Internet of Things 28 (2024) 101419

Contents lists available at ScienceDirect

Internet of Things

journal homepage: www.elsevier.com/locate/iot

Research article

ZFort: A scalable zero-trust approach for trust management and traffic engineering in SDN based IoTs

Usman Ashraf [a,*], Mohammed Al-Naeem [b], Muhammad Nasir Mumtaz Bhutta [c], Chau Yuen [d]

[a] School of Business, Torrens University, Sydney, 2010, NSW, Australia
[b] Department of Computer Networks and Communications, King Faisal University, P.O. 380, Hofuf, 31982, Alahsa, Saudi Arabia
[c] Abu Dhabi University, P.O. Box 15551, Al-ain, United Arab Emirates
[d] Nanyang Technological University, 50 Nanyang Avenue, Singapore, 639798, Singapore

ARTICLE INFO

Keywords:
IoT
Critical nodes
Zero trust

ABSTRACT

The Internet of Things (IoT), is a promising solution, but faces critical security challenges in the backdrop of evolving and sophisticated threats. Traditional security models are not well-adopted to protecting these diverse and resource-constrained devices against evolving threats like Advanced Persistent Threats (APTs). We introduce *ZFort*, a zero-trust framework that prioritizes the security of critical nodes in IoT networks. ZFort dynamically evaluates the risk status of nodes based on node's criticality and vulnerability scores derived from Common Vulnerabilities and Exposures (CVE) data ZFort dynamically assesses node risk based on criticality and vulnerability scores derived from Common Vulnerabilities and Exposures (CVE) data, and Common Vulnerability Scoring System (CVSS). ZFort uses a stochastic differential equation model for dynamic and continuous trust evaluation between nodes. Based on this evaluation, it dynamically adjusts security measures and routing decisions in real-time. Additionally, ZFort quickly isolates nodes that are likely compromised and prevents routing across them. ZFort uses Mixed Integer Linear Programming (MILP) and efficient heuristics, guaranteeing scalability and resource efficiency even in large networks and enhances the resilience and trustworthiness of key IoT infrastructure.

1. Introduction

The Internet of Things (IoT) has revolutionized several industries such as mechanized production, healthcare surveillance and smart cities among others. IoT systems are made up of a large number of inter-connected small devices including environmental data collecting sensors, mission-critical actuators and gateways that bridge the gap between physical and digital worlds. IoTs offer exceptional functionality and efficiency. However, this interconnected environment also creates a wide and vulnerable target area for possible attacks. Traditional security models were primarily designed for networks and need to be adapted to safeguard the heterogeneous and ever-changing IoT ecosystem especially in the face of evolving threats. Devices in these networks typically have restricted processing capabilities and insufficient security measuresDevices within these networks often possess limited processing power and inadequate security measures, making them attractive targets for attacks. Moreover, the fluctuating network topology

* Corresponding author.
E-mail addresses: usman.ashraf@torrens.edu.au (U. Ashraf), naeem@kfu.edu.sa (M. Al-Naeem), muhammad.bhutta@adu.ac.ae (M.N.M. Bhutta), chau.yuen@ntu.edu.sg (C. Yuen).

https://doi.org/10.1016/j.iot.2024.101419
Received 14 July 2024; Received in revised form 28 September 2024; Accepted 24 October 2024
Available online 29 October 2024

and the broad spectrum of possible attack vectors, which include basic denial-of-service attacks and advanced exploits, require a security approach that is flexible and responsive.

There are several significant challenges such as Advanced Persistent Threats (APTs). This is because they have the ability to infiltrate networks secretly and remain there for a while collecting data, manipulating it then releasing disruptive payloads. Due to the sophisticated nature of these threats, firewalls can sometimes fail to stop these attackers, which means that we need a security solution that can proactively prevent attacks. There is a significant need for an active, all-inclusive safety scheme.

In this paper, we propose and implement a novel method of zero-trust approach for reinforcing security of crucial nodes in IoT networks against these evolving threats. Software-Defined Networking (SDN) has become a key enabler for security and control over IoT networks as it provides a logical and programmable control plane. It enables the IoT networks to be re-configurable dynamically which is essential to deal with the changing security challenges that IoT systems present. This makes SDN based IoTs ideal for ZFort. ZFort leverages SDN's centralized control for orchestrating ZFort's zero-trust approach for improving security through active trust verification and effective dynamic traffic engineering. The centralized control of SDNs makes enforcement of security policies such as micro-segmentation and node isolation very easy even at a granular level for ZFort, making the security posture more effective against evolving threats.

Zero-trust provides a shift in the security mindset that requires continuous confirmation of trust within the network and adaptable reliance. All interactions are subject to evaluation and authorization before they lead to any serious harm. The following three principles underlie ZFort as the implementation of zero-trust philosophy:

- *Dynamic Zero-Trust Management*: Nodes in the ZFort network continuously evaluate the trust level that they can assign to their peers based on real-time interactions with them. The introduction of this dynamic verification capability allows ZFort to detect new threats and quickly block persistent adversaries.
- *Risk-Based Node Prioritization*: ZFort assigns risk scores to nodes based on their criticality and vulnerability by using industry-standard CVSS scores. This ensures that critical nodes are prioritized and that nodes that can be likely compromised are tagged as high-risk.
- *Strategic Micro-segmentation and Node Isolation*: By logically segmenting the network, ZFort isolates essential nodes and resources from those of lesser criticality. This segmentation strategy, rooted in the zero-trust model, includes the proactive isolation of nodes that are likely compromised. This prevents the ability of attackers to freely traverse the network and escalate their privileges.

ZFort combines mathematical modeling with practical heuristic techniques to achieve efficiency and scalability. We develop a Mixed Integer Linear Programming (MILP) model that optimizes the allocation of security controls strategically while adjusting trust levels dynamically. Simultaneously, this efficient heuristic algorithm has been customized for real-world applications in a wide variety of network settings. The algorithm provides near-optimal results and minimizes computational efforts considerably by leveraging a greedy heuristic approach. Extensive simulations involving different attack scenarios as well as network configurations show that ZFort is more effective in improving the overall security posture. These results indicate the capability of zero-trust architecture for enhancing security and resilience of critical IoT networks, thereby ensuring safety, confidentiality, accessibility and integrity of data assets. The rest of this paper is structured as follows: Section 2 discusses related work for IoT networks and contrasts it with the approach of ZFort, while highlighting the need for a modern, zero-trust approach. Section 3 introduces the prioritization and trust management framework of ZFort, including micro-segmentation and the motivation for selecting the trust metrics that ZFort incorporates. Section 4 introduces the network model and builds ZFort's Node Prioritization problem (NPP) formally. Section 5 shows the proof of NP-Hardness model and Section 6 shares two separate heuristic based greedy algorithms that can efficiently solve the NP-Hard problem. Section 7 shares the simulation results for a detailed evaluation of ZFort compared with existing solutions. Finally, Section 8 concludes the paper and discusses future research directions.

2. Related work

Securing critical nodes in IoT networks, remains a serious concern due to their vulnerability and central role in data communication and management. Our proposed approach ZFort offers a broad and comprehensive security framework to deal with issues like trust management, compromise detection and mitigation against emerging cyber threats. More specifically, for trust management, ZFort uses a zero-trust model with continual trust assessment to mitigate against emerging threats. By assigning risk scores based on node criticality and vulnerability based on industry-standard CVSS scores, ZFort achieves efficient compromise detection. ZFort complements this with micro-segmentation and node isolation to minimize the impact of likely compromised nodes. We now contrast our proposed approach to recent research efforts. Existing solutions for IoT security in this area can be broadly categorized as follows:

SDN-based Security Approaches for IoT Networks: SDN based security approaches have gained popularity recently for IoT networks [1–5] due to their centralized control capabilities which naturally offer excellent control over policy design and enforcement. In [1], authors introduce a real-time intrusion detection and mitigation system for SDN-based IoT approaches using ML (random forest classifiers) for traffic classification and threat detection. An interesting aspect of the paper is devoted to explainability of the rationale behind the system decisions, so that human experts can understand the decisioning process, thereby improving transparency. The proposed approach is effective in intrusion detection in an automated context, however, the limitation is that it is only restricted to threats that it can detect and neutralize; the threats that it knows. ZFort offers a broader security coverage as it employs continuous management of trust of users of the system, dynamically re-allocates risks and uses micro-segmentation to not merely find and suppress risk but to prevent its proliferation and actively control the level of trust that we place in each individual

node. In addition, ZFort implements CVSS based metrics for managing of vulnerabilities, which extends its capabilities beyond just threat detection to include vulnerability management.

In [2], authors propose a security architecture applicable to IoT SDN networks and utilizing Deep learning techniques for identification and prevention of Distributed Denial of Service (DDoS) attacks. The integration of SDN's features into the framework increases the efficiency of managing traffic in real time and boosts the detection of malicious DDoS attacks using deep learning models. While this solution is very effective in accurately detecting DDoS attack but its application is limited to only that threat vector. ZFort takes a broader approach that not only considers attack detection but also trust management and node prioritization. Through its continuous trust evaluation, micro-segmentation and node isolation, ZFort is effective beyond just DDoS attacks and can address attacks such as APTs.

