无线网络中基于共享密钥的轻量级匿名认证协议

钟 成 李兴华 宋园园 马建峰

(西安电子科技大学网络与信息安全学院 西安 710071)

摘 要 随着人们对隐私保护问题的关注,无线网络环境下身份认证的匿名问题越来越引起人们的重视.目前大部分匿名身份认证方案都是基于非共享密钥,此类方案计算量大导致资源消耗严重,对于一些计算能力有限的设备并不适用.同时对于基于共享密钥的方案,存在易被追踪或存储开销较大等问题.通过分析和实验证明 Li 等人所提出的基于共享密钥的方案不能抵抗时间关联攻击,从而泄露用户身份信息,进一步考虑现有常数时间认证方案存储开销较大的问题,引入用户分组机制,在 Li 等人基于共享密钥认证方案的基础上提出了一种基于共享密钥的轻量级匿名认证方案.通过对用户进行分组并且分配对应的组标识,认证阶段用户仅需要发送组标识和共享密钥的哈希信息到认证服务器,认证服务器根据组标识遍历对应分组的共享密钥验证认证用户的真实身份信息,并完成认证过程.形式化的安全证明说明了协议的安全性和匿名性,进一步的安全分析和实验表明,所提方案不仅具有更高的安全性,而且具有计算开销、通信开销和存储开销小等优点.

关键词 无线网络;共享密钥;匿名认证;时间关联攻击;轻量级

中图法分类号 TP309 **DOI 号** 10.11897/SP.J.1016.2018.01157

A Lightweight Anonymous Authentication Protocol Based on Shared Key in Wireless Networks

ZHONG Cheng LI Xing-Hua SONG-Yuan-Yuan MA Jian-Feng

(*School of Cyber Engineering*, *Xidian University*, *Xi'an* 710071)

Abstract With the rapid development of wireless communication technology and the popularity of mobile intelligent terminal equipment. Wireless network has been used in many field, but if a user leaks his personal identity while using the wireless network, he will expose his personal privacy information. Since people pay more and more attention to the protection of their privacy, anonymous authentication in wireless networks has become a hot topic. Currently, most anonymous authentication schemes are based on the asymmetric keys. However, in those methods, the clients have to perform complex calculation, such as asymmetric encryption or decryption, which leads to serious resource consumption. Therefore, they are unsuitable for mobile devices with limited computing power and resource in wireless network environment. At the same time, the existing anonymous authentication scheme based on shared key has many problems, such as easy tracking or storage overhead issues. In this paper, theoretical analysis and experiments prove that Li proposed scheme based on the Shared key can't resist the attack on time correlation, and there is a linear relationship between the location of user identifier in the k-pseudonym set and the authentication time. Attacker can obtain the user's real identity in a very high probability using this correlation. Although existing constant time authentication scheme has a certain extent to

收稿日期:2017-06-07;在线出版日期:2017-11-29.本课题得到国家自然科学基金(U1708262,61672413)资助.**钟 成**,男,1994 年生,硕士研究生,主要研究方向为网络与信息安全、隐私保护.E-mail:czhongcs@126.com.**李兴华**(通信作者),男,1978 年生,博士,教授,博士生导师,主要研究领域为网络与信息安全、隐私保护、云计算、安全协议形式化方法.E-mail:xhli1@mail.xidian.edu.cn.**宋园园**,男,1990 年生,硕士研究生,主要研究方向为网络与信息安全、隐私保护.**马建峰**,男,1963 年生,博士,教授,主要研究领域为信息安全、编码理论、密码学.

solve the problem of inconsistent authentication time, but such solutions have the problem of storage overhead, and with the user number increasing, the search efficiency will be greatly reduced. In view of these problems, a lightweight anonymous authentication scheme for wireless networks is proposed in this paper. It is based on shared key, with a user grouping mechanism. In this scheme, the registered users are grouped and each group of users is assigned the corresponding group identifier (*GID*). In the authentication process, the user sends his group identifier and the hash information of his shared keys to authentication server. After the authentication server traversals the shared key with each of the users in the group and verifies the authentication information, it can determine the real user and complete the authentication safely and anonymously. In this methodology, the construction of the group is a key issue. Compared with the existing schemes, our scheme outperforms them in the security and practicality. With the grouping mechanism, even if the attacker knows the user's group identifier, the user cannot be distinguished from other users in the group, so that the specific user's information cannot be obtained. It is also very efficient because our scheme only use shared keys and only need to traversals the shared keys with each of the users in the group. The formalized security evidence illustrates the security and anonymity of the protocol. Security analyses and experiment results show this scheme can resist multiple attacks, such as association analysis attack, replay attack etc. and is very efficient at the same time. We also compare the scheme of this paper with the existing ones, the results show that our scheme has the advantages of small computing cost, storage cost and communication cost etc.

Keywords wireless networks; shared key; anonymous authentication; time correlation attack; lightweight

1 引 言

随着无线通信技术的迅猛发展和移动智能终端设备的普及,无线网络逐渐被应用于各个领域,然而在为人们提供便利的同时,也给人们的隐私安全带来了严重的威胁与挑战.如用户在使用无线网络的过程中泄露个人身份,从而暴露个人隐私信息.因此无线网络的匿名性已经成为一种基本的安全需求,目前对匿名认证方案的研究也已经成为一个研究热点.

现有的无线网络匿名认证方案主要分为两种,一类基于非共享密钥;另一类基于共享密钥.基于非共享密钥的匿名认证方案[1-3]主要是通过签名算法、零知识证明[4]等方法来实现匿名.此类方案需要的计算开销和存储空间需求均较大,考虑无线网络中大多是计算能力以及存储能力受限的移动设备,所以此类方案不适用于无线网络环境.而目前基于共享密钥的匿名认证方案[5-7]主要通过假名机制、Hash 机制等来实现匿名.最近,Li 等人[8]针对非共享密钥匿名认证方案的局限性,通过将包含用户真实身份的 k-假名集合和用户与认证服务器间的共享密钥发送给认证服务器进行匿名认证,认证服务器仅需要通过 k 次遍历即可完成对用户的认证过程,因此该方案可以显著降低认证过程的计算开销和存储开销.但是经过我们的分析与实验验证,认证过程所花费的时间和真实用户在 k-假名集合中的位置存在线性关系,如果攻击者利用时间关联分析进行攻击,那么可以以非常高的概率获取用户的真实身份,从而导致认证用户的身份信息泄露.同时现有常量时间认证方案虽然在一定程度解决认证时间不一致所带来的问题,但是此类方案以牺牲存储空间为代价,并且在用户增多的情况下搜索效率将显著降低,方案性能将下降.

针对以上问题,对无线网络中匿名认证方案进行研究,提出了一种基于共享密钥的更加安全高效的轻量级无线网络匿名身份认证协议.具体工作如下所述:

(1)通过对 Li 所提匿名认证协议进行分析,发现现有协议中认证服务器对用户认证请求的处理时延会导致用户真实身份信息的泄露,并进行攻击实验,通过对实验数据进行量化与分析证明现有协议确实存在上述安全缺陷.

(2)通过对现有匿名认证协议的改进，引入用户分组机制，使用组标识 *GID* 来替代用户的真实身份进行认证，保证了用户真实身份的安全，认证服务器最多对分组中所有成员进行验证即可完成认证，显著提高了认证效率，进而提出了无线网络中基于共享密钥的轻量级匿名认证协议，并给出安全模型，对本文所设计方案进行了形式化的安全性证明.

(3)对本文所设计协议从抗攻击性、双向认证性、前向保密性、后向保密性等方面进行安全性分析，并从认证时间、通信量、计算量和存储开销等方面进行性能分析. 结果表明与同类协议相比，所设计协议在安全性和性能方面均具有更好的表现.

本文在第 2 节中讨论相关工作；在第 3 节中分析 Li 所提方案的安全缺陷并通过仿真实验验证分析结果；第 4 节提出基于用户分组的更加安全高效的轻量级匿名认证方案，且进行安全证明；并在第 5 节进行实验验证并分析结果；最后在第 6 节对本文进行总结.

2　相关工作

2.1　基于非共享密钥的认证方案

Liu 等人[9]通过对无线体域网(Wireless Body Area Network，WBAN)的特点进行分析，提出一种无证书签名的远程匿名认证协议用于保证病人在使用远程医疗服务时的隐私安全. 该方法能够有效防止应用程序或者服务提供商获取用户的真实身份，从而保证了用户的安全. He 等人[10]对 Liu 等人的方案进行分析并指出其方案不能抵抗伪装攻击，进而通过将认证数据存储于网络管理器来保证用户认证过程中的匿名性. 文献[11-12]对车载网(Vehicular Ad-hoc Network，VANET)中的匿名认证方案进行了研究. Yein 等人[11]提出了一种基于椭圆曲线和零知识证明的安全匿名身份认证方案，在用户的认证过程中采用了双向匿名认证算法来保证用户的隐私，并通过别名机制和签名证书来隐藏车辆的真实身份，并且定期更新别名和证书保证车辆不被非法跟踪.

