D. Ferraris, L. Russo and L. Monti, "DrATC+: A Divide et Impera Extension to Trust-Based Dynamic Routing", *The 21st International Workshop on Security and Trust Management (STM 2025)*, 2025.
http://doi.org/10.1007/978-3-032-06155-3_7
NICS Lab. Publications: https://www.nics.uma.es/publications

DrATC+: A *Divide et Impera* Extension to Trust-Based Dynamic Routing

Davide Ferraris[1], Letizia Russo[2], and Lorenzo Monti[2]

[1] Network, Information and Computer Security Lab, University of Malaga, Malaga, Spain
ferraris@uma.es
[2] Cubit Innovation Labs, Via Mario Giuntini, 25, Cascina, Pisa, Italy
{letizia.russo,lorenzo.monti}@cubitlab.com

Abstract. Trust is a critical yet often overlooked factor in network routing, especially in decentralized and dynamic environments such as IoT and ad-hoc networks. The DrATC algorithm addressed this gap by incorporating a comprehensive set of trust characteristics into dynamic routing decisions. In this paper, we propose DrATC+, an extension of the original model that applies the *divide et impera* (divide and conquer) principle to trust-based routing. DrATC+ enhances resilience and security by splitting messages into segments and routing them through multiple trusted paths, selected based on composite trust evaluations. We present the design and operational workflow of the algorithm, supported by evaluation scenarios and a real-world IoT use case. Our results demonstrate that DrATC+ improves fault tolerance, mitigates the impact of malicious nodes, and adapts effectively to changing trust conditions. This work lays the foundation for more robust and adaptive trust-aware routing protocols in future networked systems.

Keywords: Trust · Network · Routing

1 Introduction

In the evolving landscape of network communication, the reliability and security of data transmission have become fundamental. Traditional routing protocols, while efficient in determining optimal paths based on metrics such as distance or latency, often overlook a critical factor: trust. The original DrATC (Dynamic routing Algorithm based on Trust Characteristics) [6] addressed this gap by introducing a trust-aware routing mechanism that evaluates nodes based on a rich set of trust characteristics—including direct and indirect trust, transitivity, context-dependence, and more. This approach significantly improved the robustness of routing decisions in dynamic and potentially adversarial environments.

However, relying on a single trusted path, even when carefully selected, introduces inherent vulnerabilities [17]. A compromised or failing node along the path can disrupt the entire communication, leading to data loss or security breaches. Moreover, concentrating traffic on a single route may result in congestion or

inefficient resource utilization [16]. To address these limitations, we propose an extension to the DrATC algorithm that integrates the classical *divide et impera* (divide and conquer) strategy into trust-based routing.

Our approach, termed DrATC+, enhances the original model by dividing messages into multiple segments and routing them through distinct, highly trusted paths. This strategy not only increases resilience against node failures and malicious behavior but also enables better load balancing and improved adaptability to network dynamics. By leveraging multiple trusted routes, DrATC+ ensures that even if one path is compromised, the overall integrity and delivery of the message can be preserved.

This paper presents the design, implementation, and evaluation of DrATC+, demonstrating how the integration of message segmentation and multipath trust-aware routing can significantly enhance the security and reliability of network communication. We also explore various trust characteristics and their role in the path selection process, providing a flexible framework adaptable to different network scenarios and security requirements.

The paper is composed as follows. In Section 2, we discuss the background and related works. In Section 3, we reintroduce DrATC algorithm and present the trust characteristics. In Section 4, we present the extended dynamic routing algorithm based on the *divide et impera* strategy. In Section 5, we explain the algorithm design and in Section 6, we propose a set of possible scenarios for implementing our approach. Then, in Section 7, we extend such scenarios in order to present a deeper analysis on how our algorithm works. Finally, in Section 8, we conclude and present the future work.

2 Background and Related works

In this section, we will firstly present definitions of trust, then we will discuss about which algorithm exists in routing and finally we discuss about existing works about trusted routing.

2.1 Trust in general and in Network Routing

Trust is a foundational concept that spans multiple disciplines, including philosophy, psychology, and computer science. In computer science, trust is generally understood as the quantified belief or expectation that an entity will behave as intended in a given context. Marsh [10] was among the first to formalize trust as a computational construct, introducing a model where trust is dynamic and context-dependent. Abdul-Rahman and Hailes [2] described trust as a subjective probability that an entity will perform a particular action that is beneficial or at least not harmful.

