计算机科学2002Vol.29№.3

# 主动安全技术研究*)

The Research of Proactive Security

滕 猛 邹 鹏 王怀民

(国防科技大学计算机学院网络技术与信息安全所 长沙410073)

**Abstract** Proactive security provides a method for maintaining the overall security of a system, even when individual components are repeatedly broken into and controlled by an attacker. Proactive security is highly desirable in many realistic settings, in particular to ensure acceptable level of security using available operating systems with weakly security. We describe the proactive approach and review some methods, implementations, and applications. We also report a practical implemention on the key proactive security mechanisms.

**Keywords** Proactive security, Threshold cryptographic, Share

## 一、引言

为保证数据的安全,密码学提供了一系列安全手段。然而,针对密码学安全机制的最常见的攻击不是通过分析密码来破译密文,而是进行系统攻击。一旦获得系统控制权,密码学中的密钥对攻击者就变成了公开信息。在现实世界里,系统攻击容易进行,是攻击者大量使用的攻击手段。

在计算机网络中引入一种入侵检测机制,能有效地侦测入侵行为,提醒管理员及时发现入侵并恢复对主机的控制权。这样还能及时发现安全漏洞,增强安全性。然而,攻击者通常都是尽可能地避免被侦测到,哪怕是为此而放弃对主机的控制。入侵检测技术本身对数据的安全性并没有改善,而且如果攻击者在被侦测到之前就已经成功地控制了主机,仍然可能将秘密窃取走。

周期性地刷新密钥是一种增强安全的方法,如口令刷新、安全通信协议 IP-SEC 和 SSL/TSL 中的会话密钥刷新,其思想是通过刷新使旧的密钥对攻击者无用。如果攻击者窃取到一个密钥后,为避免被侦测到,放弃对主机的攻击,则刷新之后的密钥对攻击者来说,又是未知的,因而攻击者为了窃取秘密,要冒更大的被侦测到的风险而一直进行攻击活动以获得有效密钥。

但刷新密钥的方法还是不能避免在一段时间内泄漏密钥引起的安全问题。另一种增强安全的方法是将密码学上的信任(trust)发布到几个服务器上,其中最有代表性的就是秘密共享算法,并由此发展出一个新的密码学分支——门限密码学。在门限密码学中,秘密被拆分成多个影子,每个影子分给不同的服务器。这些服务器一起运行,来模拟原来在集中方式下单个服务器的行为。这种协议可以保证在攻击者攻破 t 个服务器(预先规定的一个数字)之前,系统是安全的。门限密码学在许多情况下都增强了系统的安全性。但它也有局限:只要有充足的时间,攻击者一定能够逐一攻破服务器,因而最终能窃取秘密。这种长期攻击对有些系统的安全性威胁很大,如 CA 系统、秘密恢复很困难的安全通信系统。

## 二、主动安全的提出

主动安全是一种可以对付这种长期攻击的保护技术[1,2],它将以上的入侵检测、秘密刷新和秘密共享技术有机地结合起来,到达保护的目的。主动=分布+刷新+检测。即,首先在几个服务器之间分布密码学的功能,然后使服务器周期性地执行刷新协议。这个协议使得服务器共享一个新的影子而不改变秘密数据的值。更重要的是,攻击者在刷新之前收集到的信息对今后的攻击毫无用处。检测使得服务器能够发现入侵(不一定是全部入侵)并清除它,然后利用刷新协议可以恢复服务器的原来功能。在今后将会看到,入侵者有时也能阻止服务器完全自动地恢复系统安全性,但这样入侵者也无疑会被发现。此时,极具价值的信息可以用手工方法恢复出来。总而言之,只要在一个周期内被攻击者控制的服务器不超过一个定值,系统的安全性就有了保证。这时,即使系统中的每一个服务器(在不同周期中)都被入侵过,系统也是安全的。除主动安全中刷新协议本身有部分检测功能外,系统中的检测功能是由普通的检测手段来实现,因而本文不过多地讨论检测技术。