基于非共享密钥的匿名认证方案需要复杂的数学计算并且需要较大的存储需求，这对于计算能力和存储能力受限的移动终端并不适用.

2.2　基于共享密钥的认证方案

Gódor 等人[7]提出基于 Hash 机制的匿名认证方案，服务器在认证过程中查看用户具有匿名特性的证书来为用户提供匿名服务. 但是用户认证过程中所使用的具有匿名特性的证书中有一个 Holder 标识，且该标识的值对于同一个用户固定不变，因此攻击者可以通过该标识对用户进行跟踪. 文献[13-15]中通过将用户的共享密钥进行哈希处理或者 CRC 校验发送给认证服务器进行匿名认证，认证服务器根据数库中存储的共享密钥实现对用户真实身份的认证. 由于该过程需要认证服务器进行数据库遍历，最坏情况需要遍历数据库中所有的共享密钥并和原消息进行匹配，当数据库中数据量很大时，将会严重消耗认证服务器的资源，攻击者如果利用该漏洞发起大量的匿名认证请求则可以耗尽服务器资源，影响服务器对其他用户的正常认证. Li 等人[8]提出基于共享密钥的轻量级 k-假名匿名认证方案，用户在进行身份认证时将包含自己真实身份标识的 k-假名集合以及与认证服务器间的共享密钥的哈希信息发送给认证服务器，认证服务器在最多对 k 个用户的共享密钥进行遍历并验证其对应的哈希信息后就能够完成对用户的认证，避免了认证服务器资源过度消耗，并指出该方案能够有效抵抗增强 Dolev-Yao 模型[16]中的求交集攻击.

Li 所提方案虽然一定程度的解决了认证效率低和求交集攻击的问题，但是由于在认证过程中需要遍历 k-假名集合来验证用户身份，通过分析我们发现认证时延和用户在假名集合中的位置具有线性关系，攻击者通过对认证时间的分析，依然能够以较大概率发现用户的真实身份，因此该方案存在安全缺陷. 文献[17-18]考虑了由于认证时间不一致导致对用户关联的问题，通过提前计算所有可能需要的假名信息和认证计数值的哈希值并存储在数据库中，从而提出了常量时间的认证方案，但是由于该方案引入计数值来避免假名被跟踪从而需要存储大量的哈希值，导致认证服务器存储开销巨大，并且需要在全部数据库中搜索特定的用户. 文献[19]考虑上述搜索开销，提出基于二次剩余的常量时间认证方案，但是该方案需要进行大合数的模运算，故仍然具有较高的计算开销. 文献[20]采用分布式数据库的方式避免被跟踪，但是存储开销和计算开销依然较大.

考虑现有成果中存在的上述问题，本文引入分组机制提出了一种轻量级的匿名认证方案，用户在认证时只需要发送对应的组标识 *GID*，认证服务器仅需要根据分组在组内验证用户身份. 因此该方案更加安全高效.

3 Li 匿名认证协议安全缺陷

Li 提出的使用 k-假名集合的匿名认证协议引入了 Dolev-Yao 模型和增强的 Dolev-Yao 模型，并说明所设计出的方案可以有效抵抗在这两个模型中所涵盖的攻击.但是该模型没有考虑攻击者基于认证时间的关联分析，并且由于该方案的 k-假名集合中仍然包括了用户的真实身份标识，通过对认证时间的统计和分析，可以进一步推断出用户真实身份标识在假名集合中的位置，从而确定用户的真实身份，因此 Li 所设计出的协议具有安全缺陷.

3.1 认证时间关联分析

为了进一步分析 Li 方案的安全缺陷，这里首先给出 Li 方案的认证过程，具体如图 1，表 1 给出了协议中所使用的符号.

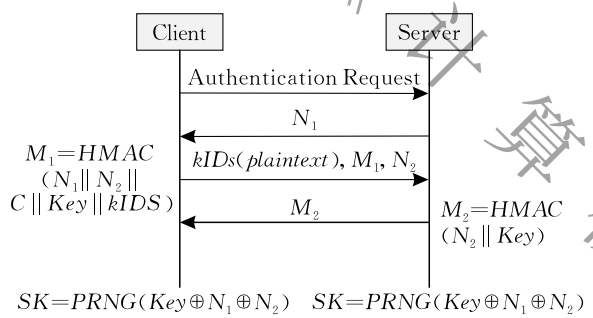

图 1 Li 方案认证交互图

表 1 Li 协议符号对照表

符号	意义
N_1	认证服务器生成的随机数
N_2	用户生成的随机数
C	用户真实身份标识
$HMAC$	哈希函数
$kIDs$	用户发送的 k 个身份标识集合
Key	用户和认证服务器的共享密钥
SK	用户和认证服务器的会话密钥
$PRNG()$	随机数生成器

Step1. 用户→认证服务器.首先由用户请求开始匿名认证，消息内容是 32 比特长的字符串信息.

Step2. 认证服务器→用户.认证服务器接收到用户的匿名认证请求后，发送随机数 N_1 给用户.

Step3. 用户→认证服务器.用户接收到认证服务器发来的随机数 N_1 后自己生成随机数 N_2，然后计算消息 M_1，并将 $kIDs$，M_1，N_2 消息发送给认证服务器，其中 $kIDs$ 为 k-假名集合.

Step4. 认证服务器→用户.认证服务器接收到用户发来的认证信息，按照 $kIDs$ 集合中用户身份标识顺序到数据库查询对应共享密钥.并计算消息 $M_1'=HMAC(N_1\|N_2\|C'\|Key\|kIDs)$，其中 C' 为 $kIDs$ 中的某个假名，比较 M_1' 和 M_1 是否相等，若相等则服务器对用户认证成功并停止遍历 k-假名集合中的其他成员.随后计算消息 M_2，并将 M_2 发送给用户端.

由于整个交互过程所需的时间消耗主要在 Step4 阶段，因此我们主要分析该阶段的时间信息.将用户真实身份标识 C 在 k-假名集合中的位置记作 r 且 Step4 的处理时间记作 Δ_t，其中易知时间 Δ_t 为

$$\Delta_t=T_{\text{verify}}+T_{\text{compute}}+T_{\text{delay}} \tag{1}$$

其中 T_{verify} 表示认证服务器遍历 k-假名集合验证用户身份所需要的时间，T_{compute} 表示认证服务器计算消息 M_2 所需要的时间，T_{delay} 表示消息接收与发送的时延.

T_{verify} 是认证服务器按照 k-假名集合中用户身份标识的顺序对用户身份进行验证所需要的时间，对 k-假名集合中每一个用户身份标识主要执行 3 项操作：(1) 按顺序取出集合中下一个用户身份标识，并到数据库查询对应的共享密钥；(2) 根据查询到的共享密钥计算消息 M_1'；(3) 比较 M_1 和 M_1' 是否相等，相等则结束遍历，不相等则返回(1)继续执行.由于对每一个用户身份标识的处理过程都是相同的，因此所耗费的时间也相同，设验证一个用户身份标识所需时间为 T_{average}，易知 T_{average} 为常数时间，那么 $T_{\text{verify}}=r\times T_{\text{average}}$，其中 r 为用户真实身份在假名集合中的位置.

T_{compute} 是计算消息 M_2 所需的时间，因此对于所有用户而言 T_{compute} 是常数时间.

T_{delay} 是消息接收与发送的时间，主要由两部分组成.一是认证服务器接收 Step3 用户发送的消息需要的时间，包括消息在信道上传输的时间.二是认证服务器向用户发送消息 M_2 需要的时间，包括消息在信道上传输的时间和竞争无线信道损耗的时间.由于消息在信道上传输速度很快，因此消息长度对传输时间的影响很小.此外根据文献[21-22]研究结果可知，当用户要发送的数据长度固定时，信道竞争时延与参与竞争的用户数量呈近似线性关系.因此在并发请求认证用户数量相对稳定的环境下，信道竞争时延相对稳定，因此 T_{delay} 为常数时间.基于上述描述可以进一步得到

$$\Delta_t=r\times T_{\text{average}}+T_{\text{compute}}+T_{\text{delay}} \tag{2}$$

根据前面分析可知 Δ_t 与 r 存在近似线性关系.假设公式是一个线性关系，其中 T_{average} 为比例系数，$T_{\text{compute}}+T_{\text{delay}}$ 为常数.因此如果能够确定该线性关

系的相关参数就能得到 Δ_t 与 r 之间的准确关系式. 使用得到的关系式并根据 Δ_t 就能计算出用户真实身份标识在集合中的准确位置 r，从而导致协议匿名性被破坏、用户身份信息泄露.