In [3], a Secure Control and Data Plane Algorithm (SECOD) is proposed that creates a two-layered mechanism to prevent Distributed Denial of Service (DDoS) attacks on SDN-based IoT networks. SECOD protects both the data and control planes, which facilitates the rapid detection and remediation of DDoS attacks – a problem that is especially acute in IoT contexts. The framework however focuses primarily on DDoS and does not incorporate attributes of comprehensive trust management and traffic engineering that ZFort provides. ZFort is not limited to responding to a DDoS attack, but also continuously evaluates the trustworthiness of its nodes through micro-segmentation and real-time trust evaluation whereby any infected nodes are contained within the affected sub-network. In [4], a new architecture named "DistB-SDCloud" is proposed for Industrial Internet of Things (IIoT) by incorporating Blockchain (BC) and Software Defined Networking (SDN). By combining the two complementary approaches, the proposed architecture offers security, confidentiality and integrity of all data in a decentralized manner making it a viable option. This is a complementary solution to ZFort which can be incorporated with the broader ZFort approach. In [5], an interesting approach is proposed which uses the open source Snort IDS which is integrated with the SDN controller to monitor and protect the IoT network traffic against unwanted traffic in real-time. This approach helps to detect malicious activities including DDoS, IP spoofing, port scanning, etc. which enhance network security. ZFort's capabilities for continuous evaluation of the trust and dynamic traffic engineering enable it to manage the threats proactively.

The SDN based approaches discussed above and similar other approaches do offer security advantages and can be considered complementary to the focus of ZFort. ZFort offers a broader and proactive security coverage by employing continuous trust evaluation and management, micro-segmentation to isolate compromised nodes as well as uses industry-standard CVSS scoring system to efficiently handle vulnerability management. This offers a more holistic solution well-aligned with industry best-practices.

Network Resilience and Redundancy Techniques: There are a number of ways to implement resilience attributes into IoT systems. Broadly, these can be categorized into *operational* and *developmental*, wherein operational refers to application during the operation of the system, and developmental solutions focus on resilience-by-design [6]. A number of solutions [7–10] employ network resilience and redundancy techniques to solve the problem of failures in IoT networks. In [7], authors propose a new key distribution scheme with self-healing for secure communication in a community healthcare IoT network. Their solution is based on a two-layer system model with one layer responsible for managing a trust-based key distribution scheme and the bottom layer handles self-healing capability. Their solution does not directly addresses the continuous verification and least privilege control principles of zero-trust. Some solutions [8] propose installing additional nodes to replace faulty ones, improving network lifetime and reliability. However, they do not address the evolving nature of threats and the core ideas of a zero-trust approach. In [9], authors use re-configurable hardware to achieve self-healing. This technique is focused on breakdown of hardware at certain sensor nodes and the solution enables them to adapt the setup of repairs. But this solution does not generally address other types of security issues in IoT networks for example, those related to software issues and communication problems. Dynamic redundancy schemes are proposed by authors [10], to improve fault tolerance and to isolate compromised nodes, by using network virtualization. While it is an interesting solution, it focuses on establishing connections between virtual objects which represent real devices. This can become an issue if a virtual object or the cloud server hosting it fails, in which case, the communication between connected devices may end up broken.

The core focus of network resilience and redundancy solutions is to enhance the availability of networks and their fault tolerance by enabling continued functionality despite node malfunction or hardware failures. However, these mechanisms do not fully address the ever-changing complexity of cyber threats. ZFort offers a holistic solution that incorporates multiple aspects and adopts a more dynamic approach which is based on real-time trust assessment. This allows it to adapt to evolving cyber threats which mitigates novel attack vectors that traditional redundancy techniques may be incapable of handling.

Leveraging Authentication and Access Control Mechanisms: Lightweight authentication protocols [11–14] continue to evolve, with promising results. In [11] a three-tiered authentication scheme (IoT device, trust center at the edge, cloud service provider) is proposed, which focuses on low-resource IoT devices. Their solution addresses eavesdropping attacks, Denial of Service (DoS) attacks and other threats. In contrast to their solution, ZFort adopts a zero-trust security model that involves constant trust level evaluation, particularly for vital nodes (e.g., control units, gateways or devices that handle sensitive data).

In [12], authors propose a security-focused energy-efficient solution for key exchange agreement, authentication, and ownership transfer of patient data in IoT networks for e-health. Their solution provides lightweight operation for resource-constrained sensors. In contrast, our proposed ZFort's Zero-Trust model offers broader and dynamic security by continuously validating trustworthiness along with prioritizing critical nodes as well as micro-segmentation to route around compromised nodes. By using industry-standard CVSS scores, a more accurate risk allocation is achieved. In [13] a lightweight and secure mechanism of authenticating users is presented, which aims at enabling access to IoT sensor data in cloud. This is made possible through the use of cryptographic methods in conjunction with biometrics. In [14], the proposed solution addresses secure cross-domain authentication for resource-constrained devices. They achieve this by employing symmetric-polynomial based approach which is lightweight and offers efficient mutual authentication. An interesting solution is proposed in [15], in which a metric-based RPL Trustworthiness Scheme (MRTS) to secure

IoT networks. Their solution incorporates trust evaluation in the RPL routing protocol. This enables MRTS to significantly improving the routing security by avoiding malicious nodes and choosing the most trustworthy path based on different metrics like energy, honesty and selfishness. Additionally, they incorporate game-theoretic techniques to facilitate cooperation among nodes, resulting in improved network stability and packet delivery. In contrast, our proposed ZFort uses zero-trust model along with continual trust assessment and prioritizes critical nodes. Additionally, by using micro-segmentation and CVSS-based traffic engineering, ZFort offers a broader and more comprehensive security framework to deal with issues like trust management, compromise detection and mitigation against emerging cyber threats not specifically covered by MRTS.

While these are promising solutions to address the security challenges in IoT networks, ZFort's emphasis is more holistic and broader in nature. ZFort offers a solution that is more agile in the face of evolving threats by continuously evaluating levels of trust and prioritizing protection of key nodes. Additionally, the concept of micro-segmentation allows it to eliminate regions of network with very high probability of comprised nodes. Moreover, these solutions are complementary to ZFort and can be incorporated in ZFort to further strengthen security. These solutions would blend perfectly well with ZFort to secure data exchange within IoT environments.

Proactive and Zero-Trust Security Solutions: Zero-trust security is rapidly gaining popularity as a framework for securing critical infrastructure in IoT networks [16–18]. In [16], authors leverage blockchain for secure data sharing in IoT networks. The key contribution is to balance anonymity with other important factors including authentication, data privacy and fair incentives for participants. In [17], a novel Intrusion Detection System (IDS) is proposed for IoT networks which employs a Federated Learning (FL) approach. Federated learning is also becoming a significantly important solution for securing IoT networks. The proposed approach employs two complementary mechanisms: Convolutional Neural Networks (CNN) and Bidirectional Long-Term Short Memory (BiLSTM) in order to properly address the diverse facets of security attacks such as temporal and spatial. This approach emphasizes intrusion detection but ZFort offers a much broader security framework. This solution can be considered complementary to ZFort since incorporating intrusion detection with ZFort can provide further strengthening of the security posture of IoT networks. In [18], an interesting solution is proposed based on the use of an RFF (Radio Frequency Fingerprint) authentication method in combination with zero-trust architecture to enhance security. The approach eliminates traditional security models' limitations, namely over-dependence on trust zones and boundaries. It is a promising solution, but ZFort's capability of micro-segmentation, continuous trust verification and prioritizing critical nodes in conjunction with using industry-standard metrics provides a more comprehensive solutions. Generally, these solutions are complementary to ZFort and can be incorporated within the ZFort framework to further enhance security.

CVSS Scores for Trust: Recent contributions have explored integrating industry-standard CVE (Common Vulnerabilities and Exposures) [19] entries and CVSS (Common Vulnerability Scoring System) [20] into trust evaluation, specifically for IoT networks. In [21], authors have proposed an extension to the CVSS scoring system by proposing a new model to reduce data privacy issues in IoT networks. Their primary contribution is to incorporate factors such as Intervenability (TUI), Unlinkability and Transparency to ensure a more holistic scoring score for vulnerabilities. In [22], authors analyze the importance of CVSS for intelligent systems, who also propose a smart vulnerability quantification model for IoT that complies with CVSS v3.1. They have primarily based their solution on blockchain technology. In [23], authors compute risk levels based on CVE/CVSS not only for the nodes, but also for the attack path and the entire graph. Their primary contribution is proposing and applying a modified Depth First Algorithm (DFS) that finds all attack paths for source and target nodes, and filters out attack paths based on dominance level using computational metrics. In [24], an improved CVSS system is proposed for IoT. The key contribution is to evaluate the "defense effectiveness" of IoT networks. Authors propose a dynamic defense solution to enhance the security and "defense-ability" of IoT systems.

