scientific reports

Check for updates

OPEN

A data plane security model of SR-BE/TE based on zero-trust architecture

Liang Wang[1], Hailong Ma[1]✉, Ziyong Li[1], Jinchuan Pei[1], Tao Hu[1] & Jin Zhang[2]

Facing the untrusted threats of network elements and PKI/CA faced by SR-BE/TE (Segment Routing-BE/TE) data plane in the zero-trust network environment, firstly, this paper refines it into eight specific security issues. Secondly, an SR-BE/TE data plane security model ZbSR (ZTA-based SR) based on zero-trust architecture is proposed, which reconstructs the original SR control plane into a "trust-agent" two-layer plane based on 4 components of the controller, agent, cryptographic center and information base. On one hand, we distinguish between the two segment list generation modes and proposes corresponding data exchange security algorithms, by introducing north–south security verification based on identity authentication, trust evaluation, and key agreement before the terminal device establishes an east–west access connection, so reliable data exchange between terminal devices can be realized. On the other hand, for the network audit lacking SR-BE/TE, a network audit security algorithm based on solid authentication is proposed. By auditing the fields, behaviors, loops, labels, paths, and SIDs of messages, threats such as stream path tampering, SID tampering, DoS attacks, and loop attacks can be effectively detected. Finally, through the simulation test, the proposed model can provide security protection for the SR data plane with a 19.3% average incremental delay overhead for various threat scenarios.

Multiprotocol Label switching (MPLS), proposed by Internet Engineering Task Force (IETF), is a new data exchange standard for efficient transmission of data guided by labels on open communication networks. The essence of MPLS is to use label distribution technology. Mapping IP addresses into simple, fixed-length labels and using the labels to forward data has been widely deployed on wide area networks. However, the control plane of MPLS relies on complex Label Distribution Protocol (LDP) and Resource Reservation Protocol (RSVP) protocols, which leads to poor scalability and difficulty in deployment and maintenance.

Therefore, Segment Routing (SR) was born out of MPLS, and it revolutionized MPLS by deleting LDP and RSVP label distribution protocols and adding source routing features, which significantly improved the simplicity of network co24ntrol and the ability of super-large-scale networking[1]. Because of its stateless, easy deployment, cross-domain, and other excellent features, SR fully embodies the new network development concept of "application-driven network". Now, it has been supported by OpenDaylight open source SDN controller and Linux system, which can strongly support the end-to-end traffic scheduling of IP network and programmable reconfiguration of the software-defined network[2] and become the key technology of SDN/NFV(Software Defined Network/Network Functions Virtualization) in the next step[3].

SR can be divided into SR-MPLS and SRv6[4] according to the data plane encapsulation method, and can also be divided into SR-BE(SR Best Effort) and SR-TE(SR Traffic Engineering) according to the implementation mode, in which the SR-BE mode determines the Segment list by the head node through the IGP(Interior Gateway Protocol) shortest path; in SR-TE mode, the SDN controller or SR PCE sends the Segment list to the head-end node through PCEP(Path Computation Element Communication Protocol), BGP(Border Gateway Protocol), BGP-LS, XML, and NETCONF, or the head-end node can automatically generate the Segment list through ODN(On-Demand Next-hop) mechanism, or the operator can explicitly configure it through CLI, NETCONF, etc. The source routing characteristics of SR enable it to specify the key nodes of the traffic path at the head node, and guide the traffic through any path based on the Segment ID (SID), which achieves a delicate balance between control granularity and control simplicity, but also brings new available conditions for attackers to accurately attack the specified links or devices in the domain. However, at present, the academic circles focus on the functional research of SR, such as principle analysis[5], protocol expansion[6], technology implementation[7],

[1]National Digital Switching System Engineering and Technology Research Center, PLA Strategic Support Force Information Engineering University, Zhengzhou 450003, China. [2]Network Communication and Security Purple Mountain Laboratory, Nanjing 210000, China. ✉email: longmanclear@163.com

and system integration[8], the research on its security is insufficient, especially the systematic solutions to the threats such as message forgery, identity fraud and node failure faced by its data plane is less.

What aggravates the threat faced by SR is that the network environment is also accelerating the transition to weak trust and zero trust, and Zero-Trust Architecture (ZTA) emerges at the historic moment. This architecture focuses on replacing the default trust granted by traditional network boundary security models (such as firewall, NAT, VPN) through dynamic trust based on multi-factor authentication and fine-grained authorization, to change the security boundary form between the host and the object from fixed hardware to software definition, to fundamentally solve zero trust threats. The progressive nature of ZTA is mainly reflected in the following aspects: the network boundary security models grant long-term trust based on single verification, lacks the traffic inspection inside the boundary, and is challenging to resist threats such as traffic eavesdropping and loopback attacks; while ZTA grants temporary trust based on each verification, which changes the paradigm of trust granting, and implements the security policy of "binding users and devices as network agents, authenticating and granting trust based on network agents, and dynamically authorizing according to trust"[9], replacing fixed boundaries with dynamic identities, and blocking the lateral movement of attackers[10].

Focusing on providing the SR-BE/TE data plane security scheme for the zero-trust network environment, this paper applies ZTA to SR-BE/TE and proposes a data plane security model of SR-BE/TE based on ZTA: ZbSR (ZTA-based SR), which focuses on the security of data plane switching device. In this model, the original SR control plane is transformed into a trust plane and an agent plane based on four security components: controller, key center, agent, and information base. Aiming at the two untrusted functions of data exchange and network audit of SR-BE/TE data plane, two list acquisition modes, Segment list generated by switching device in SR-BE/TE and list issued by controller in SR-TE, are distinguished, and the corresponding data exchange security algorithms based on trust evaluation are proposed respectively, that is, before the data exchange in east–west direction data plane terminal device via routing device of SR-BE/TE, in the north–south direction, firstly, it carries out identity authentication based on device information comparison, trust evaluation based on recommendation trust reasoning, and key negotiation based on encryption and digital signature to support it to establish a trusted connection; besides, this paper proposes a network audit security algorithm based on strong authentication, which can detect the attack representations of different threats by auditing the fields, behaviors, loops, labels, paths and SID information of the messages in various directions. Through simulation test and analysis, the proposed model is helpful to deal with different threats faced by SR-BE/TE data plane.

In summary, the main contributions of this article are as follows:

(1) 8 kinds of SR-BE/TE data plane security problems facing zero-trust network environment are put forward, and the technical combination basis of SR and ZTA basic function models is summarized;
(2) A security model of SR-BE/TE data plane based on ZTA is designed and implemented. For the untrusted function of the SR data plane, two security algorithms of data exchange and network audit are proposed, and 4 sub-algorithms of identity authentication, trust evaluation, key agreement, and loop audit are proposed;
(3) The effectiveness of the proposed architecture is proved by cost analysis and simulation, and the shortcomings of high cost were also found.

This paper mainly consists of 5 sections, among which, the second section summarizes and puts forward SR native security mechanism, primary routing security mechanism, SR-BE/TE data plane security problem for the zero-trust network environment, and basic function models of SR and ZTA. The third section expounds the architecture design, component functions, security algorithms, security overhead, and so on of the ZbSR model. In the fourth section, based on the EVE-NG simulation environment, the security performance and overhead of the ZbSR model and the other SR/SDN function models are compared and tested. The fifth section summarizes the full text and looks forward to the following research.

Related works

At present, as there is no molding solution to the threats faced by SR in the zero-trust network environment, this section mainly summarizes 7 kinds of SR native security mechanisms and 6 kinds of existing mainstream routing security mechanisms, puts forward 8 kinds of SR-BE/TE data plane security problems for the zero-trust network environment, and analyzes the coupling basis of SR and ZTA basic function models.

SR native security mechanism and primary routing security mechanism. Literature[2,11] points out the native security mechanisms adopted by SR, summarized into 7 categories in this paper. As shown in Table 1, these security mechanisms can't cope with Zero-trust security threats such as control plane message tampering, denial of service attack, topology based on devices in the domain, and label detection.

Literature[12–22] puts forward a variety of mainstream routing security mechanisms, summarized into 6 categories in this paper, as shown in Table 2. These mechanisms did not consider the label and source routing characteristics of the SR network, and could not directly migrate to the SR scene, nor did they consider and deal with the threats they faced as a whole, so their universality was limited.

The comparison between the above scheme and the scheme proposed in this paper is shown in Table 3.

SR-BE/TE data plane security for zero-trust network environment. Based on the above analysis and the premise that "no user, device or application are trusted in the zero-trust environment", this paper defines the SR-BE/TE data plane security problem in the zero-trust network environment as threats of untrusted

security mechanism	Implementation method	Threat against
Source routing[2,11]	The head node of the flow encapsulates the label stack to specify the flow path	Malicious drainage
Trust domain[2,11]	Only the source route is used in the domain, and the source route information is cleared by setting the C-flag flag in SRH when the data packet leaves the domain	Label leakage
Package validation[2,11]	RFC8754 stipulates that the optional TLV (Type-Length-Value) object field of SRH in SRv6 message carries HMAC TLV	SRv6 data message tampering
Load leveling[2,11]	Anycast-SID will balance the traffic from a single node to multiple nodes	Single point failure
Fault detect[2,11]	Local trigger (such as BFD(Bidirectional Forwarding Detection)), remote intra-domain trigger (IGP flooding), remote cross-domain trigger (updated by BGP-LS), end-to-end SR Policy survivability detection, explicit candidate path verification and dynamic candidate path recalculation	–
Failure recovery[2,11]	TI-LFA (Topology-Independent Loop-free Alternate) node protection	–
Service hiding[2,11]	Use the "mpls ip-ttl-propagation disable" command to hide the multi-hop MPLS network as a single-hop network, thus invalidating the traceroute command	Traditional topology detection, inter-domain topology detection
	By binding the SR Policy of the specified domain to BSID, users outside the domain cannot obtain the topology within the domain based on the candidate path information	

Table 1. SR native security mechanism.

Security mechanism	Examples
Identification inspection	StackPi algorithm for judging the security of forwarding path based on check stack identification[12]; SNAPP algorithm for verification by adding message integrity verification code (MIC) at sender and intermediate node[13]
Node verification	The ICING mechanism checks the received data packets by deploying authentication servers in each node of the network, but it brings high transmission overhead[14]; OSP algorithm grants a certificate between the source and the router, and the intermediate node verifies the data packet according to the certificate, which improves the inspection efficiency but increases the management overhead[15]. RPKI uses digital signature and certificate to authenticate routing source, which can effectively prevent route hijacking[16]; due to the limited deployment of RPKI infrastructure, Tomas and others put forward DISCO, which is based on distributed trust architecture to authenticate routing[17]
Trusted hardware	TrueNet mechanism deploys TCB(Trusted Computing Base) in each node of the network, and determines malicious links through multi-node security information negotiation[18]
Centralization of control	SDN architecture is usually adopted, such as VeriDP algorithm, which verifies whether the data is transmitted normally through control plane policy, thus improving the accuracy of network behavior detection[19]. DFL mechanism collects the verification information of nodes in the transmission path in a centralized way, but it is difficult to avoid a single point of failure[20]
Collaborative filtering	RISP uses RPKI to protect the inter-domain communication of source address, and completes traffic filtering through the cooperation of server, alliance center and AS border router[21]
New technology	Using blockchain to build a distributed trust framework can be used for inter-domain routing protocol to realize IP address prefix authentication[22]

Table 2. Main routing security mechanisms.

