Contents lists available at ScienceDirect

Journal of Network and Computer Applications

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

Research paper

Seraph: Towards secure and efficient multi-controller authentication with (t, n)-threshold signature in multi-domain SDWAN

Wendi Feng [a,b,*], Ke Liu [a], Shuo Sun [a], Bo Cheng [b], Wei Zhang [a,c,d]

[a] School of Computer Science, Beijing Information Science & Technology University, 55 Taihang Rd, Changping District, 102206, Beijing, China
[b] State Key Laboratory of Networking and Switching Technology, Beijing University of Posts and Telecommunications, 10 Xitucheng RD, 100876, Beijing, China
[c] Beijing Laboratory of National Economic Security Early-warning Engineering, BISTU, 55 Taihang Rd, Changping District, 102206, Beijing, China
[d] Beijing Advanced Innovation Center for Future Blockchain and Privacy Computing, BISTU, 55 Taihang Rd, Changping District, 102206, Beijing, China

ARTICLE INFO

Keywords:
SDWAN
Authentication
Threshold signature

ABSTRACT

The multi-controller scheme is widely adopted in Software-Defined Wide Area Networks (SDWANs), where a WAN is segmented into multiple domains, each controlled by one controller. These controllers communicate with each other in-band, necessitating authentication before exchanging control messages. However, relying solely on identification of a single node for authentication exposes the network to spoofing attacks, jeopardizing its security. To address this issue, we present Seraph, an innovative (t, n)-threshold signature-based authentication scheme that verifies not only the node itself but also its "endorsement" nodes to establish its identity. We have investigated the best practice for defining the "endorsement" relationships concerning security and overheads, formulating the problem as an integer programming problem. We have demonstrated the polynomial-time hardness (NP-hardness) of the problem and proposed an efficient Seraph algorithm. Through our rigorous simulation analysis, we show that Seraph can provide comparative performance with Optimal and reduce time usage by over 90%.

1. Introduction

Software-defined Networking (SDN) (McKeown et al., 2008) is a prevalent paradigm that extracts the *control plane* from the *data plane*, facilitating the control plane with a global view of the whole network. This new feature significantly benefits the network not only for finer-grained controlling and management but also for fast enforcement of network policies, making it easier and more efficient for traffic engineering for Wide Area Networks (WANs) engineering (Hong et al., 2018, 2013; Poutievski et al., 2022). Thus, the SDN-powered WAN becomes an *SDWAN*.

In large-scale commercial SDWAN networks, multiple controllers are employed for enhanced resiliency (Guo et al., 2022) and performance (Ferguson et al., 2021). As a result, the network is divided into several domains, each under the control of one (logical) controller. By sharing control messages with other controllers, each controller has a comprehensive view of the entire network. Typically, these control messages are sent *in-band*, leveraging the underlying data plane network to relay them (Hong et al., 2013). Hence, authentication is necessary before any controller communications to ensure network security.

The current authentication mechanisms for controllers primarily rely on the *Public Key Infrastructure* (PKI) approach, where a *Certificate Authority* (CA) is utilized to validate the identity of each controller. In most cases, SDWAN owners establish their own CA (Osborn et al., 2016), but it can become a single point of vulnerability, as the entire network is compromised if the CA is hacked. Recent approaches have explored utilizing blockchain technology (Yang et al., 2019; Li et al., 2022), which creates a distributed database for storing authentication information. However, these approaches encounter challenges in terms of scalability and performance, as they require the participation of all controllers for authentication and can significantly impact the efficiency of network policy enforcement.

In this paper, we present Seraph, a (t, n)-threshold signature-based approach. When authenticating a controller (*i.e.*, source node), Seraph judiciously chooses a few "endorsement" nodes to prove the authenticity of the source node. The intelligence behind is the (t, n)-threshold signature, where *siglets* (*i.e.*, signature pieces) of t nodes out of the total number of n nodes can be consolidated and verified (Shamir, 1979). However, employing all other nodes for "endorsement" can incur significant performance overhead, while using fewer can compromise the

[*] Corresponding author at: School of Computer Science, Beijing Information Science & Technology University, 55 Taihang Rd, Changping District, 102206, Beijing, China.
E-mail address: wendifeng@bistu.edu.cn (W. Feng).

https://doi.org/10.1016/j.jnca.2024.103920
Received 16 October 2023; Received in revised form 25 March 2024; Accepted 16 May 2024
Available online 21 June 2024

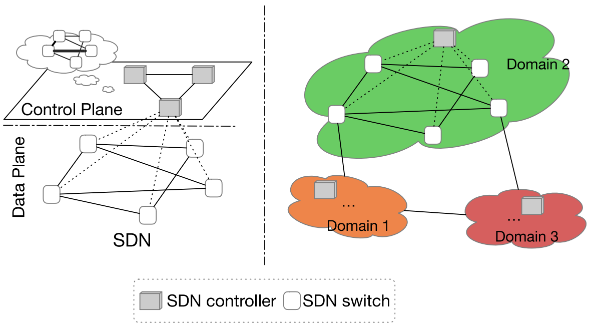


Fig. 1. Two forms of multi-controller SDNs.

security as well. The question is: *how can we achieve both security and performance simultaneously?*

We define the challenge as the *"Endorsement" Mapping Problem* (EMP). Our observation is that the "endorsement" relationship plays a crucial role in achieving the goals of controller authentication in a multi-domain SDWAN network. Consequently, we formulate the EMP problem by considering both authentication security and the performance overhead associated with "endorsement". Essentially, the problem aims to minimize the performance overhead while meeting specific security requirements. We mathematically formulate the EMP problem as an integer programming problem and prove its Non-polynomial hard (NP-hard) time complexity. To efficiently tackle the problem, we propose the Seraph algorithm, which iteratively selects a node based on two criteria: the minimum communication overhead (distance) and the minimum probability of compromise. Seraph repeats the process until the "endorsement" mapping satisfies the security requirement. Through extensive simulations, we demonstrate that Seraph achieves comparable performance to the Optimal solution while significantly reducing the time required by over 90%.

To summarize, our contribution is three-fold:

- We identify the drawbacks of existing PKI-based point-to-point authentication schemes in multi-controller SDWAN and present the EMP problem and the Seraph protocol.
- We mathematically formulate the EMP problem as an integer programming problem, prove its NP-hard time complexity, and propose an efficient heuristic algorithm called Seraph.
- We conduct rigorous simulations to evaluate the performance of the Seraph algorithm and prove it can achieve near-optimal performance with over 90% time-usage reduction.

The rest of the paper is structured as follows. Section 2 provides a concise overview of the essential background knowledge on SDWAN and the (t, n)-threshold signature mechanism, followed by a survey of related work. In Section 3, we illustrate the EMP problem with a concrete example and introduce our attack model. The detailed mechanism of the Seraph protocol is described in Section 4, while in Section 5, we formally formulate the EMP problem. We further demonstrate the NP-hard time complexity of EMP and propose an efficient heuristic algorithm in Section 6. We conduct comprehensive simulations and analyze the simulation results in Section 7. Finally, We discuss Seraph's various properties in Section 8 and conclude the paper in Section 9.

2. Background and related work

In this section, we brief the SDN and multi-controller SDN, followed by the drawbacks of existing controller authentication mechanisms.

2.1. SDN and Multi-Controller SDN

SDN, a revolutionary network paradigm, separates the *control plane* from the *data plane*. This approach enables the deployment of a (conceptually) centralized controller that provides a comprehensive overview of the network and directs the forwarding of packets in the data plane (consisting of controlled devices). Communication between the controller and SDN switches can be established through either *in-band* or *out-of-band* methods, utilizing SDN protocols such as OpenFlow (McKeown et al., 2008). In the in-band approach, control messages are transmitted over the links shared by the data plane. Conversely, the out-of-band method employs dedicated "controller – switch" links for communication.

Multi-Controller SDNs. In cases where the network comprises a large number of forwarding devices or when the data plane is distributed far apart geographically, a single controller may not be sufficient to handle the requests effectively. Thus, multi-controller SDNs are employed. In this approach, multiple controllers are utilized, where each controller is responsible for a subset of SDN switches that are located in a specific region and exchanges topology information. Despite the distribution of controllers across different locations, the control plane remains logically centralized, achieving increased resilience to failure and enhanced performance. The multi-controller scheme is not only applied in large SD-WAN networks but also in smaller local SDN networks (see Fig. 1). This paper focuses on multi-domain SDWANs.

