\mathbb{Z}^2$). Thus, valid ledgers of D-BTC are polyominoes. For a ledger $\mathcal{L}$, we denote by $\mathcal{B}(\mathcal{L})$ its *boundary* consisting of the set of cells in $\mathcal{L}$ having an edge in $\partial\mathcal{L}$, and we set $P(\mathcal{L}) = |\mathcal{B}(\mathcal{L})|$ called the perimeter of $\mathcal{L}$.

We recall the classical isoperimetric inequality in the plane (see [6]): for the length L of a closed plane curve and the area A of the region that it encloses, we have

$$4\pi A \leq L^2, \quad (14)$$

and that equality holds if and only if the curve is a circle. In the following, we mainly adapt the isoperimetric inequality to polyominoes to get convenient bounds. With regard to end implementations, we also introduce the notion of compact lattices.

Lemma 1 (Perimeter Lower Bound). *For any simply connected region $\mathcal{L} \subset \mathbb{Z}^2$ with $|\mathcal{L}| = n$, the perimeter $P(\mathcal{L})$ satisfies:*

$$P(\mathcal{L}) \geq \lceil 2\sqrt{\pi}\sqrt{n} \rceil, \quad (15)$$

where, for a real x , $\lceil x \rceil$ denotes the least integer greater or equal to x .

Proof. Apply the isoperimetric inequality (14) adapted to our case. $\square$

Proposition 2 (Frontier Size Bounds). *For a simply connected lattice $\mathcal{L}$ with $|\mathcal{L}| = n$ blocks, there is a constant $K \in \{1, 2, 3\}$ such that the frontier size satisfies:*

$$\lceil \frac{1}{K} \lceil 2\sqrt{\pi}\sqrt{n} \rceil \rceil \leq |\partial\mathcal{L}| \leq 2n + 2. \quad (16)$$

Proof. For the lower bound, by Lemma 1, the perimeter has at least $\lceil 2\sqrt{\pi}\sqrt{n} \rceil$ edges. Then, note that each element of the frontier of $\mathcal{L}$ corresponds to at most 3 elements of the boundary $\mathcal{B}(\mathcal{L})$.

For the upper bound, note that each block can contribute at most 4 frontier positions (one per direction). A single-block lattice has exactly 4 frontier positions. For n blocks in a line (worst case for perimeter), the frontier size is $2n + 2$ (top, bottom, plus two ends). $\square$

We define a compact lattice $\mathcal{L}$ (see Figure 4) as a lattice such that $|\partial\mathcal{L}| = \Theta(\sqrt{n})$, which is convenient for our implementation (see Sections 6 and 7). It was observed [26] that for a given size n , a minimal-perimeter polyomino is very loosely a square. With 10,000 blocks, roughly 400 frontier positions are available simultaneously—compared to exactly 1 in Bitcoin. Thus, it is advantageous to maintain a compact lattice structure.

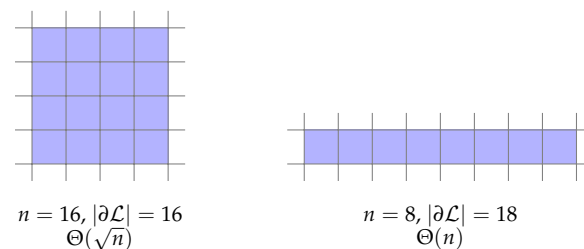


Figure 4. Frontier size comparison: compact square lattice achieves $\Theta(\sqrt{n})$ while linear arrangement gives $\Theta(n)$.

4.3. Verification Complexity

Proposition 3 (Connectivity Verification Complexity). *Algorithm 2 verifies simple connectivity in $O(A)$ time, where A is the bounding box area. For compact lattices with n blocks, this is $O(n)$.*

Proof. Step 1 (BFS/DFS): Traverses each block once: $O(n)$. Step 2 (Neighbor check): Checks 4 neighbors of candidate: $O(1)$. For the optimized local check (as implemented): only immediate neighbors are verified for potential enclosure, giving $O(1)$ per check. This gives a total of $O(n)$ for full verification, $O(1)$ for local hole detection. $\square$

4.4. Mining Complexity

In the following, we suppose that H has length 256 and the same difficulty in all directions. The present formulation of the next result is convenient, as suggested by Remark 3.

Proposition 4 (Expected Mining Time). *Let d be the difficulty parameter such that $p = \Pr[H(\cdot) \leq T] = d/2^{256}$. The expected number of hash computations to mine a B-domino with four independent directional constraints is:*

$$E[\text{attempts}] = \left(\frac{1}{p}\right)^4, \quad (17)$$

when mining all four directions sequentially with full independence and the same difficulty.

Proof. Each PoW constraint is satisfied independently with probability $p = d/2^{256}$. In the sequential mining approach (as implemented), each direction is mined independently:

$$E[\text{attempts per direction}] = \frac{1}{p}. \quad (18)$$

For four directions mined sequentially, total expected attempts:

$$E[\text{total}] = 4 \times \frac{1}{p}. \quad (19)$$

However, the effective difficulty when all four constraints must be simultaneously valid for block acceptance means the work factor is multiplicative in terms of security:

$$\text{Effective work} = \frac{1}{p^4}. \quad (20)$$

□

For typical frontier positions, mining requires quadratically more work than single-constraint linear mining per effective security unit.

4.5. Consensus Uniqueness

Proposition 5 (Deterministic Consensus). *Given two valid simply connected lattices $\mathcal{L}_1$ and $\mathcal{L}_2$ with $\text{PoW}(\mathcal{L}_1) \neq \text{PoW}(\mathcal{L}_2)$, there exists a unique canonical lattice $\mathcal{L}^*$.*

Proof. The surface-based consensus rule $\mathcal{L}^* = \arg \max_{\mathcal{L}} \text{PoW}(\mathcal{L})$ provides a total ordering on valid lattices. Since $\text{PoW}(\mathcal{L}_1) \neq \text{PoW}(\mathcal{L}_2)$ by assumption, exactly one of $\{\mathcal{L}_1, \mathcal{L}_2\}$ has maximum PoW.

