MANET中TCP数据包的乱序问题及其解决方案

孙 伟[1] 温 涛[2] 郭 权[2]
(东北大学软件中心 沈阳 110004)[1] (东北大学东软信息学院 大连 116023)[2]

摘 要 在MANET中,节点的移动性会导致不同的TCP数据包沿不同的路径到达接收端,进而在接收端产生大量的乱序数据包,影响TCP协议的性能。提出了一种延时响应TCP协议(TCP-D),通过延时触发拥塞控制算法来提高TCP协议在MANET中的性能。对延时响应TCP协议的吞吐量分析表明,增加延时定时器后,TCP_D协议仍能保证对标准TCP协议的友好性。仿真实验表明,TCP_D算法可以明显减少乱序数据包的数量,获得较标准TCP协议更高的网络吞吐量。而在网络结构稳定、无乱序数据包的情况下,改进协议仍具有很好的公平性和友好性。

关键词 拥塞控制,MANET,TCP,吞吐量
中图法分类号 TP393.04 **文献标识码** A

Mechanism for Packet Reordering in MANET

SUN Wei[1] WEN Tao[2] GUO Quan[2]
(Software Center, Northeastern University, Shenyang 110004, China)[1] (Neusoft Institute of Information, Dalian 116023, China)[2]

Abstract The standard implementation of TCP experiences heavy throughput degradation when packets are reordered. This paper presented a new version of the TCP protocol, called TCP_D, which improves its performance to mobile-induced reordering packets in mobile ad hoc network (MANET). In TCP_D, the response to the receipt of duplicate acknowledgement is delayed by a short period to allow the receiver to receive the packets that travel in different path. We analyzed TCP_D to show that its steady state throughput is similar to the native implementations of TCP when packet reordering does not occur. The simulation results show that TCP_D performs consistently better than standard implementations of TCP under persistent packets reordering. In the case that packets are not reordered TCP_D maintains the same throughput as typical implementation of TCP (TCP-NewReno) and shares network resource fairly.

Keywords Congestion control, MANET, TCP, Throughput

MANET[1]网络由于其组网灵活、方便,近年来得到了迅速的发展。对MANET的研究,一直是网络研究的热点[1,2]。TCP协议作为一种可靠、按序交付的传输协议,承载了目前Internet上大部分的数据流[3]。大量的研究表明,它对具有移动、随机丢失特性的无线通信环境不能很好地适应[4,6]。尤其在MANET中,TCP协议的性能下降非常严重。

近年来,有大量的研究致力于提高TCP在MANET中的性能,一些适用于MANET网络的TCP协议被提出并被广泛地讨论研究[5-13]。这些研究重点考虑的是无线链路的高误码率对于TCP拥塞控制算法的影响,将无线链路误码丢包误认为拥塞丢包,进而降低发送速率,其是影响TCP在MANET中性能的主要因素。因此,这些新的TCP协议的共同思想是修改现有的TCP协议中的拥塞发现和速率控制机制,通过区分链路误码丢包与拥塞丢包来提高TCP在MANET中的性能[5,6,9-11,13,18]。

然而,这些研究并未考虑到因移动性导致的乱序数据包对TCP性能的影响。在MANET中,所有的节点都可以自由移动。这样,不同的TCP数据包可能沿不同的路径到达接收端,因而在接收端产生大量的乱序数据包,影响TCP协议的性能。本文在分析移动性对于TCP性能影响的基础上,提出了一种延时响应策略,当发送端收到重复应答时,并不马上进入快速重传阶段,而是假定经过一定延时时间后,丢失的数据包会到达接收端,在此期间,发送端仍正常发送数据包。理论分析与仿真实验表明,该算法可以明显提高TCP协议在MANET中的性能。

1 移动性对TCP协议性能的影响

利用NS2[15]进行了仿真实验,以考察移动性对TCP协议性能的影响。仿真场景为包含60个节点的2000m×600m矩形区域。节点分布是通过调用NS2中的随机场景生成器来实现的。节点在矩形区域随机移动,节点移动方式为random waypoint,即所有节点随机选择目的地,然后以直线方式向目的地移动。当节点到达目的地后,经过一段时间暂停,再次随机选择下一个目标位置。本文设定暂停时间为0。MAC层使用IEEE802.11协议,网络接口服从WaveLan无线接口规范,带宽为2Mbps,传输半径为250m。TCP数据包

