# 基于 P2P 环境的分布式数字签名研究及应用*)

刘汝正
(广东海洋大学网络与教育技术中心 广东湛江 524088)

**摘 要** 随着网络安全问题越来越受到重视,数字签名(Digital Sign)技术由于可以提供网络身份认证功能,因而其应用越来越广泛。本文从分析 P2P 网络的路由算法入手,利用门限密码技术设计了一个基于 P2P 网络的分布式多重数字签名平台,重点讨论了 VSS 可验证子密钥的生成和分发,最后对其进行了安全性和密钥分发(VSS)时间的测试。

**关键词** 多重数字签名,P2P,VSS

## Study and Realization of Digital Multi-sign Based on P2P Network

LIU Ru-zheng
(Network and Education Technology Centre, Guangdong Ocean University, Guangdong Zhanjiang 524088, China)

**Abstract** With more and more important the internet security becomes, digital sign technology is widely used because it can authenticate person on internet. Beginning with the analyzation of router algorithm, by using threshold cryptography, we design a distributed fault tolerant digital multi-sign system based on P2P network. Then the form and distribute of VSS child-key are discussed particularly, lastly the security and VSS distribute time are tested.

**Keywords** Digital multi-sign, P2P, VSS

## 1 引言

数字签名技术[6](Digital Sign)是一项用途非常广泛的技术,它以公钥密码系统为基础,可以提供如密钥加密、身份验证等网络交易中非常关键的功能。在网络交易、认证服务以及 DRM 等领域都需要数字签名技术,同时由于数字签名可以确保信息的完整性,使得其在更广泛的领域都发挥巨大的作用。认证功能是公钥密码技术区别于对称加密技术的独有的特征。

本文重点研究了基于 P2P 网络的分布式多重数字签名的 VSS 问题,并提出了一个分布式多重数字签名方案,最后对其进行了理论和实验测试。

## 2 P2P 系统基本问题及其路由算法

在基于 Peer to Peer 网络结构的系统中,两个节点之间的通信是通过消息在各节点之间的传递来实现的。Peer to Peer 路由问题[1]就是关于如何确定系统中任意两个节点之间的消息传递路径的问题。Peer to Peer 路由算法就是关于如何确定系统中任意两个节点之间消息传递路径的方法。一般情况下,为了提高通讯的可靠性,Peer to Peer 路由算法能保证任意两个节点之间存在多条消息传递路径。确定了消息传递路径,也就确定了系统中各节点的连接关系,从而也就确定了 Peer to Peer 网络的拓扑结构。因此,P2P 系统的最核心问题就是对象定位的路由算法。

下面,我们首先对 Peer to Peer 网络中的重要概念[2]进行定义。Peer to Peer 系统中的一台主机称为一个节点。如果两个节点互知对方的 IP 地址,则称在这两个节点之间存在一个连接。延迟是指一次通信过程中,消息从源节点到目标节点所经过的连接数,用 hop 来描述。为了实现 Peer to Peer 网络,每个节点上需要保存一个与其有连接关系的节点的 IP 地址列表,称为邻居列表。同时,为了支持通信,每个节点还需要保存一个建立在 IP 地址列表基础上的消息转发目标表,称为路由表。一种路由算法和一个 Peer to Peer 网络之间存在一种对应关系,路由算法的一些特征可以通过一些网络特征体现出来。

近几年来,人们提出了一系列专门针对 Peer to Peer 路由问题的通讯机制:[Plaxton 1997],Can[Ratnasamy 2001],Chord[Stoics 2001],GLS[Li2000],Tapstry[Zhao 2001]和 Pastry [Rowstron 2001]等。这些路由算法综合考虑了系统的可扩展性、通信的效率和通信的可靠性,在三者之间做了有效的折衷,从而更适合构造大规模分布式系统。

## 3 可验证秘密分享 VSS(Verified Secret Share)介绍

在建立秘密分享(secret share)系统时,首先要将秘密组分(share)分发到各 share 服务器上。每个 share 服务器将收到一个秘密组分,但是并不知道这个秘密组分与其它 share 服务器收到的秘密组分是否能协同运作,即重构出完整秘密。因此,需要一种检验算法,即能够验证所收到的秘密组分的正确性,还要避免泄露秘密。

可验证秘密分享 VSS(verified secret share)技术就是人们为解决这个问题而提出来的。早期由 Chaum, Ben-Or [3]等人提出了无条件安全的 VSS 方案并将该方案用来设计安全的分布式协议。但是这些 VSS 方案都是交互性(interactive)的,为了验证秘密组分,share 服务器之间需要交互信息。在这些方案中,都允许最多只有 1/3 的 share 服务器出错,这就使得这类方案的使用受到很大限制。