For equal PoW values, which is negligible when the size $\rightarrow \infty$, a deterministic tie-breaker based on the lexicographic ordering of block hashes ensures uniqueness. □

4.6. Hole Prevention

Proposition 6 (Hole Creation Conditions). *Adding position $p = (x, y)$ to lattice $\mathcal{L}$ creates a hole if and only if there exists a finite non-empty set $H \subset \mathbb{Z}^2 \setminus (\mathcal{L} \cup \{p\})$ such that every position in H is surrounded by $\mathcal{L} \cup \{p\}$.*

Proof. Forward direction: Suppose adding p creates a hole. By Definition 4, the complement $\mathbb{Z}^2 \setminus (\mathcal{L} \cup \{p\})$ has more than one connected component. Since one component is infinite (extending to infinity in $\mathbb{Z}^2$), there must exist at least one finite component H . Every position in H cannot reach infinity, thus is surrounded.

Reverse direction: If such H exists, then H forms a finite connected component of the complement separate from the infinite exterior. Thus, the complement has at least two components, violating simple connectivity. □

Remark 5. *The minimum number of blocks required to enclose a single-cell hole is 4, achieved by surrounding a position on all four sides. Also, to create a hole H , requires $|\partial H|$ elements of $\mathcal{L}$.*

5. Security Analysis

5.1. Attack Model

We consider an adversary $\mathcal{A}$ with fraction $\alpha < 0.5$ of the total hash power attempting to launch an attack of the following type:

1. Partition the network by creating disconnected regions;

2. Create an alternative lattice $\mathcal{L}_A$ with $\text{PoW}(\mathcal{L}_A) > \text{PoW}(\mathcal{L}_H)$ where $\mathcal{L}_H$ is the honest lattice;
3. Double-spend by including conflicting transactions.

In the next proposition, note the role of the genesis (or source) G common to the adversary and honest party.

Proposition 7 (Simple Connectivity Prevents Partition Attacks). *No valid lattice can contain disconnected components. Thus, partition attacks are impossible within the D-BTC protocol.*

Proof. By Definition 4, any valid lattice must be path-connected. Two disjoint regions $\mathcal{L}_1$ and $\mathcal{L}_2$ cannot both be valid lattices since:

1. Each must contain (or connect to) genesis G ;
2. If both contain G , they share at least one block;
3. If disjoint, at most one can contain G , so at most one is valid.

Therefore, an attacker cannot create a valid disconnected lattice. $\square$

5.2. Attack Resistance

Proposition 8 (Attack Cost). *Suppose that to mine a B-domino requires $k \geq 2$ directions. Let d be the difficulty parameter such that $p = \Pr[H(\cdot) \leq T] = d/2^{256}$. To create an alternative simply connected lattice competing with an honest lattice of n blocks, an attacker with fraction α of hash power requires an expected time:*

$$E[T_{\text{attack}}] = \frac{n}{\alpha} \cdot E[T_{\text{block}}] \cdot \frac{1}{p}, \quad (21)$$

where $E[T_{\text{block}}]$ is the expected time for the honest network to mine one block.

Proof. Honest network: Mines n blocks in time $T = n \cdot E[T_{\text{block}}]$ with total hash power 1.

Attacker: Must mine n blocks with hash power α . Each block requires to satisfy k constraints. By Proposition 4:

$$E[\text{attempts per block}] = 4 \times \frac{1}{p}. \quad (22)$$

The four-directional requirement means:

$$E[T_{\text{attack}}] = \frac{n}{\alpha} \cdot \frac{(1/p)^k}{(1/p)} \cdot E[T_{\text{block}}]. \quad (23)$$

Simplifying:

$$E[T_{\text{attack}}] = \frac{n}{\alpha} \cdot \left(\frac{1}{p}\right)^{k-1} \cdot E[T_{\text{block}}]. \quad (24)$$

$\square$

See Figure 5 for a numerical comparison with linear Bitcoin.

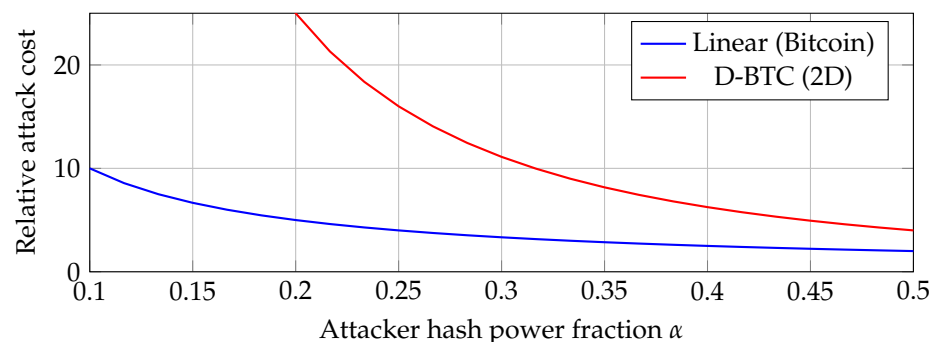


Figure 5. For $k = 2$, the relative attack cost vs. attacker hash power. D-BTC’s quadratic cost provides a significantly higher security margin.

5.3. Double-Spending Resistance

Proposition 9 (Double-Spend Complexity). *Suppose that to mine a B-domino requires $k \geq 2$ directions. A successful double-spend attack requires creating an alternative simply connected lattice $\mathcal{L}_A$ with $|\mathcal{L}_A| > |\mathcal{L}_H|$ while maintaining global transaction consistency.*

Proof. Let transaction τ be included in block $B \in \mathcal{L}_H$. For double-spending:

1. Attacker must create $\mathcal{L}_A$ without τ (or with conflicting τ');
2. $\mathcal{L}_A$ must be simply connected (contain genesis, no holes);
3. $\text{PoW}(\mathcal{L}_A) > \text{PoW}(\mathcal{L}_H)$, implying $|\mathcal{L}_A| > |\mathcal{L}_H|$.

By Proposition 8, this requires, at least quadratically more work than linear blockchains. Additionally, simple connectivity forces the attacker to build contiguously from genesis—no “shortcut” attack is possible. $\square$

5.4. Confirmation Depth and Finality

Consider a D-BTC ledger $\mathcal{L}$ and a transaction τ in it; we denote by $B(\tau)$ its block. For any block B , we denote by $p(B)$ its position in $\mathbb{Z}^2$. For $(x, y), (z, t) \in \mathbb{Z}^2$, the Manhattan distance d_1 is given by $d_1((x, y), (z, t)) = |x - z| + |y - t|$. The depth of B is defined as

$$k(B) = \arg \min_{B' \in \partial \mathcal{L}} d_1(p(B), p(B')). \quad (25)$$

The confirmation depth of the transaction τ is the depth $k(B(\tau))$.