	SR native security solution	Main routing security scheme	ZbSR security solution
Means	Source routing, trust domain, packet authentication, load balancing, fault detection, fault recovery, service hiding	Identity verification, node verification, trusted hardware, centralized control, collaborative filtering, new technologies	Introduce security component based on ZTA concept
Advantages	Helps to improve security autonomously without additional security mechanisms	Provide adaptive security solutions for a variety of specified network scenarios	Design for segmented routing; Provide comprehensive protection; It can be used in new zero-trust application scenarios
Disadvantages	The source routing feature of the segmented route has security vulnerabilities, which makes it difficult to face some new zero-trust security threat scenarios	Features such as source route and segment label of a segmented routing network are not combined. Lack of comprehensive means of protection	The existing SDP architecture needs to be improved for segmented routing. You can control only the terminal devices that access the domain, but cannot directly control the intra-zone routing devices

Table 3. Comparison of 3 types of security solutions.

network element and PKI/CA, which is divided into 8 categories, as shown in Table 4. It can be seen that these problems can be attributed to the unreliable data exchange and network audit function of the SR-BE/TE data plane and the lack of relevant security mechanisms.

Coupling foundation of SR and ZTA basic function model. Figure 1a,b are the basic functional models of SR and ZTA, respectively, in which components with similar functions are identified with the same color. As shown by the red line and blue line in Fig. 1a, there are two generation modes of Segment list in SR basic function model: self-generation of network element and issuance of the controller; SDN controller, as its control

Security issue		Specific description	Major threats
Untrusted network element	Eavesdropping and replay[23,24]	After the tag of source node and data packet is obtained by eavesdropping, the explicit path of downstream traffic of eavesdropping point will be obtained directly, and replay attack can be realized by replaying normal identity messages to access the network	Confidentiality and Authenticability
	Message Forgery[25,26]	SR is usually only verified at domain boundary devices. By means of routing protocol flooding mechanism, forging control plane protocol messages, or modifying packet headers, it can change critical flow paths or occupy specific link bandwidth, create routing loops, drop traffic, intentionally report errors and other consequences, and destroy link load balance or block network communication	Integrity
	Denial of service attack[2]	According to the SR protocol, when the "segment left" field is non-zero, the router in the domain needs to send ICMP messages to the source address of the data packet. Attackers can use this to force SR nodes to generate and send a large number of ICMP messages, thus realizing DoS/DDoS attacks	Usability
	Identity Deception[27]	Because all the nodes in the SR trust domain are under the unified control, it is usually impossible to implement identity deception in the domain, but the nodes outside the domain may access the SR network as nodes in the domain	Confidentiality and controllability
	Intra-domain detection based on back door of device[27]	By detecting and using the back door of network device, the tag information and data packet payload generated by control protocols such as OSPF for SR are obtained by grabbing packets or tampering with forwarding table entries, and the MPLS/IPv6 tag stack information in them is analyzed to obtain the node tags, link tags and topological relations of downstream device	Controllability and confidentiality
	In-domain detection based on social engineering attack[27]	Log in to the device in SR domain without credit by means of social engineering such as cheating passwords, obtain the label and topology information stored by the device by means of show command, CLI (Command-Line Interface), SNMP (Simple Network Management Protocol) and NETCONF (Network Configuration Protocol) with the help of device maintenance and management tools, and use the device as a sniffing springboard and OAM functions such as MPLS tracert and MPLS ping of SR-MPLS network. Construct attack messages with different label stacks and specific TTL to detect network topology, nodes and links hop by hop. In the network which multi-source manufacturers' devices using different control standards coexist, springboard detection is easier to succeed	Controllability and confidentiality
	Failure of intra-domain node[28]	Due to the lack of stable label release mechanism, modifying SRGB (SR Global Block) of SR router, assigning used labels to it or configuring MPLS label range will lead to service interruption, and the device needs to be restarted	Usability
PKI/CA failure	Failure of infrastructure[29]	Attackers captured network security infrastructure such as PKI/CA (Public Key Infrastructure/Certificate Authority) through APT attacks, which led to the failure of traffic encryption in SR domain	Confidentiality and Authenticability

Table 4. SR security issues for zero-trust network environment.

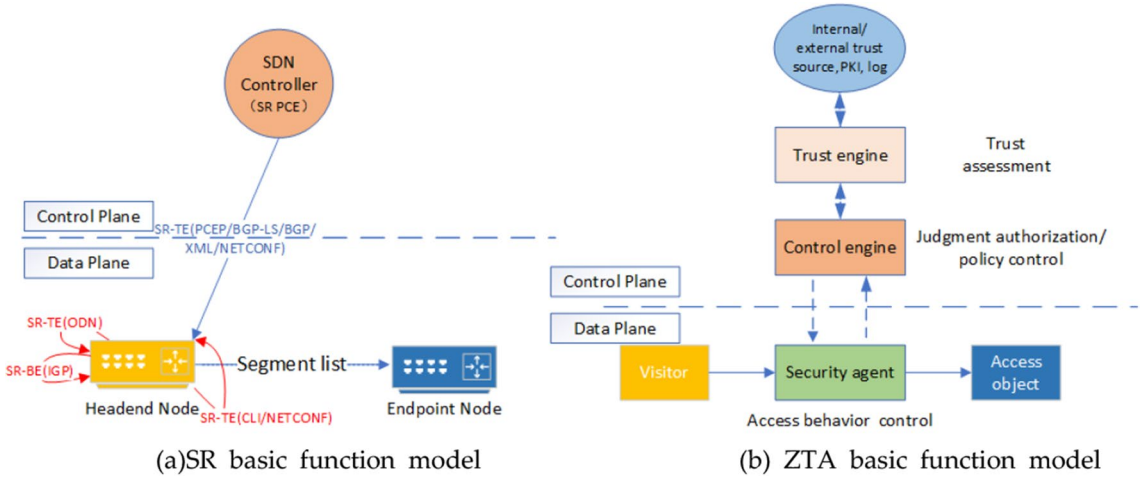

Figure 1. SR basic function model and ZTA basic function model.

engine, lacks the trust engine for managing trust and internal and external information sources, PKI, logs and other components for storing identity in ZTA model of Fig. 1b, which leads to its unreliable data exchange and network audit functions. Therefore, the two functional models have a certain coupling foundation, and the ZTA trust engine can be integrated into the SDN controller.

The previous work. In the early stage, we mainly have a related research result on SR data plane security model, and the difference between it and this work is shown in Table 5.

	SbSR (SDP based SR)[30]	ZbSR (ZTA based SR)
Problem oriented	The terminal device of the SR network data plane	The switching device of the SR network data plane
Modeling	The migration model of the mature SDP model is carried out	Based on the concept of ZTA, a new ZTA model is designed by adding security components and reassembling the original functional components
Assessment	Port scanning; Traffic monitoring; DoS attack; Topology detection based on label detection; Routing loop attack based on directional label; Performance overhead	Control plane message tampering; Data plane loop attack; Identity deception; Back door utilization; DOS attack; Performance overhead

Table 5. Comparison between previous work and this work.

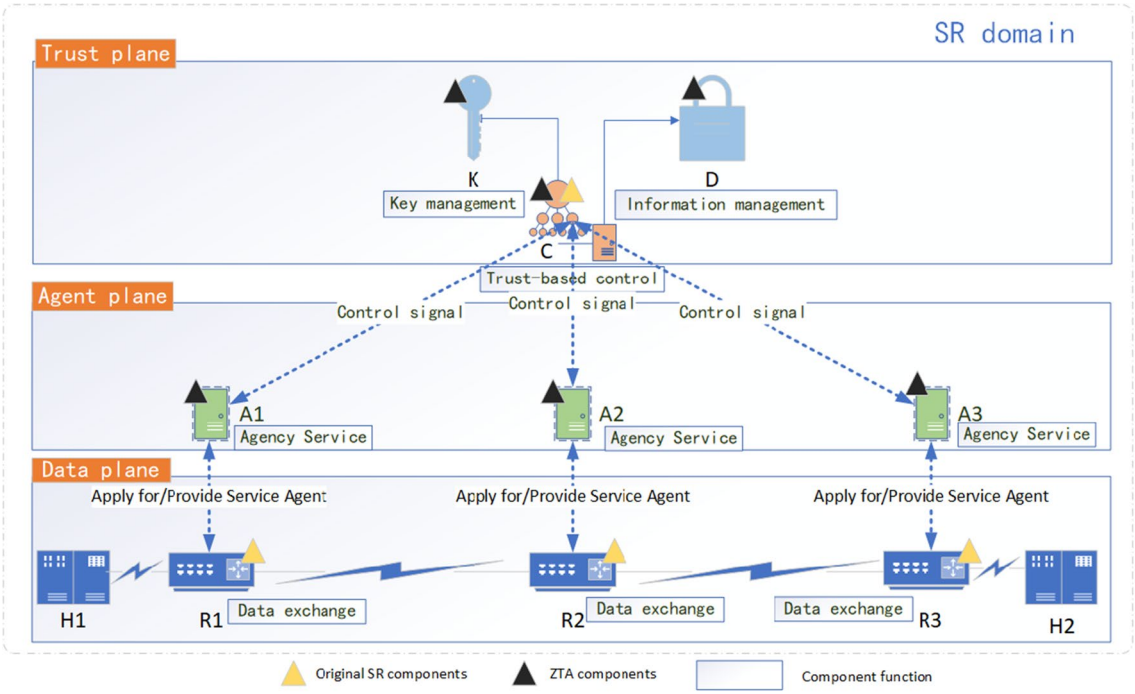

Figure 2. ZbSR security model.

SR-BE/TE security model (ZbSR) based on ZTA

Based on the above analysis, the ZbSR security model proposed in this paper is mainly composed of trust plane, agent plane, and data plane, as shown in Fig. 2, in which the trust plane is composed of the controller (C), key center (K) and information base (D), which is connected to the data plane through agent plane, and whitelist access control is established between planes, which is responsible for centralized control, authentication and trust calculation of data plane devices, in which the controller is based on the expansion of the original SDN controller of SR architecture. The agent plane consists of agent (A) connected in series to each SR data plane device, responsible for providing security agency services such as encryption, auditing, and reporting. The data plane is composed of switching devices such as SR router (R) and terminal device such as host (H), which is responsible for generating data and transmitting traffic.

The ZSR model is modeled with symbols and definitions in Table 1.

ZbSR model component function and modeling. The function of the ZbSR model component is that the controller calls the key center (for managing keys) and the information base (for managing and storing identity information) to control the subordinate plane based on trust, and the agent provides security agency services for the data plane.

The controller (C). The controller consists of the original SDN controller and the trust engine expansion module. It is responsible for realizing access control, path delivery, and other functions based on authentication and trust calculation.

