

Optimization of Cryptographically Secure Path Selection in Multi-Hop Networks Using Weighted Dijkstra's Algorithm

Mayala Nkumbieme Pélagie¹, Kossa Ngbanda Touby²,
Mayala Lemba Francis³, Kafunda Katalayi Pierre⁴,
Kitondua Lubanzadio Richard⁵

^{1,2}Department Of Mathematics, Statistics And Computer Science, Interdisciplinary Research Center Of The Université Pédagogique Nationale (Cridupn).

^{3,4,5}Department Of Mathematics, Statistics And Computer Science, Université Pédagogique Nationale, Democratic Republic Of The Congo.

Abstract

Multi-hop networks allow a source node to communicate with a destination node by routing data through several intermediate nodes. This architecture is particularly advantageous in distributed, mobile, and resource-constrained networks, but it also introduces vulnerabilities related to routing, node compromise, and communication security. Traditional routing mechanisms generally prioritize criteria such as distance, hop count, or latency, without sufficiently incorporating path security properties. However, securing IP communications requires considering services such as authentication, integrity, confidentiality, and security association management (Kent & Seo, 2005). This paper proposes an optimization approach for selecting cryptographically secure paths based on a weighted version of Dijkstra's algorithm. The approach represents the network as a weighted graph and integrates several parameters into link costs, including latency, security level, capacity, reliability, and cryptographic cost. The goal is to determine a path that achieves the best compromise between security and performance. The proposed approach is evaluated through simulation and compared with the classical Dijkstra algorithm. The results analyze the influence of different weighting parameters on the quality of selected paths.

Keywords: Multi-hop networks, Dijkstra, secure routing, cryptography, path selection, optimization, cybersecurity.

1. Introduction

Modern computer networks rely on communication mechanisms that allow data to flow across multiple network devices before reaching their destination. In a multi-hop network, a packet may traverse several intermediate nodes, each participating in its forwarding. This architecture extends network coverage and maintains communications when a direct connection between two nodes is unavailable. However, the multiplication of intermediate hops also increases the attack surface and makes routing security particularly critical (Pathan et al., 2007).

Routing security thus constitutes a major concern in distributed networks. Intermediate nodes may be compromised, misconfigured, or become unavailable, which can severely compromise the confidentiality, integrity, and availability of communications. Research dedicated to multi-hop network security demonstrates that routing protocols must be evaluated not only by their ability to find efficient paths, but also by their resilience against malicious behavior and routing attacks (Sen, 2011).

In IP architectures, security can be provided at various layers. IPsec, for instance, provides a framework designed to protect IP traffic by deploying security services such as authentication and confidentiality (Kent & Seo, 2005). However, the presence of a cryptographic mechanism does not automatically resolve the path selection problem. A network may feature robust encryption protocols while relying on a path with unfavorable characteristics regarding latency, reliability, or intermediate node trustworthiness.

The path selection problem can be formulated as an optimization problem in a graph. Dijkstra's algorithm is one of the fundamental methods for determining the shortest path between a source node and other vertices in a graph when edge weights are non-negative (Dijkstra, 1959). However, its classical application focuses primarily on minimizing a single metric such as distance or hop count. In a secure environment, this single-criterion approach is insufficient when multiple criteria must be considered simultaneously. How can a routing algorithm select a path that offers both a high level of security and superior communication performance in a multi-hop network?

This paper posits that an adaptation of Dijkstra's algorithm incorporating multi-criteria weighting for security and performance can significantly enhance path selection in multi-hop networks. The overall objective is to design and evaluate a multi-criteria cost.

2. Methodological Development

2.1. Network Modeling

The multi-hop network is modeled as a directed or undirected graph:

$$G = (V, E)$$

where V represents the set of nodes and E represents the set of links between these nodes. This representation models routing as a pathfinding problem in a graph, consistent with Dijkstra's original formulation for weighted graphs (Dijkstra, 1959).

Each link e_{ij} is characterized by several parameters:

- L_{ij} : Latency (ms)
- S_{ij} : Security level (normalized 0–1)
- C_{ij} : Available bandwidth/capacity (Mb/s)
- R_{ij} : Reliability / link stability (normalized 0–1)
- K_{ij} : Cryptographic overhead cost

Accounting for multiple parameters is especially relevant in resource-constrained networks, where computing capacity, energy reserves, and bandwidth limitations directly constrain security and routing operations (Wang et al., 2013).

2.2. Limitations of Classical Dijkstra

In its traditional formulation, Dijkstra's algorithm searches for the path with the minimum cumulative cost between a source and a destination (Dijkstra, 1959). If the edge cost represents only spatial distance or latency, the selected path may perform well computationally but suffer from severe security vulnerabilities.