3.2　认证时间关联的攻击

根据 Dolev-Yao 攻击模型中攻击者所具有的能力和对上述安全缺陷的分析，攻击者可能在 Li 匿名认证协议交互过程中发起如下攻击来计算用户真实身份标识在 k-假名集合中的位置. 攻击主要分为两个阶段，第一阶段确定 Step4 处理时间 Δ_t 与用户真实身份标识在集合中位置 r 的确切线性关系；第二阶段通过拦截消息计算出被攻击用户的 Δ_t 值，并利用所得线性关系计算 r 从而得到用户真实身份.

1. 第一阶段. 确定 Δ_t 和 r 的确切线性关系，这里假设攻击者作为合法用户参与认证过程来获取相关数据，攻击过程如图 2 所示，具体操作如下：

(a-1) 攻击者作为合法用户参与认证过程，并发起通信信道的窃听攻击. 首先攻击者拦截自己在 Step3 中发送的 N_2、M_1 和 k-假名集合，然后存储 k-假名集合并记录下拦截到消息的时间记为 T_{start}.

(a-2) 攻击者截获 Step4 中认证服务器发送给自己的消息 M_2，并记录下拦截到该消息的时间 T_{end}.

(a-3) 根据步骤(a-1)和(a-2)可以计算 Step4 的处理时间 $\Delta_t = T_{\text{end}} - T_{\text{start}}$，并且攻击者作为合法用户参与认证过程因此知道自己真实身份标识在 k-假名集合中的位置 r.

(a-4) 为了提高攻击的准确度，攻击者可以同时串联多个其他攻击者来进行多次认证过程，并执行步骤(a-1)、(a-2)、(a-3)即得到多组对应的(Δ_t，r)二元数据. 然后将得到的数据使用最小二乘法进行线性拟合，计算出相关参数的值从而确定 Δ_t 和 r 的准确线性关系方程(2).

2. 第二阶段. 拦截被攻击用户的认证消息并计算出 Δ_t 值，代入 Δ_t 和 r 的线性关系式(2)进行处理，计算出用户真实身份在 k-假名集合中的位置 r. 具体操作如下：

(a-5) 攻击者拦截被攻击用户 Step3 中发送的 N_2、M_1 和 k-假名集合，然后存储 k-假名集合并记录拦截到消息的时间记为 T_{start}.

(a-6) 攻击者截获 Step4 中认证服务器发送给被攻击用户的消息 M_2，记录拦截到该消息的时间 T_{end}.

(a-7) 攻击者计算 Δ_t，并根据第一部分得出的 Δ_t 和 r 的线性关系计算出 r，最终从先前得到的 k-假名集合中确定被攻击用户的真实身份标识.

以上是对攻击过程的分析描述，根据分析可知，

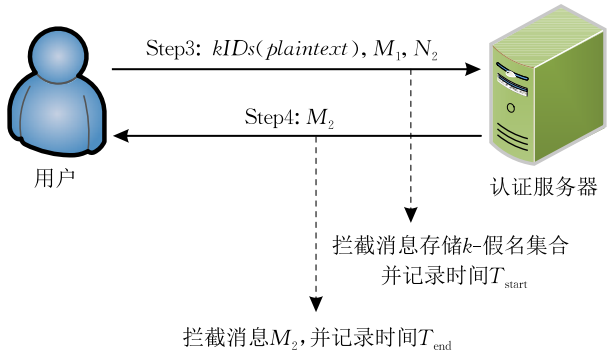

图 2　攻击示意图

Li 匿名认证协议存在上述安全缺陷并可能导致用户真实身份信息的泄露.

3.3　安全缺陷的实验证明

为了证明安全缺陷的真实存在，我们设计了仿真实验来验证我们的分析结果. 按照上述分析，攻击实验主要分为两阶段. 第一阶段建立认证处理时间 Δ_t 与用户真实身份在 k-假名集合中位置 r 的线性关系，需要攻击者作为合法用户参与认证过程；第二阶段是通过 Δ_t 以及得到的线性关系计算被攻击用户的 r 值，从而计算出用户真实身份. 为了提高攻击实验的准确性，采取了如下两项措施：

(1) 模拟了并发用户数量稳定的情况，以减小 T_{delay} 对认证处理时间 Δ_t 的影响. 在攻击实验中，并发认证用户数量设定为 15 人.

(2) 对每一个用户 Δ_t 值的计算都进行了 20 次认证并求平均值. 为了抵抗增强 Dolev-Yao 模型中攻击者发动求交集攻击，用户在每次认证过程中必须使用相同的 k-假名集合以免身份信息泄露.

下面分别对攻击实验的两个阶段进行介绍：

1. 第一阶段. 通过时间关联建立 Δ_t 与 r 的确切的线性关系式.

该阶段攻击试验中通过串联 10 名攻击者并执行攻击步骤(a-1)、(a-2)、(a-3)、(a-4)，即得到如表 2 中所示的 10 组(Δ_t，r)二元数据，其中为了保证实验的随机性和普适性，位置 r 在实验过程随机给出.

表 2　二元组(Δ_t，r)数据表

序号	位置 r	Δ_t/ms
1	3	12.2583
2	5	12.2591
3	2	12.2576
4	7	12.2597
5	6	12.2595
6	8	12.2603
7	11	12.2614
8	9	12.2607
9	19	12.2639
10	23	12.2645

根据所得到的 Δ_t 与 r 的数据，使用最小二乘法对表 1 中的数据进行线性拟合，从而可以求出参数 $1/T_{\text{average}}$ 和 $1/T_{\text{average}}\times(T_{\text{compute}}+T_{\text{delay}})$ 的值即可得到式(3)的准确形式. 通过使用最小二乘法可以使得所有 r 误差的平方和最小，各参数计算公式如下：

$$r=\frac{1}{T_{\text{average}}}\times\Delta_t-\frac{1}{T_{\text{average}}}\times(T_{\text{compute}}+T_{\text{delay}}) \quad (3)$$

$$\frac{1}{T_{\text{average}}}=\frac{1}{C}\times\sum_{k=1}^{n}(\Delta_{t_k}-\bar{\Delta}_t)\times(r_k-\bar{r}) \quad (4)$$

$$-\frac{1}{T_{\text{average}}}\times(T_{\text{compute}}+T_{\text{delay}})=\bar{r}-\frac{1}{T_{\text{average}}}\times\bar{\Delta}_t \quad (5)$$

$$C=\sum_{k=1}^{n}(\Delta_{t_k}-\bar{\Delta}_t)^2 \quad (6)$$

用最小二乘法进行线性拟合计算得到线性关系式(3)的准确形式为

$$r=2686.016\,846\times\Delta_t-32\,923.066\,825 \quad (7)$$

其中参数值为 $1/T_{\text{average}}=2686.016\,846$，$1/T_{\text{average}}\times(T_{\text{compute}}+T_{\text{delay}})=-32\,923.066\,825$.

根据最小二乘法，对表 2 中数据进行线性拟合后各组数据的分布情况如图 3 所示，由图可以更直观地看出 Δ_t 与 r 之间确实存在近似线性的数学关系.

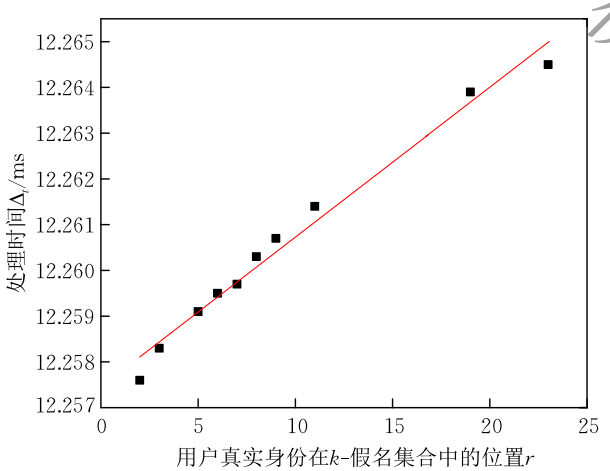

图 3 线性拟合数据图

2. 第二阶段. 计算被攻击用户的 r 值.

首先通过执行攻击步骤(a-5)、(a-6)拦截用户认证 Step3 和 Step4 的消息，拦截到的数据如图 4 所示. 并记录相应的时间 T_{start} 和 T_{end}，然后根据 $\Delta_t=T_{\text{end}}-T_{\text{start}}$ 计算出 Δ_t 的值，代入式(7)进行计算得到计算位置 R 值.

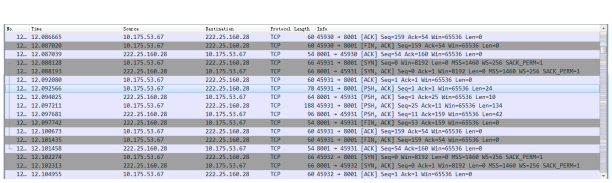

图 4 消息拦截图

然后将计算得到的 R 值与对应的 k-假名集合中用户的真实身份标识的位置 r 进行比较，如果相等则攻击成功，否则攻击失败，并记录 R 值与 r 值的绝对偏差.

