International Journal of Service and Knowledge Management
International Institute of Applied Informatics
2023, Vol. 7, No. 1, IJSKM702

Suitable Scalability Management Model for Software-Defined Perimeter Based on Zero-Trust Model

Yangchen Palmo [*], Shigeaki Tanimoto [*],
Hiroyuki Sato [†], Atsushi Kanai [‡]

Abstract

The software-defined perimeter (SDP), a zero-trust model developed by the Cloud Security Alliance, has been attracting attention in the technological industry since its introduction to a world adapting to digital transformation. Many trust models have been introduced to meet the growing demands for cyber security, such as the public key infrastructure, software-defined network, and virtual private network. SDP has particularly gained importance as a zero-trust model because no one in the digital world can be trusted. With the introduction of new models and technical devices, there is now a need to improve newly introduced technology on various grounds when customers adapt to devices. In this work, we discuss how to overcome the current issues of SDP relating to scalability, reliability, usability, etc. As the number of organizations that share information online continues to expand, there is a need for scalable and reliable SDP models that are both easy to maintain and cost efficient for evolving organizations. To meet this need, we proposed several scalable SDP models that enable easier installation management of real networks of organizations with different organizational structures. Specifically, we propose hierarchical, bridge, hybrid, and mesh models. The results of qualitative and quantitative evaluations showed that the bridge model is the most suitable of the four as an extension of SDP.

Keywords: Software-defined perimeter, Digital transformation, Zero-trust model, Scalability

1 Introduction

The need for cyber security is becoming more important than ever as the Internet continues to develop. The following is a well-known metaphor for the need for cyber security. Peter Steiner, an American cartoonist, once published a cartoon in 1993 that prompted the adage "On the Internet, nobody knows you're a dog." Almost three decades later, the adage still stands [1]. In the digital-transformation era, Internet-of-things (IoT) devices are rapidly evolving, and excess data are now being stored and processed in the cloud. A new approach is required to protect the modern network infrastructure, whether it is located in a public or

[*] Faculty of Social System Science, Chiba Institute of Technology, Chiba, Japan
[†] Information Technology Center, The University of Tokyo, Tokyo, Japan
[‡] Faculty of Science and Engineering, Hosei University, Tokyo, Japan

private cloud or on-premises, and to handle the increasing number of mobile or dispersed users. This new approach will involve the software-defined perimeter (SDP), which is a zero-trust model proposed by the Cloud Security Alliance (CSA).

SDP is becoming the de-facto standard for secure network access. SDP focuses on user context, not credentials, to grant access to corporate assets. The SDP market is expected to grow from USD 3,141 million in 2019 to USD 10,613.87 million by 2025 globally at a CAGR of 22.49% during the forecast period [2]. The growing importance of the fast-growing era of digitalization has led to inevitable challenges and consequent demands for the implementation of new services (e.g., scalability and installation management) in the real networks of organizations with different organizational structures along with high security features. Hence, developers and researchers are tasked with creating and improving a tool that can support the emerging needs of organizations and users [3]. Currently offered SDP products are insufficient in terms of scalability, which means they cannot solely meet the future needs of growing organizations. Our literature review shows that current SDP products can support a maximum of 10,000 resources, and while some products can support unlimited scalability, they utilize several third-party clouds that may be too costly for organizations with many end points.

To address the above issues with the current architectural model of SDP, we propose several scalable SDP models that enable easier installation management of real networks. In Section 2, we describe the current status and issues of SDP. In Section 3, we discuss various SDP products. In Section 4, we describe in detail the four scalable SDP models we propose and evaluate them by means of qualitative and quantitative assessments. The results demonstrate that the bridge model is the most suitable of the four scalable SDP models. We conclude the paper in Section 5 with a brief summary and mention of future work.

2 Current Status and Issues of SDP

2.1 Current Status of SDP

SDP is used to control access to resources based on identity by establishing a perimeter through software versus hardware. Specifically, it hides an organization’s infrastructure from outsiders irrespective of its location, while enabling authorized users to access it. SDP focuses on user context, not credentials, to grant access to corporate assets. It creates a secure perimeter based on policies used to isolate services from unsecured networks. Such policies utilize the access control principle of least privilege to secure devices, giving users and devices only the access they require to perform the task at hand. Unauthorized users and devices cannot connect to protected resources. SDP consists of three main components [4]: the SDP controller, the initiating host (IH), and several accepting hosts (AHs). The details of each are given below.

The SDP controller is the most important component of existing SDP as it authorizes and authenticates end users by an AH and the IH searching for network resources. It forms a secure control channel using the mutual TLS authentication (mTLS) protocol.
The IH is the end point of an SDP product that requests the SDP controller to gain access to a resource in the network. When it is authenticated by the SDP controller, the SDP controller establishes a secure data channel between the IH and an AH (resource).
AHs provide resources to the IH after replying permission to SDP controller from IH's request via SDP controller. An IH is authorized by the SDP controller to form a secure data channel

enabled with the mTLS protocol to the AH. When an AH receives a request from the IH, it authenticates it and then the authorized IH secures a data channel for communication [5]– [6].

These components are securely connected to each other over the mTLS protocol using a duplexed connection [2]. Figure 1 shows the architecture of SDP [4].

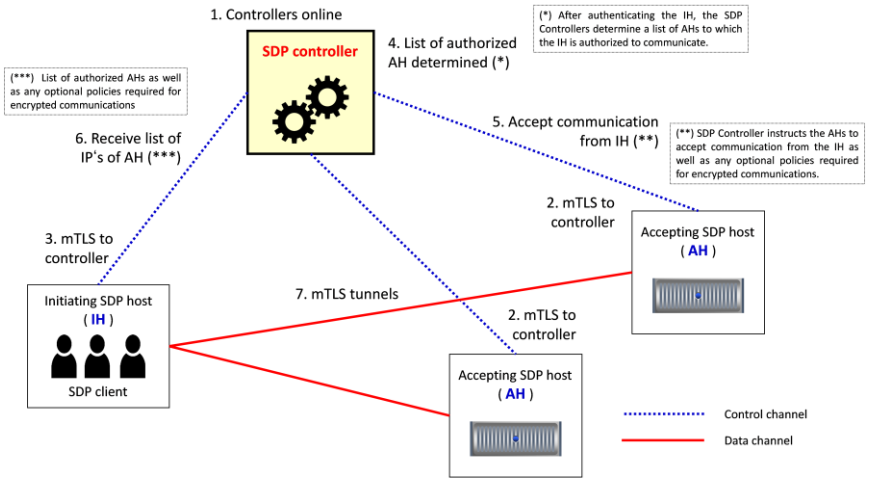


Figure 1: Trust architecture of existing SDP model (figure from [4]).

