固定移动融合网络中基于资源挑战的垃圾语音防范方法

何光宇[1),2)] 闻英友[2)] 赵 宏[1)]
[1)](东北大学信息科学与工程学院 沈阳 110004)
[2)](东软集团研究院 沈阳 110179)

摘 要 以经典单向函数为基础,设计并验证了一种用于垃圾语音(Spam over Internet Telephony,SPIT)防范的方法.该方法利用资源挑战机制,要求垃圾语音的发送者消耗大量系统资源来破解谜题才可以发送语言会话请求.谜题设计算法避免了相关研究存在的缺陷,使得方法更加安全、可靠;谜题破解算法要求发送者对CPU与Memory进行双重消耗,从而缩小不同配置终端在破解过程中的消耗差距.对方法进行仿真实验,分析结果表明方法具有很好的有效性与适用性.

关键词 SPIT;单向函数;资源消耗;谜题设计;谜题破解
中图法分类号 TP393 **DOI号**:10.3724/SP.J.1016.2012.00038

SPIT Prevention Method Based on Resource Challenge for FMC Network

HE Guang-Yu[1),2)] WEN Ying-You[2)] ZHAO Hong[1)]
[1)](*School of Information Science & Engineering, Northeastern University, Shenyang* 110004)
[2)](*NeuSoft Research, Shenyang* 110179)

Abstract Using classic One-Way Function as the basis, a SPIT prevention method is proposed and verified in this paper. Adopting the resource challenge mechanism, spitter is forced to consume huge system resource to solve the puzzle in order to send the session invite. The puzzle design algorithm avoids deficiencies of relevant research, makes the method more security and reliable. The puzzle solving algorithm consumes both the CPU and Memory of spitter, then reduces the consumption gap of puzzle solving during terminals with different configuration. Experiment and analysis show the efficiency and applicability of the method.

Keywords SPIT; one-way function; resource consumption; puzzle design; puzzle solving

1 引 言

在传统IP网络环境下,Spam被概括地定义为非预期的信息通信.Email Spam即垃圾电子邮件,作为典型代表给传统互联网带来了严重的安全威胁和经济损失[1].随着IP应用的普及以及基于IP多媒体子系统(IMS)的固定移动融合(Fixed Mobile Convergence,FMC)网络的发展,基于SIP[2]协议的VoIP(Voice over IP)无疑将成为全IP融合网络环境下最为重要的业务应用.然而SIP协议在设计之初缺乏完善的安全机制[3-4],导致垃圾语音信息(Spam over Internet Telephony,SPIT)作为一种非预期的语音发送行为,成为全IP语音业务发展中重要的安全威胁.据相关国际标准化组织及相关研究机构的预测,电话营销、非法信息以及恶意骚扰等类型的SPIT将成为未来VoIP业务中最为棘手的安全隐患[5].

收稿日期:2010-03-08;最终修改稿收到日期:2011-12-13.本课题得到国家"九七三"重点基础研究发展规划项目"云应用软件架构技术研究"(2010CB735907)、核高基重大专项"网络化应用支撑工具"(2011ZX01043-001-001)资助.**何光宇**,男,1980年生,博士,主要研究方向为网络与信息安全.E-mail:hegy@neusoft.com.**闻英友**,男,1974年生,博士,研究方向为网络安全、移动通信技术.**赵 宏**,男,1954年生,教授,博士生导师,主要研究领域为网络与信息安全、分布式多媒体、网络管理.

与 Email Spam 不同，SPIT 具有实时性和直接性的特点，因此具有更大的威胁. 垃圾语音以音频作为内容承载手段，使得较为成熟的基于内容过滤的 Email Spam 检测方法难以实施. 融合网络的开放与互通也使得简单的黑白名单机制不再适用. 针对 SPIT 检测与防范的重点和难点，相关组织、学者在综合检测与防范模型[6-10]、实体认证方法[15-16]、行为分析方法[17-19]、信任评判方法[20-22]、音频检测方法[23-24]、资源消耗方法[27-29]等方面展开研究，并进行仿真验证[25-26]与分析评估[11-14]，其中资源消耗方法能够对潜在的 SPIT 发送者进行资源挑战，发送者需要消耗大量系统资源与时间才能应对挑战，对以高频、广域为目标的 SPIT 攻击来说是一种行之有效的防范手段. 然而目前研究成果也存在一些问题与不足. Dwork 等人[27]与 Banerjee 等人[28]提出的方法仅对发送者的 CPU 资源进行消耗，由于不同终端的 CPU 频率存在很大差距，因此该方法的性能波动较大. Abadi 等人[29]提出的方法主要对发送者的内存资源进行消耗，由于不同终端的内存频率差距相对较小，因此规避了性能波动的问题. 然而以上 3 种方法的挑战空间有限，SPIT 发送者可以对挑战的计算结果进行缓存，也可以预先计算出不同挑战对应的结果，从而轻松应对挑战.

