第33卷 第11期 计 算 机 工 程 2007年6月
Vol.33 No.11 Computer Engineering June 2007
·安全技术· 文章编号：1000—3428(2007)11—0147—04 文献标识码：A 中图分类号：TP309

基于入侵容忍的CA认证中心设计

郭 萍
(南京信息工程大学计算机系，南京 210044)

摘 要：从PKI的核心部件CA入手，将入侵容忍的概念引入CA中，给出了一个可行的基于入侵容忍技术的CA认证中心设计方案。论述了基于入侵容忍CA认证中心的体系结构、各组件间的相互作用、基于入侵容忍的CA签名方案及整个系统的工作过程。针对系统的不足之处，指出了未来工作中需要改进的地方。

关键词：PKI；CA；入侵容忍；数字签名

CA Design Based on Intrusion Tolerance

GUO Ping
(Dept. of Computer, Nanjing University of Information Science & Technology, Nanjing 210044)

【Abstract】 Beginning with the kernel of PKI, which is certificate authority, this paper brings the concept of intrusion tolerance to CA, and gives a feasible scheme of CA based on intrusion tolerance. The most important part is that it discusses the system architect, a digital signature of CA based on intrusion tolerance, and the working process of the whole system. Aiming at some shortcomings, it points out where to be improved in the future.

【Key words】 PKI; CA; Intrusion tolerance; Digital signature

公共密钥基础设施[1](public key infrastructure, PKI)是提供公钥加密和数字签名服务的系统，目的是为了管理密钥和证书，保证网上数字信息传输的机密性、真实性、完整性和不可否认性。PKI可通过一个基于认证的框架处理所有的数据加密和数字签名工作，为所有网络应用透明地提供有效的安全保障。构架PKI体系[2]的核心技术就是建立功能完善的、安全的认证中心CA(Certificate Authority)，CA是当前网络安全领域研究的热点之一，其实现具有重大的实用价值和社会价值。不管多么努力，足够复杂的计算机系统总会有一些漏洞。假设攻击者利用了漏洞并控制了系统是比较合理的。在有这样攻击的情况下保证保密性并保持可以接受的服务性能是入侵容忍技术的目标。

1 入侵容忍的概念

入侵容忍[3]就是当一个网络系统遭受入侵，传统安全技术都失效或者不能完全排除入侵所造成的影响时，他就可以作为系统的最后一道防线，即使系统的某些组件遭受攻击者的破坏，整个系统仍能继续为用户提供全部或降级的服务。为了击溃系统，攻击者需要在短时间内攻击多个组件或频繁地攻击某一组件。而这两种攻击都比单个孤立的攻击更有可能被检测到。一个入侵容忍系统能够在面对攻击的情况下仍然连续地为预期的用户提供及时的服务。入侵容忍系统能够限制一些用攻击避免和预防手段无法检测的信息攻击。这些攻击可能渗透过外层防御，即用攻击避免和预防手段设置的防御，如防火墙系统、认证和加密系统等。系统将采取一些必要的措施保证关键应用的功能连续正确。这些措施包括从限制被怀疑的代码和数据到重新配置硬件和软件资源等。

一般而言，入侵容忍系统技术[4]包括两个方面：

(1)容忍技术(Tolerance Technologies)，这是目前商用系统所缺乏的功能。容忍技术可以让系统对入侵和攻击具有可复原性能(弹性)。这些技术包括资源重新分配、系统冗余等。容忍技术包括错误容忍和入侵容忍两个方面，其中错误容忍主要集中在对硬件/软件错误的容忍，而入侵容忍则集中在对恶意攻击的容忍。入侵容忍是在错误容忍的基础上发展起来的，主要是对静态属性和调整研究的基础上增加了对动态属性和调整的研究。通常将这两种容忍技术合并在一起进行系统设计，并按照危害的后果而不是原因进行分类(入侵检测系统是按照危害的原因而不是后果进行分类的)，此时将该系统称为入侵容忍系统。