如表 3 所示 R 为根据式(7)以及时间 Δ_t 计算得到的位置值，r 为用户真实身份在 k-假名集合中的位置. 偏差值为 R 与 r 的偏差的绝对值. 攻击结果 T 表示攻击成功即 $R=r$，F 表示攻击失败即 $R\neq r$.

表 3 攻击结果统计表

序号	时间 Δ_t/ms	计算值 R	真实值 r	偏差值	结果
1	12.2576	1	3	2	F
2	12.2599	8	8	0	T
3	12.2661	26	24	2	F
4	12.2582	2	2	0	T
5	12.2597	7	6	1	F
6	12.2584	3	3	0	T
7	12.2619	13	12	1	F
8	12.2672	29	29	0	T
9	12.2589	5	5	0	T
10	12.2577	1	3	2	F
11	12.2657	25	25	0	T
12	12.2617	13	13	0	T
13	12.2664	27	28	1	F
14	12.2598	7	7	0	T
15	12.2685	33	33	0	T

本次实验共对 200 名用户进行攻击(每个用户取 20 次认证过程 Δ_t 的平均值来计算 R 值)，成功 121 次，失败 79 次，成功率为 60.5%，因此证明该协议具有较严重的安全缺陷，表 3 为 200 次攻击中实验的 15 条实验结果. 根据表 3 可以看到即使攻击失败，R 与 r 的绝对偏差也不会超过 2，即 r 的值以较大概率落在以 R 值为中心大小为 5 的集合内. 因此即使攻击失败不能得到准确的用户身份标识但却可以将其限定在一个较小的范围内，这降低了协议的匿名性. 原来攻击者猜中用户真实身份的概率为 $1/k$，匿名成功率为 $P=(k-1)/k$. 通过本次攻击用户身份被猜中的概率为 1/5，匿名成功率为 $P=4/5$，匿名成功率下降为 $\Delta_p=(k-5)/5\times k$. 因此通过上述实验，可以确定 Li 所提匿名认证方案确实存在安全缺陷，从而导致匿名成功率大幅度下降，甚至是直接暴露用户的身份信息.

4 改进 Li 匿名认证协议

上述攻击过程说明 Li 匿名认证协议不能有效地抵抗基于认证时间的关联分析. 分析可以得出引发该安全缺陷的因素主要有：(1) 用户在向服务器发送 k-假名集合进行认证时，由于集合中包含用户

的真实身份标识，因此增加了用户身份信息泄露的风险；(2) 认证服务器对用户请求的处理时间与用户真实身份在 k-假名中的位置存在近似的线性关系，通过对认证时间的拟合即可得到二者的准确关系，从而根据处理时间确定用户的真实身份；(3) 为了增强抵抗 Dolev-Yao 模型中攻击者发动求交集攻击，用户在每次认证时必须使用相同的 k-假名集合，因此攻击者能够观察多次认证过程，从而利用求平均值的方法提高攻击的准确性.

针对上述问题，我们引入用户分组机制，用户在认证之前需要预先在认证服务器进行分组. 并向认证服务器索取自己所在组的组标识 GID、共享密钥 Key 等. 用户在后续的认证过程中，即可使用 GID 替代自己的真实身份来进行认证. 在保证了用户真实身份安全的同时分组机制不会降低协议的性能，认证服务器只需按照分组进行遍历即可完成认证过程，保证了协议的高效性.

4.1　协议交互过程

改进协议的提出过程中主要采用如下技术：

(1) 采用共享密钥技术，使所设计协议能更好的适用于能力受限的移动设备.

(2) 采用假名技术，使用组标识 GID 替代用户真实身份来实现匿名性.

(3) 采用用户分组机制，改进了 Li 所提协议的安全缺陷，增强了安全性的同时提高性能.

在该协议中，需要认证服务器预先给用户进行分组并存储分组信息，然后将用户所在组的组标识 GID 分发给用户. 用户在认证过程中将 GID 连同用共享密钥加密的认证消息发送到认证服务器. 认证服务器至多遍历 GID 对应组中所有用户的密钥. 通过验证其对应认证消息即可验证用户的身份. 在该协议中用户使用 GID 来进行认证，即使被攻击者拦截也无法关联分析出用户真实身份. 而且认证服务器至多遍历 GID 对应组中所有用户的密钥即可完成认证. 避免了认证服务器资源浪费. 且保证了所设计协议的高效性. 其中本文所设计协议中符号含义如表 4 所示.

表 4　协议符号对照表

符号	意义
$SNonce$	认证服务器生成的随机数
$CNonce$	用户生成的随机数
C	用户真实身份标识
GID	用户所在组的组标识
$HMAC$	哈希函数
Key	用户和认证服务器的共享密钥
SK	用户和认证服务器的会话密钥
$PRNG()$	随机数生成器

协议交互过程如下，其中交互图如图 5 所示.

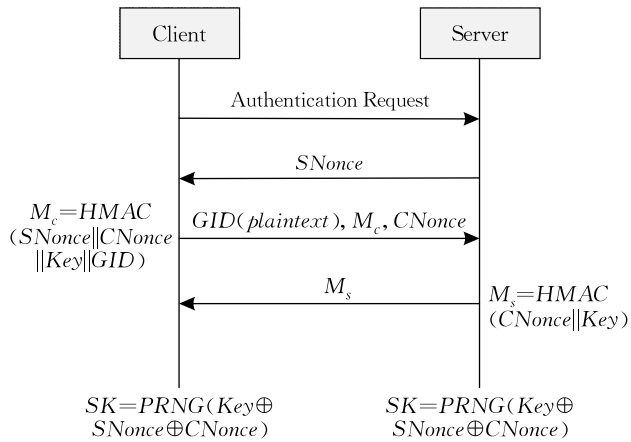

图 5　本文所设计方案交互图

Step1. 用户→认证服务器

首先由用户请求开始匿名认证，消息内容是 32 比特长的字符串信息.

Step2. 认证服务器→用户

认证服务器收到用户发送请求消息. 产生随机数 $SNonce$ 并发往用户，长度为 64 bit. 服务器为了提高处理速度，可以预先生成一些随机数存储在本地，当有用户请求认证时就选取一个随机数发送出去.

Step3. 用户→认证服务器

随机数 $SNonce$ 到达用户端后，用户生成随机数 $CNonce$，然后计算消息 M_c.

$$M_c = HMAC(SNonce \| CNonce \| Key \| GID) \quad (8)$$

其中“‖”是字符串连接符. 然后用户发送 GID，M_c，$CNonce$ 到服务器端进行认证. 此步骤发送的消息内不包含用户真实身份. 主要是为了防止攻击者通过结合认证时间等信息计算出用户的真实身份信息. 其中 $SNonce$ 和 $CNonce$ 实现挑战-应答. 确保认证过程传输消息的新鲜性，抵抗消息重放.

Step4. 认证服务器→用户

匿名认证服务器接收 Step3 中用户发送的消息. 随后遍历 GID 对应组内的用户进行认证. 首先认证服务器按照组内(组标识为 GID 的组)用户身份标识的先后顺序计算对应 M_c'，如式(9)，并验证 M_c' 与 M_c 是否相等. 如果相等则停止查询组内其他用户，服务器对用户认证通过. 如果遍历完组内所有用户都不能使 M_c' 与 M_c 相同则认证失败. 若服务器认证成功，则服务器计算消息 M_s，然后将 M_s 发送给用户，M_s 的计算公式如式(10)所示.

$$M_c' = HMAC(SNonce \| CNonce \| Key' \| GID) \quad (9)$$

$$M_s = HMAC(CNonce \| Key) \quad (10)$$

用户接收到消息 M_s，按照相同的方式计算 M_s，然后与收到的消息比对. 消息相等则客户端认证服务器成功，否则即是认证失败. 在用户完成对认证服务器的认证后，双方生成会话密钥. 认证服务器端以及用户端的会话密钥 SK 的计算公式如式(11)所示.

$$SK=PRNG(Key\oplus SNonce\oplus CNonce) \quad (11)$$

在认证消息 M_s 和 M_c 的计算过程中所使用的 Hash 算法选取了资源消耗较小的 SHA-1 算法，有助于提高协议的认证性能.

4.2 安全性证明

4.2.1 安全模型

为了保证本文所设计协议的安全性，我们使用形式化证明说明本文协议的安全性. 我们假设攻击者具有 Dolev-Yao 攻击模型中攻击者所具有的全部能力. 因此攻击者可以通过发起如下 Oracle 查询来进行攻击，其中 C 代表用户，S 代表认证服务器：

(1) $Listen(C,S)$. 模拟攻击者进行的被动攻击，可以窃听 C 和 S 之间交换的所有消息.

(2) $Send(C,S)$. 模拟攻击者可以扮演合法 C 向 S 发送消息，并接受 S 的应答消息；

(3) $Send(S,C)$. 模拟攻击者可以扮演合法 S 向 C 发送消息，并接受 C 的应答消息；

(4) $Union(C)$. 模拟攻击者可以联合 C 的能力，使 C 泄露自己的秘密信息 Key；

(5) $Test(C)$. 模拟攻击者从 $Union(C)$ 中获取的 Key，用来度量 C 中 Key 的语义安全性，通过一个比特 $b\leftarrow\{0,1\}$，如果 $b=1$，返回用户的真实 Key，如果 $b=0$，返回一个与真实 Key 同等长度的随机串.

