基于安全博弈论的中间人攻击防御策略

Defense Strategy against Man-In-The-Middle Attack based on Security Game Theory

工程领域：计算机技术
作者姓名：李姝昕
指导教师：李晓红 教授
企业导师：郭晓和 正高工

天津大学计算机科学与技术学院
二零一七年十一月

独创性声明

本人声明所呈交的学位论文是本人在导师指导下进行的研究工作和取得的研究成果，除了文中特别加以标注和致谢之处外，论文中不包含其他人已经发表或撰写过的研究成果，也不包含为获得 天津大学 或其他教育机构的学位或证书而使用过的材料。与我一同工作的同志对本研究所做的任何贡献均已在论文中作了明确的说明并表示了谢意。

学位论文作者签名: 李姝昕　　签字日期: 2017 年 12 月 11 日

学位论文版权使用授权书

本学位论文作者完全了解 天津大学 有关保留、使用学位论文的规定。特授权 天津大学 可以将学位论文的全部或部分内容编入有关数据库进行检索，并采用影印、缩印或扫描等复制手段保存、汇编以供查阅和借阅。同意学校向国家有关部门或机构送交论文的复印件和磁盘。

(保密的学位论文在解密后适用本授权说明)

学位论文作者签名: 李姝昕　　导师签名: 李佳

签字日期: 2017 年 12 月 11 日　　签字日期: 2017 年 12 月 11 日

摘　　要

中间人攻击是一种常见的网络攻击方法，攻击者可以通过使用一些技术手段获取用户的隐私敏感信息，从而给用户造成一定的财产或是人身安全隐患。因此，目前针对中间人攻击防御问题的研究具有重要的意义并受到了广泛的关注。

本文基于安全博弈论的相关理论从一个全新的方面研究中间人攻击防御问题——假设在中间人攻击不可避免的情况下，以降低防御方遭受攻击时的损失为目标，从整体和个体两个角度出发研究如何为防御方设计最优防御策略。从整体角度出发是将服务器与用户看作一个整体作为防御方，将防御方和攻击者之间的交互过程建模为Stackelberg博弈模型并采用强Stackelberg均衡作为最优防御策略。为了能够高效求解防御策略，提出了一种减小计算最优防御策略时的搜索空间的新方法。从个体角度出发是把每个用户看作一个独立的防御方，使用同时行动博弈模型对多个防御方与攻击者之间的交互过程进行建模并采用模型的Nash均衡策略作为最优防御策略。为了能够在实际中求解出Nash均衡策略，分别为防御方和攻击者设计了自适应的学习算法，使得双方能够在重复交互过程中学习收敛到Nash均衡。最后实验结果表明从整体角度出发所设计的防御方最优防御策略在降低整体损失方面明显优于其他非策略性的防御策略，从个体角度出发所提出的学习算法能够保证防御方策略收敛到Nash均衡，攻击者的期望收益稳定在Nash均衡。

基于安全博弈论的相关理论研究中间人攻击防御问题，不仅是对中间人攻击防御问题研究方面的一个补充，对安全博弈论的发展具有一定的理论意义，也为安全博弈论的应用提供了新的发展方向。

关键词： 中间人攻击，安全博弈论，Stackelberg模型，强Stackelberg均衡，Nash均衡

ABSTRACT

The man-in-the-middle attack is a common method of cyber attack. The attacker can obtain the sensitive information of users by applying some technologies, that brings certain economic losses and security issues to users. Therefore, it is of great significance to study the man-in-the-middle attack defense problem and it has attracted extensive attention.

This paper studies the man-in-the-middle attack defense problem from a novel perspective based on the security game theory. Given that the man-in-the-middle attacks are inevitable, we study how to design the optimal strategy from the perspective of the whole and the individual with the goal of reducing the defender's losses when it suffers from attack. From an overall point of view, the server and all users are considered as a whole which regarded as a defender. The interaction between the attacker and the defender is modeled as a Stackelberg security game, and strong Stackelberg equilibrium is adopted as the defender's optimal defense strategy. In order to compute the optimal defense strategy efficiently, a novel method is proposed to reduce the searching space of computing the optimal defense strategy. From an individual point of view, each user is treated as a defender. The simultaneous-move game is used to model the interaction between multiple defenders and an attacker, and Nash equilibrium is adopted as the optimal defense strategy. In order to solve Nash equilibrium in practice, we proposes practical adaptive algorithms for the defender and the attacker which enable the both sides learn towards Nash equilibrium through repeated interaction. Finally, the experiment results indicate that optimal defense strategy which is designed from the perspective of the whole significantly outperforms other non-strategic defense strategies in terms of decreasing the total losses against the man-in-the-middle attack. It also shows that the learning algorithms proposed from the perspective of the individual can ensure the strategy of the defender converges to Nash equilibrium and the expected payoff of the attacker approximates the expected payoff in Nash equilibrium.

This paper uses the security game theory to study the man-in-the-middle attack defense problem. Not only does it complement the existing man-in-the-middle attack defense approaches and have certain theoretical significance for the development of security game theory, but also it provides a new development direction of the security

game theory.

KEY WORDS: Man-In-The-Middle Attack, Security Game Theory, Stackelberg Game, Strong Stackelberg Equilibrium, Nash Equilibrium

目　录

第1章 绪论

1.1 研究背景及意义

近年来，互联网的快速发展给人们的生活以及工作带来了许多便利，但是越来越多的网络安全问题层出不穷。目前，人们通过网络进行着越来越多的涉及隐私或是财产信息的交易，一旦遭遇到网络攻击，攻击者便可从中获取到这些交易信息，这将会对人们的人身和经济财产安全构成极大的威胁。而且，人们越来越关心隐私信息保护的问题，从而使得对于网络攻击防御问题的研究受到越来越多的关注。中间人攻击（Man-In-The-Middle Attack，MITM）就是攻击者窃取用户敏感信息较常用的一种网络攻击方式。所谓中间人攻击就是攻击者与原本正常通讯的用户分别建立通讯连接，然而正常通讯的用户并不知道，仍旧认为自己还在跟对方直接通讯，但实际上他们的通讯已经被攻击者控制，这样攻击者就可以获取用户的通讯信息或者对信息进行篡改[1]。

其实中间人攻击由来已久，离我们并不遥远就存在于我们的生活中。据报道称，2013年1月26日，中国大陆的用户访问GitHub.com时曾遇到SSL证书无效警告的情况，之后的证据表明这是一起中间人攻击事件。2014年，Google IPv6教育网、Yahoo和微软也都曾遭遇过SSL证书中间人攻击。而且根据相关报道，目前95%的HTTPS服务器由于没有正确实施HSTS（HTTP Strict Transport Security）安全功能很容易遭受中间人攻击。因为如果没有正确实施HSTS安全功能，用户可能就是通过HTTP协议访问服务器，那么攻击者甚至不需要伪造TLS证书就可以实施中间人攻击。随着物联网技术慢慢进入我们的生活，一些物联网设备也同样可能遭遇中间人攻击。据报道称，中间人攻击者可以对智能汽车进行中间人攻击，从而控制汽车的一些基本功能，例如刹车、转弯和加速等。因此，中间人攻击对人们的隐私信息甚至生命安全造成了极大的威胁，针对中间人攻击防御问题的研究刻不容缓。

安全博弈论的早期研究主要是为了保护关键公共基础设施，研究如何进行有限安全资源的最优部署。安全博弈论是一个以实际应用为导向的研究领域，通常使用Stacelberg博弈模型对安全部门和攻击者之间的交互进行建模，过去几年中，美国不同领域的安全机构已经开始使用基于Stacelberg安全博弈框架设计的实际应用系统[2]。目前，其他更多安全领域也开始尝试使用安全博弈论来解决问题，一个新的应用领域就是网络攻击的防御问题[3]。网络攻击防御问题中的攻击

者和防御方之间的交互过程可以被建模为一个两个参与者的博弈，其研究的主要目标是为防御方设计最优的防御策略。目前随着应用领域的不同以及实际问题规模的增大，现有的求解算法不可能适应于所有安全问题的求解，亟需设计出能够解决针对不同问题或是大规模问题的高效求解算法。

本文应用安全博弈论的相关理论研究中间人攻击防御问题，这是对中间人攻击防御问题方面研究的一个补充；其次，使用安全博弈论的相关理论来解决网络攻击问题，为安全博弈论的应用提供了新的发展方向；最后，针对所建立的博弈模型进行了理论分析并提出了有效的求解算法，对安全博弈论的发展具有一定的理论意义。

1.2 国内外研究现状

目前，国内外有很多对于不同类型的中间人攻击进行防御的研究[4–6]。针对中间人攻击防御措施的研究主要有以下两个方面，一个方面的研究旨在增加攻击的难度，例如使用复杂的加密算法对通讯数据进行加密或是在密钥交换阶段增强其安全性，这样即使攻击者获得了通讯数据包也很难进行解密，从而获取不到用户通讯时的隐私信息[7,8]。但是随着时间的推移和攻击者计算能力的增加，加密算法还是可能会被破译。因此，此类的防御措施只是增加了攻击者的攻击难度，并不能完全消除中间人攻击的存在。

另一方面的研究就是对中间人攻击的实时检测，检测到攻击后就可以采取相应的防御措施。国内外有很多针对中间人攻击检测方法的研究，常用的检测方法有数字证书的验证以及利用通讯数据包的一些特性来进行检测。郭卫兴等人[9]通过对内部网络通讯危害较大的ARP欺骗技术的分析，提出了一种交换网络环境下基于ARP缓存超时机制的中间人攻击检测方法；Vallivaara等人[10]通过利用TCP数据包报头中的时间戳信息，提出了一个检测中间人攻击的方法；Dacosta等人[11]提出了一种直接验证证书的方法，不需要使用第三方就可以对中间人进行有效的检测；Huang等人[12]通过对虚假SSL证书的分析，提出了一种针对大规模网站的SSL中间人攻击的检测方法。一旦检测到中间人攻击的存在，例如用户端收到的数字证书与网站合法数字证书不匹配，那么就会尝试重新连接或是其他防御措施来抵御中间人攻击[13,14]。

安全博弈论早期的研究是为安全部门解决安全领域中的有限安全资源的分配问题，也就是如何将有限的安全资源进行最佳配置以获得最优的安全保护[15]。Stackelberg博弈模型是安全博弈论中经常使用的对将防御方和攻击者的交互过程进行建模的博弈模型。虽然该博弈模型在20世纪30年代就已经被提出来[16]，但是直到2006年，Conitzer和Sandolm发表了奠基性论文后，Stackelberg博弈模型才

开始广泛应用于安全领域中有限资源的优化调度问题[17]。近些年来，关于安全博弈论的应用研究受到越来越多的关注并取得了很大的进展，特别是在防御物理攻击保护关键公共基础设施方面[18–21]。

近年来，研究者们不断提出适应于不同安全问题场景的安全博弈模型及其最优策略的求解算法。目前，安全博弈论也开始应用于一些新兴领域，例如保护大规模的城市网络，如电力网络[22]、计算机网络、交通网络和一些可以用网络来建模的问题。Tsai等人[23]就将孟买警方在城市道路上设置车辆检查点的问题建模为城市道路网络上孟买警方与恐怖袭击者之间的安全博弈问题，并且针对这个大规模的实际问题提出了新的Stackelberg博弈均衡策略求解算法。还有一些研究将安全博弈论应用于保护自然资源领域，例如保护森林资源免受乱砍乱伐[24]、保护濒危物种[25]、保护海洋的鱼类资源[26]等。目前，也有一些研究开始将安全博弈论应用到网络攻击防御领域[27,28]，例如Laszka等人[29,30]将安全博弈论应用于防御钓鱼邮件攻击，通过建立Stackelberg博弈模型，为防范钓鱼邮件攻击制定了最优的防御策略。但是，目前还没有使用安全博弈论的方法来解决中间人攻击防御问题的相关研究。

然而，安全博弈论的研究中并不是全部基于Stackelberg博弈模型进行建模，因为Stackelberg博弈模型并不可能适用于全部的安全问题场景。针对不同的安全问题场景，需要寻找合适的博弈模型对其进行建模，例如当攻击者并不能提前获取防御方的防御策略时，同时行动博弈模型会更适合对这种情况进行建模，或者当攻击可能会重复出现时，就可以使用重复博弈的模型对该问题进行建模。目前，对于在重复博弈过程中如何设计防御方的策略也有一些相关的研究。由于在重复博弈过程中，防御方可以利用之前博弈的一些信息来帮助自己在下一次的博弈中做决策，因此这部分的研究在设计防御策略时大多会涉及一些学习的算法。徐海峰等人[31]在没有博弈先验信息的情况下，通过建立重复安全博弈模型，设计了一个在线对抗学习框架来计算有效的防御策略；Klíma等人[32,33]为解决边境安全资源分配问题，将攻击者与防御方之间的交互过程建模为重复博弈模型，防御方利用交互的历史信息使用一些在线学习算法进行学习，从而计算出自己的最佳响应策略也就是最优防御策略。

1.3 研究内容

目前，针对中间人攻击的防御方法主要集中在增加攻击者攻击的难度或是对中间人攻击进行实时检测。这些防御措施并不能完全消除中间人攻击的存在，一旦被攻击者发现漏洞还是会马上遭到攻击。如果攻击者只是单纯地获取敏感信息而不进行修改的话，一些检测方法也很难成功对攻击进行检测。基于上述分析，

本文利用安全博弈论的相关理论从一个全新的角度研究中间人攻击防御问题。

假设中间人攻击不可避免的情况下，分别从整体角度和个体角度出发，以降低遭受攻击时防御方所承受的损失为目标，为防御方设计最优的防御策略。通过使用博弈模型对中间人攻击问题不同情景进行建模，针对不同的模型提出相应的高效求解算法，进而得到防御方的最优防御策略，主要研究内容如下：

从整体角度出发，以降低所有用户的整体损失为目标设计防御策略。使用Stackelberg博弈模型对防御方与攻击者之间的交互过程进行建模，采用强Stackelberg均衡作为防御方的最优防御策略，通过提出一个降低计算最优防御策略搜索空间的算法对最优防御策略进行求解，最后通过实验对比来验证最优防御策略的有效性。

从个体角度出发，以降低用户自身损失为目标设计防御策略。将多个用户和一个攻击者之间的交互过程建模为同时行动博弈模型，采用Nash均衡作为防御方的最优防御策略，通过为防御方和攻击者分别设计自适应学习算法，使得防御方能够学习收敛到最优防御策略，最后通过实验对学习算法进行评估。

1.4 本文的组织结构

本文梳理了国内外关于中间人攻击防御措施以及安全博弈论的研究现状，通过对现有中间人攻击防御问题研究的分析，提出从一个新的角度来研究中间人攻击防御问题，之后分别从整体角度和个体角度出发，构建不同的安全博弈模型，并对模型进行理论分析，最后提出了有效的求解算法和相应的最优防御策略。本文的结构安排如下：

第一章介绍了中间人攻击防御问题的研究背景及意义，分析了中间人攻击防御问题以及安全博弈论的国内外研究现状，提出了研究的主要内容，最后介绍了文章的组织结构。

第二章对博弈论的相关理论、中间人攻击的相关技术以及防御问题进行了简单介绍。首先介绍了博弈论的一些基本概念以及安全博弈论的相关基础知识；然后介绍了中间人攻击的相关技术；最后对中间人攻击防御问题进行了简单描述。

第三章介绍了从整体的角度出发，如何为防御方设计最优防御策略。首先介绍了Stackelberg博弈模型；然后对所建立的模型进行了理论分析，提出了一种降低计算最优防御策略时搜索空间的方法，从而能够高效求解防御方的最优防御策略；最后实验结果表明最优防御策略在降低整体损失方面明显优于其他非策略性的防御策略。

第四章从个体的角度出发，介绍了如何为独立的防御方设计最优防御策略。首先介绍了同时行动博弈模型；然后对博弈模型进行了理论分析，证明了模

型Nash均衡的存在唯一性；之后介绍了防御方与攻击者的自适应学习算法；最后通过实验表明学习算法能够保证防御方策略收敛到Nash均衡，攻击者的期望收益稳定在Nash均衡。

