mMLSnet: Multilevel Security Network with Mobility

Mingli Yu, Quinn K. Burke, Thomas La Porta and Patrick McDaniel

***Abstract*—Multilevel security (MLS) network systems regulate data access and impose security policies to prevent unauthorized data disclosure among users of different trust domains. Current MLS network systems are generally based on a static security label configuration and fail to adapt to the dynamic network events such as device movement. In this paper, we introduce $mMLSnet$, a new MLS network model that can efficiently relabel the network while enforcing security policies. A simple mobility protocol is proposed to support routing of mobile devices. Our evaluation shows that our model can calculate a new flow path and reroute the flows of mobile devices quickly at a modest cost.**

***Index Terms*—Software-defined networking, mobility, multi-level security**

I. Introduction

In computer networks, Multilevel Security (MLS) systems implement policies to prevent unauthorized data disclosure between network service points that produce/consume data [1] and isolate network traffic among different cloud tenants which belongs to different trust domains [2], [3]. MLS in Software Defined Networks (SDN) has been proposed for static [4] and dynamic [5] networks. These works consider changes in topology due to link failures, and the entry and exit of new flows into networks, but do not directly address node mobility. In this paper we introduce mobile MLSNet (mMLSNet) to support mobile devices with differing security requirements and show its performance is superior to the baseline MLSNet system.

MLS systems regulate data access by employing a reference monitor, which oversees requests to access data from different sources. This reference monitor utilizes security labels and a specified security protocol to establish allowed and disallowed data transfers between various sources and endpoints, such as between database servers and their clients [6]. A security level is an ordered hierarchical attribute that indicates the relative authorization sensitivity of an object, for example, the public label is less sensitive than the secret label. The ordering describes the necessary access constraints to maintain confidentiality.

Mingli Yu and Thomas La Porta are with the Pennsylvania State University, University Park, PA, USA. Email: {mxy309, tfl12}@psu.edu. Quinn K. Burke and Patrick McDaniel are with the University of Wisconsin–Madison, Madison, Wisconsin, USA. Email: {qkb, mcdaniel}@cs.wisc.edu.

This research was partially sponsored by the U.S. Army Combat Capabilities Development Command Army Research Laboratory and was accomplished under Cooperative Agreement Number W911NF-13-2-0045 (ARL Cyber Security CRA).

The main difference of MLS and traditional networks is that MLS networks compute an end-to-end routing path for each flow while in the traditional networks the path is found distributively with BGP or OSPF. Meanwhile, security constraints must be satisfied along the entire path to prevent exposure to intermediate subjects like an Ethernet switch. The implementation of MLS networks are substantially simplified by Software-defined networks (SDN) [4]. By separating the control plane and data plane, SDN centralizes network management and allows MLS systems to run as controller applications with complete visibility over the network status and the capability to compute and install secure flow rules.

However, the challenge of enforcing MLS polices in networks is that the fixed configuration of security labels in existing MLS systems is incapable of handling dynamic network events such as topology changes and device movement and may fail to route flows for the mobile users.

We address the challenge by introducing $mMLSnet$, an SDN controller application that routes flows securely following MLS policies and dynamically changes the security labels of switches and access points. The flexibility of security label configuration allows the network to adapt to the unstable traffic and support routing for mobile users. We develop a mobility protocol and formulate the problem as a shortest path search while following MLS security policies. We then introduce an algorithm to find the shortest path which is quick and efficient.

A. Related Work

Multilevel Security Networks: Lu et al. [7] were the first to introduce a MLS system for networks, which statically assigned security labels to network switches but unwieldly required each network endpoint to install specialized software. MLS networks has been also used in network clouds to segregate network traffic among different tenants [1]. More recently [4], [8] leverage SDN to enforce MLS policies without requiring specialized software. Instead, controller applications are launched to logically assign and maintain security labels for each network node while enforce the security policy via flow rules. While these approaches leverage similar MLS techniques as $mMLSnet$, they are limited in that they assume static network behavior and may have a low utilization rates of network resources in face of unstable network traffic.

More closely related to $mMLSnet$, $MLSnet$ [5] assigns to network endpoints a security label and periodically enables the security label adjustment to respond to dynamic network events such as topology change, link failure and burst traf-

fic. However, $MLSnet$ can't respond quickly to the device movement because of the long relabeling interval. While $MLSnet$ provides a global relabeling framework to achieve maximum flow coverage of various security domains, we focus on searching an available flow path for a single mobile node.

