

Secure Link State Routing Protocol: A Framework for Network Survivability

Dijiang Huang, Amit Sinha, Deep Medhi
School of Interdisciplinary Computing & Engineering
University of Missouri-Kansas City
Kansas City, MO 64110 USA

Abstract—

Network survivability has been studied extensively from the view of node and link failures. The domain of survivability goes beyond just the physical failures and one needs to address this issue when faced with security threats that can render the network logically dysfunctional without causing any physical damage. There are several security threats a network routing protocol such as the link-state protocol may face. In this paper, we argue that this leaves network routing and, in turn, network susceptible to a number of attacks. We discuss these attacks and present a new secure routing framework based on security techniques (authentication and confidentiality) for a link state routing protocol. The central idea of the new framework is to divide a routing domain into multiple virtual routing domains and creating a hierarchical trust among them to address the survivability issue of the network. We illustrate that imbedding this framework to an existing protocol can be done without major modifications to the existing protocol, and without affecting its operational features. Our assessment shows that the framework is capable of preventing many currently unchecked security threats without unduly overloading routing elements in routers. Our framework also provides survivable capability for the subset of functional network routers in the event of formidable attacks like the insider attack.

*Index Terms—*Survivable Network, Link State Network Routing, Security.

I. INTRODUCTION

In this paper, we are interested in developing a robust framework for network survivability against security threats, by making the link-state routing protocol secure. Currently, link-state protocols such as OSPF and IS-IS are widely used for intra-domain routing. But, there are many security issues that the current routing protocol framework does not address. We thus start with two broad security issues to lay out the purpose of this paper.

The *first* security issue with the current routing framework is its unclarity of non-host based threats for link state network routing protocol, such as the threats targeting at

origination, verification and transmission of routing data. Thus, it is possible that an attacker can wiretap on the transmission link to inject anything he wants, and an attacker can also have the capability to “hijack” the network router and then breach the network as desired.

The *second* security issue which has not received enough attention in network routing is confidentiality. For example, confidentiality was not considered in OSPFv2 or has been considered as optional in OSPFv3 [11]. An advantage of confidentiality is that it can guard against passive attacks, such as wiretapping. The routing information (clear text) can be easily intercepted on unprotected network segments in its absence. Since the fundamental communication operation of link-state routing protocol is flooding, the attacker can very easily intercept all routing information just from one network segment and can use it to analyze network topology and traffic patterns. These information can help attacker in exploiting the weaknesses of the network and launch more devastating and highly efficient attacks. For example, an attacker can split the network and disable it by attacking the minimal number of nodes that partition the network¹.

In both cases above, the network could face dire consequences if it is not designed to survive such threats. Keeping the above in mind, we propose a framework for secure link-state routing protocol to address some threats not addressed currently in the literature. A key feature of our framework is the creation of a concept called *virtual routing trust domain* (VTRD) via systematically hiding routing information; this is addressed *without* changing the overall routing administration domain. In order to make this work, we propose using a scalable secure group keying scheme for secure group communication. We also identify what changes/modifications are necessary in current link-state routing protocols for our framework. We

¹We show an attack tree of network routing denial of service in Appendix III. It shows how important it is to prevent attacker from knowing the network topology.

then perform computational analysis to show the impact of the overhead compared to other approaches. Once convinced that there is no significant increase in computation cost, we further evaluate the survivability of the network in our proposed framework.

The rest of the paper is organized as follows. In Section II we present the related work. Section III we describe the preventive cryptographic countermeasures for various attack forms against the link state routing protocol. In Section IV we highlight the security requirements to build the secure link state routing framework. In Section V, we present a simple way to construct multiple virtual routing domain. Then, we discuss the computational impact on router's CPU and communication overhead in section VI. In Section VII, we present the evaluation for survivability for our proposed framework. Finally, Section VIII summarizes our work.

II. RELATED WORK

There have been several work to understand the security threats in order to make link state routing protocol robust against attacks. A trend-setting work in this area was done by Perlman [28]. She developed a secure link state routing infrastructure to tackle Byzantine failures. In this model, a strong asymmetric cryptographic key scheme (public key scheme) was used to provide signatures for each link-state advertisement (LSA), thus have a high computation overhead making it difficult to implement.

The importance of security in the design of routing protocols and network robustness has received some attention in recent years. The current implementations of the two most popular link state routing protocols, OSPF [23] of TCP/IP suit and IS-IS [17] of OSI reference model, provide packet-level authentication. It was however found that since authentication was done hop-by-hop, it could not prevent subverted router modifying or injecting malicious routing packets [27]. Therefore, other researchers have proposed hash chain based schemes [6][12] to provide data origin authentication (from validation point of view, this was an end-to-end authentication and every router was part of the end system). These schemes used hash values as a credential or a key used for Keyed-Hashing for Message Authentication (HMAC)[19], which was computationally efficient; but it required a loose synchronous mechanism, therefore conflicting with the operation mode of the routing mechanism, which is not synchronized. Murphy *et al* have proposed the use of digital signature for OSPF (RFC 2154 [26]) that used public-key encryption. Around the same time frame, Vetter, Wang and Wu studied the insider attacks for OSPF [31]. They claim that this issue can be left to the intrusion detection

system as in most cases the intruder will be recognized if within a certain amount of time. While this might work at the cost of some unavoidable damage to the network, a preventive method is more desirable.

As the necessity to secure the routing infrastructure grows, it can be argued that security enhancements is desired to be globally present. The latest internet draft by Gupta and Melam [11] suggests that OSPF for IPv6 [8] should rely on the security framework of IPsec [18]. IPsec provides security services such as authentication, integrity, replay detection, and encryption to protect routing protocol traffic. The global deployment of IPsec protocols will provide a set of powerful tools that would seamlessly interoperate throughout the Internet, under the assumption that the next generation of IP, IPv6, is also globally present. However, the mere provision of such services can not secure the routing protocol itself. IPsec can deter outsider attacks and disallow the injection of unauthorized routing traffic by securing the point-to-point exchange of routing updates at the network layer, but can not enforce or guarantee correct operation of the routing protocol in case of an insider attack. Furthermore, multicasting using IPsec is still being researched.

Recently, the Internet Engineering Task Force (IETF) has formed a new working group – Routing Protocol Security Requirements (rpsec) [16] to address the security requirements for routing protocols. A recent survey by Papadimitratos and Hass highlights the fact that the counter measures proposed so far has not eradicated the vulnerability of the routing infrastructure [27]. Another recent work by Chakrabarti and Manimaran provides a taxonomy of security attacks and stresses the need to develop architecture, algorithms and protocols for securing the Internet infrastructure [5].

III. MOTIVATIONS

We cite the following paragraphs from the CERT® document [14] to highlight the importance of security for routing infrastructure:

“...One of the most recent and disturbing trends we have seen is an increase in intruder compromise and use of routers. ... Reports indicate routers are being used by intruders as platforms for scanning activity ...”

“Routers make attractive targets for intruders ... a part of the network infrastructure ... routers are often less protected by security policy and monitoring technology ...”

“... attacks based on direct attacks against the routing protocols that interconnect the networks comprising the Internet. We believe this to be an

imminent and real threat with a potentially high impact...”

The above is an indication of the fact that bringing down a network infrastructure without causing any physical damage to the network entities, is quite conceivable. The lack of strong protection by using cryptographic techniques as part of the original routing protocol design has allowed malicious users to exploit the network in numerous ways. In time, the importance of making the routing protocol robust has also emerged and several standards have come up to handle a variety of threats. In spite of this, the network routing remains vulnerable in both the routing protocol itself and routing functionalities involved with sending and receiving routing data. While most of the work done so far, as discussed in Section II, in order to contain the threats quoted above has been to analyze and contain the security threats, but not from the point of view of network survivability. Therefore, our first goal is to identify the vulnerabilities of existing routing system and then to develop security enhancements to be deployed for network routing protocols and a survivable routing framework.

In this section, our discussion focuses on the origination, verification and transmission of routing data. In Appendix I, we give a description of the security threats specifically for the link state routing. We discuss here, the security mechanisms that is known, as of now, to prevent some of these threat actions from taking place. Based on the threat model presented in Appendix I, we discuss the vulnerabilities of existing network routing security mechanisms and the possible improvements therein.

A. Preventive Cryptographic Countermeasures Against Attacks

The challenges posed due to the enormity and diversity of the threats has led to several work in the recent years that address techniques to safeguard a network. On the other hand, the current standards for network routing protocols have not incorporated all the techniques required to make it as foolproof as possible. That is, a set of unplugged security holes remains there that an adversary can use to paralyze the network. In this section, we analyze the possible preventive cryptographic countermeasures first and then describe how they can prevent attacks from taking place totally or partially.

1) *Preventive cryptographic countermeasures*: Table I enlists two preventive cryptographic countermeasures that are described in the literature, including those that have found a place in protocol standards. The two main preventive cryptographic countermeasures that has been suggested for routing protocols are authentication and con-

identiality. In network routing environment, peer entity authentication and data origin authentication are two types of authentication services. Peer entity authentication happens mostly in neighborhood relation set-up procedure and data origin authentication happens mostly in routing information exchanging procedure. Note that data origin authentication service provides verification that the identity of the original source of a received data unit is as claimed; there can be no such verification if the data unit has been altered. Thus, by definition, authentication services depend on companion data integrity services. Confidentiality ensures that no unauthorized device can decipher the routing information on its way to the destination.