(1) Segment list issuing

In SR-TE, a path is issued for the data plane by generating a Segment list, the Segment list *SL* is shown in formula (1).

$$SL = \{SID_1, SID_2, SID_3, \ldots, SID_i, \ldots, SID_j, \ldots, SID_n\} \quad (1)$$

(2) Authentication and authorization

Referring to the security design based on identity control access[31,32] in Software-Defined Perimeter (SDP)[33], the trust engine of the controller performs identity authentication and trust evaluation on the devices in the domain, and then implements the minimum authorization[34], and then the authentication and authorization results are handed over to the SDN controller, which issues control signaling. The access subject and object 5-tuple authorization information Credit is modeled as shown in formula (2), where *Smac* and *Dmac* represent the MAC addresses of the access device and the visited device respectively, *SID* and *pSID* represent the Prefix Node SID assigned by the access device and the visited device respectively, and *P* represents the access protocol.

$$Credit \stackrel{def}{=} \{\text{Smac} + Dmac + SID + P + pSID\} \quad (2)$$

According to the ZTA concept, the authorization mode can be divided into centralized authorization and separate authorization. Centralized authorization means that after authentication and trust evaluation are carried out on the network-connected devices, the list of accessible devices and protocols is granted in a centralized way in the form of an authorization list. As shown in formula (3), the authorization list contains 6 types of information, among which, D^i, $Cert^i$, t_{Certi}, PK_{Di}, P and $K_D(i)$ respectively represent the accessible device i, the access certificate of device I, the lease period of the access certificate of device i, the public key of device i, the access protocol and Separate authorization means that device A needs to verify authorization every time it accesses device B through the new protocol. At this time, the authorization information is shown by formula (5), including the access certificate of device B, the lease period of the access certificate of device B, the public key of device B, the access protocol, and the traffic encryption key. Compared with centralized authorization, separate authorization not only achieves fine-grained control but also brings more overhead. Therefore, this paper sets two authorization modes that can be switched as needed.

$$List_A = \left\{D, Cert_A, t_{Cert_A}, PK, P, K_D^A\right\} \quad (3)$$

$$\begin{cases} D = \{D^1, D^2, D^3, \ldots, D^n\} \\ Cert_A = \left\{Cert_A^1, Cert_A^2, Cert_A^3, \ldots, Cert_A^n\right\} \\ t_{Cert_A} = \left\{t_{Cert_A^1}, t_{Cert_A^2}, t_{Cert_A^3}, \ldots, t_{Cert_A^n}\right\} \\ PK = \left\{PK_{D^1}, PK_{D^2}, PK_{D^3}, \ldots, PK_{D^n}\right\} \\ P = \left\{P_{D^1}, P_{D^2}, P_{D^3}, \ldots, P_{D^n}\right\} \\ K_D^A = \left\{K_D^A(1), K_D^A(2), K_D^A(3), \ldots, K_D^A(n)\right\} \end{cases} \quad (4)$$

$$List_A(B) = \left\{Cert_A^B, t_{Cert_A^B}, PK_B, P_B, K_D^A(B)\right\} \quad (5)$$

(3) Rules issuing

Before the SR source node starts streaming according to the Segment list, the controller issues security rules for preventing path tampering to the agents of each node in the list, detailed in Section "Network audit security algorithm based on solid authentication".

(4) Device control

The controller centrally controls all devices in the domain, centrally configures their Prefix-SID to prevent the attackers from tampering, and timely removes the failed devices from the list of available devices and recycles their SIDs; storing the suspected malicious device behavior found in the detection into the information base, disabling its access credentials and reporting to the network administrator when the negative feedback accumulation causes its trust to be lower than the threshold; provide the central working clock for each component of the system and provide a unified time reference.

(5) Keys scheduling

Through the agent plane of the controller, the key center is called to centrally distribute the traffic encryption key and other keys to the protocol peers that have been authorized successfully.

Key center. The key center is used to centrally control the keys in the domain and prevent the potential safety hazard of key decentralized configuration[35]. It adopts the popular "symmetric password-asymmetric password" mixed encryption mechanism[36], in which the fast symmetric password is used for traffic encryption/decryption, and the slow asymmetric password is used for key exchange and signature verification; because ZTA doesn't trust public PKI/CA, the key center is used as the private CA in the domain to issue digital certificates to the terminal devices in the domain[37]. The managed keys include traffic encryption key K_D, key-encryption key *KeK*, its own public and private keys K_{pub} and K_{pri}, and the public and private keys R_{pub} and R_{pri} of each terminal device. All

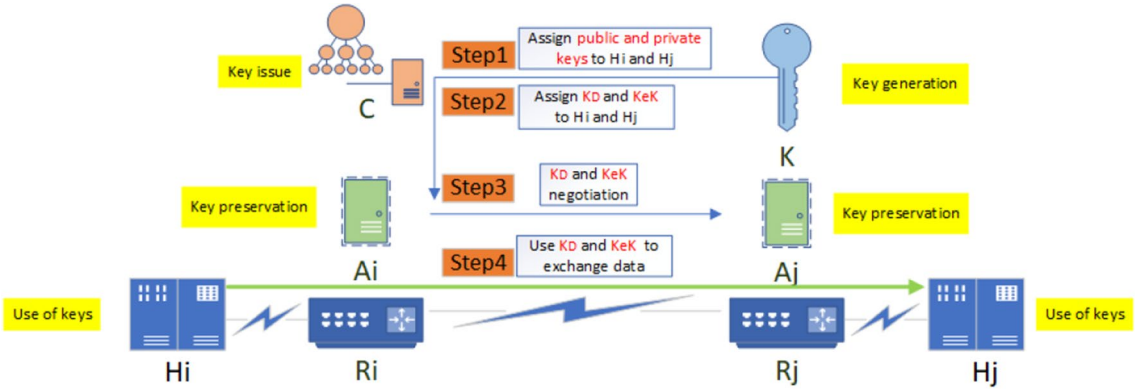

Figure 3. ZbSR key usage process.

keys are replaced regularly to prevent abuse. To simplify the configuration, duplicate keys can be configured for the nodes in an SR Anycast group. The use of all kinds of keys can be divided into 4 steps, as shown in Fig. 3.

Step1 K preallocates the public and private keys for all terminal devices, sends them through C, deposits them in A, and replaces them regularly;
Step2 K allocates K_D and KeK as needed for data exchange between H_i and H_j, which is issued by C, stored in A, and replaced regularly;
Step3 A_i and A_j use the public and private keys of H_i and H_j to negotiate K_D and KeK;
Step4 H_i and H_j exchange data with K_D and KeK.

Information base. The information base is used to store and manage device authentication information and protocol authorization information. The device authentication information is related to authentication information of the device itself, such as username/password, SID, router-mac, etc.,[38] which is determined by the formula (6), and the protocol authorization information is related to the protocol authorization. Such as *Whitelist* of connection, routing protocol type P_R, link Adjacency Adj_{ij}, port number *Interface*, peer IP address IP_p, etc., are determined by the formula (7), in which Adj_{ij} is determined by the adjacency matrix of link, as shown in formula (8), which describes the adjacency of device, with 1 indicating adjacency and 0 indicating non-adjacency; the Whitelist of connections is determined by the formula (9), which specifies all permitted connections in the domain, and the information in the information base is dynamically updated with the change of network devices.

$$I_a(i) \stackrel{def}{=} \{Uname + Upass + SID + Rmac\} \tag{6}$$

$$P_a(i,j) \stackrel{def}{=} \left\{Interface_i + Adj(i,j) + Whitelist + P_R + IP_j\right\} \tag{7}$$

$$Adjacency = \begin{pmatrix} Adj_{11} & \cdots & Adj_{1m} \\ \vdots & Adj_{ij} & \vdots \\ Adj_{m1} & \cdots & Adj_{mm} \end{pmatrix}, Adj_{ij} = (1,0) \tag{8}$$

$$Whitelist = \{(SID_i, SID_j), \ldots, (SID_i, A_k), \ldots, (A_k, C), \ldots\} \tag{9}$$

Agent. The agent is used to provide security agency service for data plane devices, and it is directly connected with each SR switching device[39]. There is no direct connection channel between agents, to prevent malicious nodes from bypassing trust plane supervision and direct communication. The agent mainly has 4 functions: key management, path report, log record, and behavior audit. Key management means that the agent provides key negotiation agent services for data plane devices; path report implies that after the SR head node generates the flow path, it needs to report the path to the controller through the agent for decision-making; logging refers to recording the behavior log of SR switching device to trace the malicious behavior; behavior audit refers to auditing the behavior of data plane devices together with the controller according to the network audit security algorithm in Section "Network audit security algorithm based on solid authentication".

Data exchange and network audit security algorithm of ZbSR model. To ensure the integrity, confidentiality, and availability of data in the SR domain, the ZSR model introduces five security mechanisms: packet authentication, data encryption, check and filtering, security audit, and trust renewal.

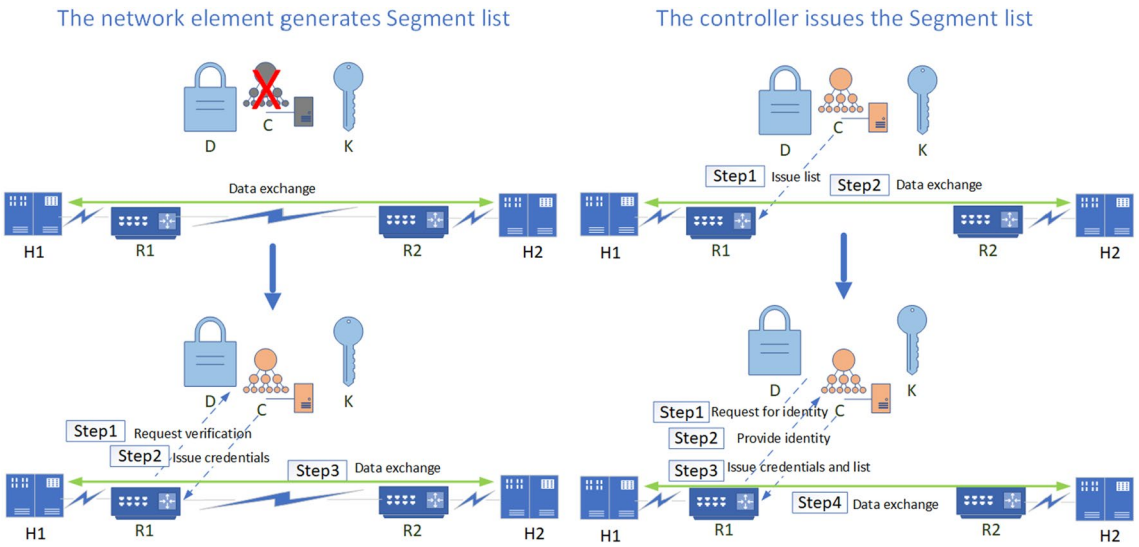

Figure 4. Simplified ZbSR access process.

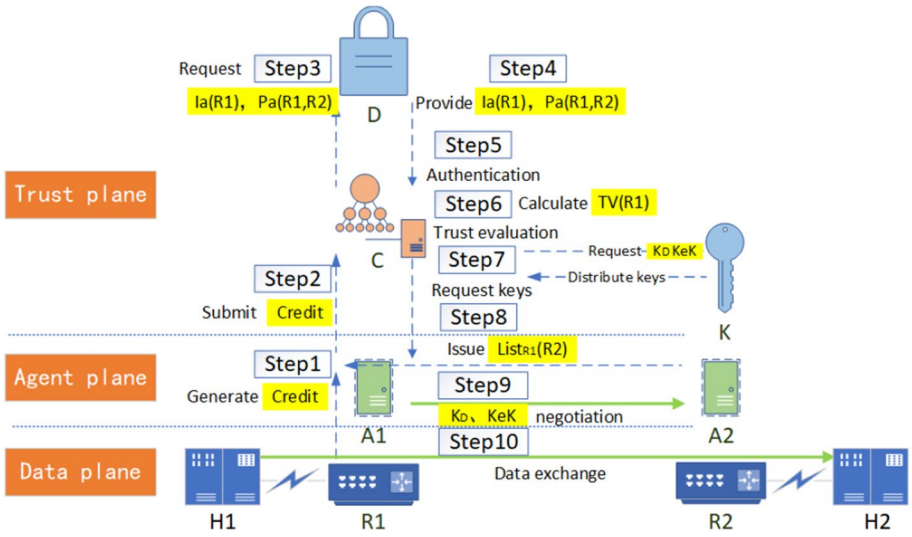

Figure 5. Data exchange process in the mode for the network element to generate Segment list.