到稿日期:2008-09-09 返修日期:2008-12-04 本文受国家高技术研究发展计划(863)项目(No. 2004AA113020)资助。
孙 伟(1978—),男,博士研究生,主要研究方向为无线网络、拥塞控制,E-mail:sunwei@neusoft.edu.cn;温 涛(1962—),男,教授,博士生导师,主要研究方向为无线网络、网络安全;郭 权(1973—),男,博士,副教授,主要研究方向为网格计算。

大小为 1000Bytes，节点接口队列长度为 50 个数据包。采用 DSR[16] 作为路由层协议。仿真持续时间为 1000 秒。利用 NS2 的流量生成器从 60 个节点中随机选择 10 对节点，作为 TCP 发送端和接收端。仿真过程中，保证该流量模式不变。网络仿真结构如图 1 所示。

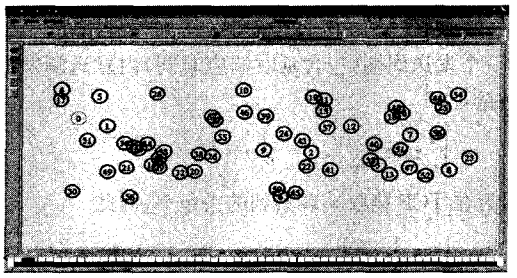


图 1　网络仿真结构

表 1 记录了不同速率下 TCP 数据流的平均吞吐量，以及发生乱序数据包的次数。可以看出，节点的移动性导致了大量乱序数据包的出现，影响了网络吞吐量。

表 1　不同速度模式下 TCP 协议的吞吐量及发生乱序数据包的次数

平均速度(m/s)	吞吐量(kbps)	发生乱序数据包的次数
0	802.35	89
2	670.02	247
5	587.35	298
10	420.35	325
15	401.13	319
20	379.87	341
25	368.76	302
30	359.71	336

TCP 通过数据包的序列号来保证数据的按序交付，假设序列号为 n 的数据包丢失，接收端每收到一个序列号大于 n 的数据包都会认为这是一个乱序(out-of-order)数据包，在数据包 n 被正确接收之前，每一个这样的数据包都将产生一个对于 n 的重复应答。如果发送端收到一定数量的重复应答，将认为该数据包丢失，此时，重传丢失的数据包，并将拥塞窗口减半。

这种方式在有线网络中可以得到很好的应用，因为有线网络结构相对稳定，所有数据包按照同一路径到达接收端。因此，数据包的到达遵循先发先到的原则。而在 MANET 中，节点可以自由移动，这将频繁导致路由的拆毁和重建，数据包也将通过多条路径到达接收端，此时，将产生大量的乱序数据包。而这种乱序数据包并非由网络拥塞引起，很可能是数据包通过一条更耗时的路由到达接收端。因此，盲目进行拥塞控制，降低发送速率，将降低网络的利用率。

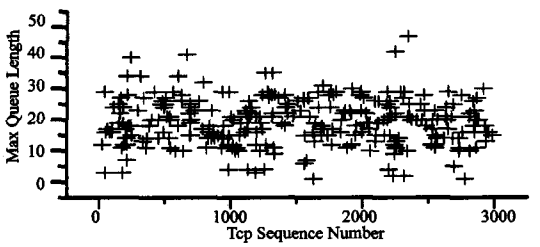


图 2　发生乱序数据包时接口队列长度(Mean Speed=10m/s)

为了统计乱序数据包与网络拥塞状况之间的关系，记录了发生乱序数据包时网络节点接口队列的最大长度。图 2 显示了节点平均移动速度为 10m/s、TCP 序列号为 0－3000[1]、发生乱序数据包时网络中接口队列的占用情况(改变节点的移动速度，会得到类似的实验结果)。