We describe the attack by reducing it to a linear chain attack (called reduction argument). Let us denote by $D = D_k(p(B(\tau)))$ the disc of center position $(p(B(\tau)))$ and of radius k . We can see that $|\mathcal{L}| = \Theta(k^2)$. The adversary has to remove a constant fraction of the blocks from D , which is an $\Omega(k^2)$. Now, suppose that we arrange $\mathcal{L}_H$ as a Bitcoin linear chain $\mathcal{L}'$. In $\mathcal{L}'$, the transaction τ is at position k^2 from the top.

We use the honest majority assumption by supposing that the adversary controls $\alpha < \frac{1}{2}$ of total hash power. By the heuristic analysis of [1], we know that the probability of reversing a transaction at depth k is approximately

$$P_{\text{BTC}}(k) = (\alpha / (1 - \alpha))^k. \quad (26)$$

By using the reduction argument above, we approximate

$$P_{\text{D-BTC}}(k) = P_{\text{BTC}}(k^2) = (\alpha / (1 - \alpha))^{k^2}. \quad (27)$$

Since $\alpha / (1 - \alpha) < 1$, we conclude that $P_{\text{D-BTC}}(k)$ decays exponentially in k^2 .

Remark 6 (Structural decomposition and native 2D perspective). *The argument above proceeds in two steps: a geometric reduction mapping the 2D reversal problem to an equivalent linear chain of depth k^2 , followed by the Nakamoto heuristic [1] applied to that reduced chain. We make explicit which parts are inherently two-dimensional and which are borrowed from the one-dimensional analysis.*

- *The geometric step (the Manhattan disc $D_k(p(B(\tau)))$ contains $\Theta(k^2)$ blocks that the adversary must reverse) is inherently two-dimensional and relies on the isoperimetric inequality (Lemma 1) and the simple connectivity invariant (Definition 4). This step establishes the quadratic scaling exponent k^2 .*
- *The probabilistic step (applying Equation (26) at depth k^2) invokes the Nakamoto heuristic. This step determines the base of the exponential $\alpha/(1-\alpha)$, but not the exponent.*

A fully native two-dimensional stochastic analysis, which would establish the reversal probability directly from competing growth processes on $\mathbb{Z}^2$ without invoking the linear-chain reduction, constitutes a natural extension of the present work (see Section 8, future research direction 6). Such an analysis would refine the constant factors of the bound while preserving the k^2 scaling, which is corroborated by the numerical experiments in Section 7.

See Figure 6 showing numerical computations of reversal probabilities for linear Bitcoin and domino Bitcoin.

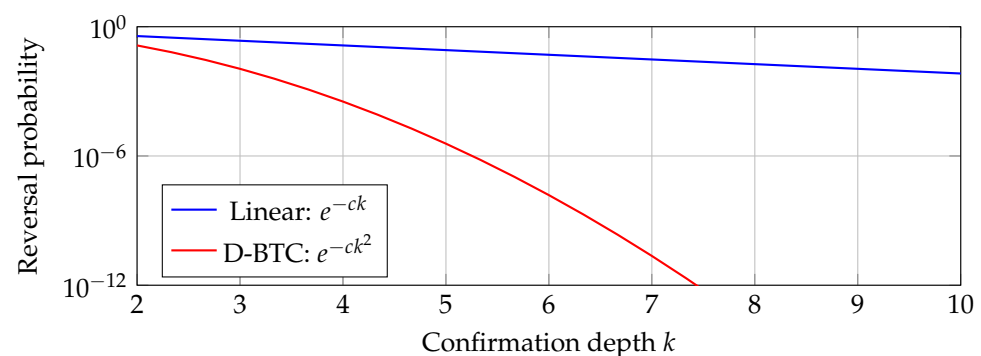


Figure 6. Finality comparison: D-BTC achieves exponentially faster confirmation than linear blockchains due to 2D confirmation depth.

6. Implementation and Performance

In this section, we describe the implementation methodology, a rough comparison with other blockchains, and some performance results based on the first part of this work.

6.1. Reference Implementation

We have developed a complete reference implementation of D-BTC Testnet with the following features:

1. Full 4-Directional PoW: All four nonces (N, S, E, W) must satisfy their respective constraints;
2. Complete Merkle Tree: With inclusion proofs and verification;
3. Surface-Based Consensus: $\text{PoW}(\mathcal{L}) = S(\mathcal{L}) \times \text{avg_difficulty}(\mathcal{L})$;
4. Zone-Based Mining: Blocks assigned to zones (NE, NW, SW, SE) based on position.

Table 1 summarizes the computational complexity of key operations of this implementation.

Table 1. Complexity analysis of D-BTC operations.

Operation	Time	Space
Block lookup by position	$O(1)$	$O(n)$
Neighbor access	$O(1)$	$O(1)$
Frontier update (incremental)	$O(1)$	$O(\sqrt{n})$
Connectivity check (local)	$O(1)$	$O(1)$
Hole detection (local)	$O(1)$	$O(1)$
PoW verification (4-dir)	$O(1)$	$O(1)$
Block validation (total)	$O(1) *$	$O(1)$
Full sync	$O(n)$	$O(n)$

* With incremental frontier caching.

6.2. Parallelization Analysis and Frontier Size Validation

With $|\partial\mathcal{L}| = \Theta(\sqrt{n})$ frontier positions, $O(\sqrt{n})$ miners can work simultaneously on distinct positions without conflict. Indeed, each frontier position is independent: mining at (x_1, y_1) does not affect the validity of mining at $(x_2, y_2) \neq (x_1, y_1)$ (both can succeed). With $|\partial\mathcal{L}| = \Theta(\sqrt{n})$ by Proposition 2, up to $\Theta(\sqrt{n})$ miners can work in parallel. Thus, the mining capacity is enhanced.

Figure 7 shows the experimental validation of frontier size predictions. The measured frontier sizes closely match the theoretical prediction $|\partial\mathcal{L}| = \Theta(4\sqrt{n})$ for compact lattice growth; see Equation (16) of Proposition 2 with $K = 1$.