*)广东省科技计划项目资助(项目编号:2006B36501009,项目名称:海洋义栖生物活性物质的普查、分离和提取研究)。刘汝正 讲师,主要从事面向对象技术研究。

后来由 Pedersen，Feldman[4] 等人提出了非交互的 VSS 方案，这类方案基于秘密分发者广播证据信息，share 服务器利用证据验证的做法，避免了 share 节点间的交互，可用性比起交互式 VSS 方案大大加强，同时需要传输的验证信息也减少了。

## 4 t-out-of-n 门限密码算法介绍

t-out-of-n[5] 秘密分享算法非常简单有效，不仅重构时计算量小，而且有很好的安全性。

具体算法如下：

将私钥 $d$ 分解成 $t$ 个随机数之和：

$d=d1+d2+\cdots+dt$

再将 $d_i$ 分配到 $i$ 台服务器中，签名时将需要签名的信息 $M$ 发送到这 $t$ 台服务器中，各服务器将计算结果 $M_i=M_{di}$ 送回客户机，客户机计算：

$S=\prod M^{di}=M^{d}$

就得到了签名后的信息。

为了提供冗余性，将密钥 $d$ 作多组拆分，得到

$d=d_{11}+d_{12}+\cdots+d_{1t}$

$d=d_{21}+d_{22}+\cdots+d_{2t}$

…

这样，就把密钥分成了多组子密钥，任意一组子密钥组合在一起可以重构出 $d$。将这多组子密钥按一定方法放入 $n$ 台服务器中，每台服务器有多个子密钥，目标是其中任意 $t$ 台服务器可以找到一组完整的子密钥来进行重构。换句话说，就是允许系统有任意不多于 $(n-t)$ 台服务器崩溃。因此整个系统获得了 $(n,t)$ 门限的容错能力。

## 5 基于 P2P 网络的分布式多重数字签名系统

本文提出了一个分布式数字签名系统，基于 t-out-of-n 门限算法，利用 peer 节点提供数字签名服务。系统框架如图 1。

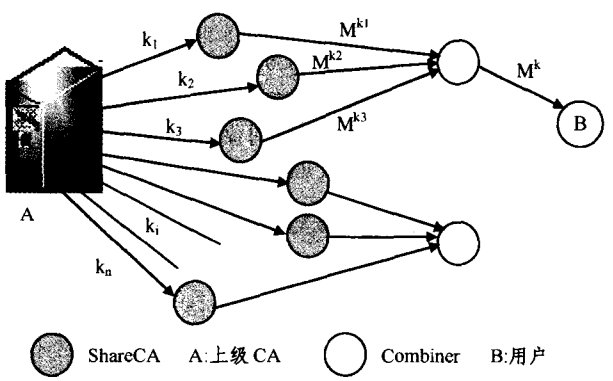


图 1 P2P 分布式多重数字签名系统

在系统平台中，上级服务器负责产生和分发子密钥。由于上级服务器是整个系统中唯一掌握完整私钥的，因此只有在分发子密钥时才会在线，平时都是离线状态。

该系统由 share 节点和 combiner 服务器组成两层结构，其中每 $t$ 个 share 节点对应一个 combiner 服务器，构成一个组。系统由很多个组构成。每一个 share 节点保存一个子密钥，同时这种两层结构也起到了节点隐藏的作用，只有 combiner 节点才知道 share 节点的位置，且 share 节点之间也都不知道其他节点的地址。用户将签名请求发给 combiner 节点即可。系统总共有至少 $t$ 个 combiner 节点，而 share 节点总数 $n$ 也至少为 $t2$ 个。这种两层结构解决了 t-out-of-n 算法固有的密钥管理的问题，同时还使 share 节点具有一定的隐藏性，提高了系统的安全。当用户将需要签名的信息 $M$ 发送给 combiner，combiner 将 $M$ 转发给 $t$ 个 share 节点，每台 share CA 服务器利用自己的子密钥 $d_i$ 计算：

$S_i=M^{di} \bmod N$

传回 combiner，combiner 收到 $t$ 份 $S_i$ 后，计算：

$S=M^{d} \bmod N=\prod_{i=1}^{t} S_i \bmod N$

得到签名证书 $S$ 后，利用公钥进行验证：

$M=S^{e} \bmod N$

其中 $e$ 和 $N$ 是公钥。

如果验证正确，就将签名的证书 $S$ 发送给用户。

该系统是一个分布式数字签名系统，可以对外提供公钥数字签名服务，因而私钥的安全性是非常重要的。尽管利用门限密码算法提高了安全性，但是仍然在密钥分发、密钥签名等各个阶段受到攻击。因此，该系统除了使用 t-out-of-n 门限密码进行门限签名以外，还使用了多个安全协议算法，在各个不同阶段进行容侵。

