主动网络中实现逐站安全的原型框架

李毅超* 张君雁 傅 彦 杨国纬

(电子科技大学计算机科学与工程学院 成都 610051)

【摘要】针对Abone实验平台上实现大规模主动网络的安全问题，集成多种非主动安全组件和传统安全功能，提出了多级分布式密钥法和创建安全关联的策略，扩展KLIPS和警报指示，建立了逐站的包认证和完整性验证的原型框架。通过实验，对LKH方法和EK方法进行分析比较，证实了LKH适合多播组成员的撤回。

关 键 词 主动网络; 逐站; 安全; 撤回

中图分类号 TP309 文献标识码 A

Implementing Hop-by-Hop Security Prototype Frameworks in Active Networks

Li Yichao Zhang Junyan Fu Yan Yang Guowei

(School of Computer Science and Engineering, UEST of China Chengdu 610054)

Abstract This paper focuses on implementing security of large scale Active Networks on the Abone platform. We set up hop-by-hop security prototype framework with packet authentication and integrity by integrating many non-active components and traditional secure functions. We put forward hierarchy-distributed key and security association policy, and extend KLIPS and alarm indications. Finally, we compare LKH with EK using experiments, and conclude that LKH is suitable for the multicast members' revocation.

Key words active networks; hop-by-hop; security; revocation

实现大规模主动网络存在许多与寻址相关的安全问题，采取的方法是将安全集成在一个主动节点内，并定义安置在节点操作系统或执行环境中的机制来保障系统内部安全。在实验平台Abone(Active Network Backbone)上配置主动网络时，把主动网络的安全服务分为逐站和端到端两类。逐站保护为无外来者向其包流中插入包的节点提供安全保障，确保相邻节点间任何信号的安全。端到端保护将源实体和包绑定，访问节点的服务基于和该源规则相关的特权被保护。

本文为逐站安全提供包认证和完整性验证的框架，满足Abone对逐站安全的需求，且更具通用性。其中使用一个集中式密钥服务器CKS(Centralized Keying Server)动态建立节点集合中的安全关联，并对节点发送和接收的包实施逐站的包认证和完整性验证。CKS维护一个包含安全链路和组的安全拓扑。构成该安全拓扑的组件应在CKS上注册，并以加密载荷的形式处理所有相关的密钥信息。任何对安全拓扑的改变都必须立即使用更新的加密消息通知被影响的节点。

使用IPsec构建安全关联(Security Association，SA)[1]，其重放保护、消息认证和完整性验证等特性都直接集成在框架中。CKS和节点之间的密钥交换用Internet Key Exchange (IKE)协议实现[2]。原型采用多级分布式密钥法，集成了对多播IPsec安全关联的支持[3]，便于撤回多播组成员。

该原型框架要求每站都要保证包的完整性和真实性，并在主动网络中转化为保障任意恶意外部节点不

2003年1月21日收稿

* 男 33岁 硕士 讲师 主要从事计算机网络、信息安全方面的研究

向其他节点的包流中引入恶意包，而且假定CKS被分别检测，并从恶意攻击中恢复。

1 原型框架设计和功能集成

1.1 原型框架设计

逐站安全原型框架如图1所示，其中CKS构成原型框架的中央实体。安全拓扑以链路和组的形式定义，前者对应于单向安全关联，后者对应于不同的多播组。当有服务器出现时，服务器能静态或动态使用配置命令将该拓扑配置成CKS。对于一个给定的CKS，可信任集合RS定义为：