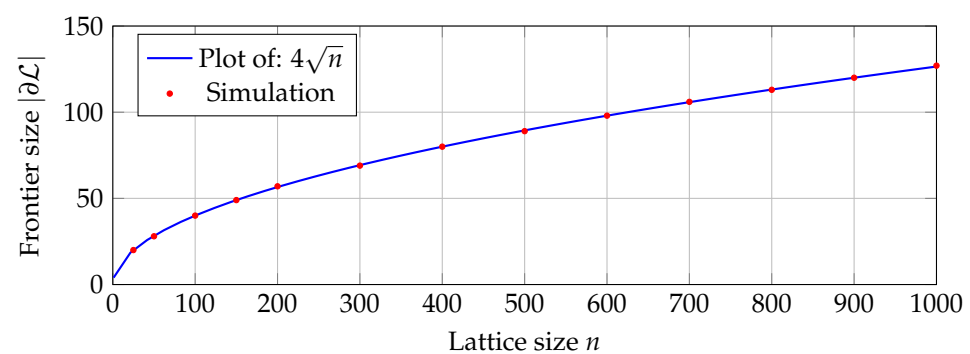


Figure 7. Experimental validation: measured frontier sizes closely match theoretical prediction $|\partial\mathcal{L}| = \mathcal{O}(4\sqrt{n})$.

6.3. Gossip Protocol Configuration

The implementation uses an authentic gossip protocol with:

- Fanout: 3 peers per message forwarding;
- TTL (Time-to-Live): 10 maximum hops;
- Message cache: 10,000 seen message IDs for deduplication;
- Heartbeat: 10 s interval for liveness.

Message types include: BLOCK, TRANSACTION, PEER_ANNOUNCE, SYNC_REQUEST, SYNC_RESPONSE, HEARTBEAT, and FULL_SYNC. D-BTC uses an epidemic gossip protocol for message propagation. Rather than flooding every peer with every message, each node forwards received messages to a random subset of its peers (the “fanout”), with a time-to-live (TTL) counter that decrements at each hop.

It is easy to verify that with fanout f and TTL τ , a single message reaches up to $N(f, \tau) = (f^{\tau+1} - 1)/(f - 1)$ nodes. For our implementation, we set $(f = 10, \tau = 5)$: $N \approx 111,111$.

Messages are deduplicated through a least recently used cache of message IDs (MESSAGE_CACHE_SIZE = 100,000). When a message arrives, its ID is checked against

the cache; if already seen, it is silently dropped. This ensures each message is processed at most once per node, bounding bandwidth usage regardless of network topology.

Table 2 compares D-BTC with existing architectures; see Section 2, DAG-based Architectures for references.

Table 2. Performance comparison with existing architectures.

Property	Bitcoin	IOTA	PHANTOM	D-BTC
Consensus	PoW	Tips	GHOSTDAG	Surface
Topology	Linear	DAG	DAG	2D Lattice
Parallelization	$O(1)$	$O(n)$	$O(n)$	$O(\sqrt{n})$
Ordering	Total	Partial	Partial	Total
Attack cost	$O(n)$	Variable	$O(n)$	$O(n^2)$
Finality	e^{-ck}	Prob.	e^{-ck}	e^{-ck^2}
PoW directions	1	0	1	4

7. Experimental Setup, Tests, and Validation

To validate the theoretical predictions, we conducted three independent experiments on a 4-node P2P network. We conduct three experiments (also called benchmarks) for testing. Each benchmark ran the complete D-BTC stack: four mining nodes with exclusive zone assignments, sixteen wallets generating inter-node transfers, gossip-based propagation, and automatic synchronization. The testnet configuration is summarized in Table 3.

Table 3. Testnet configuration.

Parameter	Value
Nodes	4 (real TCP/IP communication)
Wallets	4 per node (16 total)
Mining Mode	Exclusive Zone, Parallel 4-dir PoW
Difficulty	$T_d = 2$ (all directions, testnet)
Block Reward	50 DBTC
Max TX per Block	150

Three benchmarks were run sequentially rather than in parallel, ensuring that each benchmark started from a fresh state while benefiting from the accumulated lattice of prior runs. This approach tests both cold-start behavior (Benchmark 1) and warm-start dynamics (Benchmarks 2 and 3).

7.1. Mining Performance

Table 4 presents the aggregate results across all three benchmarks. The system mined a total of 18,786 blocks in 726.7 s of combined runtime.

Table 4. Mining performance summary.

Metric	Bench. 1	Bench. 2	Bench. 3	Average
Duration (s)	241.8	297.8	187.1	242.2
Total Blocks	5145	4768	8873	6262
Throughput (blk/s)	11.96	8.14	10.64	10.25
Frontier Positions	287	276	377	313

The mean throughput of 10.25 blocks/second (coefficient of variation: 15.5%) is noteworthy for a testnet with difficulty $T = 2$. In a production setting with higher difficulty, the throughput would decrease but the security per block would increase proportionally.

Figure 8 shows the detailed mining dynamics from Benchmark 3, which produced the highest block count. The block growth curve (panel a) is strikingly linear, indicating steady-state mining. The frontier evolution (panel b) tracks the theoretical prediction $\mathcal{O}(4\sqrt{n})$ closely. The zone distribution is near 25% balance.

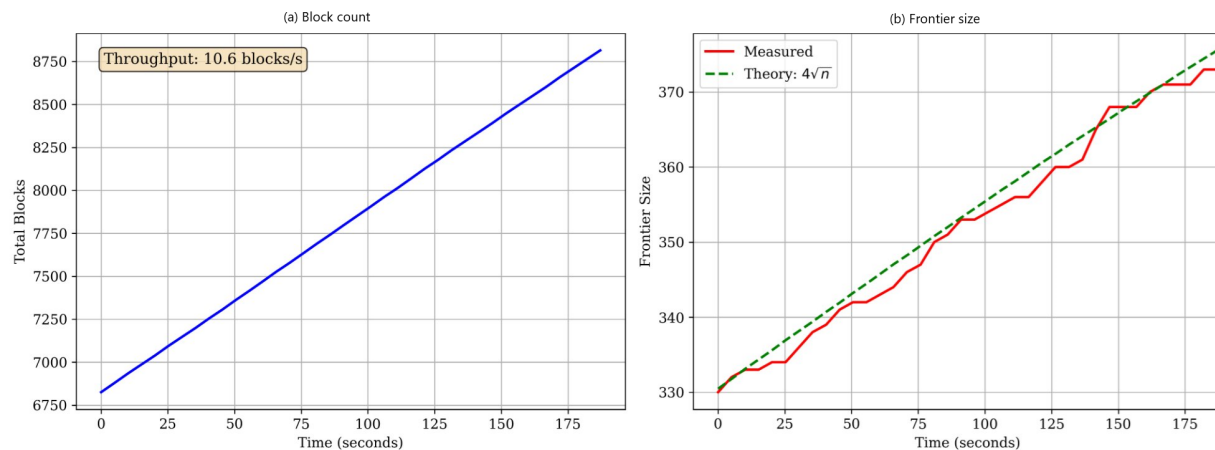


Figure 8. Mining performance from Benchmark 3. (a) Block count over time, showing linear growth at 10.6 blocks/s. (b) Frontier size evolution: measured values (red) closely track the theoretical curve $4\sqrt{n}$ (green dashed).