In this paper, our discussions focus on data origin authentication and confidentiality. More specifically, these two countermeasures can provide protection at the Packet Level or Information Level (we call these two types of data origin authentications as PA and IA respectively). By PA we mean, the authentication is processed for a routing update packet or an IP packet that contains the routing update as payload. IA provides protection for each and every routing information carried within a routing update packet. Besides PA and IA, there are another two important concepts we need to introduce into our discussion; they are *hop by hop* (HBH) and *end to end* (ETE). HBH means that the generation and verification of authentication code are performed by every forwarding router. ETE means that the generation of authentication code is performed only at the source; all the forwarding routers and termination routers are part of the end system, and they only perform verification. We analyze the combinations between PA, IA and HBH, ETE. For brevity, we identify each mechanism with a label; this is noted in Table I.

In link state routing protocol, pieces of routing information – *link state advertisements* (LSAs) are encapsulated in a *link state update* (LSU) packet. Most of current implementations fall into the category of PA_H . If PA_H is provided for entire LSU (a routing packet), then PA_H can guard against “man-in-the-middle” attack [29]. But, in link state routing, flooding is used for distributing LSAs within a link state routing domain; PA_H can not prevent any intermediate subverted router from modifying forwarded LSAs or router from originating forged LSAs. ETE is more desirable to provide stronger protection for LSAs. But, another difficulty of link state routing is that multiple LSAs are encapsulated within a single LSU packet and the content of each LSU that originated from different router may be *different*. This prevents PA_E from being implemented efficiently. Hence, IA_E and IA_H are required to provide information level protection. OSPF with digital signatures [26] is an exam-

TABLE I
SECURITY MECHANISMS

Methods		Label	Description	Protection
Authentication (A)	Packet Level	PA_H^\dagger	Packet level, hop by hop authentication	Data Origin Authenticity
		PA_E^\S	Packet level, end to end authentication	
	Information Level	$IA_H^\mathcal{L}$	Information level, hop by hop authentication	
		IA_E^\ddagger	Information level, end to end authentication	
Confidentiality (C)		C_P^\S	Confidentiality for the whole packet	Information Availability
		C_I^\P	Confidentiality for the information within the packet	

† : OSPFv2 RFC2328. ‡ : OSPF extension RFC2154. § : OSPFv3 tentative Internet draft.
 $^\mathcal{L}$: Proposed by Huang et al [13]. ¶ : Have not yet proposed.

ple of IA_E , while the double authentication scheme [13] is an example of IA_H .

For confidentiality too, we differentiate between packet level and information level, which is shown in Table I. OSPF running over IPsec [11] is an example of providing C_P , which provides confidentiality for IP payload. Providing confidentiality for each LSA individually is represented by C_I .

2) *Using preventive cryptographic countermeasures to guard against attacks:* Here, we analyze how to use cryptographic countermeasures presented in Table I to guard against the threat actions illustrated in Fig. 6 of Appendix I. Table II presents the mapping of threats and corresponding countermeasures. The threat actions marked with \checkmark are all outsider attacks. Attacks (b) can be easily guarded against by using PA_H . The dummy routing traffic due to attack (c)-(ii) can be filtered out using PA_H . Although, cryptographic-based operation can aggravate the CPU computation burden, the overload attack is usually limited within a small range where it happens. This is because the excess routing traffic can not get through a router. This may be useful in preventing *distributed denial of service* (DDOS).

Note that preventive countermeasures, such as authentication and confidentiality, can not prevent attacks that are marked with \times ((e)-(ii) and (f)-(i)). These attacks need other security mechanisms (such as admission/access control, intrusion detection, etc.), which are not addressed in this paper.

In this paper, our discussion is focused on the countermeasures marked from \star to $\star\star\star$. The countermeasures marked by \star are specified in current literature, such as OSPF with digital signatures² [26]; countermeasures marked by $\star\star$ is barely addressed in

²End-to-end authentication is considered as a strong preventive cryptographic countermeasures. The only proposal that widely accepted is the one that uses public key scheme to sign each LSA. The reason we separate it from PA marked by \checkmark is because of the deployment difficulty of digital signature, which comes with high computation overhead compared with traditional authentication scheme based on keyed

the current literature, while countermeasures marked by $\star\star\star$ have not been addressed so far.

Guard Against Attacks on Communication Links: As shown in Table II, attacks from (a) to (d)-(i) are injected on the communication link. Here, we investigate the possible use of preventive cryptographic countermeasures when attacks (a), (c)-(i) and (d)-(i) occur (marked with $\star\star$ and $\star\star\star$).

For scenario (a): C_P or C_I can be used to prevent outsiders from sniffing packets containing routing information. This is the very straight forward method to prevent passive attack. When C_P is provided for whole IP payload, the outsider can not know some general information, such as link state type, advertising router, sequence number, etc which are contained within routing packet header. These information can help attacker to derive network topology. C_I can not prevent attacker from knowing the information within the routing packet header, but it can prevent some subverted router from decrypting the routing information when they use different encryption/decryption keys. The combination of C_P and C_I will provide strong security features to guard against ineligible entities.

For scenario (c)-(i): We assume that there is an admission control mechanism to prevent outsiders from using some network tools, such as “traceroute”, to derive network topology. This can also be done by simply disabling those network services. Then, an attacker might arbitrarily wiretap any possible communication links to intercept the routing information. Plain text routing information can help attackers to derive network topology and traffic allocation pattern. Due to flooding of the routing information by link state routing protocol, tapping one link can help intercept all flooded LSAs within its routing domain. The intercepted routing information can be valuable for attackers to decide the location of an attack target, such as

hash function (HMAC) [19].

THREATS AND CORRESPONDING CRYPTOGRAPHIC PREVENTIVE COUNTERMEASURES

Threats [¶]		Attack Types		Preventive Countermeasures	Remarks	
Threat actions (Attacks)	Label	I/O	P/A			
Wiretapping		(a)	O	Passive	C_P or C_I	★★
Outsider falsification & masquerade	Substitution	(b)-(i)	O	Active	PA_H	√
	Insertion	(b)-(ii)	O	Active	PA_H	√
	Masquerading	(b)-(iii)	O	Active	PA_H	√
Outsider obstruction	Interference	(c)-(i)	O	Active	C_P or C_I	★★★
	Overload	(c)-(ii)	O	Active	PA_H	√
Replay	Outsider replay	(d)-(i)	O	Active	New keys	★★
	Insider replay	(d)-(ii)	I	Active	New keys & C_I or IA_E	★★★
Insider falsification & masquerade	Substitution	(e)-(i)	I	Active	IA_E	★
	Insertion	(e)-(ii)	I	Active	n/a	✕
	Masquerading	(e)-(iii)	I	Active	IA_E	★
Insider obstruction	Stop forwarding	(f)-(i)	I	Active	n/a	✕
	Overload	(f)-(ii)	I	Active	C_I	★★★
Repudiation	False denial of origin	(g)-(i)	I	Active	IA_E or IA_H	★
	False denial of receipt	(g)-(ii)	I	Active	PA_H	★★★
Exposure	Insider undeliberate exposure	(h)-(i)	I	Active	C_P^{\dagger} or C_I^{\ddagger}	★★
	Insider deliberate exposure	(h)-(ii)	I	Active	C_I	★★★

[¶]: Threats description and label refer to Appendix I.

I/O: Insider/Outsider (attacks). P/A:Passive/Active (attacks).

[†]: Guard against outsider attacks. [‡]: Guard against insider attacks.

√: Solvable via well known solution and widely deployed.

★: Solvable via well known solution but less deployed.

★★: Solvable via proposed solution in this paper.

★★★: Partially solvable via proposed solution in this paper.

✕: Unsolvable via authentication and confidentiality.

the weakest communication links or the partition routers. We note that providing confidentiality can not prevent attacker from doing active attacks. But, without network topology and traffic pattern information, it will make it harder for the attacker to deploy attack successfully. Thus, C_P and C_I can be helpful. Most of the attacks we present in this paper need some network topology or traffic pattern information. We present a particular attack tree for overload attack in Appendix III.

For scenario (d)-(i): Link state routing protocols typically use a non-decreasing sequence number to prevent replay attack. But, the replay attack can still take place when the sequence number is rolled over or a router reboots. OSPF based on IPSec [18] can benefit from its anti-replay window mechanism to prevent replay attack. This mechanism can detect most of the replay attacks, but it is possible that some attacks might go undetected. For e.g, consider this scenario: an anti-replay window size is 20 for a router. It starts with sequence number 1; when the sequence number reaches 10, the router reboots. Then it restarts and begins with sequence number 1 again. Now the LSA with sequence number 10 is captured by an attacker, and after the router restarts,

the network link metric is changed. Assume that all security parameters to be the same before and after the router restarts. Then, the attacker can replay the old LSA successfully. If we change security parameters after a router restarts, the outsider replay can be prevented. For example, if authentication or confidentiality is provided for routing packet/information, a new updated authentication/encryption/decryption key can be used for new LSUs/LSAs. This requires the key management to be involved with general network routing operations.

Guard Against Attacks on Routers: We discuss the possible use of preventive cryptographic countermeasures when attacks take place on a router. Insider attacks (d)-(ii), (e)-(i), (f)-(ii), (g) and (h) are some examples of attacks on routers (marked with ★, ★★ and ★★★).

For scenario (d)-(ii): Analysis of (d)-(ii) is similar to (d)-(i). The difference is that the information security protection is required. We illustrate the reason through a simple example: LSA contains the bandwidth information of a particular link l which is c_l . When all routers share a common key to sign/encrypt the LSA, any subverted insider can replay the LSA and possibly change the crypto

key. If we allocate different set of keys to routers and we sign/encrypt only sub-portion of bandwidth of link l , say c'_l , the insider attacker can only replay the old routing information c'_l , which only affect part of network resource. This would mean that not all routers within the link state routing domain share the same network resource information. If we consider a link state routing domain as a trust domain, via C_I , we can differentiate it as multiple sub-trust domains. We call the sub-trust domains as *virtual trust routing domains* (VTRDs) which will be explained in more details in the next section. The fundamental idea of VTRDs is to build multiple sub-routing systems on top of the same physical network, so as to protect them from each other in case of attacks that we have discusses earlier, so that, if one of the sub-routing system is compromised, it does not affect others or can only cause minimal damages to other sub-domains. For example, a router belonging to a VTRD-A is considered as an outsider of VTRD-B. Thus, providing confidentiality (C_I) is desired for each individual LSA, and only the eligible VTRD members can decipher these LSAs. The described attack scenario of (d)-(i) can still occur. Since all the router share the crypto key, updating it will not be helpful to prevent the insider from replaying the old routing packets. Thus, building multiple VTRDs through C_I can limit the effect of attack d-(ii) on the network. We will discuss the structure of VTRD in details in Section IV.