Figure 2 shows the flow of existing SDP, in which the IH requests the SDP controller for a connection to access the network. The IH is first authenticated by the SDP controller. After successful authentication, the SDP controller checks the number AHs associated with it and then sends an instruction to an AH to accept the connection request of the IH. Finally, a secure connection is formed between the two parties after being authenticated by the SDP controller.

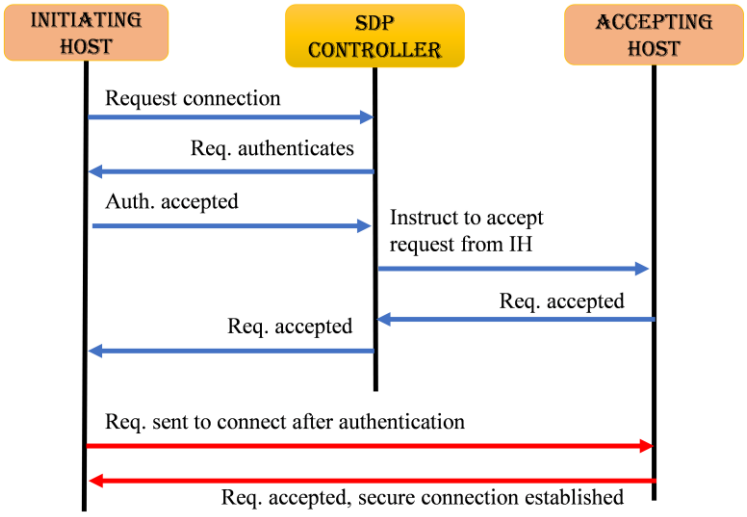


Figure 2: Trust sequence of existing SDP model (figure from [4]).

2.2 Current Issues of SDP

Although SDP is a zero-trust model, the following issues have become apparent when it is implemented in real networks.

Lack of affordable scaling: As organizations require additional user connections and deployments across multiple cloud instances, they need to purchase multiple SDP products for managing end devices. This results in rapid escalation of costs due to additional licenses and more powerful appliances.

Cloud-scale issues: Operationally, once an organization has thousands of internal employees and third parties remotely accessing applications, defining policies and synchronizing them across locations becomes complex. In addition, deploying, configuring, and managing SDP products in tens or hundreds of cloud instances is expensive, time-consuming, and risk prone.

Complex management of SDP for scalability: Additional complexity and difficulty in managing multiple SDP products for ensuring the security of devices arise when installing multiple SDP products for an organization's devices.

Installation of multiple SDP products: Installation of multiple SDP products is costly and difficult, as a problem with the installation of one SDP product may affect others since communications are interlinked.

3 Related Work and Existing SDP Products

3.1 Related Work

First, we describe the results of a literature survey on SDP. E.L.R. Lucion et al. proposed an improvement to the SDP architecture based on single packet authorization (SPA) [7]. Similarly, A. Sallam et al. proposed a security framework for SDP to be utilized in smart homes [8]. A. Moubayed et al. conducted a security evaluation by implementing SDP and described its effectiveness [2], and F.D. Tsokos conducted a similar evaluation [9]. Elsewhere, A. Sallam et al. proposed a scalable security solution by integrating software-defined networking (SDN) and SDP, but no specific SDP scalability was mentioned [10]. Most of these prior works have focused on the implementation and security evaluation of the SDP architecture, with little attention paid to scalability.

At the same time, several products related to SDP have been released, and many descriptions of scalability have been made within this context. The following subsections provide an overview of these SDP products, with a particular focus on the scalability.

3.2 Existing SDP Model-Based Products

Examples of commercial models based on the SDP model are discussed below. Table 1 lists the extensibility based on the specifications of each model.

3.2.1 Cisco Software-Defined Access

Cisco software-defined access (SDA) is a network security solution within the Cisco Digital Network that integrates intent-based network principles. The core benefit of Cisco SDA is its provisioning of visibility-based automated end-to-end segmentation to separate users, devices, and application traffic without the need to redesign the underlying physical network. By automating the use access policy, organizations can establish the appropriate polices for the devices connected over the network. Without compromising security, such policies provide a consistent

user experience over the network. Cisco SDA comes with specified features including mobility, visibility, policy determination, IoT/Cloud, and data integration, but there is no specification of scalability, such as the number of end points that a model can support [11].

3.2.2 Perimeter 81 SDP

Perimeter 81 SDP is a virtual private network (VPN) software built on the architectural principle of SDP that simplifies online security for employees by enabling them to work through secure remote access. Its core principle is to provide a simple and secure network with fast and easy deployment, cloud, and application access to a modern mobile workforce by transforming traditional network security technology into one unified zero-trust network as a service. It provides mobility and discards the need for expensive external equipment or installation. The target customers are small-to-medium businesses. Scalability can range from less than five devices to over 100. Thus, if users wish to apply it in an organization that is expanding globally, they may need to adopt multiple products to achieve scalability, which is expensive and cost inefficient [12].

3.2.3 Twingate SDP

Twingate SDP enables rapid implementation of a zero-trust model that provides superior security and easy maintainability for organizations of all sizes to secure remote access to their private applications, data, and environments, whether they are on-premises or in the cloud. By delivering a cloud-based service, Twingate SDP enables IT teams to easily configure the SDP without changing the infrastructure. It also offers a central management of user access to internal apps, whether they are on-premises or in the cloud. It was developed to make life for developers, IT teams, and end users easier. It is an attractive alternative to outdated, non-secure, and difficult-to-maintain corporate VPNs, which were not built to handle a world in which "work from anywhere" and cloud-based assets have increasingly become the norm. Small and medium business as well as mid-market organizations also need to provide secure remote access to developers and other remote workers. The unique points of Twingate SDP are that it is user-friendly for both clients and administrators, is fast and easy to deploy, does not require a public gateway, offers effortless scalability, places a low burden on IT teams, and achieves zero-trust access and scalability. It is easily scalable with minimal maintenance from 10 to 10,000 resources. However, it does not meet the desired low-cost scalable SDP requirements for large organizations [13].