These solutions underscore the importance of leveraging CVSS scores in the context of dynamic trust-management as well as risk-assessment in IoT networks. ZFort builds on these concepts by employing CVSS-based traffic redistribution, prioritization of critical nodes as well as the key component of continuous trust evaluation to significantly enhance the overall security and reliability of IoT networks.

3. ZFort: Prioritization and trust management

As discussed above, ZFort uses a zero-trust model based on continuous trust evaluation along with risk assignment based on node criticality and vulnerability. Additionally, node prioritization along with node isolation plays an important role in enforcing micro-segmentation. Fig. 1 shows the overall architecture of ZFort and next, we will discuss key components of ZFort.

3.1. Node prioritization

A key component of ZFort is node prioritization which helps achieve a more effective allocation of security resources in IoT networks. Nodes are prioritized in ZFort based on a combination of node-criticality and vulnerability scores based on CVE and CVSS. The prioritization of nodes is based on the two keys factors: *Node Criticality* and *Vulnerability Score*. *Node Criticality* is a reflection of the relative importance of that node in the wider context of the overall network. Nodes are considered as critical if they have been delegated to perform important network functions such as acting as gateways. This concept can be further leveraged by network operators for other important nodes such as nodes involved in handling sensitive data. The second factor is *Vulnerability Score*, in which we utilize CVE (Common Vulnerabilities and Exposures) [19] entries and CVSS (Common Vulnerability Scoring System) [20] scores to evaluate the security risk as a consequence of vulnerabilities present in the node. By combining CVE and

Fig. 1. ZFort architecture.

CVSS, ZFort offers a widely known and industry-standard framework for enumerating vulnerabilities. Each node i is assigned a risk score $R(i)$ calculated as:

$$R(i) = \alpha \cdot C(i) + \beta \cdot S(i) \tag{1}$$

where $C(i)$ is a binary value indicating node criticality (1 for gateways, 0 otherwise), $S(i)$ is the highest CVSS score among the CVEs applicable to node i, α and β are weights that reflect the relative importance of each factor. These weights can be adjusted based on preferences and can vary from network to network. There is no one-size-fits-all solution for assigning weights to these metrics. Instead, we believe that these parameters offer network operators flexibility to fine-tune their systems based on empirical data or expert assessment.

Node risk scores are used to calculate node priority levels. These risk scores are integrated into the network model we build later on. To ensure algorithm convergence and minimize re-calculations due to minor changes, we plan to use a segmented approach where nodes are classified as belonging to three main priority classes: *high*, *medium* and *low*:

$$\text{High Priority} : R(i) > T_H \tag{2}$$

$$\text{Medium Priority} : T_M < R(i) \leq T_H \tag{3}$$

$$\text{Low Priority} : R(i) \leq T_M \tag{4}$$

where T_H and T_M are thresholds for High, Medium and Low priority, respectively. These thresholds can be pre-defined (which is our assumption in this work for simplicity, as this is complementary to the main thread of contribution) but can also be dynamic depending on the specific requirements of the network operator.

3.2. Trust management

In order to quantify node prioritization within the ZFort framework, we employ a priority function $P(i)$, which classifies nodes into High, Medium, and Low priority levels. This depends on the node role and vulnerability assessments. This priority function influences various aspects of security and operational protocols, particularly trust management.

The priority function $P(i)$ for each node i in the network is defined as:

$$P(i) = \begin{cases} 1 & \text{if node } i \text{ is High Priority} \\ 0.75 & \text{if node } i \text{ is Medium Priority} \\ 0.5 & \text{if node } i \text{ is Low Priority} \end{cases} \tag{5}$$

- $I_{ij}^{+}(t)$: A binary indicator that is 1 if a positive interaction is detected between nodes i and j at time t, and 0 otherwise.
- $I_{ij}^{-}(t)$: A binary indicator that is 1 if a negative interaction is detected between nodes i and j at time t, and 0 otherwise.

- δ_{ij}^{+}: Base trust increment value for interactions between nodes i and j.
- δ_{ij}^{-}: Base trust decrement value for interactions between nodes i and j.

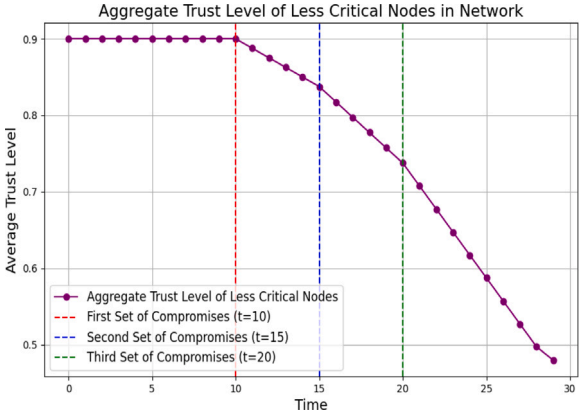


Fig. 2. Trust level evolution of less critical nodes.

There is a dynamic adjustment of trust levels with the network which is based on the interactions and node priority for the involved nodes:

$$t_{ij}(t+1) = \min\left\{t_{ij}(t) + \delta_{ij}^{+} \cdot I_{ij}^{+}(t) \cdot P(i), T_{max}\right\} \tag{6}$$

$$t_{ij}(t+1) = \max\left\{t_{ij}(t) - \delta_{ij}^{-} \cdot I_{ij}^{-}(t) \cdot P(i), T_{min}\right\} \tag{7}$$

ZFort offers a holistic solution by incorporating several real-world risk factors. It is noteworthy to point out that ZFort does not ignore less critical nodes. ZFort framework evaluates trust dynamics dynamically for nodes based on their current interactions and behavior patterns. This strategy makes it possible to monitor even less critical nodes and their trust and to update it if anomalies are observed. It should be noted that ZFort uses a simplified form of LACO [12] for authentication, which is used to establish the initial trust between nodes before ZFort's dynamic trust management takes over. In modern SDN-enabled IoT networks, ZFort can be complemented with task-offloading techniques including edge computing or mobile edge computing which allow for more resource-intensive tasks to be offloaded to more powerful devices including edge servers or cloud controllers. This can significantly reduce the processing burden on the resource-constrained IoT devices. ZFort's architecture can easily integrate with such offloading techniques, which can effectively ensure that resource constraints do not impair the system's ability to execute dynamic trust management efficiently.

3.3. Compromised node isolation

Another interesting contribution of ZFort is the dynamic isolation of nodes that are likely compromised. Nodes which have their trust levels fall below a certain threshold are flagged as compromised. This decision is based on the understanding that due to being high-risk, either these nodes are already compromised, are highly likely to be compromised in the near-future. This decision is based on historical interaction data and the current trust levels. When a node i is identified as compromised ($x_i = 1$), constraints are applied to isolate it from the network:

$$\sum_{k \in K} f_k^{i,j} \leq \psi(i,j) \cdot (1 - x_i) \quad \forall (i,j) \in \mathbb{F}(i),\ i,j \notin \{S,T\} \tag{8}$$

$$\sum_{k \in K} f_k^{i,j} \leq \psi(j,i) \cdot (1 - x_i) \quad \forall (j,i) \in \mathbb{R}(i),\ j,i \notin \{S,T\} \tag{9}$$

These constraints set the link capacities to zero for compromised nodes, effectively isolating them from the network and more importantly, maintaining the security and integrity of the IoT network. ZFort's node isolation mechanism encompasses to any node whose trust falls below a certain trust threshold, regardless of its criticality. To illustrate this point, we have share results from simulation of a simple 10 node IoT network with 3 gateways and 7 IoT nodes randomly distributed node topology and chosen the less critical nodes (non-gateway and at edges) and plotted their aggregate trust levels over time as tampering incidents occur. As Fig. 2 shows, the dynamic trust levels of even less critical nodes can fall below a certain threshold with their resulting isolation. This indicates that ZFort considers trust as a dynamic factor.

To illustrate the impact of node isolation and the resulting traffic redistribution, we share results from simulation of a simple 10 node IoT network with 2 gateways and 8 IoT nodes randomly distributed. Heat maps are generated for the risk levels (based on CVSS scores) for this topology with varying trust levels based on assigned CVSS scores. Fig. 3 shows the risk before and after traffic redistribution based on trust levels. This simple experiment shows that, ZFort employs a more conservative, risk-averse approach and redistributes traffic based on risk levels.