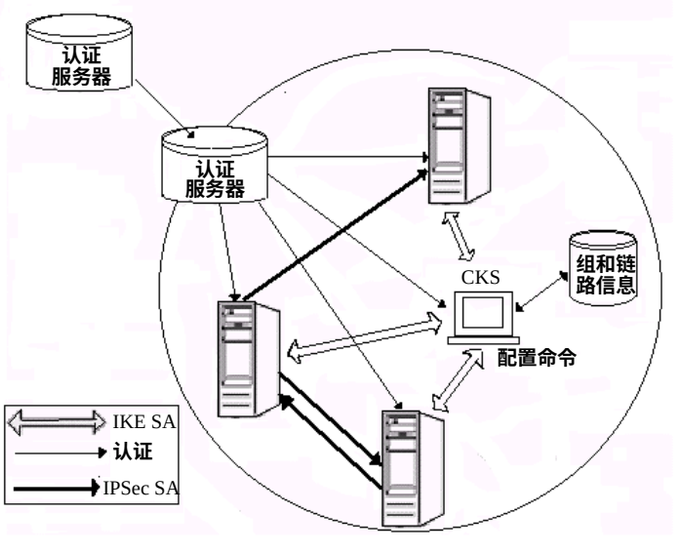

图1　逐站安全原型框架

RS={N | N为某个安全链路一个端点的节点}，或RS={N | N为安全拓扑中一个多播组成员节点}。

密钥管理模块(Key Management Module，KMM)运行在构成安全拓扑的每个节点上。KMM通过扩展IKE后台程序实现，可以将IPsec服务无缝地集成在该框架中。

为建立安全关联，节点需在CKS注册。如果一个节点在多个服务器上注册，将引发冲突问题，这时就由节点通知相应的服务器，同时采用适当策略决定哪个节点最先注册。

为使安全拓扑能动态改变，必须提供一个配置命令集合，包括初始化节点注册、增加/删除链路和组、刷新安装在给定节点上密钥集合。

在节点注册过程中，如果相关的CKS配置改变，则CKS向该节点发送与链路和组有关的密钥信息，再利用该链路和组信息建立节点内部安全关联。上述操作所需的认证服务由认证服务器(Authentication Server，AS)提供，AS作为一个DNSSEC服务器实现[4]。不同节点的公开密钥作为一个DNSSEC CERT 资源记录(Resource Record，RR)存储，每个节点运行一个能确认RR的DNSSEC Resolver。该原型使用BIND 9.1分配DNSSEC服务[5]，使用lwres分配Resolver能力。

1.2 集成IKE

本文使用FreeS/WAN的Pluto作为原型框架的IKE模块[7]。Pluto在Linux网络节点上作为一个后台程序运行，能实现与其他Pluto和IKE之间的互操作。

1.2.1 节点注册

节点在相应CKS上注册后，才能成为安全拓扑的一部分。注册时应先在节点和CKS之间建立IKE安全关联，CKS发送节点所需的各种信息来描述和对等实体之间的安全关联。由CKS发送的信息以加密IKE有效载荷形式传送，因此受到保护。

CKS发送节点所需的属性来建立可靠的安全关联，包括SA方向性，以及该SA提供的IPsec转换类型(AH或ESP)和其生命周期。CKS为SA定义外部值SPI，用于在接收端识别特定的SA，因此该值在接收方唯一确定。CKS为每个节点创建一个不同的SPI，用来维护可信任集合。CKS必须确保SPI创建过程不产生重复值，以保证没有两个主机使用同一SPI。

Pluto保持一张和SA相应的连接表。每个连接项记录创建一个SA所必需的信息，包括两个SA端点、SA生命周期、对物理接口的引用和实际连接信息。每个Pluto连接保持SA的有关策略，策略是标识值的集合，其值指示创建何种SA，例如用POLICY_AUTHENTICATE认证，POLICY_ENCRYPT加密，POLICY_TUNNEL进行隧道模式等。本文还定义一种新的策略POLICY_KEYSERVER，用来指示IKE为一个特定连接创建SA。一旦给定的SA集合被安装在注册节点上，CKS规定的附加操作就可执行：