3.2.4 Cato SDP

Cato developed an SDP product based on scalability and optimization that enables secure access to all users and applications. Specifically, their SDP product provides global, cloud-scale, optimized, and secure access to anyone through an integrated client-based and clientless remote access solution as part of the Cato Cloud. Users benefit from optimized and secure access to all applications on-premises and in the cloud while at home or on the road. Cato enforces strong authentication and granular access control as well as deep packet inspection of all traffic to protect against threats. While Cato's global, cloud-scale platform seamlessly supports any number of users and applications globally, it is implemented as a third-party abstract cloud, which can be costly for organizations with many employees [14].

Table 1: Scalability specifications of SDP products for real networks.

SDP product	Scalability specifications (end points / resources)
1) Cisco SDA	No description
2) Perimeter 81	< 5 over 100
3) Twingate	10–10,000 resources
4) Cato	Provides third-party cloud (costly)

3.3 Scalability Issues of Current SDP Products

While there are currently several SDP products on the market, as discussed above, scalability has not yet been fully considered. At present, the scalability is about 10,000 units even for products for which scalability is not specified (see 1) in Table 1) and products for which scalability is specified (see 2) and 3) in Table 1), which is not necessarily sufficient considering the future development of IoT and other technologies. As for 4) in Table 1, while scalability is ensured by linking with a third-party cloud, this may prove too costly [15].

In light of this background, it is important to consider the scalability of SDP from an economic viewpoint when assuming the future development of the IoT.

4 Proposed Scalable SDP Models

In this section, we propose four scalability models for SDP with reference to the public key infrastructure (PKI) [16], which has been established as a practical network for trust models. In general, the trust model of PKI introduces the architecture, as shown in Table 2 [17].

Table 2: Scalability specifications of SDP products for real networks.

PKI trust model	(1) Explanation of model	(2) Adaptability to scalable SDP
Single CA model	A single CA issues certificates to all users.	Not adaptable
Hierarchical model	Multiple CAs are configured in a hierarchical (tree) structure.	Adaptable
Web model	A list of root CAs is embedded in the client application beforehand and is then used in Web browsers.	Not adaptable
Mesh model	Multiple CAs are connected through mutual authentication.	Adaptable
Bridge CA model	Multiple CAs are connected via a bridge CA.	Adaptable

In general, SDP is the same kind of trust model as PKI. Therefore, when considering the scalability of SDP, we decided to use three of the PKI models in Table 2. Specifically, we applied the hierarchical, mesh, and bridge models to take advantage of their networking capabilities. We also developed a new hybrid model that combines the hierarchical and bridge models.

Our proposed architecture adds new functions to the existing SDP controller to ensure the scalability of SDP. In other words, our proposed SDP controller has network functions

(layering, bridging, etc.) that extend SDP, in addition to the conventional functions to provide SDP authentication. In the following sub-sections, we describe our proposed scalability architecture for SDP using these models [18].

4.1 Hierarchical Model

As shown in Fig. 3 the hierarchical model, or tree model, is based on the hierarchical PKI architectural trust model, in which there is a root SDP controller at the top that provides and stores all the information of the lower/intermediate child controllers, and two parent SDP controllers, which are authenticated and authorized by the root SDP controller. This scalable architectural model helps segregate the accessibility to the resources of an organization depending on each resource's importance, where the end points that accesses resources at the CEO level is authenticated and authorized by the root SDP controller, level 2 resource access is authenticated by the parent SDP controller, and so on. The hierarchical model enables tight control over the level of importance of the resources to be accessed. While this model is scalable, it may suffer from network performance issues, as shown in Fig. 4.

Figure 4 shows the connection between an AH and the IH of two different parent SDP controllers. If the IH wants to form a connection to the AH of another parent SDP controller, it first sends a connection request to its parent SDP controller, which then authenticates the IH. After successful authentication, the parent SDP controller sends a request to the root SDP controller with the address of the IH. The root SDP controller verifies the parent SDP controller of the associated AH and sends an instruction for the connection to be established.

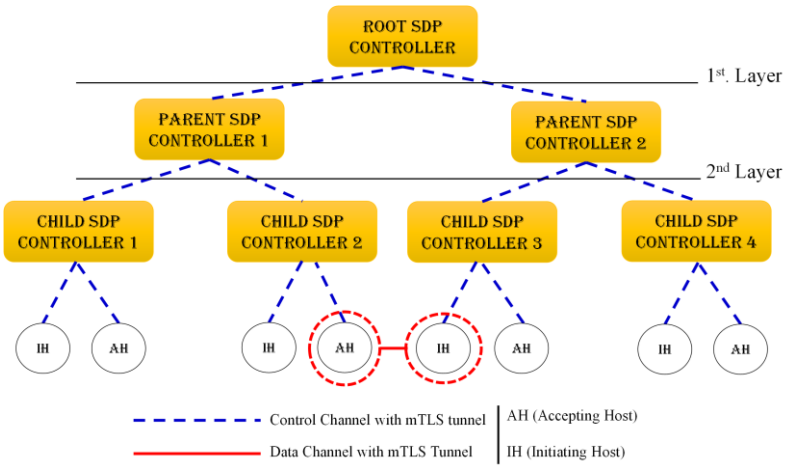


Figure 3: Proposed hierarchical SDP model.