2.2. Controller authentications

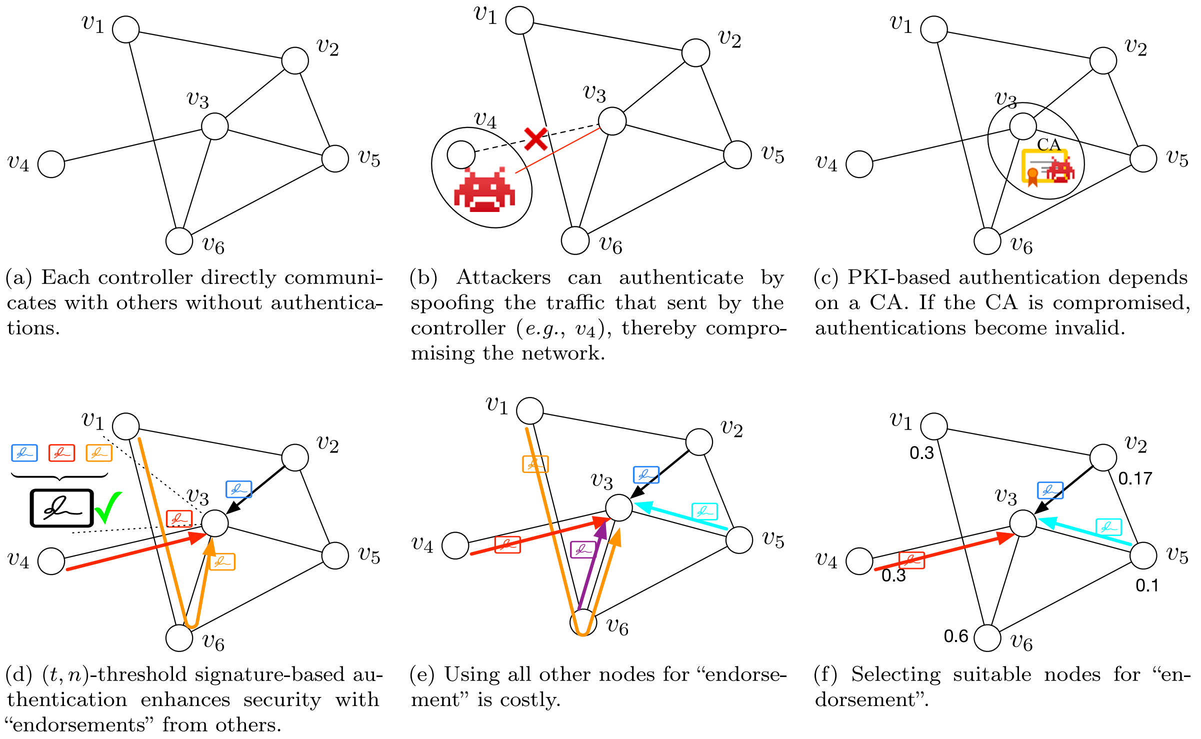
According to the de-facto standard of SDN protocols (*i.e.*, OpenFlow (Open Networking Foundation, 2022)), controller authentication is not mandatory in SDN. The specification only recommends configuring control channels between controllers and switches using the SSL/TLS protocol without explicitly handling authentications over controllers (due to the multi-controller paradigm is not specified). Consequently, controller authentication is typically achieved through Public Key Infrastructure (PKI), where a Certificate Authority (CA) issues certificates for controllers to serve as their identification. The switch verifies the received certificate before establishing a secure channel with the controller. Existing SDN frameworks such as Orion (Fu et al., 2014), NOX (Gude et al., 2008), POX (Kaur et al., 2014), and Ryu (Ryu SDN Framework Community, 2023) employ this approach for controller authentications.

2.3. Distributed threshold signature

The threshold signature scheme (Shamir, 1979) allows for many-to-one signing and verifying, whereby a group of participants can generate their own partial signatures (*i.e.*, siglets), and the verifier can consolidate the individual siglets and verify the *global signature*. The scheme requires the number of participants to exceed a certain *threshold*, hence the name "threshold signature". The process consists of five steps as follows: **(i) Parameter negotiation:** The scheme first determines the value of t, n, two large primes p, q, and each participant's polynomials. These parameters can be configured by the operator. **(ii) Key calculation:** Individual public keys pk_i and private keys qk_i are calculated for each participant using the negotiated parameters, and the individual public keys are distributed. After receiving the individual public keys, the global public key pk can be constructed. **(iii) Siglet generation:** Each participant calculates the siglet using their private key. **(iv) Global signature generation and verification:** The verifier, which can be any participant, receives t siglets and calculates (consolidates) the global signature with them. The verifier is then able to verify the global signature using pk. **(v) Signature sharing:** The global signature is shared among the participants who are required to present the signature for various purposes.

2.4. Related work

This subsection surveys relevant work, which can be classified into three categories, detailed as follows.



(a) Each controller directly communicates with others without authentications.

(b) Attackers can authenticate by spoofing the traffic that sent by the controller (*e.g.*, v_4), thereby compromising the network.

(c) PKI-based authentication depends on a CA. If the CA is compromised, authentications become invalid.

(d) (t, n)-threshold signature-based authentication enhances security with “endorsements” from others.

(e) Using all other nodes for “endorsement” is costly.

(f) Selecting suitable nodes for “endorsement”.

Fig. 2. Motivation examples. A multi-domain SDN network with six controllers, each node represents a controller for brevity.

2.4.1. Point-to-point authentication

Point-to-point authentication mechanisms rely solely on the authenticating and authenticated nodes (*i.e.*, source and destination). This mechanism is commonly used in multi-controller SDN schemes. For example, DISCO (Phemius et al., 2014) and SPARC (Li et al., 2019) utilize the Simple Authentication Security Layer (SASL) (Myers, 1997), which leverages an authentication ID issued by the system operator to authenticate controllers. Similarly, Onix (Koponen et al., 2010) and Orion (Ferguson et al., 2021) employ remote procedure calls (RPCs) between controllers, employing SSL/TLS for security. Authentication in these cases is achieved using certificates issued by a CA. CIDC (Benamrane et al., 2017) and Hyperflow (Tootoonchian and Ganjali, 2010) are also examples of this kind. However, the point-to-point scheme requires a central management agent, such as the CA, which can introduce security vulnerabilities and become a single point of compromise and failure, thus impairing overall security enhancements.

2.4.2. Blockchain-based distributed authentication

Blockchain-based schemes replace the centralized CA with a distributed blockchain (Zheng et al., 2018). The blockchain can be seen as a decentralized database where blocks are connected through cryptographic hash functions. Each block contains a replica of the database, including a cryptographic hash of the previous block, a timestamp, and transaction data (records). In blockchains, records are immutable, ensuring that the stored identifications cannot be forged. BlockCtrl (Yang et al., 2019) and BlockREV (Li et al., 2022) are examples of authentication schemes that utilize blockchain. While these blockchain-based schemes address the issue of a single point of failure associated with CAs, attackers can still attempt to spoof authentication information by intercepting the traffic and potentially compromising the network's security. Moreover, some blockchain consensus mechanisms like Proof-of-Work (PoW) rely on computational power. An attacker with more than 51% of the computational resources can “rule” the consensus and compromise the system. Therefore, additional measures should be implemented to protect against such attacks and maintain the integrity of the authentication process.

2.4.3. Threshold signature-based schemes

da Silva et al. (2020) suggest attesting the authenticity of controller states using a signature generated with a threshold encryption mechanism. FastHand (Roy et al., 2022) addresses the limitations of traditional PKI-based authentication and employs a threshold signature mechanism for user equipment (UE) handover across 5G base stations. However, both schemes treat each node equally without considering best practices for selecting an appropriate number of suitable nodes to enhance security since proper consideration of node selection is essential to improve the overall security of the threshold signature mechanism (see Section 3.1).