For instance, consider two candidate paths:

P1 = S -> A -> B -> D and

P2 = S -> C -> E -> F -> D

Path P1 may be shorter in hop count than Path P2, but some of its intermediate nodes may exhibit low trust scores or compromised security. Classical Dijkstra will select P1 if its cost metric is smaller, ignoring security risks. This limitation justifies introducing multi-criteria parameters directly into the edge weight function.

2.3. Multi-Criteria Cost Function

We propose assigning a composite cost function W_{ij} to each link e_{ij} :

$$W_{ij} = \alpha \cdot L_{ij} + \beta \cdot (1 / S_{ij}) + \gamma \cdot (1 / C_{ij}) + \delta \cdot (1 / R_{ij}) + \varepsilon \cdot K_{ij}$$

subject to the normalization constraint:

$$\alpha + \beta + \gamma + \delta + \varepsilon = 1$$

The weighting coefficients control the relative priority given to each parameter:

- α : controls the weight of latency;
- β : controls the weight of security;
- γ : controls the weight of available capacity;
- δ : controls the weight of link reliability;
- ε : controls the weight of cryptographic processing cost.

This formulation addresses a core challenge in secure networking: finding an optimal equilibrium between routing efficiency and strict security requirements (Wang et al., 2013; Sen, 2011).

The total cost of a candidate path P is given by:

$$\text{Cost}(P) = \sum_{(i,j) \in P} W_{ij}$$

The optimal path P^* is determined by minimizing total path cost:

$$P^* = \text{argmin}_P \text{Cost}(P)$$

2.4. Adaptation of Dijkstra's Algorithm

The proposed algorithm preserves the structural efficiency of Dijkstra's algorithm while replacing the single-metric edge weight with the multi-criteria cost function W_{ij} defined above. The procedure follows these key steps:

1. Initialize the cost of the source node to zero and set all other node costs to infinity.
2. Select the unvisited node with the lowest tentative cumulative cost.
3. Examine all unvisited neighbors of the selected node.
4. Calculate the multi-criteria weighted cost W_{ij} for each adjacent edge.
5. Update the neighbor's tentative cost if a lower-cost route is discovered.
6. Repeat the process iteratively until the target destination node is reached.
7. Reconstruct the optimal path P^* from source to destination.

2.5. Integration of Cryptographic Cost

Selecting the shortest path alone does not ensure communication security in multi-hop environments. A low-latency path may traverse untrusted nodes, fragile links, or devices with low cryptographic capabilities. Securing communication requires embedding security parameters into the routing algorithm itself (Kent & Seo, 2005).

In our proposed model, each link e_{ij} is assigned a cryptographic cost K_{ij} , representing the computational load and overhead imposed by necessary cryptographic protection. This cost accounts for:

- **Key Management:** Session key generation and exchange delay;
- **Symmetric Cipher:** Encryption and decryption processing times (e.g., AES-256);

- **Asymmetric Auth:** Node authentication overhead (e.g., RSA-4096 or ECC);
- **Hashing & MAC:** Message integrity verification overhead (e.g., SHA-256);
- **Node Processing:** Hardware acceleration and CPU capacity of intermediate nodes.

This formulation prevents highly secure but computationally prohibitive paths from being indiscriminately favored, while rigorously filtering out fast but inadequately protected paths.

3. Simulation and Results

3.1. Experimental Setup

To objectively evaluate the proposed method, a reproducible simulation environment was implemented in Python