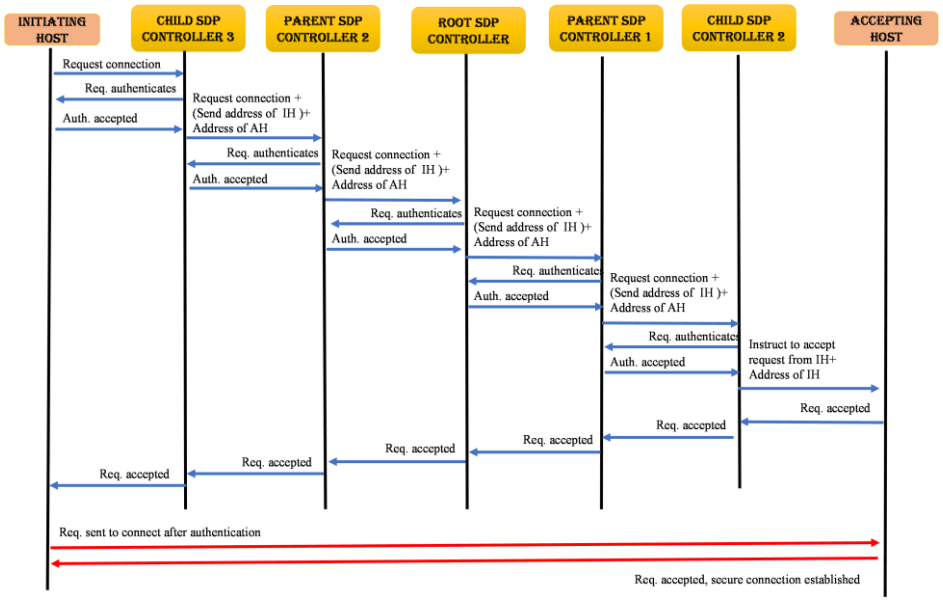


Figure 4: Flow of proposed hierarchical SDP model.

4.2 Bridge SDP Model

As shown in Fig. 5, the proposed bridge model consists of a group of hierarchical models and an existing SDP. There is a peer-to-peer (P2P) connection between the root SDP controller of one hierarchical model and another hierarchical model or a P2P connection between the SDP controller of the existing SDP and the root SDP controller of one hierarchical model. This enables two SDP controllers to authenticate each other and share resources. An implementation such as this can scale the resource-sharing capability and balance the load among SDP controllers to authenticate and authorize the multiple end points connected to them. In this model, each intermediate SDP controller can only communicate with its root and child; if they want access to the information of the bridged SDP controller, they need to send a request to the relevant root SDP controller. Additional flexibility and interoperability between organizations are the primary advantages of the bridge model. The bridge model is highly cost efficient and can be applied to major organizations. It is also easy to manage.

Figure 6 shows the flow of the bridge model when the IH connected to the bridged SDP controller requests a connection to the AH connected to the hierarchical model. The IH sends the request to the bridged SDP controller, which authenticates it and then broadcasts the request to another SDP controller connected to it. The root SDP controller associated with the corresponding AH authenticates the bridged SDP controller connected to it and checks for the list of its AHs and their respective parent SDP controllers. It then forwards the instruction to accept the request from the AH connected to the bridged SDP controller to its lower level. Finally, a secure communication channel is formed between the two parties.

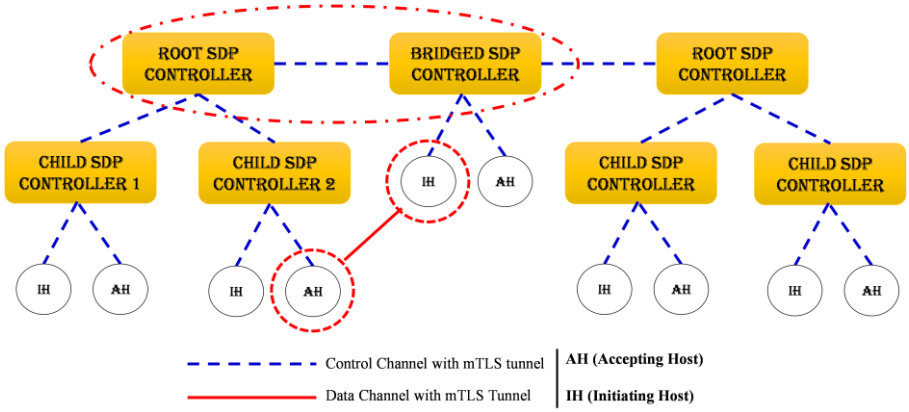


Figure 5: Proposed bridge SDP model.

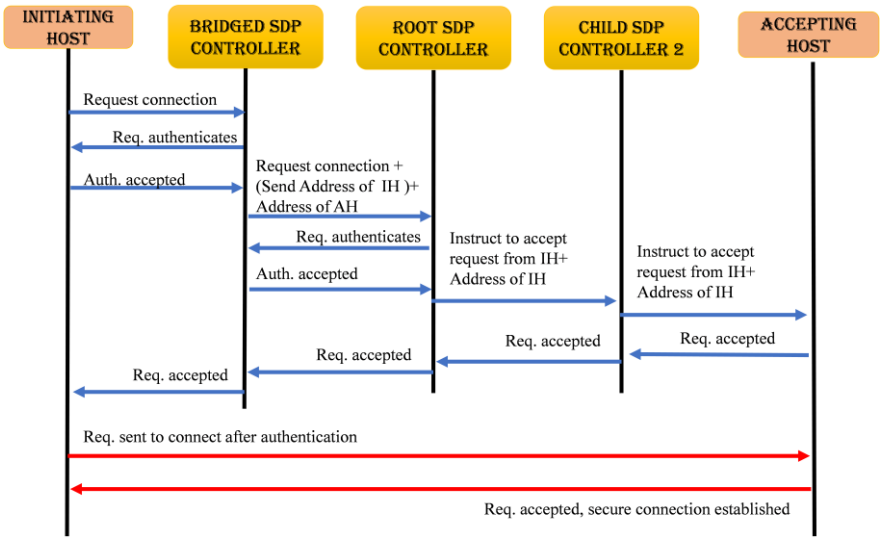


Figure 6: Flow of proposed bridge SDP model.

4.3 Hybrid SDP Model

As shown in Fig. 7, the hybrid model is a combination of the hierarchical model and bridge model. As such, it is highly scalable and can be applied to major organizations. With this model, we can achieve a high degree of flexibility, scalability, and maintainability of the network architecture. The downside is that it is costlier than the other models and can be quite difficult to manage. In this model, a child/parent SDP controller can be directly connected to another child/parent SDP controller.

