基于洋葱路由的双向匿名秘密通信协议

赵梦瑶 李晓宇

(郑州大学信息工程学院 郑州 450001)

摘 要 在网络中,通信双方的身份是一项重要的隐私,匿名通信可以隐藏通信者的身份。对于匿名通信的研究,大部分都是关于发送者匿名,而对于接收者匿名以及双向匿名通信的研究比较少。洋葱路由系统使用源路由协议和层层加密的思想构造洋葱路径,消息按照洋葱路径经过有序中转节点进行转发,隐藏了发送者的地址,实现了发送者匿名,能够有效地防止窃听和流量分析。基于洋葱路由,提出了一种新的双向匿名秘密通信协议。发送者构造的洋葱路径包含系统中所有的节点,每到一跳中转节点,都要判断该节点上是否有接收者用户。如果没有,则继续转发消息;如果有,则接收者收到消息,同时终止转发。发送者(接收者)的身份不会被对方或者任意的其他用户获取,而且除了通信双方之外,任意的中转节点和侵入者都不能获取消息,因此该协议很好地实现了双向的匿名秘密通信。该协议不使用组播实现接收者匿名,有效地减少了系统中的流量;且只基于洋葱路由一种匿名系统,实现简单。实验结果表明,随着系统用户的增加,平均响应时间和平均双向通信时间近似呈线性增长,说明该系统在用户数量很多的情况下仍然工作稳定,健壮性较好。

关键词 双向匿名秘密通信,洋葱路由,网络安全,RSA公钥系统

中图法分类号 TP319 **文献标识码** A **DOI** 10.11896/j.issn.1002-137X.2019.04.026

Bidirectional Anonymous Secret Communication Protocol Based on Onion Routing

ZHAO Meng-yao LI Xiao-yu

(School of Information Engineering, Zhengzhou University, Zhengzhou 450001, China)

Abstract In the network, the identity of communicators is an important privacy. Anonymous communications can hide the sender and the recipient. Most of the research on anonymous communication is about the sender's anonymity. There is less research on the receiver's anonymity and bidirectional anonymity. In onion routing system, onion path is constructed by using source routing protocol and layer by layer encryption. The message is forwarded through orderly transit nodes according to onion path, which hides the sender's address, realizes the sender's anonymity and effectively prevents eavesdropping and traffic analysis. A new bidirectional anonymous secret communication protocol was proposed based on onion routing in this paper. The onion path constructed by the sender contains all the nodes in the system. Every hopping transfer node must judge whether the node is the receiver or not. If not, the message continues to be forwarded, and else, the recipient receives the message and the forwarding terminates. The identity of the sender(receiver) is not captured by the other party or any other user. Besides both sides of the communication, any transit node or intruder can't get the message. Therefore, the protocol achieves a two-way anonymous secret communication well. The anonymity of the receiver is realized without multicast, which effectively reduces the traffic in the system. The protocol is only based on onion routing anonymity system and is relatively simple. The experimental results show that with the increase of system users, the average response time and the average bidirectional communication time increase almost linearly, which indicates that the system is still stable and robust in the case of a large number of users.

Keywords Bidirectional anonymity secret communication, Onion routing, Network security, RSA public key system

1 引言

随着Internet的发展,网络安全与隐私保护变得尤为重要。加/解密技术可以保护网络中的数据安全,有效防止窃听和流量分析。但是,在网络中传递消息的IP包的包头信息是未被加密的,可以从中获取源地址、目的地址等相关信息,从

到稿日期:2018-02-09 返修日期:2018-06-27 本文受国家自然科学基金项目(61472412),河南省教育厅自然科学基金项目(14A520012)资助。

赵梦瑶(1994—),女,硕士,主要研究方向为移动信息安全、匿名通信;**李晓宇**(1974—),男,博士,副教授,CCF会员,主要研究方向为量子计算与量子信息、移动计算,E-mail:iexyli@zzu.edu.cn(通信作者)。

而暴露发送者和接收者的身份。而且，在情报机构、国家安全部门等一些对网络安全性要求很高的机构，保护通信双方的身份是必要的。双向匿名通信系统可以较好地隐藏通信双方的身份信息。

文献[1]基于公钥密码体制提出了 Mix 系统，通过混淆节点转发消息，混淆节点用私钥解密接收到的消息得到消息的转发地址并转发消息。该系统不需要可信的第三方，通过消息的重路由、填充数据包和批量发送能够防止流量分析。文献[2]提到美国海军实验室研发了洋葱路由系统，其由 Mix 系统发展而来，是一种基础设施，可以在公共网络上提供隐私交流。消息按照构建的洋葱路径，通过有序的洋葱路由器的转发到达接收者。洋葱路由提供实时、双向的匿名链接，可有效防止窃听和流量分析。对于网页浏览、电子邮件、远程登录等网络应用，不用修改，其自身的网络应用协议通过代理就可以使用洋葱路由的匿名链接。文献[3]提出了第二代洋葱路由 Tor，其能提供基于链路的低时延的匿名通信服务，在发送消息前首先建立一条通信链路，通过链路转发消息。Tor 解决了第一代洋葱路由的一些局限性。相比于第一代洋葱路由，Tor 增加了前向保密性、拥塞控制、目录服务器、完整性检查、可配置的出口策略和节点同步等，在匿名性、使用性和高效性之间进行了合理的权衡。

文献[4]提出将 Crowds、Tor 和组播相结合来实现双向匿名秘密通信，发送者是 Crowds 群中的一员，其发送的信息先经过 Crowds 随机转发，再通过 Tor 转发，最后组播给接收者，实现双向匿名秘密通信。该方法能有效抵抗流量分析。文献[5]用 Crowds 实现发送者匿名，用组播实现接收者匿名，以达到双向匿名秘密通信。文献[6]提出让消息先经过 Hordes 转发，再经过 Tor 转发，接着转交给 Hordes，并通过携带的 Id 确认接收者。这种方式有效抵抗了通信流和端对端分析，同时实现了双向匿名秘密通信。