本文提出一个基于资源挑战的 SPIT 防范模型，并对模型做形式化的描述. 针对模型中关键算法进行重构与改进，克服了现有研究成果的问题与不足. 第 2 节给出基于资源挑战的 SPIT 防范模型；第 3 节描述模型的具体算法；第 4 节通过实验分析模型及算法的性能与效用；第 5 节对全文进行总结.

2　基于资源挑战的 SPIT 防范模型

2.1　体系结构与流程逻辑

为了实现资源挑战机制，需要对传统 VoIP 环境进行改造，增加一些功能组件，如图 1 所示.

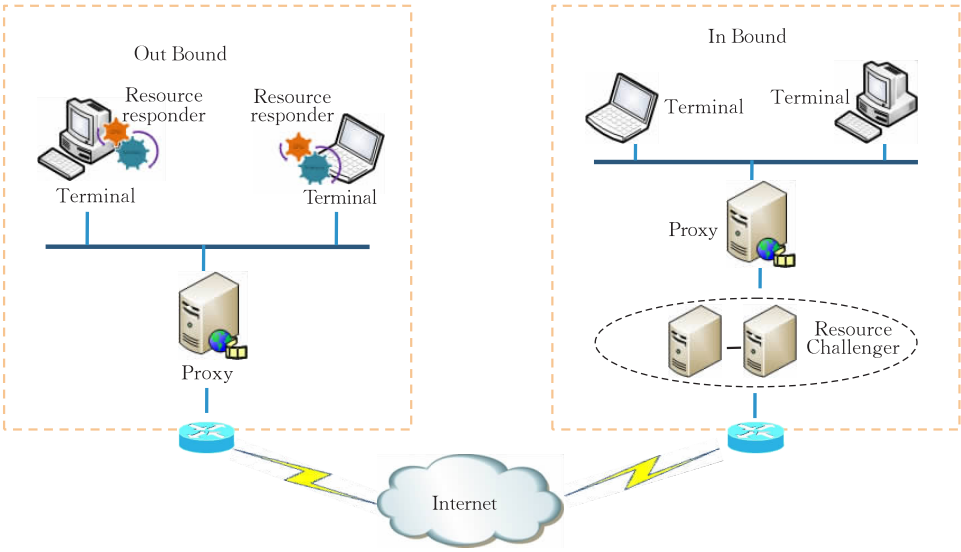


图 1　体系结构

改造后的体系结构增加了 Resource Challenger 与 Resource Responder，二者为逻辑功能组件，可以部署为独立的物理实体，也可以集成到 Proxy 与 Teminal 中，分别实现对 caller 系统资源的挑战及对 callee 挑战的应答. 图 2 给出了改造后基于 SIP 协议的 VoIP 会话流程.

(1) 每次 Out Bound 的 Terminal 发来 Invite 请求，In Bound 的 Resource Challenger 都会设计一个迷题，作为 Challenge 发送给 caller；

(2) caller 利用功能组件 Resource Responder 来破解迷题，并将答案作为 Response 返回给 Resource Challenger；

(3) Resource Challenger 对答案进行验证，如果答案错误则发送 Bye 拒绝请求，如果答案正确则向 callee 转发 Invite 请求；

(4) callee 判断是否接受请求，如果拒绝则发送 Bye 请求，如果接受则发送 OK 应答；

(5) 如果 callee 返回 OK 应答，则 caller 发送 ACK 请求，进而双方建立媒体会话；

(6) 会话完成后，callee 发送 Bye 请求，caller 返回 OK 应答，会话结束.