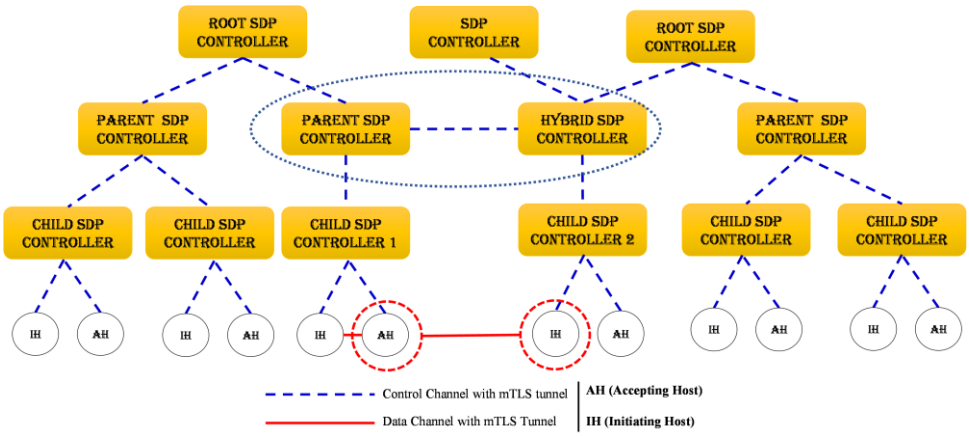


Figure 7: Proposed hybrid SDP model.

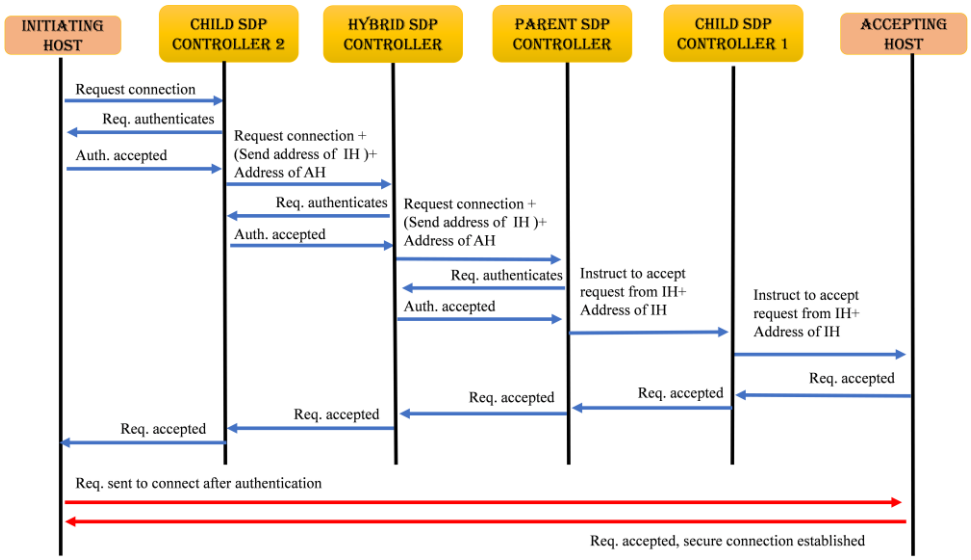


Figure 8: Flow of proposed hybrid SDP model.

Figure 8 shows the connection between the AHs and the IHs of two different parent controllers. If an IH wants to form a connection to an AH of another parent controller, it first sends a connection request to its parent SDP controller, which then authenticates the IH. After successful authentication, the parent SDP controller broadcasts the request of connection to other SDP controllers connected to it with the address of the IH. The root SDP controller verifies the parent SDP controller of the associated AH and sends the instruction for the connection to be established.

4.4 Mesh SDP Model

As shown in Fig. 9, the mesh model enables mutual authentication for the resources of individual SDP controllers. Many root SDP controllers of the hierarchical model and SDP controllers of the existing SDP are inter-connected in a network. This model has high network speed, as all the

SDP controllers are inter-connected and can share resources with much ease and flexibility. This model is also easy to manage, easy to install, and highly scalable. However, it may be costly due to the large number of connections that need to be installed among the SDP controllers.

Figure 10 shows the flow of the mesh model. The IH of one SDP controller requests access to the AH of another meshed SDP controller by sending it a connection request. The SDP controller then broadcasts the connection request to the associated AH and receives a response from the associated meshed SDP controller with authentication. After that, the meshed SDP controller forwards the instruction to the AH to accept the connection request. In this way, a secure data communication channel is established.

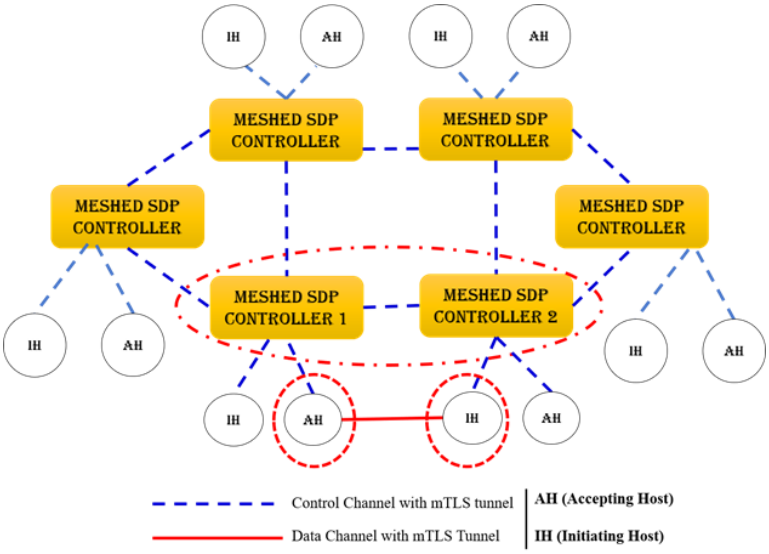


Figure 9: Proposed mesh SDP model.