本文对洋葱路由系统进行改进，提出了一种新的双向匿名秘密通信协议。洋葱路径中包含系统中所有的节点，针对每一个转发节点，判断该节点上是否有接收者用户。用组播实现接收者匿名时，系统中会产生大量的流量，造成拥塞，而本文方法有效地减少了系统中的流量，并且只用一种匿名系统，更加简单。

2　双向匿名秘密通信协议

2.1　双向匿名秘密通信的含义

双向匿名秘密通信就是在使用当前网络协议的基础上，通过某种方法隐藏通信双方的身份，使通信双方不能获取对方的身份，并且只有消息的接收者能够获取到消息，任何其他用户或者节点都不能获取到消息，因此窃听者或者恶意用户难以获取或推测通信双方的身份信息以及通信内容，保护了消息的发送者和接收者的身份以及消息本身的机密性，从而保证了网上用户的身份不被泄露。

在双向匿名秘密通信系统中，通信双方知道对方的存在，某个发送者知道要发送消息给特定接收者。而且，通信双方持有可以唯一标识对方的信息，例如身份标识符或者公开密钥等，这样在消息的转发过程中就可以确定消息的合法接收者，并且只有消息的合法接收者能够获取到消息，任意的其他用户或者节点均不能获取到消息。在通信过程中，通信双方始终不了解对方的真实身份，其他用户也不可能知道通信双方所在的网络节点信息，从而确保了通信双方的身份隐私不被对方或者其他用户获取。

在情报机构、国家安全部门等一些对网络安全性要求很高的机构中，保护通信双方的身份信息是必需的。双向匿名秘密通信技术可以隐藏通信双方的身份，因而有着重要的应用价值。

2.2　洋葱路由机制

洋葱路由是一种灵活的通信基础设施，仅位于应用层之下。发送者发送的消息通过有序的洋葱路由器转发，间接地发送给接收者。各种网络服务不用修改自身的网络应用协议，通过代理就可以应用洋葱路由。洋葱路由为应用提供了实时的、双向的匿名链接，能够防止窃听和流量分析。

洋葱路由系统使用源路由协议，在消息发送之前确定消息的转发路径。根据确定的转发路径，用上一跳的密钥对下一跳的数据和 IP 地址进行加密，这样通过层层加密就构成了洋葱路径。通过洋葱路径转发消息形成了路径的多级混淆，有效地隐藏了最终的目标节点。洋葱路由就是按照洋葱路径经过一系列中转节点转发消息的过程。路由信息是由源路由协议确定的路径节点，逆序将节点地址和消息用其上一跳节点的密钥进行层层加密，得到洋葱路径。每到一跳洋葱路由节点，节点只能解密洋葱的最外一层，得到下一跳节点地址，并且按其转发消息。

所有使用端口链接的地方都可以应用洋葱路由系统。网页浏览、电子邮件等应用通过一系列的代理，不用修改自身的网络应用协议，可直接应用该系统。发送者发送的消息首先到达入口代理，入口代理构造洋葱路径，按照洋葱路径转发消息。每到一跳路由节点，节点解密最外层洋葱，根据得到的下一跳节点地址继续转发消息，直到消息到达接收者。返回消息按原路返回。每一个洋葱路由器看到的都是剥去了最外一层的洋葱。洋葱包经过洋葱路由的解密，在进出时呈现的形式不同。与发送者相连的入口代理知道发送者和接收者地址，与接收者相连的出口路由知道接收者地址，而路径上的中转路由节点只知道前一跳和后一跳路由。消息经过层层加密与洋葱路由的转发，可以有效地防止窃听和流量分析。

2.3　基于洋葱路由的双向匿名秘密通信协议

匿名通信系统[7]可以进一步划分为使用中心混淆网络系统和点对点系统。在使用中心混淆网络的系统中，用户连接到提供匿名通信的混淆池。入口代理构造一条转发路径，消息通过这条路径转发到达接收者。洋葱路由属于这类匿名通信系统。在点对点匿名通信网络中，网络中的每一个节点都是一个混淆节点，每一个节点既可以是发送者也可以是接收者。显而易见，这种网络很庞大，使用匿名服务的用户越多，网络中产生的流量越多，系统提供的匿名性就越好。本文提出的双向匿名秘密通信协议属于此类匿名通信系统。

2.3.1 协议基于洋葱路由的重要修改

洋葱路由运用公钥密码体制(RSA)以及源路由协议为因特网上的通信提供安全和匿名,阻止了恶意节点的窃听和流量分析。本文的双向匿名秘密通信协议对洋葱路由进行了重要修改,具体包括以下几点。

1)使用匿名通信协议的所有节点都需要向服务器注册,使用基于公开密钥系统的数字证书机制,加入公开密钥系统。系统中的每一个节点既可以是发送者,也可以是接收者,或者作为中转路由节点。

2)用户和节点分开注册。用户与其所在节点的对应关系只有该用户知道,对其他任何用户和节点都保密。

3)每一个发送者构造洋葱路径(Onion Path,OP),该洋葱路径 OP 包含所有节点,且构造的洋葱路径节点的顺序是随机的。每一层洋葱都包含接收者用户名 RUN,用于让节点判断其上的用户是否包括接收者用户 RUN。发送者每次发送消息时都需要重新构造 OP,各个节点不存储路径信息。