可以看到，当乱序数据包出现时，网络路由节点的接口队列并未达到饱和(很少超过最大缓冲的 75%)，可见，乱序数据包并非由网络拥塞引起。而标准 TCP 协议把乱序数据包作为网络拥塞的指示，降低了发送速率，影响了网络的吞吐量。

2　延时响应算法

2.1　基本算法

既然是节点的移动性导致 TCP 数据包经过不同的路径到达接收端，进而导致乱序数据包的产生，那么直接的想法是，当发送端收到重复应答时，并不马上进入拥塞控制阶段，而是等待一定的时间 ϕ，使缺失的数据包可以到达接收端。如果经过 ϕ 时间后，缺失的数据包仍未到达，则认为该数据包丢失，进入拥塞控制阶段。

如图 3 所示，发送端发送序列号为 1－5 的数据包，由于节点的移动性，数据包 2 并未按序到达接收端，随后到达的 3－5 数据包被认为是乱序数据包，都将产生对数据包 2 的应答，3 个重复的应答将导致发送端进入快速重传阶段，还要减小不必要的拥塞窗口。

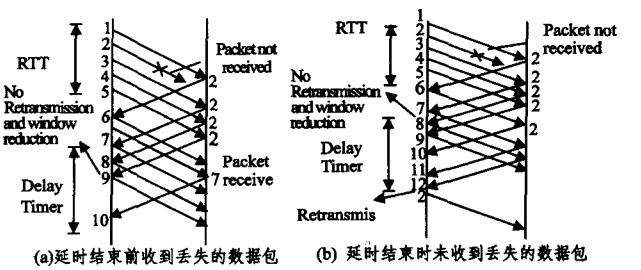


图 3　延时响应 TCP 算法

在延时响应策略中，当收到第一个重复应答时，发送端启动一个延时计时器，如果在延时计时器超时之前，2 号数据包正确地到达了接收端，这将导致接收端产生对于数据包 7 的应答。发送端收到后，可以判定数据包 1－6 已经正确到达接收端。这样，避免了对于 2 号数据包的重传和拥塞窗口的减小。在延时等待阶段，每收到一个重复的应答，发送端将继续发送新的数据包，这种算法类似 TCP 的快速恢复算法，保证在延时等待阶段链路的利用率，如图 3(a)所示。

如果数据包 2 由于拥塞或者链路错误丢失，那么在延时定时器结束时，仍未收到对于 2 号数据包的确认，如图 3(b)所示。此时，重传 2 号数据包，进入拥塞控制(标准 TCP)或者是一些为 MANET 设计的控制算法[5,6,9,11,13]。

由于延时响应算法仍然是应答驱动的(ack-clocked)，即只有收到一个重复应答才向网络中注入新的数据包，重复应答表明已有数据包离开网络，此时可以向网络中注入相等数量的数据包。因此，延时响应策略并不会加重网络的拥塞。

2.2　延时 ϕ 的选择

下面讨论延时定时器 ϕ 的设定。TCP 通过重传超时和

[1] 该序列号为 NS2 中 TCP 数据包的序列号。

重复应答来判断网络拥塞。当收到重复应答时，认为网络中度拥塞，将拥塞窗口减小一半，进入快速重传和快速恢复阶段。当重传超时发生时，TCP 认为网络严重拥塞，此时，拥塞窗口设为一个数据段，进入慢启动阶段。因此，延时响应定时器的设置，应避免 RTO 的发生。即 ϕ 的选择上限为 RTO。

如图 4 所示，S，R 分别为发送端和接收端。A，B，C 为中间路由节点。在 TCP 实现中，每个到达发送端的应答都将导致平滑计算 RTT，因此发送端保持的 RTT 为 $S'\rightarrow C\rightarrow R$ 的往返时延。当 S 移动到 S' 时，先前发送的数据包将缓存在 A 或 B。设由此到达 R 再返回 S' 的往返时延为 t，由乱序数据包的定义可知，当从 S' 发送的数据包达到 R 时，缓存在 A 或 B 的数据包仍未到达，所以有 $RTT<t$。可见，延时 ϕ 应满足 $\phi>t$。