第五章首先对研究工作进行了总结，之后提出了几点未来进一步工作的研究方向和目标。

第2章 相关理论及技术

为了更好地理解研究工作，本章主要介绍了一些相关的理论基础和技术。首先对博弈论的基本概念、安全博弈论的相关理论、中间人攻击的相关技术等基础知识进行简单介绍，之后详细介绍了研究中所采用的中间人攻击防御措施和所要解决的关键问题和研究难点。

2.1 博弈论

2.1.1 博弈论的基本概念

在实际生活中，人们的行为决策可能会互相影响，所以人们在进行决策时要考虑到对方的行为影响。博弈论研究的基本假设就是参与博弈的每个人都是理性的，即他能够在充分考虑了人们之间行为的相互作用及其可能的影响之后，做出合乎理性的选择。特别需要注意的是博弈论中理性的假设不仅指参与博弈的每个人都是理性的，而且彼此也都知道对方是理性的。所谓合乎理性是指参与博弈的个体为了最大化自己的收益目标函数，通常选择使其收益最大化的策略[34]。从实际生活来看，人们作为一个独立的行为主体在生活的不同情形下都有自己内心的收益函数或是目标函数。当面临策略选择问题时，抛开情感道德规范等因素的影响，都会倾向于选择使其收益最大化的最佳策略，因此博弈论的研究假设是符合大众心理的。一个完整的博弈过程通常包括参与者、行动集、收益、信息和均衡等几个基本要素，下面对博弈论的几个重要概念给出简单的解释。

参与者是指参与博弈过程的能够独立做决策的理性主体，参与博弈的参与者可以是一个单独的个体也可以是一个群体[35]。根据博弈论研究中关于参与者的理性假设可知，一个理性参与者会以实现自身收益的最大化为目标来选择自己合乎理性的行动。

行动集指参与者在博弈过程中可以执行的全部行动所组成的集合。在博弈过程中，参与者需要从其行动集中做出选择，也就是参与者需要选择一个行动作为他当前博弈的策略。一个博弈过程中，当每一个参与者都选定了一个策略，所有参与者选择的策略就构成了一个策略组合，也称为“局势”[36]。在博弈论中，当给定其他参与者策略不变的情况下，参与者的能够最大化其自身收益函数的策略称为其最佳响应策略。

收益是参与者在博弈过程中所能够获得的利益或是效用[37]。在博弈过程中各参与者之间的行为决策互为影响，因此参与者的收益不仅仅决定于自身的行动选择，而是取决于所有参与者选择的行动策略，所以参与者的收益通常是策略组合的函数。

信息是指在博弈过程中参与者在进行决策时所能够了解的有关其他参与者的信息，例如其他参与者的行动集、收益函数等知识信息。博弈论中的信息主要包括两类：完全信息和完美信息。完全信息是指博弈过程中所有参与者的行动集合、收益函数等全部的博弈信息，而完美信息则是指参与者获取到的在已发生博弈中所有参与者的策略信息，也就是之前博弈的所有信息[38]。

均衡是指一个博弈中由所有参与者的最佳响应策略所组成的一个策略组合，是博弈过程中最可能出现的结果[39]。均衡是博弈论研究中最重要、最基础的解概念，针对不同的博弈问题，研究者最关心的问题就是如何找出博弈的均衡解概念。Nash均衡、Stackelberg均衡等就是针对不同类型的博弈问题所形成的特定的均衡解概念。

2.1.2 安全博弈论

安全博弈论是博弈论在安全领域的有限安全资源分配问题及调度方面应用而产生的理论[15]，它是一个基于计算和行为博弈理论的一个新的研究领域，同时还结合了机器学习以及不确定情况下人工智能规划的相关理论元素。

在安全博弈论中，Stackelberg博弈模型通常被用于对安全领域中的攻击防御类的交互行为进行建模，此类Stackelberg博弈模型就称为Stackelberg安全博弈模型。经典的Stackelberg博弈通常是一个由两个参与者组成动态博弈过程，即一个领导者和一个跟随者，跟随者在做决策前能够观察到领导者的策略。首先，领导者从自己的行动策略集中选择策略，跟随者在观察了领导者的策略之后，从自己的行动策略集中选择自己的最佳响应策略作为自己的策略，也就是最大化自己利益的策略。Stackelberg博弈中所描述的场景很好地刻画了安全领域中安全部门所遇到的安全问题。首先，安全部门需要对所要保护的目标部署好自己的防御策略，例如巡视策略，攻击者在实施攻击前则会通过观察、监视等方式来了解安全部门的防御策略，之后选择能够最大化自己期望收益的目标进行攻击。因此，在Stackelberg安全博弈模型中，一般将防御方看作是领导者，首先选择策略，将攻击者看作是跟随者，在观察了防御方的策略之后选择自己的攻击策略。图2-1给出了Stackelberg安全博弈模型的一个简单示例。

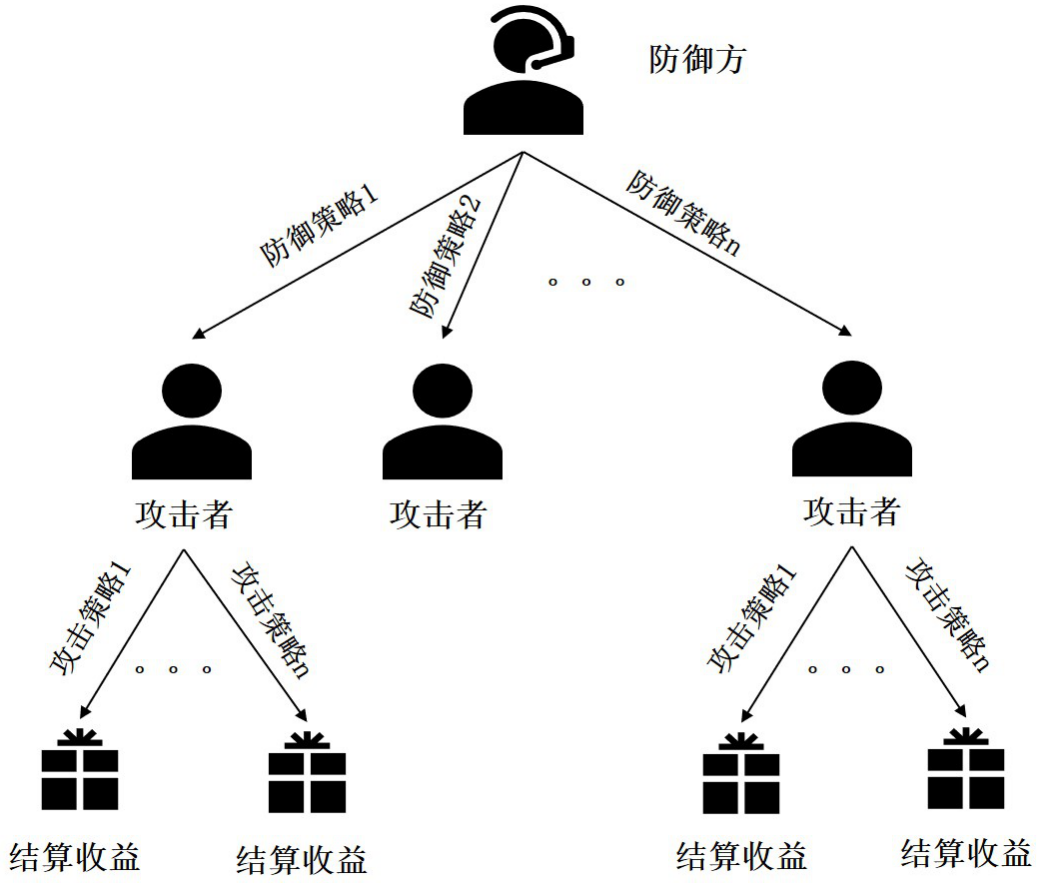


图 2-1 Stackelberg安全博弈模型示例

在博弈论中，一个博弈最重要、最基础的解概念就是均衡，最常见的就是Nash均衡。在Nash均衡下，所有参与者的策略都是自己的最佳响应策略，也就是任何人都不可能通过单方面改变自己的策略来增加自己的收益[40]。Stackelberg均衡是在Stackelberg博弈中Nash均衡的一种精炼，它是一种子博弈完美均衡[3]。在Stackelberg均衡下，每个参与者在原博弈的每个子博弈中都会选择自己的最佳响应策略。但是当多个策略对于攻击者来说没有区别时，并不能保证博弈有唯一的Stackelberg均衡。为了解决这个问题，Leitmann提出了两种Stackelberg均衡的概念[41]，随后被Breton等人命名为“强Stackelberg均衡”和“弱Stackelberg均衡”[42]。强Stackelberg均衡是指当攻击者在多个策略下收益相同时，假设攻击者选择对防御方最有利的策略作为其均衡策略，而弱Stackelberg均衡则假设攻击者选择对防御方最不利的策略作为其均衡策略[42]。强Stackelberg均衡在所有Stackelberg博弈中都是存在的，而弱Stackelberg均衡却不一定存在。在安全博弈论的相关研究中，通常是从防御方的角度出发来分析问

题，因此大多采用强Stackelberg均衡作为博弈模型的解概念。

2.2 中间人攻击

2.2.1 中间人攻击技术原理

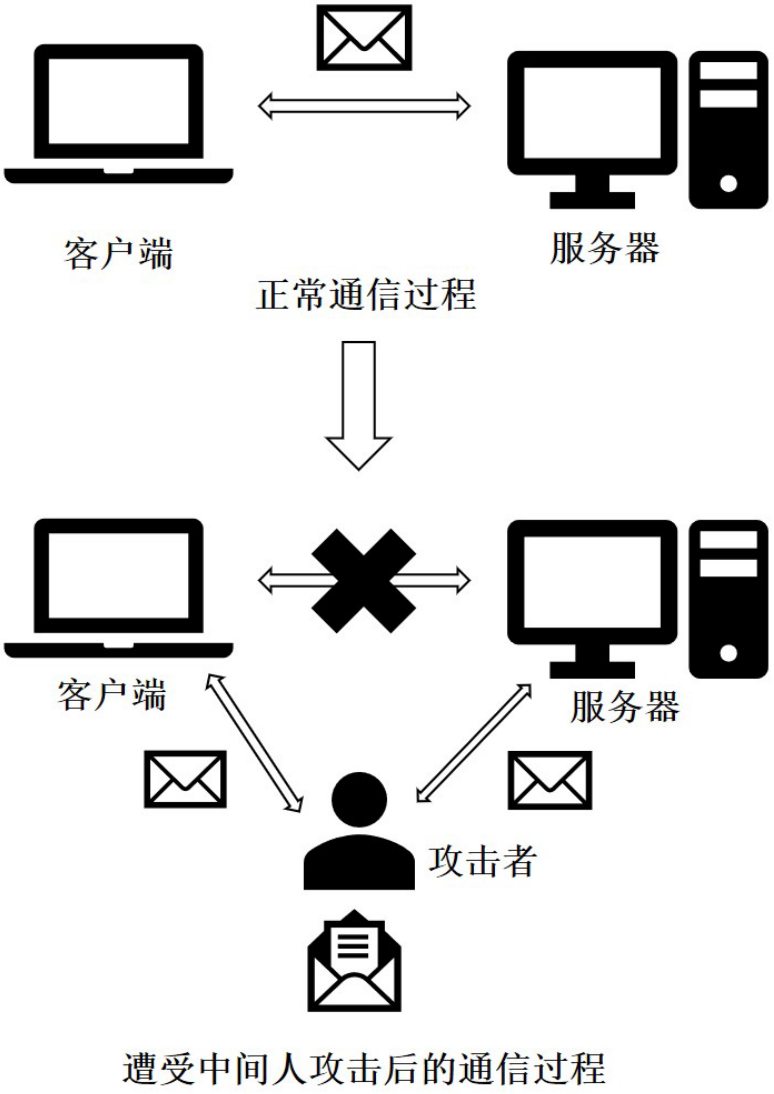


图 2-2 中间人攻击简单示例

在密码学和计算机网络安全领域中，中间人攻击是指攻击者通过使用某种技术手段介入到原本正常通讯的双方，也就是与双方分别建立独立的连接，这样攻击者就可以作为中间人监听整个通讯过程，然而通训的双方对此却毫不知情，仍旧以为自己在与对方直接通讯[43]。在中间人攻击中，攻击者可以通过截取通

讯双方的数据包获得双方的敏感信息，也可以为了达到自己的某种目的对数据进行修改，图2-2展示了中间人攻击的一个简单示例。目前实施中间人攻击的技术有很多，下面简单介绍几种常见的中间人攻击技术。

基于ARP欺骗的中间人攻击技术主要是指攻击者通过对计算机中ARP缓存的网关地址进行修改来达到攻击的目的。攻击者在实施攻击时，通过将计算机中ARP缓存的网关地址修改为自己的MAC地址，同时修改网关设备上ARP缓存中远程计算机的MAC地址为自己的MAC地址，这样一来计算机在上网时的通信数据都会流经攻击者，从而攻击者就可以窃取用户的敏感信息[44]。

SSL协议是浏览器与网站服务器之间的一种安全通信协议，该协议为通信双方提供端点认证，并建立一条可靠的安全连接来保障数据不被窃听、篡改或是伪造[45]。SSL中间人攻击技术是指攻击者首先通过其他相关技术实现会话劫持，例如DNS欺骗等，之后攻击者利用SSL协议在建立安全连接时证书认证不完善的缺陷或漏洞等，通过伪造数字证书或自签名证书充当SSL中间人进行欺骗和攻击[46]。

2.2.2 中间人攻击防御措施

在当前网络通讯过程中，服务器/客户端模型是主要的较常使用的通讯框架，服务器常常成为中间人攻击的主要攻击目标。因此，本文主要研究针对服务器的中间人攻击防御问题，下面将详细介绍研究中所采用的中间人攻击防御措施以及所要解决的关键问题。

服务器与客户端之间通讯的数据包主要是通过服务器的端口转发完成的，每个端口都有自己对应的协议，每个协议都有相应的服务。客户端（用户）需要不同的服务就会通过不同的端口与服务器进行数据包的交换，例如，80端口就是超文本传输协议（http协议）的默认端口，可以为用户提供网络服务，例如浏览网页等。显然，这样的默认端口设置给中间人攻击者提供了便利，攻击者可以通过解析特定服务所对应的协议来进行攻击，从而可以监听该服务所对应端口所转发的通讯数据，获得自己想要的敏感信息。因此，目前有一些研究提出使用端口跳变技术来迷惑攻击者，从而防御这样类型的攻击[47,48]。端口跳变的基本原理是当客户端向服务器发出通讯请求时，服务器并不使用默认的服务端口与客户端进行通讯，而是利用某些算法技术等将默认端口映射到一个没有使用的随机端口，使用映射后的端口与客户端进行通讯，这样攻击者就不能很快地准确知道客户端与服务器是通过哪个端口进行通讯的，从而达到防御攻击的目的。

基于端口跳变技术的一些设置，所有可用的端口可以依据其可提供的服务分成不同的组，即对于每一个服务s都对应一组可以提供该服务的端口集合S_s。为了方便分析，这里只分析提供某一特定服务的一个端口集合，提供其他服务的其

他端口集合可以类似地进行分析。对于同时需要服务s的所有用户，服务器可以将能够提供该服务的端口分配给这些用户使用。为了后续能够方便地建立模型以及分析，假设每一个端口对应一个用户，即端口与用户是一一对应关系。在后文中，用户与端口将交替使用表示相同含义，都表示防御方。

由于中间人攻击的攻击者目标大多是获取用户的敏感信息，因此为了减小攻击者获取到的用户有效信息的概率，采取的一种防御方法是在用户与服务器通讯的数据包中添加一些无用的噪声数据。当攻击者截取到混有有效信息和噪声数据的通讯数据包后，其解析数据包获得用户有效信息的难度就会相应地增加。但是，在通讯过程中添加噪声数据必然会对用户与服务器之间的通讯造成一定的延时。因此，需要在降低攻击者获取有效信息的概率与通讯延时之间进行权衡折衷。