For scenario (e)-(i) and (e)-(iii): In case of attacks (e)-(i) and (e)-(iii), IA_E can prevent subverted routers from substituting the routing information and masquerading other routers, where data origin authentication is provided for each LSA and is guaranteed end-to-end.

For scenario (f)-(ii): This scenario is to address excessive routing information burden on routers through overloading routers' input buffer or CPU. It is different from outsider overload attack (c)-(ii), which only aims to overload the immediate routers' input buffers or CPU. Insider overload can cause more damage, because the attacked routers will forward the excessive routing information to next hop due to flooding. Although some link state routing protocols set minimal arrival interval to constrain a router to receive any particular LSA updates, a subverted router can invent totally new LSA instances constantly. This will spread throughout the network due to flooding.

In order to constrain the routing data traffic overload attack, we again need to divide the link state routing domain into multiple small link state routing domains (VTRDs). We assume here that the network has the link bandwidth management capability. An illustrative example is shown in Fig. 1 which represents a network segment containing three routers. The capacity of a communication link 1-2

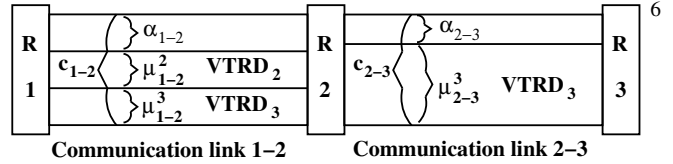


Fig. 1. Communication link under attack

is c_{1-2} and $VTRD_2$ and $VTRD_3$ are configured to run through link 1-2. The bandwidth allocated for these two VTRDs are μ_{1-2}^2 and μ_{1-2}^3 (the superscript is the identifier of a VTRD and subscript is the identifier of a link). The available bandwidth of link 1-2 is given by α_{1-2} . Then, we have $c_{1-2} = \mu_{1-2}^2 + \mu_{1-2}^3 + \alpha_{1-2}$. Similarly, the capacity allocation of link 2-3 is $c_{2-3} = \mu_{2-3}^3 + \alpha_{2-3}$. Note that the reserved bandwidth is guaranteed through both bandwidth management (such as scheduling) and confidentiality provided for each VTRD. For example, a particular encryption/decryption VTRD session key is used to provide confidentiality for a VTRD. In our example, both routers R1 and R2 can decrypt LSAs for $VTRD_2$ and $VTRD_3$, R3 can only see the available bandwidth allocation for $VTRD_3$. R3 do not possess the session key used by $VTRD_2$, it can not forge routing information to announce allocated bandwidth on link 2-3 for $VTRD_2$. When subverted router R3 overloads R2, R2 would not forward excessive routing traffic which exceeds μ_{1-2}^3 . The traffic control is done through bandwidth management. In this example, the bandwidth management is used to enforce the network resource allocation and the session key is used to setup trust among routers, which specify how much network resource is to be trusted among routers.

For scenario (g)-(i): In case of attacks (g)-(i), IA_E can prevent insiders from denying the origination of sending false routing information. The authentication code should provide the evidence that the sender can not deny, for example by using digital signature. IA_H proposed in *Double Authentication* [13] can help to build an authentication chain from the sender to the receiver, which can provide nonrepudiation services.

For scenario (g)-(ii): The acknowledgement mechanism of a link state routing protocol is neighbor-to-neighbor based. Multiple LSAs can be acknowledged by a single link state acknowledgement packet. The acknowledgement packet can use shared key between the communication peers to authenticate the received packet. But we know, using a shared key scheme and the neighbor-to-neighbor authentication mechanism, there is no way to explicitly tell who generates the packets due to key sharing. Use of IA_E and IA_H for acknowledgement of every LSA is impractical and unnecessary. Moreover, the receiver can stop responding to acknowledgement. Thus,

using PA_H for acknowledgement is optional and can only benefit from preventing “man-in-the-middle” attack.

For scenario (h)-(i): An insider may unintentionally expose routing information to outsiders or other insiders that are not necessarily receiving the routing information (for example, the communication via wireless links). The analysis of this scenario is the same as scenario (a). C_P ensures no outsider can reveal the content. Within multiple VTRDs framework, C_I ensures only eligible VTRD members can reveal the content within its VTRD.

For scenario (h)-(ii): An insider can deliberately expose the routing information to anyone. But, with the routing information protected by C_I , a subverted/compromised router can not expose the routing information of other VTRDs, which it does not belong to.

B. Summary

In Table II, attacks marked with ★ ((e)-(i), (e)-(ii) and (g)-(i)) can be foiled by using IA_E or IA_H . Attacks marked with ★★ ((a), (d)-(i) and (h)-(i)) can be thwarted by using C_P , C_I or key management. We can use preventive cryptographic countermeasures to provide degree of protection to link state routing from attacks marked with ★★★. Limiting these attacks ((c)-(i), (f)-(ii) and (h)-(ii)) is a challenging task.

We have given the security analysis of preventive cryptographic countermeasures that can be used to guard against attacks marked with ★★ and ★★★. We make the following observations:

- Current network routing lacks a framework for survivability under security threats to routing protocol.
- Since, the fundamental functions among network routers are cooperation and coordination, the attacks marked with ★★★, except (g)-(ii), can not be totally prevented via preventive cryptographic countermeasures. Thus, using confidentiality and key management (the use of C_I) provides a natural way for us to divide the routing domain into multiple small size *virtual trust routing domains* (VTRDs). Each VTRD represents a subset of network resource. In this way, we can limit attacks within the corresponding VTRD. The philosophy behind this is that a router need to know only the necessary routes go through it.
- Network routing is located within network control plane, which provides network resource allocation information for network data plane. VTRDs divides the network resource allocation into multiple small subset within the control plane. Mapping between VTRDs and network services (in data plane) is out of scope of this paper.

Since within a routing domain, an attacker can easily⁷ exploit the system for security holes and deploy attacks that can go unchecked, the flat trust structure (as it exists now) for the entire intra-domain routing is extremely susceptible to such attacks. Our proposed approach is to divide such a routing domain into multiple VTRDs with a hierarchical trust among them³. Therefore, authentication, confidentiality and virtual trust routing domains form the main components of our new framework. The *network resource management* (NRM) and *key management* (KM) are the tools that help to create and maintain this framework.

Building multiple VTRDs are required to hide network resource information for particular subset network routers. It uses encryption/decryption key to identify a VTRD member. Hiding routing information provides three-fold benefits. First, it provides a method that uses key management to control network traffic, such as using different keys to provide confidentiality for particular allocation of network resources. We know that key management is the foundation of security services, which makes it easier to integrate quality of services (for example, classes of services) and strong security services together. Our analysis of scenarios (d)-(ii) and (f)-(ii) shows the benefits for building multiple VTRDs to reduce the effect imposed by insider replay and overload attacks. Second, hiding particular network resource from other VTRDs users can avoid exposing network information that can be used by attackers to explore network vulnerabilities; this then increases the network robustness (as we had discussed in scenarios (a), (c)-(i) and (h)). Third, hiding unnecessary routing information can reduce the routing table and link state database size and which, then, reduces the amount of network information that a router needs to know.

IV. SECURE LINK STATE NETWORK ROUTING FRAMEWORK

There are many proposals to guard against attacks marked as ✓ and ★. The goal of the new security framework is to safeguard the network routing protocol from threats marked by ★★ in Table II, and to limit the extent of damage that can be caused by the attack marked with ★★★ (the attack model and corresponding preventive cryptographic countermeasures have been discussed in Section III, and the survivability analysis will be given in Section VII).

Fig. 2 gives an overview of the framework. In this section, we first discuss briefly the design requirements

³An illustrative example is presented in Section IV-D.2.

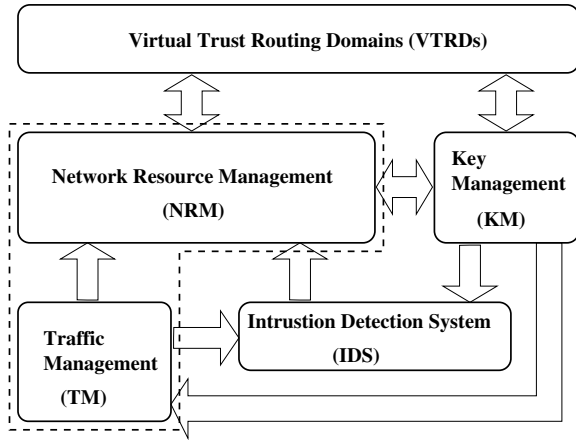


Fig. 2. Basic Framework

for the framework. We follow this up with a description of each component of the framework. Our discussion is based on the functionality of each component. Due to page limit, we leave the detail implementation for successive papers. The arrows within the figure represents the communication relations among different components. We assume they are all via secure channels.