### 5.1 子密钥生成

在该系统中，唯一掌握完整数字签名私钥的是上级服务器，同时上级服务器也负责产生所有的子密钥。该系统的数字签名基于 RSA 公钥技术。首先，上级服务器的 RSA 模块产生 RSA 公私密钥对、公钥 $e$ 和 $N$、私钥 $d$。为了运算加速，其中公钥 $e$ 取值为 17(0x11)，65537(0x10001)，257(0x101)，4097(0x1001)中的一个。将公钥 $e$ 和 $N$ 广播给所有 combiner 和用户节点。

上级服务器使用 t-out-of-n 秘密分享算法将密钥 $d$ 分割，算法如下：

$d=d_{11}+d_{12}+\cdots+d_{1t}$

$d=d_{21}+d_{22}+\cdots+d_{2t}$

……

$d=d_{i1}+d_{i2}+\cdots+d_{it}$

要将每一组 $d_{i1}--d_{i1'}$ 分配给其中一个 combiner 节点 $i$ 所对应的 $t$ 台 share 节点，每个 share 节点只需保存一个子密钥。

### 5.2 VSS 可验证子密钥分发协议

密钥分发之前，上级服务器首先要选择 $t$ 个 P2P 节点作为 share 节点，一般需要该 peer 节点具有以下两个条件：

①Peer 节点具有合法公钥证书；

②Peer 节点长期在线。

上级服务器(dealer server)对一组 share server 进行密钥分发。由于在 P2P 环境下，很难构建安全通道，因此采用加密传输子密钥来确保密钥安全。

首先，上级服务器计算出一组子密钥 $d_1d_2d_3$ 后，先用系统私钥 $d$ 对子密钥签名，然后用每个 share 节点各自的公钥对签名进行加密。这样，子密钥就可以安全地发送到 share 节点，同时对密钥的签名也保证了密钥不被篡改。除非攻击者窃取了 share 节点的私钥，否则无法得到子密钥。

密钥分发示意图如图 2。

尽管子密钥 $d_i$ 经过签名和加密两重保护，理论上使攻击者很难进行窃取和篡改，但是无法抵抗重放攻击，即将以前上级服务器分发过的子密钥来替换现有子密钥，从而使得 share 节点收到过期的或者是本应发给其他节点的子密钥，造成签

名失败。一般的抗重放攻击的方案通常需要服务器之间时间同步，通过增加时间戳的方式来抗重放，但在 P2P 环境下实现时间同步是很困难的。因此，该系统引入 VSS 可验证密钥分发(verified secret share)协议，使得每个 share 节点通过验证计算可以检验自己收到的子密钥的真伪。

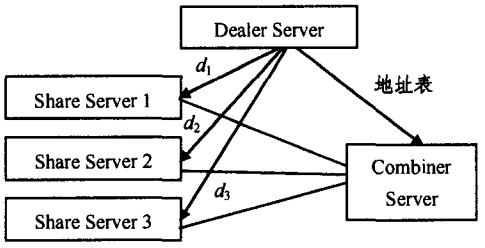


图 2　多重数字签名系统密钥分发

VSS 分发协议：

Step1　上级服务器生成验证消息 $g$，$g$ 的长度小于 $N$ 的长度。$g$ 随子密钥 $d_{ij}$ 一起加密传输给 share 节点 $j$；

Step 2　上级服务器计算验证信息：

$g^{d_{i1}} \bmod N, g^{d_{i2}} \bmod N \cdots g^{d_{it}} \bmod N$

其中，$d_{i1}+d_{i2}+\cdots+d_{it}=d$；

Step 3　上级服务器向第 $i$ 组 share 节点广播证据：

$\{g^{d_{i1}} \bmod N, g^{d_{i2}} \bmod N \cdots g^{d_{it}} \bmod N\}$；

Step 4　share 节点 $j$ 收到证据后，计算

$A=(g^{d_{i1}} \times g^{d_{i2}} \times \cdots \times g^{d_{ij}} \times \cdots \times g^{d_{it}}) \bmod N$

其中 $g^{d_{ij}}$ 项为节点 $j$ 用自己收到的子密钥 $d_{ij}$ 计算得出。

Step 5　share 节点 $j$ 验证：$g=(A)^e \bmod N$

其中，$e$，$N$ 为 RSA 公钥。

算法证明：

$$\begin{aligned}(A)^e \bmod N &= \{(g^{d_{i1}} \times g^{d_{i2}} \times \cdots \times g^{d_{ij}} \times \cdots \times g^{d_{it}}) \bmod N\}^e \bmod N \\ &= (g^d \bmod N)^e \bmod N \\ &= g \bmod N \\ &= g\end{aligned}$$

分发子密钥的同时，服务器会将包含这 $t$ 个 share 节点地址的地址表同样先签名，再用该组 combiner 服务器的公钥加密后传给 combiner。这样，combiner 节点就知道了该组 $t$ 个 share 节点的地址，而 share 节点之间互相不知道对方地址。

