第 34 卷 第 6 期 计 算 机 工 程 2008 年 3 月
Vol.34 No.6 Computer Engineering March 2008
·安全技术· 文章编号：1000—3428(2008)06—0140—03 文献标识码：A 中图分类号：TP309

双主体认证协议抵御 DoS 攻击的安全方案

陈建辉[1,2]，徐 涛[1]
(1. 南京航空航天大学信息科学与技术学院，南京 210016；2. 郑州航空工业管理学院计算机科学与应用系，郑州 450015)

摘 要：认证协议存在的一定的拒绝服务(DoS)攻击隐患。该文在 cookie 机制和工作量证明方法思想的基础上，采用弱认证和强认证相结合的方法，提出了双主体认证协议抵御 DoS 攻击的安全方案，给出了方案的实现细节和部分实验数据，分析了该方案的安全性。应用该方案对 Lowe 的 NSPK 协议进行改进，并分析了改进后协议的性能。
关键词：认证协议；拒绝服务；安全协议

Security Solution for Protecting Two-party Authentication Protocols Against Denial of Service Attack

CHEN Jian-hui [1,2], XU Tao [1]
(1. College of Information Science and Technology, Nanjing University of Aeronautics and Astronautics, Nanjing 210016;
2. Department of Computer Science & Application, Zhengzhou Institute of Aeronautical Industry Management, Zhengzhou 450015)

【Abstract】 Authentication protocol has a DoS attack weakness. Based on the idea of cookie solution and proof of work, this paper uses weak authentication and strong authentication, presents the security solution for protecting two-party authentication protocols against DoS attack, gives the detail and some experimental data, analyses the solution's security. It improves the Lowe's NSPK protocol using the solution, and analyses the improved protocol's performance.
【Key words】 authentication protocol; Denial of Service(DoS); security protocol

1 认证协议存在的 DoS 隐患

拒绝服务(Denial of Service, DoS)攻击作为互联网上一种常见的攻击方式，对网络安全和信息安全造成了很大的威胁。根据攻击者的攻击点不同，可以将DoS攻击分为带宽消耗型攻击、存储资源消耗型攻击、计算资源消耗型攻击和系统漏洞型攻击 4 种[1]。

认证协议为保证其安全性，通常要用到复杂的密码计算。在认证协议运行过程中，如果不区分协议参与者的身份即保存协议运行状态信息、进行指数级的解密或签名运算，容易遭到存储资源消耗型和计算资源消耗型DoS攻击[2]。下面以Lowe改进的NSPK协议为例进行说明。

Lowe 改进的 NSPK 协议如下：

(1)Msg1:$A \rightarrow B: \{N_a, A\}_{Kb}$

(2)Msg2:$B \rightarrow A: \{B, N_a, N_b\}_{Ka}$

(3)Msg3:$A \rightarrow B: \{N_b\}_{Kb}$

其中，A, B表示协议的参与者；K_a, K_b表示相应参与者的公钥；K_a^{-1}, K_b^{-1}表示相应参与者的私钥；$\{X\}_K$, $\{X\}_K^{-1}$分别表示对X用公钥K和私钥K^{-1}加密和解密；N_a为现实(nonce)。

分析协议运行过程，可以发现该协议存在一定的DoS隐患[2]。针对该协议，恶意的攻击者可以冒充发起方A向B发送大量虚假数据Msg1，响应方B接收到Msg1 后，不做任何判断，就开始使用自己的私钥对Msg1 进行解密计算，再生成临时值、进行数据级连，并再次用K_a进行加密计算。对于假冒的发起方来说，只需要简单地发送虚假数据就让响应者进行两次复杂的公钥密码计算，该协议很容易受到计算资源消耗型DoS攻击。

同时，在响应者发送Msg2 后，需要在内存中保存N_a, A, N_b等认证状态信息，对于假冒的发起者来说可能不会执行协议的步骤(3)，这样响应者在本次认证等待时间结束之前就不会释放相应内存空间，因此，该协议还存在存储资源消耗型DoS攻击的隐患。

2 抵御 DoS 攻击的安全方案

对于安全协议的抵御DoS攻击问题，有众多研究者提出过不同的方法和解决思路。最初Aura提出了无状态连接的方法，该方法解决了存储资源消耗的DoS攻击问题，但计算资源消耗的压力没有减轻。随后，Karn和Simpson在Photuris协议中提出cookie机制，并被应用在IKE协议中。Cookie机制可以较好地抵御虚假网络地址的DoS攻击，但却无法防范使用真实IP地址的DoS攻击。Dwork和Naor提出了工作量证明(proof of work)的方法以解决垃圾邮件的问题，基于工作量证明的思路，Juels和Brainard提出了client puzzle的方法，用于抵御SYN-flood攻击。后来，Aura和Nikander将工作量证明的思路用于认证协议[3]，但该方法仍需要在响应方保存状态信息。