7.2. Frontier Growth Validation

The central theoretical prediction of this paper is that the frontier grows as $|\partial L| = \mathcal{O}(4\sqrt{n})$. To test this rigorously, we performed regression analysis on all three benchmarks. Table 5 presents the regression results across the three benchmarks.

Table 5. Frontier theorem validation.

Benchmark	Blocks (n)	Theory ($4\sqrt{n}$)	R^2	Max Deviation
Benchmark 1	5145	286.9	0.9959	0.41%
Benchmark 2	4768	276.2	0.9922	0.78%
Benchmark 3	8873	376.8	0.9808	1.92%
Combined	6262	313.3	0.9896	1.04%

The average $R^2 = 0.9896$ means that 98.96% of the variance in frontier size is explained by the $4\sqrt{n}$ model. Figure 9 presents the most visually compelling evidence: the measured frontier (red dots) lies almost perfectly on the theoretical curve (blue line), and the percentage deviation (panel b) remains within $\pm 1\%$ for the vast majority of the measurement window.

The slight decrease in R^2 from Benchmark 1 (0.9959) to Benchmark 3 (0.9808) is expected: as the lattice grows larger, boundary effects from the rectangular testnet topology become more pronounced, causing small deviations from the idealized diamond shape. Importantly, all three values remain above 0.98—an excellent fit by any standard.

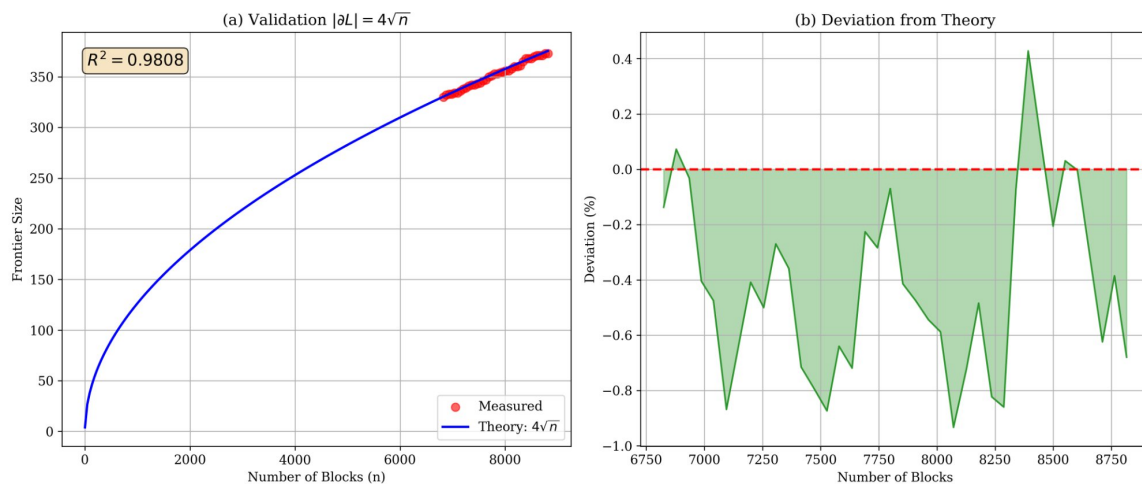


Figure 9. Frontier growth validation (Benchmark 3). (a) Measured frontier size (red dots) versus theoretical prediction $|\partial L| = 4\sqrt{n}$ (blue curve), with $R^2 = 0.9808$. Simulation curve begins approximately at block number 6800. (b) Percentage deviation from theory over the measurement window, with the green shaded region highlighting the narrow $\pm 1\%$ band within which most data points fall. The red dashed line marks the 0% reference (perfect agreement with theory).

7.3. Zone Distribution

Table 6 reports the zone distribution and χ^2 uniformity tests for each benchmark. All three benchmarks pass the χ^2 uniformity test at 95% confidence ($\chi^2_{\text{crit}} = 7.815$ with 3 d.f.). The largest deviation from the expected 25% is only 0.54% (NE zone in Benchmark 1), and the zone balance improves with lattice size—Benchmark 3's maximum deviation is just 0.23%. This demonstrates that the exclusive zone mining mechanism works as intended.

Table 6. Zone distribution and χ^2 uniformity tests.

Bench.	NE%	NW%	SE%	SW%	Max Dev.	χ^2	p-Value
1	25.5	25.1	24.8	24.6	0.54%	1.143	0.767
2	25.4	25.1	24.9	24.6	0.40%	0.677	0.879
3	25.2	25.1	24.9	24.8	0.23%	0.360	0.948

7.4. Transaction Processing

Table 7 summarizes the transaction processing results across all benchmarks. Across all three benchmarks, 5312 transactions were submitted and every single one was confirmed, yielding a 100% confirmation rate. Figure 10 provides a detailed view from Benchmark 3.

Table 7. Transaction processing results.

Metric	Bench. 1	Bench. 2	Bench. 3	Average
Transactions Sent	1523	2178	1611	1771
Transactions Confirmed	1523	2178	1611	1771
Confirmation Rate	100%	100%	100%	100%
Avg. Confirmation Time	48.14 s	47.48 s	47.26 s	47.63 s
Min Confirmation Time	3.29 s	4.65 s	3.96 s	3.97 s
Max Confirmation Time	90.00 s	90.01 s	90.05 s	90.02 s