假设每个用户在其与服务器通讯的数据包中以比例f插入噪声数据，本质上来说这个比例决定了攻击者获取该用户有效敏感信息的概率。原始通讯数据包中插入的噪声数据的比例f越大，那么攻击者获得有效信息的概率p就越小。f与p之间的关系可以用最简单的线性关系$p = 1 - f$来表示，从公式可以看出，当原始通讯数据包中没有插入噪声数据时（即$f = 0$），攻击者将以概率$p = 1$获得用户的有效信息，而当插入的噪声数据比例$f = 1$时，攻击者获取有效信息的概率变为0，这与实际情况比较吻合。另一方面，在通讯过程中插入噪声数据必然导致在传输相同数量的有效信息时需要交换更多的数据包，因此就会造成一定的通讯延时。由于插入噪声数据所造成的通讯延时程度用符号q来表示，它的大小同样决定于噪声数据的比例f。因为当用户与服务器进行通讯时，插入的噪声数据的比例f越大，就需要发送更多的数据包来完成有效信息的交换，必然造成更长的通讯时间，那么延时程度q就会变大。

为了找到一个合适的噪声数据的比例f，不仅需要考虑信息被窃取的损失也要考虑通讯延时所造成的损失，因此，需要权衡攻击者获取有效信息的概率p与插入噪声数据所造成的通讯延时程度q之间的关系。p与q之间的关系可以使用函数$q = F(p) : [0, 1] \mapsto [0, 1]$来表示。直观分析来看，如果在通讯过程中没有添加噪声数据（即$f = 0$），那么攻击者获得有效信息的概率$p = 1 - f = 1$，而且也就没有了噪声数据所造成的通讯延时，即$q = 0$；反之，如果在通讯的数据包中都是噪声数据（即$f = 1$），那么攻击者获得有效信息的概率$p = 1 - f = 0$，此时的通讯延时程度将会是最大的，最大的通讯延时程度用$q = 1$来表示。对于一些非极端的情形，可以采用实验仿真的方法来获得，从而得到函数$q = F(p)$的对应关系。获得函数$q = F(p)$的实验仿真的具体方法会在3.3.1小节详细介绍，为了后续理论分析方便，假设函数$q = F(p)$是关于p的连续的严格递减的凸函数。

在后续研究中，将基于上述方式对中间人攻击进行防御，因此关键亟待解决

的问题是如何为防御方设计合适的噪声数据的比例f，才能使得防御方在遭受中间人攻击时所承受的损失最小，即信息泄露的损失以及通讯延时造成的损失，这是主要的研究内容与重点。

2.3 本章小结

本章主要介绍了研究中所涉及到的相关理论知识和相关技术。首先，在基础的层次上对博弈论进行了总体的、轮廓性的概述，阐述了博弈论的几个一般概念，之后简单介绍了安全博弈论的相关理论知识；然后对中间人攻击的概念及其相关技术做了简单的说明；最后详细描述了所要研究的中间人攻击防御措施并提出了研究的关键问题。

第3章 基于Stackelberg博弈模型的中间人攻击防御策略

本章主要介绍在中间人攻击不可避免的情况下，从整体的角度出发，以降低整体在遭受攻击时的损失为目标为防御方设计防御策略。首先，基于上一章节对中间人攻击防御措施的介绍，将防御方与攻击者之间的交互过程建模为Stackelberg博弈模型；之后对Stackelberg博弈模型进行了理论分析，采用强Stackelberg均衡作为最优防御策略并提出一个减小计算最优防御策略有效搜索空间的算法，从而可以高效求解最优防御策略；最后，通过实验对比验证了在遭受中间人攻击时，防御方的最优防御策略能够有效地降低整体所遭受的损失。

3.1 Stackelberg博弈模型

从整体的角度分析，把服务器与用户看作一个整体作为防御方，研究的主要目标就是为防御方制定最优的防御策略来保证在遭受中间人攻击时与服务器通讯的所有用户的整体损失最小。在实际的攻击情景中，一个理性的攻击者在实施攻击之前，通常会对当前服务器所部署的防御策略进行充分的调研，然后制定自己的攻击策略。基于这种情况提出如下假设：在攻击开始之前，攻击者可以事先知道防御方所部署的防御策略。那么攻击者和防御方之间的交互就可以描述如下：首先防御方选取一个防御策略进行部署，攻击者在观察了防御方的防御策略后，选择自己的攻击策略进行攻击。这是一个有先后顺序的博弈即动态博弈，与Stackelberg博弈模型比较吻合。因此，在这种情况下采用Stackelberg博弈模型来对攻击者和防御方之间的交互行为进行建模。为了方便理解分析，表3-1整理总结了模型及理论分析中用到的一些符号表示。下面从参与者、行动策略集以及收益函数构成博弈模型的基本要素对模型进行介绍。

参与者：Stackelberg博弈模型的参与者通常有两个，一个是首先做决策的领导者，一个是观察了领导者策略后再做决策的跟随者。因此，所建立的博弈模型的参与者就是实施中间人攻击的攻击者和作为防御方的服务器，其中防御方首先选择防御策略，是Stackelberg博弈模型中领导者，攻击者是Stackelberg博弈模型中的跟随者。

行动策略集：基于2.2.2节对中间人攻击防御措施的介绍，防御方采取的防御措施是在通讯过程中，在用户的通讯数据包中添加一定比例的噪声数据来迷惑攻击者，从而降低攻击者获得用户有效信息的概率。因此防御方可以采取的

表 3-1　模型及分析中用到的符号

符号	描述
N	所有可用端口的集合 $\|N\| = n$
S	攻击者所攻击的端口集合（攻击者的策略）
K	攻击者可以攻击的端口个数 $\|S\| \leq K$
p_i	攻击者从端口i中获得有效信息的概率（端口i的策略）
$F(p_i)$	端口i采用p_i策略时的通讯延时程度
v_i	使用端口i进行通讯的用户的信息价值
c_i	使用端口i进行通讯的用户由于通讯延时所遭受的损失
l_i^A	遭受攻击的端口i的期望损失函数
l_i^N	未遭受攻击的端口i的期望损失函数
p_i^A	遭受攻击的端口i的最优策略p值
p_i^N	未遭受攻击的端口i的最优策略p值

行动是为每个用户设置其通过端口进行传输的数据包中噪声数据的比例f，为了方便后文对模型进行理论分析，基于已知攻击者获得有效信息的概率p与f之间的关系为$p = 1 - f$，这里定义防御方的行动是为每个端口（用户）i确定一个攻击者获得有效信息的概率p_i，其中$p_i \in [0, 1]$。防御方的策略用向量$\mathbf{p}$来表示，使用N表示所有可用端口组成的集合，端口总数为n，即$|N| = n$，那么防御方的策略就可以形式化地表示为$\mathbf{p} = (p_1, p_2, ..., p_n)$。对于攻击者来说，由于时间成本以及计算资源等能力限制，不可能攻击全部的端口，所以攻击者可以采取的行动是从n个端口中策略性地选择K个端口实施攻击。因此，攻击者的策略是从端口集合N中选择一个端口子集S，对集合S中的端口进行中间人攻击，其中$|S| \leq K$，K为一个定值，代表了攻击者的能力限制。

收益函数：为了方便定义参与者双方的收益函数，首先介绍一些符号定义。符号v_i表示使用端口i进行通讯的用户的信息价值，同时也代表了攻击者成功攻击了端口i所能获得的敏感信息价值即攻击者的收益值。由于插入噪声数据而造成的通讯延时给使用端口i的用户所带来的延时损失用符号c_i来表示。双方的收益是策略组合的函数，因此给定一个策略组合（$\mathbf{p}$，S），攻击者的收益就是通过截取端口集合S中的端口所传输的数据包所获得的有效信息的价值之和。因此，攻击者的收益函数定义如下：

$$U_{\text{attacker}} = \sum_{i \in S} p_i v_i \tag{3-1}$$

为方便分析，这里直接分析防御方的损失值，也就是其收益值的相反数。防御方的损失就是所有被攻击的端口（用户）和未被攻击的端口（用户）的损失之和。如果端口i被攻击了，那么使用端口i通讯的用户的损失包括其敏感信息被攻击者获取的损失（$p_i v_i$）以及由于噪声数据所造成的通讯延时带来的损失（$F(p_i)c_i$）。

因此，其损失函数定义如下：

$$l_i^A = p_i v_i + F(p_i)c_i \tag{3-2}$$

如果端口i没有被攻击，那么使用端口i进行通讯的用户所遭受的损失就只有噪声数据所造成的通讯延时损失，它的损失函数定义如下：

$$l_i^N = F(p_i)c_i. \tag{3-3}$$

最后，防御方的损失函数定义如下：

$$\begin{aligned} L_{\text{defender}} &= \sum_{i\in S} l_i^A + \sum_{i\notin S} l_i^N \\ &= \sum_{i\in S}(p_i v_i + F(p_i)c_i) + \sum_{i\notin S} F(p_i)c_i \\ &= \sum_{i\in S} p_i v_i + \sum_{i\in S} F(p_i)c_i + \sum_{i\notin S} F(p_i)c_i \\ &= \sum_{i\in S} p_i v_i + \sum_{i\in N} F(p_i)c_i \\ &= U_{\text{attacker}} + \sum_{i\in N} F(p_i)c_i \end{aligned} \tag{3-4}$$

3.2 模型理论分析

上一小节完成了对中间人攻击防御问题的形式化建模，将其建模为Stackelberg博弈模型。本小节将对模型进行理论分析，采用强Stackelberg均衡作为Stackelberg博弈模型的解概念，也就是防御方的最优防御策略，并提出一个降低计算最优防御策略时搜索空间的方法。

Stackelberg均衡下，每个参与者的策略都是其最佳响应策略。但是当攻击者存在多个最佳响应策略时，攻击者可能会随机选择一个策略，就可能导致无法保证Stackelberg均衡的唯一性。为了避免这个问题，提出了强Stackelberg均衡和弱Stackelberg均衡两个概念，在第2.1.2章节已经介绍。在上述的模型中，从公式3-1和公式3-4可以看出，当多个策略对攻击者来说没有区别时，即攻击者的多个攻击策略所产生的收益U_{attacker}相同，无论攻击者选择哪个策略，对防御方来说其损失值都相同。这就表明当出现均衡选择问题时，无论攻击者通过什么方式来选择其均衡策略，对防御方来说是没有影响的，当然也就说明防御方的最优防御策略等价于模型的强Stackelberg均衡策略，同时保证了防御方最优防御策略的唯一性。

计算Stackelberg博弈模型的Stackelberg均衡的一种常用方法是逆向归纳法。该方法的思想是向前展望，向后推理，也就是首先思考自己的策略可能引起的所有后续响应，以及后续响应的后续响应，直到这个博弈过程结束；然后从最后一

步开始依次倒推，找出自己在每一步的最佳响应策略。以中间人攻击防御问题为例，首先防御方考虑自己的每一个防御策略下攻击方可能的攻击策略，然后得到每个策略组合下的收益值，这样博弈就结束了。基于攻击者的收益值，为攻击者找出在每个防御策略下的最佳响应策略，最后防御方基于计算出的攻击者的最佳响应策略在自己的防御策略集中选择一个能够最小化自己损失的防御策略。但是，在所建立的模型中，因为$p_i \in [0,1]$并且$p_i \in R$，防御方的策略集合是无限的，不可能穷举出全部的防御策略。因此，逆向归纳法并不能直接用来对模型进行求解。因此，本文提出了一种有效的方法来降低计算强Stackelberg均衡时的搜索空间，从而可以高效求解防御方的最优防御策略。在介绍计算最优防御策略的算法之前，先来分析一个最优防御策略的子问题——假设给定攻击者的最佳响应策略，如何为防御方计算最优防御策略。

3.2.1 防御策略子问题求解

首先对攻击者的最佳响应策略进行分析，最佳响应策略就是当给定其他参与者的策略不变的情况下，能够最大化其自身收益函数的策略。从攻击者的收益函数（公式3-1）可以看出，当给定一个防御方的策略$\mathbf{p}$后，攻击者的最佳响应策略就是从端口集合N中选择$p_i v_i$值最大的K个端口组成集合S，作为其策略。

为了方便后续分析，先来介绍一些符号定义。符号p_i^A和p_i^N分别表示端口i在遭受攻击和没遭受攻击情况下的最优的策略p值，其本质就是端口i的损失函数l_i^A和l_i^N在区间[0,1]上的最小值点，并且从公式3-2和公式3-3很容易可以看出p_i^A和p_i^N对端口i来说都是唯一的。根据上述分析可以知道攻击者的最佳响应策略是包含$p_i v_i$值最大的K个端口的集合S，所以可以将防御方的最优防御策略的搜索问题限制在一个空间里，在这个空间里的集合S中的端口有较高的$p_i v_i$值，也就是集合S是攻击者的最佳响应策略。命题3.1给出了一个特殊情况下（在集合S中的端口参数值与剩下的端口参数值大大不同）防御方的最优防御策略。

命题 3.1 假设给定集合S是攻击者的最优策略，那么防御方的目标就是选择一个针对集合S的最优防御策略。如果$\min_{i\in S} p_i^A v_i \geq \max_{i\notin S} p_i^N v_i$，那么防御方的针对集合$S$的最优防御策略就是对于$i \in S$的端口，选择$p_i^A$作为其策略，对于$i \notin S$的端口，选择$p_i^N$作为其策略。

证明： 首先，从攻击者的收益函数$U_{\text{attacker}} = \sum_{i\in S} p_i v_i$以及$|S| \leq K$可以知道攻击者的最佳响应策略是选择$pv$值最高的$k$个端口，已知当前集合$S$就是攻击者的最优策略，所以在集合$S$中的端口相比不在集合$S$中的端口拥有较高的$pv$值。因此，在给定防御方策略$\mathbf{p}$后，集合$S$是攻击者的最优策略当且仅当集合$S$满足$\min_{i\in S} p_i v_i \geq \max_{i\notin S} p_i v_i$。其次，根据命题中的条件$\min_{i\in S} p_i^A v_i \geq \max_{i\notin S} p_i^N v_i$，可知在当

前防御方的策略下，也就是$i \in S$的端口，选择p_i^A作为其策略，对于$i \notin S$的端口，选择p_i^N作为其策略时，集合S是攻击者的最优策略，同时根据p_i^A 和p_i^N的定义可知，p_i^A 是端口$i \in S$ 的最优策略，p_i^N是端口$i \notin S$的最优策略，也就是当每个端口都选择自己的最优策略时，仍能保证攻击者的最优策略S不变。因此，命题3.1中所提出的防御策略就是针对集合S的最优的防御策略。 □

命题3.1给出了基于攻击者的最佳响应策略已知的情况下一个特殊情况下的防御方的最优防御策略。因为命题中的双方策略都是其最佳响应策略，所以由攻击者的最佳响应策略策略S以及由命题3.1给出的针对策略S的防御方策略$\mathbf{p}$所组成的策略组合（$\mathbf{p}$，S）是一个Nash均衡，但是在Stackelberg博弈模型中，这个Nash均衡并不一定是一个强Stackelberg均衡。命题3.1只是描述了求解最优防御策略的一个特殊情况，下面将考虑一般情况并提出一个关于最优防御策略的必要条件。

定理 3.1 假设给定集合S是攻击者的最佳响应策略，防御方的目标就是选择一个针对集合S的最优防御策略。那么，在防御方的最优防御策略$\mathbf{p}$中存在一个λ使得：

- 对于任意端口$i \in S$，如果$p_i^A v_i < \lambda$，那么$p_i = \frac{\lambda}{v_i}$；否则，$p_i = p_i^A$。
- 对于任意端口$i \notin S$，如果$p_i^N v_i > \lambda$，那么$p_i = \frac{\lambda}{v_i}$；否则，$p_i = p_i^N$。