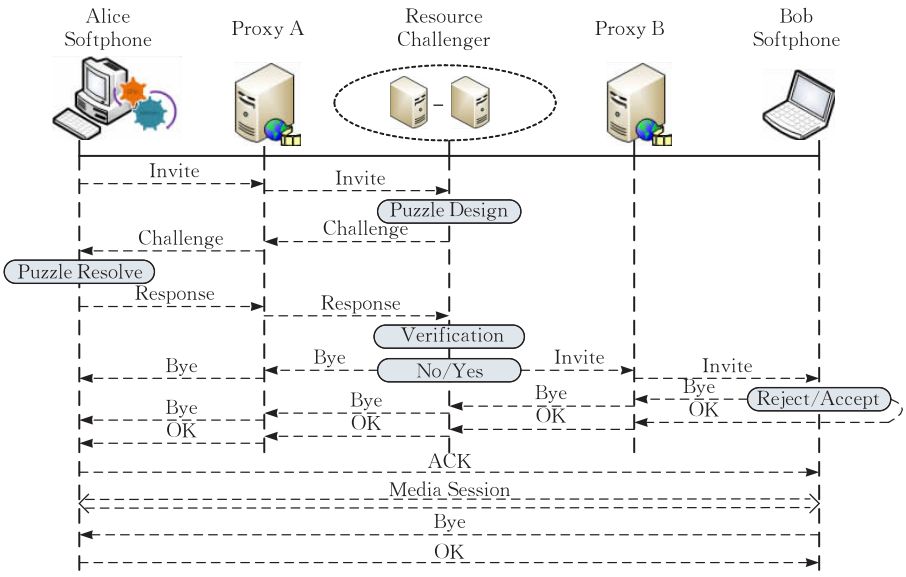


图 2 流程逻辑

2.2 形式化描述

为了准确、规范地展现模型，支撑模型完备、无二义性的实现，本小节给出形式化的描述. 由于 SIP 会话基于事务，因此对 SIP 事务有限状态机(SIP Transaction Finite Automation，ST-FA)进行改进，使其能够支持 SPIT 防范的需要. ST-FA 按照处理的请求是否为 INVITE 分为 INVITE 类型与 Non-INVITE 类型，本文阐述的 SPIT 防范方法作用于连接建立阶段，因此下面给出改进后的客户端(UAC)与服务器端(UAS)的 INVITE 类型的 ST-FA.

定义 1. 一个用于 SPIT 防范的客户端 ST-FA 是一个五元组$(Q_c,\Sigma_c,\delta_c,q_c,F_c)$，其中：

Q_c是一个有穷状态的集合，用来表示客户端会话建立过程中的各个状态，定义为{Calling，Proceeding，Computing，Pending，Completed，Terminated}；

Σ_c是导致状态变迁的各种信息的集合，定义为{Request，Response}；

$\delta_c: Q_c \times \Sigma_c \rightarrow Q_c$是状态转换函数，图 3 给出了在任一状态$q_{\text{pre}} \in Q_c$时，发送请求$req \in \Sigma_c$或收到应答$resp \in \Sigma_c$后应该进入的下一状态$q_{\text{post}} \in \Sigma_c$；

$q_c \in Q$是起始状态，即发起会话请求时的状态 Calling；

$F \in Q$是接受状态，即会话结束后的状态 Terminated.

定义 2. 一个用于 SPIT 防范的服务器端 ST-FA 是一个五元组$(Q_s,\Sigma_s,\delta_s,q_s,F_s)$，其中：

Q_s是一个有穷状态的集合，用来表示服务器端会话建立过程中的各个状态，定义为{Proceeding，Designing，Pending，Verifying，Completed，Confirmed，Terminated}；

Σ_s是导致状态变迁的各种信息的集合，定义为{Request，Response}；

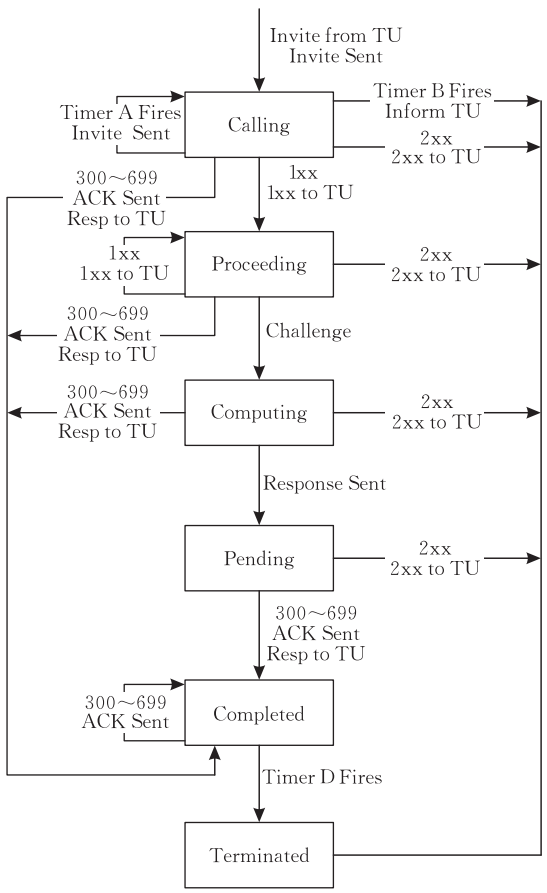


图 3 客户端 ST-FA

$\delta_s: Q_s \times \Sigma_s \to Q_s$ 是状态转换函数，图 4 给出了在任一状态 $q_{pre} \in Q_s$ 时，发送请求 $req \in \Sigma_s$ 或收到应答 $resp \in \Sigma_s$ 后应该进入的下一状态 $q_{post} \in \Sigma_s$；$q_s \in Q$ 是起始状态，即收到会话请求时的状态 Proceeding；

$F_s \in Q$ 是接受状态，即会话结束后的状态 Terminated.