3. Motivation and attack model

3.1. Motivation

In this subsection, we motivate the EMP problem through illustrative examples. To facilitate explanation, we employ a synthetic logical topology of an SDN network consisting of six nodes, and each represents controllers or domains with controllers in a multi-domain SDWAN setup (refer to Fig. 2(a)). This simplified representation allows us to demonstrate the key concepts and challenges involved in achieving efficient controller authentication.

3.1.1. Needs for controller authentications

In the depicted multi-controller network with six nodes (see Fig. 2(b)), no security enhancement measures are applied. Consequently, if an end-host (*e.g.*, a man-in-the-middle attacker) within the network or a domain decides to impersonate a controller, say v_2, the attacker can easily intercept the communication messages. By constructing packets to hijack the traffic, the attacker can then proceed to send misleading information to other controllers. As a result, the integrity of the network is compromised and susceptible to malicious activities.

3.1.2. *PKI-based authentication is vulnerable*

PKI-based authentication is a commonly employed mechanism for authenticating network entities. It relies on a CA to issue certificates for these entities. However, this centralized approach poses a single point of vulnerability. As shown in Fig. 2(c), when the CA is compromised, an attacker can exploit this situation to grant a legitimate identity to a device under their control. This compromised device can then be utilized to compromise the network, enabling potential unauthorized access and malicious activities.

3.1.3. *(t, n)-Threshold signature brings new hope*

In the (t, n)-threshold signature authentication scheme, the authorization of an entity by other entities requires "endorsement" from other $t - 1$ entities, thus significantly enhancing security. This process is illustrated in Fig. 2(d), where v_2 intends to communicate with v_3. A (t, n)-threshold signature-based authentication is initiated. Firstly, v_2 sends a siglet, generated using its private key, to v_3. Subsequently, v_3 requests siglets from two other nodes (*e.g.*, v_1 and v_4) to "endorse" the communication. By consolidating the three siglets (from v_1, v_2, and v_4), v_3 creates a *global signature*. Upon successfully verifying this signature using its own public key, v_3 can determine that v_2 is a *trustworthy partner*, thereby allowing further communication to proceed with confidence.

3.1.4. *Choosing the "endorsements" is non-trivial*

Nevertheless, it is important to note that the (t, n)-threshold signature-based authentication can have a dual impact. While it enhances security, it also introduces performance overhead that needs to be taken into consideration.

A Larger number of "endorsements" improves security but also results in increased overhead. This can be observed in the example depicted in Fig. 2(e). The overhead introduced includes the time taken to obtain each siglet (*e.g.*, half round-trip time (RTT) between v_2 and v_3, and RTT among v_1 and v_3, v_4 and v_3), the overhead for consolidating the siglet into a global signature, and the overhead of verifying the global signature. It is clear that the overhead is directly related to the value of t, with larger t values leading to higher overhead. Additionally, the selection of nodes for "endorsement" also affects performance, as nodes located farther away contribute to longer RTT, further impacting the overhead.

In addition, the probability of each controller being compromised can vary depending on factors such as the number of hosts and operators. It is worth noting that to control the communication, attackers would need to compromise all "endorsement" nodes. This highlights the significance of carefully **selecting the appropriate nodes** for "endorsement" in the authentication process as an essential task.

Hence, **judiciously selecting t and the nodes with the consideration of efficiency and security is important.** As demonstrated in Fig. 2(f), it jointly considers security and efficiency, facilitating the network with security at a minimum performance overhead.

3.2. *Attack model*

Our attack model is based on the traffic spoofing attack (Tugnait, 2018), where the attacker can intercept the traffic of a controller within the same domain. We make four assumptions in this attack model: **(i)** Each attack is conducted by capturing the traffic only from the controller that resides in the same domain as the attacker's host. **(ii)** The attacker is unaware of the total number of controllers in the network and which specific nodes are used for "endorsing" an authentication. **(iii)** The attacker cannot compromise the host running the controller and thereby cannot gain access to the public/private keys used for generating and verifying signatures. **(iv)** The attack can neither discover the bitmap structure (as described in Section 4) nor identify the "endorsing" nodes. While our attack model focuses on spoofing attacks, common authentication attacks like password guessing and dictionary attacks can be inherently defended because our system relies on a threshold signature scheme, not individual passwords. An attacker would need to compromise a minimum number of nodes (t) to forge a valid signature, making these traditional attacks ineffective.

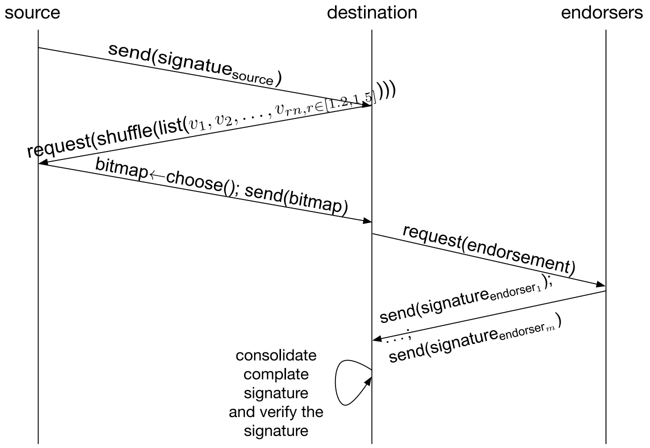

source
destination
endorsers
send(signatue$_{source}$)
request(shuffle(list($v_1, v_2, \ldots, v_{rn, r\in[1.2,1.5]}$)))
bitmap←choose(); send(bitmap)
request(endorsement)
send(signature$_{endorser_1}$);
. . . ;
send(signature$_{endorser_m}$)
consolidate
complate
signature
and verify the
signature

Fig. 3. Protocol procedures and phases.

4. The Seraph authentication protocol

This section outlines the Seraph protocol, taking into account the assumptions of our attack model as described in Section 3.2.

The initialization of the Seraph protocol is performed by the network operator, who configures the t value, "endorsement" mapping, and a polynomial required to establish the threshold signature scheme for authentication purposes, as described in Section 2.3. The "endorsement" mapping is derived from the Seraph algorithm, which is elaborated upon in Section 6.2.

Once the network begins, controllers exchange routing or control information among controller nodes. However, prior to this procedure, the Seraph authentication protocol is essential for security measures. It consists of three phases: *authentication requesting*, *"endorsement" information passing*, and *joint signing*.

As depicted in Fig. 3, when a controller (referred to as the "source") intends to transmit information to another entity (referred to as the "destination"), a series of steps should be followed: **(i)** Firstly, the source sends its signature to the destination. **(ii)** Upon receiving the signature, the destination acknowledges the authentication attempt and sends a shuffled list of node IDs back to the source. It is important to note that this list contains a greater number of nodes (ranging from 20% to 50%) than the actual topology to enhance security. **(iii)** Subsequently, the source receives the shuffled list, selects the necessary "endorsement" nodes, and returns the sequence bitmap to the destination. In this bitmap, the i^{th} bit represents the usability of the i^{th} node in the received shuffled list. By not revealing the construction of the bitmap, the attacker is unable to obtain the required "endorsement" nodes. **(iv)** The destination then receives the bitmap and requests signatures from the corresponding "endorsement" nodes. **(v)** Finally, the destination consolidates the signatures obtained from the source and the "endorsement" into a single global signature, which is subsequently verified. If the verification process succeeds, the source is authenticated, and subsequent control messages can be transmitted.

5. Formulation

In this section, we begin by providing a mathematical description of the multi-controller network and introducing the metrics employed to evaluate its security and performance. Subsequently, we outline the constraints and objective functions for the EMP problem, formulating it as an optimization problem.

Table 1
Notation definitions.