1) REFRESH　SA集合后作为一个新集合处理，任何早期注册或更新过程的旧SA被删除；

2) ACK_REQD　节点需回送已接收并安装SA的确认信息，CKS使用返回的时间戳来决定SA在节点的

安装顺序。为辨别ACK与状态号的对应关系，节点还返回其ACK服务器的时间戳。

当节点有了构建一个SA所需信息，就在其安全关联库(Security Association Database，SADB)中增加一项记录[1]。

1.2.2 SA的删除

为删除一个SA，CKS仅需发送足够的信息来唯一识别SA即可。当一个节点收到来自CKS的针对某个特定SA的DELETE消息时，就从其SADB中删除对应目录项，并识别和删除所有与之相关的Pluto状态和连接信息。

1.3 警报指示类型

如果未能获得预期效果，受影响的节点就回送错误指示，服务器将这些指示作为日志或反馈来纠正出现的问题。下面定义警报指示的类型：

1) SA_EXPIRED SA的生命周期有限，两个节点间的SA在当前一个过期之前需要重新加密，以保证链路安全。若在最近一个密钥过期前无新密钥产生，节点会继续使用已有SA，并将生命周期更新为缺省值，同时周期地向CKS回送SA_EXPIRED警报指示；

2) MALFORMED_PACKET 指示节点不能理解服务器发送的包，CKS接收到该警报，通常是发生了来自某个服务器的欺骗或重放(不成功)，或表示CKS和节点之间同步的丢失，通过在CKS上重新注册，节点能再次同步；

3) SERVER_CONFLICT 配置错误，如两个CKS分别定义同一节点作为相同多播组的一部分，能在节点处检测到，如果这种情况发生，节点就向两个服务器发送SERVER_CONFLICT警报；

4) SIGN_FAILED 用在多播组安全的上下文中。

1.4 信息包

原型框架使用ISAKMP的Notification有效载荷实现CKS与节点加密信息的交换，在加密服务器框架内使用的Notification消息类型和私有值如下：

1) KEYEXCHANGE_ACK = 32,768；

2) KEYEXCHANGE_REGISTER = 32,769；

3) KEYEXCHANGE_DELETE = 32,770；

4) KEYEXCHANGE_ALARM = 32,771。

1.5 建立IPsec安全关联

FreeS/WAN KerneL IPsec Support (KLIPS)为节点内的安全关联提供了建立和维护机制。原型使用KLIPS作为单播SA。安全多播是完全不同的范例，因为一个包有多个不同的接收方，且多个发送方都能向同一多播地址发送。在一个多播组内，整个SPI空间被所有多播节点共享，只有通过复杂的协商过程，才能保证SPI值在所有目的地都唯一，一种较好的解决方法是用多播服务器(即CKS)为每个多播组定义和分配SPI。为支持多播，需要对KLIPS进行一些修改：

1) 能识别和接受多播分组，当该节点上注册的应用都不能读出多播分组时，多播分组才被抛弃；

2) 能处理具有大于24字节头部的IP包；

3) 在SADB内不能同时为多播地址维持一个入站SA和一个出站SA，因为入站SA和出站SA的目标地址和SPI相同。由于预先不知道发送者，使用源地址不能消除SA的二义性。因此，仅将多播地址的出站SA添加到SADB，输入包也使用同一出站SA对发送给该多播地址的包解密；

4) 为使任何接收方都能为一个多播分组找到唯一SA，每个节点对发送包和接收包都使用一个公共组SPI，以这种方式定义的多播SA总是双向。

1.6 集成分级密钥框架

为防止成员在发送给组的数据已经被撤回后仍对其访问，需要对已有的组成员进行重新加密。在最简单的情况下，组控制器或者CKS能产生一个新密钥并单独将其发送给所有节点。但当节点数很大时，这种方法就不能很好地进行调节，本文用原型框架结合LKH方法来解决这一问题[6]。