(2)容忍机制的触发器(Tolerance Triggers)：入侵检测系统可以成为一个这样的触发器。但即使是目前最顶级的入侵检测系统(IDS)[5]也具有太高的误警率和太低的入侵识别范围。理论上，触发器应该具有很高的覆盖范围和零误警率。很高的覆盖范围是指对由于任何攻击和入侵导致的错误都能检测出来。同时，错误在系统传播之前就应该被检测到。例如，如果容忍系统依赖于单元的冗余备份，就必须在所有备份单元崩溃之前发现攻击入侵错误。在错误影响到容忍机制之前检测到错误也是很重要的。

2 基于入侵容忍的CA认证中心方案

2.1 系统体系结构

系统由秘密存储库、CA、RA(Register Authority)、应用实体(用户)和入侵容忍部件构成。秘密存储库、CA、RA、应用实体已在其它文章中有所介绍，本文主要对入侵容忍机构进行详细描述。入侵容忍系统体系结构如图1所示。

作者简介：郭 萍(1973－)，女，讲师、硕士，主研方向：信息安全，密码学

收稿日期：2006-07-10 **E-mail：**guohelen_8@yahoo.com.cn

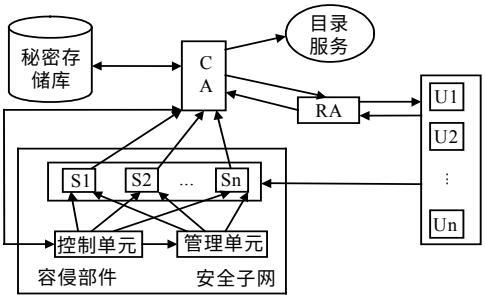

图 1 入侵容忍系统体系结构

2.1.1 共享服务器

S1,S2,…,Sn 表示 n 个共享服务器。不必保证它们每个都是安全的，但是须保证它们中的绝大多数是安全的。它们拥有 CA 签名私钥或用户私钥的一份并在每个共享服务器上都运行着一个服务程序；共享服务器能管理多个密钥而且可以为很多用户服务。

2.1.2 用户单元

用户单元是任何使用共享服务器的应用程序，如 CA 服务器、用户 Ui 等。当一个用户与共享服务器建立连接时，它首先要证明自己是已被授权的。接着，它与共享服务器相互作用，使用存储在共享服务器上的私钥 di 来对消息 M 进行数字签名或解密。

2.1.3 管理单元

管理单元用来管理所有的共享服务器。准确地说，是管理存储在共享服务器上的密钥。如有必要的话，它能关闭或挂起某个服务器，或者当某些共享服务器遭受攻击后，指示其它服务器采取合适的措施进行处理。根据实际情况管理单元可以对共享服务器发出重新生成新密钥、关机、挂起、刷新等命令。

2.1.4 控制单元

控制单元负责共享服务器的初始化、子密钥份额的分配、时钟同步、签名过程中收集共享服务器及 CA 的信息，分析这些数据后，告诉共享服务器在特定的时间内该做什么。比如：CA 会告诉控制单元需要 1、2、5 号服务器来使用密钥份额 d1、d3、d5 的组合给某个信息签名。片刻以后，共享服务器 1、2、5 每个都要告诉控制单元它们正在产生一个签名。

2.2 基于入侵容忍的 CA 签名方案

2.2.1 系统模型

如图 2 所示，系统由影子服务器组S= $\{s_1, s_2,\ldots, s_n\}$、用户组U=$\{u_1, u_2,\ldots,u_k\}$、定时刷新服务器TFS、密钥分配服务器KDS和发布公告服务器PBS组成。在初始化阶段，密钥分配服务器KDS完成影子服务器组S的密钥计算及分配；其后在整个签名过程中，定时刷新服务器以一定的周期对各个影子服务器进行刷新，同时密钥分配函数f(Key)对各影子服务器置以新的密钥分配方案，并使得新的影子服务器仍然共享同一个秘密(考虑到有攻击者存在，刷新阶段也要对被攻破服务器的影子进行重构，从而确保共享秘密的长期正确性)，在刷新影子时，各服务器的老影子均被销毁；发布公告服务器PBS用来存放系统的公开参数，系统各方均可访问此公告栏的内容，但无权修改；服务器与用户之间保持有认证的通道连接。