Notations	Description
$\mathcal{V}$	Network node set. $\mathcal{V} = \{v_1, v_2, \dots, v_n\}$.
$\mathcal{E}$	Edge set. $\mathcal{E} = \{e_1, e_2, \dots\}$.
n	Number of domains in the network.
T	Threshold set. $T = \{t_1, t_2, \dots, t_n\}$.
X	Matrix of the "endorsement" mapping. $X = \{x_{ij}\}, x_{ij} \in \{0, 1\}$.
$\mathcal{P}_i$	Set of (shortest) paths between node v_i and other nodes. $\mathcal{P}_i = \{\vec{p}_1, \vec{p}_2, \dots\}$.
$\mathbf{L}_i$	Set of lengths of paths in $\mathcal{P}_i$. $\mathbf{L}_i = \{\lvert\vec{p}_1\rvert, \lvert\vec{p}_2\rvert, \dots\} = \{l_1, l_2, \dots\}$.
$P_X(v_i)$	Probability of compromising the network under the "endorsement" mapping X.
p_i	Probability of compromising controller v_i.
$C_X^{v_{i_1}}(v_{i_2})$	Communication overhead for v_{i_1} authenticate at v_{i_2}.
$A_X^{v_{i_1}}(v_{i_2})$	Computational overhead for v_{i_1} authenticate at v_{i_2}.
α	Per-node signature consolidation time usage.
λ	Global signature verification time usage.

5.1. System description

In this subsection, we present the mathematical formulation of the multi-controller network. For brevity, each domain is represented as a single node, and the detailed topology within each domain is omitted. The definitions for the notation used can be found in Table 1.

The network can be represented as a graph $\mathcal{G} = (\mathcal{V}, \mathcal{E})$, where $\mathcal{V} = \{v_1, v_2, \dots, v_n\}$ is the set of network domain controllers. n represents the total number of domains, and v_i represents the ith domain's controller[1] in the network. $\mathcal{E}$ is the links set. Each node in $\mathcal{V}$ can choose a number t as the minimum number of nodes (including "endorsement" nodes and the source) required for authenticating the source. We use $T = \{t_1, t_2, \dots, t_n\}$ as the set of threshold values for authenticating each node. Let X be the matrix of "endorsement" mapping, and $x_{ij} = 1$ represents authenticating node i requires node j for "endorsement". Let $\mathcal{P}_i = \{\vec{p}_1, \vec{p}_2, \dots\}$ be the set of (shortest) paths of v_i and other nodes, and the length of each path is represented as the cardinality of the path vector set $\lvert\vec{p}_k\rvert$. Hence, the length set of path set $\mathcal{P}_i$ is written as $\mathbf{L}_i = \{\lvert\vec{p}_1\rvert, \lvert\vec{p}_2\rvert, \dots\}$.

5.2. Metrics

5.2.1. Probability-based security

Suppose an attacker can launch a spoofing attack by spoofing the controller of the same domain. To be authenticated, the attacker must obtain siglets of the "endorsement" nodes. However, since the construction of the bitmap (as mentioned in **Phase (iv)**) cannot be recovered by the attacker, the probability of being compromised (*i.e.*, the probability that the attacker successfully passes the authentication) is composed of two main factors: the probability of successfully guessing the "endorsement" pattern and the probability that all "endorsement" nodes are compromised. Thus, it can be expressed as

$$P_X(v_{i_1}) = \frac{1}{C_{n-1}^{t_{i_1}} + \cdots + C_{n-1}^{n-1}} p_{v_{i_1}} \prod_{j=1}^{n} p_j^{x_{i_1 j}}, \tag{1}$$

where v_{i_1} is the source node, and $p_{v_{i_1}} \prod_{j=1}^{n} p_j^{x_{i_1 j}}$ indicates the probability of compromising the source node and all "endorsement" nodes.

5.2.2. Communication overheads

As the threshold signature-based authentication necessitates additional assistance from the "endorsement" nodes, communication between the destination node and these nodes introduces authentication overheads. This can be expressed as the length of the corresponding point-to-point paths, which can be written as

$$C_X^{v_{i_1}}(v_{i_2}) = \mathbf{L}_{i_2} \cdot \overrightarrow{x_{i_1}}, \tag{2}$$

where $\mathbf{L}_{i_2}$ is the set of shortest lengths of paths between v_{i_2} and other nodes, and $C_X^{v_{i_1}}(v_{i_2})$ is the total length of the paths required for "endorsing" v_{i_1}'s authentication.

5.2.3. Signature computational overheads

When the destination node receives siglets from the source node and the "endorsement" nodes, it needs to consolidate these siglets into a single global signature and then verify it. The time required for verifying the global signature is constant, while the time taken to consolidate the siglets is proportional to the number of "endorsement" nodes. Thus, the overhead can be formulated as

$$A_X^{v_{i_1}}(v_{i_2}) = \alpha \sum_{j=1}^{n} x_{v_{i_1} j} + \lambda, \tag{3}$$

where α and λ are the constants that represent the calculation overhead of a per-node signature consolidation and signature verification, respectively.

5.3. Constraints

This subsection introduces the constraints considered in this work.

5.3.1. "Endorsement" constraint

Our scheme utilizes threshold signatures to enhance security, and the authentication of each node necessitates "endorsement" from at least one additional node. Therefore, the endorsement nodes should neither be the source nor the destination. This can be expressed as follows

$$x_{i_1 i_1} = 0, \tag{4}$$

where $v_{i_1} \in \mathcal{V}$ is the source node.

5.3.2. Threshold signature constraint

Furthermore, the total number of nodes involved in "endorsing" the authentication should not exceed the number of nodes in the network. This can be formulated as

$$t_i \le \sum_{j=1}^{n} x_{ij} \le n - 2, t_i \ge 2, \tag{5}$$

where $n - 2$ represents the authenticating nodes (*i.e.*, source and destination nodes) should be omitted, and t_i is the threshold in the (t, n) threshold signature.

5.3.3. Overall security constraint

Each network may support different services or applications that require varying levels of security. For instance, a network supporting military industrial control is considered more security-sensitive than a home network (Feng et al., 2023). Although application-level security requirements may vary, the fundamental security requirements for the physical WAN's domains rarely change. Hence, we utilize the level of security to assess whether a setup is suitable for safeguarding the system. Therefore, we can write

$$\log\left(p_i \prod_{j=1}^{n} p_j^{x_{ij}}\right) \le \ell_{sec} \log\left(p_i \prod_{j=1}^{n} p_j\right), \tag{6}$$

where ℓ_{sec} is the level of security assigned by the network operator, and the level of compromising probability across nodes should not exceeds ℓ_{sec}. $p_i \prod_{j=1}^{n} p_j$ denotes that the possibility of compromising all nodes in the network when all are chosen as "endorsement" nodes. To diminish the impact of the size of different topologies, we apply logarithm calculations to the compromising probabilities.

[1] We say ith controller to represent the ith domain's controller in the remainder of the paper.

5.4. Objective function

The EMP problem focuses on securing authentication (security) while minimizing performance overhead (efficiency). As each network operator can determine the necessary security level based on network usage, our problem transforms into finding the authentication "endorsement" mapping that attains the minimum performance overhead for a given security level. Hence, we can represent the objective function as

$$\begin{aligned} obj &= \sum_{v_{i_1} \in \mathcal{V}} \sum_{v_{i_2} \in \mathcal{V}} \left(C_X^{v_{i_1}}(v_{i_2}) + A_X^{v_{i_1}}(v_{i_2}) \right) \\ &= \sum_{v_{i_1} \in \mathcal{V}} \sum_{v_{i_2} \in \mathcal{V}} \left(\mathbf{L}_{i_2} \cdot \overrightarrow{x_{i_1}} + \alpha \sum_{j=1}^{n} x_{v_{i_1} j} + \lambda \right) \\ &= \sum_{v_{i_1} \in \mathcal{V}} \sum_{v_{i_2} \in \mathcal{V}} \sum_{j=1}^{n} x_{i_1 j} \left(l_{i_2 j} + \alpha \right) + (n-1)^2 \lambda. \end{aligned} \tag{7}$$

5.5. Problem formulation

As mentioned in the previous subsection, the EMP problem concerns finding a secure endorsement scheme X that attains minimum performance overhead (obj). This problem can be formulated as

$$\begin{aligned} \min_{x} \quad & \sum_{v_{i_1} \in \mathcal{V}} \sum_{v_{i_2} \in \mathcal{V}} \sum_{j=1}^{n} x_{i_1 j} \left(l_{i_2 j} + \alpha \right) + (n-1)^2 \lambda \\ \text{s.t.} \quad & (4)(5)(6), \\ & x_{ij} \in \{0, 1\}, \\ & \forall i, j, i_1, i_2 \in \mathcal{V}, i_2 \neq j, i_1 \neq i_2, \end{aligned} \tag{P}$$

where $\{x_{ij}\}$ are the designed binary integer variables, and $\mathbf{L}_i$ is the calculated path length vector. Symbols α, λ, and ℓ are constants indicating the overhead and security levels, respectively. Since the variable $\{x_{ij}\}$ is binary, Problem (P) is an *integer programming* problem.