主动安全有着广泛的应用空间。一般来说,主动安全可以应用于任何需在相对较长时间内维护安全的地方。包括 CA、电子货币。也可在电子投票系统中用于解密数据以及用可认证的方式安全地存储敏感数据。主动安全技术削弱了应用系统安全性对操作系统安全性的依赖,适于在安全性较低的商用计算机操作系统中建立安全性较高的应用。

## 三、主动安全技术

主动安全包括三类在多数系统中可以使用的技术:主动秘密共享、主动签名和主动安全通信。

### 1. 主动秘密共享

秘密共享用于保护在多个服务器上分布共享的秘密。典型的门限密码方案是将 n 个服务器共享秘密,其中任意 t+1 个可以恢复秘密。

如,在 Shamir 共享协议中,在整数集[0…p-1]中共享一个秘密 s 的方案如下(p 为素数):分发者 Dealer 产生 t 个[0…p-1]中随机的数 $a_1,\cdots,a_t$。给定多项式 $f(X)=s+a_1X+\cdots+a_tX^t$,Dealer 给服务器 i 一个影子 $s_i=f(i) \bmod p$。显然,从任意 t 个服务器的影子中并不能得到秘密的任何信息,而(≥)t+1

*)本文研究得到863-306-ZD-02资助。滕 猛 博士生,主要研究领域为分布式系统安全。邹 鹏 教授,博士生导师,主要研究方向为操作系统、分布计算。王怀民 教授,博士生导师,主要研究领域为分布对象技术、网络安全。

个服务器的影子就可以通过插值重构秘密值。

为维护秘密共享方案的安全性，可以周期性地刷新影子，这样即使存在攻击者能够最终入侵所有的主机，只要在一个时间周期内入侵的主机数目少于一个上限，系统也可以是安全的。刷新协议应该保证新的影子与旧的影子除都共享同一个秘密外，没有其它相关性。

周期性地刷新秘密可以如下执行。每个服务器 i 选择一个随机的 t 阶多项式 $f_i(X)$ 使得 $f_i(0)=0$。服务器 i 将值 $s_{ij}=f_i(j) \bmod p$ 传送给服务器 j。服务器 j 的新的影子为 $\hat{s}=s_j+s_{1j}+\cdots+s_{nj} \bmod p$。然后清除旧的影子 $s_j$。可以看出，新的影子位于一个新的多项式 $\delta(X)=f(X)+f_1(X)+\cdots+f_n(X)$ 之上，且其常数项仍为 s。

以上过程只能在攻击者进行被动攻击的情况下正常工作。如果有主动攻击存在，则必须使用可验证的秘密共享(VSS)协议，其中 Pedersen 共享被证明是安全的[3]。这样构成的一个完整的秘密共享解决方案就是主动共享方案[2]。

文[1]引入了主动安全的概念，并给出了信息理论上安全的秘密共享模型。文[2]具体化了主动安全的概念，并给出了一个密码学上有效且鲁棒的方案。

**2. 主动签名**

公钥系统的安全性严重地依赖于私钥的完整性和私有性，因而主动签名系统必须在提供系统可用性的同时，保护私钥。

最自然的方法是使用主动秘密共享来共享该私钥。这种方法可以长期保证私钥的安全性。然而，为了签名，私钥必须在某个地方重构，这样就丧失了分布的优点：攻击者只要攻击该点就可获取私钥。因而，主动签名必须使得服务器能够联合产生有效的签名，同时阻止攻击者伪造出有效的签名。

主动签名[4~7]包括三个阶段：密钥产生阶段、联合签名产生阶段以及主动密钥刷新阶段。密钥产生由一个可信的 dealer 首先完成。签名由所有服务器利用它们共享秘密的影子来分布产生。并且，主动签名还必须保证尽管因为主动化，共享秘密的影子会不断地变化，它们产生的关于同一个信息 m 的签名必须是相同的。在两个连续的刷新阶段期间，该方案可以容忍一定数目(如，一半)的服务器被攻击者控制。目前已有多种签名方案被主动化。文[4]给出了一种 RSA 签名的主动方案；文[5]对文[4]进行了简化；文[6]从另一个角度给出了一种与文[4]完全不同的主动 RSA 签名方案；文[7]给出了基于离散对数签名方案(如 DSS)主动化的一个通用方法。