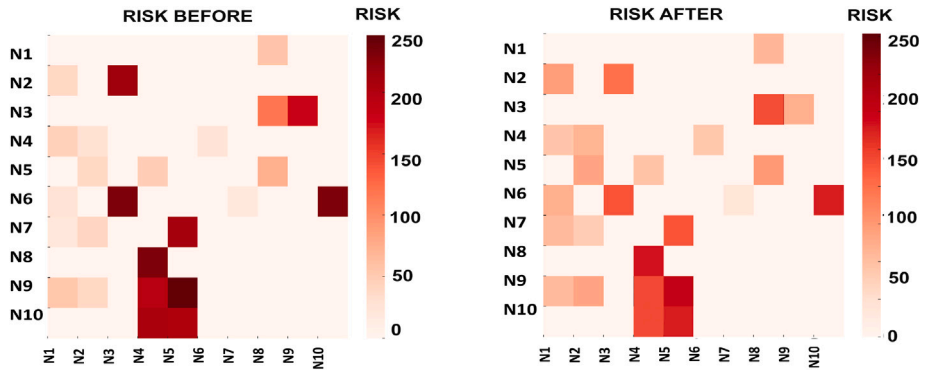


Fig. 3. Risk heat maps.

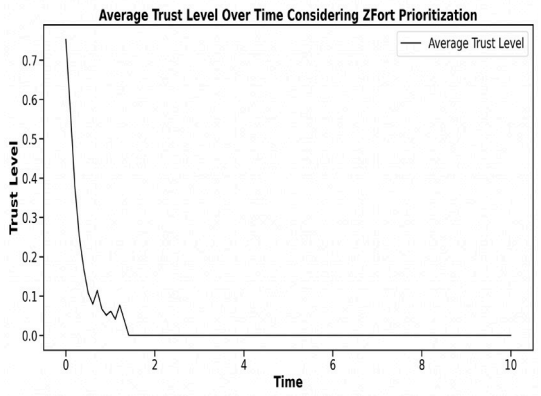


Fig. 4. Trust stability.

The ZFort model provides an all-inclusive approach to improve security on IoT networks by incorporating node prioritization, dynamic trust management and isolation of compromised nodes.

3.4. Network stability and trustworthiness

The stability of trust-levels over time is an important consideration from a performance perspective. To illustrate the impact of ZFort's node prioritization on the stability of trust levels over time, we model the trust dynamics using a differential equation and apply Lyapunov's stability theory. The rate of change of trust level $t_i(t)$ for node i considering positive and negative interactions, and cumulative risk is given by:

$$\frac{dt_i(t)}{dt} = \delta^+ I_i^+(t)P(i) - \delta^- I_i^-(t)P(i) - \gamma \int_0^t R(i,\tau)\, d\tau \tag{10}$$

To show stability, we define a Lyapunov function $V(t)$ as:

$$V(t) = \frac{1}{2}\sum_{i=1}^{N}(t_i(t) - \bar{t})^2 \tag{11}$$

where $\bar{t}$ is the average trust level. We calculate the derivative of $V(t)$ and substitute it in the trust dynamics equation, we have:

$$\frac{dV(t)}{dt} = \sum_{i=1}^{N}(t_i(t) - \bar{t})\left(\delta^+ I_i^+(t)P(i) - \delta^- I_i^-(t)P(i) - \gamma\int_0^t R(i,\tau)\, d\tau\right) \tag{12}$$

Which suggests that trust levels stabilize over time instead of having rapid fluctuations which can be counter-productive. In order to further drive this point home, we simulate the trust dynamics over time for each node (same topology as discussed above) considering ZFort's prioritization. As result suggest in Fig. 4, over time the trust levels eventually move to stabilization. These results suggest that ZFort's node prioritization and trust management can lead to stable trust levels while at the same time, reducing the overall risk. This analysis, while basic, underscores the effectiveness of ZFort's approach in enhancing the stability and trustworthiness of IoT networks.

3.5. Stochastic differential equations for trust dynamics

We employ a stochastic differential equation (SDE) framework for modeling the trust dynamics in the network. This framework aims at incorporating the drift (systematic patterns of trust change) as well as diffusion (random fluctuations caused by external influences and internal uncertainties).

3.5.1. Model formulation

The trust level of a node i towards node j at time t, denoted as $X_{ij}(t)$, is modeled using the following SDE:

$$dX_{ij}(t) = \mu_{ij}(X_{ij}(t), t)\,dt + \sigma_{ij}(X_{ij}(t), t)\,dW_t, \tag{13}$$

where:

- $\mu_{ij}(x,t)$ represents the drift coefficient which models the change of rate for trust caused by systematic factors including prior interactions as well as ongoing evaluations.
- $\sigma_{ij}(x,t)$ represents the diffusion coefficient which models trust volatility caused by network attacks for example.
- W_t is a standard Brownian motion which models stochastic trust evolution.

3.5.2. Drift and diffusion functions

We define the drift and diffusion functions as follows:

$$\mu_{ij}(x,t) = -\kappa(x - \bar{x}_{ij}) + f_{ij}(t), \tag{14}$$

$$\sigma_{ij}(x,t) = \gamma\sqrt{x(1-x)}, \tag{15}$$

where:

- κ is the rate at which the trust level is returned to its long-term mean $\bar{x}_{ij}$.
- $f_{ij}(t)$ models the external and internal trust adjustments, which can be based on several factors including recent interactions or changes in policies enforced across the network.
- γ restricts the randomness in trust changes meaning that the intensity of randomness is toned down and is reflective of uncertainty and risk based on node interactions.

3.5.3. Solution and analysis

This Stochastic Differential Equation (SDE) gives us the distribution of trust levels over a period of time, which can assist in predicting possible breaches or failures due to trust deterioration below a threshold. We can analyze the stability and long-term behavior of $X_{ij}(t)$ by examining the stationary distribution of the SDE, if it exists.

Additionally, we conduct a Fokker–Planck equation analysis which is derived from the stochastic differential equation. By doing so, we can obtain the density function of trust levels across the network, which would enable us to compute the probability of different trust use cases and their impacts on the security of the overall network.

$$\frac{\partial p}{\partial t} = -\frac{\partial}{\partial x}(\mu_{ij}(x,t)p) + \frac{1}{2}\frac{\partial^2}{\partial x^2}(\sigma_{ij}^2(x,t)p), \tag{16}$$

where $p = p(x,t)$ is the probability density function of $X_{ij}(t)$.

3.6. SDN in ZFort's zero-trust architecture

Software-Defined Networking (SDN) plays a key role in implementing and enforcing it is dynamic security approach. It effectively decouples the control plane from data plane and manages the network traffic in a centralized and programmable manner, which offers flexibility in security enforcement. By leveraging SDN's centralized control mechanism, ZFort dynamically re-configures traffic flows based on trust levels. When a node's trust level falls below a certain threshold, the SDN controller steps in and isolates the compromised node automatically by rerouting traffic, effectively preventing lateral movement by potential attackers. Moreover, due to SDNs ability for creating virtual network segments, ZFort has also enabled the use of micro-segmentation, reducing cross network attack potentials. This function together with the programmability of SDNs makes ZFort's zero trust system even more effective as changes will be done rapidly to threats that may affect the network thereby reducing chances of exposing untrusted actors within the network.

Table 1
ZFort mathematical model variables.

Variable	Description
N	Set of all nodes in the IoT network
A	Set of directed links connecting nodes in N
$\psi(i,j)$	Capacity of the directed link from node i to j
$g(X_{ij}(t))$	Function to scale capacity based on trust level
$\mathbb{F}(i)$	Set of outgoing links from node i
$\mathbb{R}(i)$	Set of incoming links to node i
S	Set of source nodes where data originates
T	Set of destination nodes where data is received
K	Set of different data flows between src-dest pairs
d_k	Demand (amount of data) for flow $k \in K$
B	Max backup paths network operator can install
$f_k^{i,j}$	Flow of data k on the link between nodes i and j
$t_{ij}(t)$	Trust level between nodes i and j at time t
x_i	Binary variable for if node i is compromised
λ	Penalty weight for low-trust flows in objective
$N_b(t)$	Number of backup paths used at time t
c_b	Cost per backup path
δ_{ij}^+	Trust increment for +ve interaction b/w i and j
δ_{ij}^-	Trust decrement for −ve interactions b/w i, j
$I_{ij}^+(t)$	Binary indicator for +ve interaction b/w i, j at t
$I_{ij}^-(t)$	Binary indicator for −ve interaction b/w i, j at t
$\mathrm{V}(X_{ij}(t))$	Variance of trust levels at time t
$v_{\max}$	Maximum trust variance

4. Network model

Table 1 summarizes the mathematical symbols used for modeling. The IoT network is modeled as a directed graph $G = (N, A)$ where $n \in N$ represents the set of $|N|$ IoT backbone nodes (IoT gateways, IoT relays, and Control Devices). Each directed arc $(i,j) \in A$ represents a directional link between nodes i and j, with $\psi(i,j)$ representing the capacity of the directional link (i,j). The function $\psi(i,j) \neq \psi(j,i)$ accounts for the asymmetric nature of communication channels such that capacity may be asymmetric in both directions.