分级树结构中所有成员都位于叶节点，因此将分级树构建成B+树，采用以节点IP地址为B+树的索引，

LKH不需要树平衡，所以该树不需要更优化的表示。

CKS为一组成员定义了一个多播地址，向每个成员发送密钥更新信息。控制信道是每个节点预先知道的，或作为CKS发送的第一个注册消息。一个特定CKS的可信任集合需要和控制地址绑定，以便能接收CKS发送来的密钥更新消息。CKS发送的每个消息都使用其私有密钥进行签名，而不包含有效签名的更新消息被忽略，同时向CKS回送一个SIGN_FAILED警报，从而防止欺骗。

1.7 集成DNSSEC

在BIND 9.1中提供的lwresd程序是轻型DNS解法的后台程序，使用liblwres库为客户提供名字查找服务。输入的轻型解法请求由lwresd解码，再使用DNS协议分解。当DNS查询完成时，lwresd采用轻型解法对来自命名服务器的解答进行编码，并返回给最初请求的客户。

为获得和主机名、类和类型相关的资源记录，Liblwres库提供了lwres_getrrsetbyname( )API。成功调用该函数后，就完成了资源记录表和潜在的包含签名信息的记录表的填写。Lwres后台程序自动检查签名的有效性，并通过设置/清除RRSET_VALIDATE标识来指示成功/失败。使用lwres_getrrsetbyname( ) API，CKS及其可信任集合的节点能查询公开密钥，并通过检测RRSET_VALIDATED标识得知是否可信，将公开密钥用在IKE消息交换过程中或在多播密钥更新信道内来验证消息。

2 时间评估

使用单密钥保护多播信道中的数据，当一个成员被撤回时，多播密钥就被改变并通知所有其他成员。通过IKE SA方法，CKS和可信任集合中的每个节点共享一个密文，因此新密钥可以使用这些共享密钥进行保护而不需要任何私有密钥签名，其时间评估的步骤如下：

1) 定义CKS的一个多播组包含许多不同测试节点；
2) 每个节点都在CKS上注册；
3) 每个节点按注册编码递增顺序从多播组中撤回；
4) 在使用EK法时执行同样的操作；
5) 步骤1)~ 4)根据不同的LKH树级数重复，其中的级数表示优先B+树级数。

表1和图2对EK法和LKH法在加密时间和多播组成员撤回时间上的差别进行了比较。为了研究LKH的行为，必须研究B+树的创建方法。对应级数为1、2和3的LKH树如图3所示。LKH树的时间行为直接依赖于在撤回过程中执行签名操作的私有密钥数量，对于order=1的LKH树，当测试节点10被撤回时，所需的签名操作数量为3：一个用于兄弟节点(5，6)，一个用于兄弟节点(7，9)，另一个用于相邻分支。测试节点9的撤回也有相同的签名操作数量：一个用于测试节点7，一个用于兄弟节点(5，6)，最后一个用于相邻分支。结果显示，简单地增加LKH树的级数并非改善密钥更新时间的途径。更新时间的增加很平均，因为更大的级数并没有更多的传送需求。为了传播密钥的撤回，需要更宽的LKH树向每个树中当前位置的兄弟节点发送更新引起的延迟费用非常大。从图2可以看出，由于基于签名的私有密钥操作代价非常昂贵，LKH比EK在时间上不止低了一个数量级。

表1　EK和LKH的加密时间比较　单位ms

测试节点	Order=1		Order=2		Order=3	
	EK	LKH	EK	LKH	EK	LKH
10	4.0	226	4.0	350	4.0	441
7	3.2	218	3.2	205	3.2	335
4	1.5	161	1.5	85	1.5	200
2	0.5	86	0.5	96	0.5	90
1	…	…	…	…	…	…