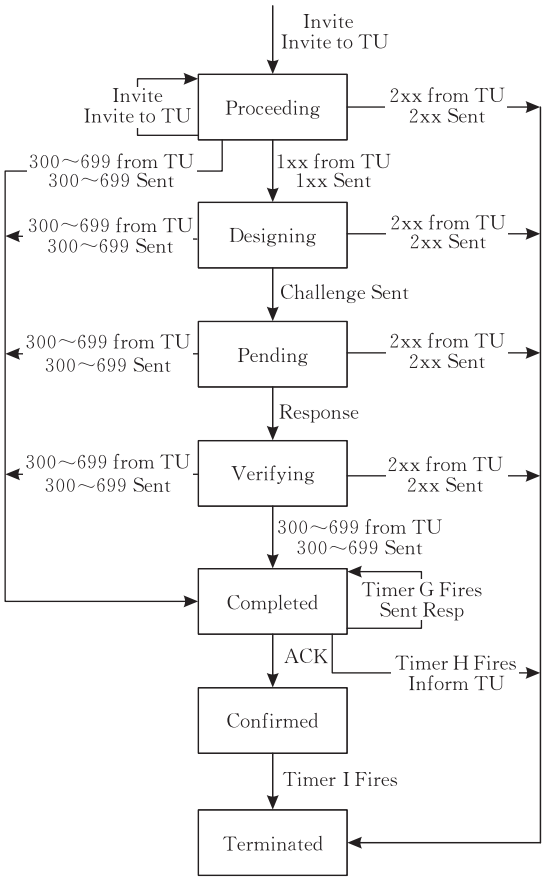


图 4　服务器端 ST-FA

3　模型中关键算法

3.1　算法设计原则

基于资源挑战的方法本质上是通过对 caller 系统资源的消耗，利用技术手段从经济性层面实现对 SPIT 攻击的防范. 为了解决现有研究成果存在的问题，提高方法的鲁棒性与实用性，在设计时需要遵循以下原则：

(1) caller 需要耗费大量系统资源来破解 callee 设计的迷题才可以成功发送语音请求，而 callee 只需消耗极少资源便可设计谜题并验证答案的正确性；

(2) 迷题需要对 caller 的 CPU 与 Memory 进行双重消耗，最大程度缩小配置不同的终端破解迷题时的差距；

(3) 实现一次一问，避免结果被多次使用；

(4) 随机挑选迷题空间，使得缓存前续计算结果与预先计算不可行.

本节后续给出迷题设计及破解的算法，并讨论给出的算法如何满足设计原则的要求.

3.2　迷题设计算法

迷题设计的核心思想是 Resource Challenger 利用单向函数的不可逆性迫使 Resource Responder 通过穷举的方法来计算其逆函数，从而达到资源消耗的目的.

典型的单向函数有大素数因数乘积与大指数求幂模运算，为了与主流密码系统相一致，并控制自变量与值域的范围，本文选择后者.

$$y = f(x) = a^x \bmod n \tag{1}$$

其中 a 称为底数，x 称为指数，n 称为模数，y 称为余数. 此时 Resource Challenger 的挑战是一个三元组 $Challenge = \{a, n, y\}$. Resource Responder 需要遍历 $x \in Domain(x) = \{x \mid 1 \le x \le n-1\}$，寻找 x^* 使得 $y = f(x^*)$. 在该算法中，设离散型随机变量 M 表示 Resource Responder 找到 x^* 需要计算的次数，其概率分布为 $P\{M = m_j = j\} = p_j = \dfrac{1}{n-1}$，$j = 1, 2, \cdots, n-1$. 于是 M 的数学期望和方差分别为

$$E(M) = \sum_{j=1}^{n-1} m_j p_j = \frac{n}{2} \tag{2}$$

$$D(M) = E(M^2) - (E(M))^2 = \frac{n^2 - 2n}{12} \tag{3}$$

可见 Resource Responder 找到特定的 x^* 的偶然性比较大，即不同挑战对 Resource Responder 的资源消耗不够稳定. 因此对该算法进行改进，利用多次遍历来降低单次遍历存在的高偶然性问题.

$$x_{i+1} = f(x_i) \tag{4}$$

改进后的算法中，Resource Challenger 首先选定 x_0，然后利用式(4)重复执行 k 轮得到 x_k. 为了针对每个 x_{i+1} 可以找到唯一的 x_i 与之对应，式(1)中模数选择素数记为 p，a 选择群 $G = \langle Z_p^*, \times \rangle$ 的本原根，$Domain(x) = \{x \mid 1 \le x \le p-1\}$. 此时挑战是一个四元组 $Challenge = \{a, p, k, x_k\}$. Resource Responder 收到 Challenge 后，首先遍历 $x_{k-1} \in Domain(x)$，找到 x_{k-1}^* 使得 $x_k = f(x_{k-1}^*)$. 如此重复遍历 k 轮以找到 $x_0^* = x_0$.

设离散型随机变量 $M_1, M_2, \cdots, M_k$ 分别表示第 k 轮找到 x_{k-1}^* 需要计算的次数，M 表示表示找到 x_0^* 需要计算的总次数，$M = \sum_{i=1}^{k} M_i$. p 选择接近 $\dfrac{n}{k}$ 的素数，$M_1, M_2, \cdots, M_k$ 相互独立，于是 M 的数学期望和方差分别为

$$E(M)=\sum_{i=1}^{k}E(M_i)=k\cdot\frac{n/k}{2}=\frac{n}{2} \tag{5}$$

$$D(M)=\sum_{i=1}^{k}D(M_i)=k\cdot\frac{(n/k)^2+2(n/k)}{12}=\frac{n^2+2nk}{12k} \tag{6}$$