Each node i has a set of outgoing links $\mathbb{F}(i) = \{(i,j) \in A : \mathrm{link}(i,j)\}$ is an outgoing link at i and a set of incoming links $\mathbb{R}(i) = \{(j,i) \in A : \mathrm{link}(j,i)\}$ is an incoming link to node i.

We assume a multi-flow IoT network with multiple flows between different source–destination pairs. Most traffic travels from end-devices across relays and gateways towards the cloud. Let $i \in S$ represent the set of source nodes, and $j \in T$ represent the set of destination nodes. The set $d_k \in D$ models the demand for each of the $k \in K$ flows in the network. The operator can install a maximum number of backup paths B to ensure robustness.

We now build on the stochastic modeling discussed above and aim at these dynamics into the network to dynamically mitigate as well as manage the risks. The network's routing protocol is modified so that it can leverage the real-time trust evaluations $X_{ij}(t)$, derived from the stochastic differential equations. Routing is basically done based on these trust levels which adapt to the fluctuations in trust across the network. For all $(i,j) \in A$, routing is enabled through (i,j) if $X_{ij}(t) \geq \theta_{ij}(t)$ where $\theta_{ij}(t)$ is an adaptable trust threshold that automatically adjusts based on network behaviors that have been observed as well as any potentially detected anomalies.

Network link capacities need to be adjusted so that they represent the latest trust assessments, such that lower trust levels yield restricted capacities to minimize risk exposure:

$$C_{ij}(t) = \psi(i,j) \cdot f(X_{ij}(t)), \tag{17}$$

with $f(X_{ij}(t))$ designed to represent a function that decreases with decreasing trust, indicating enhancement of security protocols and resulting in restricted data flow across links which are less reliable.

Stability of operations is ensured by bounding trust level variance within limits:

$$\mathrm{Var}(X_{ij}(t)) \leq v_{\max}, \tag{18}$$

which prevents systemic failures or security breaches due to high variability in trust. We formally define the ZFort model as follows: (see Table 1).

Objective:

$$\max \sum_{t=1}^{T} \sum_{i,j \in N} \sum_{k \in K} t_{ij}(t) f_k^{i,j} - \lambda \sum_{t=1}^{T} \sum_{i,j \in N} (1 - t_{ij}(t)) f_k^{i,j}$$

Subject to:

$$\sum_{(i,j)\in\mathbb{F}(i)} f_k^{i,j} - \sum_{(j,i)\in\mathbb{R}(i)} f_k^{j,i} = d_k \quad \forall i\in S,k\in K \tag{19}$$

$$\sum_{(i,j)\in\mathbb{F}(i)} f_k^{i,j} - \sum_{(j,i)\in\mathbb{R}(i)} f_k^{j,i} = 0 \quad \forall i\in N\setminus\{S,T\},k\in K \tag{20}$$

$$\sum_{(i,j)\in\mathbb{F}(i)} f_k^{i,j} - \sum_{(j,i)\in\mathbb{R}(i)} f_k^{i,j} = -d_k \quad \forall i\in T,k\in K \tag{21}$$

$$\sum_{k\in K} f_k^{i,j} \le \psi(i,j)\cdot g(X_{ij}(t)) \quad \forall (i,j)\in A \tag{22}$$

$$\sum_{k\in K} f_k^{i,j} \le \psi(i,j)(1-x_i) \quad \forall_{(i,j)} \quad \in\mathbb{F}(i)\ i,j\notin\{S,T\} \tag{23}$$

$$\sum_{k\in K} f_k^{i,j} \le \psi(j,i)(1-x_i) \quad \forall_{(j,i)\in\mathbb{R}(i)}\ j,i\notin\{S,T\} \tag{24}$$

$$\sum_{t=1}^{T} c_b * N_b(t) \le B \tag{25}$$

$$t_{ij}(t+1) = \min\{t_{ij}(t) + \delta_{ij}^{+}\cdot I_{ij}^{+}(t)\cdot P(i), T_{max}\} \tag{26}$$

$$t_{ij}(t+1) = \max\{t_{ij}(t) - \delta_{ij}^{-}\cdot I_{ij}^{-}(t)\cdot P(i), T_{min}\} \tag{27}$$

$$\mathrm{V}(X_{ij}(t)) \le \upsilon_{\max} \tag{28}$$

$$x_i \in \{0,1\} \quad \forall i\in N$$

$$f_k^{i,j}, \psi(i,j) \ge 0 \quad \forall_{(i,j)\in A}$$

Objective function: The objective function directly maximizes the total weighted flow ($t_{ij}(t) * f_{(k)}^{i,j}$) based on trust and penalizes low-trust flows with weight λ. The objective function is optimizing two goals in parallel: maximizing trust-weighted flow while simultaneously penalizing flow through low-trust links. The first term prioritizes data flow across connections which have a high level of trust. The second term, multiplied by a penalty weight factor λ, discourages flow through less-trusted links. The objective balances security and flow optimization.

Flow Conservation Constrains: The flow conservation constraints ensure that the amount of traffic entering a node is equal to the traffic leaving the node except for source and destination nodes which have them equal to d_k and $-d_k$ respectively.

Capacity Constraint: The capacity constraint limits the total flow on each directed link (i,j) to its capacity $\psi(i,j)$ and prevents link overloading.

Least-Privilege: Ensures that the trust level $t_{ij}(t)$ between a critical node i and node j is greater than or equal to the minimum trust that is required for accessing critical resources R_r. This constraint restricts flow through compromised nodes ($x_i = 1$) to reduce attack exposure. In order to eliminate compromised (or more specifically, likely compromised nodes) we set x_i to 1 for compromised nodes. This binary decision variable is used to reduce the capacity for compromised nodes by multiplying $(1 - x_i)$ with the link capacity $\psi(i,j)$. This achieves the effect of limiting the total flow $\sum f_k^{(i,j)}$ on affected links.

Budget Constraint: The budget constraint $\sum_{t=1}^{T}(c_b * N_b(t)) \le B$ reflects total cost over time T. In ZFort, backup paths are dynamically calculated to ensure that if the primary path becomes compromised, a backup path can be substituted in order to maintain the security of network. This cost models overheads on the controller such as additional processing or memory. This is a supplemental metric meant to give more refined budgeting capability to operators. The cost per backup path is represented by c_b. The symbol $N_b(t)$ represents the number of backup paths used at time t. If all communication flows are using primary paths without any failures, $N_b(t)$ would be zero. However, over time if the primary paths are compromised or are not available, the controller can activate backup paths, increasing the value of $N_b(t)$. The symbol B represents the total budget or resource constraint for security overhead.

For trust variable, $\upsilon_{\max}$ is the maximum allowable variance for trust levels. This constraint helps prevent excessive fluctuations in network behavior, which can lead to instabilities.

5. Proof of NP-hardness

In this section, we establish the NP-hardness of the node prioritization problem within the ZFort model for IoT networks. The problem is to determine an optimal prioritization of nodes such that the overall risk is minimized while maintaining operational requirements. Formally, this problem can be described as follows:

Definition 1 (*Node Prioritization Problem (NPP)*)**.** Given an IoT network modeled as a directed graph $G = (N, A)$ where N is the set of nodes and A is the set of directed links, along with a risk score $R(i)$ for each node $i \in N$, the goal is to assign a priority p_i to each node such that the sum of the products of the node's risk and its priority across all nodes is minimized:

$$\min \sum_{i\in N} p_i \cdot R(i) \tag{29}$$

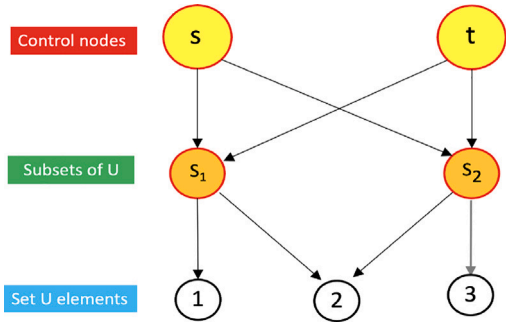


Fig. 5. Illustrative topology for reduction.

subject to operational constraints and the condition that p_i takes on a value from a fixed set of priorities.

Definition 2 (*Minimum Set Cover Problem (MSCP)*)**.** Given a universal set U and a collection of subsets $S = \{S_1, S_2, \ldots, S_m\}$ where each $S_i \subseteq U$, the objective is to find the smallest subset $C \subseteq S$ such that:

$$\bigcup_{S_i \in C} S_i = U \tag{30}$$

The goal is to minimize $|C|$, the number of subsets in the cover.

If a problem can be reduced to a known NP-Hard problem, then we can conclude that the problem is in fact, NP-Hard. For this purpose, we reduce from the Minimum Set Cover problem, which is known to be NP-hard. The reduction is as follows:

Theorem 1. *The Node Prioritization Problem (NPP) is NP-hard.*

Proof. Consider a specific instance of the Minimum Set Cover (MSC) problem, in which we have a universal set U and a subset collection represented by $S_1, S_2, \ldots, S_m$ of U. The goal is to select the minimal number of these subsets with the condition that the union of these subsets equals U.