Confidentiality Protection: The prevalent methods of confidentiality protection and traffic isolation, including perimeter firewalls, encryption, and VLAN, only mitigate the issue of confidentiality to some extent and do not adapt well to changes in network dynamics. Configuring firewalls can be a complex and mistake-prone task [9]. Moreover, encryption itself is susceptible to traffic analysis through side channels [10]. Routing methods such as VLANs provide a modest degree of separation by assigning an individual VLAN identifier to each user group. VLANs often suffer from limited granularity of policy and complex configuration [11]. For example, many campus network faces the problem that only a limited number of hosts are supported per VLAN [12]. Conversely, $mMLSnet$ achieves a comparable level of segregation without the requirement of tag and port management, as it verifies access control conditions during the establishment of flow rules.

Policy based Routing: There exists a substantial amount of research focused on the implementation, validation, and reconciliation of network policies based on SDN [13]–[16]. This includes the introduction of constructions and specification languages that assess reachability, ensure loop-free forwarding, and enforce network-level access controls (ACLs) on a per-service and per-user-identity basis, among other network invariants. However, many of these systems exhibit a significant limitation: they do not adapt well to dynamic network events. These policies are either preset according to the user identity or service or neglect the security of intermediate nodes within the shared network infrastructure.

Mobility Management: Mobile IP [17] is the most well-known solution among existing network-layer macro-mobility solutions in mobile Internet. Multiple variants of Mobile IP are proposed such as Mobile IP Regional Registration [18] and Hierarchical Mobile IP [19]. We borrow the design idea of Mobile IP to design our mobility protocol. On the other hand, HAWAII [20] provides a micro-mobility solution where the mobility is restricted in a subnet domain, which is similar to the mobility in the MLS network scenario.

B. Summary of Contributions

- We provide a mobility protocol to support mobile nodes in MLS networks.
- We formulate dynamic relabeling and re-routing for a single mobile node as a shortest path problem and develop a path finding algorithm that incurs a short running time and has good scalability.
- We demonstrate the necessary properties of a MLS network topology to minimize the disruption on the traffic of lower security groups while supporting mobility.

Roadmap. Section II introduces our models, problem formulation and the mobility protocol. Section III presents the routing path finding algorithm for a mobile node. Section IV validate the proposed mobility solution. Section V discusses the network topology design for mobility in MLS networks. Section VI concludes the paper.

II. Problem Formulation

A. Network Model

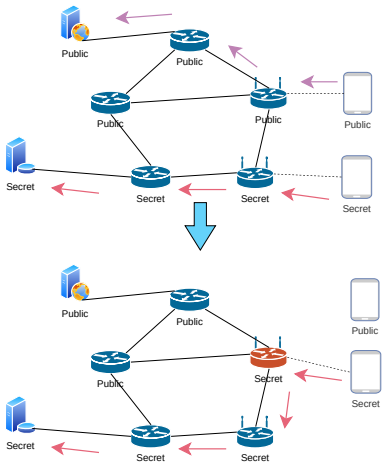

Fig. 1. Network scenario describing how relabeling operates.

We model the network as a graph $G(V, E)$ where the node set V contains mobile device, access points, switches and servers and the link set E includes wired and wireless links. An SDN controller monitors the whole network and assigns each node v a security label l_v. The upper part of Fig. 1 shows a network scenario with two security levels–public and secret. Any network flow f is also assigned a security label l_f that equals to the security label of its source node. In the upper part of Fig. 1 a public host is sending a public flow (purple) toward a public server while the a secret host is sending a secret flow (pink) to a secret server. We assume the security labels are assigned properly in advance based on a relative security assessment. In fact, the security labels of the switches and access points can be initialized randomly as they may be changed later.