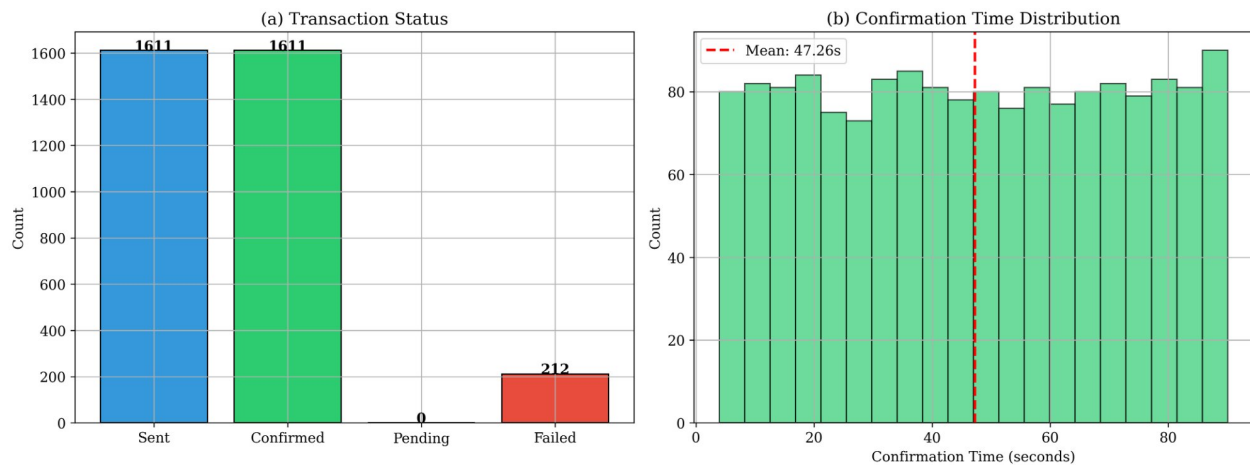


Figure 10. Transaction analysis (Benchmark 3). (a) All 1611 submitted transactions were confirmed (212 additional attempts were rejected at submission time due to balance constraints, not post-confirmation failures). (b) Confirmation time distribution showing a roughly uniform spread between 0 and 90 s, with mean at 47.26 s.

The average confirmation time of 47.63 s deserves interpretation. Unlike Bitcoin, where the confirmation time reflects the block interval (10 min on average), D-BTC's confirmation time is determined by how quickly a pending transaction is included in a mined block. The near-uniform distribution between 0 and 90 s suggests that transactions enter the mempool (memory pool) at random times relative to the block production cycle, which is consistent with a well-functioning system.

7.5. Security Analysis Revisited and Numerical Comparison

We give more comparisons concerning the reversal probabilities of Bitcoin and D-BTC based on Equations (26) and (27) of Section 5 assuming an honest majority. The advantage of D-BTC over Bitcoin is the transition from linear to quadratic decay in reversal probability.

Table 8 and Figure 11 quantify this advantage for $\alpha = 30\%$, a commonly studied attack threshold.

Table 8. Security comparison: D-BTC vs. Bitcoin ($\alpha = 30\%$).

k	Bitcoin $P(\text{rev})$	D-BTC $P(\text{rev})$	Advantage	BTC Equivalent
2	1.84×10^{-1}	3.37×10^{-2}	$5 \times$	$k_{\text{BTC}} = 4$
4	3.37×10^{-2}	1.30×10^{-6}	$3 \times 10^4 \times$	$k_{\text{BTC}} = 16$
6	6.20×10^{-3}	5.66×10^{-14}	$10^{11} \times$	$k_{\text{BTC}} = 36$
8	1.14×10^{-3}	2.82×10^{-24}	$4 \times 10^{20} \times$	$k_{\text{BTC}} = 64$
10	2.09×10^{-4}	1.59×10^{-37}	$10^{33} \times$	$k_{\text{BTC}} = 100$

The headline result in Table 8: six D-BTC confirmations provide security equivalent to 36 Bitcoin confirmations. At $k = 6$, D-BTC achieves a reversal probability of 5.66×10^{-14} , which is 10^{11} times smaller than Bitcoin's 6.20×10^{-3} .

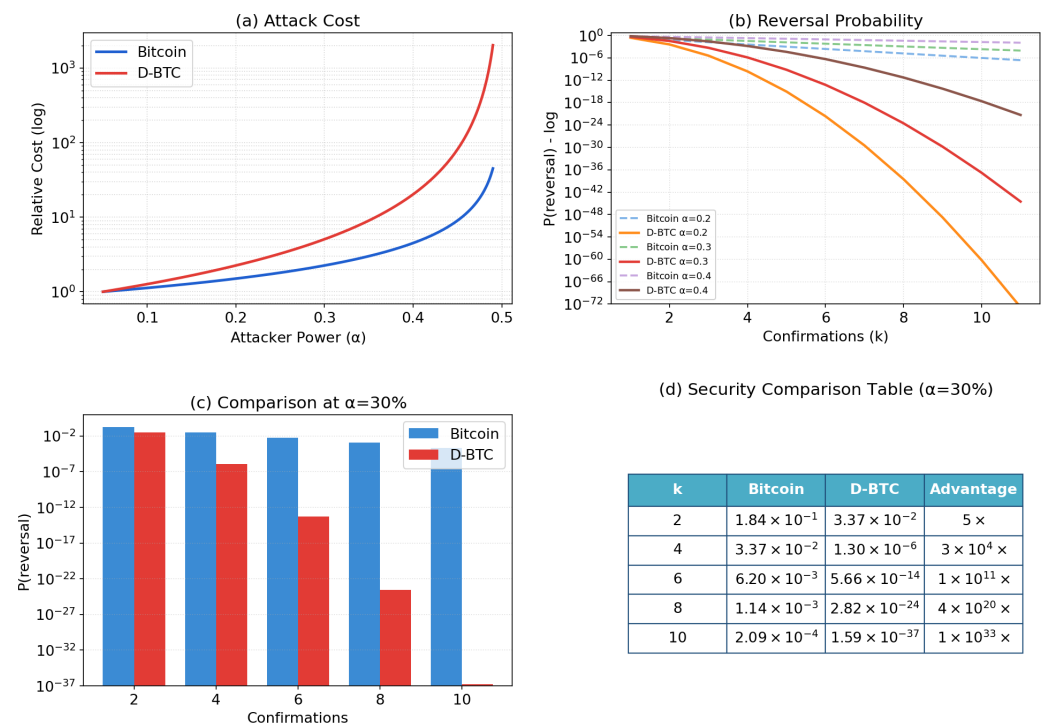


Figure 11. Security analysis. (a) Relative attack cost as a function of attacker hash power α : D-BTC's cost grows super-exponentially while Bitcoin's grows exponentially. (b) Reversal probability (log scale) versus number of confirmations for three attacker strengths ($\alpha = 0.2, 0.3, 0.4$): D-BTC curves plummet far below Bitcoin at every level. (c) Bar chart comparison at $\alpha = 30\%$. (d) Numerical comparison table embedded in the figure.