Further, Gambetta [7] emphasized trust's role in reducing complexity and uncertainty in systems composed of autonomous agents. This is especially relevant in open, dynamic, and heterogeneous environments such as peer-to-peer

systems, cloud computing, and the Internet of Things (IoT), where entities must interact without centralized control or prior knowledge of each other.

In recent works, trust is increasingly considered a multidimensional property, encompassing parameters such as direct experience, reputation, context-awareness, and time sensitivity. As highlighted in [13], trust in IoT systems must be integrated throughout the full system development life cycle to address heterogeneity and ensure secure interactions between diverse entities.

So, Trust is a multifaceted and context-dependent concept that plays a critical role in secure and reliable communication, especially in decentralized and dynamic networks. In traditional routing protocols such as Dijkstra's algorithm, Open Shortest Path First (OSPF), or Border Gateway Protocol (BGP), trust is often assumed implicitly—nodes are expected to behave correctly. However, this assumption does not hold in environments where nodes may be compromised, misconfigured, or malicious.

To address this, trust-aware routing protocols have emerged, incorporating trust metrics into the route selection process. These metrics are typically derived from direct interactions (e.g., successful packet forwarding), indirect observations (e.g., recommendations from other nodes), or hybrid approaches. Trust-based routing has been particularly effective in ad-hoc networks, wireless sensor networks (WSNs), and peer-to-peer systems, where centralized control is limited or absent.

The original DrATC algorithm introduced a flexible framework for trust-based routing by leveraging a comprehensive set of trust characteristics, including direct and indirect trust, transitivity, directionality, context-dependence, and dynamic adaptability. This model allowed for fine-grained control over how trust is computed and applied in routing decisions, enabling more secure and context-aware communication.

2.2 Routing Algorithm and *Divide et Impera* in Network Systems

Routing algorithms are essential in determining efficient and reliable paths for data packets to traverse a network. Traditional routing algorithms fall into two broad categories: static routing and dynamic routing.

Static routing uses predefined paths and is suitable for simple, unchanging networks. In contrast, dynamic routing adapts to network changes and conditions in real time. Some of the most common dynamic routing algorithms include:

- Distance Vector Routing: Nodes share information with their neighbors to update routing tables, with algorithms like Bellman-Ford used to calculate the shortest paths.
- Link State Routing: Each node has complete visibility of the network and independently calculates the optimal path using algorithms like Dijkstra's.
- Path Vector Routing: Used in protocols such as BGP, where each route advertisement carries the entire path information, enabling loop prevention and policy-based decisions.

Divide and conquer strategies have been employed in various routing protocols to manage complexity and improve scalability. In hierarchical routing, for example, large networks are divided into regions or clusters. This approach reduces overhead by allowing intra-cluster communication to use simpler routing methods, while inter-cluster communication is managed through designated gateway nodes. Such strategies are especially beneficial in large-scale systems like IoT and MANETs (Mobile Ad Hoc Networks), where routing decisions must remain efficient despite high node mobility and network dynamics.

The *divide et impera* (divide and conquer) paradigm has been widely applied in computer science to improve performance, scalability, and fault tolerance. In networking, this principle underpins several strategies, such as load balancing, parallel data transmission, and multipath routing. By splitting a task into smaller, manageable parts and distributing them across multiple paths or nodes, systems can achieve higher efficiency and resilience.

In the context of routing, multipath strategies have been explored to enhance throughput, reduce latency, and improve fault tolerance. Protocols such as Multipath TCP (MPTCP) and Ad hoc On-demand Multipath Distance Vector (AOMDV) routing demonstrate the benefits of using multiple paths for data transmission. However, these approaches typically focus on performance metrics and do not incorporate trust as a primary factor in path selection.

2.3 Trust-Aware Multipath Routing

While traditional routing focuses on metrics such as hop count, latency, or bandwidth, trusted routing algorithms integrate security-related parameters, especially trust, into the routing decision process. These algorithms are critical in environments where nodes might be compromised or unreliable, such as ad hoc networks, sensor networks, and IoT systems.

One of the earliest approaches, by Pirzada and McDonald [15], introduced trust as a metric in ad hoc networks to isolate malicious nodes and improve route stability. Since then, several trust models have been proposed using direct trust (based on personal experience), indirect trust (reputation), and hybrid models. These trust values are then incorporated into routing decisions to prefer more reliable paths.

Recent advancements have considered trust interoperability, especially in IoT contexts where heterogeneous trust models co-exist. As discussed in [14], devices may interpret and calculate trust differently, necessitating frameworks that can translate and unify trust values across domains.