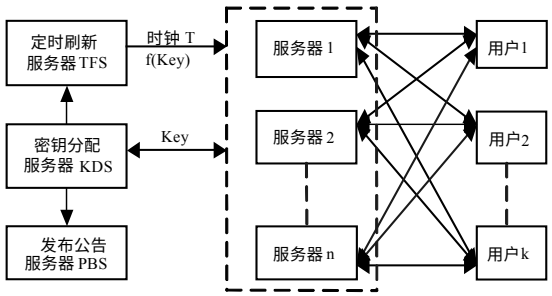

图 2 入侵容忍 CA 签名系统模型

2.2.2 密钥生成

密钥分配服务器完成以下步骤：

(1)参数准备。获得椭圆曲线系统参数[6]

$T = (p, F, a, b, G, n, h)$

其中 p 是奇素数，F 是一有限域，G 表示椭圆曲线上的一个基点，n 为素数且为点 G 的阶，h 是单向哈希函数，如 SHA-1。

(2)密钥分配服务器 KDS 在区间 $[1, n-1]$ 上选择一个随机数 d。

(3)计算 $Q = dG$，如果 Q 是无穷远点或 G 转(2)，否则 Q 是公钥。

(4)按照t-out-of-n的共享方案[7]，将秘密 d 分割成多种组合，d 的每种组合都与包含 t 个不同的服务器子集相对应。这里参数 t 为签名门限值，不少于 t 个服务器可以实现签名，n 为全部服务器数目。配置方法如下：密钥分配服务器从影子服务器组S中选择 t 个服务器 $s_1, s_2, ..., s_t (t \le n)$，将已拆分好的子密钥 d_i ($i = 1, 2, ..., n$，且有 $d = \sum_{i=1}^{t} d_i$)与密钥组标识绑定后，通过安全方式分别秘密传送给 n 个服务器 $s_1, s_2, ..., s_n$。表 1 是一个 3-out-of-4 的秘密共享方案。

表 1 3-out-of-4 秘密共享方案

密钥组标识 key	影子服务器 1	影子服务器 2	影子服务器 3	影子服务器 4
1	d_1	d_2	d_3	d_3
2	d_1	d_2	d_4	d_4
3	d_1	d_1	d_3	d_4
4	d_2	d_2	d_3	d_4

它有以下几种组合：

1) $d = d_1 + d_2 + d_3$；

2) $d = d_1 + d_2 + d_4$；

3) $d = d_1 + d_3 + d_4$；

4) $d = d_2 + d_3 + d_4$；

(5)对 d_i (i =1, 2,…, n)，KDS 计算公钥 $Q_i = d_i G$，Q_i 与用户标识 ID_{U_i} 及密钥组标识 ID_{key_i} 绑定后，将{ ID_{U_i}，ID_{key_i}，Q_i }发布在公告服务器上。

2.2.3 签名生成

用户签名请求如下：

(1)签名者 u_i (1≤i≤t)选取一密钥组的组合，同时选定一服务器组。

(2)设签名消息为 M，u_i 随机产生一整数 k_i (1≤ k_i ≤n-1)，计算 $R_i = k_i G$，并将 R_i 广播至公告栏。

(3) u_i 向第 1 步选定的服务器组发送签名请求 $<\mathrm{Re}\,quest, ID_{u_i}, M, ID_{key_i}, k_i, R_i>$，同时启动本机定时器进入计时等待状态。

服务器组响应：选定的服务器组 s_r $(1 \le r \le t)$ 接收到 u_i 的请求后，认证 u_i 身份，然后执行如下步骤：

(1)计算 $r = R_{ix} (\mathrm{mod}\, n)$，$R_{ix}$ 是点 R_i 的 X 轴坐标值；$s_i = d_i r + k_i h(M) (\mathrm{mod}\, n)$。

(2) s_r 向 u_i 发送响应消息 $<\mathrm{Re}\,sponse, r, s_i>$。

2.2.4 用户端签名重构

在签名时间定界内，u_i 收到 t 个 $<\mathrm{Re}\,sponse, r, s_i>$ 响应消息后(i =1, 2,…, t)，计算 $s = \sum_{i=1}^{t} s_i$，然后输出 (r, s) 作为签名者的数字签名。