4)消息在报文中的位置不同。发送者发送给接收者的消息不是位于构造的洋葱路径的最内层,洋葱路径的最内层没有消息,消息和洋葱路径连接在一起生成发送报文。

5)加密方式不同。对构造的随机路径上的每一跳节点 N 的 IP 地址 IP_n 加上接收者用户名 RUN,用 AES 算法生成的对称密钥 SK 加密,得到 $\{IP_n,RUN\}_{SK}$,再用上一跳节点 $N-1$ 的公钥 PK_{N-1} 加密对称密钥,得到 $\{\{IP_n,RUN\}_{SK},\{SK\}_{PK(n-1)+}\}$,路径信息就这样被层层加密成洋葱路径 OP。消息部分 P 用接收者的公钥 PK_{RUN} 加密为 $\{P\}_{PKRUN+}$。另外,将路径信息 OP 和消息部分 $\{P\}_{PKRUN+}$ 连接起来得到 $\{OP,\{P\}_{PKRUN+}\}$,再用 AES 算法生成的一个对称密钥 SK' 对 $\{OP,\{P\}_{PKRUN+}\}$ 进行加密,得到 $\{OP,\{P\}_{PKRUN+}\}_{SK'}$,用下一跳节点 $N+1$ 的公钥 PK_{N+1} 加密对称密钥 SK',得到 $\{SK'\}_{PK(n+1)+}$,从而形成报文 $\{\{OP,\{P\}_{PKRUN}\}_{SK'},\{SK'\}_{PK(n+1)+}\}$,并发送给下一跳节点 $N+1$。下一跳节点 $N+1$ 收到报文后,首先用节点私钥 NSK_{n+1} 对加密的对称密钥部分 $\{SK'\}_{PK(n+1)+}$ 进行解密,获得对称密钥 SK',再对用对称密钥加密的报文 $\{OP,\{P\}_{PKRUN}\}_{SK'}$ 进行解密,得到洋葱路径 OP。之后,再用节点私钥 NSK_{n+1} 解密对最外层洋葱路径加密的对称密钥 $\{SK\}_{PK(n+1)+}$,再用获得的对称密钥 SK 解密洋葱路径 OP,获得 IP 地址 IP_{n+2} 和接收者用户名 RUN。节点判断其上的用户是否有接收者用户 RUN,若有则把消息转交给接收者用户 RUN,停止转发;否则,继续转发消息。网络中传输的所有信息,包括洋葱路径信息 OP 和消息 P,都是经过加密的,从而有效地防止了窃听和流量分析。

6)返回消息的方式不同。当接收者向发送者返回消息时,消息不是通过原路返回,而是将接收者与发送者调换角色,此时,消息的接收者作为返回消息的发送者,消息的发送者作为返回消息的接收者,将返回消息发送回去。

2.3.2 协议的系统架构

该协议的系统架构由以下几部分组成。

1)节点目录服务器:用来存储双向匿名秘密通信节点群的相关信息,包括 IP 地址、节点公钥。

2)用户目录服务器:用来存储系统用户的相关信息,包括用户名、用户公钥。

3)双向匿名秘密通信群:包括系统中的节点和用户。

假设该系统中注册了 N 个节点,分别为节点 A、节点 B、…、节点 N;注册了 M 个用户,分别为用户 1、用户 2、…、用户 M。为了方便叙述,本节只讨论 $N=M$(节点数与用户数相等),且每个节点上只有一个用户,节点和用户一一对应情况下的协议内容。关于 N 与 M 其他关系情况下的协议内容,将在 2.4 节中进行讨论。协议的系统架构如图 1 所示。

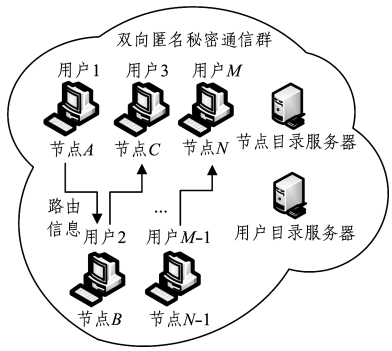

图 1 系统架构图

Fig. 1 System architecture diagram

2.3.3 匿名通信系统的建立与使用过程

该匿名通信系统的建立与使用包括 3 个阶段。

Step 1 匿名通信网络的建立,包括搭建节点目录服务器和用户目录服务器。

Step 2 初始化,包括节点和用户的注册。节点向节点目录服务器进行注册,用户向用户目录服务器进行注册。用户与其所在节点的对应关系是保密的,只有用户自己知道。搭建匿名通信网络之后,想要使用该网络的用户可以在注册的节点上进行注册和登录。若要在未被注册的主机节点上进行匿名交流,则用户需要先注册节点再进行使用。

Step 3 数据传送。数据通过该匿名通信系统传送,可以实现发送者与接收者的双向匿名。

2.3.4 节点与其上用户关系的保密性

每一个用户均可通过节点目录服务器获取该匿名通信系统中所有节点的 IP 地址和节点公钥,并且可以通过用户目录服务器获取该匿名通信系统中所有用户的用户名和用户公钥。但是,每一个用户只知道自己所在的节点,不知道其他用户所在的节点。当用户发送消息时,其知道消息接收者的用户名与其公钥,不知道接收者用户所在的节点;知道消息所经过的洋葱路径,不知道每一个路径节点上的用户名。

假设用户 Alice 发送消息给用户 Bob。Alice 可以通过用户目录服务器获取 Bob 的公钥 PK_{Bob},但不知道用户 Bob 所在的节点。当用户 Alice 想要与用户 Bob 进行匿名通信时,Alice 首先把自己所存储的双向匿名秘密通信群的所有节点信息以随机的顺序构成洋葱路径 OP。洋葱的每一层包括下一跳节点 IP 和接收者用户名 Bob,表示为 $\{IP,\text{Bob}\}$;用 AES 算法生成对称密钥 SK,用 SK 加密该层洋葱 $\{IP,\text{Bob}\}$ 得到 $\{IP,\text{Bob}\}_{SK}$;用对应转发节点的公钥 PK 加密 SK 得到