Moreover, [6] introduced DrATC (Dynamic Routing Algorithm based on Trust Characteristics), which integrates both direct and indirect trust metrics to dynamically select routes. This algorithm is context-sensitive, supporting flexible trust evaluation depending on application needs (e.g., high confidentiality vs. performance). It also supports selecting alternative routes based on specific trust values to avoid risky nodes, enhancing both the reliability and security of communications.

The inclusion of divide and conquer strategies in trusted routing is also gaining traction. By organizing networks into trust-based clusters or zones, each with its own local trust management, systems can isolate faults, reduce overhead, and scale more efficiently. These approaches allow trust evaluation and route selection to occur locally within clusters, while high-level coordination ensures inter-cluster trust routing. Such hybrid models benefit from the robustness of trust-based security and the efficiency of hierarchical or modular designs.

While trust-based routing and multipath routing have been studied independently, their integration remains limited. Some research has explored trust-aware multipath routing in specific domains, such as WSNs or IoT, but often with simplified trust models or static trust assumptions. These approaches may not fully capture the dynamic and context-sensitive nature of trust, nor do they leverage the full potential of message segmentation and distributed delivery.

The proposed extension, DrATC+, addresses this gap by combining the strengths of trust-based routing and *divide et impera*. By splitting messages and routing segments through multiple trusted paths, DrATC+ enhances resilience, mitigates the risk of single-point failures, and adapts to dynamic trust conditions. This novel integration offers a more robust and secure routing framework, particularly suited for environments where trust is variable and critical.

3 Trust-Based Routing Recap (DrATC Overview)

The DrATC (Dynamic routing Algorithm based on Trust Characteristics) model was introduced to address the limitations of traditional routing protocols that often overlook the trustworthiness of nodes in a network. Unlike conventional approaches that prioritize metrics such as hop count or latency, DrATC integrates a rich set of trust characteristics into the routing decision process, enabling more secure and context-aware communication.

3.1 Trust Characteristics

At the core of DrATC is a comprehensive taxonomy of trust characteristics, derived from interdisciplinary research in computer science, sociology, and psychology. These characteristics, summarized in Table 1 along with the research papers that highlighted them, define how trust is perceived, evaluated, and applied in routing decisions. The key dimensions include:

- Direct Trust: Based on firsthand interactions between nodes.
- Indirect Trust: Derived from recommendations or observations by other nodes.
- Transitive Trust: Trust can propagate through chains of trusted nodes.
- Directed Trust: Trust is not necessarily reciprocal between two nodes.
- Dynamic Trust: Trust values evolve over time based on behavior.
- Context-Dependent Trust: Trust varies depending on the type of communication or application.

- Local vs. Global Trust: Trust can be specific to node pairs or universally known.
- Specific vs. General Trust: Trust may apply to specific services or be broadly applicable.
- Subjective vs. Objective Trust: Trust can be based on personal experience or measurable metrics.
- Composite and Measurable Trust: Trust is often a weighted combination of multiple attributes and can be quantified.

These characteristics allow DrATC to adapt to a wide range of network environments and security requirements.

Table 1. Characteristics of trust

Direct	[3]
Indirect	[1]
Transitive	[5, 18]
Directed	[18]
Dynamic	[8, 4, 13]
Context-dependent	[1, 12]
Local	[1, 5]
Global	[1]
Specific	[9, 11]
General	[9, 11]
Subjective	[8, 18]
Objective	[1]
Composite-property	[8, 18]
Measurable	[18]

3.2 Operational Workflow

The DrATC algorithm operates dynamically, continuously updating trust evaluations and routing decisions based on real-time network conditions. Its core workflow includes the following phases:

- **Initialization**: Each node initializes trust scores based on prior interactions and available recommendations.

- **Path Discovery**: When a node needs to send data, it initiates a discovery process to identify potential routes.
- **Trust Evaluation**: Each path is evaluated using a composite trust score, aggregating the trustworthiness of all intermediate nodes.
- **Path Selection**: The path with the highest trust score is selected for data transmission.
- **Feedback and Learning**: After transmission, nodes monitor performance and adjust trust scores based on success, failure, or anomalies. This feedback loop ensures that the algorithm remains responsive to changes in node behavior and network topology.

In this paper, we basically implement different trusted paths to include the *divide et impera* strategy in the algorithm, in order to avoid the limitations explained in the previous sections.