2.2.5 验证签名

(1)椭圆曲线系统参数 $T = (p, F, a, b, G, n, h)$、用户的公钥 Q 和签名 (r, s)。

(2)用户 u_i 检查 r 是曲线上的有效点，并且 $1 < s, r < n-1$。

(3)用户 u_i 计算 $e = h(M)$，$w = e^{-1}$ 和 $V = swG - rwQ$。

(4) v_x 是点 V 的 x 坐标模 n。如果 v_x =r，则接受签名，否则拒绝签名。

如果成立，(r, s) 即为 M 的签名，用户 u_i 接受 (r, s)。反之，在服务器组中存在一个或多个欺诈服务器或故障服务器，u_i 需检查每一份签名的正确性，识别出欺诈(故障)服务器，过程如下:

(1) u_i 计算 $e = h(M)$，$w = e^{-1}$ 和 $V_i = s_i wG - rwQ_i$，v_{ix} 是点 V_i 的 x 坐标模 n。检查 $v_{ix} = r$ 是否成立，这里 $1 \le i \le t$。如果不成立，则签名 s_i 存在欺诈，相应的服务器 s_r 存在故障或已被入侵；如果成立，则可认为 s_r 是诚实的，u_i 需继续识别其他服务器的响应消息，直到发现欺诈(故障)服务器。

(2)可根据系统的运行情况和安全策略每间隔一段时间进行一次密钥刷新。这样，一次泄漏就不会造成太大的危害。密钥刷新只需要重新随机选择并计算子密钥 d_i，并按照既定的配置完成分配即可。由于密钥 d 保持不变，因此，可以保证方案是前向安全的。但若某服务器由于通信或系统部件故障等原因，在刷新周期内未送回签名消息，则 u_i 应重新选择服务器组。u_i 回到用户签名请求阶段，重新进行签名请求。

本签名方案以门限密码学[8]为基础，各服务器(也称参与者)分别持有该秘密信息的一个共享(或称影子)，周期性地刷新影子，并使得新的影子仍然共享同一个秘密(考虑到有攻击者存在，刷新阶段也要对被攻破服务器的影子进行重构，从而确保共享秘密的长期正确性)，在刷新影子时，各服务器老的影子均被销毁，使得攻击者获得的以前周期的影子信息对其在当前和以后周期进行的攻击没有任何帮助作用。因此攻击者只有在一个周期内攻破的服务器个数超过门限值，并获得它们当前的影子，才能真正攻破系统；否则，由于攻击者在每一个周期所获得的影子值都没有达到门限值，因而每个周期中都不能获得秘密。基于这种思想来共享秘密的算法被称为主动秘密共享算法[9]，与门限系统相比，主动秘密共享算法构成的系统大大提高了CA私钥的安全性。