$\{SK\}_{PK+}$，洋葱的每一层为$\{\{IP,\text{Bob}\}_{SK},\{SK\}_{PK+}\}$。Alice 不知道接收者 Bob 所在的节点信息，洋葱路径的最后一跳不一定是接收者 Bob 所在的节点。该洋葱路径 OP 包括所有的系统节点，而且以随机的顺序排序，接收者 Bob 所在的节点包含在其中，位于构造的洋葱路径的某个节点上。洋葱路径 OP 只包括路径信息，不包括要传递的消息 P。消息 P 用接收者用户 Bob 的公钥 PK_{Bob} 加密为$\{P\}_{PKBob+}$，OP 和 P 连接在一起作为数据部分$\{OP,\{P\}_{PKBob+}\}$进行传递。发送者 Alice 传送数据之前，首先用 AES 算法生成对称密钥 SK'，用 SK'加密数据得到$\{OP,\{P\}_{PKBob+}\}_{SK'}$，然后用下一跳节点的公钥 PK_{n+1}加密 SK'得到$\{SK'\}_{PK(n+1)+}$，再传送报文$\{\{OP,\{P\}_{PKBob+}\}_{SK'},\{SK'\}_{PK(n+1)+}\}$。每一个节点向下一个节点传送数据时，都要先进行加密再传送，以防止流量分析。下一跳节点收到消息时，首先用节点的私钥 NSK_{n+1} 解密被加密的对称密钥$\{SK'\}_{PK(n+1)+}$，获得 SK'，然后用 SK'对消息$\{OP,\{P\}_{PKBob+}\}_{SK'}$进行解密，提取出$\{P\}_{PKBob+}$，获得洋葱路径 OP，用节点私钥 NSK_{n+1} 解密被加密的洋葱最外层的对称密钥 $\{SK\}_{PK(n+1)+}$，得到 SK，接着用 SK 解密 OP，获得下一跳的 IP 地址和接收者用户名 Bob。节点判断接收者用户 Bob 是否为该节点上的用户，若是则将消息转交给 Bob；否则，继续转发消息。节点剥去 OP 的最外一层得到 OP'，将其和 $\{P\}_{PKBob+}$连接在一起得到$\{OP',\{P\}_{PKBob+}\}$；用 AES 算法生成一个对称密钥 SK''，对$\{OP',\{P\}_{PKBob+}\}$进行加密，得到$\{OP',\{P\}_{PKBob+}\}_{SK''}$，再用下一跳节点的公钥 PK_{n+2}加密对称密钥 SK''，得到$\{SK''\}_{PK(n+2)+}$，将数据$\{\{OP',\{P\}_{PKBob+}\}_{SK''},\{SK''\}_{PK(n+2)+}\}$转发出去。当 Bob 想要返回消息给 Alice 时，双方交换角色即可，即 Bob 只需要按照上述步骤发送回复消息给 Alice 即可。

2.3.5　协议的详细描述

下面详细描述基于洋葱路由的双向匿名通信协议[8]的各个阶段。

1）初始化

①节点注册

节点 N 加入匿名秘密通信系统时，首先在本地生成公钥密码对 PK_n 和 SK_n，节点 N 保存私钥 SK_n。节点向节点目录服务器 S_n 发送注册信息，包括节点 N 的公钥 PK_n、IP 地址 IP_n。S_n 收到消息后验证身份，将节点 N 的公钥 PK_n 和 IP 地址 IP_n 添加到 S_n 的注册列表，并把所有注册的节点信息用节点 N 的公钥 PK_n 加密发送给节点 N。S_n 向匿名群的所有节点广播 N 的加入并加上时间戳，以防止重放攻击。节点 N 完成注册。收到节点 N 注册广播的节点将 N 添加到存储的节点列表。节点的注册协议如图 2 所示。

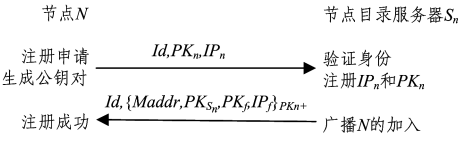

图 2　节点注册协议

Fig. 2　Node registration protocol

其中，PK_f，IP_f 表示节点目录服务器中注册的所有节点信息；Id 表示消息标识符；PK_{S_n} 表示节点目录服务器 S_n 的公钥；M_{addr} 表示节点要加入的广播组；f 表示匿名群的任一节点。

②节点注销

节点 N 要退出匿名通信系统时，向 S_n 发送注销请求。请求消息用 S_n 的公钥加密。S_n 收到请求后，从节点列表中删除 N 并广播 N 的注销，同时向节点 N 返回注销成功。收到节点 N 注销广播的节点从存储的节点列表中删除节点 N。节点的注销协议如图 3 所示。

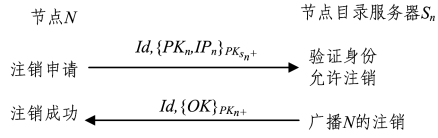

图 3　节点注销协议

Fig. 3　Node cancellation protocol

③用户注册

用户的注册与节点注册类似。用户 Alice 申请注册时，首先在本地生成公钥密码对 PK_a 和 SK_a，用户保存私钥 SK_a。接着，用户向用户目录服务器 S_u 发送注册信息，包括用户的公钥 PK_a、用户名 Alice 和使用用户公钥加密的密码 $\{Password\}_{PKa+}$。S_u 收到消息后，验证身份，将信息添加到用户列表，并向 Alice 返回 S_u 的公钥 PK_{S_u} 以及其它用户信息，同时用 Alice 的公钥加密。S_u 向所有用户广播 Alice 的加入。用户的注册协议如图 4 所示。

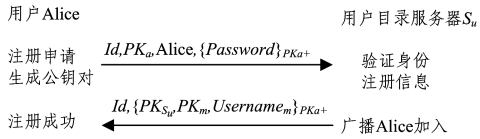

图 4　用户注册协议

Fig. 4　User registration protocol

其中，PK_m，$Username_m$ 表示用户目录服务器中注册的所有用户信息；m 为匿名群中任一用户。

④用户注销

用户的注销与节点注销类似。用户 Alice 想要注销时，向 S_u 发送注销请求。S_u 收到请求后验证身份，注销用户信息，并向用户返回注销成功。S_u 向所有用户广播 Alice 的注销。用户的注销协议如图 5 所示。