We enforce a strict security policy which maintains the absolute confidentiality of network flows and the total isolation among different security level groups. Specifically, a flow of l_f can only traverse the nodes v such that $l_v = l_f$, which is shown in Fig. 1. Recall that in SDN, packets not matching existing rules in the flow table will typically be forwarded to the controller for further processing. Therefore, $mMLSnet$ can run as a controller application that computes flow rules that meet the policy. When receiving a new flow, $mMLSnet$ will: (1) identify the security label of the flow based on the source node (2) confirm the destination node has the security label of the flow; and (3) potentially relabel some switches or access points to build a routing path. Here we leverage the SDN property of centralized management such that the controller maintains the up-to-date network status and topology. This

enables the controller to assign security labels without hosts or servers being aware.

B. Relabeling

Device movements are common in wireless networks and relabeling is an essential function that enables multilevel security networks to support mobile devices. Fig. 1 shows that $mMLSnet$ can dynamically relabel the access points and switches. In this example, a secret device moves out of the range of a secret access point and into the range of a public access point. Then $mMLSnet$ relabels and reboots the public access point as a secret access point so that the secret host can have access to the network. Meanwhile, the public wireless devices that attached the relabelled access point are blocked. Note that only the public hosts and servers are blocked. The links between switches, routers and access points remain untouched.

Relabeling allows timely reaction to dynamic network events such as link failure or device movement. In practice, relabeling can be deployed both periodically and re-actively. Periodic relabeling optimizes the flow coverage and switch or link utilization rate for all the security groups but is time-consuming. Therefore, it is normally scheduled with a long time interval [5]. On the contrary, active relabeling that only searches and affects a small fraction of the network can respond to the urgent network event such as the device movement immediately.

C. Mobility Support Design

We follow the design principles of Mobile IP [17] to design multilevel security network mobility protocols. In the Mobile IP framework, packets destined to a mobile device currently resident in a foreign network are first forwarded to the home network of the device before the home network agent tunnels the packets to the foreign network agent. However, in multilevel security networks, the tunnel between the home agent, that is the previous access point the mobile device was attached, and the foreign agent, that is the current access point the mobile device is attached, may not be able to be set up directly because of the potential violation of the security policy. For example, the current access point may be public but the mobile device is secret. This implies under some circumstances the ongoing TCP sessions of mobile devices may be disrupted.

This challenge can be addressed by relabeling the current access point to match the previous one and then finding a shortest and secure path between them. The controller will install forwarding flow rules on the switches along the path. In this way, we formulate the mobility problem as a shortest path problem between two nodes. In summary, the mobile protocol is: (1) the mobile node disassociates with the previous access point and associates with the new access point (2) the new access point informs the controller about the arrival of the mobile node (3) the controller calculates the shortest "secure" path P between the two access points and possibly relabels the access points along the path (4) the controller installs such flow rules that the packets destined to the mobile node via the previous access point are now forwarded along the path P to the new access point. (5) the mobile node now uses the indirect routing to resume the ongoing TCP sessions.

Another possible solution [5] is to recalculate all the flow paths that are sourced from or destined to the mobile node. The main difference between our approach and this method is that we leverage and extend the existing flow path to accommodate the mobility at the cost of potentially assigning a higher security level on some switches. In fact, our approach can apply incrementally to the multilevel security network in [5]. While global relabeling is scheduled periodically to achieve maximum flow coverage for all the security groups, our method enables the network to react more immediately to mobile users so that the mobile user can get network access in tens of seconds instead of waiting hundreds of seconds which is the typical relabeling time interval.

For simplicity but without loss of generality, we study a network with two security levels: public and secret. We assume that the mobile user stays in the reach of the multilevel security network. We also assume there are only $X\%$ of the secret nodes, where $0 \leq X \leq 20$. This means that most nodes are public and the solution to the mobile problem of a public user are well studied [20]. Therefore, we only consider the cases when the mobile user is secret.

III. Relabeling Algorithm

In this section we illustrate the relabeling process and propose a modified Dijkstra's algorithm shown in Alg. 1 to find the shortest path. The relabeling and re-routing procedure includes three steps: (1) finding shortest flow paths and recording conflicts on the paths (2) relabeling the switches and access points of conflicts (3) rebooting and resetting the switches and access points of conflicts and installing the new flow rules.

We adopt an early stop trick in line 8 of Alg. 1. While the normal Dijkstra's algorithm searches for the short distances from the mobile node to all the other nodes, our algorithm stops when the shortest path to the previous access point is found. The algorithm returns a "shortest" path between the previous access point and the current access point. Here we define the link cost as the security level difference to minimize the number of access points or switches to be relabeled. This not only reduces the disruption to the public network traffic, but also diminishes the waiting time for the switch and access point reboot. Then in the second step, we can traverse the path and relabel the switches and access points with a public security level along the path to the secret security level.