Data exchange security algorithm based on trust evaluation. To realize reliable east–west data exchange between switching devices, firstly, based on ZTA's security design of "first authentication, then connection", a UDP-based SPA (Single Packet Authorization) method is adopted to initiate pre-authentication to the trust plane, and the trust plane carries out the north–south security authentication based on identity authentication, trust evaluation, and key negotiation. Secondly, the terminal device realizes the encrypted traffic exchange by encrypting traffic with a key. Taking the separate authorization mode as an example, the simplified process of terminal H1 accessing H2 is shown in Fig. 4, implemented in two modes: Segment list generation by the network element and Segment list distribution by the controller.

(1) Mode for the network element to generate Segment list

In this case, the data exchange process is shown in Fig. 5, and the pseudo-code of the process is shown in Algorithm 1.

Algorithm1 data exchange security algorithm in the mode for the network element to generate Segment list

Input: MAC address *Smac* and Prefix Node SID *SID* of routing device R1,
MAC address *Dmac* and Prefix Node SID *pSID* of routing device R2,
Segment list *segment* generated by device R1, residual transmission flow from R1 to R2, trust threshold *Th*,
public key MP_{pub} and private key MP_{pri} of device H_1, public key SP_{pub} and private key SP_{pri} of device H2

Output: the result of data exchange between terminal devices H_1 and H_2 (1: success; 0: fail)

1. Deploy IACL filtering policy for R1, and filter the access of out-of-domain nodes to the intra-domain segments according to source IP, destination IP, SRH, etc. //Prevent service theft.
2. Headend node R1 automatically generated *segment* and prepare to send data
3. A1 and A2 collect *Smac, Dmac, SID, pSID* of R1 and R2 respectively, and A1 combines above information with P to generate *Credit*
4. A1 submits *Credit* as SPA package load to C //A1 initiates single verification package to C
5. C applies for $I_a(R1), P_a(R1, R2)$ from D
6. D provides $I_a(R1), P_a(R1, R2)$ to C
7. C ***call*** $subalgorithm_{IA}$ according to Credit, $I_a(R1)$、$P_a(R1, R2)$ // Call for authentication sub-algorithm
8. ***if*** $subalgorithm_{IA}(R1, R2)=1$
9. ***then*** C ***call*** $subalgorithm_{TE}$ //Call for trust evaluation sub-algorithm
10. ***if*** $subalgorithm_{TE}(R1) \geq Th$
11. ***then*** C applies to K to allocate K_D and *KeK*
12. K allocates K_D and *KeK* for C
13. Establish a bidirectional encrypted connection between C and A1 //mTLS[5] can be adopted
14. C sends $List_{R1}(R2)$ to A1
15. A1 and A2 represent H_1 and H_2 respectively, and call $subalgorithm_{KN}$ // Call for key negotiation sub-algorithm
16. **if** $subalgorithm_{KN}(MP_{pub}, MP_{pri}, SP_{pub}, SP_{pri}, K_D, KeK)$ =1
17. ***while*** *flow*>0 ***do***
18. H_1 and H_2 use K_D and *KeK* to transport encrypted traffic
19. *flow=flow-*
20. ***end while***
21. ***output*** 0
22. ***end if***
23. ***output*** 1
24. ***else return*** 0
25. ***end if***
26. ***else return*** 0
27. ***end if***

(2) Mode for the controller to issue Segment list

In this case, the data exchange process is shown in Fig. 6, and the pseudo-code of the process is shown in Algorithm 2.

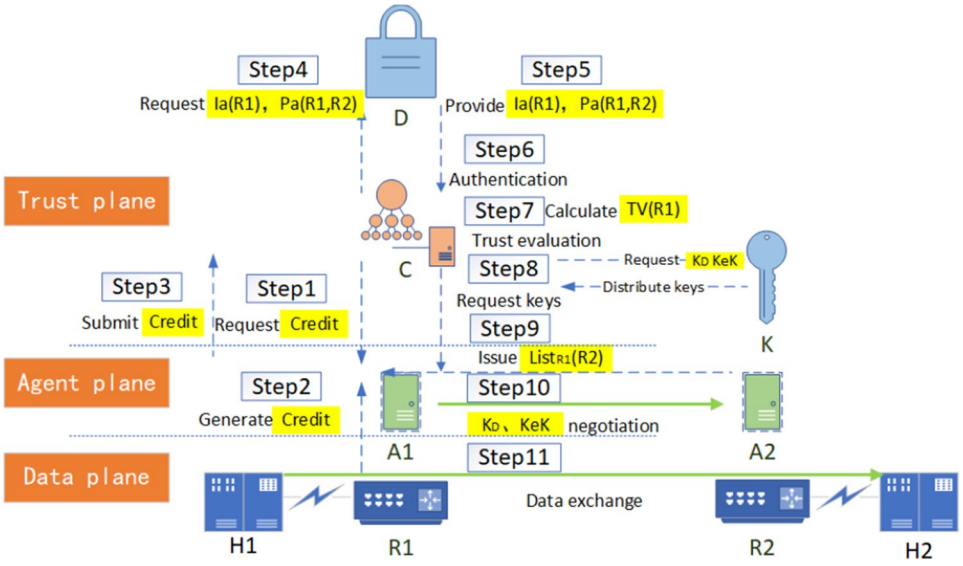

Figure 6. Data exchange process in the mode for the controller to issue Segment list.

Algorithm2 data exchange security algorithm in the mode for the controller to issue Segment list

Input: MAC address *Smac* and Prefix Node SID *SID* of routing device R1,

MAC address *Dmac* and Prefix Node SID *pSID* of routing device R2,

Segment list *segment* generated by controller, residual transmission flow from R1 to R2, trust threshold *Th*,

public key MP_{pub} and private key MP_{pri} of device H_1, public key SP_{pub} and private key SP_{pri} of device H2

Output: the result of data exchange between terminal devices H_1 and H_2 (1: success; 0: fail)

1. C generated *segment* and prepare to send it to the headend node, then C ask the headend node R1 and the endpoint node R2 to upload their *Credit* //If there are intermediate nodes between R1 and R2, it is necessary to upload their macs and SIDs.
2. A1 and A2 collect the *Smac, Dmac, SID, pSID* of R1 and R2 respectively, and , and A1 combines above information with P to generate *Credit*
3. A1 submits *Credit* as SPA package load to C //The following is the same as algorithm 1.
4. …***call*** … $subalgorithm_{IA}$ **…**
5. …***call*** … $subalgorithm_{TE}$ **…**
6. …***call*** **…** $subalgorithm_{KN}$ **…**

The authentication sub-algorithm, trust evaluation sub-algorithm, and key negotiation sub-algorithm called by algorithms 1 and 2 are shown as $subalgorithm_{IA}$, $subalgorithm_{TE}$, and $subalgorithm_{KN}$. In $subalgorithm_{TE}$, trust renewal can be implemented for temporary trust granted based on security metrics, but this scheme has not been implemented in this paper due to limited research energy.

subalgorithm$_{IA}$ authentication sub-algorithm

Input: *Credit* of devices Ri and Rj,
device authentication information $I_a(Ri)$, protocol authentication information $P_a(i,j)$,
number of devices to be authenticated l

Output: device identity authentication result (1: true; 0: false)

1. ***while*** l>0 ***do***
2. C extracts macs from the *Credit* of Ri and Rj respectively, and records them as $Mac1_{Ri}$, $Mac1_{Rj}$
3. C extract SIDs and macs of Ri and Rj from$I_a(Ri)$, and record them as SID_i, SID_j, $Mac2_{Ri}$, $Mac2_{Rj}$
4. C extract Whitelist from $P_a(i,j)$
5. ***if*** $Mac1_{Ri} = Mac2_{Ri} \vee Mac1_{Rj} = Mac2_{Rj}$ // Compare the identity of devices Ri and Rj
6. ***if*** $(SID_i, SID_j) \in Whitelist$ // Check whether the connection relationship between Ri and Rj belongs to the *whitelist*
7. ***then return*** 1
8. ***else return*** 0
9. ***end if***
10. ***else return*** 0
11. ***end if***
12. l=l-1
13. ***end while***

subalgorithmTE trust evaluation sub-algorithm

Input: device r, device set R^m, number of devices in device set m

Output: device trust value $TV(r)$

1. Initialize $TV(r)=0$, $TV(r)\in[-1,1]$ // $TV(r)=1$ means trust device r completely, while $TV(r)=-1$ means that don't trust device r completely
2. Set the evaluation set to $R_a=R-r$, the trust value of C to the evaluation node R_a^i is T_{C,R_a^i} , the threshold value of T_{C,R_a^i} is $Th(T_{C,R_a^i})$, which meeting $-1\le T_{C,R_a^i}\le Th(T_{C,R_a^i})\le 1$, $T_{C,R_a^i}=1$ //The controller does not grant subjective trust to the switching device, but determines it according to the recommendation of other nodes; set the controller's initial trust value to all recommenders as 1.
3. ***while*** $0<n\le \mathrm{m}-1$ ***do*** //filter evaluation set
4. For the evaluation node R_a^i , calculate the probability that the monitored device r faithfully forwards packets and record it as $F_{R_a^i,r}$ //Take $F_{R_a^i,r}$ as the recommended trust value of R_a^i to r[40]
5. Calculate the current trust value of device r: $TV_{tem}=\sum_{i=1}^{n}(\frac{T_{C,R_a^i}}{\sum_{i=1}^{n}T_{C,R_a^i}}\times F_{R_a^i,r})$ //Get the instantaneous trust value of device r based on the current and previous recommendation values of all recommenders by weighted average
6. Calculate the difference $dv_{C,R_a^i}=|TV_{tem}-F_{R_a^i,l}|$ between the recommended trust value $F_{R_a^i,r}$ from R_a^i and current instantaneous trust value of r//This result can reflect the deviation degree between the given recommended opinion and the overall opinion, which is used as the basis for feedback on the trust degree of the recommender R_a^i based on his recommendation
7. Set the threshold of the difference dv_{ir} to $Th(dv_{ir})$, and the reward and punishment factors are rd and p respectively, which meet the requirements $p>rd$
8. Calculate the trust value of C to the evaluation node $R_a^i: T_{C,R_a^i}=\begin{cases} T_{C,R_a^i}+rd, dv_{C,R_a^i}<Th(dv_{ir}) \\ T_{C,R_a^i}-p, dv_{C,R_a^i}\ge Th(dv_{ir}) \end{cases}$ //evaluate the trust of the evaluation node based on the recommendation feedback
9. ***if*** $T_{C,R_a^i}<Th(T_{C,R_a^i})$
10. ***then*** remove R_a^i form R_a *// When the recommender's trust level falls below the threshold, it will be removed out of the evaluation set*
11. ***end if***
12. ***end while***
13. Set the number of interactions between R_a^i and r to $n_{R_a^i,r}$
14. $TV(r)=\sum_{l=1}^{card(R_a)}(\frac{CR_{ir_l}\times n_{R_a^i,r}}{\sum_{l=1}^{n}CR_{ir_l}\times n_{R_a^i,r}}\times F_{r_l,j})\times e^{-1/card(R_a)}$ //Trust value of device r is the weighted average of the recommended values from the evaluation set, and the evaluation set size $card(R_a)$ and the number of interactions $n_{R_a^i,r}$ between the evaluation set elements and device r are used for optimization
15. ***output*** $TV(r)$

subalgorithmKN key agreement sub-algorithm

Input: public key MP_{pub} and private key MP_{pri} of H_i^k connected with device R_i, public key SP_{pub} and private key SP_{pri} of H_j^l connected with device R_j, data encryption key K_D, key encryption key KeK