6. Solution

In this section, we first prove the NP-hard complexity of the EMP problem and then present an efficient heuristic algorithm – Seraph.

6.1. Complexity analysis

This subsection proves the NP-hardness by reducing a special case of the EMP problem to the *traveling salesman problem* (TSP).

Theorem 1. *For a special case where (1) each node requires exactly one node to "endorse" the authentication; (2) each node only "endorses" the authentication of one other node; and (3) the attacker cannot guess the "endorsement" nodes, the EMP problem is NP-hard.*

Proof. We first introduce the traveling salesman problem (TSP). The TSP problem involves studying the problem of finding the optimal route for a salesman to traverse all the locations exactly once. A typical formulation of the TSP problem is as follows:

$$\begin{aligned} \min_{x} \quad & \sum_{i} \sum_{j} c_{ij} x_{ij} \\ \text{s.t.} \quad & \sum_{j} x_{ij} = 1, \sum_{i} x_{ij} = 1 \\ & \sum_{i} \sum_{j} x_{ij} \leq |S| - 1, \\ & \forall S \subset V, 2 \leq |S| \leq n - 2, \end{aligned} \tag{8}$$

where S is the set of all tours of the graph G. $\{x_{ij}\}$ indicates if the salesman visits city j immediately after city i, $x_{ij} = 1$; and 0 otherwise. It has been proven that the TSP problem is NP-hard (Jünger et al., 1995).

Algorithm 1: The Seraph algorithm.

```
Input: $G = (V, E)$: The topology;
Input: $\{p_i\}$: Compromise probability vector.
Input: $\ell_{sec}$: Configured security level.
Output: $X$: "Endorsement" mapping.
$X = \{0, \dots, 0\}$;  Calculate the node distance matrix $D$ with the
    Dijkstra algorithm (Johnson, 1973);
Derive list of vectors $D'$, where the $D'_i$ is vector of nodes sorted
    by distance between $v_i$ in ascending order;
Sort each vector in $D'$ by $\{p_i\}$ in ascending order as $D''$;
for $i \in [1, n]$ do
    $X' \leftarrow X$;
    for $v_j \in D''_i$ do
        $X'_{ij} = 1$;
        if $P_{X'}(v_i) < \ell_{sec}$ then
            $X \leftarrow X'$;
            break;
return $X$
```

We then prove under the aforementioned special case, Eq. (P) and the TSP problem are equivalent. Given the spacial case in Theorem 1, the objective function can be derived as $\sum_i \sum_j l'_{ij} x_{ij}$, and Constraint (6) is eliminated (given Condition (3)). It is noted that Constraint (5) can be derived to $n \leq \sum_{i=1}^{n} \sum_{j=1}^{n} x_{ij} \leq n(n-2)$. Let $S' = \sum_{i=1}^{n} \sum_{j=1}^{n} x_{ij} - (n-2)$. Hence, the constraint can be written as $2 \leq S' \leq (n-1)(n-2)$. Consequently, Eq. (P) can be reformulated as

$$\begin{aligned} \min_{x} \quad & \sum_{i=1}^{n} \sum_{j=1}^{n} l'_{ij} x_{ij} \\ \text{s.t.} \quad & \sum_{i=1}^{n} x_{ij} = 1, \sum_{j=1}^{n} x_{ij} = 1, \\ & 2 \leq S' \leq (n-1)(n-2), \end{aligned} \tag{P'}$$

where it is easy to prove that $(n-1)(n-2) \geq n-2$. Hence, $2 \leq S' \leq (n-1)(n-2)$ has a larger range. Problem (P') aims to minimize the overall transmission performance overhead. We can consider the "endorsement" mapping x_{ij} as the traveling sequence in the TSP problem and treat S' as the tour set S in TSP. We can prove that the solution with the minimum performance overhead exists if and only if the TSP problem has an optimal solution that uses the minimum cost to travel to all cities. The problem construction can be derived in polynomial time, but finding the optimal solution for Problem (P') is NP-hard due to the NP-hardness of the TSP problem.

Since Problem (P') is a special case of the TSP problem, we can conclude that

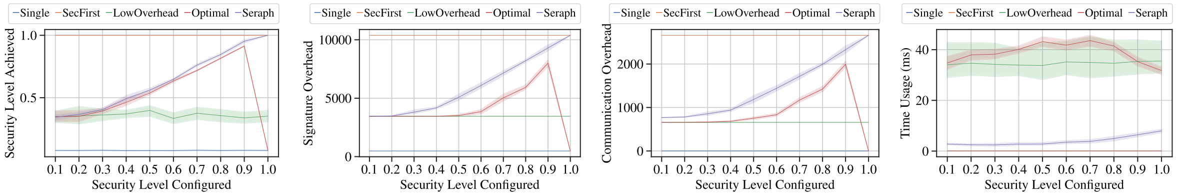
Theorem 2. *The "endorsement" mapping problem problem is NP-hard.*

6.2. The Seraph algorithm

The complexity of the EMP problem arises from the variations in the compromise probability of each node. Due to its NP-hard complexity, we propose an efficient heuristic algorithm to solve it.

As demonstrated in Algorithm 1, the intelligence behind Seraph lies in utilizing the *most secure and closest* nodes for authentication "endorsement". The Seraph approach is a three-step iterative process described as follows.

(1) Initialization. Seraph first computes the shortest paths for each pair of nodes with the Dijkstra Algorithm (Johnson, 1973), generating a list of "endorsement" candidates for each node. The list is then sorted in ascending order based on distance and in ascending order based on the probability of compromise.



(a) Security ratio. The higher, the better. (b) Signature overhead. The lower, the better. (c) Communication overhead. The lower, the better. (d) Time usage. The lower, the better.

Fig. 4. Performance of the compared algorithms on the Abilene topology.

(2) Security-first "endorsements". After initialization, Seraph iteratively assigns nodes from the sorted candidate list to "endorse" the authentication and updates the endorsement mapping x_i. This ensures that the selected nodes are both the closest and the most secure.

(3) Constraint comparing. Finally, Seraph calculates the compromise probability of the node (*i.e.*, $P_X(v_i)$) based on the "endorsement" mapping x_i. If the security level of x_i aligns with the network operator's configuration, Seraph proceeds to the next node. Otherwise, node v_i requires additional "endorsement" nodes to be assigned from the candidate list.

7. Simulation

This section describes the simulation of Seraph. We first present the simulation setup and then introduce the compared algorithms. Finally, we exhibit the simulation results and analyze them. **In summary, the simulation results indicate that Seraph can not only achieve comparable performance to Optimal but also scale well with larger topologies.**

7.1. Simulation setup

We utilize topologies from the Topology Zoo (Knight et al., 2011) in our simulations. It includes 262 real-world backbone network topologies provided in `gml` file format. Our simulation is implemented in Python, utilizing the popular `python-igraph` (Csardi et al., 2006) graph library for reading `gml` files and performing shortest path calculations. In the simulation, each node in a topology represents a domain with one deployed controller. The compromise probability for each controller (node) is randomly generated within the range of [0.1, 0.9] following a uniform distribution. The setting is based on the statistics for various vulnerability databases (National Vulnerability Database, 2024; Chinese National Vulnerability Database, 2024). The configured security level of the whole network is set from 0.1 to 1.

7.2. Compared algorithms

We compare three algorithms in the simulation described as follows.

- Single: this is the lower bound algorithm, where no "endorsement" authentications are required.
- Optimal: this is the optimal solution to the "endorsement" mapping problem problem.
- SecFirst: this solution aims at achieving the best security and is the upper bound.
- LowOverhead: this solution minimizes the "endorsement" overhead without considering security.
- Seraph: this solution is detailed in Section 6.2.

7.3. Simulation results and analysis

We compare the performance of the mentioned algorithms, evaluating *security ratios*, *signature overhead*, *communication overhead*, and *time usage*. Our results show that Seraph achieves near-optimal performance on small topologies and scales efficiently to larger topologies with stricter security requirements, indicating significantly faster execution times for Seraph.

