主动安全技术发展综述*)

滕 猛 邹 鹏 王怀民
(国防科技大学计算机学院 长沙410073)

A Survey of Proactive Security

TENG Meng ZOU Peng WANG Huai-Ming
(School of Computer, National University of Defense Technology, Changsha 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 a non-trivial extension of (t+1,n)-threshold cryptosystems, where shares are periodically renewed (without changing the secret) in such a way that information gained by the adversary in one time period is useless for attacking the secret after the shares are renewed. In this paper, we describe the proactive approach and review some basic technologies. We also report the problems need to be study.

Keywords Proactive security, Threshold cryptographic, Proactive secret sharing, Adaptive-secure

1 引言

在电子商务和开放网络中,有一类高度机密且长期有效的数据需要保护和使用.直接使用传统密码学甚至门限密码学提供的方法都不能很好地保证其安全性.而主动安全技术则可以较好地解决这类问题.

以CA的私钥为例.一旦攻击者窃取了该私钥,就可以冒充该CA随意签发证书.这将造成不可估量的损失.因此如何保护CA的私钥就成为重要的安全问题.直接将其放在某一台机器上并使用它提供签名服务,这种办法显然不安全(尽管有许多系统正是这样做的).因为在现实世界里的攻击者完全可以通过系统攻击,获得对该机器的控制权,进而窃取该私钥;即使这台机器不与外部网络连接、离线提供签名服务(在线提供签名服务在电子商务的某些应用中是必须的),也不得不解决防止内部人员窃取私钥的问题.

使用门限密码学技术对CA私钥进行处理也不能完全保证其安全性.在使用门限密码学的签名系统中,一组服务器(也称参与者)来共同完成CA私钥的签名功能.所有参与者都不实际拥有CA私钥,而是以门限的方式分别持有该私钥的一个共享(也称影子).当需要签名时,一批参与者只有通过合作才能产生签名.由于私钥的各个共享只存在于它们所属的服务器上,别的机器不可能将其泄露.因而,对(t+1,n)-门限方案来说,攻击者要攻破这种安全系统,需要攻破超过t个服务器.这种方法大大提高了私钥的安全性.但是,CA私钥的有效期是数年以上,在如此长的时间里,攻击者仍然有足够的时间来逐一攻破整个门限系统的各个服务器,进而计算出CA私钥.

主动安全[1~3]是一种有能力保护象CA私钥一样的长效秘密的技术.它将时间分成一个个的周期,并综合使用门限密码学、数据刷新和入侵检测技术来达到保护秘密的目的.它以门限密码学为基础,各服务器分别持有该秘密信息的一个共享,周期性地刷新共享,并使得新的共享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的机器的共享进行重构,从而确保共享秘密的长期正确性).在刷新共享时,各服务器的老的共享均被销毁,使得攻击者获得的以前周期的共享信息对其在当前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超过门限值个服务器,并获得它们当前的共享,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共享值数目都没有达到门限值,因而每个周期中都不能获得秘密(本周期过后当前共享被销毁,因此以后周期攻击者也不会再获得本周期的其它共享).入侵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.对CA私钥应用主动安全技术,就构成了一个主动签名系统.主动签名系统系统可以有效保证CA私钥的安全性.

主动安全技术不仅可以应用于CA私钥,而且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.可以如下描述主动安全系统:有n个服务器的系统,如果在任何周期内攻击者只能成功地入侵最多t个服务器(t<n)的情况下,系统仍然是安全的和可用的,则被称为主动安全系统.

在主动安全的研究发展过程中,逐步形成了以下几种技术.1)在普通的网络平台上维护主动安全所要求的通信模型的技术被称为主动通信.2)在主动通信模型上完成共享产生、共享刷新、共享重构的部分被称为主动秘密共享.主动通信和主动秘密共享是主动安全系统的基础.3)使用共享完成签名或加密功能的部分被称为主动签名或主动加密(本文以主动签名为例).4)最初的主动技术(包括门限密码学),都没有考虑到自适应攻击者存在时的安全性问题.为此又有了主动系

*)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.