Output: the result of key negotiation between H_i^k and H_j^l (1: completed; 0: failed)

1. A_i uses KeK to encrypt K_D: $C_{K_D} = E_{KeK}(K_D)$
2. A_i uses SP_{pub} to encrypt KeK : $C_{KeK} = E_{SP_{pub}}^{H_j^l}(KeK)$
3. A_i uses MP_{pri} to sign K_D : $D_{K_D} = D_{SP_{pri}}^{H_i^k}(K_D)$[41]
4. A_i binds the above results to triple message *CCK (i, j)*, so that $CCK(i,j) \overset{def}{=} \{C_{K_D}, C_{KeK}, D_{K_D}\}$
5. A_i sends $CCK(H_i^k, H_j^l)$ to A_j via C
6. A_j receives $CCK(H_i^k, H_j^l)$
7. A_j uses SP_{pri} to decrypt C_{KeK} : $KeK = D_{SP_{pri}}^{H_j^l}(C_{KeK})$
8. A_j uses KeK to decrypt C_{K_D} : $K_D = D_{KeK}(C_{K_D})$
9. A_j uses MP_{pub} to authenticate signature D_{K_D} : $K_D' = E_{SP_{pub}}^{H_i^k}(D_{K_D})$
10. ***if*** $K_D = K_D'$
11. ***then*** H_i^k、H_j^l save K_D、KeK
12. ***else*** alerts the operator, ***return*** 0
13. ***return*** 1
14. ***end if***

Network audit security algorithm based on solid authentication. Due to the lack of audit mechanism for threat representation in SR network, a network security audit algorithm is proposed based on ZTA's strategy of solid verification of all behaviors in the domain. The pseudo-code of related process is shown in algorithm 3. The audit content includes the following 6 aspects.

(1) Field audit: audit whether the TTL value of the packet header is legal and whether the outbound traffic of the domain egress router has removed the SRH.
(2) Behavior audit: audit whether the rate of ICMP information generation reaches the threshold for enabling the ICMPv6 rate-limiting mechanism and whether the traffic which cannot find next-hop to be malicious.
(3) Loop audit: if the label stack only uses Prefix-SID, then directly determine whether there is a loop according to the following subalgorithmLP; if the label stack contains Adjacency-SID, restore the network topology according to the label stack, and then determine the loop according to subalgorithmLP.
(4) Label audit: audit the validity of SRGB labels, SRLB labels of specific border routers, and other external labels.
(5) Path audit: audit whether the flow path has been tampered with by malicious intermediate nodes. As shown in Fig. 7, the controller issues a segment list {16,007} to node 3, and according to the list, issues security rules to all intermediate nodes (node 5 at this time) along the path: the top label of the received packet from the interface from node 3 to node 5 should be 16,007; otherwise, it is discarded.
(6) SID audit: audit whether the SID of a flow path node has been tampered with by malicious intra-domain nodes; it can be divided into two steps. As shown in Fig. 8, the controller centrally configures the Prefix-SID, router-id of each device node, imports them into the information database in advance, and synchronizes them to each device node through LSA notifications. Each device node caches its own and other node SIDs to Label Manager (LM); in the first step, when each device node receives a new LSA notification, it will be audited and compared with the SID cached by the LM. It will be considered valid and received only if the matching is successful. If the matching fails, it will report an exception to the controller, then the controller determines whether there is an attack; the second step is to refer to LM and FIB (Forwarding Information dataBase) to audit whether the SID has been tampered with during streaming. If an unrecognized SID is found in the LM, it will be further matched in the information database. If the matching fails, it will be reported to the operator.

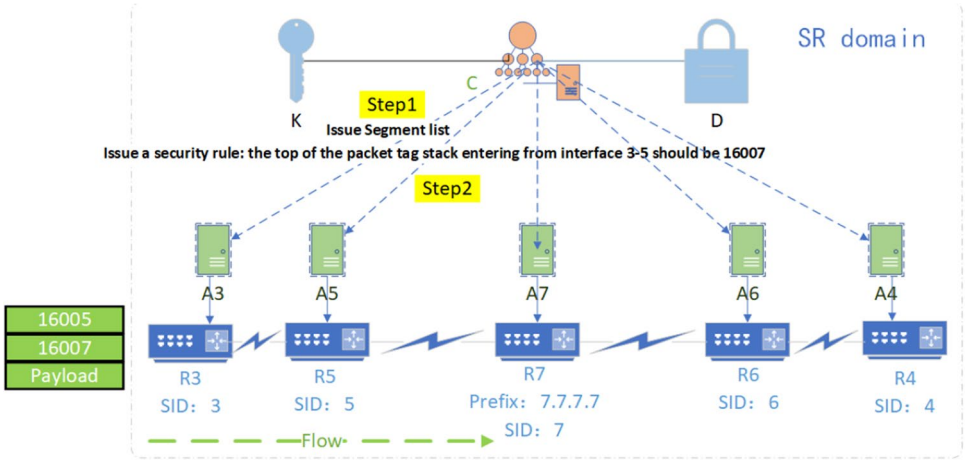

Figure 7. Schematic diagram of SR rule audit.

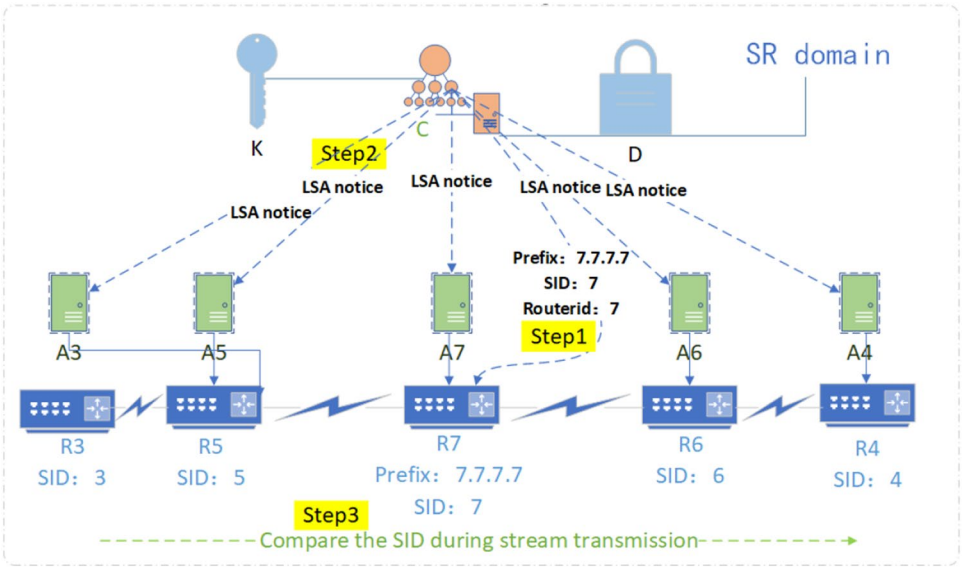

Figure 8. Schematic diagram of SR SID audit.

Algorithm3 Network audit security algorithm based on strong authentication

Input: Topology $G = (V, L)$, number of nodes n, number of flows k,
Segment list $SL = \{SID_1, SID_2, SID_3, .., SID_i, \ldots, SID_j, \ldots, SID_n\}$

Output: Safety audit results (1: passed; 0: failed)

Initialize variables j, t, v = 0//respectively represent the current ICMP message generation amount, time and ICMP message generation rate
Initialize T,V//Threshold of variables t, v
while $t \le T \vee v \le V$ ***do***// Within the requirements of streaming period and ICMP information rate range
  j increments with each ICMP message generated
  $v = \frac{j}{t}$
  for $I = 1$ ***to*** M // Traverse each agent
    Initialize set $P, Q = \{\emptyset\}$//P，Q are used to store all the message headers and label stacks of the flows corresponding to each agent respectively
    for $i = 1$ *to* K_i // For each data flow of the agent
      Extract all headers in the data stream to P
      if TTL field of headers are legal
      Extract all tag stacks in the data stream to Q.
      call $subalgorithm_{LP}$ // Call the loop audit sub-algorithm
      if $subalgorithm_{LP}(SL) = 1$
        if the order of the node labels in the label stack corresponds to the actual topology //Tag audit
          if all intermediate nodes of flow comply with security rules and SIDs have not been tampered with //Path audit and SID audit
            then start streaming
              if the flow which reaches the node can find the corresponding tag // Behavior audit
                Stream continues
              else return 0 // Behavior audit failed
              end if
              return 1
            else return 0 //Path audit and SID audit failed
            end if
          else return 0 //Tag audit failed
          end if
        else return 0 // Loop audit failed
        end if
      else return 0 // Field audit failed
      end if
    end for
  end for // Field audit completed
end while
Discard packet// The packet timed out or ICMP rate exceeded the limit
return 0

The subalgorithmLP called in algorithm 3 is as follows.

Symbol	Definition
M	Number of agents
N	Number of device nodes
N_t	Number of nodes in communication
N_f	Flow quantity
N_f^i	Number of streams that agent i flows through
k1	Flow speed
k2	Constant
n_i	Number of message segments in each stream
μ	Key update period

Table 6. Definitions of symbols in the ZbSR model.

subalgorithm$_{LP}$ loop audit sub-algorithm

Input: Segment list $SL = \{SID_1, SID_2, SID_3, ..., SID_i, ..., SID_j, ..., SID_n\}$

Output: Loop detection result (1: none; 0: exist)

1. Set the number of intermediate nodes between SID_i and SID_j to N_{ij}, and the set of intermediate nodes to $F^{N_{ij}}$

2. ***for*** $i = 1; i \le n; i++$

3. ***for*** $j = i; j \le n; j++$

4. ***if*** $SID_i \notin \{SID1, SID2, ..., SID_{i-1}\} \vee \{SID1, SID2, ..., SID_{i-1}\} \notin F^{N_{ij}}$ //Determine whether there are duplicate tags in SL, and the intermediate nodes between any two nodes R_i and R_j must not contain the nodes before R_i in the Segment list

5. ***then output*** 1

6. ***else output*** 0

7. ***end if***

8. ***end for***

9. ***end for***

ZbSR model security overhead. According to different components, the security cost of the ZbSR model is divided into 6 parts, that is, controller cost, key center cost, information base cost, agent cost, encryption cost of terminal devices, and component synchronization cost. Because all kinds of security components run in parallel, in addition to the one-time hardware cost brought by the introduction of devices, the evaluation of system performance cost only needs to pay attention to the time delay item that has the most significant influence on streaming transmission. The related symbols and definitions are shown in Table 6.