Now we analyze the time complexity of the relabeling algorithms. First, the shortest-path algorithm and the path tracing is executed once for each mobile node. Then the time complexity is: $O\left((|V| + |E|) \log |V|\right)$. The time complexity of the second and third step can be upper-bounded by $|V|$. Therefore, the running time for this step is: $O\left((|V| + |E|) \log |V|\right)$.

Algorithm 1 Path find algorithm

Input current access point s, previous access point d, graph G=(V,E)
Variable distance array d, previous hop array p, priority queue q
Output shortest path ans

$d[s] = 0, p[s] = null, q.add(0, s)$
for $v \in V \setminus s$ **do**
    $d[v] = \infty$, $p[v] = null$
end for
while $queue$ not empty **do**
    u = node in q with smallest $d[\cdot]$
    $d[u], u = q.pop()$
    if $u == d$ **then**
        while $p[u]! = null$ **do**
            $ans.add[u]$
            $u = p[u]$
        end while
        return ans
    end if
    for each neighbor v of u **do**
        $w = abs(j.label - v.label)$
        if $d[u] + w < d[v]$ **then**
            $d[v] = d[u] + w$, $p[v] = u$
        end if
    end for
end while

IV. Evaluation

We evaluate the proposed relabeling algorithms on different network topology by synthetic simulations and compare it with the prior work $MLSnet$ [5].

A. Simulation setting

We evaluate $mMLSnet$ on two types of topology: random graph and real network topology sampled from Topology Zoo [21]. The switch and access points are assigned a security label of secret and public randomly but the number of the secret switch and access point only is $X\%$ of the total switch and access points. 20 hosts are connected to each access point or switch and each host is assigned the same security label as the switch or access point to which it is attached. Around 50 flows are randomly generated per second as the background traffic except that the source and destination node are sampled from the same security group. $X\%$ of the flows are secret while the rest are public. Flow duration average are set to be 5s. The initial flow paths and security label configuration are conducted following $MLSnet$.

B. Random graph

The random graph can be characterized by two parameters: the number of nodes $|V|$, and the number of edges $|E|$. Hence we evaluate our relabeling algorithm in comparison with $MLSnet$ on different $|V|$ and $|E|$ settings. Here we define the edge density $p = \frac{2|E|}{|V|(|V|-1)}$. If $p = 1$, the graph becomes the mesh topology which is popular in wireless networks. If $p = 0$, the graph becomes a set of isolated nodes.

We set $|V| = 50, X = 20$ and Fig. 2 shows that our method is significantly faster than the benchmark at the cost of longer flow paths. The left y axis denotes the running time

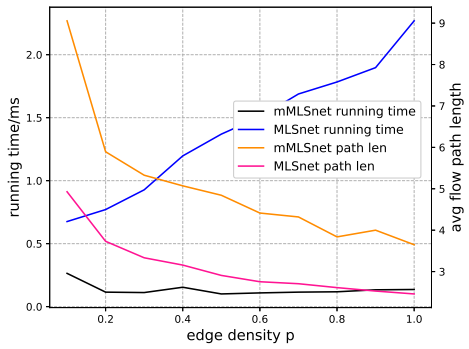

Fig. 2. Effect of p on the running time and average path length

of the path finding algorithm when one secret node moves and the right y axis denotes the average flow path length of the mobile secret node. More specifically, our method is at least ten times faster when the graph is dense like when $p = 0.8$ but still two time faster when the graph is sparse. The advantage mainly stems from the incremental path extension idea and the early stop trick. As the graph becomes more dense, the early stop yields more benefit. This short running time and the fact that the running time is resistant to the edge density make a timely solution for mobility possible. On the other hand, our average flow paths are nearly 1.5 hops longer than $MLSnet$. In practice, our method can provide fast mobility support at the cost of longer paths, and a shorter path can be found and replace our path when the periodic recalculation of flow rules occurs.

We also measure the number of secret switches and access points after relabeling and the portion of blocked public flow when $|V| = 50, X = 20$. It turns out that both $MLSnet$ and $mMLSnet$ have the same number of secret nodes after relabeling and a similar portion of blocked public flow regardless of p. This implies that our method achieves a faster running time without more disruption to the public traffic.