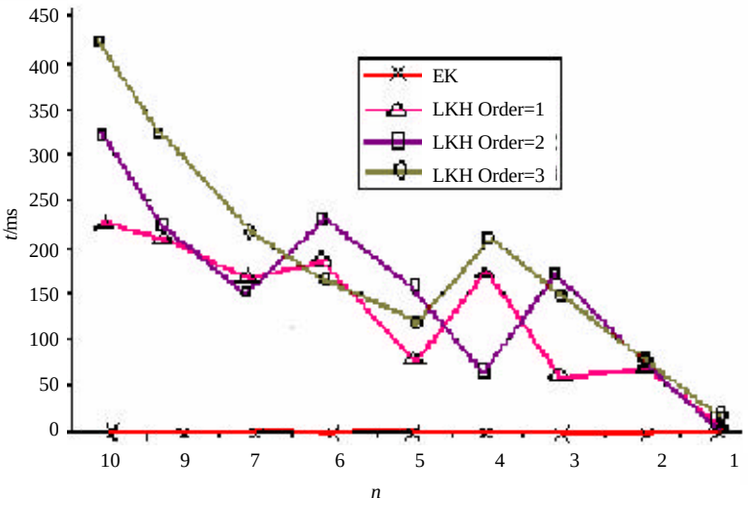

图2 EK和LKH撤回多播成员时间的比较

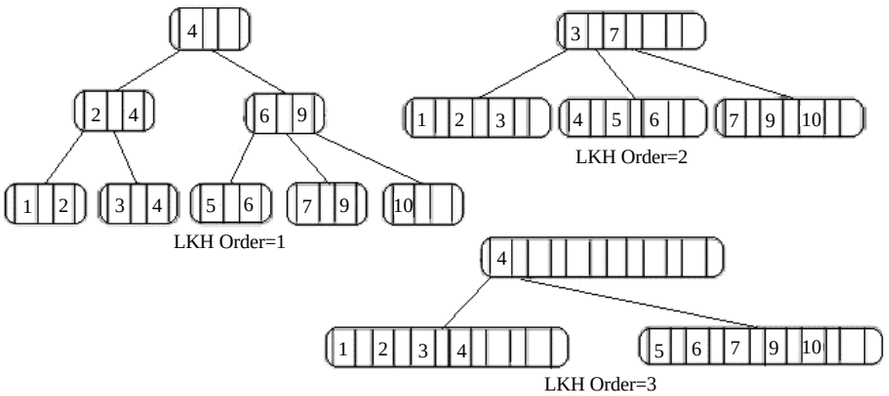

图3 对应级数为1，2和3的LKH树

综上所述，在节点数较少时，LKH算法的优点被该实现中所需要的基于签名操作的私有密钥的代价抵消，而每个对等实体EK不能很好地进行调整，当节点数增加时，EK和LKH相差不多。

3 结 论

本文提出了在主动网络中使用CKS动态建立逐站安全的原型框架。其中，使用DNSSEC集成了对资源认证的支持，使用IPsec SA集成了对对等认证和正确性验证的支持，还实现了多播安全关联，集成了一个使用IKE的加密构件。通过实验，对EK法和LKH法进行了比较，证明后者更便于实现多播组成员的撤回。

参 考 文 献

[1] Kent S, Atkinson R. Security Architecture for the Internet Protocol[S]. IETF Network Working Group, RFC 2401

[2] Harkins D, Carrel D. The Internet Key Exchange (IKE)[S]. IETF Network Working Group, RFC 2409

[3] Canetti R, Cheng P C, Giraud F, *et al.* An IPsec--based host architecture for secure internet multicast[C]. Proceedings of NDSS ’2000, 2000, 54-70

[4] Murphy S, Lewis E, Watson R. Strong security for active networks[C]. IEEE Openarch, 2001, 21-28

[5] Peterson L, Gottlieb Y, Hibler M, *et al.* An OS interface for active routers[J]. IEEE JSAC, 2001, 19(3), 77-85

[6] Wallner D, Harderet E, Agee R. Key Management for Multicast: Issues and Architectures[S]. IETF Network Working Group, RFC 2627

编 辑 徐培红