The first is the controller overhead. If the performance allows, controller can be deployed single, and it can also be deployed multiple to realize load balancing and disaster recovery. The cost is related to the number of devices N it controls, and the cost is associated with the number of streams N_f when it issues paths and rules. When authenticating, the cost is related to N; when controlling the device, it is only performed when the device leaves the network, or malicious device is generated, and the occurrence probability is small and can be ignored; when scheduling the key, it is only issued to the nodes in communication after the key is updated, so the computational complexity of the controller overhead is $O(N + N_f)$. The second is the key center overhead. There is only one set in the SR domain, and the cost mainly comes from its regularly key updating, and its computational complexity is $O\left(N_t \times \frac{k2}{\mu}\right)$. The third is the information base overhead. In-domain devices cache commonly used verification information to the local agent, and the information base only needs to import information when the topology is established, verify information when new users access the network, and update the information when devices change, so the overhead is negligible compared with the controller. The fourth is the agent overhead. The agent is used in every streaming for key management, path reporting, and logging, which is related to N_f^i; in behavior audit, the time complexity of field audit is $\sum_1^{N_f}(k1 \times n_i)$: Behavior audit is triggered only when abnormal traffic occurs, and the overhead can be ignored. Other auditing functions are only related to the number of streams N_f^i, so the computational complexity of agent overhead is $O\left(\sum_1^{N_f}(k1 \times n_i) + \sum N_f^i\right)$, namely $O\left(\sum_1^{N_f}(k1 \times n_i) + N_f\right)$. The fifth is the encryption overhead of terminal devices. If hardware devices are used for encryption, the efficiency is high, so the time delay can be ignored. However, if software devices are used for

	Time overhead	Storage overhead	Hardware overhead
ZbSR	More time overhead is introduced for the authentication and encryption mechanisms are introduced at the same time	The storage overhead is focused on the newly added security components, so the original data plane devices have no new security overhead	The key center, agent and information base are newly introduced
ICING[14]	More than 10,000 cycles	23.3% more expensive than a standard IP router	93% more expensive than a standard IP router
OSP[15]	Less than 1000 cycles	Lower storage consumption compared to ICING[14]	No assessment
MFRA[42]	9% of the traditional OpenFlow solution	No assessment	No assessment

Table 7. Security overhead comparison of various security models.

	ZbSR	Baseline	MFRA
Physical machine/virtual machine operating system	Ubuntu18.04/Ubuntu16.04		
Physical machine CPU	Core i7-7700 3.6 GHz		
Physical/virtual machine memory	32 GB/2 GB		
Controller	OpenDaylight		
Southbound interface	PCEP/NETCONF		
Data plane	MPLS		
Additional security mechanism	Data exchange security algorithm, network audit security algorithm	None	Backup, multi-failure recovery and re-failure avoidance

Table 8. Test object configuration.

encryption, it will take more time. The sixth is the component synchronization overhead. Usually, there is only one controller deployed in the domain, and the information between agents does not have to be identical, so only the key negotiation and authentication need to be synchronized. Here, the Raft state synchronization technology is implemented according to the flow information between devices, and the overhead is low and can be ignored. It can be seen that the security overhead of the ZbSR model is concentrated in 3 parts: controller, agent, and terminal device encryption. The security cost comparison between this model and other similar routing security models is shown in Table 7. It can be seen that compared with other models, the ZbSR model brings more hardware cost and time cost due to the introduction of new security components and security mechanisms, but this is necessary, and the reasons have been explained in Table 3.

Simulation test and analysis

Simulation settings. OpenDaylight open-source controller is installed based on KVM virtual machine in EVE-NG 3.0.1-16 PRO, and its function is programmed to realize ZbSR controller. Dedicated Linux virtual machine is used as agent. Because it is challenging to build private CA, information base, and encryption hardware, and it is not the focus of research, this paper adopts a simplified design and uses virtual machines based on X.509 protocol and DES encryption software to simulate key center. Virtual machine simulation information base based on MySQL database. Due to the lack of mature and comparable SR security models, the ZbSR model is compared with the SR Baseline model, the MFRA model, the SDN cross bitmap algorithm model[43], and the DoS attack detection model based on C4.5[44], among which the SR Baseline model has been introduced in Section "Coupling foundation of SR and ZTA basic function model". In the MFRA model, the multi-fault quick recovery and avoidance mechanism based on SR pre-deployment link ring backup is mainly applied, and the configuration of test objects is shown in Table 8. There are 4 security tests and 1 overhead tests: control plane message tampering, data plane loop attack, identity deception, and DoS attack. The simulation network topology is shown in Fig. 9, in which the components of the SR Baseline model and MFRA model are shown by the red box in the figure, that is, they include the SR native network composed of 5 Cisco xrv9k routers and 1 OpenDaylight controller based on KVM; ZbSR model, based on KVM, additionally set an information base, an expansion controller and a key center, and each router is connected with another KVM virtual machine as an agent. The control plane components and data plane topology of the SDN cross bitmap algorithm model and the DoS attack detection model based on C4.5, are consistent with the SR Baseline model, except that the data plane uses SDN switches.

Safety performance test and analysis. Due to the lack of comprehensive SR security model facing multiple threats, 4 models are introduced here, namely SR Baseline model, MFRA model, SDN cross bitmap algorithm model and DoS attack detection model based on C4.5, which are respectively compared with ZbSR model proposed in this paper in different types of attack tests. The threat model is set as follows: the attacker will implement 4 kinds of attack based on different switching devices and terminals, which one is the message tampering, namely the attacker tamper with the control plane message of a switching equipment, through the routing protocol flooding mechanism or other ways. This attack will induce the original flow path changes, to test whether the ZbSR model, the SR Baseline model, and the SDN cross bitmap algorithm model can prevent this attack. The second is the routing-loop attack, that is, by pressing the specified MPLS label stack into the head node of the traffic, the loop attack packet is constructed, so that the traffic transmission path will generate

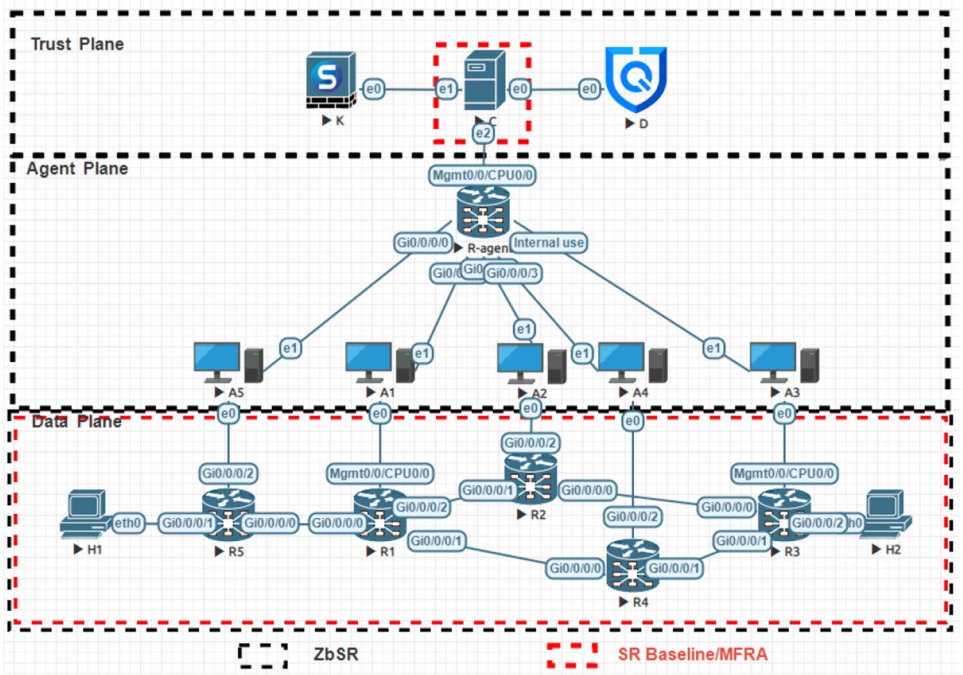

Figure 9. Simulation network topology.

Actual label	Predicted results	
	Malicious	Benign
Malicious	*TP*	*FN*
Benign	*FP*	*TN*

Table 9. Reference table of indicators.

a loop, and whether the ZbSR model, the SR Baseline model can prevent the attack consequences is tested. The third is identity spoofing, that is, the Iperf tool is used to inject background traffic with a specified proportion of traffic characteristics, and part of the traffic is regarded as malicious traffic generated by identity spoofing, then the precision rate and recall rate (the probability of malicious traffic being detected and identified) of the ZbSR model, the SR Baseline model, and the MFRA model are tested. Fourth, DoS attack, that is, according to the simulated network set in Fig. 9, the packet generation rate threshold in the network audit security algorithm is set as 10,000/s, and the hping3 3.0.0 tool deployed on host H1 launches a DoS attack on the data plane with a packet sending rate of about 14,000/s, to test whether the ZbSR model, the SR Baseline model, the DoS attack detection model based on C4.5 and the MFRA model can effectively deal with DoS attack. In order to better evaluate the safety performance of each model, the precision rate and recall rate indexes are defined according to Table 9 as shown in the formula.

$$precision = \frac{TP}{TP + FP} \tag{10}$$

$$recall = \frac{TP}{TP + FN} \tag{11}$$

Message tampering. Configure the Loopback 0 address of the 2.2.2.2/32 for the R2 device, and assign it SID: 16,222. At this time, it is found that the next-hop corresponding to the tag 16,222 in the source route on the R1 device points to R2 by grabbing the packet with the Wireshark tool, as shown in Fig. 10. At this time, the attacker tampered with the control plane message using routing protocol flooding mechanism, etc., set the Loopback 0 address of R4 device to the same 2.2.2.2/32 as R2 and set the link cost value between R1 and R4 to half of the link cost value between R1 and R2. At this time, because the SR Baseline model use none additional security mechanisms, the next-hop corresponding to R1 selection label 16,222 will prefer R4. As shown in Fig. 11, in this case, the traffic path has been tampered. However, According to the SID audit mechanism in the security audit algorithm, the ZbSR model finds that this tampering is a malicious operation and rejects the packet tampering, so

```
RP/0/RP0/CPU0:ios#show mpls forwarding prefix 2.2.2.2/32 detail
Tue Jan 19 03:13:48.354 UTC
Local  Outgoing    Prefix            Outgoing     Next Hop        Bytes
Label  Label       or ID             Interface                    Switched
------ ----------- ----------------- ------------ --------------- ------------
16222  Pop         SR Pfx (idx 222)  Gi0/0/0/0    10.1.1.2        320
     Updated: Jan 19 03:13:09.725
     Version: 696, Priority: 1
     Label Stack (Top -> Bottom): { Imp-Null }
     NHID: 0x0, Encap-ID: N/A, Path idx: 0, Backup path idx: 0, Weight: 0
     MAC/Encaps: 4/4, MTU: 1500
     Outgoing Interface: GigabitEthernet0/0/0/0 (ifhandle 0x01000018)
```

Figure 10. Original R1 device packet.

```
RP/0/RP0/CPU0:ios#show mpls forwarding prefix 2.2.2.2/32 detail
Tue Jan 19 03:10:11.352 UTC
Local  Outgoing    Prefix            Outgoing     Next Hop        Bytes
Label  Label       or ID             Interface                    Switched
------ ----------- ----------------- ------------ --------------- ------------
16222  Pop         SR Pfx (idx 222)  Gi0/0/0/1    40.1.1.4        640
     Updated: Jan 19 03:07:03.111
     Version: 691, Priority: 1
     Label Stack (Top -> Bottom): { Imp-Null }
     NHID: 0x0, Encap-ID: N/A, Path idx: 0, Backup path idx: 0, Weight: 0
     MAC/Encaps: 4/4, MTU: 1500
     Outgoing Interface: GigabitEthernet0/0/0/1 (ifhandle 0x01000028)
```

Figure 11. R1 device packet after tampering.

the traffic path is still shown as Fig. 10. The cross-bitmap algorithm model can also resist the similar tampering attack. The ZbSR model and the SDN cross bitmap algorithm model, which can defend against packet tampering attacks, are repeatedly executed 100 times based on SR network and SDN network respectively. The detection accuracy of the two models are both higher than 97%, and there is no significant difference. However, the SDN cross bitmap algorithm model can only defend against tampering attacks that can cause flow rule conflicts, and its universality is limited.