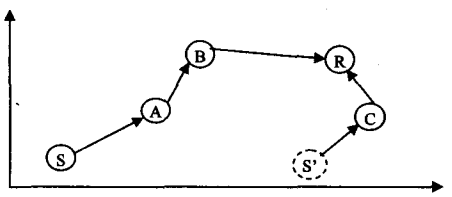


图 4　移动性对延时的影响

由以上分析可知，延时 ϕ 应满足 $RTT<\phi<RTO$。RTT，RTO 均为发送端已知数据，采用低通过滤参数 λ 设置 ϕ 的取值。令

$$\phi=\lambda RTT+(1-\lambda)RTO \tag{1}$$

当 $\lambda<0.5$ 时，可保证 $RTT<\phi<RTO$。对延时响应机制的分析可知，ϕ 的选择不会影响延时响应协议对标准 TCP 协议的公平性，如图 3 所示。因此，ϕ 应接近 RTO，以保证有足够的延时等待缺失的数据包到达。而如果 ϕ 过于接近 RTO，会造成延时到达时刻与超时重传时刻的间隔太短，容易发生误判。仿真实验表明，$\lambda=0.3$ 可以保证延时响应协议取得较优的性能。

3　延时响应 TCP 协议的性能分析

由 2.1 节中的分析可知，延时响应 TCP 协议的窗口变化规律如式(2)：

$$\begin{aligned} &no\ packet\ loss: W_{i+1}=W_i+\alpha \\ &packet\ loss: W_{i+1}=W_i+\alpha; t_{dack}<t<t_{dack}+\phi \\ &\qquad\qquad W_{i+1}=\beta W_i; t\geqslant t_{dack}+\phi \end{aligned} \tag{2}$$

α 和 β 分别为加性增加和乘性减小参数。t_{dack} 为收到第一个重复应答的时刻。W_i 为拥塞窗口大小。拥塞窗口变化规律如图 5 所示。接下来分析延时响应 TCP 协议的稳态吞吐量。

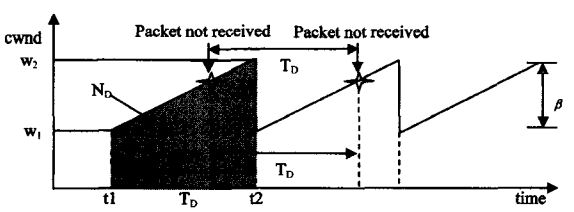


图 5　延时响应 TCP 协议的窗口变化

w 为拥塞窗口大小，因此有：

$$\frac{dw}{dt}=\frac{\alpha}{RTT}\Rightarrow w=\frac{\alpha\cdot t}{RTT}+C\Rightarrow t=\frac{(w-C)RTT}{\alpha} \tag{3}$$

T_D 为从接收端观察到的连续两次发生丢包的时间间隔，由图 5 可知：

$$T_D=\frac{(w_2-C)RTT}{\alpha}-\frac{(w_1-C)RTT}{\alpha}=\frac{RTT}{\alpha}(w_2-w_1) \tag{4}$$

由式(2)可知 $w_1=\beta w_2$，带入式(4)，有

$$T_D=\frac{RTT}{\alpha}w_2(1-\beta) \tag{5}$$

N_D 为一个发送周期(包含延时响应时间)内发送端发送的数据包数量：

$$N_D=\int_{t_1}^{t_2}w(t)\frac{dt}{RTT}=\frac{1}{2\alpha}w_2^2(1-\beta^2) \tag{6}$$

p 为在 TCP 接收端观察到的丢包率，因此

$$N_D=\frac{1}{p}\Rightarrow\frac{1}{p}=\frac{1}{2\alpha}w_2^2(1-\beta^2)\Rightarrow w_2=\sqrt{\frac{2\alpha}{p(1-\beta^2)}} \tag{7}$$

延时响应 TCP 协议的吞吐量 $B=\frac{N_D}{T_D}$。将式(5)和式(6)代入，可得：

$$B=\frac{\frac{1}{2\alpha}w_2^2(1-\beta^2)}{\frac{RTT}{\alpha}w_2(1-\beta)}=\frac{w_2(1+\beta)}{2RTT} \tag{8}$$