We construct an instance of NPP by creating a node for each subset S_j and a node for each element in the universal set U. For each element $u \in U$, a risk score is assigned $R(u) = 1$. For each subset node S_j, we assign a risk score $R(S_j) = |U| + 1$ which ensures that prioritizing subset nodes is always less preferable unless it can lead to a solution which covers all the elements in U.

We connect an element node u to a subset node S_j if and only if $u \in S_j$ in the Set Cover instance. The problem now becomes that of prioritizing nodes so that we minimize the sum $\sum_{i \in N} p_i \cdot R(i)$ and prioritizing an element node corresponds to covering that element in the Set Cover problem.

Since the Minimum Set Cover problem is known to be NP-hard, and we have shown that any instance of it can be polynomially reduced to NPP, it follows that NPP is also NP-hard. □

To show how we can perform the reduction from the Minimum Set Cover Problem to the Node Prioritization Problem, we consider a simple topology shown in Fig. 5. This topology represents a basic instance of the Minimum Set Cover problem with three elements $\{1, 2, 3\}$ and two sets $\{S_1, S_2\}$, where: - S_1 covers elements 1 and 2, - S_2 covers elements 2 and 3. Nodes labeled 1, 2, and 3 represent elements of the universal set U. Nodes S_1 and S_2 represent the subsets of U. Arrows from S_1 and S_2 to elements depict the inclusion of elements within these subsets. The special nodes s and t could represent additional control nodes within a network model.

In order to ensure that all the elements are covered, requires giving priority to the nodes for S_1 and S_2. The sum of risk-product priorities is minimized by assigning risk scores and priorities in such a way that it becomes equivalent to finding the smallest cover of the universal set.

This example demonstrates that solving the NPP involves addressing the underlying Set Cover problem, thus proving that the NPP is NP-hard.

6. Greedy algorithms

This section presents two efficient greedy algorithms that leverage heuristics to find an near-optimal solution quickly. Greedy approaches are most popular for their computational efficiency, which is a major factor in IoT real-time applicability. However, it is noted that meta-heuristics (like Genetic Algorithms) or other exact approaches (Linear Programming) could be considered in future, especially when we are dealing with networks that do not require a time-efficient solutions are larger in size. For these situations, these alternate solutions could potentially perform well. However, one disadvantage of all these approaches is, although there is a

Algorithm 1 Greedy Algorithm 1 for ZFort Model

Initialize all flows $f_k^{i,j}$ to zero.
Initialize available capacity $AC_{i,j} = BW_{i,j}$ for all links (i, j).
Sort commodities $k \in K$ in descending order of trust thresholds T_k.
for each commodity k in sorted order **do**
    Find a shortest path P_k from source s_k to sink t_k with highest trust levels, considering available capacity and compromised nodes:
    - Prioritize links with higher trust levels $t_{ij}(t)$.
    - Exclude links connected to compromised nodes ($x_i = 1$).

    $f_{min} = \min\{AC_{i,j} : (i, j) \in P_k\}$ (Considering only usable links in P_k)
    for each link $(i, j) \in P_k$ **do**
        $f_k^{i,j} \leftarrow f_k^{i,j} + f_{min}$
        $AC_{i,j} \leftarrow AC_{i,j} - f_{min}$
    end for
end for

Algorithm 2 Greedy Algorithm 2 for ZFort Model

Initialize all flows $f_k^{i,j}$ to zero.
Initialize available capacity $AC_{i,j} = BW_{i,j}$ for all links (i, j).
Construct a trust-weighted network dynamically at each iteration:
- Define weights on edges (i, j) as $w_{ij}(t) = \alpha * t_{ij}(t) + (1 - \alpha) * (1 - x_i) * BW_{ij}$.
- The weight incorporates current trust level $t_{ij}(t)$ and excludes links from compromised nodes ($x_i = 1$).

for each commodity $k \in K$ **do**
    Find a **minimum-cost path** P_k from source s_k to sink t_k in the dynamically weighted network.
    $f_{min} = \min\{AC_{i,j} : (i, j) \in P_k\}$ (Considering only usable links in P_k)
    for each link $(i, j) \in P_k$ **do**
        $f_k^{i,j} \leftarrow f_k^{i,j} + f_{min}$
        $AC_{i,j} \leftarrow AC_{i,j} - f_{min}$
    end for
end for

potential to give a better global optimize solution, it could be too computationally intensive, which may limit their applicability to dynamic IoT networks.

Greedy algorithms are particularly well-known due to their computational efficiency, which is a ke3y requirement for real-time IoT environments. However, we note that alternative optimization methods such as meta-heuristics (e.g., Genetic Algorithms) or exact approaches (e.g., Linear Programming) could potentially be explored in future work, especially when we consider larger or less time-sensitive networks. One challenge with these approaches is that while they could potentially provide better global optimization, however, this may come at the cost of increased computational overhead, which may not be ideal for dynamic IoT scenarios.

6.1. Greedy Algorithm 1

The key idea of the first algorithm is trust and avoidance of compromised nodes. To initialize, each link has flows and available capacity. The algorithm scans through flows in order of decreasing trust thresholds. It selects the shortest path with highest trust levels among the available links and prefers links which have higher trust levels between nodes, based on $t_{ij}(t)$ and does not have any connection to compromised nodes ($x_i = 1$). It calculates the minimum available capacity on the chosen path f_min in order to allocate flows. The algorithm then assigns f_{min} to each link along the path and updates remainder capacity on those links as a result. The algorithm basically prioritizes high-trust paths and maximizes trust-weighted flow while avoiding compromised nodes.

6.2. Greedy Algorithm 2

The second algorithm takes a different approach, which leverages adaptive trust weights to find the optimal path for each flow in the network. Initialization is similar to the first algorithm and we set up flows and link capacities that are available. In each iteration, for each flow, the algorithm constructs a new weighted network such that link weights are calculated as a combination of trust level $t_{ij}(t)$ and link capacity BW_{ij}. Links that connect compromised nodes ($x_i = 1$) are assigned reduced weights by multiplying by $(1 - x_i)$. The algorithm then finds out the minimum-cost route in this dynamically weighted network. Trust and available capacity are the two factors that the cost models. For flow allocation and assignment, similar to Algorithm 1, f_{min} is calculated based on

Table 2
Simulation parameters and scenarios.

Parameter	Value(s)
Network topology	Randomly distributed
Node count	50–200
Trust model	Numerical (0–1)
Trust threshold	0.85
Traffic model	Random flows
Flow generation	Poisson process
SDN controller	OpenDaylight
Flow size	1–5 units
Algorithms	ZFort, GreedyAlgo1, GreedyAlgo2, MRTS
Incident type	Tampering (malicious nodes - high CVSS)
Incident count	1–10
Repetitions per scenario	5

the chosen path and flow is assigned to each link and then updates remainder available capacity. The weights are adjusted by the algorithm as trust changes or nodes are compromised with the result that the algorithm chooses a path that optimizes both trust and flow efficiency while at the same time, avoiding compromised nodes.

7. Performance evaluation

This is the performance evaluation section and Table 2 summarizes the simulation parameters and settings. Mininet was employed to emulate the network topology, install and control SDN controllers and monitor the network traffic flows. We used NetworkX to generate and manipulate the network topology and evaluate the shortest paths. This facilitated the implementation of the ZFort algorithms, which rely on dynamic trust management and node criticality. The solutions were implemented using Python. We evaluate five solutions: ZFort (the optimal model), GreedyAlgo1, GreedyAlgo2 and a simplified implementation of Minimum Risk Trust-based Secure routing (MRTS) algorithm [15] that avoids compromised nodes based on shortest path algorithms. Additionally, we also implemented Secure Control and Data Plane Algorithm (SECOD), a very relevant related solution that specifically enhances security of SDN based IoTs by mitigating DDoS attacks (see Table 2).

We use the following performance metrics to compare the performance:

- Flow Satisfaction: Percentage of successful data flows between source and destination nodes
- Trust Compliance: The degree to which the selected paths comply with the trust requirements.
- Average Path Trust: The average trust level of the selected paths.
- Runtime (s): The computational time required to compute the solution.

7.1. Varying compromised nodes

We first analyze the performance based on increasing number of tampering incidents, which we simulate by setting up CVSS score very high in an iteration for selected nodes. The aim is to study the traffic re-engineering if there are compromised nodes and how efficiently does the micro-segmentation and re-routing around malicious nodes works.

Flow Satisfaction: In terms of flow satisfaction, as Fig. 6(a) shows, ZFort starts high but declines steeply, reflecting that while it can optimize trust (seen from next figures), it may be less effective in ensuring all flows are satisfied as more nodes are compromised. Both the greedy algorithms indicate relatively high flow satisfaction Both greedy algorithms demonstrate relatively high flow satisfaction, achieving a balance between flow optimization and trust management, showing a balance between flow satisfaction and trust management. MRTS offers the highest flow satisfaction despite increasing compromised node instances, but at the cost of trust metrics. SECOD offers a reasonably well performance with it being second to only MRTS, and offers a balanced approach in terms of trust compliance.