本文综合 cookie 机制和工作量证明的思想，提出一种新的主动 cookie 的方法。该方法在协议开始时不保存状态信息，将 cookie 和 puzzle 并用，并能根据自身系统资源状况主动调整 puzzle 问题难度，能较好地抵御包括虚假网络地址和真实网络地址在内的存储资源消耗攻击和计算资源消耗攻击。

2.1 设计方案

安全方案采用弱认证和强认证相结合的方法。在认证协议运行的第 1 阶段采用弱认证，协议响应方不保存任何状态

作者简介：陈建辉(1978－)，男，讲师、硕士研究生，主研方向：网络安全，信息集成；徐 涛，教授
收稿日期：2007-03-28 **E-mail：**zzcjh@126.com

信息，不进行复杂的计算，只是向协议请求方发送特定cookie，同时向协议请求方提出puzzle问题要求解决，然后根据请求方返回的cookie和puzzle问题解答对其进行身份真实性确认，此阶段正常完成后才进行后续的强认证。方案框架如下：(1)协议发起方A生成临时值S_a，存储S_a，将自身标识A和S_a级连作为信息 1 发送给响应方B。(2)B收到信息 1 后，根据特定算法生成临时值S_b作为cookie，并根据当前的资源使用情况产生一个puzzle，将cookie S_b、时间戳T和puzzle作为消息 2 发给A。(3)A存储S_b, T，求解puzzle，将自身标识A用A的私钥K_a^{-1}签名，连同临时值S_a, S_b, 时间戳T和问题解答solution作为消息 3 发送给B。

响应方首先在无状态情况下验证S_b，然后验证solution的正确性，通过后再确认发起者A的身份，并在所存储的表中检查A是否存在，如果已存在则说明是重复认证，立即中止协议。以上任何环节验证没有通过则中止协议，全部通过后开始为本次认证分配资源，执行协议后续步骤。方案框架形式表示如下：

Msg1: $A \rightarrow B: A, S_a$

Msg2: $B \rightarrow A: S_b, T$, puzzle

Msg3: $A \rightarrow B: S_a, S_b, T$, solution, $\{A\}_{Ka^{-1}}$

对本设计框架，要想保证其实现过程中的有效性，需满足下列需求：

需求 1：puzzle 问题必须易于产生和验证，但求解困难；

需求 2：puzzle 问题的难度可以动态调整；

需求 3：发起方不能预先求解 puzzle，也不能根据原有的 puzzle 求解结果降低 puzzle 的求解代价；

需求 4：响应方在没有确认发起者付出一定的代价前，不保存与发起方有关的任何状态信息；

需求 5：作为cookie的S_b必须易于产生、易于验证，但不容易伪造；

需求 6：为防止重放Msg3，须在S_b和puzzle问题的产生和验证中加入时间戳。

2.2 方案实现及实验

基于 2.1 节中所述需求，方案框架的进一步描述如下：S_a是以时间为因子生成的随机数，长度为 64 比特，记为$S_a = rand(ta)$。S_b是以S_a, T, 发起方A的网络地址(IPa)和响应方B的本地秘密为因子Hash得到的，记为S_b＝hash(S_a, T, Ipa, local secret)，其中本地秘密保存在内存中，并定时更换。

Puzzle问题的产生机制[4]：设X是响应方以双方标识、时间戳及本地秘密为因子Hash产生的比特串，记为X＝hash(A, B, T, local secret)，其长度为L，$X<m,n>$表示从X第m位到第n位的子串，其中m, n为正整数，且 $0 \le m < n \le L$。令$x_0 = X<1,d>$, $x_1 = <d+1,L>$，则$X = x_0 \| x_1$。令y＝hash(X)，则y＝hash($x_0 \| x_1$)。已知Hash函数，如果给出y,d和x_1，则可以求解出x_0。响应方产生的puzzle问题即对x_0的求解。在协议运行过程中，d的大小可根据系统资源的使用情况动态调整。定义资源使用率为β，由CPU使用率和内存使用率共同决定，则$\beta = c\beta_1 + m\beta_2$，其中$\beta_1$, β_2分别为CPU和内存的当前使用率，c, m为CPU和内存在系统资源使用率计算中的权重。

根据puzzle问题的产生原理，方案框架中的puzzle可表示为y, d, $X<d+1,L>$，solution应为x_0'，如果B验证($x_0' \| x_1$)= hash(A, B, T, local secret)，则说明A求解出正确结果。