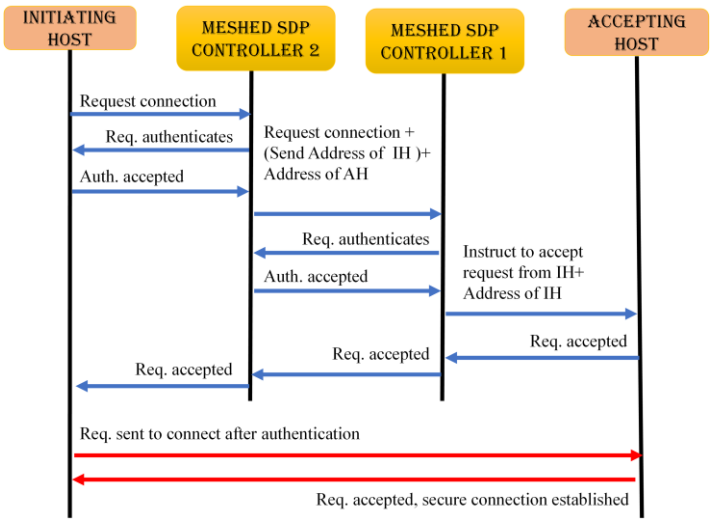


Figure 10: Flow of proposed mesh SDP model.

4.5 Qualitative Evaluation using Desktop Simulation

The results of our qualitative analysis of the proposed SDP models from the perspective of a typical network implementation are shown in Table 3. The evaluation items were scalability, ease of installation, cost, ease of management, and scalability overhead, which were ranked in three levels (H: high, M: middle, and L: low) on the basis of literature review and discussion among the authors.

The results demonstrate that, among the four proposed scalable SDP models, the bridge model is the most suitable due to its high scalability, high ease of implementation, low cost, high ease of management, and low overhead. The mesh model has the lowest ease of implementation, the highest cost, and the lowest ease of management because it requires all SDP controllers to implement the extensions. For the hierarchical model, the disadvantage is the high scalability overhead. In the case of the hybrid model, the disadvantage is the high cost due to the combination of hierarchical and bridge models.

Table 3: Qualitative evaluation results of proposed SDP models.

Scalable SDP model	Scalability	Ease of installation	Cost	Ease of management	Overhead of scalability	Overall evaluation
A: Hierarchical	H	M	M	H	H	M
B: Bridge	H	H	L	H	L	H
C: Hybrid	H	M	H	L	M	M
D: Mesh	H	L	H	L	L	L

(H: high, M: middle, L: low)

From these findings, we conclude that the bridge model can be applied to any type of organization. However, note that these results are for a qualitative analysis based on a desk simulation rather than an actual implementation.

4.6 Quantitative Evaluation

Next, for our quantitative evaluation, we use the average number of signals required from authentication to the final connection with the host in the zero-trust model. The specific calculation of the average signal count is based on the sequence flow diagram of each proposed model.

4.6.1 Hierarchical Model

We utilize the service sequence diagram of the hierarchical model in Fig. 4 to calculate the number of signals until the IH requests service from the SDP controller and is granted it by the AH through the SDP controller. The specific number of signals in the hierarchical model is shown in Fig. 11. This number is then used to calculate the time until the IH is authorized to service the AH.

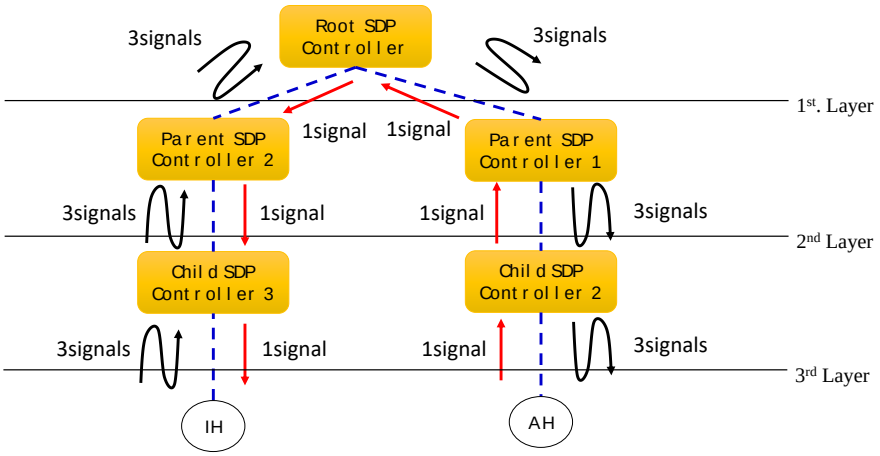


Figure 11: Number of signals before IH is authorized for service from AH (hierarchical model).

The number of signals is deduced as follows.
Step 1) Signaling from IH to SDP controller: 3 × n (n: number of layers in the hierarchy)
Step 2) Signaling from SDP controller to AH: 3 × n
Step 3) Signaling from AH to SDP controller: 1 × n
Step 4) Signaling from SDP controller to IH: 1 × n

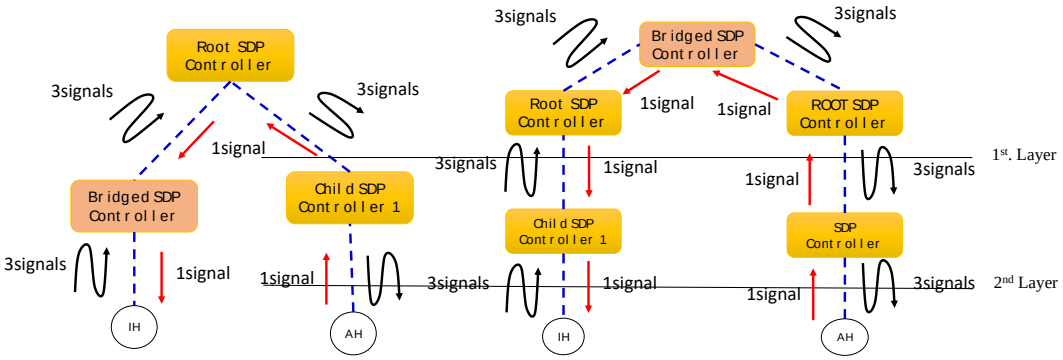
On the basis of the above, the number of signals in the hierarchical model is calculated as

Number of signals = 3 × n + 3 × n + n + n = 8 × n, (1)

where n is the number of levels.

4.6.2 Bridge Model

Similarly, in the case of the bridge model, the number of signals until the IH receives the service authorization of the AH is calculated from the sequence diagram in Fig. 6. The number of signals in this case depends on the number of paths through the bridge. By referncing the shortest and longest paths in Fig. 12(a) and (b), respectively, we obtain the number of signals for the bridge model.