7.6. Consolidated Validation

As a conclusion of these tests, Table 9 provides a complete mapping between theoretical claims made in this paper and the experimental evidence supporting it. Table 10 combines the statistics of the three benchmarks.

Table 9. Theory-to-experiment validation summary.

Theoretical Claim	Experimental Evidence	Status
Frontier $ \partial L = 4\sqrt{n}$	$R^2 = 0.9896$ average, max deviation 1.92% (3 runs)	✓
Zone uniformity = 25%	24.5–25.5%, $\chi^2 p > 0.05$ in all 3 benchmarks	✓
Quadratic security	Computed $10^{11} \times$ advantage at $k = 6, \alpha = 30\%$	✓
Eventual consistency	100% TX confirmation, ≤ 1 block divergence	✓
Fork impossibility	0 structural forks across 18,786 blocks	✓
Parallel mining	Avg. 313 frontier positions vs. 1 for Bitcoin	✓

Remark 7 (Limitations and extensions). *The current testnet operates at low difficulty ($T_d = 2$) and small node count ($K = 4$) by design, prioritizing theoretical validation over production-scale performance. All three benchmarks confirm the isoperimetric prediction $|\partial \mathcal{L}| = \mathcal{O}(4\sqrt{n})$ (Table 5, $R^2 = 0.9896$), the 100% transaction confirmation rate (Table 7), and the absence of structural forks across 18,786 mined blocks. Towards a practical scaled proposal, we must vary the parameters such as K and target difficulty T_d .*

We emphasize that the key theoretical predictions are scale-invariant by construction: Proposition 2 holds for any n , and Proposition 9 depends only on the confirmation depth k , not the absolute lattice size. Larger-scale experiments will primarily test the engineering behavior (gossip

delays, memory management, fork resolution latency) rather than the mathematical predictions, which are established by proof.

Table 10. Combined statistics from three benchmarks.

Metric	Value
Total Blocks Mined	18,786
Total Transactions Processed	5312
Total Benchmark Duration	726.7 s (12.1 min)
Average Mining Throughput	10.25 blocks/second
Average Frontier R^2	0.9896
Transaction Confirmation Rate	100%
Peak Burst Throughput	70.8 TX/s
Average Confirmation Time	47.63 s
Maximum Node Divergence	1 block

8. Conclusions

We have shown that the D-BTC protocol has the potential to improve scalability and at the same time to enhance security. This is due to the introduction of geometric and topological notions in the realm of blockchains.

The main results are: (i) the four-directional PoW structure (Definition 6) cryptographically links each B-domino to its four cardinal neighbours; (ii) the simple connectivity invariant (Definition 4, Algorithm 2) prevents partition attacks by construction (Proposition 7) and enables deterministic fork resolution (Proposition 5); (iii) the discrete isoperimetric inequality (Lemma 1) yields a frontier of size $|\partial\mathcal{L}| = \Omega(\sqrt{n})$ (Proposition 2), enabling approximately $\sqrt{n}$ miners to work in parallel without sharding; (iv) the surface-based consensus provides a total ordering on valid lattices; and (v) the reduction argument establishes that the double-spend reversal probability decays as $(\alpha/(1-\alpha))^{k^2}$ (Proposition 9).

Some directions for future research are suggested, for example:

1. Adaptation of D-BTC to higher dimensions; again, isoperimetric inequalities will play a central role.
2. Adaptation to other consensus such as proof of stake, or hybridization of consensus.
3. Impact of other attacks such as selfish mining [10].
4. The zone-assignment strategy for miners in our tests is somewhat limited. It is possible to avoid this by using advanced hashing techniques.
5. Computation of maximum transaction production per unit of time.
6. More in-depth security analysis following the works of [27–29], including a native two-dimensional finality proof based on competing growth processes on $\mathbb{Z}^2$ under the simply connected constraint. Such an analysis would provide a self-contained 2D stochastic bound, strengthening the reduction argument of Section 5.4.
7. Integration of Smart Contracts.

Author Contributions: Conceptualization, S.B. and H.B.-a.; methodology, S.B. and H.B.-a.; software, S.B.; validation, S.B. and H.B.-a.; formal analysis, S.B. and H.B.-a.; investigation, S.B. and H.B.-a.; resources, S.B. and H.B.-a.; data curation, S.B. and H.B.-a.; writing—original draft preparation, S.B.; writing—review and editing, S.B. and H.B.-a.; visualization, S.B. and H.B.-a.; supervision, S.B. and H.B.-a.; project administration, S.B. and H.B.-a.; funding acquisition, S.B. and H.B.-a. All authors have read and agreed to the published version of the manuscript.

Funding: This research received no external funding.

Informed Consent Statement: Not applicable.

Data Availability Statement: The original contributions presented in this study are included in the article. Further inquiries can be directed to the corresponding author.

Acknowledgments: We are grateful to the reviewers for their help improving the quality of the paper.

Conflicts of Interest: The authors declare no conflicts of interest.

Abbreviations

The following abbreviations are used in this manuscript:

D-BTC	Domino Bitcoin
PoW	Proof of Work
BFS	Breadth-First Search
DFS	Depth-First Search
B-domino	Bitcoin domino
DAG	Directed Acyclic Graph
ID	Identifier
TTL	Time-to-Live
TX	Transaction