统的自适应安全技术。以下我们将分别介绍这几种技术。

2 主动秘密共享技术

主动秘密共享包括秘密产生、秘密共享、周期性刷新以及共享恢复。主动秘密共享构成主动安全系统的整体框架。

2.1 秘密共享

秘密共享(Secert Sharing,SS)协议由文[4]引入,通过将某个秘密值分布到多个服务器上对其进行保护。典型的秘密共享是将秘密值分布到 n 个服务器上,任何 t+1个服务器均可通过其拥有的共享恢复该秘密。如在 Shamir 共享方案中,秘密值 $s\in\Sigma=\{a|a\in Z$ 且$0\leqslant a<q\}$,其中 q 为素数。秘密的分发者随机选择 Σ 中 t 个数 $a_1,\cdots,a_t$。取多项式 $f(z)=s+a_1z+\cdots+a_tz^t$,分发者将共享 $s_i=f(i) \bmod q$ 分发给服务器 i。显然从任何 t 个服务器的共享中不能得到 s 的任何信息,而任何 t+1个服务器都可通过多项式插值计算出 s。

2.2 周期性刷新

主动秘密共享不仅要将秘密值分布到多个服务器上共享,而且要周期性地刷新这些服务器所拥有的共享。刷新协议保证新的共享独立于老的共享,但新的共享仍然共享着同一个秘密值。设秘密 s 上一个周期共享在 t 阶多项式 F(z)上,即 $s_i=F(i)$。周期性地刷新可以按如下方法进行[2]。每一个服务器 p_i 选择一个 t 阶随机多项式 $f_i(z)$使得 $f_i(0)=0$。然后 p_i 传送值 $s_{ij}=f_i(j) \bmod q$ 到 p_j。p_j 计算其新共享为 $s'_j=s_j+s_{1j}+\cdots+s_{nj} \bmod q$ 并清除掉老的共享 s_j。新的共享 s'_j 位于多项式 $G(z)=F(z)+f_1(z)+\cdots+f_n(z)$上,且 G(0)=s,即新的共享 s'_1,$s'_2,\cdots,s'_n$ 仍然共享着同一个秘密 s。

2.3 共享恢复

共享恢复用于被攻破的服务器(不妨设为 p_r)在成功地清除了攻击后恢复原来持有的共享。设秘密 s 共享在 t 阶多项式 f(z)上,则 p_r 待恢复的共享 $s_r=f(r)$。共享恢复可以按如下方法进行[2]:每一个服务器 p_i 选择一个 t 阶随机多项式 $g_i(z)$使得 $g_i(r)=0$(选择适当的常数项即可满足等式)。p_i 将值 $g_{ij}=g_i(j) \bmod q$ 传送给 p_j,p_j 计算一个辅助共享 $g'_j=g_{1j}+g_{2j}+\cdots+g_{nj} \bmod q$。辅助共享 g'_j 位于 t 阶多项式 $g'(z)=g_1(z)+g_2(z)+\cdots+g_n(z)$上,且 $g'(r)=0$。然后,每一个 p_i 给 p_r 传送一个新的辅助共享 $s_i+g'_i \bmod q=f(i)+g'(i) \bmod q$。$p_r$ 根据这些值,可构造出 t 阶多项式 $f(z)+g'(z) \bmod q$,进而计算出 $f(r)+g'(r)=f(r)+0=s_r$,即要恢复的共享值。

2.4 主动秘密共享

如果攻击者只进行被动攻击,即只能"读"服务器的数据,不能修改服务器的数据或导致服务器行为改变时,2.1～2.3节给出的过程就可完成对秘密的长期保护。但当攻击者可以进行主动攻击时,就要使用可验证的秘密共享(Verifiable Secert Sharing,VSS)扩展上述的方法,使得系统能够容忍一定数目的服务器持有错误的共享值。如可验证的 Feldman 方案的思想是:给定一个 t 阶多项式 $f(z)=a_0+a_1z+\cdots+a_tz^t$,分发者 Dealer 给 p_i 传送共享值 $s_i=f(i) \bmod q$,同时广播值$(\alpha_0,\alpha_1,\cdots,\alpha_n)$,其中 $\alpha_i=g^{a_i} \bmod p$。于是就有 $g^{s_i}=\prod_j(\alpha_j^{i^j}) \bmod p$。每个服务器都可以通过检查该式是否成立来判断 Dealer 是否有欺骗行为(其中的 p,q,g 满足一定的关系)。

Feldman 共享方案有一个不安全的因素[5]:在进行随机值共享时,如果有两个恶意参与者联合,可能破坏随机值中某一位的随机性。文[5]采用更复杂的 Pedersen 共享方案(Feldman 共享的一种扩展)对主动共享方案中的 Feldman 共享进行了替换,并证明了替换后的安全性。