7.3.1. Performance comparison on Abilene

We utilize the Abilene topology (Wikipedia, 2023), which consists of 11 nodes and 14 links. Abilene, the former version of Internet2 (Internet2, 2023), is extensively as a case studied in the networking community. Besides, due to the time-consuming nature of signature calculation and verification, which takes milliseconds to derive results, we preconfigure α (calculation overhead) and λ (signature verification overhead) to 9 ms and 4 ms, respectively. We believe the performance setting is sound as manifested in Stathakopoulou and Cachin (2017), Yu and Wang (2022). Finally, we conduct 10 simulations for each run to minimize errors.

(1) Security. We employ a probability-based approach that calculates the ratio of the compromised probability of a specific solution X to the ratio of that in the most secure setting (*i.e.*, the SecFirst algorithm). This can be derived from Eq. (6) and expressed as follows:

$$sec = \frac{\log\left(p_i \prod_{j=1}^{n} p_j^{x_{ij}}\right)}{\log\left(p_i \prod_{j=1}^{n} p_j\right)}, \tag{9}$$

where sec is the ratio, and we utilize logarithm to mitigate the impact of topology sizes (*i.e.*, the number of nodes in a topology). As shown in Fig. 4a, Single only achieves an average of 0.079 compared to the most secure scheme, failing to meet all security requirements. LowOverhead fails to satisfy 60% of the cases as it does not consider any security factors. Additionally, the variation (see the error bar) in LowOverhead is larger than that of other algorithms due to the larger solution space it considers. SecFirst serves as our baseline in this evaluation, representing the most secure scheme. Despite incurring significant overheads (see Section 7.3.1**(2)**), SecFirst satisfies the security requirements even when they are not explicitly demanded. Furthermore, we observe that Optimal drops to 0 when the security requirement is set to 1.0. This is because Optimal fails to generate a feasible solution for the EMP problem. On the other hand, Seraph can still fulfill the requirement thanks to its iterative approach. **Overall, Seraph demonstrates comparable performance to Optimal while satisfying all security requirements.**

(2) Overheads. We present the signature verification and communication overheads of the compared algorithms in Figs. 4(b)–4(c). SecFirst exhibits the highest overhead for both signature verification and communication due to its utilization of all nodes (excluding the source and destination) for "endorsement". This is anticipated since it pursues maximum security. In contrast, Single achieves the lowest overhead, with a 0 communication overhead as it solely authenticates with the source node itself. LowOverhead delivers a steady low-overhead performance in both signature verification and communication when compared to Optimal and Seraph. This is attributable to the relaxation of the security constraint in LowOverhead. Compared to Optimal, Seraph incurs up to 22.6% more overhead for signature verification and as high as 22.9% more overhead for communication, contributing less than 2 s for authentication. We consider the overhead to be acceptable for SD-WAN since the BGP protocol (a distributed protocol for multi-domain WAN) updates its routing information every 120 s (Gray et al., 2020). Furthermore, as Optimal fails to generate a feasible solution for the 1.0 security requirement, the overhead drops to 0 in both figures.

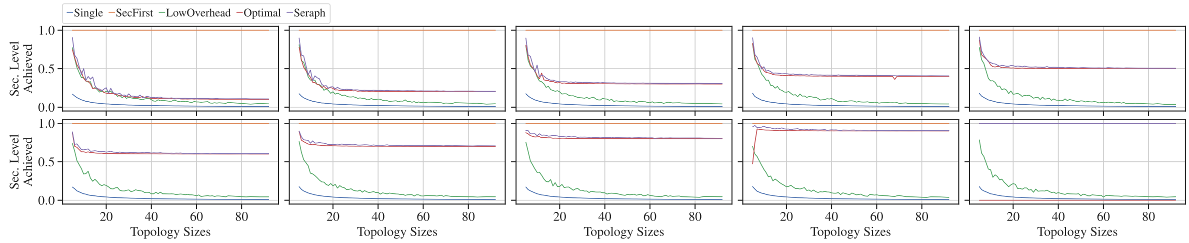


Fig. 5. Security performance of different topologies under different security requirements. The higher, the better.

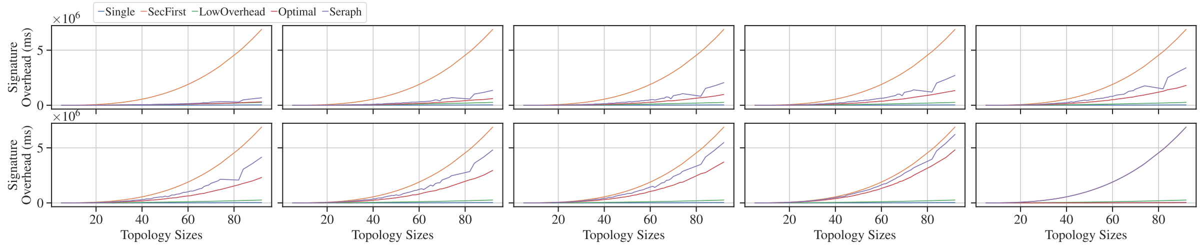


Fig. 6. Signature verification overhead of different topologies under different security requirements. The lower, the better.

(3) Time usage. We present the execution time comparisons of the algorithms. Single and SecFirst do not utilize any "endorsement" nodes, resulting in a fixed zero matrix solution and a 0 time usage. LowOverhead disregards constraints, enabling it to execute in a shorter duration but exhibiting greater variation due to the larger solution space mentioned in Section 7.3.1**(1)**. On the other hand, the Optimal algorithm achieves better security performance at the expense of longer time usage. Seraph significantly reduces time expenses by up to 93.6% while attaining near-optimal performance on security and overhead.

7.3.2. Performance across various topology sizes

To demonstrate the scalability of Seraph, we conducted simulations across multiple topologies ranging from 5 nodes to 92 nodes from Topology Zoo. We used the same α and λ values as mentioned in Section 7.3.1. For each simulation, we ran it 10 times to reduce errors, with average values used to show performance. **In summary, Seraph achieves comparable security performance to Optimal while attaining scalable to stricter constraints.**

(1) Security. We utilize the same security metric as shown in Eq. (9). Fig. 5 demonstrates the achieved security level under different topology sizes and security requirements (each displayed in each subplot). We observe that Single, SecFirst, and LowOverhead remain unchanged across different subplots as they are not affected by varying security requirements. Optimal and Seraph exhibit comparable performance and can fulfill most security requirements (ranging from 0.1 to 0.8). However, under security requirements of 0.9 and 1.0, Optimal encounters difficulties in deriving feasible solutions (*e.g.*, 5-node topology under the 0.9 security requirement and all topologies under the 1.0 security requirement), whereas **Seraph successfully generates corresponding solutions that satisfy all requirements.** Furthermore, we observe "L"-shape plots when the security requirement is below 0.9. This phenomenon arises from the fact that, with small topology sizes, the limited number of nodes leads to a significant increase or decrease in security with the "endorsement" of just one more or one less node. Additionally, we notice a peculiar occurrence where Optimal achieves a value close to 0.5 for the 5-node topology under the 0.9 security requirement, even though the value should either be 0 (indicating no feasible solution) or above 0.9. This discrepancy is due to averaging, where 5 out of 10 simulation rounds fail to derive feasible solutions.

(2) Overheads. As demonstrated in Figs. 6 and 7, Single once again achieves the minimum overhead across all requirements and topologies. The disparity between Optimal and Seraph becomes more pronounced with increasing topology size. Generally, Seraph contributes up to 23.3% more signature verification overhead and up to 21.8% more communication overhead. Moreover, communication overhead experiences irregularities in the Ulaknet topology (Topology Zoo, 2023) with 82 nodes. This can be attributed to Ulaknet's star-like topology, where three main nodes are interconnected while all other nodes connect to these three nodes. Turns out, the communication overhead is significantly reduced due to the maximum path length being limited to four.

(3) Time usage. As observed in Fig. 8, LowOverhead and Optimal, which solve the integer programming problem, exhibit comparable time usage across all security requirement cases and topologies. Furthermore, an exponential-like increase can be observed as the size of the topologies increases. Conversely, Seraph achieves linear performance improvement as the number of nodes increases, thanks to its iterative design. Additionally, it can reduce time usage by approximately 93% while achieving comparable security performance to Optimal.