将式(7)带入，可得延时响应 TCP 协议在稳定状态下的吞吐量：

$$B=\sqrt{\frac{\alpha(1+\beta)}{2(1-\beta)}}\cdot\frac{1}{RTT\sqrt{p}} \tag{9}$$

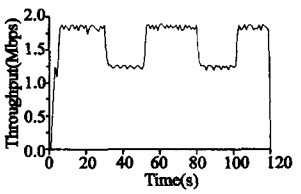
Padhye J 在文献[17]中推导出稳定状态下 TCP 协议的吞吐量：

$$B=\sqrt{\frac{3}{2\beta}}\cdot\frac{1}{RTT\sqrt{p}}+o\left(\frac{1}{\sqrt{p}}\right) \tag{10}$$

根据文献[17]对 TCP 友好性定义可知，改进后的 TCP 协议仍可以保证对标准 TCP 协议的友好性(Friendly)。

4　性能仿真试验结果及分析

首先验证延时响应协议对标准 TCP 协议的公平性。由第 3 节中的分析可知，在无乱序数据包的情况下，延时响应协议可以获得和 TCP 类似的吞吐量。考虑以下网络结构，发送端与接受端通过一条带宽为 2M 的瓶颈链路，链路误码率为 0.5%，往返时延为 40ms。在 30s 到 50s 和 80s 到 100s 之间加入一个 UDP 数据流，发送速率为 0.5M。由于网络结构稳定，因此数据包遵循先发现到的原则。图 6 与图 7 显示了延时响应协议和标准 TCP 协议所获得的网络吞吐量的情况。



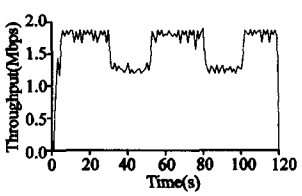


图 6　TCP 数据流吞吐量变化　　图 7　TCP_D 数据流吞吐量变化

从图 6 和图 7 可以看出：(1)延时响应协议可以获得与标准 TCP 协议相似的吞吐量，可以保证与 TCP 协议在竞争网络资源上的公平性。(2)增加延时定时器后，协议仍然可以迅速对网络状态的变化做出响应。如图 7 所示，在 30s 和 80s 加入 UDP 数据流后，TCP_D 的吞吐量迅速稳定在当前可用带宽附近。(3)延时响应协议的吞吐量较标准 TCP 协议表现

出较大的抖动性。这是因为在延时 ϕ 超时前，拥塞窗口仍然在增长，因此当延时 ϕ 超时后，乘性减小参数 β 作用于一个更大的拥塞窗口，造成较大的拥塞窗口变化。

图 8 记录了 TCP_D 与 TCP NewReno 数据流的拥塞窗口变化情况。由图 8 可以看出，虽然 TCP_D 的拥塞窗口均值和 TCP 类似，但是变化幅度更大。

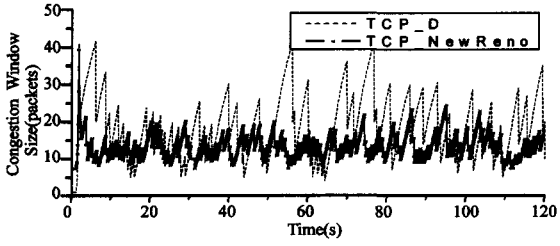


图 8　两种协议的拥塞窗口比较

进一步考察增加延时定时器后对发送速率平滑性的影响。定义相邻时间间隔的吞吐量比例 $R_i=\frac{T_i}{T_{i-1}}$。T_i，T_{i-1} 分别为第 i 个时间间隔内发送端所获得的吞吐量。$R_i<1$ 意味发送速率增加，反之意味发送速率减小。设定时间间隔为0.1秒。

按 0.1 为间隔，将[0,2]分为 20 个相等的间隔，统计吞吐量比例在各个间隔中的数量占总数量的比例，即吞吐量的概率分布。图 9 显示了 10 条 TCP 数据流和 TCP_D 数据流通过一条 2M 瓶颈链路时的吞吐量比例的概率分布情况。