Hash函数计算速度快，本身具有单向性，同时设计良好的Hash函数计算结果在较大范围和较长时间内具有唯一性，因此puzzle问题中求解x_0只能使用逐个尝试的暴力破解方法求解，这满足了 2.1 节中的需求 1。在puzzle问题的产生过程中，d的大小可以动态调整，而d的大小直接决定了puzzle问题的难度，其值越大，求解问题需要付出的代价也越大。当系统资源空闲、没有受到DoS攻击时还可以设置d为 0，此时相当于不启用puzzle机制，这一点满足了 2.1 节中的需求 2。此外X的产生是以时间戳和本地秘密为因子Hash得到的，其结果具有时效性和唯一性，基于Hash函数本身的特性，发起方无法预处理puzzle问题，也不能利用以前的求解结果，因此满足 2.1 节中的需求 3。2.1 节中的需求 4 是通过协议响应方在验证Msg3 中的信息通过后才肯分配资源的措施实现的。S_b和puzzle问题中的X都是使用时间戳和本地秘密Hash得到的，Hash函数容易计算，验证方便，本地秘密保证了攻击者无法在短时间内伪造S_b和X，时间戳和本地秘密又可以防止攻击者对Msg3 进行重放，因此方案实现满足了 2.1 节中的需求 5 和需求 6。

根据不同的系统资源利用率，设定相应 puzzle 问题的难度，在 Windows 2000 Server, 2.53 GHz P4, 256 MB 内存的计算机上进行puzzle破解测试(CPU和内存的权值c, m均取0.5，Hash 函数采用 MD5，对每种难度系数的 puzzle 问题均进行 100 次求解，求解时间取其平均值)，测试结果如表 1 和图 1 所示。

表 1　资源使用率、puzzle 问题难度与求解时间关系

资源使用率 β/(%)	难度系数(d)	平均求解时间/ms
<30	0	0.00
30～40	16	70.32
40～50	18	418.13
50～60	20	1 434.35
60～70	22	7 782.83
70～80	24	31 390.00
>80	26	126 307.00

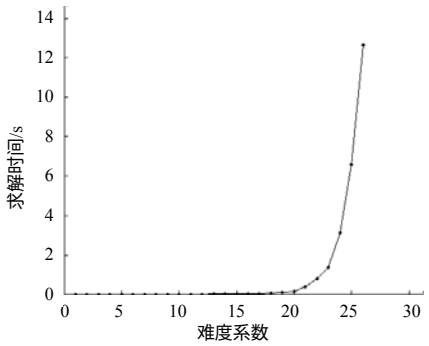

图 1　puzzle 问题难度与求解时间关系

从表 1 中可以分析出，当协议响应方在自身系统资源使用率不断上升时，puzzle 问题难度不断提高，要求协议发起者进行协议认证计算时间急剧增加，从而大大增加了攻击者的攻击难度。从图 1 中可以看出，puzzle 问题难度和求解时间呈指数关系变化，这样，根据响应者系统资源使用情况设定 puzzle 问题难度，可以对 DoS 攻击及时作出有效的反应，降低系统资源压力。

2.3 安全性分析

2.3.1 抗 DoS 攻击

方案中，响应方在第 1 次收到认证请求时，快速生成cookie S_b，并根据系统资源使用情况决定是否产生puzzle和puzzle问题的难度，在这个过程中不保存任何本次认证状态

信息，根据需要S_b和puzzle可使用软件或高速硬件产生。因此在这个阶段不存在存储资源消耗型和计算资源消耗型DoS攻击隐患。

当响应方收到Msg3 后，分阶段对S_b, solution和A的身份进行验证。对于使用虚假网络地址的攻击者而言，如果其无法接收到Msg2，则无法知道S_b，也不能伪造可通过验证的S_b，则其攻击无法继续进行；如果攻击者能窃听到Msg2，或者攻击者使用真实网络地址，要想继续进行攻击也必须要求解出puzzle问题，这将大大增加其攻击成本。

最后，使用A的公钥对A的身份进行验证，如果验证不能通过或验证后发现A已经认证过，都将中止本次认证，从而避免了可能的后续攻击。在这个步骤，非法的攻击者要想成功欺骗，必须在Msg3 中使用A的私钥正常签名，如果其没有进行窃听，伪造或破解私钥基本上是无法实现的。如果通过窃听获得了信息 $\{A\}_{Ka^{-1}}$，一方面，攻击者的网络地址如果与A不一致则无法通过S_b和solution的验证；另一方面，B使用A的公钥对信息进行解密后会发现A已经认证过，而协议不运行重复认证，则同样会中止本次认证，从而避免了攻击的继续进行。

2.3.2　抗重放攻击

重放攻击使指攻击者使用自己或其他用户在协议执行过程中生成的旧信息，以实现欺骗的目的。因为在Msg3 中包含时间戳T，同时S_b和puzzle的生成因子中都包含时间戳和本地秘密，如果篡改T的值则S_b和puzzle无法验证通过，所以本方案中的Msg3 可以抗重放攻击。