2.5 分布式秘密产生

使用主动安全系统替换已有的非主动安全系统时,要使用秘密共享方案把已知的秘密共享到多个服务器上。但如果一个安全系统一开始就被设计成主动安全系统,则可以让多个服务器直接分布式产生秘密的共享(即分布式秘密产生)。而并不使用以下方法获得共享:首先实际产生一个秘密,再使用秘密共享方案对其进行共享,最后销毁秘密。这种技术的关键在于需要分布式地产生一些公开值,而且要与并不实际存在的秘密满足已知的关系。文[6]给出了多个参与者联合产生 RSA 密钥的有效技术。在其协议结束时,RSA 的模数 N=pq 被公开计算出来;任何参与者都不知道 N 的因子;公钥是计算后公开的;每一个参与者拥有私钥的一个共享,并能够进行门限加密/签名。文[7]对其进行了扩充,使得在恶意参与者(不超过总数的一半)存在的情况下也能够正确计算出所要求的值。

2.6 联合秘密共享

在周期性刷新时,由于已经没有了可信的分发者,因此需要使用联合秘密共享算法产生刷新时需要使用的一些中间值。联合秘密共享是由多个服务器联合产生关于某个秘密的共享而不需要分发者的一种算法。由它产生的秘密 s 仍然满足从任何 t 个服务器的共享中不能得到 s 的任何信息,而任何 t+1个服务器都可通过插值计算出 s。其思想是:n 个服务器同时运行秘密共享协议,每一个服务器 p_i 都作为分发者 Dealer 分发一个秘密 s_i 到服务器中去共享。最终共享的秘密是 $S=s_1+\cdots+s_n$。在联合随机秘密共享中,每个服务器 p_i 随机地挑选 s_i,因而 S 也是随机的。在联合零秘密共享时,每个服务器 p_i 选择 $s_i=0$,因而 S=0。无论是哪一种情况下,服务器只需要将它收到的所有部分值相加即可获得秘密值 S 在本地的共享。

3 主动通信技术

实现主动安全系统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。该方案存在两个缺陷。一是如果攻击者控制了通信链路,则可以进行中间人攻击。二是若攻击者同时入侵了两台服务器,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,即使在入侵者被清除掉之后,这两个服务器仍然失去了互相认证的能力。

文[8]通过以下方法解决了上述问题:各服务器持有一个签名私钥的共享,签名私钥本身并不实际存在,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;在每个刷新阶段,各服务器合作对每个服务器选择的公钥都进行签名,这个签名结果用于各方验证其它服务器选择的公钥。

4 主动签名技术

主动签名技术主要有两类:针对 RSA 机制的 RSA 类主动签名和使用离散对数的 DSS 类签名。

4.1 DSS 类

对于 DSS 类的主动签名，可以直接应用某些门限 DSS 签名方案[9]。DSS 签名算法主要思想是：p 和 q 是两个大素数，q 整除 p-1；g 是 Z_p 上的一个 q 阶元素；x 是签名者的私钥，$1 \leqslant x \leqslant q$；$y = g^x \bmod p$ 是验证密钥（公钥）；除 x 保密，只有签名者知道外，其它信息（y、p、q、g）均公开。当签名者要对信息 M 进行签名时，先用 SHA-1对该信息进行 hash。令 m 是 hash 后的结果值。签名者随机选取一个1～q 中的随机数 k，并置 $r = (g^k \bmod p) \bmod q$，和 $u = k^{-1}(m + xr) \bmod q$。数对(r,u)即为 M 的签名。针对标准 DSS 方案中签名部分，相应地构造主动 DSS 签名算法如下：

1）使用联合随机秘密共享算法，产生一个秘密值 k

2）各服务器计算 $r = (g^k \bmod p) \bmod q$ 的共享。

3）使用计算两个秘密乘积的共享的协议计算值 $u = k^{-1}(m + xr)$。

4）每一个服务器输出数对$\langle r, u_i \rangle$

利用$\langle r, u_i \rangle$计算出最终的签名$\langle r, u \rangle$。

文[10]基于主动秘密共享和门限方案，给出了一个主动化 DSS 类签名系统的通用方法，并证明了其安全性和有效性。使用该方法可以主动化的签名系统包括 DSS 签名、AMV 签名、Schnorr 签名、Undeniable 签名、Chaum/Pederson(CP)签名。