攻击者在本文所提协议中的目的是对认证用户的身份进行区别和跟踪，以获取用户的隐私信息. 基于上述信息将本文中攻击者 A 设定为概率多项式时间算法，在多项式时间 $t(n)$ 内，其中 n 为时间参数可忽略，攻击者可以尝试最多 $q(n)$ 次攻击，并将该攻击定义为恶意攻击者 A 与用户和认证服务器之间进行的游戏，攻击分为两个阶段：

(1) 训练阶段

攻击者 A 可以发送任意的 $Listen$，$Send$ 和 $Union$ 查询. 并在 $t(n)$ 时间内收集最多 $q(n)$ 条知识，其中根据认证协议的过程，这些知识的组成包括 $I(n)=\{SNonce,CNonce,GID,M_c,M_s\}^{q(n)}$，并将 $I(n)$ 作为下一阶段的帮助信息.

考虑本文攻击者 A 的目的是为了识别用户 C，由于本文中用户的真实身份均被组标识 GID 所替代，那么攻击者仅能够通过跟踪 GID 来识别用户 C，因此假设攻击者的攻击目标是 GID_c，并且训练阶段攻击者 A 在 $t(n)$ 内的 $q(n)$ 次查询来自于 GID_c 的概率为 $\alpha(0\leq\alpha\leq1)$，那么攻击者 A 在训练阶段知识 $I(n)$ 中能够得到 GID_c 有关的信息数目为 $q_c(n)=\alpha\times q(n)$.

(2) 挑战阶段

在挑战阶段，恶意攻击者 A 在游戏的某个时刻，参与用户和认证服务器的会话过程. 攻击者 A 在协议的执行过程中发起 $Test(C)$ 查询，由于是多项式时间的攻击算法，多项式时间 $t(n)$ 内，A 可以进行 $q(n)$ 次攻击实验，设每次攻击实验 A 获胜的优势为 Adv_A^i，则最终赢得游戏的概率优势为 $Adv(A)=\sum_{i=1}^{q(n)}Adv^i(A)$.

4.2.2 安全定义

定义 1(语义安全). 在 $I(n)$ 的基础上对 b 进行猜测，输出比特 b'. 如果 $b'=b$，则输出结果为 1，否则输出 0，因此攻击者在实验 $EXP_A(t,q)$ 中获胜的概率优势为：$Adv(A)=\left|pr(EXP_A(t,q)=1)-\frac{1}{2}\right|$，其中如果在该实验中，攻击者获胜的优势 Adv_A 是可忽略的，则说明本文所设计协议是语义安全的.

定义 2(匿名性). 对于用户 C_i 和 C_j，分别表示 GID 分组中的两个用户，同时用 $M(C_i,S)$ 和 $M(C_j,S)$ 表示两个用户执行协议的消息抄本，如果 $Dist[M(C_i,S)]=Dist[M(C_j,S)]$，其中 $Dist[M(C,S)]$ 表示 $M(C,S)$ 的概率分布，说明本协议具有匿名性.

4.2.3 安全性证明

定理 1. 设攻击者 A 是概率多项式时间算法，在多项式时间 $t(n)$ 内，攻击者最多进行 $q(n)$ 次随机预言 Oracle 查询. 则攻击者获胜的概率优势 $Adv(A)\leq q(n)/2^{|key|}+neg(l)$ 是可忽略的，其中 $neg(l)$ 表示关于安全参数 l 的一个可忽略函数，因此我们的协议是安全的.

证明. 我们采用混合实验的方法来证明协议的语义安全性，通过一系列的攻击实验对模拟规则进行改变，直到攻击者获胜的优势为可忽略的函数为止. 我们用 $Adv^i(A,EXP_j)$ 表示在第 i 次攻击第 j 个混合实验中的优势.

实验 EXP_0：此实验模拟在随机预言模型下的真实协议运行，在实验中攻击者 A 可以进行 Oracle 查询，则有 $Adv^i(A)=Adv^i(A,EXP_0)$.

实验 EXP_1：在本次实验中，我们通过建立信息列表来模拟随机预言函数，分别将 Step3、Step4 中的 $HMAC$ 和最后 $PRNG$ 函数记为 H_1, H_2, P，同时我们模拟私有的随机预言函数 H_1', H_2', P'. 随机预言函数的模拟规则如下：

(1) H_i 查询列表 T_{H_i} ($i=1,2$). 对于训练阶段每次的随机预言查询 $H_i(m)$，将记录 (i,m,r) 添加到 T_{H_i}，同时对于挑战阶段的新的随机预言查询 $H_i(m)$，如果列表中存在记录 (i,m,r)，则返回 r；否则随机选择 $r\in\{0,1\}^l$，将 r 返回给攻击者，并将记录 (i,m,r) 添加到 T_{H_i}.

(2) P 查询列表 T_P. 对于训练阶段每次的随机预言查询 $P(m)$，将记录 (m,r) 添加到 T_P，同时对于挑战阶段的新的随机预言查询 $P(m)$，如果列表中存在记录 (m,r)，则返回 r；否则随机选择 $r\in\{0,1\}^l$，将 r 返回给攻击者，并将记录 (m,r) 添加到 T_P.

(3) H_i' 查询列表 T_{H_i} ($i=1,2$). 对于训练阶段每次的随机预言查询 $H_i'(m)$，将记录 (i,m,r) 添加到 $T_{H_i'}$，同时对于挑战阶段的新的随机预言查询 $H_i'(m)$，如果列表中存在记录 (i,m,r)，则返回 r；否则随机选择 $r\in\{0,1\}^l$，将 r 返回给攻击者，并将记录 (i,m,r) 添加到 $T_{H_i'}$.

(4) P' 查询列表 $T_{P'}$. 对于训练阶段每次的随机预言查询 $P'(m)$，将记录 (m,r) 添加到 $T_{P'}$，同时对于挑战阶段的新的随机预言查询 $P'(m)$，如果列表中存在记录 (m,r)，则返回 r；否则随机选择 $r\in\{0,1\}^l$，将 r 返回给攻击者，并将记录 (m,r) 添加到 $T_{P'}$.

除了模拟随机预言函数外，我们还根据协议描述模拟所有的 $Listen$，$Send$ 和 $Test$ 查询，因此由模拟规则可知 $Adv^i(A,EXP_0)=Adv^i(A,EXP_1)$.

实验 EXP_2：在该阶段实验中，我们排除一些发生碰撞的会话，如果会话中的消息发生碰撞或者随机预言函数的输出发生碰撞，则取消会话的运行，则由生日攻击原理可知，实验 EXP_2 和 EXP_1 不可区分. 因此 $|Adv^i(A,EXP_2)-Adv^i(A,EXP_1)|\leq neg(l)$.

实验 EXP_3：在该阶段实验中，我们修改对 $Listen$ 查询的模拟，在被动会话中将随机预言函数 H_1, H_2, P 分别替换为 EXP_1 中定义的 H_1', H_2', P'，并且随机选择随机预言函数中的输入 Key. 如果攻击者想要区分实验 EXP_3 与 EXP_2，由于其它的值都是公开的，仅共享密钥 Key 和会话密钥 SK 秘密的，因此攻击者如果想获取 SK，必须从以下两个角度：

(1) 攻击者必须可以正确地恢复共享密钥 Key 从而进一步恢复会话密钥 SK，但是共享密钥 Key 与协议中其它值并无任何关系，从训练阶段积累的知识中也无法推导出 Key，因此攻击者在区分被动会话中共享密钥时没有任何优势，Key 值完全随机产生，假设此实验为 EXP_3^1，因此实验 EXP_3^1 与 EXP_2 不可区分，即 $|Adv^i(A,EXP_3^1)-Adv^i(A,EXP_2)|\leq neg(l)$.

(2) 攻击者必须通过训练阶段积累的知识对 SK 进行直接恢复，由于用户认证时每次使用相同的共享密钥 Key，因此如果被动攻击实验中所使用的 $SNonce$ 和 $CNonce$ 在训练阶段出现，那么攻击者则可以通过随机语言函数 P 直接恢复 SK. 假设此实验为 EXP_3^2，现在证明 EXP_3^2 与 EXP_2 是不可区分的.