4 Proposed Extension: *Divide et Impera* Trust Routing (DrATC+)

While the original DrATC algorithm provides a robust framework for selecting trusted paths in dynamic networks, it relies on a single-path routing strategy. This approach, although effective in many scenarios, presents limitations in terms of fault tolerance, load distribution, and resilience to malicious behavior. To address these challenges, we propose an extension to DrATC termed DrATC+ which integrates the classical *divide et impera* (divide and conquer) principle into trust-based routing.

4.1 Conceptual Overview and Design Principles

DrATC+ enhances the original model by dividing a message into multiple segments and routing each segment through a different, highly trusted path. This multipath strategy increases the robustness of data transmission by:

- **Mitigating single-point failures**: If one path is compromised or fails, only a portion of the message is affected.
- **Enhancing security**: Distributing message segments across paths reduces the risk of full message interception or tampering.
- **Improving load balancing**: Traffic is spread across multiple routes, reducing congestion and improving performance.
- **Adapting to trust dynamics**: Paths are selected based on real-time trust evaluations, allowing the system to respond to changing network conditions.

Moreover, the DrATC+ extension is built on the following key principles:

- **Trust-Weighted Path Selection**: Instead of selecting a single path with the highest trust score, DrATC+ identifies the top-N most trusted paths. Each path is evaluated using a composite trust score derived from the trustworthiness of its intermediate nodes.

- **Message Segmentation Strategy**: The message is divided into segments, which can be:
  - Equally sized, for simplicity.
  - Weighted by trust, where more trusted paths carry larger or more critical segments.
  - Context-aware, where sensitive data is routed through the most secure paths.
- **Redundancy and Reassembly**: Optional redundancy can be introduced to improve fault tolerance. At the destination, segments are reassembled in order, with mechanisms to detect and request retransmission of missing or corrupted parts.
- **Dynamic Trust Updates**: As in DrATC, trust scores are continuously updated based on feedback, allowing the system to adapt to malicious behavior or node failures.

4.2 Integration with Trust Characteristics

DrATC+ retains the flexibility of the original model by allowing selective inclusion of trust characteristics. For example:

- Direct and Indirect Trust: Used to evaluate known and unknown nodes.
- Transitive Trust: Enables trust inference across multi-hop paths.
- Context-Dependent Trust: Allows different trust thresholds for different types of data.
- Composite Trust: Combines attributes such as reliability, security, and competence into a single score.

This modularity ensures that DrATC+ can be tailored to specific application domains, such as IoT, vehicular networks, or critical infrastructure.

4.3 Updated Operational Workflow

Considering the extension proposed, we can design a new operational workflow. Thus, the DrATC+ routing process consists of the following steps:

- **Path Discovery**: Identify multiple candidate paths from source to destination.
- **Trust Evaluation**: Compute a composite trust score for each path.
- **Path Selection**: Select the top-N trusted paths based on trust thresholds or diversity criteria.
- **Message Segmentation**: Divide the message into segments and assign them to selected paths.
- **Transmission**: Send segments concurrently through their respective paths.
- **Reassembly and Verification**: Reconstruct the message at the destination, verifying integrity and completeness.
- **Feedback and Trust Update**: Update trust scores based on delivery success, latency, and anomalies.

This workflow ensures that routing decisions are both trust-aware and resilient to dynamic network conditions adding the DrATC+ rationale to the old algorithm. In the following section, we explain better this new design.

5 Algorithm Design

The DrATC+ algorithm builds upon the foundation of the original DrATC model by introducing a multipath, trust-aware routing mechanism that divides messages into segments and transmits them across multiple trusted paths. This section presents the design of the algorithm, detailing its inputs, core components, and operational flow.

5.1 Inputs and Assumptions

The algorithm assumes the following inputs:

- **Network Graph**: A representation of the network as a graph $G = (V, E)$ where V is the set of nodes and E is the set of edges (links).
- **Trust Matrix**: A matrix T where $Tij \in [0, 1]$ represents the trust value from node i to node j based on direct, indirect, or composite trust.
- **Source and Destination**: The origin node S and the target node D.
- **Message Payload**: The data to be transmitted, which will be segmented.
- **Trust Characteristics Configuration**: A set of enabled trust characteristics (e.g., direct, indirect, transitive, context-dependent).