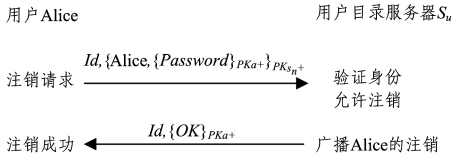

图 5　用户注销协议

Fig. 5　User cancellation protocol

2）发送消息

用户 Alice 想要向用户 Bob 发送消息时，在本地生成包含所有系统节点的洋葱路径 OP。洋葱路径 OP 不包含要发送的消息 P，只包含路径。每层洋葱包括 IP 地址和接收者用户名 Bob，并且用 AES 算法生成的对称密钥 SK 加密，得到

$\{IP,\text{Bob}\}_{SK}$，用对应节点的公钥 PK 加密对称密钥，得到 $\{SK\}_{PK+}$。消息 P 用 Bob 的公钥 PK_{Bob} 加密为 $\{P\}_{PKBob+}$，并将其洋葱路径 OP 一起作为数据部分 $\{OP,\{P\}_{PKBob+}\}$ 进行传递。数据在传送之前首先要生成一个对称密钥 SK'，用对称密钥 SK' 加密数据，得到 $\{OP,\{P\}_{PKBob+}\}_{SK'}$，用下一跳节点的公钥 PK_{n+1} 加密对称密钥 SK'，得到 $\{SK'\}_{PK(n+1)+}$，再转发数据 $\{\{OP,\{P\}_{PKBob+}\}_{SK'},\{SK'\}_{PK(n+1)+}\}$。下一跳节点收到传送的报文时，首先用自己的私钥 NSK_{n+1} 解密被加密的对称密钥 $\{SK'\}_{PK(n+1)+}$，再用获得的对称密钥 SK' 解密被加密的消息 $\{OP,\{P\}_{PKBob+}\}_{SK'}$，得到洋葱路径 OP。然后用自己的私钥 NSK_{n+1} 解密获得用来加密洋葱路径 OP 的对称密钥 SK，再用对称密钥 SK 解密洋葱路径 OP，得到下一跳的 IP 地址和接收者用户名 Bob。节点判断其上的用户是否为 Bob，若是则将消息转交给用户 Bob；否则，生成对称密钥 SK''，加密剥去最外一层的洋葱路径 OP' 和消息 $\{P\}_{PKBob+}$，得到 $\{OP',\{P\}_{PKBob+}\}_{SK''}$，用下一跳节点的公钥 PK_{n+2} 加密对称密钥 SK''，得到 $\{SK''\}_{PK(n+2)+}$，继续转发数据 $\{\{OP',\{P\}_{PKBob+}\}_{SK''},\{SK''\}_{PK(n+2)+}\}$。为了方便说明消息发送过程，假设洋葱路径即如图 1 系统架构所示；并且 $N=M=5$，节点和用户分别为 5 个，用户 1 为发送者 Alice，用户 3 为接收者 Bob，Bob 用户在其路径上的第二跳。Alice 向 Bob 发送消息的过程如图 6 所示。

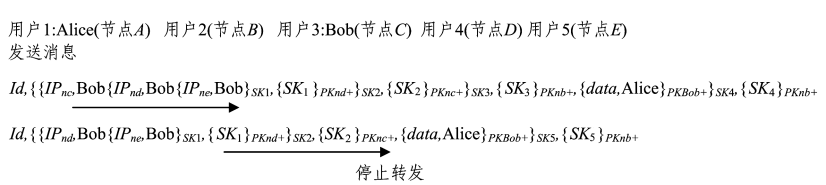

图 6 Alice 向 Bob 发送消息的过程

Fig. 6 Process of Alice sending messages to Bob

路径上的节点不存储任何洋葱路径信息，只转发数据。尽管接收者相同，但发送者每次发送消息时都要重新构建洋葱路径。这种机制能有效抵抗流量分析和中转节点的合谋攻击。

3）返回消息

当接收者用户要返回消息时，发送者与接收者互换角色即可，当作向发送者发送消息处理。接收者用户 Bob 根据接收的消息，可以知道发送者用户名为 Alice。接收者 Bob 向发送者 Alice 返回消息时，Bob 把要返回的消息当作要发送的消息发送给 Alice。此时，Bob 是消息的发送者，Alice 是消息的接收者，消息的处理过程与用户发送消息一样。

匿名通信系统中数据信元的格式[9]如图 7 所示。Command 是命令标志，表示数据负载的作用。

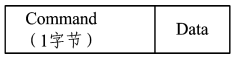

图 7 数据信元格式

Fig. 7 Data cell format

算法 1 表述了节点对接收到的数据包进行处理的过程，其中 Data 表示节点接收到的数据包，Result 表示处理操作。

算法 1 Node Scheduling Algorithm

```
1. WHILE(TRUE)
2.   FOR each received packet,get its command
3.   WITCH(Command)
4.     CASE 1:广播节点加入消息,添加节点信息;
5.           break;
6.     CASE 2:广播节点注销消息,删除节点信息;
7.           break;
8.     CASE 3:广播用户加入消息,添加用户信息;
9.           break;
10.    CASE 4:广播用户注销消息,删除用户信息;
11.          break;
12.    CASE 5:转发数据
13.          IF 节点上的用户包含接收者用户
14.          THEN 停止转发
15.          ELSE 继续转发
16.          END IF
17.          break;
18.  END FOR
19. END WHILE
```

2.4 对协议的进一步讨论

2.3 节就 $N=M$，即节点数与用户数相等，且每个节点上只有一个用户，节点与用户一一对应这种最简单情况下的协议内容进行了说明，下面就 N 与 M 的其他关系下的协议内容进行阐述。