首先给出攻击者 A 在训练过程中具备的知识对认证过程的影响. 攻击者 A 的目标 GID_{C_m} 来自某特定用户 C_m 的概率为 $pr(C_m)=1/|GID_{C_m}|$，$|GID_{C_m}|$ 为分组大小；挑战阶段用户 C_m 认证过程中使用的 $SNonce$ 在训练阶段出现过的概率为：$pr(SNonce)\leq pr(C_m)\times q_{C_m}(n)/2^{|SNonce|}$；用户 C_m 认证过程中使用的 $CNonce$ 在训练阶段出现过的概率为：$pr(CNonce)\leq pr(C_m)\times q_{C_m}(n)/2^{|CNonce|}$. 其中 $|SNonce|$ 和 $|CNonce|$ 分别表示 $SNonce$ 和 $CNonce$ 的长度. 因此攻击者通过 $SNonce$ 和 $CNonce$ 直接恢复 SK 的概率为

$$\begin{aligned}pr(SK)&=pr(SNonce)\times pr(CNonce)\times pr(C)\\&\leq pr(C)^3\times q_c(n)^2/2^{|SNonce|+|CNonce|}\\&\leq\frac{q_c(n)^2}{|GID_c|^3\times 2^{|SNonce|+|CNonce|}}.\end{aligned}$$

在实验过程中 $CNonce$，$SNonce$ 选择为 64 bit，适当用户分组大小的情况下，直接恢复 SK 的概率 $pr(SK)$ 是忽略不计的，因此 EXP_3^2 与 EXP_2 不可区分，则 $|Adv^i(A,EXP_3^2)-Adv^i(A,EXP_2)|\leq neg(l)$.

综上(1)、(2)所述，实验 EXP_3 与 EXP_2 是不可区分的，即 $|Adv^i(A,EXP_3)-Adv^i(A,EXP_2)|\leq neg(l)$.

实验 EXP_4：在此阶段实验中，我们最后处理攻击者通过 $Send$ 查询进行的主动攻击，一方面我们让攻击者接受用户的 $SNonce$，并返回 Step3 阶段消息 $(GID,M_c,CNonce)$，其中 M_c 使用 H_1 生成，共享密钥 Key 随机选择，此时服务器拒绝接受并终止协

议的进行.另一方面攻击者作为服务器接受 Step3 阶段消息,并返回 Step4 阶段消息 M_s,其中 M_s 使用 H_2 生成,Key 随机选择,此时用户拒绝接受并终止协议进行.显然实验 EXP_4 与实验 EXP_3 是不可区分的,除非攻击者能够正确计算出 Key 值,从而正确生成 M_s 或 M_c,我们将该事件定义为 $getKey$,则 $|Adv^i(A,EXP_4)-Adv^i(A,EXP_3)|\leq pr^i(getKey)$.

上述实验中,由于训练阶段没有获取任何关于 Key 的直接信息,并且 Key 值是随机选择的,在实验 EXP_2 中也已经排除了随机预言函数的碰撞,因此显然有 $pr^i(getKey)\leq 1/2^{|Key|}$,其中 $|Key|$ 表示密钥的长度.

由于在多项式时间 $t(n)$ 内,攻击者可以进行 $q(n)$ 次查询,则有 $pr(getKey)\leq q(n)/2^{|Key|}$,进一步有 $Adv(A)\leq \alpha\times q_c(n)/2^{|key|}+neg(l)$,因此定理 1 得证. 证毕.

定理 2. 本文所设计协议实现了用户身份的匿名性,并且恶意攻击者至多以 $1/|GID|$ 的概率破坏协议的匿名性.

证明. 对于用户 C_i 和 C_j 发送的所有信息中,$SNonce$ 和 $CNonce$ 随机选择,并且 GID 是相同的,因此消息是均匀分布的;并且 Key 对于同一用户固定且秘密,通过 M_s 不会泄露身份信息,因此对两个用户 C_i 和 C_j 有 $Dist[M(C_i,S)]=Dist[M(C_j,S)]$,因此协议实现了身份的匿名性.

对于用户 C,攻击者仅通过协议真实执行过程中截获的消息 $\{SNonce,CNonce,GID_c,M_c,M_s\}$ 来识别攻击者身份,在此过程中 $SNonce$ 和 $CNonce$ 都是随机的,因此攻击者通过该信息不可能有任何优势区分用户身份.根据训练阶段获取的知识 $I(n)$,攻击者可根据 GID_c 以概率 α 获取和 C 同一分组所有信息 $\{M_c^{GID_c},M_s^{GID_c}\}^{\alpha\times q(n)}$,但是对于 M_c,M_s 的生成过程都需要用户和认证服务器的共享密钥 Key 作为输入,而 $M_c^{GID_c}$,$M_s^{GID_c}$ 对攻击者并没有任何可用的帮助,因此攻击者通过该信息也不可能有任何优势以区分用户身份.此时攻击者仅能通过 GID_c 对用户进行随机猜测,假设攻击者根据知识掌握用户使用 GID_c 的信息,仍然至多以 $1/|GID_c|$ 的概率猜测出正确的用户,因此攻击者破坏协议中用户匿名性的概率至多为 $1/|GID_c|$,因此定理 2 得证.证毕.

5 安全性分析与仿真实验

为了说明本文所设计方案的安全性和高效性,我们进行了安全性分析和实验仿真.安全性方面主要从协议的抗攻击能力等方面对协议安全性进行分析,从而将所设计协议与现有的匿名认证协议的安全性进行对比,证明所设计协议更加安全可靠.在实验仿真方面则通过搭建测试环境来进行性能分析,其中使用的设备有,无线 AP:TP-Link 无线路由器,用户端设备:联想笔记本电脑,Windows7,i5-4590 CPU 3.30 GHz、4 GB 内存;认证服务器:惠普 Z620 工作站,Windows7,E5-1603 24 核 CPU,96 GB 内存.实验拓扑如图 6 所示.

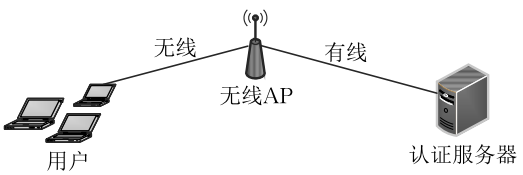

图 6 实验拓扑图

5.1 安全性分析

5.1.1 安全性分析

(1)抵抗时间关联攻击

Li 所设计方案中在认证过程中虽然使用了 k-假名集合,但是集合中依然含有用户的真实 ID,并且该假名集合在认证过程中直接以明文的方式发送给认证服务器,该假名集合容易被攻击者截获,通过实验可知,攻击者通过认证时间和用户真实 ID 在假名集合中的位置间的线性关系并结合假名集合来确定用户真实身份,从而泄露用户隐私信息.

而本文方案将用户的身份信息泛化为组标识 GID,用户在认证过程中不再需要发送有关个人的 ID 信息,只需要发送 GID 到认证服务器,从而攻击者不能获取有关用户身份的任何信息,当攻击者仅知道 GID 的情况下,无法将发起认证请求的用户与对应分组内某个特定的用户相关联,同时由于同一分组内具有多个用户,使攻击者根据特定的假名信息对用户进行追踪的难度加大,进一步保证了用户的不可追踪性,因此可以保证用户的隐私安全.

(2)抵抗重放攻击

本文所设计匿名身份认证协议的交互过程引入了挑战-应答机制.以保证认证过程中传输消息的新鲜性.认证服务器收到用户发来的认证请求后发送一个随机数 $SNonce$ 给用户.用户使用共享密钥和 $SNonce$ 生成一条认证消息 M_c,并将 M_c 和自己生成的随机数 $CNonce$ 发往认证服务器.认证服务器用共享密钥计算 M_c' 与 M_c 比较即可知道该消息是不是具有新鲜性.因为只用相同的随机数计算的消息才

会相同，而认证服务器使用的是最新产生的随机数. 同理用户也可以通过自己产生的随机数 *CNonce* 验证服务器发来的消息是否具有新鲜性，因此可以抵抗重放攻击.

(3) 抵抗伪装攻击

本文所设计匿名身份认证协议通过共享密钥加密来抵抗伪装攻击. 用户在认证之前需要先到认证服务器注册. 然后与认证服务器协商共享密钥 Key. 交互过程中用户发送认证消息 M_c 到认证服务器. 认证服务器使用对应共享密钥生成类似的消息 M_c' 与 M_c 进行比较. 因为密钥只有用户和认证服务器拥有，而攻击者没有，因此若 M_c' 与 M_c 相等则该消息是来自用户的，且其中引入挑战-应答避免了该消息是攻击者重放的，因此能抵抗攻击者伪装合法用户. 同理用户通过消息 M_s 来识别认证服务器是否被伪装. 基于上述分析可知所设计匿名认证协议能够抵抗攻击者伪装合法和认证服务器的攻击.

(4) 抵抗消息修改

本文所设计协议通过使用 Hash 算法来抵抗消息修改攻击. 认证过程中认证服务器通过计算消息 $HMAC(SNonce \| CNonce \| C \| Key \| GID)$ 来认证用户身份. 若是攻击者修改消息内容，根据散列算法的抗强碰撞性可知认证服务器同攻击者计算的消息摘要必然不同. 因此认证服务器可以检测出来自用户的消息是否被修改过. 同理用户也可以通过认证消息 $HMAC(CNonce \| Key)$ 检测来自认证服务器消息有没有被修改过. 所以论文所设计协议可以有效抵抗消息修改.