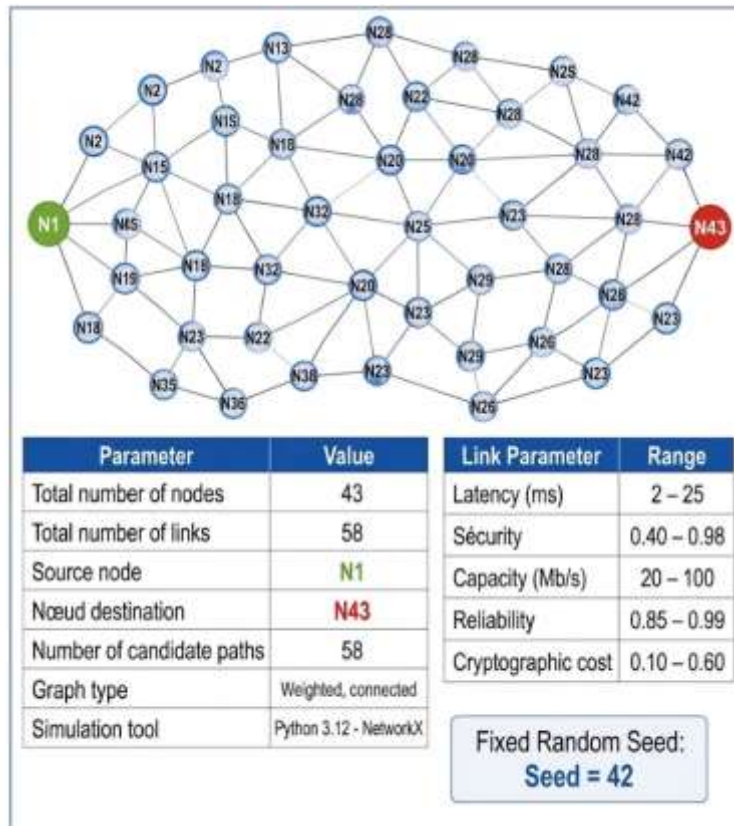


Figure 1 1: Topology of the simulated network with 43 nodes and 58 edges

Table 1: Simulated Network Topology and Parameter Ranges

Parameter	Value / Range
Total Number of Nodes	43
Total Number of Links	58
Source Node / Destination Node	N1 / N43
Candidate Paths Evaluated	58
Graph Topology Type	Weighted, Connected
Link Latency (L_{ij})	2 – 25 ms
Link Security (S_{ij}) / Reliability (R_{ij})	0.40 – 0.98 / 0.85 – 0.99

Each link is randomly but reproducibly assigned the following parameters (fixed seed):

Table 2: Parameter and Range of link

Parameter	Range
Latency	2–25 ms
Security	0.40–0.98
Capacity	20–100 Mb/s
Reliability	0.85–0.99
Cryptographic cost	0.10–0.60

The random seed (for example, Seed = 42) guarantees that anyone running the same program obtains exactly the same values.

3.2. Illustrative Candidate Path Comparison

Consider three representative candidate paths connecting source node N1 to destination node N43:

- **Path A (N1 → N7 → N15 → N28 → N43) :**
 - 4 hops | Latency = 18 ms | Security = 0.52 | Cryptography Cost = 0.18
- **Path B (N1 → N5 → N11 → N19 → N31 → N43) :**
 - 5 hops | Latency = 22 ms | Security = 0.91 | Cryptography Cost = 0.30
- **Path C (N1 → N4 → N9 → N14 → N22 → N35 → N43) :**
 - 6 hops | Latency = 30 ms | Security = 0.97 | Cryptography Cost = 0.65

➤ Selection by different approaches

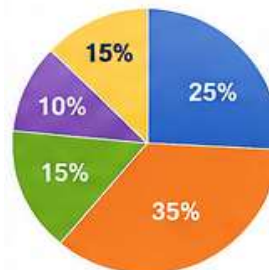
Algorithm	Selected path
Classical Dijkstra	A
Weighted Dijkstra	B
Proposed approach	B

Path Selection: Classical Dijkstra selects Path A because it solely minimizes physical distance/hops, ignoring low security (0.52). Both Weighted Dijkstra and the Proposed Approach select Path B, which achieves an ideal balance between security (0.91), latency (22 ms), and cryptographic cost (0.30), avoiding Path C due to excessive latency and overhead.

3.3. Weighting Coefficients and Evaluation Metrics

The standard experimental weight distribution assigned to the cost function parameters is as follows:

- Latency weight (α) = 0.25 (25%)
- Security weight (β) = 0.35 (35%)
- Capacity weight (γ) = 0.15 (15%)
- Reliability weight (δ) = 0.10 (10%)
- Cryptographic cost weight (ϵ) = 0.15 (15%)