随着 k 值的增加，该算法可以有效避免单次遍历的高偶然性，然而由于式(1)引入了模运算，使得 x_0 到 x_k 序列中容易出现循环，即 $x_i=x_{i+c}$. Resource Responder 在逐轮遍历中发现循环后，可以在 c 轮内找到 $x_0^*=x_0$，得到破解捷径从而逃避大量的资源消耗. 解决的方法是在谜题设计的每一轮加入该轮的信息 i，从而保证 x_0 到 x_k 序列中不存在周期性循环.

$$x_{i+1}=(f(x_i)\ Xor\ i)\bmod p \tag{7}$$

Resource Challenger 在每次计算 $f(x_i)$ 后将其与 i 做异或操作，从而避免了周期性循环. 然而异或操作的结果可能超出 $Domain(x)$，因此还要与 p 做模运算. 模运算的结果可能为 0，因此令 $f(0)=0$. 异或操作还可能破坏 x_{i+1} 与 x_i 的一一映射关系，因此需要对 x_0 到 x_k 序列求和得到 sum. 此时挑战是一个五元组 $Challenge=\{a,p,k,x_k,sum\}$. Resource Responder 收到 Challenge 后，每轮遍历都要选择 $f(x_i)\ Xor\ i$ 的值为 x_{i+1} 或 $x_{i+1}+p$. 进行 k 轮遍历计算得到 x_0^*，并用 x_0^* 到 x_k^* 的和 sum^* 与 sum 进行比较，如果相等则计算结束，否则选择其它组合继续计算，直到 sum^* 与 sum 相等.

算法目前已经具有一定的实用性，但在 k 确定的情况下，$Challenge=\{a,p,k,x_k,sum\}$ 中 x_k 可取不同值的数量为 p. 因此 Resource Responder 可以缓存以往计算得到的 x_k 对应的 x_0，也可以在闲暇的时间预先计算出每个 x_k 对应的 x_0，从而轻松的应对挑战.

针对这个问题的解决方法是，在每次设计 Challenge 时，加入临时信息从而使得预先计算与缓存结果不再有效.

$$x_{i+1}=\begin{cases}(f(x_i)\ Xor\ i)\bmod p, & i\text{ 为偶数}\\ (f(x_i)\ Xor\ i\ Xor\ token)\bmod p, & i\text{ 为奇数}\end{cases} \tag{8}$$

Resource Challenger 每次挑战前，随机选择一个变量 $token\in Domain(x)$. 当轮次 i 为奇数时，计算 $f(x_i)$ 并与 i 做异或操作，再与 p 做模运算；当轮次 i 为偶数时，计算 $f(x_i)$ 并依次与 i 和 $token$ 做异或操作，再与 p 做模运算. 重复执行 k 次得到 x_k，此时挑战是一个六元组 $Challenge=\{a,p,k,x_k,token,sum\}$. Resource Responder 如何破解该谜题，在下面给出详细阐述.

3.3 迷题破解算法

Resource Responder 收到 $Challenge=\{a,p,k,x_k,token,sum\}$ 后采用递归的方法进行破解. 递归函数的形式参数为 3 元组 $\{l,x,s\}$，其中 l 代表当前的轮次，x 代表当前轮次 x_l 的值，s 代表到当前轮次为止已得到的 x_l 的和，即 $s=\sum_{j=l}^{k}x_j$.

递归初始时，$l=k$，$x=x_k$，$s=x_k$. 如果 $l=0$ 且 $s=sum$，则递归结束，返回 $x_0=x$. 如果 $l=0$ 且 $s\neq sum$，则返回上层函数.

如果 $l\neq 0$，则令 $x'=x$，并执行如下计算：

$$x''=\begin{cases}x'\ Xor\ l-1, & i\text{ 为奇数}\\ x'\ Xor\ l-1\ Xor\ token, & i\text{ 为偶数}\end{cases} \tag{9}$$

遍历 $Domain(x)$，如果找到 x^* 使得 $x''=f(x^*)$. 递归调用该函数，此时形式参数赋值为 3 元组 $\{l-1,x^*,s+x^*\}$.

如果没有找到 x^*，并且 $x\leq N-p$，则递归调用该函数，此时形式参数赋值为 3 元组 $\{l,x+p,s\}$.

利用该递归函数进行谜题破解，至少要进行 l 轮遍历，每轮遍历都需要依次取 $x\in Domain(x)$，计算 $f(x)$. 因此理性的 Resource Responder 会首先进行一次全遍历，存储 $x\rightarrow f(x)$ 的映射关系，在每个轮次的遍历中只需要反向查询便可找到 x'' 对应的 x^*.

反向查询实质上是对 Memory 的随机访问，随机访问的效率取决于 Memory 的时钟频率，较之 CPU 的频率(通常 333MHz-n×3 GHz，几十倍)，Memory 的频率(通常 333 MHz～800 MHz，几倍)在不同终端之间差别较小，因此采取对 Resource Responder 的 Memory 进行消耗，可以在很大程度上缩小不同配置的 Resource Responder 破解谜题所消耗的资源的差距.