(5) 抵抗别名去同步攻击

别名去同步攻击主要指攻击者破坏用户与服务器二者的别名之同步. 以达到用户不能正常认证的目的. 该攻击主要针对需要进行别名或假名更换的协议，本文所设计协议不涉及这方面的问题，用户的分组 GID 是固定不变的不存在更换问题，因此不会受这种类型的攻击.

(6) 双向认证性

本文所设计身份认证方案中，认证服务器通过遍历 GID 组内所有成员并计算消息 M_c'，与 M_c 进行比较来认证对应成员的身份. 而用户通过计算 M_s 与认证服务器发送的消息 M_s 进行比较来认证认证服务器的可信性. 交互完成后二者都验证了对方身份信息的合法与否.

(7) 前向保密性和后向保密性

本文所设计认证协议中，用户与认证服务器通过两轮交互彼此验证对方身份，立即使用预先协商好的算法计算会话密钥 SK 用于加密后续通信过程传输的消息. 其中计算 SK 使用了随机数 $SNonce$ 和 $CNonce$ 即 $SK=PRNG(Key \oplus SNonce \oplus CNonce)$ 而 $SNonce$ 和 $CNonce$ 在每次认证过程中都是随机生成的，与前一次或后一次生成的随机数没有关联. 因此攻击者不能通过截获会话密钥 SK 来计算前一次或后一次的会话密钥，保证了协议的前向保密性和后向保密性.

5.1.2　安全性对比

本节主要将论文提出的无线网络中基于共享密钥的轻量级匿名认证协议和参考文献[7-8,23]中的匿名认证方案进行安全性对比，如表 5 所示.

表 5　安全性对比

安全属性	文献[7]	文献[8]	文献[23]	本文方案
抗重放攻击	Y	Y	Y	Y
抗伪装攻击	Y	Y	Y	Y
匿名性	Y	Y	Y	Y
双向认证	Y	Y	Y	Y
前向和后向保密	Y	Y	Y	Y
抗关联性分析	Y	N	Y	Y
无需要额外设备	Y	Y	N	Y
抗别名去同步攻击	N	Y	N	Y

表 5 将本文所设计匿名认证协议和现有匿名认证协议进行安全性比较，结果表明本文所设计无线网络中基于共享密钥的匿名身份认证协议安全性更高. 此外，改进的协议在不能获取 MAC 地址的移动通信网中能够抵抗攻击者根据 MAC 地址等独有信息对用户进行非法跟踪. 而在 WIFI 网络等能获取 MAC 地址的环境下，可以通过引入 MAC 地址随机化等技术抵抗攻击者对用户进行非法跟踪.

5.2　性能分析

通过进行仿真实验并对实验数据进行量化处理，对所设计匿名身份认证协议的认证时间、计算量、通信量和所需存储空间大小等方面进行分析.

(1) 认证时间

认证时间是评估一个身份认证协议性能的重要标准. 通常来讲认证时间不应超出用户的忍受限度，过长的认证时间将会降低对用户的服务质量并降低协议的使用性. 本协议通过在服务器端对用户分组来进行认证，因此分组的大小是影响认证时间的重要因素. 仿真实验主要针对两种情况来测试所设计协议的认证时间：第一种主要测试认证时间随分组的大小 n 的变化情况；第二种主要测试认证时间随同时发起身份认证的用户数量的变化情况.

针对上述第一种测试情况实验中为了得到认证时间的上限,因此考虑了最坏的情况,即每个参与测试用户的身份标识在分组的最后一个位置.实验中分别测试分组大小 n 为 1,10,20,30,40 时 20 次认证每一次的认证时间(该测试中同一时间只有一个用户进行认证),实验结果如图 7 所示.可以看到随分组大小 n 的变化认证时间有所增长,但是对认证时间影响不大.

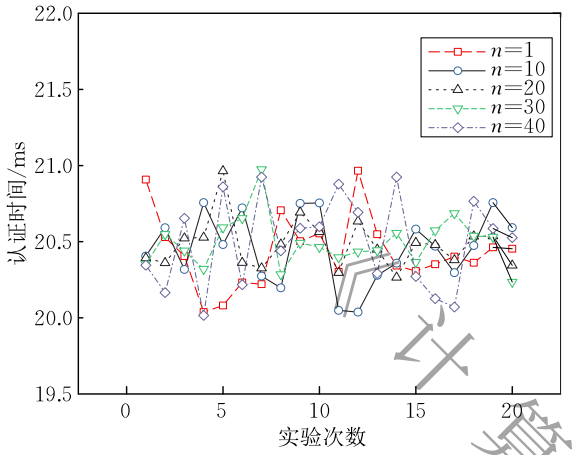

图 7 不同分组下的认证时间

用户 $N=20$ 次认证的平均认证时间随分组大小的变化情况如图 8 所示,计算可知每当分组大小 n 增加时平均认证时间也有所增加,但是对认证时间影响很小,因此所设计协议认证时间相对稳定且受分组大小影响较小,并且由于分本文方案引入了分组机制,即使当认证服务器中注册用户数量增加时,由于认证过程仅需要遍历分组内用户即可完成,因此,认证时间仅与分组大小有关,不会随注册用户数量增多而增加,认证时间较稳定,使得本文方案可以进一步抵抗时间关联攻击并具有较好的性能.

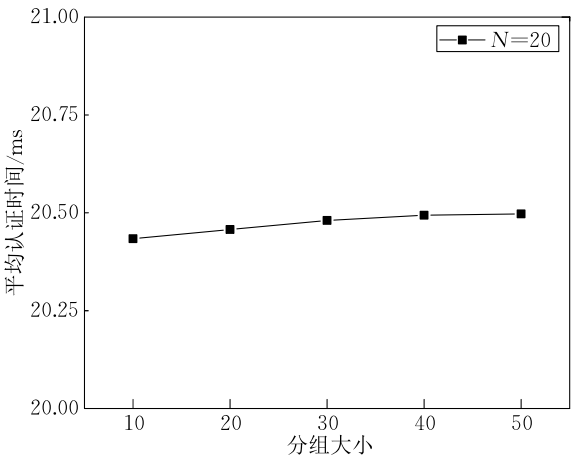

图 8 不同分组下的平均认证时间

本文所设计匿名认证协议采用了假名技术,比通过身份信息加密来实现匿名性的协议具有更高的性能.例如在 RFID 身份认证过程中,认证服务器需要遍历数据库中的所有标签来验证用户的身份.当标签数量很多时认证时间会很长,导致协议的性能低且可用性差.本文所设计匿名认证协议中认证服务器最多遍历整个组中的用户即可完成认证,通过限制分组的大小就可以保证协议的高效性.

针对第二种测试情况,多个用户同时认证时的认证时间变化,本次实验预先指定分组大小 $n=20$,并通过改变同时发起认证的用户数量来分析认证时间的变化.如图 9 所示,认证时间随同时发起认证的用户数量的变化情况.可以看到,认证时间与并发用户数呈近似线性关系.认证时间主要包括通信时间和处理时间,其中通信时间指的是消息传输所需要的时间,处理时间指的是计算认证消息并进行验证所需要的时间.在上一个测试中已经发现处理时间对认证时间的影响比较小,因此即使多个用户同时进行认证,处理时间也不会对认证时间产生太大的影响.

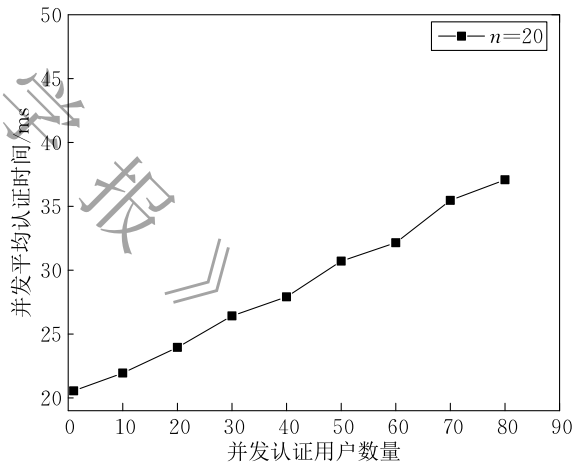

图 9 用户并发认证时间

多用户同时认证主要影响通信时间,因为在无线网络中当有多个用户同时进行认证时所有用户要通过 CSMA/CA 来竞争无线信道,因此会产生一定的时延.当同时进行认证的用户数量由 m 增加到 $m+1$ 时,原来 m 个用户的认证时间会因为新加入的一个用户竞争无线信道而有所增加.设新加入一个用户后 $m+1$ 个用户的平均信道竞争时延增加为 $delta$,那么若 m 个用户同时认证时平均认证时间为 t,那么 $m+1$ 个用户同时认证时平均时间为 $t+delta$.根据文献[21-22]可知在无线网络中由于分布式协

调功能 DCF 的作用，当所有用户要发送的数据包大小相同时用户接入网络所需要的平均时延与同时进行认证的用户数量呈线性关系. 因此本文协议所达到的并发效果符合正常的水平.