(a) Shortest path signal routing **(b) Longest path signal routing**

Figure 12. Number of signals before IH is authorized for service from AH (bridge model).

The number of signals is deduced as follows.

(a) Shortest path case
Step 1) Signaling from IH to SDP controller: 3 × 2 (layer-independent due to direct bridge connection)
Step 2) Signaling from SDP controller to AH: 3 × n
Step 3) Signaling from AH to SDP controller: 1 × n
Step 4) Signaling from SDP controller to IH: 1 × 2 (layer-independent due to direct bridge connection)

On the basis of the above, the number of signals in the shortest path of the bridge model is calculated as

$$\textbf{Number of signals} = 3 \times 2 + 3 \times n + n + 1 \times 2 = 4 \times n + 8. \quad (2)$$

(b) Longest path case
Step 1) Signaling from IH to SDP controller: 3 × (n + 1) (because there are layer-dependent parts (3 × n) and non-dependent parts (3))
Step 2) Signaling from SDP controller to AH: 3 × (n + 1) (same reason as Step 1)
Step 3) Signaling from AH to SDP controller: 1 × (n + 1) (same reason as Step 1)
Step 4) Signaling from SDP controller to IH: 1 × (n + 1) (same reason as Step 1)

On the basis of the above, the number of signals in the longest path of the bridge model is calculated as

$$\textbf{Number of signals} = 3 \times (n + 1) + 3 \times (n + 1) + 1 \times (n + 1) + 1 \times (n + 1) = 8 \times n + 8. \quad (3)$$

4.6.3 Hybrid Model

In the case of the hybrid model, the number of signals until the IH is authorized for service from the AH is calculated from the sequence diagram in Fig. 8. As in the bridge model, the number of signals depends on the number of paths through the SDP controller in the hybrid form. By referencing the shortest and longest paths in Fig. 13(a) and (b), respectively, we obtain the number of signals for the hybrid model.

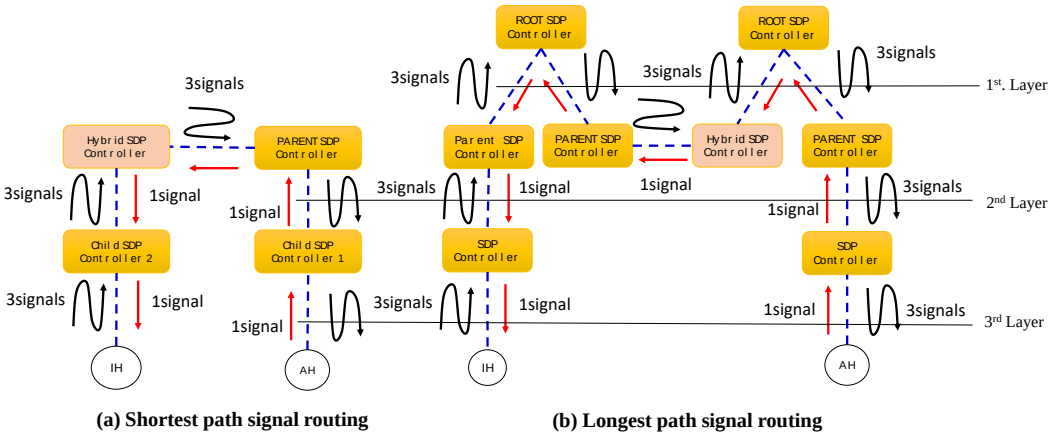


Figure 13. Hybrid model architecture for signal routing from AH-IH.

The number of signals is deduced as follows.

(a) Shortest path case
Step 1) Signaling from IH to hybrid SDP controller: 3 × (n + 1) (because the part that is connected to the hybrid SDP controller is layer-independent)
Step 2) Signaling from hybrid SDP controller to AH: 3 × (n + 1) (same reason as Step 1)
Step 3) Signaling from AH to hybrid SDP controller: n + 1 (same reason as Step 1)
Step 4) Signaling from hybrid SDP controller to IH: n + 1 (same reason as Step 1)

On the basis of the above, the number of signals in the shortest path of the hybrid model is calculated as

$$\textbf{Number of signals} = 3 \times (n + 1) + 3 \times (n + 1) + n + 1 + n + 1 = 8 \times n + 8. \quad (4)$$

(b) Longest path case
Step 1) Signaling from IH to hybrid SDP controller: 3 × (2 × n + 1) (because the part that is connected to the hybrid SDP controller is layer-independent; i.e., the layer is doubled because there is a point where it folds back at the root SDP controller)
Step 2) Signaling from hybrid SDP controller to AH: 3 × (2 × n) (same reason as Step 1)
Step 3) Signaling from AH to hybrid SDP controller: 2 × n (same reason as Step 1)
Step 4) Signaling from hybrid SDP controller to IH: 2 × n + 1 (same reason as Step 1)

On the basis of the above, the number of signals in the longest path of the hybrid model is calculated as

$$\textbf{Number of signals} = 3 \times (2 \times n + 1) + 3 \times (2 \times n) + 2 \times n + 2 \times n + 1 = 16 \times n + 4. \quad (5)$$

4.6.4 Mesh Model

Using the service sequence diagram of the mesh model in Fig. 10, we calculate the number of signals until the IH requests a service from the SDP controller and the service is granted by the AH. The specific number of signals in the mesh model is shown in Fig. 14. By referencing this number, we calculate the number of signals until the IH is granted service by the AH.

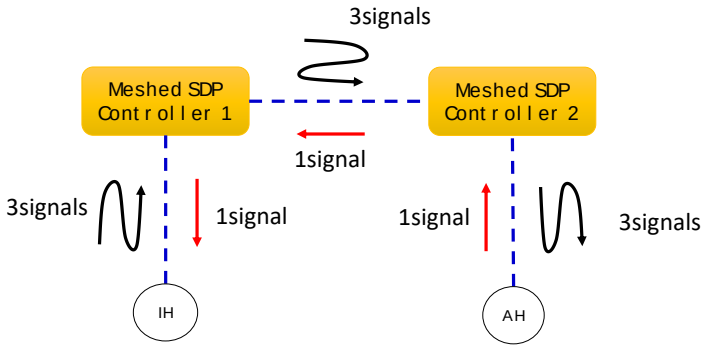


Figure 14. Mesh model architecture for signal routing from AH-IH.