证明： 根据命题3.1的证明，可以知道$\min\limits_{i \in S} p_i v_i \geq \max\limits_{i \notin S} p_i v_i$是集合$S$是攻击者最佳响应策略的充分必要条件。假设$\lambda = \max\limits_{i \notin S} p_i v_i$。

首先分析当端口i被攻击的情况，也就是$i \in S$。根据攻击者最佳响应策略的充分必要条件可知，该被攻击端口i的$p_i v_i$的值不小于λ。端口i的损失函数为$l_i^A = p_i v_i + F(p_i) c_i$，$p_i^A$是函数$l_i^A$的最小值点，也就是端口$i$被攻击时的最优策略，所以，如果$p_i^A$能够满足$p_i^A v_i >= \lambda$，那么端口$i$ 的最优策略就是p_i^A，即$p_i = p_i^A$；如果$p_i^A v_i < \lambda$，即$p_i^A < \frac{\lambda}{v_i}$，由于端口$i$的最优策略$p_i$应该满足$p_i v_i \geq \lambda$，即$p_i \geq \frac{\lambda}{v_i}$，又知函数$F(p_i)$是凸函数，可以知道函数$l_i^A$在$[p_i^A,1]$ 上是递增的，那么端口i 的最优策略就是$\frac{\lambda}{v_i}$，即$p_i = \frac{\lambda}{v_i}$ 时函数l_i^A取得最小值。

类似地，对端口i没有被攻击的情况进行分析，也就是$i \notin S$。根据攻击者最佳响应策略的充分必要条件可知，端口i的$p_i v_i$的值不大于λ。端口i 的损失函数为$l_i^N = F(p_i) c_i$，p_i^N是函数l_i^N的最小值点，也就是端口i未被攻击时的最优策略，所以，如果p_i^N能够满足$p_i^N v_i \leq \lambda$，那么端口i 的最优策略就是p_i^N，即$p_i = p_i^A$；如果$p_i^N v_i > \lambda$，即$p_i^N < \frac{\lambda}{v_i}$，因为函数$F(p_i)$ 是减函数，而且端口i的最优策略p_i应该满足$p_i v_i \leq \lambda$，即$p_i \leq \frac{\lambda}{v_i}$，那么端口$i$的最优策略就是$\frac{\lambda}{v_i}$，即$p_i = \frac{\lambda}{v_i}$ 时，函数l_i^N取得最小值。 □

定理3.1给出了一个最优防御策略的必要条件，当给定一个任意的攻击者的策略端口集合S后，可以使用搜索技术找到一个最优的λ值，从而获得防御方的最优防御策略$\mathbf{p}$。进一步来说，按照定理3.1给出的算法，可以计算出针对任意攻击者策略S下攻击者的最佳响应策略，即最优防御策略。按照这样的方法，可以先对所有K大小的端口子集按照上述方法求解出每一个子集所对应的最优防御策略，之后在所求得的最优防御策略中，进一步搜寻找到使得防御方损失最小的防御策略，从而确定防御方的最优防御策略。但是在实际中，端口数量n的大小和K的取值大小可能导致端口子集的个数（$\binom{n}{K}$）很大，从而导致搜索空间依旧会很大，因此上述方法可能并不能有效地对问题进行求解。接下来，基于上述分析介绍一种有效且实用的计算防御方最优防御策略的方法。

3.2.2 防御策略问题求解

下面分析当定理3.1中的λ值已知的情况下，如何为防御方计算其最优防御策略，具体方法通过定理3.2给出。

定理 3.2 假设给定一个λ值，那么防御方的最优防御策略就是满足$\min_{i\in S} p_i v_i \geq \lambda$和$\max_{i\notin S} p_i v_i \leq \lambda$的策略，其中$S$ 是攻击者的最佳响应策略。下面给出一个计算最优防御策略的算法：

1. 按照如下方式计算每一个端口的l_i^N值：对于端口i，如果$p_i^N v_i < \lambda$，则$l_i^N = F(p_i^N)c_i$；否则$l_i^N = F(\frac{\lambda}{v_i})c_i$；
2. 按照如下方式计算每一个端口的l_i^A值：对于端口i，如果$p_i^A v_i > \lambda$，则$l_i^A = p_i^A v_i + F(p_i^A)c_i$；否则$l_i^A = \frac{\lambda}{v_i} v_i + F(\frac{\lambda}{v_i})c_i$；
3. 计算每一个端口的D_i值：$D_i = l_i^A - l_i^N$；
4. 在所有端口中选择D_i值最小的K个端口组成集合S；
5. 对于每一个在集合S中的端口i，如果$p_i^A v_i > \lambda$，那么$p_i = p_i^A$；否则，$p_i = \frac{\lambda}{v_i}$；
6. 对于每一个不在集合S中的端口i，如果$p_i^N v_i < \lambda$，那么$p_i = p_i^N$；否则，$p_i = \frac{\lambda}{v_i}$；
7. 输出防御方最优策略$\mathbf{p} = (p_1, p_2, ..., p_n)$。

证明： 首先，假设攻击者的最佳响应策略S是给定的，这样情况就与定理3.1是类似的。那么，根据定理3.1的证明可知，上述算法中步骤5-6得出的策略就是针对S的最优防御策略。接下来，只需要确定步骤1-4生成的攻击策略S是否是对于防御方最优的集合。

这里使用反证法进行证明，假设存在一个集合S^*能够使得防御方遭受更低的期望损失。因为已知步骤5-6能够针对任意集合给出最优的防御策略，所以假设对应于集合S和S^*的防御策略都是通过步骤5-6计算出的。使用i^+表示在集合S^*中但不在集合S中的端口组成的集合，使用i^-表示在集合S中但不在集合S^*中的端

口组成的集合。那么，可以分别计算出对应于集合S和集合S^*的防御者的期望损失如下：

$$L_{\text{defender}} = \sum_{j\in S} l_j^A + \sum_{j\notin S} l_j^N \tag{3-5}$$

$$L^*_{\text{defender}} = \sum_{j\in S^*} l_j^A + \sum_{j\notin S^*} l_j^N \tag{3-6}$$

两者之间的差值计算如下：

$$\begin{aligned}\Delta L &= L_{\text{defender}} - L^*_{\text{defender}} \\ &= \sum_{j\in S} l_j^A + \sum_{j\notin S} l_j^N - (\sum_{j\in S^*} l_j^A + \sum_{j\notin S^*} l_j^N) \\ &= \sum_{j\in S} l_j^A - \sum_{j\in S^*} l_j^A + \sum_{j\notin S} l_i^N - \sum_{j\notin S^*} l_j^N \\ &= \sum_{j\in i^-} l_j^A - \sum_{j\in i^+} l_j^A + \sum_{j\in i^+} l_j^N - \sum_{j\in i^-} l_j^N \\ &= \sum_{j\in i^-} l_j^A - \sum_{j\in i^-} l_j^N - (\sum_{j\in i^+} l_j^A - \sum_{j\in i^+} l_j^N) \\ &= \sum_{j\in i^-} D_j - \sum_{j\in i^+} D_j \end{aligned} \tag{3-7}$$

从步骤4中可知集合S中的端口是D_i值最小的K个端口，所以，可以得到$\sum_{j\in i^-} D_j - \sum_{j\in i^+} D_j < 0$。因此，$\Delta L < 0$，即$L_{\text{defender}} < L^*_{\text{defender}}$，这与之前的假设矛盾，那么原命题成立，所以集合$S$是对于防御方最优的集合。 □

通过使用定理3.2给出的算法，当给定一个任意的λ值时，就可以计算出防御方的最优防御策略。使用符号$L_{\text{defender}}(\lambda)$来表示给定$\lambda$值的情况下，防御方所承受的最小的损失，本质上来说就是给定λ值后，使用上述算法所计算出的防御方最优防御策略下所对应的损失值。因为研究的目标是寻找使得防御方损失最小的防御策略，那么就可以把寻找防御方最优防御策略的问题转化为寻找$L_{\text{defender}}(\lambda)$最小值点$\lambda^*$的问题，在找到了$\lambda^*$后，利用定理3.2所给出的算法很容易地可以计算出最优防御策略$\mathbf{p}$。

3.3 防御策略实验评估

通过3.2.2节的理论分析，计算防御方最优防御策略的问题转化成了寻找函数$L_{\text{defender}}(\lambda)$最小值点$\lambda^*$的问题。因此，本节首先介绍在仿真实验中如何寻找最小值点λ^*，进而求得防御方的最优防御策略。之后为了验证最优防御策略的有效性，将与一些其他非策略性的防御策略进行实验对比，比较每个防御策略在降低防御方在遭受中间人攻击时的整体损失的效果。

3.3.1 防御策略求解实现

通过上述理论分析，可以知道求解防御方最优防御策略的关键在于如何寻找函数$L_{\text{defender}}(\lambda)$的最小值点$\lambda^*$。因此，需要知道$L_{\text{defender}}(\lambda)$与$\lambda$的对应关系，也就是给定任意的$\lambda$值要知道其对应的$L_{\text{defender}}(\lambda)$值，然后使用搜索算法找到其最小值点即$\lambda^*$。$L_{\text{defender}}(\lambda)$的计算方法就是按照定理3.2所给出的算法首先计算出该λ值下防御方的最优防御策略，然后计算该防御策略的期望损失值。通过观察定理3.2中的算法，亟需解决的问题是如何获得攻击者获得有效信息的概率p与通讯延时程度q之间的对应关系$q = F(p)$以及如何计算每个端口i的p_i^A值和p_i^N值。

首先，攻击者获得有效信息的概率p与通讯延时程度q之间的对应关系（$q = F(p)$）可以通过仿真实验得到。在仿真实验中，采用HTTP协议作为其通讯协议，也可以使用其他的通讯协议进行仿真实验。由于通讯延时是跟通讯时间有关的，所以在实验中通过记录通讯时间来衡量其通讯延时程度。假设用户与服务器之间需要交换的通讯数据为一个定值，通过在通讯数据包中插入不同比例的噪声数据使得攻击者获得有效信息的概率p在[0,1]中取值，之后记录下p值下用户和服务器之间完成通讯时所需要的时间，这样通过大量的实验就可以采样到不同p值下完成通讯所需要的时间。最后对所得到的通讯时间数据进行标准化处理，使其落入[0,1]之间，把标准化后的通讯时间数据看作通讯延时程度q。按照上述方法便可以得到攻击者获得有效信息的概率p与通讯延时程度q之间的对应关系（$q = F(p)$），如图3-1所示。

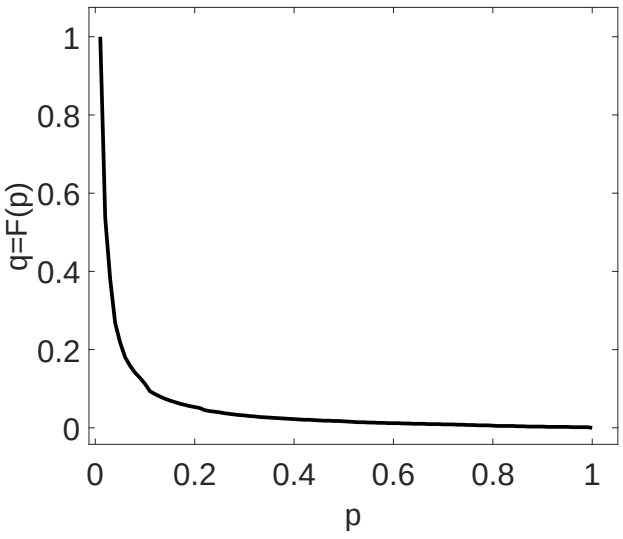


图 3-1　　p与q的关系图

在获得了攻击者获得有效信息的概率p与通讯延时程度q之间的对应关系之

后，接下来需要解决的问题是如何计算端口i的p_i^A和p_i^N的值。其实这两个值就是端口i的损失函数l_i^A和l_i^N在区间[0,1]上的最小值点。根据$l_i^N = F(p_i)c_i$（公式3-3）以及函数$F(p)$是一个减函数，很容易知道1就是函数l_i^N在区间[0,1] 上的最小值点，因此对于所有的端口i，其p_i^N值都为1。这表明当端口没有被攻击时，对使用该端口的用户来说在通讯数据包中不添加任何的噪声数据是其最好的策略，与实际情况相符合。对于如何计算p_i^A的值，已知$p_i^A = \arg\min_p l_i^A = \arg\min_p (pv_i + F(p)c_i)$，由于函数$F(p)$是通过充分采样$p$值获得的一系列的数据点组成的并且是一个凸函数，因此可以利用穷尽搜索的方法来找到函数l_i^A的最小值点p_i^A。

最后，解决了上述两个问题后，就可以通过对λ的值进行充分采样，然后根据定理3.2的算法计算出防御方的最优防御策略，进而求得防御方在该防御策略下的期望损失$L_{\text{defender}}(\lambda)$。图3-2展示了当$K = 3$，$c_i$和$v_i$的值分别服从幂律分布和正态分布时，防御方的期望损失$L_{\text{defender}}(\lambda)$关于$\lambda$的变化关系。从图中可以看出函数$L_{\text{defender}}(\lambda)$的图像是比较平滑的，因此可以使用穷尽搜索的方法找到最小值点λ^*。在找到了λ^*后，防御方的最优防御策略就可以相应地计算出来。

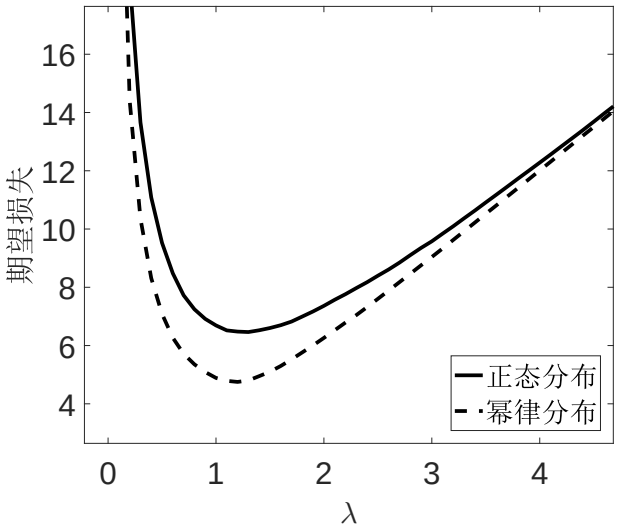


图 3-2　　期望损失关于λ的变化图($K = 3$)

3.3.2 对比实验

为了验证防御方的最优防御策略是否能够有效地降低防御方在遭受中间人攻击时的整体损失，将最优防御策略与其他非策略性的防御策略进行实验对比，观察在遭受中间人攻击时，每个防御策略下防御方所承受的整体损失。

首先介绍实验的一些基本设置。假设实验中有30个用户分别通过30个端口与服务器进行通讯，每个用户i的c_i值和v_i值需要初始化。v_i值代表用户信息的价

值，是由用户在整个组织中的相对地位或级别所决定。这里考虑两种情况，第一种情况假设v_i值服从幂律分布，因为在公司等大型组织中分层结构比较常见，也就是有很少一部分人拥有较高的级别，而绝大多数人的级别较低；另一种情况假设相对较少的人拥有较高或较低的级别，大多数人拥有中等相近的级别。这种情况比较符合正态分布的规律，因此假设v_i的值服从正态分布。c_i值表示由于通讯延时所造成的损失，它是由用户工作需求的紧急程度所决定的。因为每个用户都可能有不同的工作任务，这里同样假设c_i服从幂律分布或是正态分布。c_i和v_i的值并没有关联性，因此实验在两种情况下进行：1）c_i和v_i所服从分布的参数相同；2）c_i和v_i所服从分布的参数不同。

为了研究在中间人攻击存在的情况下，防御方的最优防御策略能否有效降低防御方的整体损失，将其与几个非策略性的防御策略进行对比实验。由于是非策略性的防御策略，这些防御策略将为每一个端口设置相同的p值。一种极端的情况就是将每个端口的p值全部设置为1，这就等价于无防御状态，因为$f = 1 - p = 0$，也就是通讯数据包中并没有添加任何噪声数据，这种情况看作是实验基准情况。除此之外，实验还对比了其他两种非策略性的防御策略，虽然这两种非策略性的防御策略并没有考虑攻击者以策略性的方式选择策略，但是是对攻击者的行为进行了基本假设后所计算出的最优策略，具体计算方式如下：