4.2 RSA 类

RSA 类算法的核心思想是：选取两个大的素数 p 和 q，计算公开值 n=pq，随机选取私钥 e，使其与(p－1)(q－1)互素，计算公钥 d，使得 $ed = 1 \bmod (p-1)(q-1)$即可。加密或签名时，计算 $c = m^e \bmod n$ 即可。在主动化 RSA 类算法与 DSS 类算法时，最大的不同在于：DSS 中公私钥满足关系：$y = g^x \bmod p$，其中的模 p 是公开值；而 RSA 中公私钥满足关系：$ed = 1 \bmod (p-1)(q-1)$，其中模数(p－1)(q－1)必须保密（否则利用代数学方法可以计算出保密值 p 和 q）。正是因为这一点，RSA 类的主动签名算法不能直接使用第3节介绍的基于多项式的主动秘密共享方法进行密钥的主动共享，而必须采用其它措施。

最早提出的方法是[11～13]：让所有服务器（设个数为 n）中的任意 k 个（门限值，k<n）服务器构成的子集都共享一个私钥值，每一个子集的所有共享值之和为私钥 e。这样同一个服务器在不同的子集中持有不同的共享值。例如，取 n=3，k=2，服务器依次为 s1，s2，s3，e=7（为简化讨论，取小的整数）。则共有3个子集 A={s1，s2}、B={s1，s3}和 C={s2，s3}分别共享私钥 e。假定 A 中 s1，s2分别持有影子4，3；B 中 s1，s3分别持有2，5；C 中 s2，s3分别持有4，3。则 s1实际持有两个影子值，$s_{1A}=4$，$s_{1B}=2$。对 s2和 s3分别有 $s_{2A}=3$，$s_{2C}=4$；$s_{3B}=5$，$s_{3C}=3$。显然，每一个子集都可以容易地计算出签名。当需要对影子进行刷新时，每一个子集分别进行刷新（如对 A 中的 s1和 s2进行刷新），产生新影子，保持影子之和仍为 e（如取 $s_{1A}=3$，$s_{2A}=4$）。

上述方法的缺点是每一个服务器持有的共享多（随着服务器数目的增加，每个服务器持有的共享呈指数增长），刷新比较复杂。特别是影子丢失后的重构更复杂。例如，上例中如果服务器 s1被攻破，则需要利用子集 C 对 A 和 B 两个子集中的共享进行重构。另外，上述方法还要求 p 和 q 都是强素数。

文[14,15]给出了另外一种方案。它将 e 分成两个部分，其中的一个部分可以使用第3节的方法进行主动共享，另一部分可以公开。令服务器的数目为 t，记 L=t!，取 $H = \gcd(d, L^2)$为最大公约数。由两个数互素的性质有：$1 = da + L^2H^{-1}b$，其中 a，b 可以通过欧几里得辗转相除法计算出来。记 $k = ebH^{-1} \bmod (p-1)(q-1)$，则有 $e = a + L^2k \bmod (p-1)(q-1)$。公开 a，共享秘密为 L^2k。这样，每次刷新时只要对 L^2k 进行刷新即可。由于共享的秘密 L^2k 是 t!的倍数，所以当选取适当系数（L 的整数倍）的多项式对其进行共享后，每一个服务器共享的影子都是 t!的倍数，这样在利用多项式插值计算签名时可以避免取模运算（插值计算时的分母必是 t!的因子，故可以整除，不必通过取模求倒数）。通过这种方法，就可以将 RSA 的私钥利用多项式共享。该方法的一个缺点是：每一次签名后都必须对共享秘密进行刷新，因而影响效率。