**3. 主动安全通信**

主动秘密共享和主动签名实现的一个前提是在各服务器之间维护一个已认证且秘密的通信。各服务器必须维护其相应的公共密钥(如会话密钥)、签名密钥、私有解密密钥以及其它服务器的公开密钥的完整性和秘密性。

解决这个问题的一个直接方法是：在每个刷新阶段，每一个服务器选择一新的公私钥对，将公钥签名后分发给其它服务器，并接收从其它服务器传来的签名的公钥。然后，用新的公钥来协商产生新的会话密钥。但是，如果攻击者同时也控制了通信链路，则可以进行中间人攻击。更严重的是，若攻击者入侵了两台服务器，则入侵者可以选择伪造的公钥，将自己永久性地插入到这两个服务器之间。这样，即使入侵者已经被清除掉，这两个服务器仍然永远失去了互相认证的能力。

文[8]解决了以上问题。服务器持有主动签名方案的影子。相关的验证公钥在所有服务器的只读存贮器(ROM)中都有一份。然后，每个刷新阶段，各服务器都联合对各个服务器选择的公钥进行签名。这个签名对应的证书用于各方验证其它服务器选择的公钥。

## 四、主动安全的实现

文[9]提出并实现了一个主动安全系统的原型。它是与操作系统捆绑在一起的，即将主动安全的一些功能放在操作系统中。这对一般性的用户来说，部署这样的服务器群，代价太大。为此，本文提出一种在操作系统之上构建主动安全系统的方案。

在设计应用级的主动安全系统时，要遵循以下原则：

1. 尽可能地将一个主动系统部署在同一个局域网网段之内，可以增加攻击者攻击的难度。主动安全系统不直接接受用户的请求，而是通过一个应用服务器转接。安装一个小型的防火墙，对主动安全系统的服务器群进行保护。各服务器均安装一个入侵检测工具，对可能的入侵进行检测。服务器群的选型要多样化(选择不同类型的计算机)，以增加攻击者的攻击难度。服务器群的管理员要选择多人担当，以防止内部安全事件的发生。

2. 将一些公共信息(如 IP 地址、端口号以及常量)放在只读存贮器中，防止入侵者进行修改。对主动安全系统本身的程序进行签名。每次重启主动服务时，首先验证程序本身的正确性。

3. 建议采用分布对象技术实施。分布对象技术有利于快速开发这种分布的系统，有利于系统的构件化，有利于功能的扩充，有利于并发性的实现。由于主动安全系统要不断地进行刷新，所以还有时间同步的考虑，分布对象技术中的 CORBA 中间件定义并提供时间服务。可选的中间件包括：(国产的) StarBus、Visibroker、Orbix、Obacus、MICO 等。

主动安全系统中时间被化分成一个个周期，1，2，…m，…。每个周期由刷新阶段和功能计算阶段组成。(1)在刷新阶段，先执行影子恢复协议(有必要的话)，然后再执行一个刷新协议，刷新阶段结束时，相对于上一个周期而言，每个服务器得到一个关于秘密 x 的新影子。(2)在功能计算阶段，服务器使用它们的影子进行相关计算(如主动签名)。直接使用文[4，5，6，7]中的主动签名算法均可。以下给出一个初始化以及刷新协议的算法，供参考。

**初始化方案**

1. Dealer 为每一个服务器 i 生成只读存贮器，其内容包括，{Scert (IPs，PORTs，C)，IPs，PORTs，C，Vcert，Scert ($P_i$)}，其中，IPs 为各服务器的 IP，PORTs 为各服务器使用的端口号，C 为各服务器使用的常量，Vcert 是与 Scert 对应的公钥，$P_i$ 为服务器 i 的主动安全程序。