5.2 Core Components

As explained in Section 4.3, the algorithm consists of the following core components which are here explored:

- **Path Discovery**
  - Identify all feasible paths from S to D using depth-limited or k-shortest path search.
  - Filter paths based on minimum trust thresholds or hop constraints.
- **Trust Evaluation**
  - For each path $P = \{v1, v2, ..., vn\}$, compute a composite trust score

$$T_P = \prod_{i=1}^{n-1} T_{v_i v_{i+1}} \tag{1}$$

  - Optionally, apply weights to emphasize certain trust characteristics (e.g., security, reliability).
- **Path Selection**
  - Select the top-N paths with the highest trust scores.
  - Optionally, enforce path diversity to avoid overlapping nodes or links.

- **Message Segmentation**
  - Divide the message into N segments considering different possibilities:
    - Uniform: Equal-sized segments.
    - Weighted: Segment size proportional to path trust score.
    - Redundant: Add parity or duplicate segments for fault tolerance.
- **Transmission**
  - Send each segment through its assigned path.
  - Include metadata for reassembly (e.g., segment ID, total segments, checksum).
- **Reassembly and Verification**
  - At the destination, reassemble the message using segment metadata.
  - Verify integrity using checksums or hash functions.
  - Request retransmission if segments are missing or corrupted.
- **Feedback and Trust Update**
  - Update trust scores based on:
    - Delivery success/failure.
    - Latency or delay.
    - Anomalies (e.g., tampering, packet loss).
  - Propagate feedback to neighboring nodes to refine indirect trust.

5.3 Pseudocode Overview

Here, in 1.1, we present a pseudocode describing how the extended algorith works.

Listing 1.1. Pseudocode Overview

```
function DrATCPlus(S, D, message, trust_matrix, config):
 paths = discover_paths(S, D, config.max_paths)
 scored_paths = []
 for path in paths:
     score = evaluate_trust(path, trust_matrix, config.trust_weights)
     if score >= config.min_trust_threshold:
         scored_paths.append((path, score))
 top_paths = select_top_paths(scored_paths, config.num_segments)
 segments = split_message(message, top_paths, config.split_strategy)
 for i in range(len(segments)):
     send_segment(top_paths[i], segments[i])
 received_segments = wait_for_segments(D, len(segments), timeout=config.timeout)
 if verify_segments(received_segments):
     return reassemble_message(received_segments)
 else:
     request_retransmission()
     return None
```

This pseudocode describes the logic of the DrATCPlus algorithm, which combines dynamic trust-based routing with a divide-and-conquer strategy. Initially, it discovers multiple candidate paths between the source and destination (line 2). Each path is then evaluated for trustworthiness using a customizable trust matrix and weight configuration (lines 3–5). Only the paths that meet a minimum trust threshold are retained, and from these, the top paths are selected based on trust scores (line 6). The original message is split into multiple segments according to a defined strategy (line 7), and each segment is transmitted

along a different trusted path (lines 8–9). The destination node waits to receive all segments within a given timeout (line 10), verifies their integrity (line 11), and reassembles the message if successful (line 12). If verification fails, a retransmission is requested (lines 13–14). This approach increases both robustness and security by distributing data across diverse and trusted routes.

6 Evaluation Scenarios

To validate the effectiveness of the DrATC+ algorithm, we present a set of evaluation scenarios that simulate different network conditions and trust dynamics. These scenarios are designed to highlight the benefits of integrating the *divide et impera* strategy into trust-based routing, particularly in terms of resilience, adaptability, and security.

Each scenario builds upon the foundational principles of the original DrATC model and introduces new challenges that are addressed through message segmentation and multipath routing.

6.1 Scenario 1: Dynamic Trust Updates with Feedback

This scenario introduces dynamic trust behavior by incorporating feedback mechanisms. Initially, the network operates similarly to Scenario 1, using three highly trusted paths. However, one of the nodes along one path begins to misbehave—delaying or dropping packets during transmission.

DrATC+ completes its first round of message delivery, but the destination detects performance issues associated with one segment. This triggers a feedback response, lowering the trust score of the problematic node. During the next execution, the trust evaluation phase excludes the now less-trusted node, and DrATC+ selects a new set of top paths that avoid the compromised route.

The scenario demonstrates the algorithm's ability to adapt to evolving conditions. Trust scores are updated in near real-time, allowing the system to reroute future transmissions dynamically. This adaptive behavior ensures ongoing message reliability and network resilience, even in the face of emerging issues.