1）$N>M$，即节点数多于用户数，且每个节点上没有用户或只有一个用户。这种情况下的协议内容与 2.3 节描述的不同之处在于：某个节点收到消息，但是该节点上没有用户。该节点收到消息 $\{\{OP,\{P\}_{PKBob+}\}_{SK'},\{SK'\}_{PK(n+1)+}\}$ 后，首先用自己的节点私钥 NSK_{n+1} 解密获得对称密钥 SK'，用对称密钥 SK' 解密消息 $\{OP,\{P\}_{PKBob+}\}_{SK'}$，得到加密的洋葱路径 OP；然后用节点私钥 NSK_{n+1} 解密被加密的用来加密洋葱路径的对称密钥 $\{SK\}_{PK+}$，再用得到的对称密钥 SK 解密洋葱路径 OP，获得下一跳地址和接收者用户名。由于该节点上没有用户，因此直接转发消息，用 AES 算法生成对称密钥 SK''，用对称密钥 SK'' 加密消息得到 $\{OP',\{P\}_{PKBob+}\}_{SK''}$，再用下一跳节点的公钥 PK_{n+2} 加密对称密钥得到 $\{SK''\}_{PK(n+2)+}$，形成报文 $\{\{OP',\{P\}_{PKBob+}\}_{SK''},\{SK''\}_{PK(n+2)+}\}$ 并转发。

2）$N<M$，即用户数多于节点数，且每个节点上有一个用户或多个用户。这种情况下的协议内容与 2.3 节描述的不同之处在于：某个节点收到消息，但是该节点上有多个用户。这时，节点将得到的接收者用户名与该节点上的多个用户进行比较，看是否有接收者用户。若该节点上的多个用户包含接收者用户，则节点不再转发消息；否则节点继续转发消息。

3）无论 N 与 M 的关系如何，在某一时间段上，匿名秘密通信系统中都有可能包含以下 3 种节点：①节点上没有用户；②节点上有一个用户；③节点上有多个用户。结合匿名系统中可能包含的这 3 种节点，综合以上对每种节点的协议讨论，可以得到以下对协议的完整描述：节点收到消息$\{\{OP,\{P\}_{PKRUN+}\}_{SK'},\{SK'\}_{PK(n+1)+}\}$后，用自己的私钥 NSK_{n+1} 解密获得对称密钥 SK'，再用对称密钥 SK' 解密消息$\{OP,\{P\}_{PKRUN+}\}_{SK'}$，得到洋葱路径 OP。同样地，节点用自己的私钥 NSK_{n+1} 解密获得洋葱路径 OP 的最外层对称密钥 SK，再用对称密钥 SK 解密洋葱路径 OP，得到下一跳的 IP 地址和接收者用户名 RUN。然后节点查看自己的用户表上是否有接收者用户 RUN，若有则节点不再转发消息；否则，节点继续转发消息。

2.5　匿名性证明

双向匿名秘密通信协议中，发送者构造包含所有节点的洋葱路径，洋葱的每一层包括下一跳中转节点和接收者用户名，洋葱路径不包含消息，洋葱路径和消息连接在一起进行传送。消息按照洋葱路径传送，每到一跳中转节点，节点剥离最外层洋葱，得到接收者用户名和下一跳 IP 地址。节点根据得到的接收者用户名判断节点上的用户是否包含接收者用户。若包含，则节点上的接收者用户接收消息，节点停止转发消息；否则，节点根据得到的下一跳 IP 地址，继续转发消息。只有接收者用户才能够获取消息，任何其他用户或者节点无法获取消息。传送的消息中包含发送者用户名，其用接收者用户的公钥加密。当接收者用户要返回消息时，发送者与接收者互换角色即可。本文的双向匿名秘密通信协议既实现了发送者匿名，又实现了接收者匿名。下面分别从发送者匿名和接收者匿名两方面来论证协议的匿名性。

1）发送者匿名

对于发送者匿名来说，接收者用户接收到消息，通过解密知道发送者的用户名、上一跳转发节点和下一跳转发节点，但是不可能知道发送者用户位于哪一个节点上。

路由节点对得到的洋葱路径进行解密，获得下一跳地址和接收者用户名。中转节点只知道上一跳路由、下一跳路由和接收者用户名，不可能知道发送者位于哪一个节点，因此，所有中转节点都不可能获知发送者的身份。发送者发送的消息和发送者用户名用接收者用户的公钥加密，中转节点不可能获取发送的消息和发送者用户名，从而实现了发送者匿名。

2）接收者匿名

对于接收者匿名来说，发送者虽然可以从节点目录服务器和用户目录服务器获得系统中所有的节点信息和用户信息，但是其除了知道本身所位于的节点地址，不可能知道其他用户位于哪个节点上，即用户和其所在节点的对应关系。发送者只知道其想要发送消息的用户名，即接收者用户名，不可能知道其所在的节点地址，即接收者地址。

发送者构造包含所有系统节点的洋葱路径，消息按照洋葱路径进行传送。洋葱路径包括接收者节点，消息一定会路由到接收者节点。发送者虽然并不知道接收者位于哪一个节点，但仍然可以将消息发送给接收者并确保接收者可以收到。

路由节点对得到的洋葱路径进行解密，获得下一跳地址和接收者用户名。中转节点只知道上一跳路由、下一跳路由和接收者用户名，不可能知道接收者位于哪一个节点，因此，所有中转节点都不可能获知接收者的身份，从而实现了接收者匿名。

2.6　消息保密性证明

发送者发送的消息以及发送者用户名用接收者用户的公钥加密，然后和洋葱路径链接在一起用下一跳路由的公钥加密后再进行传送。每到一跳路由节点，节点首先用其私钥解密，然后剥离最外层洋葱路径获得接收者用户名。节点判断其上用户是否包含接收者用户，若包含，则停止转发，将消息转交给接收者用户；否则，继续转发消息。