A. Design Requirements

To our knowledge, there is no specific discussion in the current literature on what would be appropriately classified as design requirements. Thus, below, we list which we believe are major requirements:

- The maximum computational overhead on router's CPU is due to *shortest path tree* (SPT) computation (Dijkstra for OSPF) [21]. In order to avoid unduly impacting CPU processors, it is highly desirable that any new add-in security features does not exceed shortest path computation overhead.
- Current link state routing protocols invoke all types of communication (point-to-point, multicast, broadcast, flooding, virtual-link, etc.). It is desirable that a new framework be able to support the communication diversity. Via cryptographic methods, it requires a flexible group key scheme to support the versatility of communication types.
- The number of routers deployed within a link-state routing domain is growing. Moy's surveys [22][24] show more than ten-fold increase in seven years. Therefore, it is desired that the new framework is scalable.

B. Virtual Trust Routing Domain (VTRD)

The entire routing domain would be divided into multiple virtual routing domains with a hierarchical trust

among them. We refer to each such domain as a *virtual trust routing domain* (VTRD). Note that the framework does not need/implies the division of the administrative domain (of intra-domain routing). Every router that belongs to a particular VTRD will have complete routing information of its own domain, but limited information of other VTRDs, depending on the level/group it belongs to. This feature of VTRD would ensure that the damage caused by attacks marked with ★★★ are restricted within the VTRD to which the compromised router/link belongs. The working of VTRD can be described as follows:

- Trust is built among routers, which is set up by the shared keys among routers or key certificates provided by some key certificate servers. The trust validation is processed through cryptographic operations, such as, authentication and encryption/decryption. Thus, key management plays an indispensable role in determining the trust relations among routers. Each router will have a uniquely defined identification number, called *router ID* (RID) (for example, this can be the router's static IP address). The router will also have a uniquely defined cryptographic key, referred to as *key ID* (KID). Router's individual key is assigned by a centralized control center, such as *key distribution center* (KDC), in a secure way (for example, off-line).
- During the procedure of building up adjacency, router sets up the trust relation with its neighbors, which is determined by the duple $\langle \text{RID}, \text{KID} \rangle$ of the neighbors. The router should be able to verify this duple.
- A VTRD is formed by a subgroup of routers within a link state routing domain. The creation of VTRDs is the responsibility of the *network resource management* (NRM) entity. The NRM sends the commands required for setting up VTRDs. These commands are encrypted/signed using VTRD members' individual key before being sent.
- A VTRD is identified by the triple $\langle \text{VID}, \text{SKID}, \text{VKID} \rangle^4$. A VTRD session key is used by VTRD members to encrypt routing information. The VTRD session key is encrypted by the subgroup key (served as the *Key-encrypting Key*– KEK), which is shared among the subgroup members. This framework allows to build multiple

⁴VID: VTRD ID; SKID: Subgroup Key ID – identifies the subgroup key that is used as the KEK to encrypt VTRD session key and the VTRD members form the subgroup – they all share the same subgroup key; VKID: VTRD Key ID – identifies the session key used by a particular VTRD.

VTRD among same subgroup of routers.

- NRM creates multiple VTRDs based on a comprehensive analysis of overall network performance (through a set of sub optimal functions). It involves the security requirements (for example, as less overlapping among multiple VTRDs as possible) and network resource reservation mechanism (a pipe view of network resource). With the changing of time, the parameters of network security and network resource might change. The NRM needs to change the VTRDs accordingly. In this paper, we do not discuss how NRM manages the network resource.

The VTRD session key can be used by VTRD users to derive routing information. When the network is under attack, the VTRD session key needs to be updated through subgroup key (KEK). From the NRM's point of view, the relation among VTRDs can be mutual exclusive or prioritized. NRM decides and enforces this relation by issuing the commands to build different VTRDs. *Information level confidentiality (IC)* is used to reserve network resources for a particular VTRD. These information can be decrypted only by the members of that VTRD.

C. Network Resource Management (NRM)

Together with *key management (KM)*, NRM plays an important role for our framework to provide survivability. It servers as a coordinating center to create or withdraw VTRD. In this paper, we assume it is protected well from both insiders and outsiders, and NRM has secure channel connection to every router.

The *traffic management (TM)* and *intrusion detection system (IDS)* report network status and security events to NRM. Based on this information, NRM makes the decision on creating or withdrawing a particular VTRD. We note that the implementation details is out of scope of this paper, our focuses are the functionalities provided by each component.

1) *Reports from TM*: TM maintains the database of historic network traffic pattern, network topology and network resource allocation information. To simplify the communication overhead, we consider the TM is integrated with NRM and serve as a subsystem of NRM (see Fig. 2). TM also provides historical network statistics to IDS, which helps IDS to take decisions. We are not addressing the information shared between TM and IDS in this paper. Note that any new physical information changes need to be registered at TM in advance.

TM also collects the network traffic pattern information. The granule of historic network traffic pattern can be daily, weekly or monthly, which reduce the traffic fluctuation or attack on the network. The traffic pattern infor-

TABLE III
REPORTS TO/FROM TM

Information (to NRM & IDS)	
static	topology
dynamic	topology
dynamic	VTRD
static	link capacity
dynamic	occupied bandwidth
dynamic	available bandwidth
Information (from KM)	
crypto keys for each VTRD	

mation can help NRM to allocate VTRD to avoid congestion.

TM maintains two types of network topology and network resource allocation information: static and dynamic. When new routers or links are added into the network, the static network topology and corresponding link capacity information are required to be updated via offline method. The dynamic information includes the current connectivity status of network and traffic pattern information, which can be collected periodically. Due to flooding, TM also can monitor the network routing packets propagated on the network to derive network topology changes and VTRD locations. It requires KM to inform TM about the crypto keys used by each VTRD. Table III gives a summary of the reports from TM

2) *Reports from IDS*: We assume IDS can detect any type of attacks targeting link state routing. One of phenomenal work in this area is by Wang, Yifei and his work group – JiNao [32]. The proposed wrapper based link state routing IDS is integrated with network management system. Due to our proposed strong security featured link state routing (information level authentication and confidentiality), the link state routing can provide explicit evidence to the IDS, which help reduce both false positive and false negative. Our focuses in this paper is to build a new framework (VTRD) for link state routing. So we leave the detail of IDS involvement into the framework as a part of further research. We present a summary of the content IDS report in Table IV.

3) *NRM operations*: NRM informs KM to distribute/update crypto keys that used by each router. NRM also adjust VTRD deployments based on network resource allocation (reported from TM) and security alerts (from IDS).

NRM construct VTRD via flooding procedure based on the extension of link state routing protocol, for example, opaque LSA used by OSPF [7].

NRM constructing/withdrawing VTRD needs to fulfill the following prioritized requirements:

TABLE IV
REPORTS FROM IDS

Information (to NRM)
<ul style="list-style-type: none"> • identify insider/outside attacks • identify attack locations • identify subverted router • identify attacking targets • identify attacking methods • identify compromised VTRD • identify compromised network services

- i Fulfil security requirements (S): isolating a subverted router, detouring from compromised links.
- ii Fulfil traffic requirements (T): mapping service level requirements to the VTRD. The VTRD is constructed based on network resource reservation via hidden routing information. The used crypto key is a trust token that used to identify how much trust is build among a subset of network routers. Mapping the service level requirements to VTRD requires the network router to have traffic management capability.
- iii Fulfil the survivability requirements (Γ): The overlapping of multiple VTRDs should be as small as possible. When small number of routers are compromised, it will affect the smallest number of VTRDs.

NRM uses one or multiple comprehensive optimized functions \mathcal{F} to make the decision of constructing/withdrawing VTRD; such as $\mathcal{F}(S, T, \Gamma)$, where, S, T and Γ represent security requirements, traffic requirements and survivability requirements respectively. The function \mathcal{F} can be constructed based on some experimental work for particular network.

NRM constructs/withdraws VTRD based on the information from IDS and TM. It informs network routers to adjust the VTRD setting using the unique shared key possessed by pair of NRM and routers. Whenever necessary, NRM informs the KM to issue/update key set possessed by each router.

D. Key Management (KM)

An efficient key management needs an efficient keying scheme that can reduce the management overhead. In this section, we discuss the requirements of the keying scheme and present a simple example to show its usage.

Creating VTRDs in a routing domain, providing IA and confidentiality to the routing information require an efficient symmetric keying scheme. The keying scheme would be deemed suitable for this purpose if it displays the following features:

- i Due to frequent routing information exchange, the use of shared key scheme is desired in order to minimize computational overhead.
- ii It is flexible in order to support group/subgroup communication to reduce overhead caused by subgroup formation process. When NRM builds VTRDs, it only distributes the triple $\langle \text{VID}, \text{SKID}, \text{VKID} \rangle$, encrypted VTRD session key (encrypted by subgroup key), and related network resource information to each VTRD member. It is desired that a router be able to generate the corresponding subgroup key from $\langle \text{VID}, \text{SKID}, \text{VKID} \rangle$.
- iii The size of the shared key of a router needs to be moderate.

1) *Secure Group Communication Keying Scheme (SGCKS)* : Our framework needs a *secure group communication keying scheme* (SGCKS). For SGCKS, a *key distribution center* (KDC) is needed which allocates secrets to each group member (router) in advance. We call these “secrets” as *key generation seeds* (KGS). The KGS can be used to generate the group/subgroup keys. The subgroup of routers can build a VTRD and the subgroup key is used as a KEK to encrypt routing information propagated throughout the routing domain.

A brief representation of SGCKS using mathematical notations is given below:

- G : Group, where $|G| = n$.
- S : Subgroup, which is a subset of a group and $S \subseteq G$.
- x : Group member, $G = \{x^p | p = 1, 2, \dots, n; p \in N \text{ and } n \geq 2\}$.
- K : Key, generated from KGS.
- F : Key generation function, $K = F(KGS)$.
- Superscript: Identify the source of (a) communication peer(s).
- Subscript: Identify the destination of (a) communication peer(s).

Within group G , a group member x^p , $p \in \{1, 2, \dots, n\}$, can use its KGS^p to generate subgroup KGS_s^p , where the communication subgroup $S = s \cup \{p\}$, and $s \subset G$, $S \subseteq G$. Group member x^p uses the subgroup key $K_s^p = F(KGS_s^p)$ to communicate with other subgroup members of s .