3　对 NSPK 协议的改进

3.1　NSPK 协议的改进

应用本文所述方案对 Lowe 改进的 NSPK 协议进行抵御 DoS 攻击的改进。改进后的协议如下：

(1)$A \rightarrow B$: A, S_a

(2)$B \rightarrow A$: S_b, T, y, d, $X<d+1,L>$

(3)$A \rightarrow B$: S_a, S_b, T, x_0', $\{\{A\}_{Ka^{-1}}, A, N_a\}_{Kb}$

(4)$B \rightarrow A$: $\{B,N_a,N_b\}_{Ka}$

(5)$A \rightarrow B$: $\{N_b\}_{Kb}$

协议的第(1)步、第(2)步为前述方案的Msg1 和Msg2。议第(3)步，结合方案中的Msg3 和原协议中的信息$\{N_a,A\}_{Kb}$进行改进，A的身份标识使用了A的私钥进行签名，与只使用B的公钥加密的A相比具有更强的安全性和认证能力。另外，使用B的公钥对 $\{A\}_{Ka^{-1}}$ 进行再次加密可以避免非法窃听者获取单独的 $\{A\}_{Ka^{-1}}$ 而假冒A的身份。在协议第(4)步，A通过收到的$\{B,N_a,N_b\}_{Ka}$知道B已经收到了Na，从而认证了B身份的真实性，并和A共享秘密临时值Na和Nb。协议最后一步$\{N_b\}_{Kb}$的作用是实现B对A的身份的认证。

3.2　改进的 NSPK 协议分析

改进后的 NSPK 协议与原来相比具有了较强的抵御 DoS 攻击的能力，具体分析如 2.3.1 节所述。通过 3.1 节中的分析，A, B 双方在协议第(4)步和第(5)步实现了相互认证，因此，协议具有正确的认证功能。在实现认证功能的同时 A, B 分别发送使用对方公钥加密的秘密值 Na 和 Nb，除 A, B 外没有其他用户能解密相应数据包，因此协议实现了临时秘密值的交换。

对于协议的防重放能力，除 2.3.2 节所提到的措施外，在协议第(4)步 B 发送给 A 的信息中包含自身标识 B，这是 Lowe 对最初的 NSPK 协议做的重要修改，其目的就是防止重放攻击。因此，协议具有良好的防重放攻击能力。

此外，由于改进协议继承了原协议良好的防假冒性能，因此改进的 NSPK 协议也具有较强的防假冒能力。

4　结束语

DoS 攻击的危害性使得研究者从不同的角度提出解决方法。本文就认证协议存在的 DoS 隐患进行分析，提出了针对无可信第三方的双主体认证协议抵御 DoS 攻击的安全方案，并给出了方案的实现细节。实验和分析说明方案具有一定的抗 DoS 攻击能力，但方案在效率和通用性方面还存在一定的不足。进一步的研究目标是对本文方案进行适当修改，使其具有更高的效率和更好的通用性。

参考文献

[1] 林梅琴, 李志蜀, 袁小铃, 等. 分布式拒绝服务攻击及防范研究[J]. 计算机应用, 2006, 23(8): 136-138, 151.

[2] 卫剑钒, 陈　钟, 段云所, 等. 一种认证协议防御拒绝服务攻击的设计方法[J]. 电子学报, 2005, 33(2): 288-293.

[3] Tuomas A, Pekka N, Leiwo J. DoS-resistant Authentication with Client Puzzles[C]//Proc. of Security Protocols Workshop. Berlin: Springer, 2000: 170-177.

[4] 史庭俊, 马建峰. 基于 Hash 函数的抗攻击无线认证方案[J]. 系统工程与电子技术, 2006, 28(1): 122-126.

(上接第 139 页)

[6] 张　涛, 胡铭曾. 计算机网络安全性分析建模研究[J]. 通信学报, 2005, 26(12): 100-109.

[7] Molisz W. Survivability Function——A Measure of Disaster-based Routing Performance[J]. IEEE Journal on Selected Areas in Communications, 2004, 22(9): 1876-1883.

[8] Kerivin H, Nace D, Pham T T L. Design of Capacitated Survivable Networks with a Single Facility[J]. IEEE/ACM Transactions on Networking, 2005, 13(2): 248-161.

[9] Zhang Yongzheng, Yun Xiaochun, Hu Mingzeng. Research on Privilege-escalating Based Vulnerability Taxonomy with Multi-dimensional Quantitative Attribute[J]. Journal of China Institute of Communications, 2004, 25(7): 107-114.

[10] Lin Xuegang, Xiong Hua, Xu Rongsheng. Survivability Analysis and Implementation for Network Information System[J]. Journal of Computer Engineering, 2005, 31(24): 161-163.

[11] Zhang Tao, Hu Mingzeng. An Effective Method to Generate Attack Graph[C]//Proceedings of the 4th International Conference on Machine Learning and Cybernetics. Guangzhou, China: [s. n.], 2005.