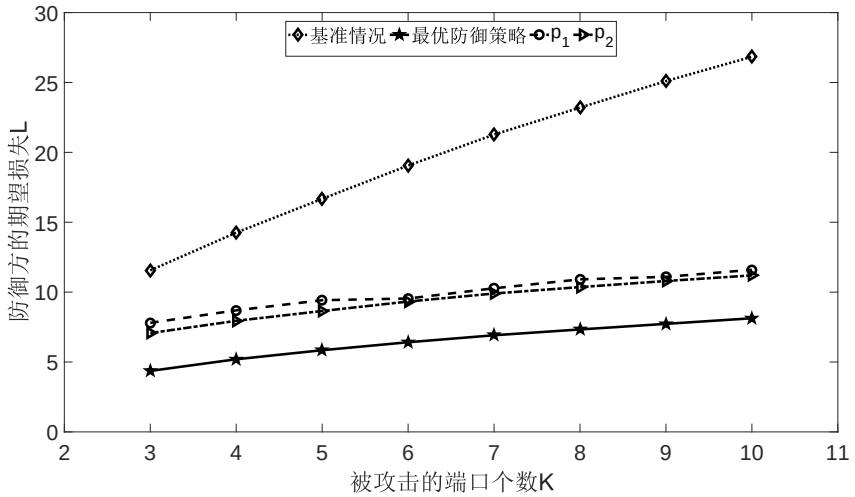
第一个防御策略假设攻击者随机选取端口进行攻击，基于这个假设，防御方计算一个能够最小化自己期望损失的p值，记为p_1，p_1值的计算方式见下式：

$$p_1 = \arg\min_{p} \left(\frac{K}{n} \sum_{i \in N} v_i \right) p + F(p) \sum_{i \in N} c_i. \tag{3-8}$$

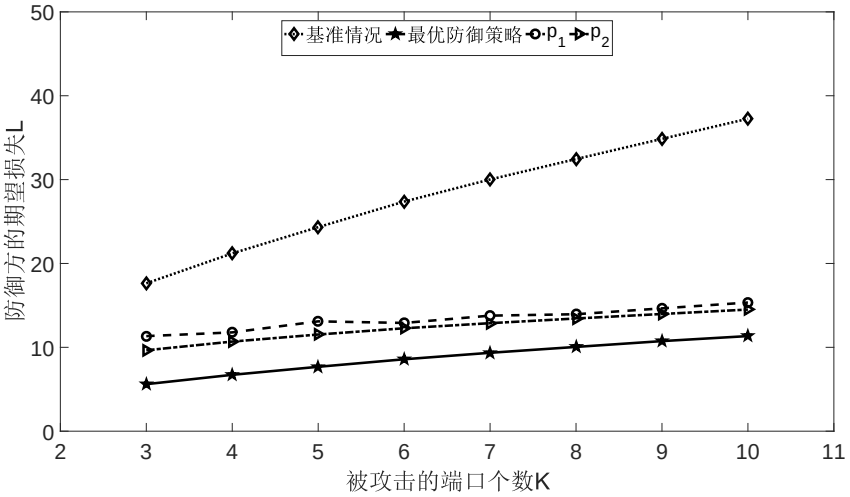
第二个防御策略假设攻击者选择那些信息价值较高（v_i值较大）的端口进行攻击，相应地，防御方基于假设计算其最优的p值，记为p_2，p_2值的计算如下：

$$p_2 = \arg\min_{p} \left(\max_{S:|S|=K} \sum_{i \in S} v_i \right) p + F(p) \sum_{i \in N} c_i. \tag{3-9}$$

最后，在中间人攻击一直存在的情况下，针对每一个防御策略假设攻击方一直选择自己的最佳响应策略，最后计算防御方的期望损失。攻击者所能够攻击的端口个数K从3一直变化到10 进行实验，用户参数c_i和v_i值服从幂律分布时的实验结果如图3-3所示，其中图3-3 (a)，3-3 (b)分别表示c_i和v_i服从相同和不同参数下的情况。图3-4展示了用户参数服从正态分布时的实验结果，其中图3-4 (a)，3-4 (b)分别表示c_i和v_i服从相同和不同参数下的情况。

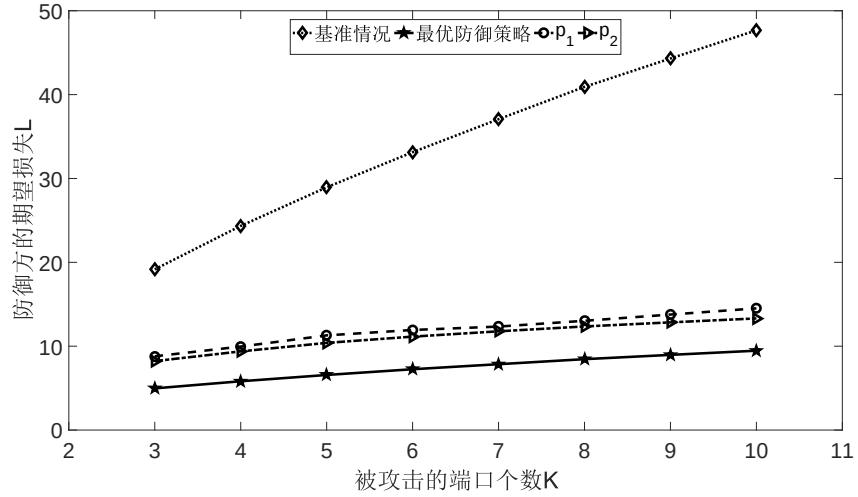


(a) c与v所服从分布的参数相同

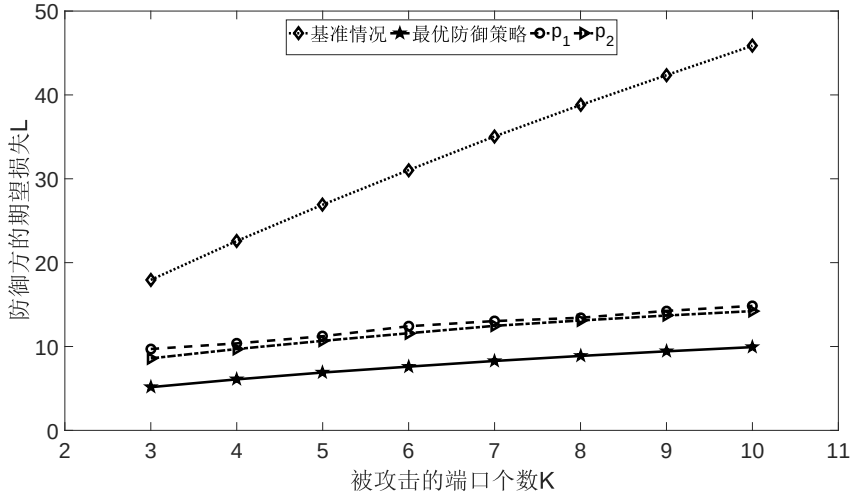


(b) c与v所服从分布的参数不同

图 3-3　　期望损失L关于K的变化图（幂律分布）



(a) c与v所服从分布的参数相同



(b) c与v所服从分布的参数不同

图 3-4　　期望损失L关于K的变化图（正态分布）

从实验结果图中可以看出，随着攻击者能够攻击的端口个数K的不断增加，每一个防御策略下防御方所承受的整体损失都在增加。但是相比实验基准情况，可以发现其他三种防御策略所对应的防御方的整体损失增加速率较慢。而且在每一种情况下，实验基准情况所对应的曲线都是最高的，这就表明基准策略下防御方所遭受的整体损失是最大的。因此，可以说明随着攻击者攻击能力的增加，采取在通讯数据包中添加一定比例的噪声数据这种防御措施能够有效降低防御方在遭受中间人攻击时的整体损失。通过仔细对比两种非策略性的防御策略，第二种策略p_2的实验效果稍稍优于第一种策略p_1，但是最优防御策略的曲线在所有情况下都是位于最下面的，这表明不论c_i和v_i值服从什么分布，最优防御策略都

能够将防御方所遭受的整体损失降到最低。

3.4 本章小结

本章从整体的角度出发，研究在中间人攻击一直存在的情况下，如何为防御方设计防御策略使得防御方所遭受的整体损失降到最低。首先采用Stackelberg博弈模型对中间人攻击防御问题进行建模，然后对博弈模型进行理论分析并提出了一个有效实用的求解最优防御策略的算法，最后通过实验验证了最优防御策略在降低防御方整体损失方面明显优于其他非策略性的防御策略。

第4章　基于同时行动博弈模型的中间人攻击防御策略

上一章节从整体角度出发，为防御方设计了最优的防御策略来降低整体在遭受中间人攻击时所承受的损失，这种防御策略比较适合全体用户合作以保全整体利益为目标的情况。但是在现实生活中，如果与服务器通讯的用户互不相识，每个人都想要最小化自己的损失，而且并不想要牺牲自己的利益来保全整体利益，那么上述的防御策略可能就不再适用。本章考虑在中间人攻击不可避免的情况下，从用户个体角度出发，研究如何为用户设计防御策略能够使得其自身所遭受的损失降到最低。

4.1　同时行动博弈模型

在实际的一些情况中，在实施中间人攻击之前，由于某些原因攻击者可能并不能准确地获取到防御方（服务器）的防御策略。比如使用某个端口与服务器进行通讯的用户结束了自己的通讯，另外的用户会接着使用该端口与服务器进行通讯，但是每个用户所对应的防御策略不同，也就是防御方的策略可能会时时变化，那么攻击者就无法时刻准确了解到该端口的防御策略。在这种情况下，Stackelberg博弈模型就不再适用，同时行动博弈模型更符合这种情况。因此，这里采用同时行动博弈模型对这种情况下的问题进行建模。为了便于理解模型及后续理论分析，表4-1列出了模型及分析中用到的符号。

表 4-1　　模型建立及分析中用到的符号

符号	描述
N	所有端口的集合 $\|N\| = n$
S	攻击者所攻击的端口集合（攻击者的纯策略）
K	攻击者可以攻击的端口个数 $\|S\| \le K$
a_i	攻击者攻击端口i的概率
$\boldsymbol{a}$	攻击者的混合策略，$\boldsymbol{a}= (a_1, ..., a_n)$
p_i	攻击者从端口i中获得有效信息的概率（端口i的策略）
$F(p_i)$	端口i采用p_i策略时的通讯延时程度
v_i	使用端口i进行通讯的用户的信息价值
c_i	使用端口i进行通讯的用户由于通讯延时所遭受的损失
$L_i^{a_i}$	当端口i被攻击的概率为a_i时，端口i的期望损失
$p_i^{a_i}$	当端口i被攻击的概率为a_i时，端口i的最优策略p 值

下面就从参与者、纯策略以及混合策略、收益函数几方面对所建立的同时行

动博弈模型进行介绍。博弈的参与者就是与服务器同时通讯的多个用户与一个中间人攻击者，由于之前假设用户与服务器的通讯端口是一一对应关系，这里把端口和用户看做一个防御方，在本章节中，将使用端口代表参与博弈的防御方。

首先来介绍端口和攻击者的纯策略。沿用3.1节的分析，由于时间成本以及计算资源等能力限制，对于攻击者来说只能从n个端口中策略性地选择K个端口进行攻击，因此攻击者的纯策略是从端口集合N中选择的K个端口组成的端口子集S，其中$|S| \leq K$，K为一个定值。对于每个端口i来说，其纯策略是设置一个合适概率p_i值，p_i表示攻击者获得有效信息的概率，是由通讯数据包中所添加的噪声数据的比例决定的。为了后续表示方便，使用向量$\mathbf{P}$表示所有端口的纯策略，即$\mathbf{p} = (p_1, p_2, ..., p_n)$。所有端口的纯策略的集合使用符号$\mathbf{P}$表示，即$\mathbf{P} = \{\mathbf{p_1}, \mathbf{p_2}, ...\}$。接下来介绍纯策略组合下的收益函数，这里沿用3.1节中对c_i和v_i的定义，当给定一个纯策略组合（$\mathbf{p}$，S），其中$\mathbf{p} \in \mathbf{P}$，攻击者的收益是攻击端口集合$S$中的端口所获得的有效信息的价值之和，因此攻击者的收益函数定义如下：

$$U_{\text{attacker}} = \sum_{i \in S} p_i v_i \tag{4-1}$$

当端口i遭受攻击时，使用它进行通讯的用户就要遭受信息泄露的损失以及通讯延时的损失，那么端口i的损失函数定义如下：

$$L_i^1 = p_i v_i + F(p_i) c_i \tag{4-2}$$

当端口i未遭受攻击时，使用它的用户只承担通讯延时的损失，所以端口i的损失函数可以按照如下定义：

$$L_i^0 = F(p_i) c_i \tag{4-3}$$

下面分析混合策略的情况。端口i的混合策略就是其纯策略（$p_i \in [0, 1]$）上的一个连续的概率分布。在第4.2小节将会证明对于一个端口来说，其最佳响应策略永远是一个纯策略，因此这里不再考虑端口的混合策略，只关注端口的纯策略。接下来分析攻击者的混合策略，使用符号A来表示所有大小为K的端口集合，m表示集合A 的元素个数，即$m = |A| = \binom{n}{K}$。如果攻击者的混合策略使用符号$\triangle A$来表示，那么$\triangle A$就是在集合A上的一个概率分布，这个概率分布可以用向量$\boldsymbol{b} = (b_1, ..., b_m)$形式化描述，其中$b_i$表示攻击者选择集合$A$中的第$i$个元素（端口子集）的概率，$b_i \geq 0$且$\sum_{i=1}^{m} b_i = 1$。为了方便起见，使用符号$a_i$来表示攻击者选择攻击端口$i$的概率。当给定一个攻击者的混合策略$\boldsymbol{b}$时，就可以相应地计算出$a_i$的值，得到一个概率向量$\boldsymbol{a} = (a_1, ..., a_n)$，并且满足约束条件$\sum_{i=1}^{n} a_i = K$。在后文中，将使用攻击者攻击每个端口概率所组成的概率向量$\boldsymbol{a}$来表示攻击者的混合策略。下面给出混合策略下收益函数的定义，对于给定的一个策略组合（$\boldsymbol{p}$，$\boldsymbol{a}$），攻击

者的期望收益函数定义如下：

$$U_{\text{attacker}} = \sum_{i=1}^{n} a_i p_i v_i \tag{4-4}$$

端口i的期望损失函数定义如下：

$$\begin{aligned} L_i^{a_i} &= a_i L_i^1 + (1 - a_i) L_i^0 \\ &= a_i p_i v_i + F(p_i) c_i \end{aligned} \tag{4-5}$$

4.2 模型理论分析

在同时行动博弈模型中，每个参与者同时做决策，Nash均衡是这类博弈模型常见的解概念。因此，这里采用Nash均衡作为所建立的同时行动博弈模型的解，也是端口的最优防御策略。对于多个端口和一个攻击者的多参与者的博弈模型，Nash均衡的定义如下：

定义 4.1 (Nash均衡) 在一个策略组合下，如果当其他参与者策略都不变的情况下，每个参与者的策略都是其最佳响应策略，即参与者不能通过单方面改变自己的策略而使得自己的收益增加，那么该策略组合就是一个Nash均衡。

本节将对博弈模型进行理论分析，证明博弈模型的Nash均衡存在且唯一。Nash均衡存在且唯一的性质消除了均衡选择的问题，同时保证了对端口来说必然存在唯一的最优防御策略。在此之前，先证明一个关于端口最佳响应策略的引理。