Trust Compliance: As shown in Fig. 6(b), ZFort offers the highest trust compliance throughout which can be explained by the fact that it finds the optimal solution to the mathematical model. GreedyAlgo2 shows performance closely mimicking that of ZFort, but just shy of it, which is understandable as it finds approximate solutions to the model. GreedyAlgo2 however, shows a steady decline in trust compliance with increasing indicating it is less robust in maintaining trust. MRTS offers the lowest trust compliance, which is significantly decreased with increasingly compromised nodes. It seems to meet flow satisfaction well, but has limited applicability to contemporary risk metrics and is thus less robust. SECOD offers a decent performance indicating SECOD's effective monitoring of both planes in real time and reroute traffic dynamically in response to compromised nodes. SECOD outperforms MRTS and GreedyAlgo1 however, it falls short of ZFort and GreedyAlgo2 primarily. SECOD employs the same dynamic trust management model; however, the implementation mechanism is different as a distributed control plane is used which complicates the real time recalculation of trust metrics. Even though SECOD is quite effective in containing the risks of violation of trust due to isolation of compromised nodes, the distributed nature means that reconfiguration of paths can at times take longer, especially while scaling up to larger tampering events. On the other hand, GreedyAlgo2 (and ZFort), have a more efficient trust evaluation mechanism and also leverages CVSS scores to help it identify potential nodes with high probabilities of compromise.

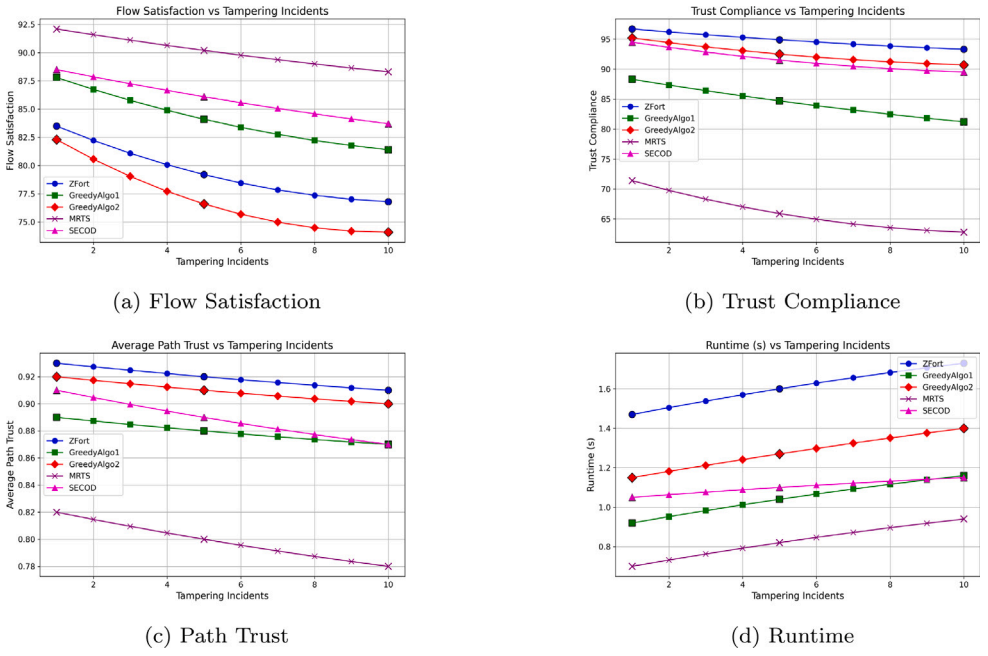


Fig. 6. Varying compromised nodes.

Average Path Trust: As Fig. 6(c) shows, in terms of average path trust, the results are very similar to trust compliance. In general, ZFort maintains the highest score due to optimal resolving of the model and employing techniques such as micro-segmentation, continuous trust evaluation and isolation of compromised nodes. The two greedy algorithms also follow the same performance pattern as they did for trust compliance and MRTS again offers the lowest performance on account of not incorporating these contemporary risk factors. SECOD's performance here follows the same pattern as that of trust compliance.

Runtime: For runtime metric, shown in Fig. 6(d), as expected, the optimal ZFort model consumes exponentially more time as the underlying optimization model becomes more complex, making the solution intractable for large problem instances. The other four solutions solve the routing and traffic-engineering problem within reasonably bounded time. In practice, the greedy algorithms would perform better as we scale the network.

7.2. Varying network size

We vary the network size from 50–200 in order to evaluate performance of the four solutions as the network scales.

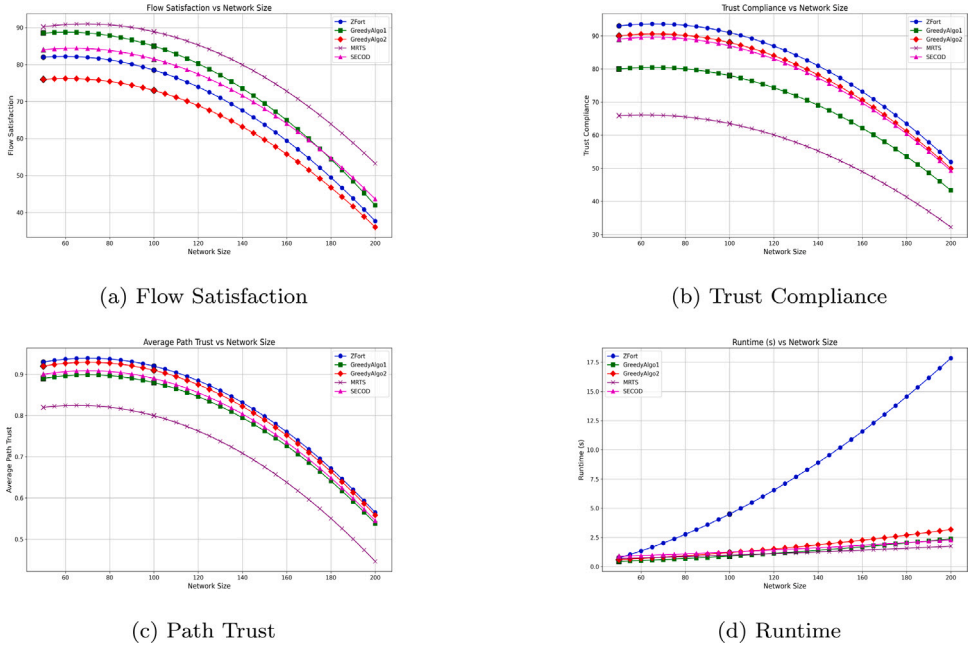
Flow Satisfaction: From Fig. 7(a) we see a similar pattern as before. ZFort starts high, but decreases as do all the other solutions. This can be attributed to longer paths and increased probability of encountering paths with failures. Overall, we see the same pattern of MRTS offering the highest flow satisfaction (at the cost of trust metrics) while ZFort and GreedyAlgo2 the lowest flow satisfaction, but with increased trust compliance. SECOD offers a moderate performance indicating an acceptable solution.

Trust Compliance: As illustrated in Fig. 7(b), ZFort offers the best trust compliance, offsetting the setback in overall flow satisfaction as we have increased trust metrics for the flow that are actually routed. GreedyAlgo2 offers a very close solution to ZFort's optimal results followed by SECOD. MRTS performs the lowest indicating that while it offers better overall flow satisfaction, but does not cater to several important risk factors, which can lead to a sub-optimal performance in terms of trust compliance. There is again a downward trend overall for all solutions due to scalability issues, longer paths and more probability of failures over longer paths.

Average Path Trust: Understandably, the average path trust performance as shown in Fig. 7(c), all solutions follow a very similar trend to that of trust compliance with ZFort offering the best solution due to optimal results at each iteration, closely followed by GreedyAlgo2, and then we have SECOD which offers a reasonable performance which is very close to GreedyAglo1 on account of its dual-layer approach, followed by MRTS which offers the worst performance.

Runtime: As Fig. 7(d) shows ZFort, the exact solution, requires exponentially longer time to resolve the optimal model as the network size increases, owing to significantly large number of constraints that are required to be solved to achieve the optimal solution. This indicates that greedy algorithms are a more practical solution when we consider network scalability. Overall, the four other solutions provide tractable runtime as the network scales.

Overall, from the results, we can conclude that ZFort is best for scenarios where trust and security are critical however, it is only feasible if we have small network size and a handful of compromise incidents. GreedyAlgo2 provides an excellent balance of trust and efficiency, providing the best results overall GreedyAlgo2 strikes an optimal balance between trust and efficiency, yielding



(a) Flow Satisfaction (b) Trust Compliance

(c) Path Trust (d) Runtime

Fig. 7. Varying network size.