There are many candidates for SGCKS, two good surveys can be found at [30] and [25]. We do not address in detail here.

2) *Implementing Components of the Framework Using SGCKS* : One of the fundamental functionality of routers is to exchange routing information. Thus, we can not prevent insiders (both “good” and “bad”) from receiving

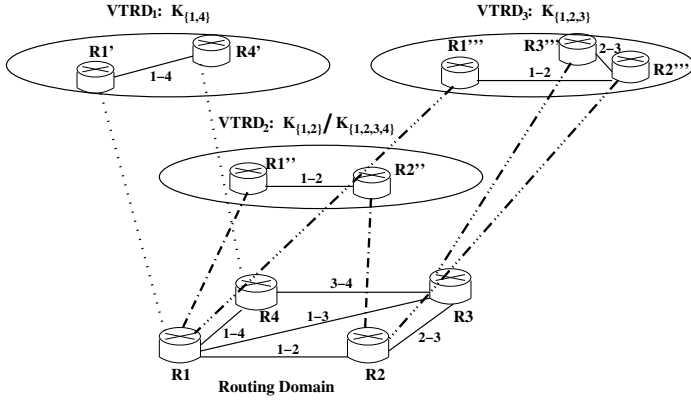


Fig. 3. An example of virtual routing

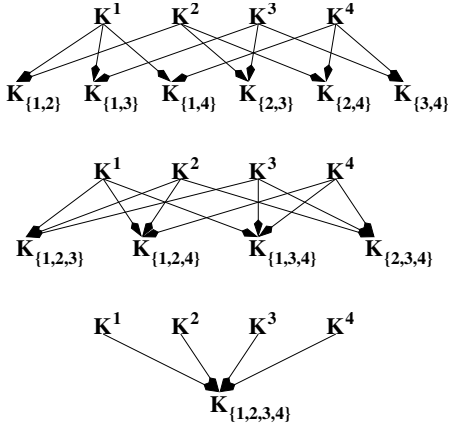


Fig. 4. An example of SGKS for virtual routing

ing/understanding the routing information. These properties determine that the network routing are vulnerable to insider attacks. When an insider attack occurs, the whole system can be compromised. To limit the attack's range, we can model the physical network as multiple virtual networks, which are represented as multiple VTRDs. Each VTRD can be configured corresponding to a certain network service or a particular group of nodes. Note that routers belonging to a VTRD share a common subgroup key.

We now present a simple illustrative example with a 4-router network shown in Fig. 3; key relations based on SGCKS are shown in Fig.4. Keys K^1 , K^2 , K^3 and K^4 are assigned by KDC to routers R1 R2, R3 and R4. Three VTRDs are constructed by the network's NRM system. To provide confidentiality for these VTRDs, routers use different subgroup keys as KEKs to encrypt routing information. Using SGCKS, subgroup keys can be derived following the direction of arrow lines. Suppose that one of the link attributes that we want to communicate is the link capacity through LSA of the link-state routing protocol, and that we want different VTRDs to use a different value of the link capacity (or rather "reduced" capacity) for its

TABLE V
MUTUAL EXCLUSIVE VTRD

VTRD	KEK	Link(capacity)
1	$K_{\{1,4\}}$	1-4(2)
2	$K_{\{1,2\}}$	1-2(2)
3	$K_{\{1,2,3\}}$	1-2(2), 2-3(2)

TABLE VI
PRIORITIZED VTRD

VTRD	KEK	Link(capacity)
1	$K_{\{1,4\}}$	1-4(3)
2	$K_{\{1,2,3,4\}}$	1-2(1), 1-3(1), 2-3(1)
3	$K_{\{1,2,3\}}$	1-2(3), 2-3(3)

own domain. For example, we set the capacity of every physical link to be 4 units. When three VTRDs are mutually exclusive and the capacity requirement of each links in VTRD is 2 units, the used subgroup keys and LSAs flooded through out the network are shown in Table V. Only the routers in the subgroup can decrypt the corresponding VTRD routing information.

When three VTRDs are prioritized, we assume that $VTRD_1$ and $VTRD_3$ are at the same priority level, and they both preempt $VTRD_2$. The required bandwidth for each link of VTRD is 3 units, then the used subgroup keys and LSAs flooded in the network are shown in Table VI. In this case, $VTRD_2$ can use group key $K_{\{1,2,3,4\}}$ to encrypt the routing information 1-2(1), 1-3(1), 2-3(1); the total capacity available for $VTRD_2$ is 2 units via to two paths (path 1-2, and path 1-3-2). Both $VTRD_1$ and $VTRD_3$ members can see that there is 1 unit (owned by $VTRD_2$) as available capacity on links 1-2, 2-3 and there are 4 units available on link 1-3. Note that the (virtual) topology of $VTRD_2$ in Fig. 3 is changed as well.

V. GENERATING VTRDS

Creating VTRDs is the job of NRM and algorithms used to create VTRDs is versatile. In this section, we propose one approach based on k -shortest path algorithms.

A. Assumptions

We propose a fixed network scenario to create VTRDs. The assumptions are listed as follows:

- Network is composed by three types of routers, they are:
 - Ingress routers: located at the edge of routing domain (local service provider side).
 - Egress routers: located at the edge of routing domain (long distance service provider side).

- Intermediate routers.
- We classify the ingress routers as sources and egress routers as sinks. And, the number of sources equal to the number of sinks.
- One VTRD is represented by a shortest path between a pair of source and sink. Multiple shortest paths may coexist and are vertex-disjoint.
- Every shortest path can fulfil network traffic requirement.
- The sources and sinks are not fixed. This means that, as long as there is a path between a pair of source and sinks, the network service can be fulfilled. For a VTRD, the path assignment is independent.

B. Paths Generating Algorithms

Bhandari summarized many k -shortest path algorithms in [4]. Our path generating algorithms are based on several algorithms proposed in that book. $k(> 2)$ vertex-disjoint paths algorithm and revised version of multiple sources and destinations disjoint paths algorithm is presented in Appendix II.

C. Relation among VTRDs

The proposed approach to build VTRDs ensure the independence among all VTRDs that provide strong survivability when a router is compromised. Any single router failure of a VTRD would not affect other VTRDs. But, this approach has some limitations. First, the number of VTRDs is constrained by the number of shortest paths that can be found among sources and sinks. Second, it is hard to fulfill the VTRD creating demands, in the scenario, that VTRD is formed between any pair of routers. Third, it does not provide the protection for a VTRD when one of its router is compromised.

To overcome the presented limitations, in Section VIII-B, we also formalized a more complete network model (based on graph theory) and list several possible problems for further research.

VI. COMPUTATIONAL IMPACT

In this section, we present the computational impact of our proposed approach. Recall that our proposal requires information level authentication and confidentiality as the preventive countermeasures for threats ★ and ★★, respectively. This means that the router needs to store the cryptographic keys, sign the LSAs that it originates and also encrypt that information before it is sent. Moreover, the number of LSAs that a router originates for each link is now a multiple of the number of VTRDs that shares

that link. Therefore, an important issue is the implementation¹² of this framework and in what way would it impact router CPU. In this section, we address this issue by doing a worst case analysis for both CPU time complexity and communication overhead. For this analysis, we use OSPF as the link state routing protocol.

A. Evaluation Assumptions

According to Moy [21], the router CPU usage is dominated by the length of time it takes to run the *shortest path first* (SPF) calculation in OSPF. The complexity of this process grows quadratically with the number of routers in the routing domain. Routers are designed to carry out this task efficiently.

For simplicity of the analysis, we make the following assumptions:

- The network is designed optimally in accordance with the guidelines provided by Aho and Lee [2]. Therefore, the worst case complexity of the number of LSAs contained in a single link state update packet is $O(N^{4/3})$ for a network with N nodes.
- Moy in his analysis [21] indicated that the average size of the LSA to be 64 Bytes back in 1991. We assume the size to have doubled for our analysis.
- The OSPF standardization report [24] states that the mode of the number of routers in a single domain in a real OSPF deployment is 350. All our analysis is done for 400 nodes.
- LSA arrival and processing usually occurs at different frequency/time than the shortest path calculation.
- The job being processed is never preempted.
- We have computed the benchmark for the speeds of encryption/decryption and hashing schemes, using crypto++ [10], on a Windows 2000 Professional system that has a x86 (family 6) processor with a clock speed of 930 MHz. We have also computed the Dijkstra's shortest path with varying number of nodes on this machine. We assume that one would observe similar (not same) trend if these computations are done on any other system.
- Each VTRD shares all the links in the network. This would be the worst case for number of LSAs generated for each link.

B. CPU usage for confidentiality

We assume that the time required to find the level of the key used for encryption, by looking up the LSA header, is accounted for in the fixed processing time. We also assume that the processing time for LS Update packet is going to be dominant. This assumption is made based on

TYPES OF LOGICAL NETWORK FAILURE

Type	Threat actions (Attacks)	Label	Remarks
I	Outsider wiretapping	(a)	★★
	Insider undeliberate exposure	(h)-(i)	★★
	Insider deliberate exposure	(h)-(ii)	★★★
II	Outsider replay	(d)-(i)	★★
	Outsider interference	(c)-(i)	★★★
	Insider replay	(d)-(ii)	★★★
	Insider Overload	(f)-(ii)	★★★
	Insider Over/Under/Mis claim	*	*

*: An insider attacker misuses the network resource under its control.