4 实验与分析

为了验证本文提出的资源挑战方法，搭建仿真环境进行实验与分析. 由于关注的重点是该方法对 Resource Responder 的 CPU 与 Memory 的消耗情况以及该方法面对不同类型与配置的 Resource Responder 所表现出的适用性，实验选择 3 台不同类型与配置的计算机作为 Resource Responder，观察它们在解决谜题过程中的表现，从而验证本文提出的方法的有效性. 表 1 给出作为 Resource Responder 的 3 台计算机的配置信息.

表 1　Resource Responder 配置信息

Machine	Model	CPU Frequency	Memory Size	Memory Frequency
A	HP 2210b	Intel Core2 Duo 1.80 GHz	1024 MB	DDR2 667
B	Lenovo KM400	AMD Athlon 1.83 GHz	512 MB	DDR 333
C	Compaq B2000	Intel Pentium Duo 1.50 GHz	512 MB	DDR 333

表 2 给出了对于不同 $Challenge=\{a,p,k,x_k,token,sum\}$，Resource Responder 进行破解的资源消耗情况. 其中大素数 $p=9973$，其本原根 $a=11$，$token=8888$. Resource Challenger 在设计谜题时，x_0 选择 1234，轮次 k 从 100 到 1500 以 100 等差递增. C/R 分别代表用 A、B、C 配置的计算机进行谜题设计与破解所需的时间，单位为 s. 文献[29]实验用到 Server、Desktop、Laptop 配置差距与本文实验用到 A、C、B 计算机配置差距相当，文献[29]实验结果表现出性能差异为 18∶24∶42，本文实验结果表现出性能差异为 20∶23∶27，可见本文算法性能明显优于文献[29]的算法性能.

表 2　谜题设计与破解的资源消耗

Challenge	C/R (s)		
	A	B	C
{11,9973,100,8895,8888,450402}	0.03/2.46	0.02/4.69	0.01/3.99
{11,9973,200,3209,8888,886802}	0.05/3.89	0.04/6.87	0.02/5.85
{11,9973,300,1561,8888,1370925}	0.06/6.45	0.05/9.36	0.02/8.43
{11,9973,400,3056,8888,1812027}	0.09/9.73	0.05/14.61	0.03/12.51
{11,9973,500,2745,8888,2258233}	0.11/14.75	0.05/20.05	0.04/17.67
{11,9973,600,860,8888,2646259}	0.13/19.96	0.06/27.12	0.04/23.58
{11,9973,700,6278,8888,3045536}	0.14/26.63	0.06/35.87	0.04/31.16
{11,9973,800,5079,8888,3441242}	0.15/33.11	0.08/44.25	0.05/39.47
{11,9973,900,921,8888,3898513}	0.16/41.07	0.08/55.40	0.06/49.12
{11,9973,1000,325,8888,4325947}	0.17/51.52	0.08/68.56	0.06/59.76
{11,9973,1100,747,8888,4743976}	0.17/61.60	0.09/81.18	0.06/71.39
{11,9973,1200,2586,8888,5229609}	0.17/72.96	0.09/96.79	0.07/84.57
{11,9973,1300,1395,8888,5655506}	0.18/84.79	0.10/112.45	0.07/98.26
{11,9973,1400,6302,8888,6025271}	0.18/98.21	0.11/129.37	0.08/113.78
{11,9973,1500,1750,8888,6441642}	0.19/112.10	0.11/148.72	0.08/130.11

图 5 给出了不同配置的 Resource Responder 破解轮次递增的 *Challenge* 时所需要的时间，该图反映出本文提出方法的一些特性：(1) 随着轮次的增加，破解谜题的时间也随之增加；(2) 破解谜题所需时间的增加从平缓趋向剧烈；(3) Resource Responder 破解谜题所需时间的差异要小于其配置的差异. 这些特性证明了本文提出的谜题设计与破解方法的有效性以及适用性.

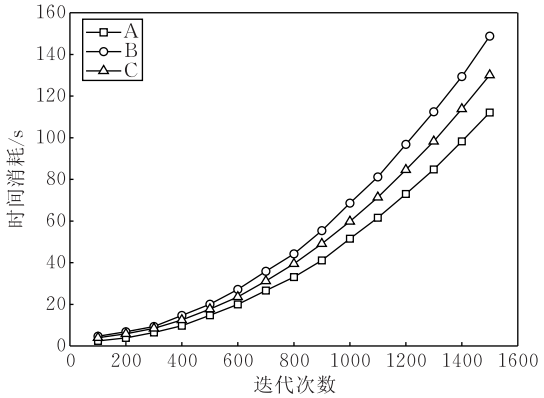


图 5　破解的时间消耗

图 6 和图 7 分别给出不同配置的 Resource Responder 破解轮次递增的 Challenge 时 CPU 与内存的消耗.