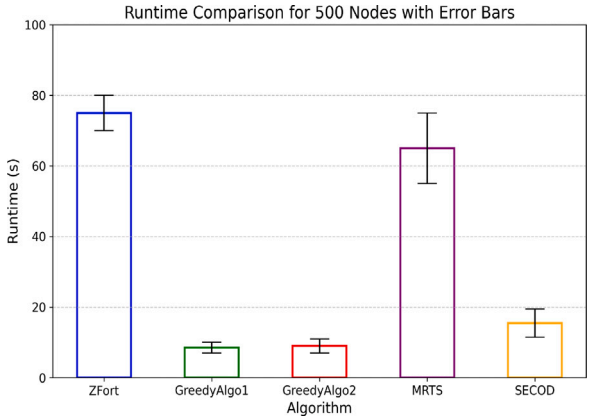


Fig. 8. Scalability of solutions.

the best overall results. GreedyAlgo1 is less effective in larger networks. MRTS is best if overall flow satisfaction and efficiency are critical MRTS is optimal when flow satisfaction and overall efficiency are prioritized. However, it sacrifices trust compliance and average path trust, making it less suitable for if security is crucial However, it compromises trust compliance and average path trust, making it less suitable when security is a top priority.

7.3. Scalability of solutions with large network size

In order to compare the time complexity of the five solutions, we simulate a separate scenario in which we use a topology comprising of 500 nodes. This would enable us to identify if the proposed approaches can be used for practical deployments. As shown in Fig. 8, we can see that the optimal ZFort model has an exponential time complexity, which is expected because MILPs are known to be NP-Hard to solve, which motivated us to design the greedy algorithms. We see that the greedy algorithms provide a reasonably bounded runtime despite a very large network comprising of 500 nodes. SECOD provides reasonable performance, however, it performs sub-par compared to the greedy algorithms which indicate the time efficiency of the greedy algorithms.

8. Conclusion

Security in IoT networks has become of paramount importance as they have become pervasive all around the world. Security in IoT networks has gained paramount importance due to their increasing pervasiveness across the globe. In this paper, we proposed ZFort, a zero-trust based traffic engineering solution for SDN based IoTs which utilizes a stochastic differential equation model for dynamic and continuous trust evaluation between nodes. By dynamically adjusting security measures and routing decisions in real-time as well as micro-segmentation and compromised node isolation, ZFort offers substantial performance benefits in terms of improving the overall security posture of the network. While we have provided extensive simulation results for a comparative performance comparison, we acknowledge that simulations can sometimes be limited and we believe it would be fruitful to evaluate the solution on a testbed. Therefore, for future, we are interested in implementing the two greedy algorithms on a small IoT testbed and see real-world performance. Additionally, by design, ZFort relies on the centralized controller for enforcing the security policies and is primarily targeted at SDN based IoTs. This is a limitation and we plan to investigate this as a future work. Another very interesting future direction for us is integrating counterfactual explanations for trust management [25]. Counterfactual explanations can significantly enhance trust in decision-making systems as they offer hypothetical situations to users that clarify how varying input conditions can influence the output. Integrating counterfactual examples in ZFort's trust management framework can offer significant room for improvement and accuracy.

CRediT authorship contribution statement

Usman Ashraf: Software, Methodology, Formal analysis, Data curation, Conceptualization. **Mohammed Al-Naeem:** Conceptualization. **Muhammad Nasir Mumtaz Bhutta:** Conceptualization. **Chau Yuen:** Methodology.

Declaration of competing interest

The authors declare that they have no known competing financial interests or personal relationships that could have appeared to influence the work reported in this paper.

Data availability

No data was used for the research described in the article.

References

[1] A.K. Sarica, P. Angin, Explainable security in SDN-based IoT networks, Sensors 20 (24) (2020) 7326.
[2] M. Cherian, S.L. Varma, Secure SDN–IoT framework for DDoS attack detection using deep learning and counter based approach, J. Netw. Syst. Manage. 31 (3) (2023) 54.
[3] S. Wang, K. Gomez, K. Sithamparanathan, M.R. Asghar, G. Russello, P. Zanna, Mitigating ddos attacks in sdn-based iot networks leveraging secure control and data plane algorithm, Appl. Sci. 11 (3) (2021) 929.
[4] A. Rahman, M.J. Islam, S.S. Band, G. Muhammad, K. Hasan, P. Tiwari, Towards a blockchain-SDN-based secure architecture for cloud computing in smart industrial IoT, Digit. Commun. Netw. 9 (2) (2023) 411–421.
[5] L. Elhaloui, M. Tabaa, S. Elfilali, E. Benlahmer, Dynamic security of IoT network traffic using SDN, Procedia Comput. Sci. 220 (2023) 356–363.
[6] A. Avizienis, J.-C. Laprie, B. Randell, C. Landwehr, Basic concepts and taxonomy of dependable and secure computing, IEEE Trans. Depend. Secur. Comput. 1 (1) (2004) 11–33.
[7] S. Han, M. Gu, B. Yang, J. Lin, H. Hong, M. Kong, A secure trust-based key distribution with self-healing for internet of things, IEEE Access 7 (2019) 114060–114076.
[8] U. Ashraf, PROSE: proactive resilience in internet of things: targeted attacks and countermeasures, IEEE Sens. J. 18 (24) (2018) 10049–10057.
[9] S. Yuan, L. Qiu, S. Gao, Y. Tong, W. Yang, Providing self-healing ability for wireless sensor node by using reconfigurable hardware, Sensors 12 (11) (2012) 14570–14591.
[10] I. Ullah, S. Ahmad, F. Mehmood, D. Kim, Cloud based IoT network virtualization for supporting dynamic connectivity among connected devices, Electronics 8 (7) (2019) 742.
[11] A. Shahidinejad, M. Ghobaei-Arani, A. Souri, M. Shojafar, S. Kumari, Light-edge: A lightweight authentication protocol for IoT devices in an edge-cloud environment, IEEE Consum. Electron. Mag. 11 (2) (2021) 57–63.
[12] S.F. Aghili, H. Mala, M. Shojafar, P. Peris-Lopez, LACO: Lightweight three-factor authentication, access control and ownership transfer scheme for e-health systems in IoT, Future Gen. Comput. Syst. 96 (2019) 410–424.
[13] M. Wazid, A.K. Das, V. Bhat, A.V. Vasilakos, LAM-CIoT: Lightweight authentication mechanism in cloud-based IoT environment, J. Netw. Comput. Appl. 150 (2020) 102496.
[14] B. Gong, G. Zheng, M. Waqas, S. Tu, S. Chen, LCDMA: Lightweight cross-domain mutual identity authentication scheme for internet of things, IEEE Internet Things J. (2023).
[15] N. Djedjig, D. Tandjaoui, F. Medjek, I. Romdhani, Trust-aware and cooperative routing protocol for IoT security, J. Inf. Secur. Appl. 52 (2020) 102467.
[16] Y. Liu, X. Hao, W. Ren, R. Xiong, T. Zhu, K.-K.R. Choo, G. Min, A blockchain-based decentralized, fair and authenticated information sharing scheme in zero trust internet-of-things, IEEE Trans. Comput. 72 (2) (2022) 501–512.
[17] D. Javeed, M.S. Saeed, M. Adil, P. Kumar, A. Jolfaei, A federated learning-based zero trust intrusion detection system for internet of things, Ad Hoc Netw. (2024) 103540.
[18] W. Jing, L. Peng, H. Fu, A. Hu, An authentication mechanism based on zero trust with radio frequency fingerprint for internet of things networks, IEEE Internet Things J. (2024).
[19] N.I. of Standards, T. (NIST), Common Vulnerabilities and Exposures (CVE), National Vulnerability Database (NVD), URL https://nvd.nist.gov/vuln-management/cve.

[20] F.F. for Incident Response, S.T. (FIRST), Common Vulnerability Scoring System (CVSS), National Vulnerability Database (NVD), URL https://www.first.org/cvss/.
[21] S. Rizvi, I. Williams, S. Campbell, TUI model for data privacy assessment in IoT networks, Internet Things 17 (2022) 100465.
[22] P. Anand, Y. Singh, A. Selwal, P.K. Singh, K.Z. Ghafoor, IVQFIoT: An intelligent vulnerability quantification framework for scoring internet of things vulnerabilities, Expert Syst. 39 (5) (2022) e12829.
[23] F. Arat, S. Akleylek, A new method for vulnerability and risk assessment of IoT, Comput. Netw. 237 (2023) 110046.
[24] W. Han, X. Liu, H. Zhang, R. Quan, L. Shen, Dynamically-enabled defense effectiveness evaluation of IoT based on vulnerability analysis, in: Proceedings of the 3rd International Conference on Multimedia Systems and Signal Processing, 2018, pp. 99–103.
[25] J. Del Ser, A. Barredo-Arrieta, N. Díaz-Rodríguez, F. Herrera, A. Saranti, A. Holzinger, On generating trustworthy counterfactual explanations, Inform. Sci. 655 (2024) 119898.