6.2 Scenario 2: Malicious Node Detection and Route Reconfiguration

Here, the focus is on evaluating the security dimension of DrATC+ in a network containing a malicious node. The malicious node has earned a high trust score from historical behavior but begins selectively dropping or tampering with packets during transmission.

DrATC+ initially includes this node in one of its trusted paths. However, upon receipt, the destination detects anomalies—either missing segments or signs of tampering. Feedback mechanisms penalize the responsible node, reducing its trust score and triggering exclusion from future routing decisions.

This scenario illustrates how DrATC+ mitigates the impact of previously trusted but now untrustworthy nodes. It shows that trust is not static but continuously refined based on observed behavior. Segment-based routing combined with trust recalibration helps maintain message integrity and isolate threats, reinforcing the system's security guarantees.

7 Scenarios Extensions

With the previous section, we wanted to show how our extended algorithm can be applied to different scenarios. In this section, we explain more deeply such Scenarios. For the second, we will provide also a deeper example.

7.1 Scenario 1 extension

In this scenario, we explore how DrATC+ dynamically adapts to evolving trust conditions in a network where node reliability changes over time. The system begins in a relatively stable environment: trust scores are consistent, and the source node has access to three paths to the destination, all surpassing the configured minimum trust threshold.

Phase 1: Initial Transmission The algorithm performs its initial *discover_paths* and *evaluate_trust* routines to identify the top-3 most trusted paths.

The message is split into three equal segments using the configured *split_strategy* and each segment is transmitted independently via one of the selected paths.

Trust scores at this point reflect historical data, with all selected paths considered reliable.

Phase 2: Feedback Detection Upon receiving the segments, the destination node notices irregularities—e.g., one segment is delayed significantly or fails to arrive within the timeout period.

A verification routine detects incomplete or untimely delivery, triggering a trust feedback mechanism.

The node responsible for the affected segment is identified, and its trust score is reduced in the *trust_matrix*.

Phase 3: Adaptive Re-routing For the next transmission, *evaluate_trust* is executed again using the updated trust scores.

The previously problematic node now falls below the trust threshold and is excluded from future path selections.

DrATC+ selects a new trusted set of paths that avoids the compromised node.

Security Implications This scenario emphasizes DrATC+'s resilience and trust plasticity. By continuously refining trust based on transmission outcomes, the algorithm self-adapts to network dynamics, ensuring that unreliable behavior is penalized and excluded from future decisions. This adaptive feedback loop is crucial for real-time systems and environments where node behavior is non-deterministic, such as mobile ad hoc networks (MANETs), IoT ecosystems, or delay-tolerant networks.

7.2 Scenario 2 extension

This scenario examines the robustness of DrATC+ in the presence of a malicious node that intentionally attempts to disrupt secure message delivery. Unlike Scenario 2, the deviation in behavior here is not accidental or due to poor connectivity, but intentional and adversarial.

Phase 1: Trust Exploitation The malicious node has historically behaved well, thereby acquiring a high initial trust score in the trust matrix.

During path discovery, DrATC+ considers it a legitimate and reliable candidate and includes it in one of the selected paths.

Phase 2: Tampering Behavior During message transmission, the malicious node selectively drops packets, introduces delays, or even alters segment content.

At the destination, integrity checks fail—either because the segment is missing, corrupted, or fails cryptographic validation (e.g., hash mismatch).

A retransmission request is triggered for the affected segment.

Phase 3: Trust Recalibration The feedback mechanism lowers the trust value associated with the malicious node.

The *trust_ matrix* is updated dynamically, penalizing the misbehaving node.

In subsequent transmissions, the path that includes the compromised node is automatically excluded from selection.

If redundancy was configured (e.g., overlapping segments or parity bits), the system may still reassemble the original message successfully despite tampering.

Security Implications This scenario highlights the algorithm's ability to detect and isolate malicious behavior using both runtime feedback and historical interaction data. Importantly, it demonstrates:

- False trust detection: Nodes with high prior trust are not permanently trusted—behavior must remain consistent.
- Dynamic mitigation: Instead of relying on centralized intrusion detection, the routing mechanism itself filters out risky nodes over time.
- Segment-based resilience: Even if a single segment is lost or corrupted, redundancy or retransmission can preserve overall message integrity.

Proposed Use Cases for Scenario 3 This behavior is particularly valuable in untrusted, decentralized networks like sensor grids, peer-to-peer overlays, or battlefield communications, where pre-defining adversarial behavior is infeasible and on-the-fly response is essential.