2. 将 Scert 分布共享于服务器群之中，其中服务器 i 的影子为 S'cert。其它长效秘密也同样地分布共享。

3. Dealer 彻底销毁拥有的 Scert。

4. 各服务器调用刷新协议 refresh(0)。

为保证主动安全，本算法引入了一对签名密钥(Scert，Vcert)。如果要实现的主动方案就包括主动签名算法，则不需要引入这对密钥，直接使用主动签名算法要使用的密钥代替它来签名。Scert($P_i$)为服务器 i 的关于主动安全程序本身的签名，用于清除攻击后的恢复。这个算法在系统初始化时执行，其目的是让可靠的分发者Dealer将秘密分发到服务器群

(下转第128页)

的容错策略结合起来，根据实时任务本身的属性、系统资源现状和外部环境，动态地为任务选择当前状态下最佳的容错策略，在保证系统可靠性要求的前提下，灵活地配置系统资源，提高资源的利用率和系统的任务吞吐量。自适应容错作为实时操作系统的一个组成部分，为应用提供通用的容错支持，减轻了应用程序编程的工作量。自适应容错的实现必须高效(开销小)、通用(可以应付不同类型的错误)、确定(及时响应外部环境的变化)，否则将得不偿失。

在本文中，我们建立了基于 RTEMS 的 AFTM 模型，并对其中的每个部分的构成、基本功能和实现方法进行了详细的描述。我们希望通过对该模型的研究和实现，深入地研究自适应容错的原理及其相关实现技术，推动其在工程实际中的应用。

## 参 考 文 献


1 Furth B, Halang W A. A Survey of Real-Time Computing Systems. International Journal of Mini and Microcomputers, 1994, 16 (3)
2 Cristian H. Understanding fault-tolerant distributed systems. Communications of the ACM, 1991, 34(2): 56～78
3 Jahanian F. State restoration in real-time fault-tolerant system, Complex System Engineering Synthesis and Assessment Technology Workshop, July 1992. 21～29
4 Thuel S R, Strosnider J K. Enhancing Fault Tolerant of Real-Time Systems through Time Redundancy, G. M. Koob and C. G. Lau, ed., Kluwer, 1994. 265～318
5 Lyu M R. Software Fault Tolerance, Wiley, 1995
6 陈宇，熊光泽. 基于非精确计算模型的容错单调比率调度. 计算机科学，已录
7 Hecht M, Hecht H, Shokri E. Adaptive Fault Tolerance for Spscecraft, IEEE 2000
8 Kim K, Lawrence T. Adaptive Fault Tolerance: Issues and Approaches. In: Proc. IEEE Workshop on Future Trends of Distributed Computing Systems, 1990. 38～46
9 Kim K. Action-level fault tolerance, in Advances in Real-Time Systems, S. H. Sang ed., Prentice Hall, 1994. 415～434
10 Bondavalli A, Stankovic J, Strigini L. Adaptive Fault Tolerance for Real-Time Systems, in Responsive Computer Systems: Steps toward Fault-Tolerant Real-Time System, D. S. Fussell and M. Malek ed., Kluwer, 1995. 187～208
11 Li Gong, Goldberg J. Implementing Adaptive Fault-Tolerant Services for Hybrid Faults: [Technique Report]. SRI International, 1994
12 Landis S, Maffeis S. Building Reliable Distributed Systems With CORBA, in Theory and Practice of Object Systems, Wiley, 1997
13 Sabnis C, Cukier M, et al. Proteus: A Flexible Infrastructure to Implement Adaptive Fault-Tolerance in AquA. In: Proc. of the 7th IFIP IWC in DCCA, 1999. 137～156
14 Shokri E, Crane P. Architecture of ROAFTS/Solaris: A Solaris-Based Middleware for Real-time OO Adaptive Fault Tolerance Support. In: Proc. COMPSAC98, 1998. 90～98
15 Shokri E, Hecht H, Crane P, et al. An Approach for Adaptive Fault Tolerance in OO Open Distributed Systems. International Journal of Software Engineering and Knowledge Engineering, 1998, 8(3)
16 Kalbarczyk Z, Iyer R K, Bagchi S, et al. Chameleon: a Software infrastructure for Adaptive Fault Tolerance, IEEE Transactions on Parallel and Distributed Systems, 1999, 10(6): 560～579
17 陈宇，罗蕾，陈红蓉，蔡建平. 一个外军嵌入式实时操作系统-RTEMS. 见：第十届全国抗恶劣环境计算机学术年会论文集. 2000. 67～72
18 陈宇，熊光泽. 支持应用软件容错的实时运行库技术