Routing-loop attack. By pressing the MPLS label stack {R1 → R2 → R3 → R4 → R1} into R1 to construct a loop attack packet, and capturing packets from R1 ~ R4, the message flow of Fig. 12a–e can be obtained in the Baseline model. It can be seen that the stream starts from R1 (the source IP is 1.1.1.1), and the MPLS labels from R1 to R4 pop up hop by hop. However, the ZbSR model detects and discards the loop attack packets through the loop audit algorithm, and the above attack consequences do not occur.

Identity deception. To simulate the real network scene, the Iperf tool is used to inject the background traffic with the ratio of normal traffic to malicious traffic of 3: 1. Let the Smac and SID in the Credit information correspond to the mac and Prefix Node SID of R5, the pSID corresponds to the Prefix Node SID of R1, R2, and R3, the traffic with P as OSPF/ISIS is normal traffic, and the others are malicious traffic. The precision rate and recall rate of malicious traffic (the probability of malicious traffic being detected and identified) can be obtained by statistics, as shown in Fig. 13. It can be seen that based on traffic characteristics, the ZbSR model can perform identity authentication according to the $subalgorithm_{IA}$ in Section "Data exchange security algorithm based on trust evaluation", identify and prevent identity deception attacks with high accuracy, and ensure the credibility of the identity of both communication parties. In contrast, the Baseline model and MFRA model can filter a small amount of malicious traffic thanks to SR native security mechanism.

DoS attack. As shown in Fig. 14, the horizontal axis is the network running time, and the vertical axis is the network processing capacity, which is measured by the retention rate of the source route generation rate (this value is 1 when the network is normal). DoS attack started at 20 s. It can be seen that after the network processing capability based on the ZbSR model is temporarily degraded, the network can locate the injection node of malicious traffic through trust estimation and traffic auditing, and recover the processing capability gradually by filtering the attack traffic. The processing ability of the Baseline model drops rapidly after malicious traffic is injected; in the MFRA model, the network processing capacity is temporarily restored because the backup link is enabled after the congested link is detected, but the backup link also quickly becomes congested. The recovery speed of DoS attack detection model based on C4.5 is faster than that of ZbSR model, because the former adopts mature machine learning algorithm to detect attack traffic. However, this model is similar to the SDN cross bitmap algorithm model, because it cannot resist other types of attacks, and its universality is limited. In

```
Frame 363: 94 bytes on wire (752 bits), 94 bytes captured (752 bits) on interface 0
Ethernet II, Src: 50:0c:00:05:00:05 (50:0c:00:05:00:05), Dst: 50:0c:00:01:00:05 (50:0c:00:01:00:05)
MultiProtocol Label Switching Header, Label: 16222, Exp: 0, S: 0, TTL: 63
MultiProtocol Label Switching Header, Label: 16333, Exp: 0, S: 0, TTL: 64
MultiProtocol Label Switching Header, Label: 16444, Exp: 0, S: 0, TTL: 64
MultiProtocol Label Switching Header, Label: 16111, Exp: 0, S: 1, TTL: 64
Internet Protocol Version 4, Src: 172.18.3.91, Dst: 1.1.1.1
Internet Control Message Protocol
```

(a) The message received by R1 pops up the 16111 label

```
Frame 448: 90 bytes on wire (720 bits), 90 bytes captured (720 bits) on interface 0
Ethernet II, Src: 50:0c:00:01:00:03 (50:0c:00:01:00:03), Dst: 50:0c:00:02:00:03 (50:0c:00:02:00:03)
MultiProtocol Label Switching Header, Label: 16333, Exp: 0, S: 0, TTL: 62
MultiProtocol Label Switching Header, Label: 16444, Exp: 0, S: 0, TTL: 64
MultiProtocol Label Switching Header, Label: 16111, Exp: 0, S: 1, TTL: 64
Internet Protocol Version 4, Src: 172.18.3.91, Dst: 1.1.1.1
Internet Control Message Protocol
```

(b) The message received b) R2 pops up the 16222 label

```
Frame 486: 86 bytes on wire (688 bits), 86 bytes captured (688 bits) on interface 0
Ethernet II, Src: 50:0c:00:02:00:04 (50:0c:00:02:00:04), Dst: 50:0c:00:03:00:04 (50:0c:00:03:00:04)
MultiProtocol Label Switching Header, Label: 16444, Exp: 0, S: 0, TTL: 61
MultiProtocol Label Switching Header, Label: 16111, Exp: 0, S: 1, TTL: 64
Internet Protocol Version 4, Src: 172.18.3.91, Dst: 1.1.1.1
Internet Control Message Protocol
```

(c) The 16333 label pops up for the message received by R3

```
Frame 577: 82 bytes on wire (656 bits), 82 bytes captured (656 bits) on interface 0
Ethernet II, Src: 50:0c:00:03:00:03 (50:0c:00:03:00:03), Dst: 50:0c:00:04:00:03 (50:0c:00:04:00:03)
MultiProtocol Label Switching Header, Label: 16111, Exp: 0, S: 1, TTL: 60
Internet Protocol Version 4, Src: 172.18.3.91, Dst: 1.1.1.1
Internet Control Message Protocol
```

(d) The message received by R4 pops up the 16444 label

```
Frame 937: 78 bytes on wire (624 bits), 78 bytes captured (624 bits) on interface 0
Ethernet II, Src: 50:0c:00:04:00:04 (50:0c:00:04:00:04), Dst: 50:0c:00:01:00:04 (50:0c:00:01:00:04)
Internet Protocol Version 4, Src: 172.18.3.91, Dst: 1.1.1.1
Internet Control Message Protocol
```

(e) R1 receives the message that the MPLS label stack is empty

Figure 12. The process of jumping out MPLS labels from R1 to R4.

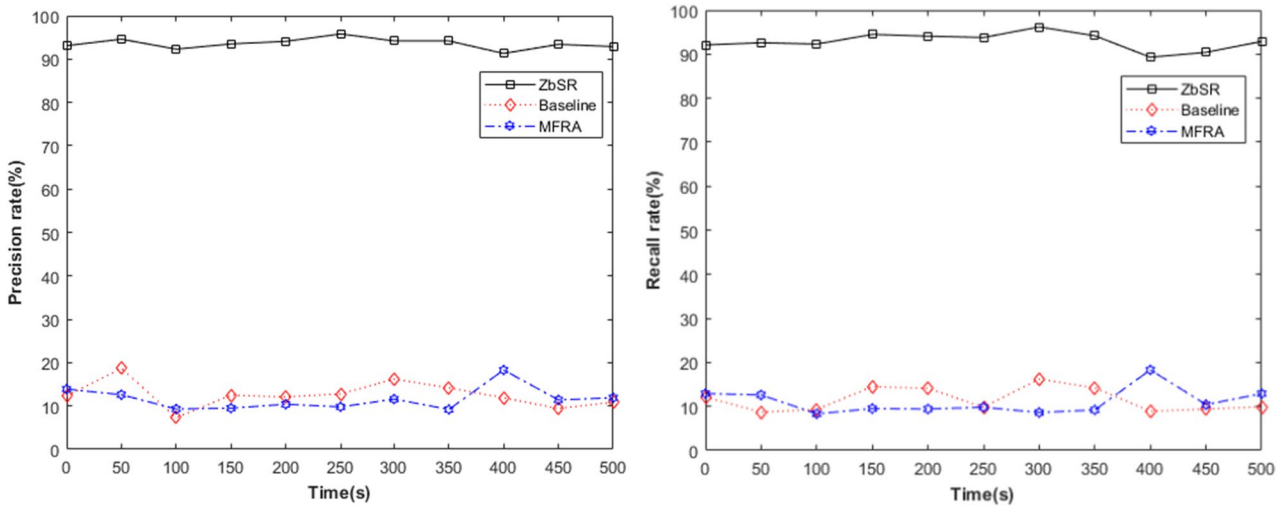

Figure 13. Precision rate and recall rate of malicious flow in 3 models.

the ZbSR model, when the traffic auditing mechanism discovers traffic anomalies, multiple network nodes need to reduce the recommendation trust evaluation value of the malicious node to locate them accurately, so it is more time-consuming.

Performance overhead test and analysis. Since the C4.5 model and the SDN cross bitmap algorithm model are essentially based on SDN and both are implemented based on OpenFlow switch flow table, they are not comparable enough in delay cost testing. Therefore, the ZbSR model, SR Baseline model and MFRA model are analyzed in this paper. The streaming transmission delay of the 3 models in the network with 5, 11, and 15 nodes is shown in Fig. 15.

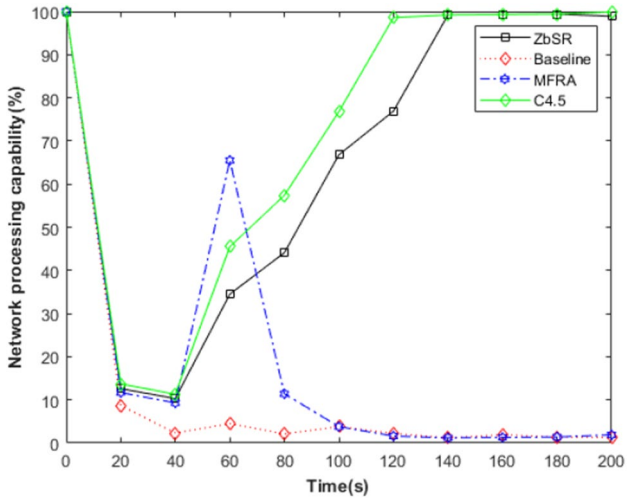

Figure 14. Comparison of the decline of the processing capacity of 4 models.