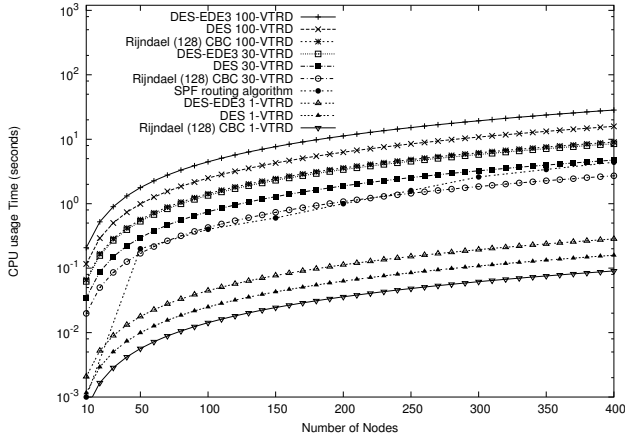


Fig. 5. CPU usage by various algorithms for Encrypting/Decrypting LSAs

the fact that there are additional functions to be carried out with each LS Update packet. The extra complexity introduced is due to a) decryption of LSAs while processing the LS Update packet, and b) encryption of LSAs retrieved from the LS database before bundling them into an LS Update packet and flooding. In order to see how the time needed to encrypt/decrypt all the LSAs in the LS Update packet, we have used the benchmark for speeds of various encryption/decryption algorithms that we have computed.

Fig. 5 is the plot to evaluate the performance of various algorithms for encryption/decryption. We have also plotted the SPF computation in the figure. We observe that the best performance (as far as CPU usage time is concerned) is given by Rijndael (AES) [1], while 3-DES [15] is the most time consuming algorithm for all the cases. We also observe that the CPU usage time increases with the increase in number of VTRDs. Moreover, with the AES encryption/decryption scheme, a router in a domain with 30 VTRDs generates the same overhead as that of the SPF. Our intention here is to show the performance of algorithms for encryption/decryption and not to suggest which among these should be used. Rather, our analysis here is to show that addition of such functions do not necessarily impose undue burden on routers.

C. Communication Overhead

The communication overhead is the size of the information that has to be carried along with the packet in order to support various schemes. In case of confidentiality, the inputs of cryptographic algorithm may be padded and the outputs may be compressed; this depends on cryptographic algorithm used. The variance may be neglected.

VII. EVALUATION OF SURVIVABLE NETWORK FRAMEWORK

In this section, we analyze the control plane failures and then set up the evaluation model to analyze the robustness of presented VTRD framework.

Conceptually, network failure include physical network devices failure and logical network failure. Physical network devices failure include nodes failure and links failure. The corresponding damages to the network is straightforward. The device is either functional or dysfunctional. In this paper, we consider the physical network devices failure as a consequence of an attack. The possible failure size/range of physical network is one of parameter to evaluate our presented framework.

A. Logical network failure

We define the logical network failure as an attacker hacking into the system and utilizing the system resource to deploy network attacks. It is also called *Byzantine* failure, which can cause more damage than just simple link or node failures. The attacker can utilize network control plane, i.e. routing, to deploy more efficient attacks to cause wide area network turbulence or intercept critical data traffic. Moreover, it is hard to locate logical network failure as compared to physical network failure, because attackers always try to hide their location and make the network suffering longer.

In this paper, all our analysis is based on the attack model presented in Table II, Section III. We focus on attacks marked by ★★ and ★★★ except the attack *false denial of receipt* (label (g)-(ii)). In addition, we add the analysis of overclaim, underclaim and misclaim, in which an insider attacker misuse the network resource under its control.

Based on attack consequences, we classify the network failure caused by attacks into two types (shown in Table VII):

Type-I Information based failure: the attacks target at deriving network resource allocation information.

Type-II Operation based failure: the attacks target at compromising or misleading network operation.

When an attacker hacks into a network router, we call the router as a subverted router. As a result, we assume an attacker takes over the router and usurp all the knowledge the subverted router has. To analyze *Type-I* failure, we use \mathcal{P} to represent the overall information of a link state routing domain, \mathcal{S}_{r_i} represents the information known by a router r_i . Thus, we have equation:

$$\mathcal{L}_{r_i} = \mathcal{S}_{r_i} / \mathcal{P} \quad (1)$$

where, \mathcal{L}_{r_i} represents the proportional information a router r_i has, where there are n routers within the link state routing domain, $i = 1, \dots, n$ and $\mathcal{P} = (\mathcal{S}_{r_1} \cup \dots \cup \mathcal{S}_{r_n})$. We define the information survivability as Γ , which represents the proportion of safe information, i.e.

$$\Gamma = \sum_i^{good} \mathcal{L}_{r_i} \quad (2)$$

where r_i is a good router. If we consider that each router has equal probability to become a subverted router, obviously, $\mathcal{L}_{r_1} = \dots = \mathcal{L}_{r_n}$ is the condition to minimize the variance $V_k(\Gamma)$ of k subverted routers, where $k < n$. Accordingly, the expectation $E_{n-k}(\Gamma)$ represents the survivability of a link state routing domain with k subverted routers. The condition $\mathcal{L}_{r_1} = \dots = \mathcal{L}_{r_n}$ specifies that the survivability is router independent. In real network, some router may get more security preference than others, such as minimum network cut routers, edge routers, etc. In this paper, to make the analysis easier, we assume all routers have same security preference.

The survivability analysis based on *Type-II* failure is similar to the analysis of *Type-I* failure. Based on our proposed VTRD framework, the consequence of failure is the incidence of overall network, i.e. the proportional number routers that recognize the VTRD session key. This is based on the fact that, within a VTRD, a subverted router can compromise or mislead other routers that use the same VTRD session key. Thus, for *Type-II* failure, we still can use the survivability Γ defined in the analysis of *Type-I* failure.

B. Survivability Analysis

To find the maximum survivability Γ , we need to define \mathcal{P} and \mathcal{L}_{r_i} to fulfil Equations 1 and 2. The VTRD framework proposed in Section IV is based on subsets of

routers within the link state routing domain, and within¹⁴ VTRD all routers use a shared VTRD session key to disseminate routing information. Thus, we define \mathcal{P} as the set of all possible VTRDs of a link state routing domain. We note that a VTRD is composed by a subset of routers, which is connected by communication links. To form a virtual routing domain, a VTRD needs to provide end-to-end guarantee to support data traffic. This means that there should be at least an ingress and an egress router. Thus, the minimal composition of a VTRD is the routers on a routing path within the link state routing domain. In our analysis the granularity of a VTRD is a routing path within the link state routing domain. We then use \mathcal{P} to represent all possible routing paths within the link state routing domain. The \mathcal{S}_{r_i} is one possible routing path that contains router r_i . Our optimization goal is to maximize Γ , in another words, to minimize the effect of a node failure. It is equivalent to making \mathcal{P} as big as possible as well as to make \mathcal{S}_{r_i} as small as possible. One way to achieve this is to distribute VTRD session keys evenly and to avoid depending on some nodes heavily.

Once router r_i is compromised, it will affect all VTRDs that contains r_i . All other VTRDs that do not contain r_i will not be affected. Thus, the survivability defined in Equation 2 is also referred as usable rate under network stress.

VIII. CONCLUSION

In this paper, we present the motivation for requiring security features in routing in order to build a framework for survivable network. This framework emphasizes the use of efficient cryptographic countermeasures for network survivability against security threats to link state routing protocol. This framework relies on providing information level authentication & information level confidentiality that can be imbedded in link state routing protocol with assistance of a key management system that uses secure group communication.

A. Pros and Cons

While we have already illustrated several benefits of our approach, we provide below a summary on pros and cons of our approach:

Pros:

- It prevents attacks marked by ★ and ★★, and limits the damage caused by ★★★.
- Proposed security features are add on components, which do not change operational functionalities of current link state routing protocols.

- Most threats targeted at link state routing can be prevented by using preventive cryptographic countermeasures. And for some “hard problems”, they can create useful network routing status information for detection and recovery systems.
- Our computational assessment shows that the routers would have the ability to handle the extra processing required for the new framework, with some increase in memory requirement.

Cons:

- Link bandwidth management is needed to help prevent certain threats, which is typically not support in current best effort network.
- Multiple VTRDs will increase the number of LSA within a LSU, which will increase the traffic load due to routing updates.

B. Further Research

Since all routers of a VTRD use the same session key to encrypt/decrypt routing information belong to that VTRD, we transform the path finding problem to VTRD session key distribution problem. It is shown as follows:

INSTANCE: Given a network, represented by a graph $G(V, E)$, where V is the set of vertices and E is the set of edges, $|V| = n$.

QUESTION:

- (1) If there exists a distribution of $n(n - 1)/2$ distinct source-sink paths (vertices can overlap) that minimize the maximum number of paths that went through a vertex, $\min_S \max_{i \in V} \{S_{r_i}\}$, here S_{r_i} is the path function that represents number of path that router r_i involved into.
- (2) If there exists a distribution of $n(n - 1)$ distinct source-sink paths (vertices can overlap) that minimize the maximum number of paths that went through a vertex. This requires each pair to have two vertex-disjoint paths. These two paths can be backup for each other and the VTRD session keys for each path is different. We call each path a sub-VTRD of a particular VTRD.
- (3) Network resource allocation issues to support VTRD routing framework.
- (4) Due to hiding routing information, each router within link state routing domain might have different view of network. It is yet to be determined whether this might cause any instability or convergence problem of link state routing protocol.

REFERENCES

[1] AES Algorithm (Rijndael) Information, <http://csrc.nist.gov/CryptoToolkit/aes/rijndael/>.

[2] A. V. Aho and D. Lee. “Hierchical Networks and the LSA N-Squared Problem in OSPF Routing”, *IEEE GlobeCom2000*.

[3] D. Beard, S. Murphy and Y. Yang, “Generic Threats to Routing Protocols”, work in progress. Available from <http://www.ietf.org/internet-drafts/draft-beard-rpsec-routing-threats-01.txt>

[4] R. Bhandari, “Survivable Networks - Algorithms of Diverse Routing”, *KLUWER ACADEMIC PUBLISHERS*, 1999.