(上接第111页)

中共享。本算法同时也产生用于刷新协议和恢复协议执行时所必须得验证信息，并将其放入各服务器的 ROM 中。

**周期 t 的刷新协议 refresh(t)**

在服务器 i 上执行：

1. 产生签名密钥对($S_i(t)$, $V_i(t)$)和加密密钥对($E_i(t)$, $D_i(t)$)，产生签名 $S_i(t)(E_i(t))$。广播 $K_i=(V_i(t), S_i(t)(E_i(t)))$。若有 $S_i(t-1)$，则产生签名 $S_i(t-1)(K_i)$。

2. 服务器联合产生一个签名 Scert(KeyTable)，其中，$KeyTable=[K_1, \cdots, K_n]$。

3. 验证 Scert(KeyTable)的有效性。

4. 为需要恢复影子的服务器运行重构协议。

5. 对包括 Scert 在内的长效秘密运行刷新协议。

本算法的目的是刷新所有的长效秘密，包括 Scert。其中步骤1中，如果收到了同一个服务器发送的两个或两个以上的经过加密的不同信息，则丢弃，一个也不要；如果收到一个加密的信息，一个明文信息，则接受加密的信息；如果收到两个明文信息，则丢弃，一个也不要。步骤2中，要恢复的服务器不参与签名的产生。步骤4中的重构协议可以选择文[7]中的重构协议。步骤5的刷新协议可以选择文[2]中的算法。

**清除攻击后的周期性恢复协议**

设对服务器 i 进行恢复

1. 从 ROM 中读出有关信息，包括 Vcert，并验证正在执行的主动安全程序 $P_i$ 是否未被修改，即程序是否与 Scert($P_i$)吻合。若一切无误，则等待进行刷新协议的恢复。否则，恢复失败。

本算法用于侦测到成功的攻击后的恢复。此时，假定攻击者已被清除掉，而且主动安全程序从磁盘上重新调入运行。

**结束语** 主动安全技术是一种新型的安全技术，必将在电子商务和开放网络中起到关键性的作用。本文给出的主动安全技术的实现算法对架构主动安全系统至关重要，它拉近了主动安全技术与应用的距离。主动安全的功能计算与具体算法相关，因此有必要进一步将给出更多的针对具体密码学算法的主动实现。

## 参 考 文 献


1 Ostrovsky R, Yung M. How to withstand mobile virus attacks. In: Proc. of the 10th ACM Symposium on the Principles of Distributed Computing, 1991, 51～61
2 Herzberg A, et al. PROACTIVE SECRET SHARING Or: How to Cope With Perpetual Leakage? Advances in Cryptology——Crypto 95 Proceedings, Lecture Notes in Computer Science, 1995, 963: 339～352
3 Gennaro R, et al. Secure Distributed Key Generation for Discrete-Log Based Cryptosystems. In Eurocrypt '99, pages 295～310
4 Frankel Y, et al. Proactive RSA. In Advances in Cryptology——Crypto'97, 1997, 440～454
5 Rabin T. A Simplified Approach to Threshold and Proactive RSA. Crypto'98, LNCS, Springer-Verlag, 1998, 1462: 89～104
6 Frankel Y, MacKenzie P, Yung M. Adaptively-Secure Optimal-Resilience Proactive RSA. ASIACRYPT'99. 180～194
7 Herzberg A, et al. Proactive Public Key and Signature Systems. In: 1997 ACM Conf. on Computers and Communication Security, 1997
8 Canetti R, Halevi S, Herzberg A. Maintaining Authenticated Communication in the Presence of Break-ins. Journal of Cryptology, 2000, 13: 61～105
9 Barak B, et al. The Proactive Security Toolkit and Applications. In: ACM Conf. on Computer and Communications Security 1999. 18～27