由图可知，破解谜题过程几乎占用 Resource Responder 的全部 CPU 资源，符合资源消耗的思想；并且对 Memory 的使用基本恒定在 23 MB，这说明进行资源挑战不要求 Resource Responder 具有大量内存，该方法可以推广至 PDA、智能手机等终端设备.

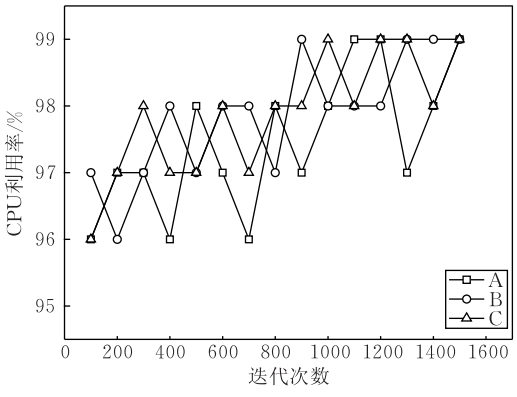


图 6　破解的 CPU 消耗

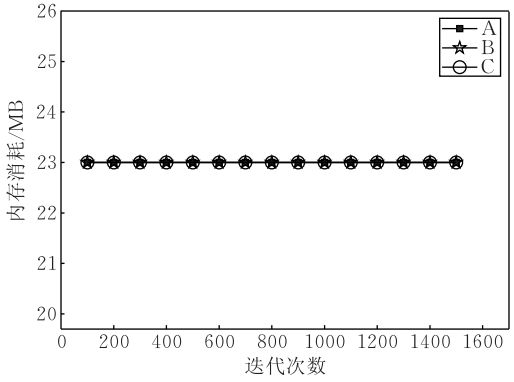


图 7 破解的存储器消耗

5 结束语

本文提出的基于资源挑战的 SPIT 防范方法，克服了相关研究中存在的消耗抖动较大，存在破解捷径、重复利用计算结果、预先计算等诸多缺陷，并且实现了对 Resource Responder 的 CPU 与 Memory 的双重消耗. 仿真实验的分析结果表明，该方法可以根据实际需要设计出任意难度的谜题，并可以扩展到 PDA、智能手机等终端设备，具有很好的有效性与适用性.

参 考 文 献

[1] Evett D. Spam Statistics 2006. http://spam-filter-review.toptenreviews.com/spam-statistics.html

[2] Rosenberg J, Schulzrinne H, Camanilo G. SIP: Session initiation protocol. Internet RFC 3261, 2002

[3] Arkko J, Torv Inen V, Camar Illo G. Security mechanism agreement for the session initiation protocol (SIP). Internet RFC3329, 2003

[4] Stefano S, Luca V, Donald P. SIP security issues: The SIP authentication procedure and its processing load. IEEE Network, 2002, 16(6): 38-44

[5] Schulzrinne H. Internet telephony-challenges and open issues. IETF-draft- schulzrinne-iptel-challenges-00. June 1, 2002

[6] Croft N J, Olivier M S. A model for spam prevention in voice over IP networks using anonymous verifying authorities//Proceedings of the 5th Annual Information Security South Africa Conference(ISSA2005). Sandton, South Africa, 2005

[7] Schlegel R, Niccolini S, Tartarelli S et al. Spam over Internet telephony (SPIT) prevention framework//Proceedings of the Global Telecommunications Conference (GLOBECOM'06). California, USA, 2006: 1-6

[8] Quittek J, Niccolini S, Tartarelli S et al. On spam over Internet telephony (SPIT) prevention. IEEE Communication Magazine, 2008, 46(8): 80-86

[9] Gritzalis D, Mallios Y. A SIP-oriented SPIT management framework. Computers & Security, 2008, 27(5-6): 136-153

[10] Dritsas S, Dritsou V, Tsoumas B et al. OntoSPIT: SPIT management through ontologies. Computer Communications, 2009, 32(1): 203-212

[11] Park S Y, Kim J T, Kang S G. Analysis of applicability of traditional spam regulations to VoIP spam//Proceedings of the International Conference on Advanced Communication Technology (ICACT 2006). Phoenix Park, Korea, 2006: 1215-1217

[12] Rosenberg J, Jennings C, Peterson J. The session initiation protocol (SIP) and spam. draft-ietf-sipping-spam-03. txt, IETF Draft, April 25, 2007

[13] Dritsas S, Mallios J, Theoharidou M et al. Threat analysis of the session initiation protocol regarding spam//Proceedings of the 3rd IEEE International Workshop on Information Assurance (WIA 2007). Louisiana, USA: IEEE Press, 2007: 426-433

[14] Marias G F, Dritsas S, Theoharidou M et al. SIP vulnerabilities and anti-SPIT mechanisms assessment//Proceedings of the 16th IEEE International Conference on Computer Communications and Networks (ICCCN'07). USA, 2007: 597-604

[15] Schwartz D, Sterman B, Katz E et al. Spam for internet telephony (SPIT) prevention using the security assertion markup language (SAML). draft-schwartz-sipping-spit-saml-01. txt. IETF Draft, 2006