7.3 Scenario 2: Detailed example

In this section, we provide a more detailed example for the second scenario. Starting from the network, it is represented in the following way:

- Nodes: A, B, C, D, E, F, G, H
- Source: A
- Destination: H
- Message: "HELLOTRUST"
- Split Strategy: Equal segments (3 segments)

In this case, we have an initial trust matrix, with trust scores between 0 and 1, which represents the paths with the trust scores shown in Table 2:

Path	**Nodes**	**Trust Score (avg)**
Path 1	A → B → E → H	0.90
Path 2	A → C → F → H	0.88
Path 3	A → D → G → H	0.87

Table 2. Trust Paths

We assume that the Min Trust Threshold is 0.85.

In our scenario the Malicious Node is G that will drop/corrupt the packet.

So we can consider two moments in time. Before and after the incident. Our trust matrix before the incident is presented in Table 3 where G behaves correctly initially.

('A','B'): 0.95	('B','E'): 0.90	('E','H'): 0.85
('A','C'): 0.90	('C','F'): 0.87	('F','H'): 0.88
('A','D'): 0.89	('D','G'): 0.90	('G','H'): 0.82

Table 3. Trust Matrix before the incident

Having the initial trust matrix, the algorithm can proceed to choose the paths and make the segmentation of the message.

Thus, in this case, DrATC+ selects 3 paths with highest average trust (above 0.85). For now all of them are trusted. So, the message "HELLOTRUST" is split into three segments:

1. "HEL"

2. "LOT"
3. "RUST"

This is represented by the pseudo-code and the defined split strategy is "equal" in Listing 1.2.

Listing 1.2. Path Selection Pseudo-code Overview

```
   # Pseudocode Fragment
paths = discover_paths(A, H, max_paths=3)
# paths = [[A,B,E,H], [A,C,F,H], [A,D,G,H]]

scored_paths = []
for path in paths:
    score = evaluate_trust(path, trust_matrix, weights)
    if score >= 0.85:
        scored_paths.append((path, score))

# top_paths = 3 paths as above
segments = split_message("HELLOTRUST", top_paths, split_strategy='equal')
```

Now, we assume that Node G behaves maliciously in Path 3 dropping the segment "RUST". In this case, the destination H receives the following pieces of information:

- From Path 1: "HEL"
- From Path 2: "LOT"
- From Path 3: None (missing or corrupted)

This is represented by the code in Listing 1.3.

Listing 1.3. Received segments pseudo-code

```
  received_segments = wait_for_segments(H, 3, timeout=5)
# received_segments = ["HEL", "LOT", None]
```

Thus, we assume that after receiving the message, integrity verification fails. So, after the analysis we can infer that G has acted maliciously. So, H sends trust feedback to penalize node G and we have a new trust matrix where in the path trust(G, H) trust drops to 0.32. This value can be adjusted according to the network administrator parameters. In our case, we set a penalization of 0.5 as presented by the pseudo-code shown in Listing 1.4.

Listing 1.4. Penalization pseudo-code

```
if not verify_segments(received_segments):
  update_trust(trust_matrix, culprit_node='G', delta=-0.5)
```

So, the new trust matrix (updated) will have modified the value representing the path from G to H: trust_matrix[('G', 'H')] = 0.32. We can see that now such path is now below the trust threshold (0.85). The trust matrix after the incident is presented in Table 4.

Thus, having the new trust matrix and considering the network modification, on the next transmission DrATC+ will exclude path A $\rightarrow$ D $\rightarrow$ G $\rightarrow$ H from the trusted ones. So, the algorithm will only consider as trusted Path 1 and 2 sending the message split into two parts.

('A','B'): 0.95	('B','E'): 0.90	('E','H'): 0.85
('A','C'): 0.90	('C','F'): 0.87	('F','H'): 0.88
('A','D'): 0.89	('D','G'): 0.90	('G','H'): 0.32

Table 4. Trust Matrix after the incident

8 Conclusion and Future Work

In this paper, we introduced DrATC+, an extension of the original DrATC trust-based routing algorithm, which integrates the *divide et impera* strategy to enhance the security, resilience, and adaptability of network communication. By segmenting messages and distributing them across multiple trusted paths, DrATC+ mitigates the risks associated with single-path routing, such as node failure, congestion, or malicious behavior.

We demonstrated how DrATC+ builds upon a rich set of trust characteristics—such as direct, indirect, transitive, and context-dependent trust—to dynamically evaluate and select the most reliable paths in a network. Through a series of evaluation scenarios and a real-world IoT use case, we showed that this approach not only improves fault tolerance and load balancing but also strengthens the overall trustworthiness of data transmission.