8. Discussion

This section discusses miscellaneous properties of Seraph. These properties exhibit Seraph is practical, resilient, and performant in its applicable scenarios.

8.1. Practical deployment

The Seraph protocol is a security enhancement for the control plane of the SDWAN. Hence, it can best facilitate the *programmability* introduced by the SDN mechanism to implement the protocol as an application running on the control plane. This eliminates the need for hardware modifications or changes to the control plane operating system. Therefore, Deploying Seraph becomes as straightforward as installing a new software program, significantly reducing deployment complexity and costs.

8.2. Resilient to changes

Seraph is designed for multi-domain SDWANs, where the network is logically segmented into domains, each managed by a controller. Within a domain, changes to the network topology or capacity are expected. However, the topology between domains, and by extension, the controllers managing those domains, tend to remain stable. Since

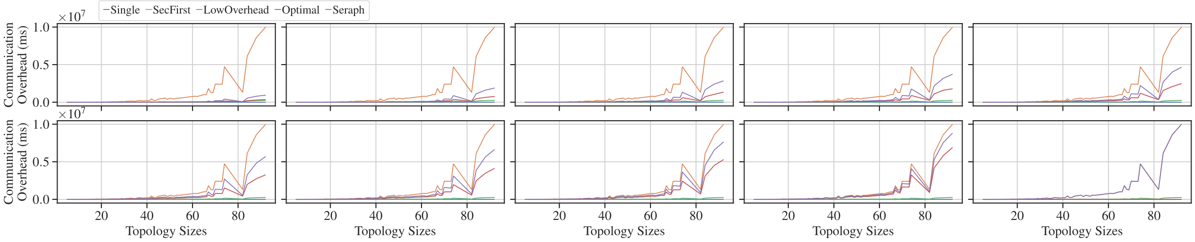


Fig. 7. Communication overhead of different topologies under different security requirements. The lower, the better.

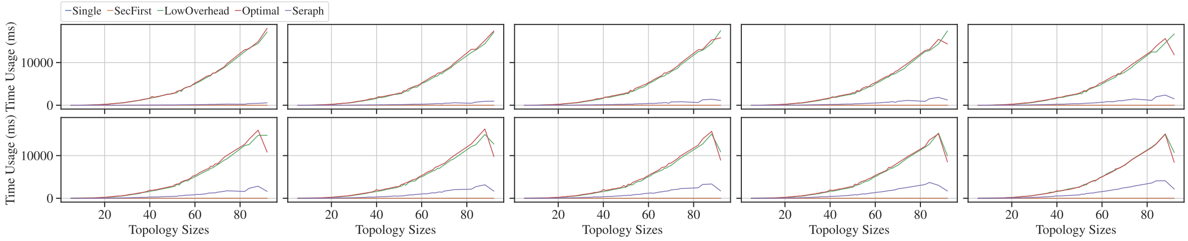


Fig. 8. Time usage of different topologies under different security requirements. The lower, the better.

Seraph focuses on authentication between controllers in these interconnected domains, frequent topological changes within domains have minimal impact on Seraph. In the rare instance where the overall network topology undergoes a significant shift, necessitating changes to the topology between domains, Seraph can be adapted through software reconfiguration. Hence, Seraph is resilient to changes.

8.3. Generality of the dataset

We leverage the Topology Zoo topological dataset in our simulations to evaluate the performance of the Seraph algorithm. We believe Topology Zoo is general, and the reason is three-fold. **(i) Diverse network types.** Topology Zoo includes a variety of real-world topologies, encompassing star (*e.g.*, Padi), ring (*e.g.*, Viatel), tree (*e.g.*, AMRES), mesh (*e.g.*, GlobalCenter), and daisy chain (*e.g.*, Chinanet) structures. This diversity provides a strong foundation for evaluating Seraph's adaptability to different network environments. **(ii) Real-World relevance.** Unlike synthetic datasets, Topology Zoo provides a total number of 262 real-world network topologies. This ensures that the performance evaluation reflects Seraph's effectiveness in realistic scenarios. **(iii) Popularity.** The popularity of Topology Zoo is evident with over 1550 citations for the JSAC paper (Knight et al., 2011) on Google Scholar. This widespread use within the network research community strengthens the dataset's credibility as a benchmark for our simulations. In conclusion, Topology Zoo's diversity, real-world focus, and established reputation make it an ideal choice for evaluating Seraph's performance across a broad range of network conditions.

8.4. Applicable scenarios

Our simulations demonstrate Seraph's efficiency in multi-domain SDWAN environments. However, it is important to acknowledge that applications with extremely strict latency requirements might benefit from alternative authentication approaches due to the overhead. To this end, we plan to optimize the verification and solution derivation processes for the Seraph protocol and algorithm as part of our future work. This optimization aims at minimizing overhead while maintaining security, making Seraph a compelling solution for a broader range of scenarios.

9. Conclusion and future work

This paper identifies the potential risk of spoofing attacks in the PKI-based multi-controller authentication schemes widely used in SDWAN. To enhance authentication security, we present an approach based on (t, n)-threshold signatures, which requires "endorsement" nodes to authenticate each node. We investigate the optimal method for setting the "endorsement" mapping, introduce the EMP problem, and formulate it as an integer programming problem. Additionally, we prove the NP-hardness of EMP and present the heuristic algorithm Seraph, which efficiently solves it. Simulation results indicate that Seraph achieves near-optimal performance with a 90% reduction in time usage. We hope our proposed (t, n)-threshold signature-based authentication paradigm will inspire the community to establish a more secure SDWAN network.

We envision two key areas for Seraph's future development. Firstly, we will conduct more comprehensive simulations using diverse compromise probability distributions. This will strengthen our evaluation of Seraph's robustness under various real-world vulnerability scenarios. Secondly, we plan to optimize the Seraph protocol and algorithm to reduce overhead, including verification calculation overhead, communication overhead between nodes, and solution derivation time. This optimization will broaden the range of applications where Seraph remains an efficient solution.

CRediT authorship contribution statement

Wendi Feng: Conceptualization, Formal analysis, Funding acquisition, Validation, Writing – original draft, Writing – review & editing. **Ke Liu:** Software, Conceptualization, Investigation. **Shuo Sun:** Software. **Bo Cheng:** Funding acquisition, Methodology, Resources. **Wei Zhang:** Funding acquisition, Project administration.

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

Data will be made available on request.

Acknowledgments

This work is supported in part by R&D Program of Beijing Municipal Education Commission under grant KM202311232005, National Key R&D Program of China under grant 2022YFC3320903, Open Foundation of State key Laboratory of Networking and Switching Technology (Beijing University of Posts and Telecommunications) under grant SKLNST-2023-1-01, and Young Backbone Teacher Support Plan of Beijing Information Science & Technology University under grant YBT202427. We acknowledge the anonymous reviewers for their insightful comments, which significantly improve the paper's quality.

References

Benamrane, F., Benaini, R., et al., 2017. An east-west interface for distributed SDN control plane: Implementation and evaluation. Comput. Electr. Eng. 57, 162–175.

Chinese National Vulnerability Database, 2024. CNNVD vulnerability reports. https://www.cnnvd.org.cn/home/report.

Csardi, G., Nepusz, T., et al., 2006. The igraph software package for complex network research. Int. J., Complex Syst. 1695 (5), 1–9.

da Silva, J.P., Alchieri, E., Bordim, J., Costa, L., 2020. A secure and distributed control plane for software defined networks. In: International Conference on Advanced Information Networking and Applications. Springer, pp. 994–1006.

Feng, W., Liu, C., Cheng, B., Chen, J., Wan, Z., 2023. An end-host-importance-aware secure service-enabled hybrid sdn deployment. IEEE Trans. Netw. Serv. Manag. 20 (2), 2056–2070. http://dx.doi.org/10.1109/TNSM.2022.3208695.

Ferguson, A.D., Gribble, S., Hong, C.-Y., Killian, C., Mohsin, W., Muehe, H., Ong, J., Poutievski, L., Singh, A., Vicisano, L., et al., 2021. Orion: Google's software-defined networking control plane. In: 18th USENIX Symposium on Networked Systems Design and Implementation. (NSDI 21), pp. 83–98.