[16] Cao F, Jennings C. Providing response identity and authentication in IP telephony//Proceedings of the Create-Net Workshop on Security and Quality of Service in Communication Networks. Athens, Greece, 2005

[17] Robert M, Dmitri V. Detection and mitigation of spam in IP telephony networks using signaling protocal analysis//Proceedings of the 2005 IEEE Symposium on Advances in Wired and Wireless Communication. New York: IEEE, 2005: 135-139

[18] Shin D, Ahn J, Shim C. Progressive multi gray-leveling: A voice spam protection algorithm. IEEE Network, 2006, 20(5): 18-24

[19] He Guang-Yu, Wen Ying-You, Zhao Hong. SPIT detection and prevention method based on signal analysis//Proceedings of the IEEE International Conference on Convergence and hybrid Information Technology (ICCIT 2008). Busan, Korea, 2008: 631-638

[20] Kolan P, Dantu R. Socio-technical defense against voice spamming. ACM Transactions on Autonomous and Adaptive Systems, 2007, 2(1): 1-44

[21] Rebahi Y, Sisalem D. SIP service providers and the spam problem//Proceedings of the 2nd Workshop on Voice over IP Security. Washington DC: ACM, 2005: 65-72

[22] He Guang-Yu, Wen Ying-You, Zhao Hong. SPIT detection and prevention method in VoIP environment//Proceedings of

the IEEE 3rd International Conference on Availability, Reliability and Security (ARES 2008). Barcelona, Spain, 2008: 473-478

[23] Pantridge M. VOIP spam counter measures[M. S. dissertation]. Technical University of Denmark, Kongens Lyngby, Denmark, 2006

[24] Quittek J, Niccolini S, Tartarelli S et al. Detecting SPIT calls by checking human communication patterns//Proceedings of the IEEE International Conference on Communications. Glasgow, Scotland, 2007: 1979-1984

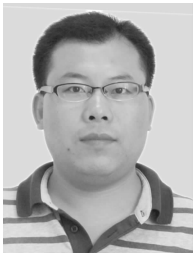
[25] Madhosingh A R. The design of a differentiated session initiation protocol to control VoIP spam. Computer Science Department, Florida State University: Technical Report, 2006

[26] Falomi M, Garroppo R, Niccolini S. Simulation and optimization of SPIT detection frameworks//Proceedings of the IEEE GLOBECOM'07. Washington, DC, USA, 2007

[27] Dwork C, Naor M. Pricing via processing or combatting junk mail//Advances in Cryptology. LNCS 740. 1992: 139-147

[28] Banerjee N, Saklikar S, Saha S. Antivamming trust enforcement in peer to peer VoIP networks//Proceedings of the ACM IWCMC'06. Vancouver, Canada, 2006

[29] Abadi M, Burrows M, Manasse M et al. Moderately hard, memory-bound functions//Proceedings of the 10th Annual Network and Distributed System Security Symposium. California, USA, 2003



HE Guang-Yu, born in 1980, Ph. D.. His research interests include network and information security.

WEN Ying-You, born in 1974, Ph. D.. His research interests include network security, mobile communication.

ZHAO Hong, born in 1954, professor, Ph. D. supervisor. His research interests include network and information security, distribution multimedia, network management.

Background

Network Fusion can make full use of resources, reduce operating costs and enrich the business style, so that the user in either fixed or mobile environment can enjoy the same service. From a global perspective, fixed and mobile convergence network (FMC) is an inevitable worldwide trend. Under such a trend, Session Initiation Protocol (SIP) which has the characteristics of simple realization, good scalability, and strong multimedia-ability has become the main control protocol of application layer. IETF and 3GPP make SIP as the core application layer control protocol of full IP fusion network. Microsoft, Cisco, IBM and other well-known IT Enterprise also try to support SIP in their products. In the numerous applications based on SIP, VoIP (Voice over IP) is one of the most competitive businesses. However, because of the lack of security consideration at the beginning of design, SPIT as an unexpected audio communication behavior will become an important security threat in the development of VoIP. According to the prediction of international organization for standardization and research institutions, SPIT will become the most terrible security hidden trouble in the future of VoIP business.

For SPIT detection and prevention area, relevant organizations, institutions and scholars have launched the research and made some achievements. But in general cannot meet the requirement of SPIT detection and prevention. This research is supported by Core Electronic Device, High General Chip and Basic Software program of China (2011ZX01043-001-001) and National Basic Research Program (973 Program) of China (2010CB735907). The project aimed at the shortcomings and deficiencies of the current research: first of all, establish multidimensional collaborative comprehensive detection and prevention framework at the system architecture layer; secondly, integrated multiple techniques, detected SPIT from multi-angle; finally, according to the security situation which SPIT related, adopt an optimal response measures which has the continuous properties.

This paper discusses the SPIT prevention method based on resource challenges, as the core methods of prevention in the overall architecture, realize the resource consumption for potential SPIT sender, so as to achieve the purpose of preventing. This method makes up for the defects of the current research, the simulation results show that the method is effective and applicable.