As future work, we plan to extend the algorithm to support adaptive message segmentation, allowing the division of data to respond dynamically to both the sensitivity of the message and the volatility of trust levels in the network. Additionally, we aim to integrate DrATC+ into existing routing protocols such as BGP or AODV, in order to assess its performance and scalability in large-scale, real-world network environments. Finally, we intend to develop a comprehensive simulation framework or testbed that will enable benchmarking DrATC+ against other trust-aware and multipath routing protocols, providing a robust basis for performance evaluation and comparison.

Acknowledgments

This work has been supported by the EU project HORIZON-MSCA-2021-SE-01 under grant agreement No 101086308 (DUCA).

This work reflects only the authors view and the Research Executive Agency is not responsible for any use that may be made of the information it contains.

References

1. Wafa Abdelghani, Corinne Amel Zayani, Ikram Amous, and Florence Sèdes. Trust management in social internet of things: a survey. In *Conference on e-Business, e-Services and e-Society*, pages 430–441. Springer, 2016.
2. Alfarez Abdul-Rahman and Stephen Hailes. Supporting trust in virtual communities. In *Proceedings of the 33rd annual Hawaii international conference on system sciences*, pages 9–pp. IEEE, 2000.

3. Thomas Beth, Malte Borcherding, and Birgit Klein. Valuation of trust in open networks. In *European Symposium on Research in Computer Security*, pages 1–18. Springer, 1994.
4. Junsheng Chang, Huaimin Wang, and Yin Gang. A dynamic trust metric for p2p systems. In *2006 Fifth International Conference on Grid and Cooperative Computing Workshops*, pages 117–120. IEEE, 2006.
5. Bruce Christianson and William S Harbison. Why isn't trust transitive? In *International workshop on security protocols*, pages 171–176. Springer, 1996.
6. Davide Ferraris and Lorenzo Monti. Dratc: Dynamic routing algorithm based on trust characteristics. In *International Workshop on Security and Trust Management*, pages 3–20. Springer, 2024.
7. Diego Gambetta et al. Can we trust trust. *Trust: Making and breaking cooperative relations*, 13:213–237, 2000.
8. Tyrone Grandison and Morris Sloman. A survey of trust in internet applications. *IEEE Communications Surveys & Tutorials*, 3(4):2–16, 2000.
9. Peter Kenning. The influence of general trust and specific trust on buying behaviour. *International Journal of Retail & Distribution Management*, 36(6):461–476, 2008.
10. Stephen Paul Marsh. *Formalising trust as a computational concept*. PhD thesis, Department of Computing Science and Mathematics, University of Stirling, 1994.
11. JL Morrow Jr, Mark H Hansen, and Allison W Pearson. The cognitive and affective antecedents of general trust within cooperative organizations. *Journal of managerial issues*, pages 48–64, 2004.
12. Francisco Moyano, Carmen Fernandez-Gago, and Javier Lopez. A conceptual framework for trust models. In *9th International Conference on Trust, Privacy and Security in Digital Business (TrustBus 2012*, volume 7449 of Lectures Notes in Computer Science, pages 93–104. Springer Verlag, Sep 2012.
13. Michalis Pavlidis. Designing for trust. In *CAiSE (Doctoral Consortium)*, pages 3–14, 2011.
14. Charles E Perkins. Ad hoc on-demand distance vector (aodv) routing, internet-draft. *draft-ietf-manet-aodv08. txt*, 2001.
15. Asad Amir Pirzada, Chris McDonald, et al. Establishing trust in pure ad-hoc networks. In *ACSC*, volume 4, page 1. Citeseer, 2004.
16. Charalambos Sergiou, Pavlos Antoniou, and Vasos Vassiliou. A comprehensive survey of congestion control protocols in wireless sensor networks. *IEEE Communications Surveys & Tutorials*, 16(4):1839–1859, 2014.
17. Yan Lindsay Sun, Zue Han, Wei Yu, and KJ Ray Liu. A trust evaluation framework in distributed networks: Vulnerability analysis and defense against attacks. In *Proceedings IEEE INFOCOM 2006. 25TH IEEE International Conference on Computer Communications*, pages 1–13. IEEE, 2006.
18. Zheng Yan and Silke Holtmanns. Trust modeling and management: from social trust to digital trust. *IGI Global*, pages 290–323, 2008.