References

1. Nakamoto, S. Bitcoin: A Peer-to-Peer Electronic Cash System. 2008. Available online: <https://bitcoin.org/bitcoin.pdf> (accessed on 4 February 2026).
2. Popov, S. *The Tangle*; White Paper; IOTA Foundation: Berlin, Germany, 2018.
3. Sompolinsky, Y.; Shai Wyborski, S.; Zohar, A. PHANTOM and GHOSTDAG: A Scalable Generalization of Nakamoto Consensus. IACR Cryptology ePrint Archive. 2018. Available online: <https://eprint.iacr.org/2018/104> (accessed on 26 March 2026).
4. Kokoris-Kogias, E.; Jovanovic, P.; Gasser, L.; Gailly, N.; Syta, E.; Ford, B. OmniLedger: A secure, scale-out, decentralized ledger via sharding. In *Proceedings of the 2018 IEEE Symposium on Security and Privacy (SP), San Francisco, CA, USA, 20–24 May 2018*; IEEE: Piscataway, NJ, USA, 2018; pp. 583–598.
5. Luu, L.; Narayanan, V.; Zheng, C.; Baweja, K.; Gilbert, S.; Saxena, P. A secure sharding protocol for open blockchains. In *Proceedings of the 2016 ACM SIGSAC Conference on Computer and Communications Security*; Association for Computing Machinery: New York, NY, USA, 2016; pp. 17–30.
6. Grimmett, G. *Percolation*, 2nd ed.; Springer: Berlin/Heidelberg, Germany, 1999.
7. Wood, G. *Ethereum: A Secure Decentralised Generalised Transaction Ledger*; Ethereum Project Yellow Paper 151; 2014. Available online: <https://ethereum.github.io/yellowpaper/paper.pdf> (accessed on 16 June 2026).
8. Garay, J.; Kiayias, A.; Leonardos, N. The Bitcoin backbone protocol: Analysis and applications. In *Advances in Cryptology—EUROCRYPT 2015*; Oswald, E., Fischlin, M., Eds.; Springer: Berlin/Heidelberg, Germany, 2015; pp. 281–310.
9. Castro, M.; Liskov, B. Practical Byzantine fault tolerance. In *Proceedings of the Third Symposium on Operating Systems Design and Implementation, New Orleans, LA, USA, 22–25 February 1999*; USENIX Association: Berkeley, CA, USA, 1999; pp. 173–186.
10. Eyal, I.; Sirer, E.G. Majority is not enough: Bitcoin mining is vulnerable. *Commun. ACM* **2018**, *61*, 95–102.
11. Gervais, A.; Karame, G.O.; Wüst, K.; Glykantzis, V.; Ritzdorf, H.; Capkun, S. On the security and performance of proof of work blockchains. In *Proceedings of the 2016 ACM SIGSAC Conference on Computer and Communications Security*; Association for Computing Machinery: New York, NY, USA, 2016; pp. 3–16.
12. Croman, K.; Decker, C.; Eyal, I.; Gencer, A.E.; Juels, A.; Kosba, A.; Miller, A.; Saxena, P.; Shi, E.; Sirer, E.G.; et al. On Scaling Decentralized Blockchains. In *Financial Cryptography and Data Security*; Clark, J., Meiklejohn, S., Ryan, P., Wallach, D., Brenner, M., Rohloff, K., Eds.; Springer: Berlin/Heidelberg, Germany, 2016; pp. 106–125.
13. Pass, R.; Seeman, L.; Shelat, A. Analysis of the blockchain protocol in asynchronous networks. In *Cryptology—EUROCRYPT 2017*; Coron, J.S., Nielsen, J., Eds.; Springer: Berlin/Heidelberg, Germany, 2017; pp. 643–673.
14. Guo, D.; Ren, L. Bitcoin’s latency–security analysis made simple. In *Proceedings of the 4th ACM Conference on Advances in Financial Technologies (AFT)*; Association for Computing Machinery: New York, NY, USA, 2022; pp. 244–253.
15. Cao, S.; Guo, D. Security, latency, and throughput of proof-of-work Nakamoto consensus. *IEEE Trans. Inf. Theory* **2025**, *71*, 4708–4731. [CrossRef]
16. Sompolinsky, Y.; Zohar, A. Secure high-rate transaction processing in Bitcoin. In *Financial Cryptography and Data Security*; Rainer, B., Tatsuo, O., Eds.; Springer: Berlin/Heidelberg, Germany, 2015; pp. 507–527.

17. Li, C.; Li, P.; Zhou, D.; Xu, W.; Long, F.; Yao, A. Scaling Nakamoto consensus to thousands of transactions per second. *arXiv* **2018**. [[CrossRef](#)]
18. Wang, Q.; Yu, J.; Chen, S.; Xiang, Y. SoK: DAG-based Blockchain Systems. *ACM Comput. Surv.* **2023**, *55*, 261. [[CrossRef](#)]
19. Lamport, R.; Shostak, R.; Pease, M. The Byzantine generals problem. *ACM Trans. Program. Lang. Syst.* **1982**, *4*, 382–401. [[CrossRef](#)]
20. Lynch, N.A. *Distributed Algorithms*; Morgan Kaufmann: San Francisco, CA, USA, 1996.
21. De Silva, V.; Ghrist, R. Coverage in sensor networks via persistent homology. *Algebr. Geom. Topol.* **2007**, *7*, 339–358. [[CrossRef](#)]
22. Karp, B.; Kung, H.T. GPSR: Greedy perimeter stateless routing for wireless networks. In *Proceedings of the 6th Annual International Conference on Mobile Computing and Networking*; Association for Computing Machinery: New York, NY, USA, 2000; pp. 243–254.
23. Grädel, E. Domino Games and Complexity. *SIAM J. Comput.* **1990**, *19*, 787–804. [[CrossRef](#)]
24. Altshuler, T.; Yanovsky, V.; Vainsencher, D.; Wagner, I.A.; Bruckstein, A.M. On minimal perimeter polyminoes. In *Proceedings of the Conference Discrete Geometry for Computer Imagery*; Kuba, A., Nyúl, L.G., Palágyi, K., Eds.; Springer: Berlin/Heidelberg, Germany, 2006; pp. 17–28.
25. Barequet, G.; Ben-Shachar, G. Properties of Minimal-Perimeter Polyominoes. In *Computing and Combinatorics. COCOON 2018*; Wang, L., Zhu, D., Eds.; Springer: Berlin/Heidelberg, Germany, 2018; pp. 120–129.
26. Vince, A. An Extremal Graph Problem on a Grid and an Isoperimetric Problem for Polyominoes. *Electron. J. Comb.* **2024**, *31*, P2.5. [[CrossRef](#)] [[PubMed](#)]
27. Rosenfeld, M. Analysis of hashrate-based double spending. *arXiv* **2014**, arXiv:1402.2009.
28. Grunspan, C.; Pérez-Marco, R. Double spend races. *Int. J. Theor. Appl. Financ.* **2018**, *21*, 1850053. [[CrossRef](#)]
29. Georgiadis, E.; Zeilberger, D. A combinatorial-probabilistic analysis of bitcoin attacks. *J. Differ. Equ. Appl.* **2019**, *25*, 56–63.

Disclaimer/Publisher’s Note: The statements, opinions and data contained in all publications are solely those of the individual author(s) and contributor(s) and not of MDPI and/or the editor(s). MDPI and/or the editor(s) disclaim responsibility for any injury to people or property resulting from any ideas, methods, instructions or products referred to in the content.