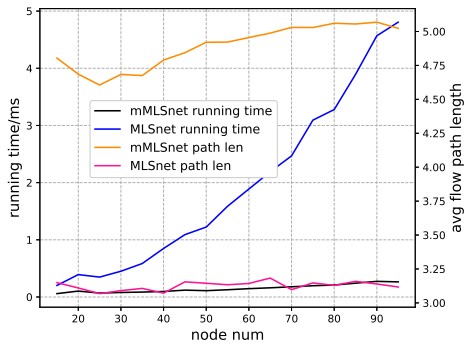

Fig. 3. Effect of $|V|$ on the running time and average path length

Set $p = 0.5, X = 20$ and Fig. 3 illustrates the running time and the average flow path hops when the node number $|V|$ varies. The running time of $MLSnet$ grows linearly with respect to $|V|$ while the running time of our methods remains

the same. The average flow path length of $mMLSnet$ is around 1.5 hops longer than that of $MLSnet$ whatever $|V|$ is. We also find that the average flow path length and the running time doesn't change much when $p = 0.5, |V| = 50$ and the secret node percentage X range from 5 to 20. This indicates that our method has good scalability and maintains a short running time on a random graph of different settings. Combining Fig. 2 and Fig. 3 we conclude that the edge density p solely decides the average flow path length difference.

C. Realistic topology

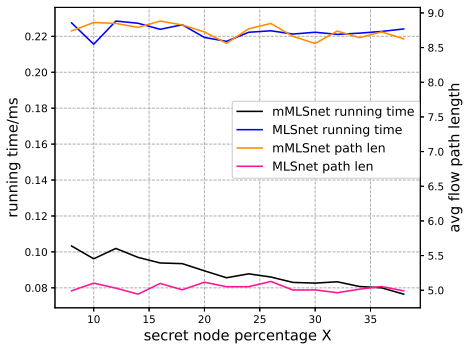

Fig. 4. Effect of X on the running time and average path length

We evaluate the algorithm on twenty topologies sampled from Topology Zoo. Fig. 4 shows that the running time and average flow path length are invariant to the percentage X of secret nodes in the network. In fact, this claim holds for all the sampled topologies. We observe that the path length difference is more distinct on sparse graphs and the running time difference is more obvious on dense graphs. Compared with Fig. 2, the average path length increases from 1.5 hops to 3.5hops. Intuitively, Fig. 4 also demonstrates the insight that $mMLSnet$ trades the average flow path length (traffic latency) for running time (flow path initialization time).

Less obviously, $mMLSnet$ pursues short running time and higher coverage for secret flows by potentially assigning more secret access points and switches and blocking more public traffic. In $MLSnet$, the flow path finding algorithm runs for multiple rounds because relabeling may include both upgrades and downgrades of the security label. For example, consider when a secret user moves from a secret access point A to a public access point B. Our method keeps A secret and uses it as an intermediate routing point. But in $MLSnet$, A may be downgraded to public if all the secret users attached to A move away and new public users coming in are attached to A. While the upgrade blocks public traffic and isolates secret traffic, the downgrade may bring new security policy violations that are not found in the first round. In the example above, a secret user that is not attached to A may find one of its flows, which is routed through A, is disrupted if A is downgraded to public. The goal of the second round of the initial MLSNet algorithm is to eliminate the possibility of this conflict. In contrast, $mMLSnet$ prioritizes routing for secret mobile nodes and only upgrades access points and switches. Therefore, we can skip the next round of security label examination because the consequence of the upgrade is isolating secret traffic from public traffic, which is approved by the MLS policy. This side effect is less intuitive and may not be observed, as it is in our experiments.