节点只将消息转交给其上的合法接收者用户，只有合法的接收者用户才能得到消息。消息用接收者用户的公钥加密，只有合法的接收者用户用其私钥才可以解密消息。每到一跳中转节点，节点用其私钥解密，得到的是用接收者用户的公钥加密的消息。所有的中转节点都不能获取消息。消息不仅用接收者用户的公钥加密，在转发过程中，还要用下一跳节点的公钥进行加密，因此窃听者或者恶意节点和用户不能获取到消息。由此可知，通信双方传送的消息是保密的，除了发送者和接收者之外，任意的中转节点和侵入者都不能获取消息。

2.7　恶意节点攻击的防御

消息在节点之间传递时都经过了加密，有效地防止了流量分析。下面对恶意节点攻击进行分析。

1）破坏发送者匿名

若恶意节点分布符合以下情况，则打破消息的发送者匿名。

①若恶意节点中包含发送者节点，则破坏从该节点发送消息的发送者匿名；

②若恶意节点中不包含发送者节点和链路上的第一个中转节点，则一条转发链路上至少有一半的节点是恶意节点（好的节点不相邻），且非转发链路上的所有节点都为恶意节点才能打破该条消息的发送者匿名，并打破该条消息的接收者匿名。

2）破坏接收者匿名

若恶意节点分布符合以下情况，则打破消息的接收者匿名。

①若恶意节点中包含接收者节点，则破坏该节点接收消息的接收者匿名；

②若恶意节点不包含接收者节点，则一条转发链路上至少有一半的节点是恶意节点（好的节点不相邻），且非转发链路上的所有节点都为恶意节点才能打破该条消息的接收者匿名，并打破该条消息的发送者匿名。

3）破坏发送者与接收者双向匿名

若恶意节点分布符合以下情况，则打破消息的双向匿名。

①若恶意节点中包含发送者节点和接收者节点，则破坏该条消息的双向匿名通信；

②若恶意节点既不包含消息的发送者，又不包含消息的

接收者，但至少包括该条消息链路上一半的中转节点(好的节点不相邻)，且包括非转发链路上的所有节点，则打破该条消息的双向匿名；

③若恶意节点包含接收者节点，不包含发送者节点，且至少包括一条链路上一半的中转节点(好的节点不相邻)，则破坏该条消息的双向匿名通信；

④若恶意节点包含发送者节点，不包含接收者节点，但至少包括该条消息链路上一半的中转节点(好的节点不相邻)，且包括非转发链路上的所有节点，则打破该条消息的双向匿名。

对于发送者任意构造的一条消息传送链路，接收者用户可能位于链路上的第 1 个节点、第 2 个节点…第 $N-1$ 个节点上，对应的转发次数分别为 0 次、1 次…$N-2$ 次。因此，对于一条消息，平均的中转次数 $C=(N-2)/2$。

若恶意节点包含接收者节点 r，不包含发送者节点 s，且至少包括一条链路上一半的中转节点(好的节点不相邻)，则打破一条消息的双向匿名通信的概率 Pr 如下。

假设一条消息的转发次数取平均转发次数 C，则链路长度 L 为 $C+1$；恶意节点集合 B 中的节点数目为 V。Pr 的表达式如下：

$$Pr=\frac{C_{N-L/2}^{V-L/2}}{C_N^V},N\geqslant V\geqslant L/2 \tag{1}$$

当 $N=20$ 时，$L=10$；取 V 分别为 5，10，15，20 时，对应的 Pr 分别为 1/15 504，1/323，1/5，1。对于其他情况，不仅需要恶意节点集合 B 至少包括一条链路上一半的中转节点(好的节点不相邻)，而且还需要非转发链路上的所有节点都为恶意节点，才能打破系统的双向匿名通信，所以更难打破系统的匿名性。其他情况的概率计算与此情况类似。

通过以上分析可以得到，随着 N 的增大，平均转发次数 L 线性增长。一般情况下，要想打破系统的双向匿名，系统中的恶意节点数 M 必须大于或等于 $L/2$。当 $V=L/2$ 时，概率 Pr 非常小；随着 V 的增大，Pr 逐渐增大；当 $V=N$ 时，Pr 为 1。因此，为了打破该系统的匿名性，随着 N 的增加，必须增加 V 的数量。由此可以得到，随着 N 的增加，系统的匿名度增大。

3 实验

3.1 实验环境及性能指标

实验的硬件环境：CPU 为 Intel(R) Core(TM)i5-3230M；主频为 2.60 GHz；内存为 4.00 GB；操作系统为 Windows 7。实验的软件环境：eclipse 平台，项目的执行环境为 JavaSE-1.7。

3.2 实验结果及分析

图 8 和图 9 分别为节点数目变化时得到的平均响应时间和平均双向通信时间。从中可以看到，随着节点数目的增多，平均响应时间和平均双向通信时间近似呈线性增加态势。随着系统中节点数目的增加，系统运行稳定，健壮性较好。在节点数目相同时，平均双向通信时间大约为平均响应时间的两倍。