本方案是将 CA 的私钥采用上述思想进行主动共享，再配合相应的主动签名算法进行证书的签发，整个 CA 系统的安全性得到极大提高，使整个系统具有较好的容侵能力。

2.3 入侵容忍 CA 认证中心的工作过程

2.3.1 用户注册

一个用户如要与其他用户进行通信，首先必须拥有自己的私钥和公钥。也就是说，必须先进行注册。假设有用户 Alice 要注册(系统参数如 2.2 节)：

(1)系统为 Alice 随机选取整数 d∈[l, n-l]作为其私钥，计算 Q=dG∈E(F)作为其公钥。

(2)随机选取整数 r∈[1, n-l]，计算：

1)V=(x, y)=rG；

2)s=r+xd (mod N)。

(3)系统检查(s, V, Q)，确保 Q 与其它用户不同，否则，重新选择密钥。

(4)生成 Alice 的数字证书并发布。

(5)将 d 通过安全渠道送给 Alice，至此 Alice 完成注册。

2.3.2 证书生成

生成数字证书的过程就是用 CA 私钥对用户注册信息进行签名的过程。

(1)用户 Alice 的主要信息(s, V, Q)；

(2)单向 Hash 函数作用于(s, V, Q)，生成消息摘要；

(3)选定 CA 签名服务器的一组影子服务器；

(4)CA 的各个影子服务器分别对消息摘要进行签名，得到各子签名；

(5)对各子签名进行重构，得到 CA 对 Alice 信息的签名；

(6)CA 生成 Alice 证书并发布。

以上过程如图 3 左边部分所示。

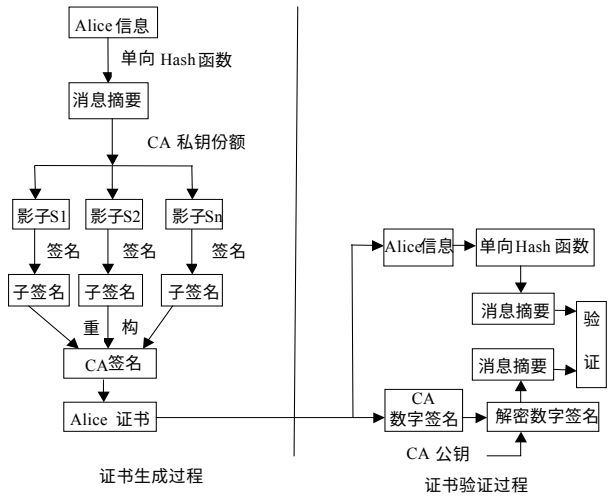

图 3 证书生成及验证过程

2.3.3 证书验证

假设用户 Bob 要和用户 Alice 通信，首先要取得 Alice 的数字证书。原则上说 CA 颁发的证书是 CA 信任域内所有用户都可信的，无需验证，但在第 1 次使用时也可以进行与所要通信用户的数字证书的验证。验证数字证书的过程就是用 CA 公钥解密 CA 对该用户的签名得到的消息摘要与对用户注册信息用 CA 签名算法重新生成的消息摘要进行比对的过程。

(1)用 CA 公钥对用户 Alice 的 CA 签名进行解密，得到消息摘要；

(2)用与 CA 签名时相同的单向 Hash 函数作用于用户 Alice 的信息(s, V,Q)，生成消息摘要；

(3)比较(1)、(2)步中得到的消息摘要；

(4)如(1)、(2)步中得到的消息摘要一致，则 Alice 的数字证书确实是 CA 颁发的，验证结束。

以上过程如图 3 右边部分所示。

3 结束语

本文在充分研究PKI技术的原理、应用、建设及运行，尤其是在CA的原理、实体模型及职能、入侵容忍技术、主动共享的门限密码方案、椭圆曲线密码体制及国内外已有入侵容忍认证中心的设计方案的基础上，设计完成了入侵容忍CA认证中心的方案。详细论述了基于入侵容忍CA认证中心的体系结构、各组件间的相互作用、基于入侵容忍的CA签名方案及整个系统的工作过程。文中给出的基于椭圆曲线算法的数字签名方案是容错和高效的，并具有结构简单、安全性强等特点。与计算有限域上的离散对数相比，椭圆曲线上的离散对数计算更困难，目前技术下，160bit模长的ECDLP与 512bit模数的有限域上的离散对数问题安全性相当[10]。这样，在某些情形(譬如内存、带宽等资源受限)下，基于椭圆曲线密码体制的主动秘密共享方案要比基于离散对数问题的主动秘密共享方案具有更多的优势。通过该方案增强的系统，不仅支持多用户的多秘密共享，而且还保持了秘密的可再用性。这两个特征对实际的系统非常有用，这是因为不仅可以使秘密被取消而不影响其他的秘密，而且可以确保秘密本身不会被泄漏，即使在解密过程秘密也不会泄漏，增强了系统的整体安全性。

由于篇幅所限，本文没有对安全性及效率等方面进行详细分析，将在下一步的研究中进行完善。

参考文献

1 Adams C, Lloyd S. 公开密钥基础设施—— 概念、标准和实施[M]. 冯登国, 译. 北京:人民邮电出版, 2001.

2 William T, Polknelson E, Malpani H A. Public Key Infrastructures that Satisfy Security Goals[J]. IEEE Internet Computing, 2003, 7(4): 60-67.