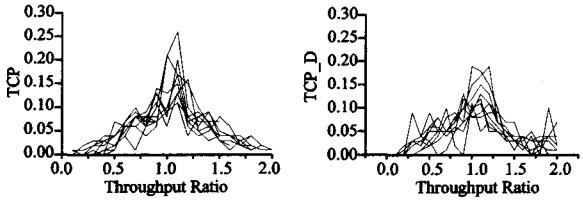


图 9　两种协议的吞吐量比例概率分布

从图 9 可以看出，TCP 数据流的吞吐量比例分布较 TCP_D 更加集中在 1.0 附近，说明发送速率的变化更加平滑。而 TCP_D 的吞吐量比例分布范围更广，发送速率表现出更强的抖动性。

由上述分析可知，延时响应协议在保证对标准 TCP 协议公平性的基础上，牺牲了发送速率的平滑性，因此并不适用于对发送速率平滑性要求较高的应用，如视频点播、流媒体播放等。

接下来考察延时响应协议在无线网络中的性能。仿真场景为包含 60 个节点的 2000m×600m 矩形区域。全部节点均为静止状态。从网络中随机选择若干对节点，在选定的一对节点间部署 TCP-NewReno 数据流和延时响应的 TCP 数据流(TCP-D)。这样可以保证 TCP-NewReno 和 TCP-D 经历相同的网络节点跳数。节点对的个数为 1,2,3,5,8,10,15，用以模拟不同的网络负载的情况。计算两种协议的归一化吞吐量[2] 及其平均值。从图 10 可以看出，无论是网络负载较轻还是较重，延时响应 TCP 协议均可获得和 TCP-NewReno 大致相等的吞吐量。这与(3)中的分析一致。

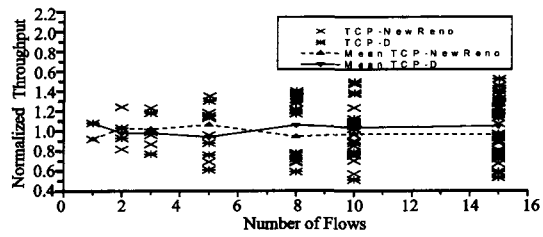


图 10　不同网络负载情况下两种协议的吞吐量比较

接下来考察在网络节点移动情况下延时响应 TCP 协议性能的提升。每次在包含 60 个节点的 2000m×600m 矩形区域仿真场景中随机选择一对节点，部署 TCP-NewReno 和延时响应 TCP 策略，改变网络节点的运行速度，为消除偶然因素的影响，每个速度模式下运行 10 次，取平均值。

图 11 显示了在节点不同的平均移动速度下，两种协议所感知的乱序数据包的数量。可以看出，由于节点的移动，导致在接收端产生了大量的乱序数据包，由于在延时响应 TCP 协议中加入了等待机制，因此在接收端感知到的乱序数据包的数量要小于标准 TCP 协议。图 12 显示了两种协议在节点的不同移动速度下的吞吐量比较。可以看出延时响应 TCP 协议所获得的网络吞吐量要明显高于标准 TCP 协议。