[5] A. Chakrabarti, G. Manimaran, “Internet Infrastructure Security: A Taxonomy”, *IEEE Network*, November/December 2002, Vol. 16 No. 6.

[6] S. Cheung, “An efficient message authentication scheme for link state routing ”, 13th Annual Computer Security Applications Conference (ACSAC ’97).

[7] R. Coltun, “The OSPF Opaque LSA Option”, RFC 2370, July 1998.

[8] R. Coltun, D. Ferguson and J. Moy “OSPF for IPv6”, RFC 2740, Dec 1999.

[9] D. Coppersmith, M. Jakobsson, “Almost Optimal Hash Sequence Traversal”. *Finacial Cryptography*, 2002.

[10] Crypto++ Version 5.0. Available from <http://sourceforge.net/projects/cryptopp/>.

[11] M. Gupta, N. Melam, “Authentication/Confidentiality for OSPFv3”, Internet draft, November, 2002. <http://www.ietf.org/internet-drafts/draft-ietf-ospf-ospfv3-auth-00.txt>.

[12] R. Hauser, T. Przygenda, and G. Tsudik, “Lowering Security Overhead in Link State Routing”, *Computer Networks (Amsterdam, Netherlands: 1999)*.

[13] D. Huang, A. Sinha, D. Medhi, A Double Authentication Scheme To Detect Impersonation Attack In Link State Routing Protocols, Accepted for publication and presentation at IEEE International Conference on Communications (ICC 2003), Anchorage, Alaska, USA on May 11-15, 2003.

[14] K. J. Houle, G. M. Weaver, N. Long and R. Thomas “Trends in Denial of Service Attack Technology”, *CERT® Coordination Center*, 2001. http://www.cert.org/archive/pdf/DoS_trends.pdf.

[15] R. Housley, “Triple-DES and RC2 Key Wrapping”, RFC 3217, December 2001.

[16] IETF Routing Protocol Security Requirements (rpsec) working group <http://www.ietf.org/html.charters/rpsec-charter.html>.

[17] “Intermediate System to Intermediate System Intra-Domain Routing Exchange Protocol for use in Conjunction with the Protocol for Providing the Connectionless-mode Network Service (ISO 8473)”, ISO DP 10589, February 1990.

[18] S. Kent, R. Atkinson, “Security Architecture for the Internet Protocol”, RFC 2401, November 1998.

[19] H. Krawczyk, M. Bellare, R. Canetti “HMAC: Keyed-Hashing for Message Authentication”, RFC2104, February 1997.

[20] M. Jakobsson, “Fractal Hash Sequence Representation and Traversal”. *IEEE International Symposium on Information Theory*, 2002.

[21] J. Moy, “OSPF Protocol Analysis”, RFC 1245, 1991.

[22] J. Moy, “Experience with the OSPF Protocol, RFC 1246, 1991.

[23] J. Moy, “OSPF version 2”, RFC 2328, April 1998.

[24] J. Moy, “OSPF Standardization Report”, RFC 2329, April 1998.

[25] M. J. Moyer, J. R. Rao and P. Rohatgi, “A Survey of Security Issues in Multicast Communications”. *IEEE Network*, November/December, 1999.

[26] S. Murphy, M. Badger and W. Wellington, “OSPF with Digital Signatures”, RFC 2154, June 1997.

[27] P. Papadimitratos, Z. J. Haas, “Securing the Internet Routing Infrastructure,” *IEEE Communication*, October 2002.

- [28] R. Perlman, "Network Layer Protocols With Byzantine Robustness," *MIT/LCS/TR-429*, October, 1988.
- [29] R. Shirey "Internet Security Glossary", RFC 2828, May 2000.
- [30] D. R. Stinson, "On Some Methods for Unconditionally Secure Key Distribution and Broadcast Encryption". *Designs, Codes and Cryptography*, 12, pp. 215-243, 1997.
- [31] B. Vetter, F. Wang and S. Wu, "an Experimental Study of Insider Attacks for OSPF Routing Protocol", *IEEE International Conference on Network Protocols*, pp. 293 - 300, October 1997.
- [32] F. Wang "Vulnerability Analysis, Intrusion Prevention and Detection for Link State Routing Protocols", PhD thesis, 2000.

APPENDIX I ATTACK MODEL

A. Link state routing model

The *open shortest path first* (OSPF) [23] is the most popular link state routing protocol. Based on it, we formulate a simple link state routing model to identify the security issues we will address in this paper.

Link state routing model is composed of physical entities (routers and communication links) and logical entity (link state routing protocol running in the routers). Within link state routing domain, each router originates the link state information for the link that has the direct connection with the router (the link state information is directional) and floods⁵ this information to its neighbors. A receiving router will forward the routing information (unmodified) via flooding again. Therefore each router will have the same view of the network. When a router joins the network, it needs to synchronize the link state database with its neighbors. The granularity of routing information in link state routing protocol is the link state of a router's interface. This information is called the *link state advertisement* (LSA). During flooding, multiple LSAs can be encapsulated in a single *link state updates* (LSU) routing packet.

The security issues related to the link state routing model can be broadly classified as security for network device, operational security and communication security. The security for network device concerns the physical access to the routers and communication links. The operational security includes the access control of operating system of a router, privilege mode of a router, etc. The communication security is related to the transmission, reception and processing of routing data (LSAs and LSUs). Note that all data security related issues we discussed in this paper is based on ROUTING DATA but not user data and we focus on the communication security aspect of the link-state routing protocol.

⁵Flooding provides reliable data transmission, in which a router forwards the routing packets to every interface except the one it receives the routing packets.

B. Threats to Link State Network Routing

In order to categorize the security threats to the routing protocol, we first need to identify the possible threat sources and their actions. We will follow definitions (such as threats, insider/outsider, etc.) provided in RFC2828 [29] and Beard et al. [3] for this purpose, and use them in the context of network routing and routing protocols.

Threat Sources: The threat sources for link state routing can be through communication links and routers. Although, there could be some malicious network operators/administrators involved, we consider that the final threat actions are originated from network components, such as communication links and routers. We classify them into two groups: insiders and outsiders. The legitimate devices that lie inside the link state routing security perimeter are called insiders. The devices that lie outside the link state routing security perimeter are called outsiders. The security perimeter defines a router/link's authorized role in network routing, which includes two parts: identity and functionality. For example, a valid router (or an insider) is authorized to perform routing functions, such as exchanging routing information, and associates with an unique router ID. But sometimes, it is easy to be confused in term of masquerading with the definition of insider and outsider. Note that an outsider can masquerade to generate routing information as an insider. It has no valid identifier and is not authorized to perform routing functions. An insider can also masquerade as another authorized router and generate forged routing information. It has valid identifier, but it is not authorized to impersonate other routers or forge other routers' routing information. To make it clear, our definition of "authorized" is from overall routing operational functionalities per se. For example, a router/link is authorized as part of routing domain to exchange routing information and possesses a valid identifier. In this sense, we always say an outsider is unauthorized and an insider is authorized.

Threat Actions: Threat actions are also called attacks. Here, all our discussions focus on the origination, verification and transmission of routing data. The attacks can be active or passive⁶. These attacks can be classified based on threat sources – insiders and outsiders. This classification helps to categorize corresponding preventive cryptographic countermeasures which will be discussed in Section III-A. Note that we discuss the attack model that an attacker compromises other routers' re-

⁶Threat actions are identical to attacks. Defined in RFC 2828: An "active attack" attempts to alter system resources or affects their operation. A "passive attack" attempts to learn or makes use of information from the system but does not affect system resources.

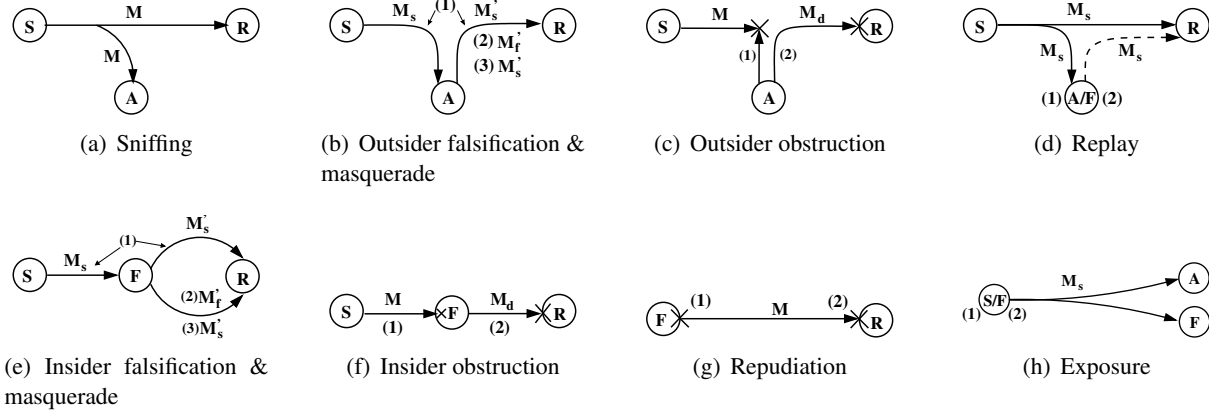


Fig. 6. Outsider attacks (a) to (c) and (d)-(i), Insider attacks (d)-(ii) and (e) to (h), (S:sender R:receiver (or victim) A: outsider (attacker) F: insider (attacker) M : any routing message M_s : routing message from sender M_s' : forged routing message of sender M_f' : malicious routing message generated by subverted routers M_d : dummy routing traffic that cause overload $-->$: delayed transmission)

source. Thus, we exclude the discussion of insider attackers overclaim/underclaim/misclaim the network resource that under its control. For example, a subverted router claim the bandwidth attach to one of its interface is w , but in fact, the actual bandwidth is r , where $w \neq r$.