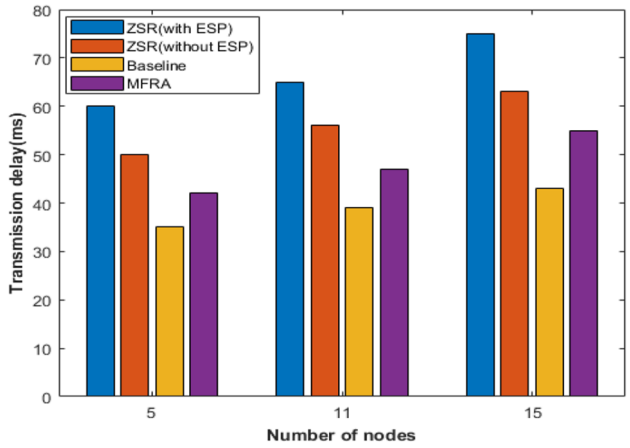

Figure 15. Comparison of streaming transmission delay of 3 models.

It can be seen that when software encryption is not turned on (ZbSR (unencrypted)), compared with the Baseline model and the MFRA model, the time delay of the ZbSR model increases by 18.2–22.7% (average incremental time delay is 19.3%) and 8.5–12.3% (average incremental time delay is 10.4%), respectively, and the proportion of time delay increased compared with MFRA model decreases with the number of nodes. After software encryption is turned on (ZbSR (encrypted)), the delay of the ZbSR model is further improved, which is 37.4–41.1% higher than that of encryption without it, which is consistent with the cost analysis in Section "ZbSR model security overhead". Considering that if the application layer of the SR network terminal device has a mature encryption mechanism, there is no need to enable the terminal device encryption of the ZbSR model, and the Baseline model and MFRA model will also significantly increase the delay after encryption is turned on, so the security cost of the ZbSR model is regarded as 19.3% of the average incremental delay compared with the Baseline model when encryption is not turned on. To reduce the security overhead of the model, we can consider introducing particular data encryption components to replace software encryption in the terminal device; besides, the ZbSR model should be configured on-demand to focus on auditing the backbone network nodes with a large degree of nodes or large flow.

Conclusion and future work

This article analyzes the untrustworthy security problems of network elements and PKI/CA in the zero-trust network environment of SR, and points out that it can be attributed to the untrustworthiness of SR data exchange and network audit functions, but there is no corresponding supporting security mechanism at present. Focusing on the application of ZTA in the SR-BE/TE network to improve its data plane security performance, this paper proposes a ZbSR data plane security model based on ZTA and the corresponding data exchange and network audit security algorithms. Through simulation test, the proposed model can provide various security protection for SR-BE/TE data plane, but also exposes its disadvantages of high-security cost. In the next step, we will focus

on the hardware design of security components, the improvement of the trust evaluation algorithm for trust renewal, and the incremental network attack surface introduced by the model.

Data availability

The data and algorithms in our graphs and tables only come from the research process itself, without using public data sets or publishing unavailable data.

Received: 3 January 2022; Accepted: 14 November 2022
Published online: 29 November 2022

References

1. Ventre, P. L. *et al.* Segment routing: A comprehensive survey of research activities, standardization efforts and implementation results. *J. IEEE Commun. Surv. Tutor.* **99**, 1–1 (2020).
2. Clarence, F., Kris, M. & Ketan, T. Segment routing-part I (2017).
3. Pier, V. *et al.* Segment routing: A comprehensive survey of research activities, standardization efforts, and implementation results. *J. IEEE Commun. Surv. Tutor.* **23**(1), 182–221 (2021).
4. Segment Routing over IPv6 (SRv6) Network Programming. RFC 8986:1-40 (2021).
5. Geng, H. Intra-domain routing protection scheme in segment routing architecture. *J. Comput. Eng. Appl.* **55**(08), 80–85 (2019).
6. Medved, J., Lopez, V., Crabbe, E. *et al.* OSPF extensions for segment routing (2015).
7. Akiya, N., Pignataro, C. & Kumar, N. Segment routing (SR) (2013).
8. Liu, Q. & Shi, L. Segment routing technology and its application analysis. *J. Telecommun. Technol.* **525**(12), 56–58 (2017).
9. Evan, G. & Doug, B. Zero-trust networks: Build security systems in untrusted networks (2019).
10. Barclay, O., Justin, M., Betsy, B. & Max, S. BeyondCorp: Design to deployment at google. *J. Login Usenix Mag.* **41**(1), 28–35 (2016).
11. Clarence, F., Kris, M., Francois, C. *et al.* Segment routing-part II (2019).
12. Abraham, Y., Adrian, P. & Dawn, S. StackPi: New packet marking and filtering mechanisms for DDoS and IP spoofing defense. *J IEEE J Sel Areas Commun* **24**(10), 1853–1863 (2006).
13. Bryan, P., Adrian, P. & Dave, A. SNAPP: Stateless network-authenticated path pinning. In *Proceedings of the 2008 ACM Symposium on Information, Computer and Communications Security*, ASIACCS 2008, Tokyo, Japan, March 18–20, 168–178 (2008).
14. Jad, N., Michael, W., Antonio N. *et al.* Verifying and enforcing network paths with ICING. In *Proceedings of the 2011 Conference on Emerging Networking Experiments and Technologies*, Co-NEXT '11, Tokyo, Japan, December 6–9, 1–12 (2011).
15. Hao, C. & Tilman, W. Source authentication and path validation with orthogonal network capabilities. In *2015 IEEE Conference on Computer Communications Workshops*, INFOCOM Workshops, Hong Kong, China, April 26–May 1, 111–112 (2015).
16. Lepinski, M. & Kent, S. An infrastructure to support secure internet routing. RFC 6480, 1–24. https://doi.org/10.17487/RFC6480 (2012).
17. Tomas, H., Italo, C., Yossi, G. *et al.* DISCO: Sidestepping RPKI's deployment barriers. In *Network and Distributed Systems Security (NDSS) Symposium*, San Diego, CA, USA, February 1–17 (2020).
18. Xin, Z., Zhou, Z. W., Geoffrey, H., Adrian, P. & Virgil, D. Network fault localization with small TCB. In *Proceedings of the 19th Annual IEEE International Conference on Network Protocols*, ICNP 2011, Vancouver, BC, Canada, October 17–20, 143–154 (2011).
19. Peng, Z., Hao, L., Hu, C. C., Hu, L. J., Xiong, L., Wang, R. L. & Zhang Y. M. Mind the gap: Monitoring the control-data plane consistency in software defined networks. In *The 12th International Conference on Emerging Networking Experiments and Technologies*, Irvine, CA, USA, December 12–15, 19–33 (2016).
20. Zhang, X. *et al.* DFL: Secure and practical fault localization for datacenter networks. *J. IEEE/ACM Trans. Netw. (ToN)* **22**(4), 1218–1231 (2014).
21. Jia, Y. H. *et al.* RISP: An RPKI-based inter-AS source protection mechanism. *J. Tsinghua Sci. Technol.* **23**(1), 1–12 (2018).
22. Hari, A. & Lakshman, T. The Internet blockchain: A distributed, tamper-resistant transaction framework for the Internet. In *Proceedings of the 15th ACM Workshop on Hot Topics in Networks, Atlanta, GA, USA, November 9–10, 204–210 (2016).*
23. Fu, Y. P. Research on segmented routing technology of 5G bearer network. *J. Digit. Commun. World* **3**, 17–18 (2020).
24. Pier, L. V., Stefano, S., Marco, P., Antonio, C., Ahmed, A., Clarence, F., Pablo, C. & Francois, C. Segment routing: A comprehensive survey of research activities, standardization efforts, and implementation results. *IEEE Commun. Surv. Tutor.* **23** (1), First Quarter (2021).
25. Behringer, M. RFC 4381-analysis of the security of BGP/MPLS IP virtual private networks (VPNs). February (2006).
26. Fang, L. E. RFC 5920—Security framework for MPLS and GMPLS networks. July (2010).
27. Tang, Y. *et al.* Deterministic security protection method for segmented routing. *J. Netw. Secur. Technol. Appl.* **11**, 3–7 (2021).
28. Zhang, N. N., Zhu, K. & Zhu, D. D. Research on DoS attack and its countermeasures. *J. Internet World* **12**, 103–105 (2016).
29. Zhu, Q. Research on PKI CA authentication technology. *J. Cybersp. Secur.* **7**(Z1), 37–39 (2016).
30. Wang, L. *et al.* A data plane security model of segmented routing based on SDP trust enhancement architecture. *Sci. Rep.* **12**(1), 1–21 (2022).
31. Yu, X. Y., Sun, G. & Zhang, Y. W. Research on software-defined boundary network stealth technology based on zero trust. *J. Commun. Technol.* **54** (5), 6.
32. Singh, J., Refaey, A. & Koilpillai, J. Adoption of the software-defined perimeter (SDP) architecture for infrastructure as a service. *J. Can. J. Electr. Comput. Eng.* **43**(4), 357–363 (2021).
33. George, A. S. & Aremu, B. Software-defined perimeter (SDP): The next-generation secure VPN solution built for future networks. In *4th International Online Multidisciplinary Research Conference (IOMRC-2020)* (2020).
34. Zhang, Q. G., Huang, H. & Wang, Y. J. Research on software-defined boundary security model based on zero trust. *J. Inf. Technol. Inf.* **248**(11), 98–100 (2020).
35. Moghadam, M. F. A key management schema based on ECC to secure the substation and control center communications in smart grids (2019).
36. Wang, L. X. & Du, G. Z. Research on data encryption technology of computer network. *J. Netw. Secur. Technol. Appl.* **234**(06), 37–38 (2020).
37. Stalling, W. *Data and Computer Communication* 3rd edn. (Tsinghua University Publishing House, 1997).
38. Arzo, S. T. *et al.* Multi-agent based autonomic network management architecture. *J. IEEE Trans. Netw. Serv. Manag.* **99**, 1–1 (2021).
39. Dierks, T. The transport layer security (TLS) protocol version 1.2. RFC (2008).
40. Hui, X. *et al.* An attack-resistant trust inference model for securing routing in vehicular Ad Hoc networks. *IEEE Trans. Veh. Technol.* **68**(7), 7108–7120 (2019).
41. Qin, B. *et al.* Progress of key agreement protocol. *J. Comput. Sci.* **09**, 9–12 (2008).
42. Huang, J. Y. *et al.* A multi-fault recovery and avoidance mechanism based on SR in SDN. *J. Electron. Sci. Technol.* **2017**(11), 195–202 (2017).

43. Liu, J. SDN network security research and implementation. Nanjing university of posts and telecommunications. https://doi.org/10.27251/d.cnki.gnjdc.2019.000346 (2021).
44. Liu, J. *et al.* DDoS attack detection based on C4.5 decision tree in SDN. *Comput. Eng. Appl.* **55**(20), 84–881+27 (2021).

Acknowledgements
Here, we are deeply grateful to all reviewers and editors.

Author contributions
Investigation, L.W.; Resources, Z.Y.L.; Writing—original draft preparation, L.W. and J.C.P.; Writing—review and editing, L.W.; J.Z. and T.H.; Supervision, H.L.M. All authors have read and agreed to the published version of the manuscript.

Funding
This research was funded by the Innovation Scientists and Technicians Troop Construction Projects of Henan Province (224000510002).

Competing interests
The authors declare no competing interests.

Additional information
Correspondence and requests for materials should be addressed to H.M.

Reprints and permissions information is available at www.nature.com/reprints.

Publisher's note Springer Nature remains neutral with regard to jurisdictional claims in published maps and institutional affiliations.