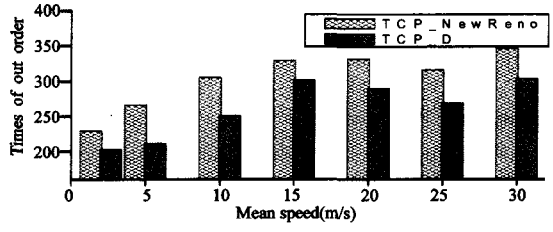


图 11　两种协议所感知的乱序数据包数量

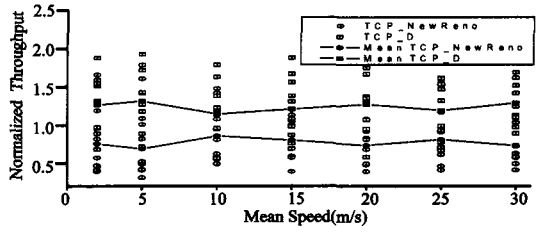


图 12　不同速度模式下两种协议的吞吐量比较

结束语　MANET 中节点的移动性会产生大量乱序的数据包，而传统的 TCP 协议在收到 3 个重复应答时，将进入拥塞控制阶段，降低发送速率，影响网络的性能。本文在分析移动性对 TCP 协议影响的基础上提出一种改进的 TCP 算法，当收到重复应答时延时进入拥塞控制阶段。理论分析和仿真实验表明，该算法在网络中存在大量由移动性导致的乱序数据包时，能有效提高 TCP 协议的性能，而在无乱序数据包时，改进协议仍具有很好的公平性和友好性。

参 考 文 献

[1] Dow C R, Lin P J, Chen S C, et al. A study of recent research trends and experimental guidelines in mobile ad-hoc network [J]. Advanced Information Networking and Applications, 2005

(下转第 108 页)

[2] 假设网络中有 n 个数据流，则第 i 个数据流的归一化吞吐量为 $T_i=\frac{x_i}{\frac{1}{n}\sum_{j=1}^{n}x_i}$。

证书可验证加密签名 ves。克服了 Gu-Zhu 方案存在恶意签名性的弱点。

4.4 可克服合谋攻击

在 Gu-zhu 方案中，注意到验证方程① 并不保证同一个 k_2 被签名者用于计算 $h'=H_3(e(Q_A,P_{pub})^{k_2})$ 和 $U_2=k_2P$，$V=hD_X+(k_1+h'k_2)P+h'Q_A$，因此签名者在不知道 h' 的情况下也可以产生有效的可验证加密签名。这一发现导致了该协议存在合谋攻击[8]。

本方案中，取消参数 k_2，用 b 替代。$a=x_iY_A$，$b=e(a,P_{pub})$，从等式中知 b 的产生需要签名者的秘密值 x_i 和仲裁者的公开钥 Y_A。验证时 $b=e(S_A,X_i)$，计算 b 需要签名者的公开钥 X_i 和仲裁者的秘密钥 S_A。因为 $h'=H_3(e(Y_A,P_{pub})^b)$ 是可验证的，所以 $U_1=h'P$ 也可以验证，因而就避免了合谋攻击者通过利用 U_1 计算自己的可验证加密签名而进行计算原始签名的可能。

结束语 本文研究分析了无证书签密方案和 Gu-Zhu 的基于身份的可验证加密签名方案后，结合无证书和可验证签名，提出了基于无证书的可验证的加密签名方案，证明了该方案的正确性和不可伪造性，以及它可以有效避免 Gu 和 Zhu 提出的可验证加密签名方案存在的恶意签名和合谋攻击。

参 考 文 献

[1] Rivest R L, Shamir A, Adleman L. A method for obtaining digital signatures and public-key cryptosystems. Communications of the ACM, 1978, 21: 120-126

[2] Shamir A. Identity based cryptosystems and signature schemes [C]// Proceedings of the Advances in Cryptology-Cryipto'84. Lecture Note in Computer Science: 196. Springer-Verlag, 1984: 47-53

[3] Boneh D, Franklin M. Identity - base encryption from the Well *pairing*[C]// *Proceedings of the Advances in Cryptology-Crypto* 2001. Lecture Note in Computer Science: 2139. Springer-Verlag, 2001: 213-229

[4] Al-Riyami S S, Paterson K. Certificateless Public Key Cryptography[C]// Proc. of Asiacrypt'03. [S. l.]: Springer-Verlag, 2003

[5] Boneh D, Gentry C, Lynn B, et al. Aggregate and verifiably encrypted signatures from bilinear maps[C]// Proceedings of the EUROCRYPT' 03. LNCS, vol. 2656. Berlin: Springer, 2003: 416-432

[6] Hess F. Efficient identity based signature schemes based on pairings[C]// Proceedings of the 9th Annual International Workshop on Selected Areas in Cryptography. Lecture notes in Computer Science ; 2595. Springer-Verlag, 2003: 310-324

[7] Gu C X, Zhu Y F. An ID-based verifiable encrypted signature scheme based on Hess's scheme[C]// Proceedings of the 1st SKLOIS Conference on Information Security and Cryptology. Lecture Notes in Computer Science: 3822. Springer-Verlag, 2005: 42-52