3 WU T, Malkin M, Boneh D. Building Intrusion Tolerant Applications [C]//Proceedings of the 8th USENIX Security Symposium, Washington. 1999: 79-91.

4 荆继武, 周天阳. Internet 上的入侵容忍服务技术[J]. 中国科学院研究生院学报, 2001, 18(2): 119-123.

5 Axelsson S. Research in Intrusion-detection Systems: A Survey[C]// Proceedings of the 22th Nstional Information Systems Security Conference. 2002: 62-75.

6 徐秋亮, 李大兴. 椭圆曲线密码体制[J]. 计算机研究与发展, 1999, 36(2): 1281-1288.

7 张峪峰, 刘锦德. 一种基于门限ECC的入侵容忍CA方案[J]. 计算机应用, 2004, 24(2): 5-9.

8 Desmedt Y, Frankel Y. Threshold Cryptosystems[C]//Proc. of Advances in Cryptology Crypto′89. Berlin: Springer-Verlag, 1990: 307-315.

9 Herzberg A, Jarecki S L, Krawczyk H, et al. Proactive Secret Sharing or: How to Cope with Perpetual Leakage[C]//Proc. of Cryptology-Crypto'95. Berlin: Springer-Verlag, 1995: 339-352.

10 张险峰, 秦志光, 刘锦德. 椭圆曲线加密体制的性能分析[J]. 电子科技大学学报, 2001, 30(2): 144-147.

（上接第 143 页）

4.4 支付网关的验证

支付网关验证程序需要的信息有：数字信封文件，支付信息文件，消费者签名公钥文件以及支付网关的加密私钥文件。验证的步骤如下：

(1)读取数字信封文件，采用支付网关的加密私钥对其进行解密，得到对称密钥。

(2)读取支付信息文件，并使用(1)中得到的对称密钥进行解密。

(3)对(2)中解密后的支付信息进行分析，通过识别“ && ”将订购信息的哈希、支付信息及双重签名分离。对于支付信息，通过识别“ || ”提取支付信息中的“ 银行账号 ”和“ 支付金额 ”。

(4)使用 SHA1 算法对支付信息进行哈希，然后使用“ && ”将其与(3)提取出来的订购信息的哈希连接在一起，再次哈希，得到本地计算出的支付信息哈希与订购信息哈希连接的哈希。

(5)使用消费者的签名公钥对双重签名进行解签名，然后和(4)中得到的哈希进行比较，如果相同，则验证成功，否则验证失败。

5 结束语

本文的实现目前用于我们开发的安全通信协议实验平台中 SET 协议的实验演示，通过演示，加深学生理解 SET 协议的基本原理。SET 协议采用双重签名，保证消费者、商家及银行进行数据传输的安全。其实不仅在 SET 协议中，只要是涉及多方通信时，需要提供身份认证、信息完整性认证及交易的防抵赖服务时，都可以采用双重签名这种技术，将发往各方的信息相互隔离，进行签名，保证多方通信安全。文中所提出的双重签名设计方案及实现，不仅可用于 SET 协议，而且也可以方便地应用到其它需要解决多方通信安全的系统中。

参考文献

1 Stallings W. 网络安全要素——应用与标准[M]. 潇湘工作室, 译. 北京: 人民邮电出版社, 2000.

2 韩宝明, 杜 鹏, 刘 华. 电子商务安全与支付[M]. 北京: 人民邮电出版社, 2001.

3 Templeman J, Vitter D. Visual Studio.NET Framework 技术内幕[M]. 邓劲生, 张晓明, 译. 北京: 中国水利水电出版社, 2003.

4 Thorsteinson P, Ganesh G G A. .NET 安全性与密码术[M]. 梁志敏, 蔡 建, 译. 北京: 清华大学出版社, 2004.

5 Rahmel D. .NET Framework 程序员查询辞典[M]. 陈 君, 译. 北京: 中国铁道出版社, 2003.

6 徐 可, 熊 伟, 袁和金, 等. Visual C#.NET 深入编程[M]. 北京: 希望电子出版社, 2001.