## 6　安全性分析及测试分析

### 6.1　安全性分析

首先，验证信息 $g$ 是和子密钥 $d_i$ 一同被加密签名保护传送到 share 节点的，因此 $g$ 可以保证没有被篡改。

Share 节点 $i$ 在验证时计算

$A=(g^{d_1} \times g^{d_2} \times \cdots \times g^{d_j} \times \cdots \times g^{d_t}) \bmod N$

$g=(A)^e \bmod N$

假设 share $i$ 收到错误的子密钥 $d'i$ 和 $g'$，如果想让下式成立：

$g^d \bmod N=(g^{d_1} \times g^{d_2} \times \cdots \times g^{d_j} \times \cdots \times g^{d_t}) \bmod N$

则攻击者需要：

①事先知道 $g^{1d} \bmod N$ 的值。在不知道私钥 $d$ 的前提下这是很困难的。

②即使攻击者事先掌握了 $g^{1d} \bmod N$，也必须将发往 share $i$ 节点的所有 $t-1$ 个 $g^{1d} \bmod N$ 都拦截并替换。

如果攻击者具备了以上两个条件，有可能欺骗过 VSS 协议。但是在接下来的数字签名中，出错的节点 share $i$ 将被子签名验证协议发现并替换。

### 6.2　测试结果及分析

生成子密钥后，就进入密钥分发阶段了。我们进行了两组实验，分别有 2 台 share 节点和 3 台 share 节点，即门限 $t=2$ 和 $t=3$。测试从 server 向 share 发送子密钥和验证信息开始，直到 share 节点对收到的子密钥验证完毕这一过程的时间效率。我们分别对 RSA 密钥长度为 512 位、1024 位和 2048 位的情况进行了对比测试，时间单位为秒。测试结果如表 1。

表 1　多重数字签名密钥分发(VSS)时间

| RSA 密匙长度 | 512bit | 1024bit | 2048bit |
|---|---|---|---|
| t=2 | 0.087 | 0.135 | 0.383 |
| t=3 | 0.095 | 0.157 | 0.415 |

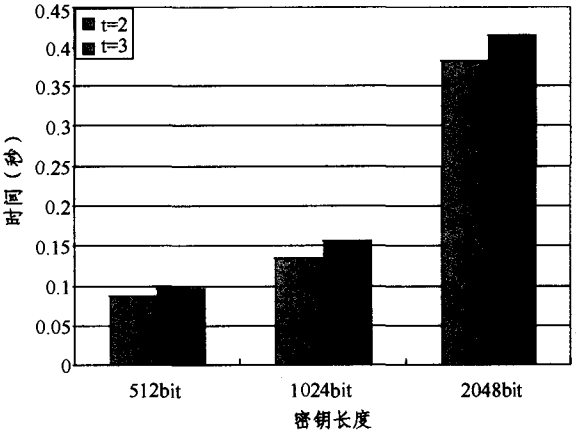


图 3　密钥分发(VSS)时间

从实验结果可以看出，在局域网的环境下，门限值 $t=2$ 和 $t=3$ 时，在三种密钥长度的对比测试中，两者的时间差距并不大，这主要是因为局域网网络情况良好。尽管如此，仍可以看到，门限值 $t$ 的大小对系统效率的影响。同时可以看到密钥长度为 512bit 和 1024bit 时，时间增长并不大，而 2048bit 时则增长很大。实际上这主要是因为密钥长度 512bit 时，由于单机的计算量与后两者相比小很多，而网络传输代价则并没有减少太多，因此造成在 512bit 时的签名时间长度接近 1024bit 时的情况。

## 参 考 文 献

[1] Flenner R, Abbott M. Java P2P 技术内幕. 高岭，等译. 人民邮电出版社，2003

[2] Liben-Nowell D, Balakrishnan, H, Karger D R. Observations on the dynamic evolution of peer-to peer networks// First International Workshop on Peer-to-Peer Systems. Cambridge, A14, Alar, 2002

[3] Ben-Or M, Goldwasser S, Widgerson A. Completeness theorems non-cryptographic fault-tolerant distributed computation// Proceedings of the annual ACM Symposium on Theory of Computing. 1988:1-10

[4] Pedersen T P. Non-interactive and Information-Theoretic Secure Verifiable Secret Sharing// Proceedings of the 11th Annual International Cryptology Conference on Advances in Cryptology. August 1991:129-140

[5] Desmedt Y, Di Crescenzo G, Burmester M. Multiplicative non-abelian sharing schemes and their application to threshold cryptography, Oakland, 1991:110-121

[6] 张世永. 网络安全原理与应用[M]. 北京：科学出版社，2003