[8] 张振峰. 基于身份的可验证加密签名协议的安全性分析[J]. 计算机学报, 2006, 29(9): 1688-1693

[9] Bao F. Colluding attacks to a payment protocol and two signature exchange schemes[C]// Proceedings of the Advances in Cryptology-ASIACRYPT 2004. Lecture Notes in Computer Science: 3329. Spring-Verlag, 2004: 417-429

(上接第 85 页)

[2] Akyildiz I F, Wang X, Wang W. Wireless mesh networks: a survey[J]. Computer Networks, 2005, 47(4): 445-487

[3] Medina A, Allman M, Floyd S. Measuring the evolution of transport protocols in the internet[J]. ACM SIGCOMM Computer Communication Review, 2005, 35(2): 37-51

[4] Holland G, Vaidya N. Analysis of TCP Performance over Mobile Ad Hoc Networks[J]. Wireless Networks, 2002, 8(2/3): 275-288

[5] Lohier S, Doudane Y G, Pujolle G. Cross-Layer Loss Differentiation Algorithms to Improve TCP Performance in WLANs[C]// Proceedings of the 11th IFIP TC6 International Conference on Personal Wireless Communications (PWC'06). vol. 4217 of LNCS. Springer, 2006: 297-309

[6] Barman D, Matta I. Effectiveness of Loss Labeling in Improving TCP Performance in Wired/Wireless Networks[C]// Proceedings of the 10th IEEE International Conference on Network Protocols (ICNP'02). IEEE Press, 2002: 2-11

[7] Biaz S, Vaidya N H. Discriminating Congestion Losses from Wireless Losses Using Inter-Arrival Times at the Receiver[C]// Proceedings of the 2nd IEEE Symposium on Application-Specific Systems and Software Engineering Technology (ASSET'99). IEEE Press, 1999: 10-17

[8] Cen S, Cosman P C, Voelker G M. End-to-End Differentiation of Congestion and Wireless Losses[J]. IEEE/ACM Transactions on Networking, 2003, 11(5): 703-717

[9] Chandran K, et al. A Feedback - Based Scheme for Improving TCP Performance in Ad Hoc Wireless Networks[J]. IEEE Personal Communications, 2001, 8(1): 34-39

[10] Eddy W M, Ostermann S, Allman M. New techniques for making transport protocols robust to corruption-based loss[J]. SIGCOMM Computer Communication Review, 2004, 34(5): 75-88

[11] 刘俊, 隆克平, 徐昌彪, 等. 两种改善无线 TCP 性能的新机制[J]. 电子学报, 2004, 32(12): 2059-2062

[12] Khayat I E, Geurts P, Leduc G. Enhancement of TCP over Wired/Wireless Networks with Packet Loss Classifiers Inferred by Supervised Learning[R]. Research Unit in Networking, University of Liège, 2004

[13] Bhandarkar S, Sadry N E, Reddy A L N, et al. TCP-DCR: A Novel Protocol for Tolerating Wireless Channel Errors[J]. IEEE Transactions on Mobile Computing, 2005, 4(5): 517-529

[14] Paul R, Trajkovic L. Selective - TCP for Wired / Wireless Networks[C]// Proceedings of the International Symposium on Performance Evaluation of Computer and Telecommunication Systems (SPECTS'06). 2006

[15] NS-2 Network Simulator (Ver. 2) LBL[OL]. http://www.mash.cs.berkly.edu/ns/

[16] Johnson D B, Maltz D A, Hu Y C, et al. The dynamic source routing protocol for mobile ad hoc networks[M]. IETF Draft MANET Working Group, 2000

[17] Padhye J, Firoiu V, Towsley D, et al. Modeling TCP throughput: A simple model and its empirical validation[C]// Proceedings of SIGCOMM'98. 1998

[18] Liu J, Singh S. ATCP: TCP for Mobile Ad-Hoc Networks[J]. IEEE Journal on Selected Areas in Communications, 2001, 19(7): 1300-1315