引理 4.1 端口（防御方）的最佳响应策略永远是纯策略。

证明： 首先，假设存在一个混合策略是端口i的最佳响应策略。混合策略就是其纯策略集上的连续的概率分布，使用$g(x)$来表示这个分布的概率密度函数，变量X服从该分布，即$X \sim g(x)$，$\int_0^1 g(x)\,dx = 1$。接下来，计算这个分布的期望值$E(X)$，$E(X) = \int_0^1 xg(x)\,dx$，并取这个期望值作为该端口的一个纯策略，即$p_i = E(X)$。当端口$i$选择该混合策略进行博弈时，攻击者攻击端口$i$的期望收益为$\int_0^1 a_i x v_i g(x)\,dx = a_i E(X) v_i$。如果端口$i$选择其纯策略$p_i = E(X)$进行博弈，攻击者攻击用户$i$的期望收益为$a_i p_i v_i = a_i E(X) v_i$。这就表明无论端口$i$选择混合策略$g(x)$或是纯策略$p_i$，对于攻击者来说，其期望收益不变，攻击者的策略不会改变，这样也就不会影响其他参与者的策略。最后，当给定攻击者的策略$\boldsymbol{a}$时，端

口i选择混合策略$g(x)$进行博弈时的期望损失为：

$$\begin{aligned} L_i^{a_i}(X) &= \int_0^1 (a_i x g(x) v_i + F(x g(x)) c_i)\, dx \\ &= a_i \int_0^1 x g(x)\, dx v_i + \int_0^1 F(x g(x))\, dx c_i \\ &= a_i E(X) v_i + E(F(X)) c_i \end{aligned} \tag{4-6}$$

端口i选择纯策略p_i进行博弈时的期望损失为：

$$\begin{aligned} L_i^{a_i}(p_i) &= a_i p_i v_i + F(p_i) c_i \\ &= a_i E(X) v_i + F(E(X)) c_i \end{aligned} \tag{4-7}$$

因为函数$F(p)$是严格的凸函数，所以根据詹森不等式[49]，可以得到$F(E(X)) < E(F(X))$，所以进一步可以得到$L_i^{a_i}(p_i) < L_i^{a_i}(X)$。这表明端口$i$选择纯策略时的期望损失小于其选择混合策略时的期望收益，那么该混合策略就不是用户的最佳响应策略。因此，端口的最佳响应策略必然是一个纯策略。 □

接下来，给出一个纯策略Nash均衡存在的充分必要条件，最后，同时考虑纯策略和混合策略并证明必然存在唯一的Nash均衡。

引理 4.2 该博弈模型存在纯策略Nash均衡当且仅当存在一个K大小的端口集合S使得$\min_{i \in S} p_i^1 v_i \geq \max_{i \notin S} p_i^0 v_i$。如果纯策略Nash均衡存在，那么它是唯一的，并且攻击者的均衡策略就是S。

证明： 假设策略组合（$\mathbf{p}$，S）是该博弈模型的一个纯策略Nash均衡，那么根据Nash均衡的定义，可知在该策略组合下，每个参与者的策略都是对于其他参与者策略的最佳响应策略。针对攻击者的纯策略S，对于在集合S中的端口i来说，其最佳响应策略为p_i^1，而对于不在集合S中的端口i来说，其最佳响应策略为p_i^0。针对防御方的策略$\mathbf{p}$，由公式4-1，知道攻击者的最佳响应策略就是选择$p_i v_i$值最大的K个端口组成集合S。因此，在纯策略Nash均衡中，端口集合S必须满足引理4.2中的条件$\min_{i \in S} p_i^1 v_i \geq \max_{i \notin S} p_i^0 v_i$，否则，集合$S$就不是攻击者的最佳响应策略。根据$p_i^{a_i}$的定义，很容易知道$p_i^0 > p_i^1$，那么引理4.2中的条件可以进行如下的转变：

$$\min_{i \in S} p_i^1 v_i \geq \max_{i \notin S} p_i^0 v_i > \max_{i \notin S} p_i^1 v_i \tag{4-8}$$

因此，可知最多存在一个集合S满足上式，如果集合S存在，那么攻击者选择集合S作为其策略，端口i选择其相应的最佳响应策略就是一个Nash均衡。 □

定理 4.1 该博弈模型必然存在一个唯一的Nash均衡。

证明： 首先给出一个Nash均衡的充分必要条件。一个策略组合（$\mathbf{p}$，$\boldsymbol{a}$）是Nash均衡当且仅当存在一个λ 使得对每一个端口i满足：

- $a_i = 0 \Rightarrow p_i = p_i^0$ and $p_i v_i \leq \lambda$;
- $0 < a_i < 1 \Rightarrow p_i = p_i^{a_i}$ and $p_i v_i = \lambda$;
- $a_i = 1 \Rightarrow p_i = p_i^1$ and $p_i v_i \geq \lambda$.

给定攻击者的策略$\boldsymbol{a}$，通过上述条件以及$p_i^{a_i}$的定义，$p_i^{a_i}$是最小化端口i期望损失的最优p值，可以知道在策略组合（$\mathbf{p}$，$\boldsymbol{a}$）下每个端口都选择了自己的最佳响应策略。当给定端口的策略$\mathbf{p}$时，根据公式4-4，可知攻击者的最佳响应策略就是在保证$\sum_i a_i = K$的条件下，对$p_i v_i$值较高的端口i赋予$a_i = 1$，对于$p_i v_i$值相同的端口就以一定的概率随机选择，即$0 < a_i < 1$，而$p_i v_i$值较低的端口i赋予$a_i = 0$。因此，必然存在一个λ值作为攻击者攻击的分界线，即以概率1攻击$p_i v_i$值高于λ的端口，以一定的概率攻击$p_i v_i$值等于λ的端口。显然，满足上述条件的攻击者策略$\boldsymbol{a}$正是攻击者的最佳响应策略。因此，策略组合（$\mathbf{p}$，$\boldsymbol{a}$）均是双方的最佳响应策略，也就是Nash均衡。

接下来，介绍如何计算一个混合策略Nash均衡。首先，定义一个p_i关于λ的函数如下：

$$p_i(\lambda) = \begin{cases} p_i^0 & \lambda \geq p_i^0 v_i \\ p_i^1 & \lambda \leq p_i^1 v_i \\ \frac{\lambda}{v_i} & p_i^1 v_i < \lambda < p_i^0 v_i \end{cases} \tag{4-9}$$

可以看出这个函数是关于λ的非递减的连续函数，而且当$p_i^1 v_i < \lambda < p_i^0 v_i$时，该函数是严格递增的。接下来，定义一个$a_i$关于$\lambda$函数如下：

$$a_i(\lambda) = \begin{cases} 0 & p_i(\lambda) = p_i^0 \\ 1 & p_i(\lambda) = p_i^1 \\ a^* & \text{其他} \end{cases} \tag{4-10}$$

其中a^*是使得等式$p_i(\lambda) = p_i^a$成立的a值。因为$p_i^1 \leq p_i(\lambda) \leq p_i^0$并且$p_i(\lambda)$是关于$\lambda$的非递减的连续函数，所以$a_i(\lambda)$是关于$\lambda$的非递增的连续函数。相似地，当$a_i(\lambda)$的函数值大于0且小于1时，函数$a_i(\lambda)$是严格递减的。最后，定义

$$E(\lambda) = K - \sum_i a_i(\lambda) \tag{4-11}$$

显然$E(\lambda)$是一个关于λ的连续且递增的函数。如果$\lambda = 0$，则$\forall i \in N, a_i(\lambda) = 1$，其中$N$表示所有$n$个端口的集合，即$|N| = n$，所以$E(\lambda) = K - n < 0$。如果$\lambda$的值很大以至于对于每一个端口$i$，都满足$a_i(\lambda) = 0$，那么$E(\lambda) = K - 0 > 0$。因此，可以找到一个使得函数$E(\lambda) = 0$的值$\lambda^*$。

如果这个博弈模型有纯策略Nash均衡，根据引理4.2可知在第K大的$p_i^1 v_i$值和第$K+1$大的$p_i^0 v_i$值之间是有差值的，当λ在这个差值中取值时，$E(\lambda) = 0$，这时的λ^*不唯一。虽然这种情况下λ^*不唯一，但是其所得出的纯策略Nash均衡是唯一的。如果这个博弈模型没有纯策略均衡，由于函数$E(\lambda)$是连续函数，那么λ^*是唯一的。

最后，可以发现由$p_i(\lambda^*)$和$a_i(\lambda^*)$计算出的双方策略所组成的策略组合是一

个Nash均衡，因为其满足Nash均衡的充分必要条件，也就是对于任意的λ^*，满足Nash均衡充分必要条件的策略组合就是由$p_i(\lambda^*)$和$a_i(\lambda^*)$计算出的双方策略所组成的策略组合。因为在一个混合策略Nash均衡中，λ^*是唯一的，所以其Nash均衡策略是唯一的。在纯策略Nash均衡中，虽然λ^*不唯一，但是其所得出的均衡策略是唯一的。因此，这个博弈模型必然存在唯一的Nash均衡。 □

定理4.1证明了模型Nash均衡的存在唯一性，并且其证明过程提供了一个理论上求解Nash均衡的方法。但是使用该方法计算Nash均衡需要知道博弈的完全信息，而在实际情况中，由于每个用户都是自私独立的个体不愿意将自己的信息共享，也可能并不了解参与博弈的其他用户信息，所以用户无法知道博弈的完全信息。通过上述方法直接计算Nash均衡并不可行的，因此提出了一个重复博弈的学习框架，分别为端口和攻击者设计了自适应学习算法，使得端口和攻击者能够在不断的重复博弈过程中，通过学习的方法不断调整自己的策略收敛到Nash均衡。

4.3 求解防御策略的学习框架

本节介绍所提出的重复博弈的学习框架，如算法1所示。首先，每个端口依据用户的信息对v_i和c_i的值进行初始化；接着，每个端口选择自己的防御策略，同时攻击者也做出自己的决策（步骤3-4）；然后，每个参与者使用自己所选择的策略参与博弈，之后就会接收到博弈的结果信息，即自己的收益等（步骤5）；最后，端口和攻击者分别依据自己的学习算法更新自己的策略（步骤6-7）。端口和攻击者的学习算法分别是对虚假博弈算法（Fictitious Play）和策略爬山法（Policy Hill-Climbing，PHC）的扩展，接下来，将分别对这两个学习算法进行详细地介绍。

Algorithm 1 重复博弈学习框架

1: 对所有的端口i，初始化v_i和c_i值;
2: **for** 每一轮t **do**
3: 每一个端口选择自己的防御策略p_i;
4: 攻击者基于其混合策略$\boldsymbol{a}$选择攻击的端口集合S;
5: 所有参与者参与博弈，收到博弈结果信息;
6: 每一个端口记录反馈信息，并根据学习算法更新自己历史信息;
7: 攻击者根据学习算法更新自己的策略;
8: **end for**

4.3.1 防御方的学习算法

通过观察端口i的损失函数$L_i^{a_i} = a_i p_i v_i + F(p_i)c_i$（公式4-5），其中$F(p)$是一个连续严格递减的凸函数，可以发现如果能够知道$a_i$的值，那么就可以很容易地计算出端口的最优防御策略，即能够最小化损失函数的p_i值。基于这个想法，通过借鉴虚假博弈算法的思想为端口设计了一个学习算法。该学习算法的主要思想就是根据之前的博弈交互信息对a_i的值进行预测，然后基于预测出的a_i值计算出端口i的最优防御策略p_i。在所设计的算法中，使用在历史交互过程中端口被攻击的频率来对a_i的值进行预测，整个学习算法流程如算法2所示。

Algorithm 2 防御方（端口i）的学习算法

1: 初始化: $attackflag \leftarrow 0$，$A_p \leftarrow 0$;
2: **for** 每一轮t **do**
3: 依据下式进行计算，选择最优的防御策略p_i
$p_i \leftarrow \arg\min_{p_i} L_i^{A_p}$;
4: 使用策略p_i参与博弈，接收博弈结果信息;
5: **if** 端口i被攻击了 **then**
6: $attackflag \leftarrow 1$;
7: **else**
8: $attackflag \leftarrow 0$;
9: **end if**
10: 依据下式更新概率值A_p
$A_p \leftarrow A_p + \frac{1}{t}(attackflag - A_p)$;
11: **end for**

算法2中的变量$attackflag$用来记录在当前轮端口i是否被攻击。当该端口被攻击时，其值为1，否则其值为0。变量A_p就是用来记录端口i对a_i的预测值，它可以被初始化为0到1的任何值，初始值的大小表示了端口对自己是否被攻击的最初估计。这里初始化为0，表明端口是乐观的，认为当前自己不会被攻击。初始化步骤后，端口基于预测值A_p计算自己的最优防御策略，即使得其期望损失最小的p_i值（步骤3）；接着，端口参与博弈过程然后接收博弈结果信息，记录在当前博弈中自己是否被攻击；最后，根据记录的信息更新变量$attackflag$和A_p的值（步骤5-10）。

4.3.2 攻击者的学习算法

通过观察攻击者的期望收益函数$U_{\text{attacker}} = \sum_{i \in S} p_i v_i$（公式4-4）以及约束条

件$\sum_{i\in N} a_i = K$，可以知道一个理性的攻击者为了实现自己收益的最大化，将会以较高的概率a_i攻击拥有较高$p_i v_i$值的端口。本质上来说，$p_i v_i$的值就是攻击者攻击了使用策略p_i的端口i所能获得的收益。然而，在博弈开始前攻击者并不能提前获知每个端口的pv值，因此使用Q值对每个端口的pv值进行估计。Q_i值也就代表了攻击者对攻击端口i能够获得的收益的一个估计。在攻击者的学习算法中，使用a_i表示攻击者攻击端口i的概率，根据Q值的大小采用基于策略梯度的算法对a_i的值进行调整更新，算法3展示了攻击者的学习算法流程。

Algorithm 3 攻击者的学习算法

1: 初始化: $\forall i \in N\ a_i \leftarrow \frac{K}{N}$，$Q_i \leftarrow 0$;
2: **for** 每一轮t **do**
3: 基于概率向量$\boldsymbol{a}$选择K个端口组成集合S，作为其策略;
4: 使用策略S参与博弈，接收博弈结果信息r;
5: **for** 每一个端口i **do**
6: **if** $i \in S$ **then**
7: $Q_i \leftarrow (1-\alpha)Q_i + \alpha r_i$
8: **end if**
9: **end for**
10: 选择Q值最大的K个端口组成集合C;
11: 选择Q值最小的K个端口组成集合D;
12: 依据下式更新$\boldsymbol{a}$

$$a_i \leftarrow \begin{cases} a_i + \delta & i \in C \text{ 且} i \notin D \\ a_i - \delta & i \in D \text{ 且} i \notin C \\ a_i & \text{otherwise} \end{cases};$$

13: **end for**

在攻击者的学习算法中，变量$\boldsymbol{a}$表示攻击者的混合策略，将a_i初始化为$\frac{K}{N}$，这表明在初始阶段由于没有历史信息，攻击者将以相同的概率攻击每一个端口。首先，攻击者基于其混合策略$\boldsymbol{a}$选择K个端口组成端口集合S，作为其参与博弈的纯策略（步骤3）；然后，攻击者对集合SS中的端口进行攻击，收到博弈的结果信息，即自己的收益；接下来，根据收到的收益信息对Q值进行更新（步骤5-9）；最后，根据Q值的大小更新攻击者的混合策略$\boldsymbol{a}$（步骤10-12）。对于攻击者混合策略$\boldsymbol{a}$的更新借鉴了策略爬山法的思想。在PHC算法中，Q值最大的对应的a值会增加，其他的都会减小。在攻击者的学习算法中，挑选Q值最大的K个端口组成集合C，Q值最小的K个端口组成集合D。在集合C中的所有端口的a_i值都会增加，在集合C中的所有端口的a_i值都会减少，剩下的所有端口的a_i值不变。与此同时，必须保证$\boldsymbol{a}$满足如下约束条件：$0 \le a_i \le 1$，$\sum_i a_i = K$。

4.4 学习算法实验评估

本节对所提出的端口和攻击者的学习算法进行实验评估。定理4.1的证明过程提供了一个计算博弈模型的理论Nash均衡解的方法，因此可以计算出模型的理论上的Nash均衡。通过将仿真实验结果与理论Nash均衡结果进行对比来验证所提出的算法的有效性。