Fu, Y., Bi, J., Gao, K., Chen, Z., Wu, J., Hao, B., 2014. Orion: A hybrid hierarchical control plane of software-defined networking for large-scale networks. In: 2014 IEEE 22nd International Conference on Network Protocols. IEEE, pp. 569–576.

Gray, C., Mosig, C., Bush, R., Pelsser, C., Roughan, M., Schmidt, T.C., Wahlisch, M., 2020. BGP beacons, network tomography, and Bayesian computation to locate route flap damping. In: Proceedings of the ACM Internet Measurement Conference. pp. 492–505.

Gude, N., Koponen, T., Pettit, J., Pfaff, B., Casado, M., McKeown, N., Shenker, S., 2008. NOX: towards an operating system for networks. SIGCOMM Comput. Commun. Rev. 38 (3), 105–110.

Guo, Z., Dou, S., Liu, S., Feng, W., Jiang, W., Xu, Y., Zhang, Z.-L., 2022. Maintaining control resiliency and flow programmability in software-defined WANs during controller failures. IEEE/ACM Trans. Netw. 30 (3), 969–984.

Hong, C.-Y., Kandula, S., Mahajan, R., Zhang, M., Gill, V., Nanduri, M., Wattenhofer, R., 2013. Achieving high utilization with software-driven WAN. In: Proceedings of the ACM SIGCOMM 2013 Conference. pp. 15–26.

Hong, C.-Y., Mandal, S., Al-Fares, M., Zhu, M., Alimi, R., Bhagat, C., Jain, S., Kaimal, J., Liang, S., Mendelev, K., et al., 2018. B4 and after: managing hierarchy, partitioning, and asymmetry for availability and scale in google's software-defined WAN. In: Proceedings of the ACM SIGOMM 2018 Conference. pp. 74–87.

Internet2, 2023. About Internet2. https://internet2.edu/community/about-us/.

Johnson, D.B., 1973. A note on Dijkstra's shortest path algorithm. J. ACM 20 (3), 385–388.

Jünger, M., Reinelt, G., Rinaldi, G., 1995. The traveling salesman problem. Handb. Oper. Res. Manag. Sci. 7, 225–330.

Kaur, S., Singh, J., Ghumman, N.S., 2014. Network programmability using POX controller. In: ICCCS International Conference on Communication, Computing & Systems, IEEE, vol. 138, sn, p. 70.

Knight, S., Nguyen, H.X., Falkner, N., Bowden, R., Roughan, M., 2011. The internet topology zoo. IEEE J. Sel. Areas Commun. (JSAC) 29 (9), 1765–1775.

Koponen, T., Casado, M., Gude, N., Stribling, J., Poutievski, L., Zhu, M., Ramanathan, R., Iwata, Y., Inoue, H., Hama, T., et al., 2010. Onix: A distributed control platform for large-scale production networks. In: 9th USENIX Symposium on Operating Systems Design and Implementation. (OSDI 10).

Li, P., Guo, S., Wu, J., Zhao, Q., 2022. BlockREV: Blockchain-enabled multi-controller rule enforcement verification in SDN. Secur. Commun. Netw. 2022.

Li, M., Wang, X., Tong, H., Liu, T., Tian, Y., 2019. SPARC: Towards a scalable distributed control plane architecture for protocol-oblivious SDN networks. In: 2019 28th International Conference on Computer Communication and Networks. ICCCN, IEEE, pp. 1–9.

McKeown, N., Anderson, T., Balakrishnan, H., Parulkar, G., Peterson, L., Rexford, J., Shenker, S., Turner, J., 2008. OpenFlow: enabling innovation in campus networks. SIGCOMM Comput. Commun. Rev. 38 (2), 69–74.

Myers, J., 1997. Simple authentication and security layer (SASL). IETF RFC.

National Vulnerability Database, 2024. NVD– NVD dashboard. https://nvd.nist.gov/general/nvd-dashboard.

Open Networking Foundation, 2022. OpenFlow switch specification. https://opennetworking.org/wp-content/uploads/2014/10/openflow-switch-v1.5.1.pdf.

Osborn, B., McWilliams, J., Beyer, B., Saltonstall, M., 2016. BeyondCorp: Design to deployment at google. USENIX ;login: J. 41, 28–34.

Phemius, K., Bouet, M., Leguay, J., 2014. Disco: Distributed multi-domain sdn controllers. In: 2014 IEEE Network Operations and Management Symposium. NOMS, IEEE, pp. 1–4.

Poutievski, L., Mashayekhi, O., Ong, J., Singh, A., Tariq, M., Wang, R., Zhang, J., Beauregard, V., Conner, P., Gribble, S., et al., 2022. Jupiter evolving: transforming google's datacenter network via optical circuit switches and software-defined networking. In: Proceedings of the ACM SIGCOMM 2022 Conference. pp. 66–85.

Roy, P.K., Sahu, P., Bhattacharya, A., 2022. FastHand: A fast handover authentication protocol for densely deployed small-cell networks. J. Netw. Comput. Appl. 205, 103435.

Ryu SDN Framework Community, 2023. Ryu SDN framework. https://ryu-sdn.org.

Shamir, A., 1979. How to share a secret. Commun. ACM 22 (11), 612–613.

Stathakopoulou, C., Cachin, C., 2017. Threshold Signatures for Blockchain Systems. Tech. rep., vol. 30, Swiss Federal Institute of Technology, p. 1.

Tootoonchian, A., Ganjali, Y., 2010. Hyperflow: A distributed control plane for OpenFlow. In: Proceedings of the 2010 Internet Network Management Conference on Research on Enterprise Networking, vol. 3, pp. 10–5555.

Topology Zoo, 2023. The ulaknet topology. http://www.topology-zoo.org/maps/Ulaknet.jpg.

Tugnait, J.K., 2018. Pilot spoofing attack detection and countermeasure. IEEE Trans. Commun. 66 (5), 2093–2106.

Wikipedia, 2023. Abilene network. https://en.wikipedia.org/wiki/Abilene_Network.

Yang, H., Liang, Y., Yao, Q., Guo, S., Yu, A., Zhang, J., 2019. Blockchain-based secure distributed control for software defined optical networking. China Commun. 16 (6), 42–54.

Yu, H., Wang, H., 2022. Elliptic curve threshold signature scheme for blockchain. J. Inf. Secur. Appl. 70, 103345.

Zheng, Z., Xie, S., Dai, H.-N., Chen, X., Wang, H., 2018. Blockchain challenges and opportunities: A survey. Int. J. Web Grid Serv. 14 (4), 352–375.

Wendi Feng is an associate professor in the School of Computer Science at Beijing Information Science and Technology University. He is also affiliated with the State Key Laboratory of Network and Switching Technology at Beijing University of Posts and Telecommunications, where he obtained his Ph.D. degree under the guidance of Prof. Junliang Chen. From 2018 to 2020, was co-supervised by Prof. Zhi-Li Zhang at the University of Minnesota - Twin Cities. His research interests encompass mobile computing, cloud computing, computer networking, software-defined networks, and network function virtualization. Wendi serves as a reviewer for prestigious journals such as JNCA and TNSM.

Ke Liu received her B.E. from Beijing Information Science and Technology University in 2020. She is currently a master student advised by Prof. Wei Zhang and Wendi Feng at Beijing Information Science and Technology University. Her research interests include cryptographic application, privacy protection, and software-defined network.

Shuo Sun received his B.E. from Beijing Information Science and Technology University in 2021. He is currently a master student advised by Prof. Wei Zhang and Wendi Feng at Beijing Information Science and Technology University. His research interests include monolithic OS kernels and network systems.

Bo Cheng received the Ph.D. degree in computer science from the University of Electronics Science and Technology of China, Chengdu, China, in 2006. He is a Professor with the State Key Laboratory of Networking and Switching Technology, Beijing University of Posts and Telecommunications, Beijing, China. His research interests include service computing, the Internet of Things, and multimedia communications.

Wei Zhang received both his B.E. and Ph.D. degree from Tsinghua University. He is currently a full Professor with School of Computer Science at Beijing Information Science and Technology University. His current research interests include big data storage and security, software and hardware co-design.