1) *Attacks by outsiders*: It is initiated by an unauthorized router/link.

- (a) Sniffing (passive): Monitoring and recording routing data transmitted on the communication links among routers; see Fig. 6(a).
- (b) Falsification and masquerading (active): This can be of three kinds: (i) Substitution: altering or replacing valid routing information with false routing information; see (1) in Fig. 6(b), (ii) Insertion: introducing false routing data that serves to deceive an authorized router; see (2) in Fig. 6(b), (iii) Masquerading: impersonating an authorized link/router; see (3) in Fig. 6(b). The masquerading is usually executed concurrently with substitution and/or insertion.
- (c) Obstruction (active): This attack can be of two types: (i) Interference: an attacker can block the transmission link by cutting off the transmission link or introduce noise into the transmission link to prevent the victims from receiving the routing information correctly; see (1) in Fig. 6(c), (ii) Overload: an attacker can place excess dummy routing traffic that can saturate the victim's input buffer or exhaust victim's CPU capacity; see (2) in Fig. 6(c).
- (d)-(i) Replay (attack): a valid routing data transmission is maliciously or fraudulently repeated by an outsider; see (1) in Fig. 6(d).

2) *Attacks by Insiders*: This form of threat actions is initiated by a subverted router/link.

- (d)-(ii) Replay (active): a valid routing data transmission is maliciously or fraudulently repeated by an insider; see (2) in Fig. 6(d).
- (e) Falsification and masquerading (active): (i) Substitution: altering or replacing valid routing data with false routing data; see (1) in Fig. 6(e), (ii) Insertion: introducing malicious routing data to overclaim/underclaim the network resource possessed by the subverted router or misclaim network resources possessed by other authorized routers; see (2) in Fig. 6(e), (iii) Masquerading: impersonating other authorized routers; see (3) in Fig. 6(e). This is the same as specified in outsiders' threat actions; the masquerading is usually executed concurrently with substitution and/or insertion.
- (f) Obstruction (active): (i) Stop forwarding: the subverted router does not forward received routing packets; see (1) in Fig. 6(f), (ii) Overload: excessive routing information processing burden is placed on the router in order to saturate the victim's input buffer or exhaust victim's CPU capacity; see (2) in Fig. 6(f).
- (g) Repudiation (active): (i) False denial of origin: a subverted router denies the operations that it had done on the transmitted routing information; see (1) in Fig. 6(g), (ii) False denial of receipt: a subverted router denies receiving the routing data; see (2) in Fig. 6(g). Although, an outsider can repudiate what it has done, it is more critical for insider attacks. A subverted router can

cause more serious problem when they are authorized to perform routing functions. Quickly identifying the subverted routers/links will help to reduce the recovery time imposed by the attacks.

- (h) Exposure (active): (i) Indeliberate exposure: a router unintentionally releases sensitive routing data to attackers (both insiders and outsiders); see (1) in Fig. 6(h), (ii) Deliberate exposure: a subverted router intentionally releases sensitive routing data to attackers (both insiders and outsiders); see (2) in Fig. 6(h).

We note that in our classification all attacks originated from outsiders occur on the routing transmission links, e.g. Fig 6(a) to 6(d). In them, attacks (c)-(i) and (d)-(ii) need to get access to transmission link first and then attackers can launch attacks. Attacks originated from insiders are generated by the subverted routers, e.g., Fig 6(d) to 6(h). When an outsider successfully takes over an authorized router, it becomes an insider (or subverted router).

APPENDIX II

k -SHORTEST PATH ALGORITHM

A. BFS algorithm

- Step1. Start with $d(A)=0$, $d(i)=\text{INF} \forall i \in V(i \neq A)$. Assign $P(i)=A \forall i \in V(i \neq A)$. Let Υ^T denote the set of vertices from which search or scanning (fanning out) takes place in a given iteration, and let Υ^I denote the set of vertices whose labels are updated in that iteration, Υ_j denotes the set of neighbor vertices for vertex j . Start with $\Upsilon^T = \{A\}$.
- Step 2. Set $\Upsilon^I = \Phi$
 $\forall j \in \Upsilon^T$,
do the following:
 $\forall i \in \Upsilon_j$ if $d(j)+l(ji) < d(i)$ and $d(j)+l(ji) < d(Z)$,
set $(d(i)=d(j)+l(ji))$, $P(i)=j$, and
 $\Upsilon^I = \Upsilon^I \cup \{i\}$;
set $\Upsilon^T = \Upsilon^T - (\Upsilon^T \cap \{Z\})$.
- Step 3. If $\Upsilon^T = \Phi$, END;
otherwise go to 2.

B. $k(> 2)$ Vertex-disjoint Paths Algorithms

In general, k Vertex-disjoint paths are obtained from a knowledge of $k - 1$ vertex-disjoint paths by applying the shortest path algorithm in a modified graph. The modified graph is obtained by replacing the edges of the $k - 1$ vertex-disjoint paths with arcs directed towards the source vertex, and making these arcs negative; in addition the vertices (except the endpoint vertices) of the $k - 1$

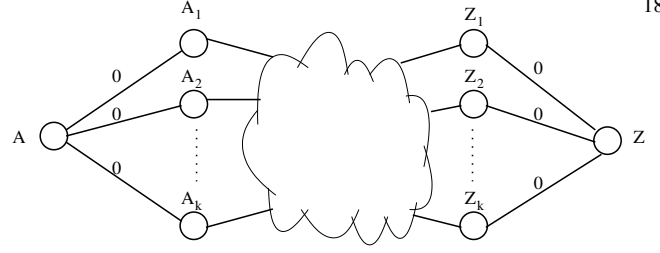


Fig. 7. k paths of the shortest pair; the cloud represents all possible one-one pair connection among A_i and Z_i ($i \leq k$)

disjoint paths are split in accordance with Vertex-Splitting method.

Vertex-Splitting method

1. For the given pair of vertices under consideration, find the shortest path using BFS algorithm.
2. Replace each edge on the shortest path by an arc directed towards the destination vertex.
3. Split each vertex on the shortest path into two collocated subvertices joined by an arc of length zero. Direct this arc towards the destination vertex. Replace each external edge connected to a vertex on the shortest path by its two component arcs (of length equal to the edge length); let one arc terminate on one subvertex, and the other arc emanate from the other subvertex such that along with the zero-length arc, a cycle results.
4. Reverse the direction of the arcs on the shortest path. Also make their lengths negative.
5. Run the BFS algorithm from the source vertex to the destination vertex in the above modified graph.
6. Remove the zero length arcs; coalesce the subvertices into their parent vertices. Replace the single arcs of the shortest path with their original edges (of positive length). Remove interlacing edges of the two paths found above to obtain the shortest pair of paths.

C. Multiple Sources and Sinks Disjoint Paths

We want to find k vertex-disjoint path among k sources A_1 to A_k and k sinks Z_1 to Z_k . We introduce two fictitious vertices A and Z into the graph, shown in Fig. 7. It does not require the path is fixed, in that A_i to Z_i , $i = \{1, \dots, k\}$. Applying the $k(> 2)$ vertex-disjoint paths algorithm between vertices A and Z . When computation is finished. remove fictitious vertices A and Z . We then derive the multiple paths for sources and sinks. Note that the results may not fulfil number of k path requirement. For different graphs, we can derive different solution.

APPENDIX III
ATTACK TREE EXAMPLE

We present the attack tree (see Table VIII) based on our discussion in Table II. Note that this attack tree may not present all possible attacks that available, but shows the importance of presented cryptographic countermeasures to prevent the attacks. We shows that preventing attackers from knowing the network topology through confidentiality can stop attackers from doing further attacks.

TABLE VIII
NETWORK ROUTING DENIAL OF SERVICE ATTACK PATTERN

<p>Network routing denial of service attack pattern</p> <p>Goal: Denial of service attack on network routing</p> <p>Precondition: Attackers can derive routers/links physical location through known network topology</p>	
<p>OR</p>	<p>1. Outsider obstruction - Interference (c)-(i)</p> <p>OR</p> <ol style="list-style-type: none"> 1. Use network tools (such as traceroute, etc.) to derive network topology 2. Wiretapping to derive network topology (a) 3. Attacker derives network topology through insider undeliberate exposure (h)-(i) 4. Attacker derives network topology through insider deliberate exposure (h)-(ii) <p>AND</p> <ol style="list-style-type: none"> 1. Attacker guesses the possible physical location of communication links 2. Attacker cuts the communication links or injects noise <p>2. Outsider obstruction - Overload (c)-(ii)</p> <p>OR</p> <ol style="list-style-type: none"> 1. Use network tools (such as traceroute, etc.) to derive network topology 2. Wiretapping to derive network topology (a) 3. Attacker derives network topology through insider undeliberate exposure (h)-(i) 4. Attacker derives network topology through insider deliberate exposure (h)-(ii) <p>AND</p> <ol style="list-style-type: none"> 1. Attacker guesses the possible physical location of communication links 2. Attacker injects dummy routing traffic to block routers <p>3. Insider obstruction - Overload (f)-(ii)</p> <p>OR</p> <ol style="list-style-type: none"> 1. Attacker derives network topology from its own routing table 2. Use network tools (such as traceroute, etc.) to derive network topology 3. Wiretapping to derive network topology (a) 4. Attacker derives network topology through insider undeliberate exposure (h)-(i) 5. Attacker derives network topology through insider deliberate exposure (h)-(ii) <p>OR</p> <ol style="list-style-type: none"> 1. Attacker injects dummy routing traffic to block routers. 2. Attacker cuts the communication links to partition the network. 3. Subverted routers partition the network and stops forwarding the traffic.
<p>Postcondition: Attackers stop routers functioning or isolate routers</p>	