4.4.1 实验设置

在实验之前，需要知道攻击者获得有效信息的概率p与通讯延时程度q之间的对应关系，即函数$q = F(p)$，这里直接沿用3.3.1节的仿真实验方法来获得p与q之间的关系。对于用户参数的设置问题，即对v_i和c_i值的初始化，依据3.3.2节的分析，这里假设v_i和c_i值服从幂律分布或是正态分布。基于上述分析，进行了以下四种情况的实验：

1. 两个用户和一个攻击者的情况，两个用户的参数相同，攻击者只能攻击一个用户即$K = 1$。
2. 两个用户和一个攻击者的情况，两个用户的参数不同，攻击者只能攻击一个用户即$K = 1$。
3. 八个用户和一个攻击者的情况，八个用户的参数服从幂律分布，攻击者可以攻击两个用户即$K = 2$。
4. 八个用户和一个攻击者的情况，八个用户的参数服从正态分布，攻击者可以攻击两个用户即$K = 2$。

4.4.2 实验结果及分析

本节所有的实验结果都是在进行了100次重复实验之后，将实验结果取平均后获得的。第一种实验情况中，用户的参数设置为：$v_1 = v_2 = 2$，$c_1 = c_2 = 1$。显然，攻击者的最佳响应策略是以相同的概率攻击每个用户。因此，可以很容易地计算出这个博弈的混合策略Nash均衡为（$\mathbf{p}$，$\boldsymbol{a}$），其中$\mathbf{p} = (0.1, 0.1)$，$\boldsymbol{a}$=(0.5, 0.5)。在该Nash均衡下，攻击者的期望收益为0.2。图4-1展示了用户策略的实验结果，从图中可以看出两个用户的策略都很快地收敛到了Nash均衡，并且最后稳定在了Nash均衡。攻击者策略的实验结果如图4-2所示，可以发现攻击者的策略在Nash均衡附近轻微波动。虽然攻击者的策略并没有最终稳定，但是在实验中攻击者的期望收益（图4-3所示）很快就稳定在其Nash均衡中的期望收益值（0.2）。

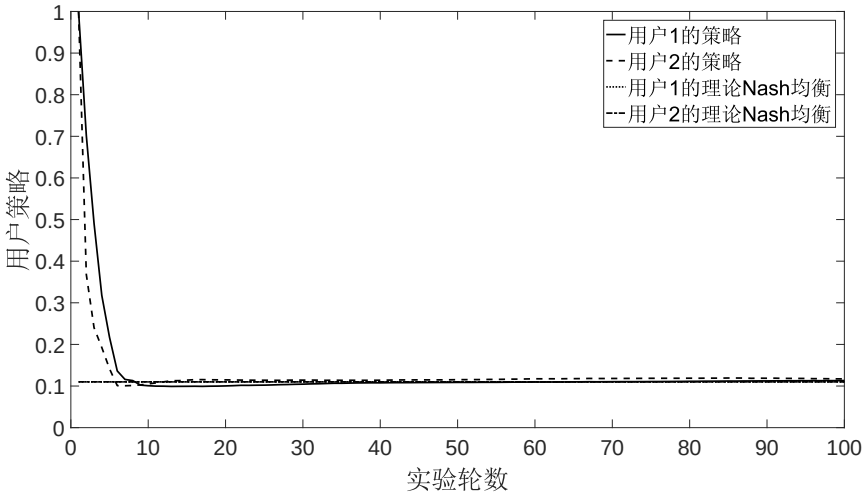


图 4-1　　用户的策略（用户参数相同）

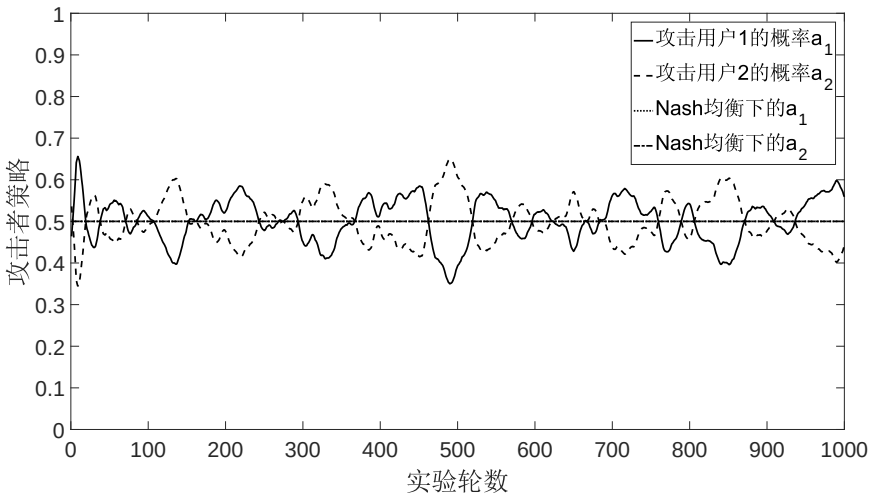


图 4-2　　攻击者的策略（用户参数相同）

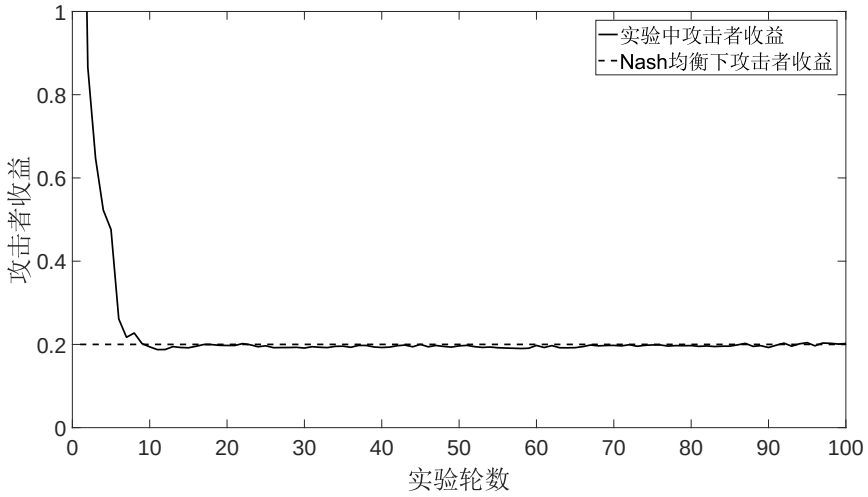


图 4-3　　攻击者的收益（用户参数相同）

第二种实验情况下，用户的参数设置为：$v_1 = 1$，$v_2 = 2$，$c_1 = 1$，$c_2 = 2$。用户策略的实验结果如图4-4所示，由于第2个用户的信息价值v_2较高，在实验刚开始的一段时间里，攻击者会以较高的概率攻击用户2，此时用户1可能就会放松自己的防御，从图中可以看出用户1的策略有一个上升的阶段，最终两个用户的策略都收敛到了Nash均衡。图4-5展示了攻击者策略的实验结果，从图中可以发现攻击者的策略依旧会在Nash均衡附近波动。通过观察攻击者期望收益的实验结果如图4-6所示，可以发现其期望收益很快收敛并稳定到了理论Nash均衡下的期望收益值（0.22）。

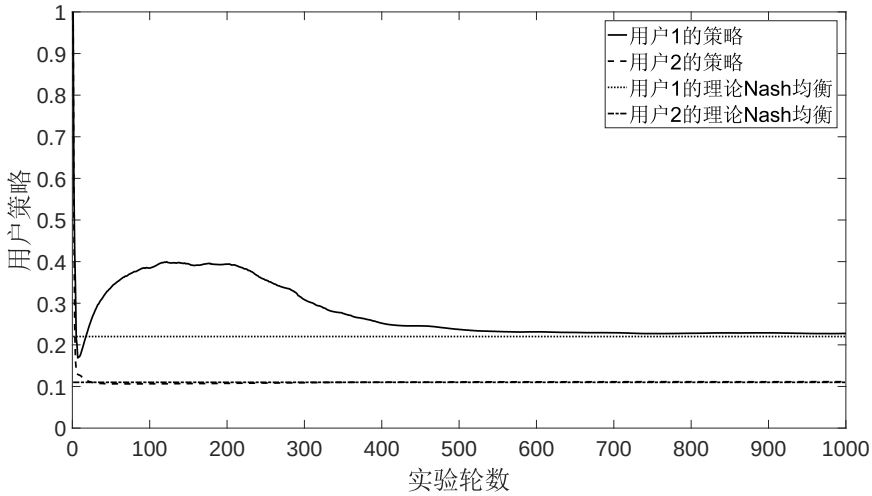


图 4-4　　用户的策略（用户参数不同）

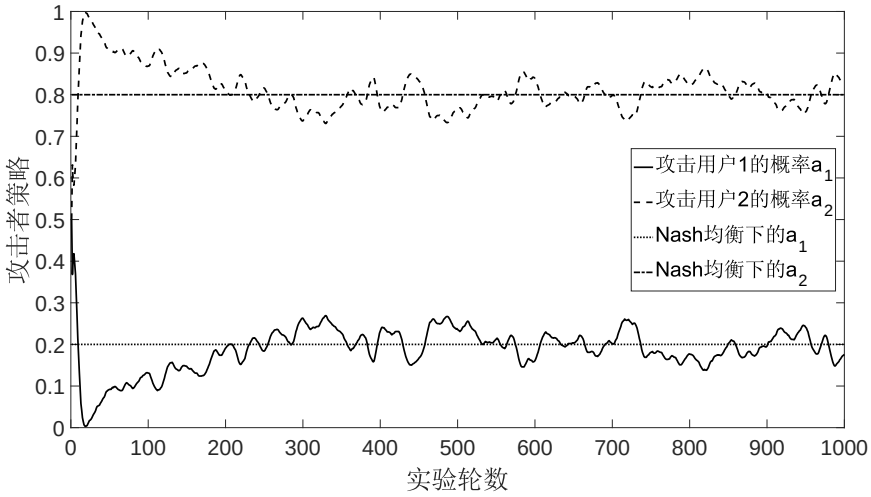


图 4-5　　攻击者的策略（用户参数不同）

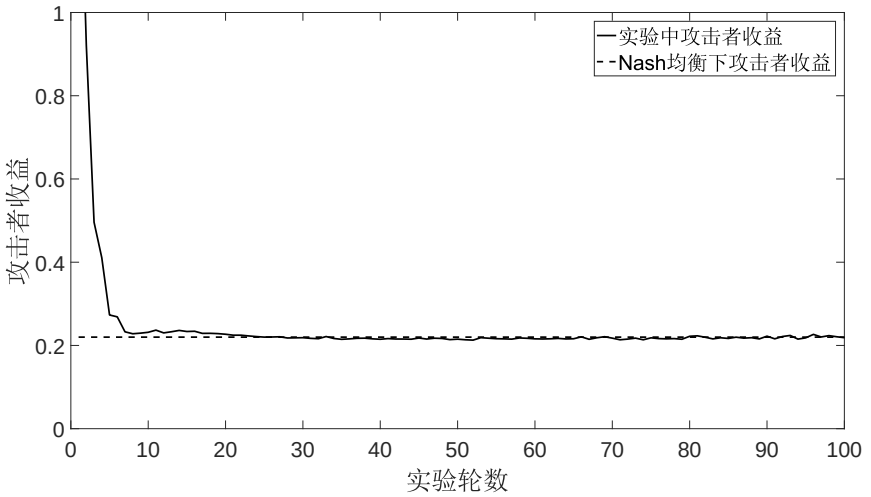


图 4-6　　攻击者的收益（用户参数不同）

第三、四种实验设置中考虑了更多用户的情况，用户的参数分别服从幂律分布和正态分布。表4-2和表4-3分别展示了当用户参数服从幂律分布和正态分布时，在实验进行到1000轮后用户策略的实验结果、理论Nash均衡以及两者的绝对误差和相对误差。由表中数据可以看出所有的相对误差都在5%以下，随着实验轮数的不断增加，这个误差将会变小，因为用户可以使用更多的历史交互信息，从而对a_i值的估计也就更加精确。对于攻击者来说，它的策略依旧会在Nash均衡附近波动。图4-7和图4-8分别展示了幂律分布和正态分布下攻击者期望收益的实验结果，通过观察可以发现攻击者的期望收益可以在较短时间里收

敛并稳定到Nash均衡下其期望收益的结果。

表 4-2　用户的策略（幂律分布）

用户编号	理论结果	实验结果	绝对误差	相对误差
user 1	0.2073	0.2176	0.0103	4.96%
user 2	0.1756	0.1764	0.0008	0.45%
user 3	0.1702	0.1726	0.0024	1.41%
user 4	0.1603	0.1654	0.0051	3.15%
user 5	0.2006	0.2105	0.0099	4.93%
user 6	0.2342	0.2451	0.0109	4.65%
user 7	0.2277	0.2320	0.0043	1.88%
user 8	0.2265	0.2248	0.0017	0.75%

表 4-3　用户的策略（正态分布）

用户编号	理论结果	实验结果	绝对误差	相对误差
user 1	0.2074	0.2044	0.0030	1.45%
user 2	0.1373	0.1434	0.0061	4.44%
user 3	0.1839	0.1838	0.0001	0.05%
user 4	0.2270	0.2378	0.0108	4.76%
user 5	0.7604	0.7882	0.0278	3.66%
user 6	0.3361	0.3492	0.0131	3.89%
user 7	0.2247	0.2326	0.0079	3.52%
user 8	0.0944	0.0958	0.0014	1.48%

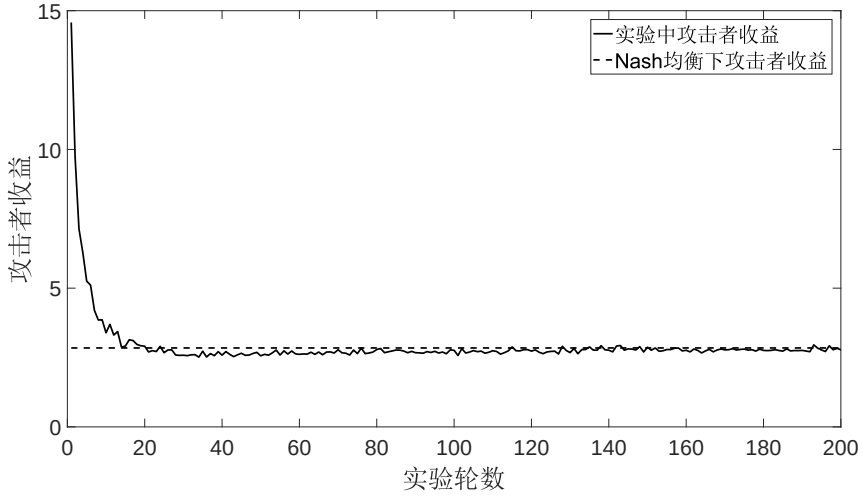


图 4-7　攻击者的收益（幂律分布）

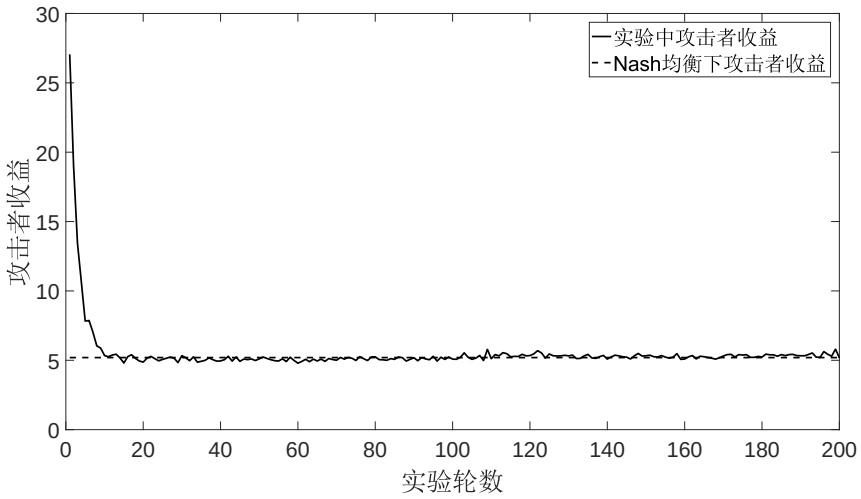


图 4-8　　攻击者的收益（正态分布）

通过对实验结果进行分析，可以知道在一定的误差范围内，学习算法可以保证用户的策略收敛并稳定在Nash均衡。虽然攻击者的策略并没有最终稳定在Nash均衡，而是会在Nash均衡附近波动，但是攻击者的期望收益基本稳定在Nash均衡所对应的期望收益。唯一不足之处在于收敛轮数会随着用户个数的增加而增加，但是用户的损失在其策略没有收敛到Nash均衡之前会很快降到一个较小的数值，这表明所提出的学习算法在进行很短的实验轮数后得到的策略可以有效降低用户损失。