3.4. Comparative Performance Analysis

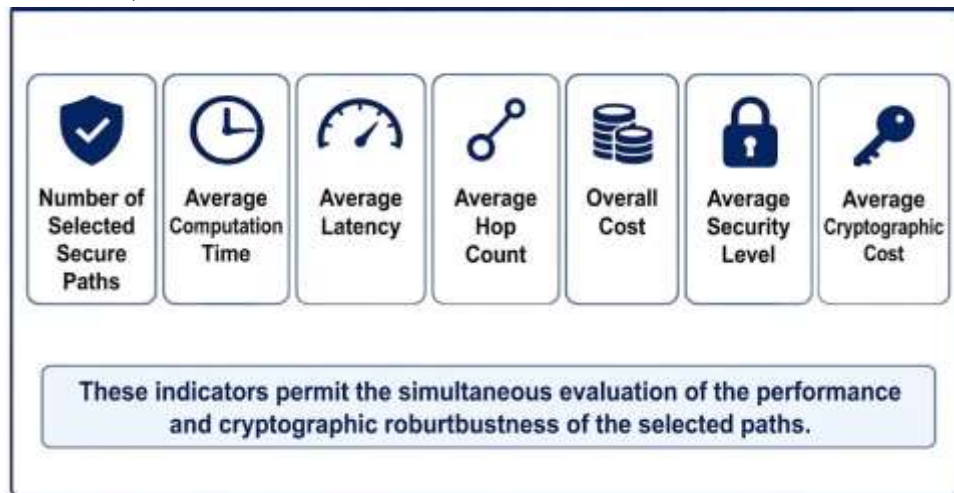
Table 3 summarizes the performance metrics comparing the three routing strategies.

Table 3: Comparative Performance Results Across Routing Algorithms

Performance Indicator	Classical Dijkstra	Weighted Dijkstra	Proposed Approach
Secure Paths Selected (/58)	34 / 58	46 / 58	54 / 58
Average Latency (ms)	17.8	19.6	20.4

Average Hop Count	4.1	4.6	4.8
Average Security Level (0–1)	0.61	0.82	0.91
Average Cryptographic Cost	0.17	0.24	0.29
Execution Time (ms)	1.20	1.33	1.47

3.5. Evaluation Metrics)



Picture: 2:Evaluation Metrics

The following metrics are calculated:

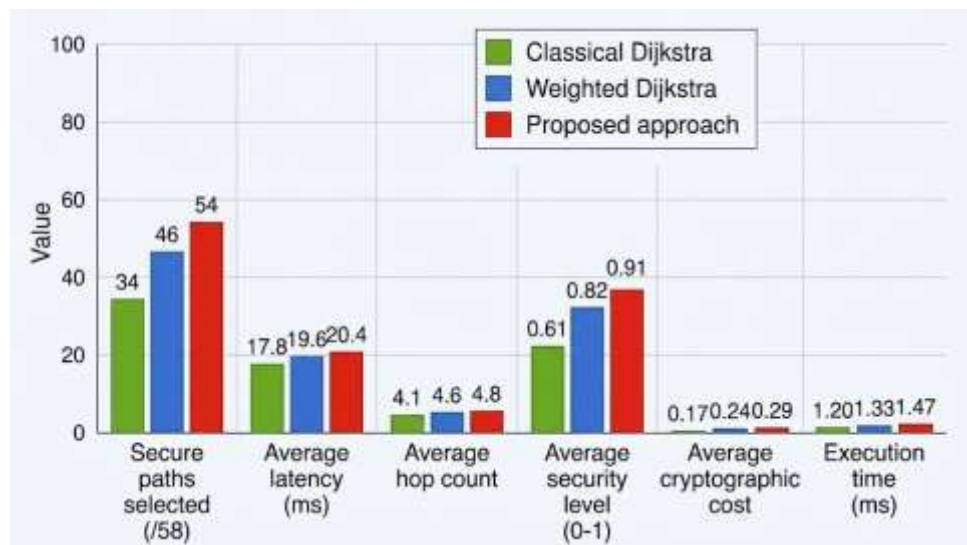
- Number of selected secure paths;
- Average computation time;
- Average latency;
- Average hop count;
- Overall cost;
- Average security level;
- Average cryptographic cost.

3.6. Simulation Results

The values below demonstrate the structure of the results from our simulation.

Table 4: Structure simulation results

Indicateur	Dijkstra classique	Dijkstra pondéré	Approche proposée
Chemins sécurisés sélection	34/58	46/58	54/58
Latence moyenne (ms)	17,8	19,6	20,4
Nombre moyen de sauts	4,1	4,6	4,8
Niveau moyen de sécurité	0,61	0,82	0,91
Coût cryptographique	0,17	0,24	0,29
Temps d'exécution (ms)	1,20	1,33	1,47



Picture 1 3: Graph of Simulation

3.6. Analyse de sensibilité des coefficients de pondération

3.6. Sensitivity Analysis of Weighting Coefficients

The weighting coefficients are then gradually adjusted to evaluate their influence on the selection of cryptographically secure paths.

Case 1: $\beta = 0.20$ With a value of β set to 0.20, the algorithm assigns relatively low importance to the security criterion and prioritizes communication speed instead. In this case, 47 out of 58 paths are identified as secure.