V. Discussion

A. Architecture for scalability

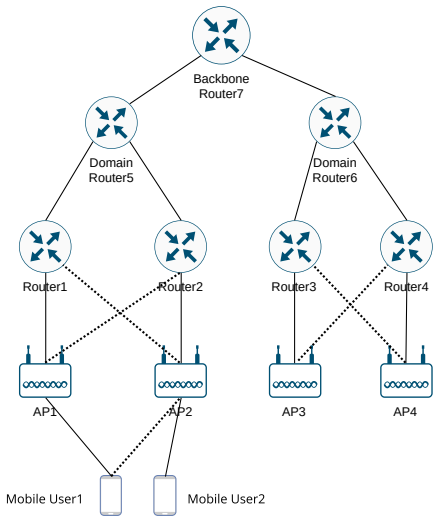

Fig. 5. wireless network architecture

While $mMLSnet$ enables routing for mobile users, the disruption on the public network traffic is not negligible. One way to overcome this weakness and enhance the scalability of MLS network is to introduce hierarchy [20]. Consider the wireless network topology in Fig. 5. At first, we assume the dot-links do not exist. If mobile user 1 connected to access point (AP) 1 is secret, AP1 and Router 1 must be secret. Now if $mMLSnet$ is enforced and the mobile user 1 moves and associates with AP2, $mMLSnet$ will find the shortest path: AP2, Router2, Domain Router5, Router1 and AP1. All these routers and APs will be relabeled as secret. Consequently, all the public users in the first domain are blocked since Router5 is the domain router. However, suppose the underlying APs have some degree of the connectivity redundancy and the dot-links exist. Then only AP2 and Router2 are relabeled as secret and other public users who are connected to Domain Router5 via routers other than Router1 and Router2 will be untouched. Similar techniques can be applied to the routers in the upper layers.

In fact, the idea of the connectivity redundancy can also found in the enterprise networks such as the fat tree topology [22]. Essentially, networks that require slicing or traffic isolation need to reduce the graph conductance to improve scalability. The graph conductance is defined as followed: For the mobility problem in MLS networks, traffic isolation happens most in the network edges such as the access points so the conductance can be defined between a single access point

and the rest of the network. On the other hand, the underlying idea of $mMLSnet$ is to temporarily increase the number of the secret nodes to allow more coverage for secret nodes. Then the conductance inside the connected components of secret nodes is reduced while the conductance for the connected components of public nodes is increased. To put it in another way, the mobility problem in MLS networks can be viewed as resource balancing between different security groups.

B. Heuristic search

The shortest path search algorithms can be further accelerated with heuristic searching. Here we introduce one possible direction in the search optimizations.

Similar to $A^\star$, we can estimate the distance from the current node to the destination and prioritize the search based on the sum of the current distance of the node and the remaining distance. In mobile MLS networks, the majority of the network should maintain the same security label since the moving nodes are limited and the moving secret nodes, which require relabeling, are rare because there are only a few secret nodes. We can possibly reuse the remaining distances calculated from previous rounds to estimate the new remaining distance.

VI. Conclusion

Observing that many studies of MLS networks are based on static security label configuration and fail to adapt to device movement, we develop a network model $mMLSnet$ that can dynamically relabel the network and quickly restore flows of mobile devices. We provide a simple mobility protocol and develop an algorithm to search for the shortest secure routing paths. We show that $mMLSnet$ can react to the device movement quickly at the cost of a modest increase in average flow path length. In demonstrating the underlying idea of $mMLSnet$, we also identify the need of new architecture designs and heuristic shortest path search algorithms.

References

[1] T. D. Nguyen, M. A. Gondree, D. J. Shifflett, J. Khosalim, T. E. Levin, and C. E. Irvine, "A cloud-oriented cross-domain security architecture," in *Proc. of IEEE MILCOM*, 2010.

[2] N. Meghanathan, "Review of access control models for cloud computing," *Computer Science & Information Science*, vol. 3, no. 1, pp. 77–85, 2013.

[3] D. Gonzales, J. M. Kaplan, E. Saltzman, Z. Winkelman, and D. Woods, "Cloud-trust—a security assessment model for infrastructure as a service (iaas) clouds," *IEEE Transactions on Cloud Computing*, vol. 5, no. 3, pp. 523–536, 2015.

[4] S. Achleitner, Q. Burke, P. McDaniel, T. Jaeger, T. La Porta, and S. Krishnamurthy, "Mlsnet: A policy complying multilevel security framework for software defined networking," *IEEE Transactions on Network and Service Management*, vol. 18, no. 1, pp. 729–744, 2020.