5 自适应主动安全技术

自适应安全问题同利用“仿真器[16]”证明分布式协议的安全性有关。以签名协议为例，分布式签名协议的安全性是指在原集中方式下安全的签名方案（不可伪造），在分布之后仍然是安全的（不可伪造）。一般采用反证法来证明。即：给定一个在分布式环境下可以伪造签名的攻击者 A，证明可以相应地构造出攻击者 B（即仿真器），他在集中签名方式下可以伪造签名。这与假设矛盾，从而得出“分布后的协议也是安全的（不可伪造）”的结论。该反证法的关键就是要构造一个仿真器。仿真器要能够仿真攻击者 A 在分布式签名协议运行时（通过攻击）所能获得的信息，同时又要将分布式情况下的攻击转化成集中方式下的攻击。以前的分布式协议安全性证明均假定攻击者在签名协议运行之前已经选定了要攻击的服务器（即静态攻击），这时比较容易构造出仿真器，因为只需要将攻击者 A 所选定服务器的内部状态保持不变，就可以构造出仿真器。但如果攻击者 A 是自适应的，即他在协议运行过程中才决定攻击哪些服务器，这时构造仿真器就很困难。原因是：攻击者可能在知道了所有服务器给出的某些信息之后才决定攻击哪个服务器，因此在构造仿真器时不能确定哪些服务器被攻击，因而必须让所有的服务器的内部状态保持不变，这样是无法构造出仿真器的。也就无法证明分布式签名协议在自适应攻击下的安全性。

文[16]解决了这个问题。它通过及时仔细地清除服务器的内部信息来消除构造仿真器的困难。为了构造仿真器，对所有内部信息都要及时清除，如果被清除的信息在以后协议运行中还要使用，就使用零知识证明进行补偿。其给出的方法对主动安全也适用。按照这个思路，文[17]给出了第一个可抵抗自适应攻击的主动 RAS 系统。文[18]实现了主动 Cramer-Shoup 密码方案，该方案可以在并发执行的条件下对抗自适应的选择性密文攻击。

结束语 主动安全是一种提高安全性的有效技术。但由于效率的原因，目前实际使用主动安全技术的系统并不多。已知的主动安全应用只有3个。文[19]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，可以对外提供主动 DSS 签名服务。文[20]只利用主动共享和主动通信，解决了信息安全分布存贮与检索问题。文[21]设计和实现了产生密码学意义上安全的伪随机数的主动协议。

主动安全中还有很多可以进一步研究的问题。在效率方面，有如何提高以下算法效率的问题：主动秘密共享、主动通信、主动 RSA 签名；在实用性方面，有如何将主动 DSS 签名

与主动 RSA 签名结合起来提供的问题，有入侵检测发现攻击后如何通知主动系统的问题，以及如何在被攻击者控制的机器上维护主动系统代码本身的安全性等问题。

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, LNCS, 1995, 963: 339～352

3 Canetti R, et al. Proactive Security: Long-term protection against break-ins. RSA CryptoBytes, 1997, 3: 1～8

4 Shamir A. How to Share a Secret. Communications of the ACM, 1979, 22: 612～613

5 Gennaro R, et al. Secure Distributed Key Generation for Discrete-Log Based Cryptosystems. Eurocrypt '99, pp. 295～310

6 Boneh D, Franklin M. Efficient Generation of Shared RSA Keys. Crypto 97, pp. 425～439

7 Frankel Y, MacKenzie P D, Yung M. Robust Efficient Distributed RSA-Key Generation. PODC 1998. 320

8 Canetti R, Halevi S, Herzberg A. Maintaining Authenticated Communication in the Presence of Break-ins. Journal of Cryptology, 2000, 13: 61～105

9 Gennaro R, et al. Robust Threshold DSS Signatures. Eurocrypt '96, LNCS 1070. 354～371

10 Herzberg A, et al. Proactive Public Key and Signature Systems. ACM Conference on Computers and Communication Security, 1997

11 Frankel Y, Gemmall P, MacKenzie P, Yung M. Proactive RSA. Crypto'97, 1997. 440～454

12 Rabin T. A Simplified Approach to Threshold and Proactive RSA. Crypto'98, LNCS vol. 1462, Springer-Verlag, 1998. 89～104

13 Gennaro R, Jarecki S, Krawczyk H, Rabin T. Robust and Efficient Sharing of RSA Functions. Journal of Cryptology, 2000, 13(2): 273～300

14 Frankel Y, MacKenzie P, Yung M. Adaptively-Secure Optimal-Resilience Proactive RSA. ASIACRYPT, 1999. 180～194

15 Frankel Y, Gemmell P, MacKenzie P D, Yung M. Optimal-resilience proactive public-key cryptosystems. IEEE Symposium on Foundations of Computer Science, 1997. 384～393

16 Frankel Y, MacKenzie P, Yung M. Adaptively-Secure Distributed Public-Key Systems. European Symposium on Algorithms, 1999. 4～27