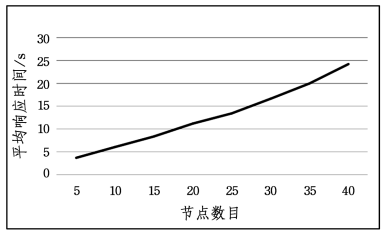

图 8 节点数目变化时的平均响应时间

Fig. 8 Average response time when number of nodes changes

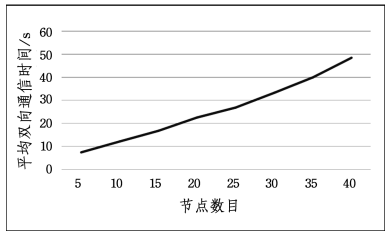

图 9 节点数目变化时的平均双向通信时间

Fig. 9 Average bidirectional communication time when number of nodes changes

图 10 给出节点数目变化时得到的平均转发次数。从中可以看到，节点数目与平均转发次数之间近似呈线性关系，平均转发次数接近于节点数目的一半。

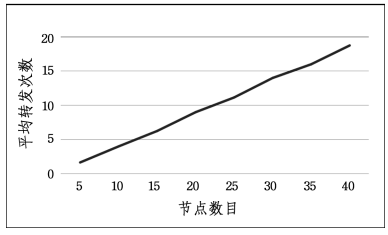

图 10 节点数目变化时的平均转发次数

Fig. 10 Average number of forwarding times when number of nodes changes

图 11 给出恶意节点数目变化时得到的匿名度。匿名度为通信双方不被恶意节点发现的概率。

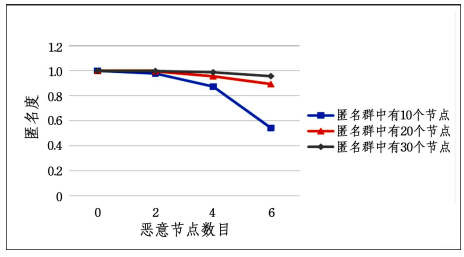

图 11 恶意节点数目变化时的匿名度

Fig. 11 Anonymity when number of malicious nodes varies

图 11 中 3 条曲线分别代表匿名群中节点数目为 10，20 和 30 时匿名度随着恶意节点数目改变的变化情况。从中可以看到，随着恶意节点数目变小，匿名度逐渐变大。同时，随着匿名群中节点数目的增多，匿名度也在增大。当系统中存在恶意节点，且系统节点较多时，系统的匿名度都在 80% 以上，达到了匿名性的 Beyond Suspicion 分类等级，攻击者不能

猜测谁是真正的通信双方，系统具有较好的匿名性。

图 12 给出节点数目变化时得到的本文协议、BACP 协议[5]和 H-Tor 协议[6]的平均双向匿名通信节点的负载和。节点处理转发消息的次数为该节点的负载。BACP 协议中，假设消息在 Crowds 群中随机转发的概率为最小值 0.5。H-Tor 协议中，假设消息在 Hordes 群中随机转发的概率值为 0.5。从图 12 中可以看出，3 个协议的平均双向匿名通信节点的负载和都随着节点数目的增多近似呈线性增长，但是本文协议增长的速度最慢。当系统中节点数目一定时，本文协议的平均双向匿名通信节点负载和几乎是 H-Tor 协议的 1/2，是 BACP 协议的 1/3。由此可以得到，与 H-Tor 协议和 BACP 协议相比，本文提出的双向匿名秘密通信协议通信时对系统资源的消耗比较少，系统中产生的流量较少。

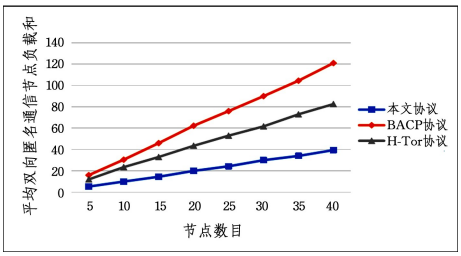

图 12　节点数目变化时的平均双向匿名通信节点负载和

Fig. 12　Average bidirectional anonymous communication node load sum when number of nodes changes

结束语　本文提出基于洋葱路由的双向匿名秘密通信协议，构造的洋葱路径包含匿名群的所有节点，处于洋葱路径上的接收者用户接收消息，从而任何其他用户或者节点都不能获取消息，实现了双向匿名秘密通信。实验结果表明：该协议具有可行性，可以有效地减少系统中的流量，且随着系统中节点数目的增加，系统的匿名性增强。本文工作为双向匿名秘密通信的研究提供了新思路。下一步可以就如何改善协议并提高通信效率进行研究。

参 考 文 献

[1] CHAUM D. Untraceable electronic mail, return addresses, and digital pseudonyms[J]. Communications of the ACM, 1981, 24(2): 84-88.

[2] REED M G, SYVERSON P F, GOLDSCHLAG D M, et al. Anonymous connections and onion routing[J]. IEEE Journal on Selected Areas in Communications, 1998, 16(4): 482-494.

[3] DINGLEDINE R, MATHEWSON N, SYVERSON P. Tor: The second-generation onion router[C]// Proceedings of the 13th USENIX Security Symposium. San Diego, USA, 2004.

[4] WU Q Z, XIE S X, JIA Y L. An improved Tor anonymous communication system[J]. Science and Technology Information, 2012(19): 125-126. (in Chinese)
吴庆震，谢圣献，贾仰理. Tor 系统的改进方案[J]. 科技信息，2012(19)：125-126.

[5] HUO C Y, WU Z Q. Study and design of bidirectional anonymity communication protocol[J]. Computer Engineering, 2008, 34(19): 174-178. (in Chinese)
霍成义，吴振强. 双向匿名通信协议的研究与设计[J]. 计算机工程，2008，34(19)：174-178.

[6] ZHENG G, XUE Z. A mixed anonymous system based on Tor[J]. Information Security and Communications Privacy, 2011(12): 76-80. (in Chinese)
郑光，薛质. 基于 Tor 的混合匿名转发系统[J]. 信息安全与通信保密，2011(12)：76-80.

[7] ZHU Y, FU X W, GRAHAM B, et al. On flow correlation attacks and Countermeasures in mix networks[C]// Proceedings of the 4th Privacy Enhancing Technology Workshop. Toronto, CANADA, 2005: 207-225.

[8] ZHAO F X, WANG Y M, WANG C J. An authenticated scheme of onion routing[J]. Chinese Journal of Computers, 2001, 24(5): 463-467. (in Chinese)
赵福祥，王育民，王常杰. 可靠洋葱路由方案的设计与实现[J]. 计算机学报，2001，24(5)：463-467.

[9] HE G F, YANG M, LUO J Z, et al. Modeling and analysis of time characteristics used in onion routing traceback techniques[J]. Chinese Journal of Computers, 2014, 37(2): 356-372. (in Chinese)
何高峰，杨明，罗军舟，等. 洋葱路由追踪技术中时间特征的建模与分析[J]. 计算机学报，2014，37(2)：356-372.