[5] Q. Burke, F. Mehmeti, R. George, K. Ostrowski, T. Jaeger, T. F. La Porta, and P. McDaniel, "Enforcing multilevel security policies in unstable networks," *IEEE Transactions on Network and Service Management*, vol. 19, no. 3, pp. 2349–2365, 2022.

[6] G. Pernul, W. Winiwarter, and A. M. Tjoa, "The entity-relationship model for multilevel security," in *International Conference on Conceptual Modeling*. Springer, 1993, pp. 166–177.

[7] W.-P. Lu and M. K. Sundareshan, "A model for multilevel security in computer networks," *IEEE Transactions on Software Engineering*, vol. 16, no. 6, pp. 647–659, 1990.

[8] V. Varadharajan, K. Karmakar, U. Tupakula, and M. Hitchens, "A policy-based security architecture for software-defined networks," *IEEE Transactions on Information Forensics and Security*, 2018.

[9] L. Yuan, H. Chen, J. Mai, C.-N. Chuah, Z. Su, and P. Mohapatra, "Fireman: A toolkit for firewall modeling and analysis," in *Proc. of IEEE Symposium on Security and Privacy (S&P'06)*, 2006.

[10] K. P. Dyer, S. E. Coull, T. Ristenpart, and T. Shrimpton, "Peek-a-boo, i still see you: Why efficient traffic analysis countermeasures fail," in *2012 IEEE Symposium on Security and Privacy*, 2012, pp. 332–346.

[11] D. Drutskoy, E. Keller, and J. Rexford, "Scalable network virtualization in software-defined networks," *IEEE Internet Computing*, vol. 17, no. 2, pp. 20–27, 2012.

[12] M. Yu, J. Rexford, X. Sun, S. Rao, and N. Feamster, "A survey of virtual lan usage in campus networks," *IEEE Communications Magazine*, vol. 49, no. 7, pp. 98–103, 2011.

[13] M. Ibrar, L. Wang, G.-M. Muntean, A. Akbar, N. Shah, and K. R. Malik, "Prepass-flow: A machine learning based technique to minimize acl policy violation due to links failure in hybrid sdn," *Computer Networks*, vol. 184, p. 107706, 2021.

[14] B. Tian, X. Zhang, E. Zhai, H. H. Liu, Q. Ye, C. Wang, X. Wu, Z. Ji, Y. Sang, M. Zhang *et al.*, "Safely and automatically updating in-network acl configurations with intent language," in *Proceedings of the ACM Special Interest Group on Data Communication*, 2019, pp. 214–226.

[15] M. Ali, N. Shah, and M. A. K. Khattak, "Dai: Dynamic acl policy implementation for software-defined networking," in *2020 IEEE 17th International Conference on Smart Communities: Improving Quality of Life Using ICT, IoT and AI (HONET)*. IEEE, 2020, pp. 138–142.

[16] S. Pisharody, J. Natarajan, A. Chowdhary, A. Alshalan, and D. Huang, "Brew: A security policy analysis framework for distributed sdn-based cloud environments," *IEEE transactions on dependable and secure computing*, vol. 16, no. 6, pp. 1011–1025, 2017.

[17] C. Perkins, "Ip mobility support for ipv4, revised," Tech. Rep., 2010.

[18] C. E. Perkins, "Mobile ip," *IEEE Communications Magazine*, vol. 40, no. 5, pp. 66–82, 2002.

[19] C.-C. Tseng, G.-C. Lee, R.-S. Liu, and T.-P. Wang, "Hmrsvp: A hierarchical mobile rsvp protocol," *Wireless Networks*, vol. 9, pp. 95–102, 2003.

[20] R. Ramjee, K. Varadhan, L. Salgarelli, S. R. Thuel, S.-Y. Wang, and T. La Porta, "Hawaii: A domain-based approach for supporting mobility in wide-area wireless networks," *IEEE/ACM Transactions on networking*, vol. 10, no. 3, pp. 396–410, 2002.

[21] S. Knight, H. Nguyen, N. Falkner, R. Bowden, and M. Roughan, "The Internet Topology Zoo," *IEEE Journal on Selected Areas in Communications*, vol. 29, no. 9, pp. 1765–1775, 2011.

[22] M. Al-Fares, A. Loukissas, and A. Vahdat, "A scalable, commodity data center network architecture," *ACM SIGCOMM computer communication review*, vol. 38, no. 4, pp. 63–74, 2008.