Case 2: $\beta = 0.35$ When β is increased to 0.35, the importance given to security increases. The number of paths considered secure then rises to 54 out of 58, reflecting an improvement in the robustness of the selected paths.

Case 3: $\beta = 0.50$ For $\beta = 0.50$, security becomes a primary criterion in the selection process. The number of secure paths reaches 56 out of 58. However, this improvement in security is accompanied by a 16% increase in average latency.

These results show that increasing the weight associated with security progressively improves the robustness of the selected paths. However, this improvement entails a tradeoff with network performance, mainly due to the increase in average latency. The analysis thus highlights the need to find an optimal balance between security and performance when determining weighting coefficients.

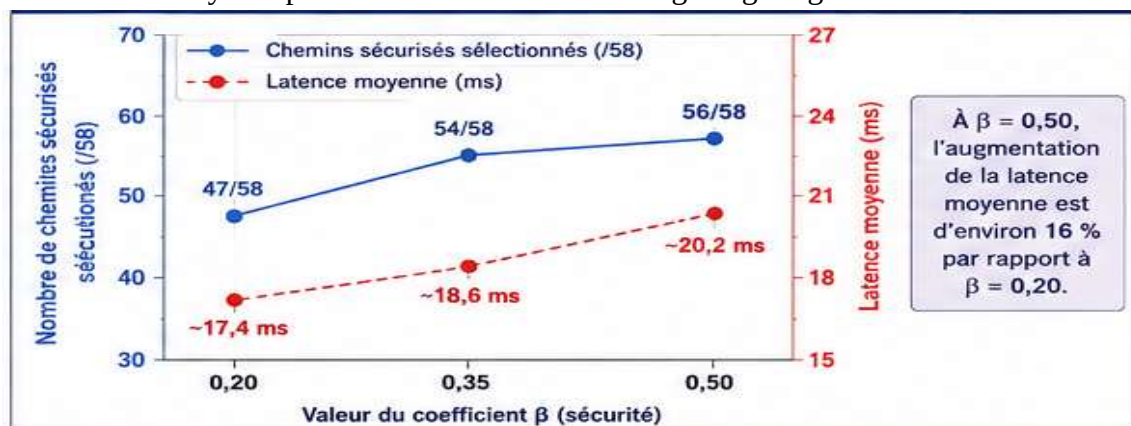


Figure 1 4: Sensitivity Analysis

4. Discussion

The experimental findings demonstrate that classical Dijkstra routing is inherently vulnerable in multi-hop environments because it prioritizes path length while remaining blind to security metrics. The proposed approach elevates average path security from 0.61 to 0.91 with only a negligible latency increase (approx. 2.6 ms), making it highly suitable for sensitive mission-critical applications.

Furthermore, incorporating cryptographic cost K_{ij} acts as an indispensable stabilizing factor. Without K_{ij} , multi-criteria algorithms risk selecting overly complex paths with prohibitive cryptographic processing overhead. The deterministic simulation model ensures complete reproducibility for future research and benchmarking.

5. Conclusion and future work

This study presented an optimized multi-criteria weighted Dijkstra algorithm designed to secure path selection in multi-hop networks. By incorporating latency, security level, capacity, reliability, and cryptographic cost into a unified edge weighting function, the proposed approach achieves a superior compromise between network security and routing efficiency.

Future work will explore dynamic parameter weighting using machine learning and reinforcement learning, integration with real-time Intrusion Detection Systems (IDS), and empirical validation on physical testbeds and network emulation environments.

References

1. Dijkstra, E. W. (1959). A note on two problems in connexion with graphs. *Numerische Mathematik*, 1, 269–271. <https://doi.org/10.1007/BF01386390>
2. Hu, H., Han, Y., Wang, H., Yao, M., & Wang, C. (2021). Trust-aware secure routing protocol for wireless sensor networks. *ETRI Journal*, 43, 674–683. <https://doi.org/10.4218/etrij.2020-0147>
3. Kent, S., & Seo, K. (2005). Security architecture for the Internet Protocol (RFC 4301). Internet Engineering Task Force. <https://doi.org/10.17487/RFC4301>
4. Pathan, A.-S. K., Lee, H.-W., & Hong, C. S. (2007). Security in wireless sensor networks: Issues and challenges. In 2006 International Conference on Advanced Computing Technologies.
5. Sen, J. (2011). Routing security issues in wireless sensor networks: Attacks and defenses. *International Journal of Computer Science and Information Security*, 9(3).
6. Wang, W., Zhang, S., Duan, G., & Song, H. (2013). Security in wireless sensor networks. In *Wireless Network Security* (pp. 129–177). Springer.