4.5　本章小结

本章从个体的角度出发，研究如何为用户设计防御策略使得其在遭受中间人攻击时自身的损失最小。基于用户是理性自私的且攻击者无法提前获知用户策略的假设，采用同时行动博弈模型对问题进行建模；证明了模型Nash均衡的存在唯一性并给出了一个计算Nash均衡的理论算法；之后，分别为防御方和攻击者设计了自适应的学习算法；最后实验结果验证了学习算法的有效性。

第5章 总结与展望

5.1 总结

随着互联网深入人们的生活，给人们带来了许多便利，但同时网络安全问题开始受到关注。层出不穷的网络攻击使得人们越来越关注对自己隐私信息的保护，所以目前网络攻击的防御问题成为研究的热点。中间人攻击是黑客们窃取用户敏感信息常用的一种网络攻击手段，本研究基于安全博弈论的相关理论来解决中间人攻击防御问题。首先介绍了安全博弈论的相关概念以及中间人攻击的常用技术并分析了所研究的中间人攻击防御措施，之后从整体和个体两个角度出发，研究如何设计防御策略使得在遭受中间人攻击时防御方的损失最小。主要研究总结如下：

（1）从整体角度分析，就是以降低整体的损失为目标为防御方设计防御策略。将服务器与所有用户看作一个整体作为防御方，设计了最优防御策略保证在遭受中间人攻击时与服务器通讯的所有用户的整体损失最小。首先，使用Stackelberg博弈模型对防御方与攻击者之间的交互过程进行建模，采用模型的强Stackelberg均衡作为防御方的最优防御策略。由于防御方的策略空间是无限大的，不能使用传统的求解强Stackelberg均衡的方法，提出了一个能够减小计算防御方最优防御策略时搜索空间的方法，从而可以有效地求解最优防御策略。最后通过与其他非策略性的防御策略进行实验对比，结果表明在遭受中间人攻击时防御方的最优防御策略在降低整体损失方面明显优于其他非策略性的防御策略。

（2）从个体角度分析，就是以降低个体自身损失为目标为防御方设计防御策略。每个用户看作一个独立的防御方，因此这是一个多个防御方与一个攻击者之间的博弈。首先将多个防御方和一个攻击者之间的交互过程建模为同时行动博弈模型，采用常见的Nash均衡作为模型的解。之后证明了模型Nash均衡的存在唯一性并提出了一种理论上求解Nash均衡的方法。由于在实际中很难知道博弈的完全信息，并不能直接计算出Nash均衡，因此分别为博弈双方设计了相应的自适应学习算法使得双方能够在重复博弈过程中收敛到Nash均衡。最后通过实验与Nash均衡理论结果进行比较，结果表明学习算法能够使得防御方策略收敛并稳定到Nash均衡，攻击者策略在Nash均衡附近波动，但其期望收益收敛到Nash均衡的期望收益。

5.2 展望

本文应用安全博弈论的相关理论来研究中间人攻击防御问题。虽然已经从不同角度分析了中间人攻击防御问题，设计了相应的防御策略来降低遭受中间人攻击时的损失，但是研究也只是中间人攻击防御问题很小的一部分。由于在研究中对问题定义有一些基本的假设，因此研究结果只适用于一些特定的情况。针对中间人攻击防御方面的研究仍存在许多问题亟待解决，未来可以从以下几方面进行深层次的研究：

（1）如果攻击者由于一些不确定因素不能攻击选定的端口或者攻击者并不是完全理性的，这就会导致攻击者并不会选择自己的最佳响应策略，也就偏离了均衡策略。如果这时防御方仍旧选择均衡策略作为防御策略，可能就不是其最优选择。此时可以考虑首先对攻击者行为进行建模，基于对攻击者的行为建模通过设计相应的算法来求得防御方的能够最小化其损失的防御策略。

（2）如果考虑服务器可以提供多种服务时，由于服务类型的不同可能导致端口被攻击的概率不同。这时可以考虑使用贝叶斯Stackelberg模型对其进行建模分析，从而获得防御方的最优防御策略。

（3）目前，安全博弈论在网络安全方面的研究还不是很多，可以考虑将博弈论的相关理论应用到其他网络攻击防御问题中去。

参考文献

[1] Conti M, Dragoni N, Lesyk V. A survey of man in the middle attacks [J]. IEEE Communications Surveys & Tutorials, 2016, 18 (3): 2027–2051.

[2] 安波. 安全博弈论 [J]. 中国计算机学会通讯, 2013, 9 (1): 58–63.

[3] 王震, 袁勇, 安波, et al. 安全博弈论研究综述 [J]. 指挥与控制学报, 2015, 1 (2): 121–149.

[4] 王立彦. HTTPS 协议中间人攻击的实现与防御 [D]. 沈阳: 东北大学, 2011.

[5] Mishra P. Analysis of MITM attack in Secure Simple Pairing [J]. Journal of Global Research in Computer Science, 2013, 4 (2): 42–45.

[6] 龚庭楠. 关于网络欺骗攻击实现和防范的研究 [D]. 南京: 南京邮电大学, 2012.

[7] Albina M N, Raju U, Revathi G K, et al. Protection against Man-in-the-middle Attack in Banking Transaction using Steganography [J]. International Journal of Scientific & Engineering Research, 2013: 457–464.

[8] Kumar C K, Jose G J A, Sajeev C, et al. Safety measures against man-in-the-middle attack in key exchange [J]. Journal of Engineering & Applied Sciences, 2012, 7 (2): 243–246.

[9] 郭卫兴, 刘旭, 吴灏. 基于ARP缓存超时的中间人攻击检测方法 [J]. 计算机工程, 2008, 34 (13): 133–135.

[10] Vallivaara V A, Sailio M, Halunen K. Detecting man-in-the-middle attacks on non-mobile systems [C]. In ACM Conference on Data and Application Security and Privacy, 2014: 131–134.

[11] Dacosta I, Ahamad M, Traynor P. Trust no one else: Detecting MITM attacks against SSL/TLS without third-parties [C]. In European Symposium on Research in Computer Security, 2012: 199–216.

[12] Huang L S, Rice A, Ellingsen E, et al. Analyzing Forged SSL Certificates in the Wild [C]. In IEEE Symposium on Security and Privacy, 2014: 83–97.

[13] 欧阳江, 李勍, 刘嘉勇. SSL-Webmail 中间人监测技术研究 [J]. 信息安全与通信保密, 2012 (5): 69–71.

[14] 郭润, 王振兴, 敦亚南. 交换式以太网中的ARP与DNS欺骗技术分析 [J]. 微计算机信息, 2005, 21 (10X): 21–23.

[15] An B, Tambe M, Sinha A. Stackelberg security games (SSG): Basics and application overview [J]. Improving Homeland Security Decisions. Cambridge University Press, forthcoming, 2015.

[16] Stackelberg H V. Marktform Und Gleichgewicht [M]. Berlin: Springer, 1934.

[17] Conitzer V, Sandholm T. Computing the optimal strategy to commit to [C]. In Proceedings of the 7th ACM conference on Electronic commerce, 2006: 82–90.

[18] Jain M, Tsai J, Pita J, et al. Software Assistants for Randomized Patrol Planning for the LAX Airport Police and the Federal Air Marshal Service [J]. Interfaces, 2010, 40 (4): 267–290.

[19] Shieh E A, An B, Yang R, et al. PROTECT: An Application of Computational Game Theory for the Security of the Ports of the United States. [C]. In Twenty-Sixth AAAI Conference on Artificial Intelligence, 2012: 2173–2179.

[20] An B, Shieh E, Yang R, et al. PROTECT - A Deployed Game-Theoretic System for Strategic Security Allocation for the United States Coast Guard [J]. Ai Magazine, 2012, 33 (4): 96–110.

[21] Kiekintveld C, Islam T, Kreinovich V. Security games with interval uncertainty [C]. In Proceedings of the 2013 international conference on Autonomous agents and multi-agent systems, 2013: 231–238.

[22] Shakarian P, Lei H, Lindelauf R. Power grid defense against malicious cascading failure [C]. In Proceedings of the 2014 international conference on Autonomous agents and multi-agent systems, 2014: 813–820.

[23] Tsai J, Yin Z, Kwak J Y, et al. Urban Security: Game-Theoretic Resource Allocation in Networked Domains [C]. In Twenty-Fourth AAAI Conference on Artificial Intelligence, 2011: 881–886.

[24] Johnson M P, Fang F, Tambe M. Patrol strategies to maximize pristine forest area [C]. In Twenty-Sixth AAAI Conference on Artificial Intelligence, 2012: 295–301.

[25] Yang R, Ford B, Tambe M, et al. Adaptive resource allocation for wildlife protection against illegal poachers [C]. In Proceedings of the 2014 international conference on Autonomous agents and multi-agent systems, 2014: 453–460.

[26] Haskell W B, Kar D, Fang F, et al. Robust protection of fisheries with COmPASS [C]. In Twenty-Eighth AAAI Conference on Artificial Intelligence, 2014: 2978–2983.

[27] Zhao M, An B, Kiekintveld C. Optimizing Personalized Email Filtering Thresholds to Mitigate Sequential Spear Phishing Attacks [C]. In Thirtieth AAAI Conference on Artificial Intelligence, 2016: 658–665.

[28] Gutierrez M P, Kiekintveld C. Bandits for Cybersecurity: Adaptive Intrusion Detection Using Honeypots [C]. In Workshops at the Thirtieth AAAI Conference on Artificial Intelligence, 2016: 165–166.

[29] Laszka A, Vorobeychik Y, Koutsoukos X D. Optimal Personalized Filtering Against Spear-Phishing Attacks. [C]. In Twenty-Ninth AAAI Conference on Artificial Intelligence, 2015: 958–964.

[30] Laszka A, Lou J, Vorobeychik Y. Multi-Defender Strategic Filtering Against Spear-Phishing Attacks [C]. In Thirtieth AAAI Conference on Artificial Intelligence, 2016: 537–543.

[31] Xu H, Tran-Thanh L, Jennings N R. Playing Repeated Security Games with No Prior Knowledge [C]. In Proceedings of the 2016 International Conference on Autonomous Agents & Multiagent Systems, 2016: 104–112.

[32] Klíma R, Kiekintveld C, Lisy V. Online Learning Methods for Border Patrol Resource Allocation [C]. In International Conference on Decision and Game Theory for Security, 2014: 340–349.

[33] Klíma R, Lisy V, Kiekintveld C. Combining Online Learning and Equilibrium Computation in Security Games [C]. In International Conference on Decision and Game Theory for Security, 2015: 130–149.

[34] 范如国. 博弈论 [M]. 武汉: 武汉大学出版社, 2011.

[35] 李光久, 李昕. 博弈论简明教程 [M]. 江苏: 江苏大学出版社, 2013.

[36] 焦宝聪, 陈兰平. 博弈论 [M]. 北京: 首都师范大学出版社, 2013.

[37] Osborne M J, Rubinstein A. A course in game theory [M]. Massachusetts: MIT press, 1994.

[38] 侯定丕. 博弈论导论 [M]. 合肥: 中国科学技术大学出版社, 2004.

[39] 马洪宽. 博弈论 [M]. 上海: 同济大学出版社, 2015.

[40] Yin Z, Korzhyk D, Kiekintveld C, et al. Stackelberg vs. Nash in security games: interchangeability, equivalence, and uniqueness [C]. In International Conference on Autonomous Agents and Multiagent Systems: Volume, 2010: 1139–1146.

[41] Leitmann G. On generalized Stackelberg strategies [J]. Journal of Optimization Theory and Applications, 1978, 26 (4): 637–643.

[42] Breton M, Alj A, Haurie A. Sequential Stackelberg equilibria in two-person games [J]. Journal of Optimization Theory and Applications, 1988, 59 (1): 71–97.

[43] 李星伟, 沈磊, 曾磐. 基于数据包的中间人攻击分析 [J]. 福建电脑, 2013, 29 (2): 91–92.

[44] 储柱学, 黄莺. 局域网下基于ARP欺骗的中间人攻击与防御 [J]. 佳木斯职业学院学报, 2012 (8): 428–428.

[45] 乔艳飞. SSL 安全分析以及中间人攻击和防范研究 [D]. 北京: 北京邮电大学, 2013.

[46] 贾静, 薛质. SSL中间人攻击原理与防范 [J]. 信息安全与通信保密, 2007 (4): 103–105.

[47] Luo Y B, Wang B S, Cai G L. Effectiveness of Port Hopping as a Moving Target Defense [C]. In SecTech, 2014: 7–10.

[48] Luo Y B, Wang B S, Cai G L. Analysis of Port Hopping for Proactive Cyber Defense [J]. International Journal of Security & Its Applications, 2015, 9 (2): 123–134.

[49] Chandler D, Percus J K. Introduction to Modern Statistical Mechanics [M]. Oxford: Oxford University Press, 1987.

发表论文和参加科研情况说明

（一）发表的学术论文

[1] Xiaohong Li, Shuxin Li, Jianye Hao, Zhiyong Feng, Bo An. Optimal personalized defense strategy against man-in-the-middle attack[C]. In Proceedings of the 31st AAAI Conference on Artificial Intelligence, 2017: 593-599.

[2] Shuxin Li, Xiaohong Li, Jianye Hao, Bo An, Zhiyong Feng, Kangjie Chen, Chengwei Zhang. Defending against man-in-the-middle attack in repeated games[C]. In Proceedings of the 26th International Joint Conference on Artificial Intelligence, 2017: 3742-3748.

（二）申请及已获得的专利

[1] 李晓红，李姝昕. 基于AADL的Web应用架构安全性的评估方法：中国，201710333755.7[P]. 2017-05-12.

（三）参与的科研项目

[1] 泛在接入条件下移动应用安全管家技术研究, 国家自然科学基金项目. 课题编号：No.61572349, 2016.01-2019.12.

致　　谢

本篇论文的工作是在我的导师李晓红教授的悉心指导下完成的，李晓红教授严谨的治学态度、精益求精的工作作风，诲人不倦的高尚师德和科学的工作方法给了我极大的帮助和影响。李晓红教授悉心指导我们完成了实验室的科研工作，在学习上和生活上都给予了我很大的关心和帮助，在此向李晓红老师表示衷心的谢意。

在研究生科研期间，从课题的选择到论文的最终完成，郝建业老师以及冯志勇老师都始终给予我细心的指导和不懈的支持。各位老师对于我的科研工作和论文帮助很多，提出了许多的宝贵意见，在此，我要向老师们深深地鞠上一躬表示衷心的感谢。

在实验室工作及撰写论文期间，同一实验室的曹茹、杨薇、何慧娟、王江娟以及刘云昊等同学对我论文中的关于中间人攻击的研究工作给予了热情帮助，实验室师兄张程伟以及陈康杰同学对我论文中的关于学习算法的研究给与了许多帮助，在此向他们表达我的感激之情。

另外也感谢我的家人，是他们的理解和支持使我能够在学校专心完成我的学业。最后，在此再次衷心感谢两年半的研究生生涯里，所有老师、同学以及家人对我的关心和指导。