17 Frankel Y, MacKenzie P, Yung M. Adaptively-Secure Optimal-Resilience Proactive RSA. ASIACRYPT'99. 180～194

18 Lysyanskaya A. Efficient Threshold and Proactive Cryptography Secure Against the Adaptive Adversary. Manuscript, 1999

19 Barak B, Herzberg A, Naor D, Shai E. The Proactive Security Toolkit and Applications. ACM Conference on Computer and Communications Security 1999. 18～27

20 Garay J A, Gennaro R, Jutla C, Rabin T. Secure Distributed Storage and Retrieval. Theoretical Computer Science, 2000, 243(1-2): 363～389

21 Chow C-S, Herzberg A. Network Randomization Protocol: A Proactive Pseudo-Random Generator. In: Proc. of the Fifth USENIX UNIX Security Symposium, Salt Lake City, Utah, June 1995

（上接第12页）

户身份退出网络以来本地资源状态变化情况，更新 LocalResourceStatusList；将当前连接的 super-peer 设为 CachedSuperPeerList 的头；根据 super-peer 的要求及自身缓存，报告资源信息（完整资源信息或仅是资源变化信息）；发布 PeerInfo Advertisement。在需要时执行的任务有：1. 发送查询（需在 QueryMessage 中设定初始 TTL，ResultsNumber，自己的 PeerID，QueryID），接收回应；2. 发送 AliveMessage；3. 为 super-peer 提供负载值；4. 本地资源发生变化时，向 super-peer 报告；5. 被要求断开时，转“对等体启动”。

在退出客户身份时不执行任何处理。

Super-peer 身份运行：基本任务包括接受客户连接，更新客户资源，记录客户连接状态，处理查询消息和结果消息。另外需周期性执行以下任务：1. 根据 AliveClientList，调用 CleanUpSelfClient；2. 调用 DeleteOtherClient；3. 发现并连接最佳的 super-peer，成功时退出 super-peer 身份，转“客户身份运行”；4. 计算负载，高负载时要求性能最高的客户断开，并为其执行 CleanUpSelfClient；5. 发布 PeerInfo Advertisement。

在退出 super-peer 身份退出前必须执行 CleanUpSelfClient 和 DeleteOtherClient。

结束语 高效的资源搜索方法对 P2P 网络扩展性和可用性的提高有着重要影响，具有 super-peer 概念的搜索方法是提高扩展性和可用性的可行途径之一。本文基于 JXTA 技术设计了一种 super-peer 搜索方法。由于基于 JXTA，使其能适应更广的应用环境。在设计中，以性能高的对等体发生网络连接断开和系统崩溃的可能性较低为假设，使用它们来共同承担搜索任务；通过客户及 super-peer 的信息缓存，有效减少对等体间消息发送数量；利用 super-peer 负载计算，合理控制网络中 super-peer 的数目。

为使本算法更加完善，在未来的工作中仍有一些问题需要解决，其中主要包括如何使 super-peer 与客户的组织更符合物理网络拓扑逻辑，如何提高 super-peer 间信息缓存复制的有效性。

参 考 文 献

1 Yang B, Garcia-Molina H. Comparing hybrid peer-to-peer systems. In: Proc. of the 27th Intl. Conf. on Very Large Databases, Sep. 2001

2 Ratnasamy S, Francis P, Handley M, Karp R, Shenker S. A scalable content-addressable network. In: Proc. ACM SIGCOMM, Aug. 2001

3 Yang B, Garcia-Molina H. Designing a Super-Peer Network: [Technical report]. Stanford University, 2002. Available at: http://www-db.stanford.edu/~byang/pubs/superpeer.pdf

4 JXTA v1.0 Protocols Specification. Available at: http://spec.jxta.org/v1.0/docbook/JXTAProtocols.html

5 Ripeanu M. Peer-to-Peer Architecture Case Study: Gnutella Network: [Technical report]. University of Chicago. Available at: http://www.cs.uchicago.edu/files/tr_authentic/TR-2001-26.pdf

6 Saroiu S, Gummadi P, Gribble S. A measurement study of peer-to-peer file sharing systems. In: Proc. of the Multimedia Computing and Networking, January 2002

7 Yang B, Garcia-Molina H. Improving efficiency of peer-to-peer search. In: Proc. of the 28th Intl. Conf. on Distributed Computing Systems, July 2002