The number of signals is deduced as follows.
Step 1) Signaling from IH to AH: 9
Step 2) Signaling from AH to IH: 3
On the basis of the above, the number of signals in the mesh model is calculated as

Number of signals = 9 + 3 = 12. (6)

4.6.5 Summary of quantitative evaluation

In this section, we summarize the quantitative evaluation of each model proposed in the previous section based on the number of signals from IH to the service authorization by AH. Table 4 lists the number of signals for each proposed model.

As we can see, the layer-independent mesh model results in the lowest number of signals. In the case of the mesh model, the number of signals is also the lowest because of the one-to-one connection between the IH or AH and the SDP controller, which can be considered a reasonable result. However, referring back to Table 3, the mesh model is not optimal when scalability is considered.

The next best model for evaluating the number of signals is the short path case of the bridge model, depending on the number of hierarchies (for n > 2). Since the number of hierarchies becomes larger when scalability is considered, the bridge model is more realistic and also more suitable quantitatively from this point of view. The qualitative evaluation results in Table 3 also show that the model is cost effective and easy to maintain. When we consider both the qualitative and quantitative evaluations, the bridge model is the best among the four proposed models.

Table 4: Average signaling equations derived for the proposed models

Scalable SDP model		Number of signals
4.6.1 Hierarchical SDP model		8 × n
4.6.2 Bridge SDP model	Shortest path case	4 × n + 8
	Longest path case	8 × n + 8
4.6.3 Hybrid SDP model	Shortest path case	8 × n + 8
	Longest path case	16 × n + 4
4.6.4 Mesh SDP model		12

n: number of layers in the hierarchy

5 Conclusion and Future Work

Constant changes in the characteristics of cyber attacks have challenged developers to come up with new security tools to overcome such threats, which has led to the introduction of SDP. However, the existing SDP models and related products are still insufficient in terms of scalable installation management when applied to the real networks of growing organizations with bil-

lions of end devices to be managed securely.

In response, we developed four new scalable SDP models that enable scalability features to be used in conjunction with original authentication features for growing organizations. Our models can be applied to various organizations depending on their working architectures or structural models by analyzing the flow of information between resources and end points.

Our qualitative and quantitative evaluations of the proposed architecture showed that the bridge model is the best among the four, as it can be applied to any type of organization irrespective of structure, financial status, or functionality. Different models can be selected in accordance with the needs and characteristics of a given organization, as all four models support the scalable installation management of real networks.

For future work, we will conduct a practical study for each proposed model and examine their associated risks to expand our research in accordance with the needs of organizations and individuals. By using a risk-analysis method combined with practical analysis, we can improve the functionality of the proposed models.

Acknowledgement

This work was supported by JSPS KAKENHI Grant Number JP 19H04098.

References

[1] Leo Taddeo, Why Security needs a Software Defined Perimeter, [Online] Available: https://www.darkreading.com/why-security-needs-a-software-defined-perimeter/a/d-id/1332666

[2] A. Moubayed, et al., Software-Defined Perimeter (SDP): State of the Art Secure Solution for Modern Networks, [Online] Available: https://ieeexplore.ieee.org/stamp/stamp.jsp?tp=&arnumber=8863736&tag=1

[3] A. Froehlich, Let's have a look at the top 7 networking technologies and architecture trends you should expect to see as we enter the New Year, [Online] Available:

https://www.networkcomputing.com/networking/7-network-trends-you-can-expect-2020

[4] CSA Software Defined Perimeter Working Group, SDP Specification 1.0, 2014, [Online] Available: https://cloudsecurityalliance.org/artifacts/sdp-specification-v1-0/

[5] M. Henderson, et al., Modelling Trust Structures for Public Key Infrastructures, [Online] Available: http://www.math.udel.edu/~coulter/papers/acisp.pdf

[6] S. Tanimoto, et al., Proposal of a perimeter line management method for fog and edge computing with SDP concept, The 23rd International Conference on Network-Based Information Systems (NBiS-2020), AISC 1264, pp.290-302, Springer, DOI: 10.1007/978-3-030-57811-4_27, 2020

[7] E.L.R. Lucion, et al., Software Defined Perimeter: improvements in the security of Single Packet Authorization and user authentication, 2018 XLIV Latin American Computer Conference (CLEI), 2018

[8] A. Sallam, et al., Securing Smart Home Networks with Software-Defined Perimeter, 2019 15th International Wireless Communications & Mobile Computing Conference (IWCMC), pp.1989- 1993, 2019

[9] F. D. Tsokos, Development of a Software Defined Security Perimeter, University of the Thessaly, 2018, https://core.ac.uk/download/pdf/159408436.pdf

[10] A. Sallam, et al., On the Security of SDN: A Completed Secure and Scalable Framework Using the Software-Defined Perimeter, IEEE Access (Volume: 7), pp.146577-146587, 2020

[11] Cisco, Cisco Software-Defined Access: Introducing an Entirely New Era in networking Solution overview, [Online] Available: https://www.cisco.com/c/en/us/solutions/collateral/enterprise-networks/software-defined-access/solution-overview-c22-739012.html

[12] Perimeter 81, Securing a Digital Work Space, [Online] Available: https://www.perimeter81.com/solutions/software-defined-perimeter

[13] Twingate, Twingate provides a simple, modern approach to secure online work, [Online] Available: https://www.twingate.com/

[14] Cato Networks, Software Defined Perimeter, [Online] Available: https://www.catonetworks.com/sdp/

[15] A. Welekwe, Six best software defined perimeter, [Online] Available: https://www.comparitech.com/net-admin/software-defined-perimeter-software/

[16] Y. Miyakawa, et al., Current Status of Japanese Government PKI Systems, [Online] Available: https://link.springer.com/chapter/10.1007/978-3-540-69485-4_8

[17] IPA, PKI Related Technical Information, 5 Trust Model, (Japanese Edition), [Online] Available: https://www.ipa.go.jp/security/pki/051.html

[18] Y. Palmo, et al., A Consideration of Scalability for Software Defined Perimeter Based on the Zero-trust Model, 2021 10th International Congress on Advanced Applied Informatics (IIAI-AAI), pp.717-724, 2021