本文方案的分组大小 n 直接影响了所设计方案的安全性和性能. 其中分组大小 n 的确定需要考虑两个方面的因素：① 用户的隐私需求，当用户具有较高的隐私需求时，应使分组大小 n 较大，易知攻击者不能以超过 $1/n$ 的概率分辨出真实用户，当 n 越大时，用户隐私泄露的概率越小；② 用户的服务质量需求，一般来说用户分组 n 越大，则认证时间越长，服务质量将会降低，同时认证服务器的性能对服务质量也有一定的影响. 因此在实际环境中，隐私需求和服务质量相互制约，我们需要综合以上两个方面因素来最终确定合适的分组大小. 在上述实验中为了仿真实际环境中不同的隐私需求，并且考虑实验设备的性能将分组大小设置在了 1 到 50 之间的取值，从而验证本文方案的高效性.

(2) 通信量

通信量指的是认证过程中用户和认证服务器之间传输消息的数据量之和. 本文所设计匿名认证协议交互过程分为四个交互步骤，传输消息如下：

Step1. 传输的消息为认证请求，长度为 32 bit.

Step2. 传输的消息为 $SNonce$，长度为 64 bit.

Step3. 传输的消息为 $CNonce$ 和 M_c，$CNonce$ 是长度为 64 bit，M_c 是使用 SHA-1 计算的认证消息长度为 160 bit，因此总消息长度为 224 bit.

Step4. 传输的消息 M_s，M_s 是使用 SHA-1 计算的认证消息，长度为 160 bit.

协议交互过程传输的通信量如表 6 所示，总的通信量为 480 bit.

表 6　协议通信量

协议步骤	通信量/bit
Step1	32
Step2	64
Step3	224
Step4	160
通信量总和	480

(3) 所需存储空间

用户端需要存储的信息包括：用户真实身份标识 C 长度为 32 bit、共享密钥 Key 长度为 128 bit、组标识 GID 长度为 32 bit、会话密钥 SK 长度为 128 bit，则用户端需存储空间总量为 320 bit.

认证服务器端为每个用户存储的信息包括：用户真实身份标识 C 长度为 32 bit、共享密钥 Key 长度为 128 bit、组标识 GID 长度为 32 bit、会话密钥 SK 长度为 128 bit，则每个用户端所需存储空间总量为 320 bit，假设认证服务器上注册的用户数量为 n，则认证服务器所需总存储空间为 $n\times320$ bit. 此外相比于现有常量认证时间方案，本文在存储开销方面具有绝对的优势. 其中表 7 中 C 为文献[17-18，20]方案中的计数值.

表 7　存储开销对比

文献	文献[17]	文献[18]	文献[20]	本文方案
存储空间	$O(C\times n)$	$O(C\times n)$	$O(C\times n)$	$O(n)$

(4) 计算量

计算量指认证过程中计算认证消息所需的计算量，分为用户端计算量和认证服务器端计算量.

1. 用户在认证过程中需要进行的计算主要包括如下内容：

① 2 次 Hash 运算. 计算消息 M_c，计算消息 M_s 公式为 M_s.

② 2 次异或运算. 计算共享密钥 SK 其中包含 2 次异或运算.

2. 认证服务器在认证过程中需要进行的计算主要包括如下内容：

①$(r+1)$次 Hash 运算. r 为用户真实身份在组中的位置，认证服务器通过组标识 GID 遍历组内用户计算 M_c 来认证用户身份共需要 r 次 Hash 运算. 计算消息 M_s 需要一次 Hash 运算.

② 2 次异或运算. 计算共享密钥 SK 其中包含 2 次异或运算.

根据上述计算量统计，并与现有的基于共享密钥的研究进行对比，其对比结果如表 8 所示. 可以看出本文所设计出方案在保证上述优点的同时并不增加计算量. 并且通过合理的设置认证服务器端分组大小，可以保证用户端和认证服务器端的性能.

表 8　计算量对比

运算方式	文献[7]	文献[8]	文献[15]	本文方案
Hash 运算	$n+10$	$r+3$	0	$r+3$
异或运算	13	4	$2n+9$	4
CRC 运算	0	0	$n+3$	0

上述性能测试与评估结果表明所设计匿名认证协议具有认证时间短、计算量小、所需存储空间和通信量少的特点，因此该方案具有轻量级的特点，能更好的适用于无线网络环境.

6 结 论

本文首先分析了 Li 匿名认证协议的安全缺陷，指出认证时间和用户在 k-假名集合中的位置存在线性关系，导致该协议无法抵抗针对认证时间关联分析. 进一步进行实验仿真验证，结果证明攻击者能够根据认证时间以较大概率获取用户的真实身份标识. 通过引入用户分组机制，提出了一种基于共享密钥的轻量级匿名认证方案. 通过安全分析和实验仿真验证，所设计方案能够抵抗时间关联等攻击，并具有计算量开销、通信量开销和存储开销小等特点，表明所设计方案更加安全，并且具备轻量级的特点，更加高效.

参 考 文 献

[1] Ren K, Lou W, Kim K, et al. A novel privacy preserving authentication and access control scheme for pervasive computing environments. IEEE Transactions on Vehicular Technology, 2006, 55(4): 1373-1384

[2] Wan Z, Ren K, Preneel B. A secure privacy-preserving roaming protocol based on hierarchical identity-based encryption for mobile networks//Proceedings of the 1st ACM Conference on Wireless Network Security. Alexandria, USA, 2008: 62-67

[3] Huang D, Misra S, Verma M, et al. PACP: An efficient pseudonymous authentication-based conditional privacy protocol for VANETs. IEEE Transactions on Intelligent Transportation Systems, 2011, 12(3): 736-746

[4] Goldwasser S, Micali S, Rackoff C. The knowledge complexity of interactive proof systems. SIAM Journal on Computing, 1989, 18(1): 186-208

[5] Alwen J, Hirt M, Maurer U, et al. Anonymous authentication with shared secrets//Proceedings of the International Conference on Cryptology and Information Security in Latin America. Florianópolis, Brazil, 2014: 219-236

[6] Yao Q, Han J, Qi S, et al. MAP: Authenticating multiple-tags//Proceedings of the International Conference on Mobile Ad hoc and Sensor Systems. Valencia, Spain, 2011: 332-340

[7] Gódor G, Imre S. Hash-based mutual authentication protocol for low-cost RFID systems//Proceedings of the Information and Communication Technologies. Budapest, Hungary, 2012: 76-87

[8] Li X, Liu H, Wei F, et al. A lightweight anonymous authentication protocol using k-pseudonym set in wireless networks//Proceedings of the Global Communications Conference (GLOBECOM). San Diego, USA, 2015: 1-6

[9] Liu J, Zhang Z, Chen X, et al. Certificateless remote anonymous authentication schemes for wireless body area networks. IEEE Transactions on Parallel and Distributed Systems, 2014, 25(2): 332-342

[10] He D, Zeadally S, Kumar N, et al. Anonymous authentication for wireless body area networks with provable security. IEEE Systems Journal, 2016, PP(99): 1-12

[11] Yein A D, Huang Y H, Lin C H, et al. Using a random secret pre-distribution scheme to implement message authentication in VANETs. Applied Sciences, 2015, 5(4): 973-988

[12] Chim T W, Yiu S M, Hui L C K, et al. SPECS: Secure and privacy enhancing communications schemes for VANETs. Ad Hoc Networks, 2011, 9(2): 189-203

[13] Safkhani M, Bagheri N, Peris-Lopez P, et al. On the traceability of tags in SUAP RFID authentication protocols//Proceedings of the IEEE International Conference on RFID-Technologies and Applications. Nice, France, 2012: 292-296

[14] Safkhani M, Bagheri N, Peris-Lopez P, et al. Weaknesses in another Gen2-based RFID authentication protocol//Proceedings of the IEEE International Conference on RFID-Technologies and Applications. Nice, France, 2012: 80-84

[15] Yi X, Wang L, Mao D, et al. An Gen2 based security authentication protocol for RFID system. Physics Procedia, 2012: 1385-1391

[16] Dolev D, Yao A. On the security of public key protocols. IEEE Transactions on Information Theory, 1983, 29(2): 198-208

[17] Avoine G, Coisel I, Martin T. Time measurement threatens privacy-friendly RFID authentication protocols//Proceedings of the 6th International Conference on Radio Frequency Identification: Security and Privacy Issues. Istanbul, Turkey, 2010: 138-157

[18] Alomair B, Poovendran R, Poovendran R, et al. Scalable RFID systems: A privacy-preserving protocol with constant-time identification. IEEE Transactions on Parallel & Distributed Systems, 2012, 23(8): 1536-1550

[19] Zhou J. A quadratic residue-based lightweight RFID mutual authentication protocol with constant-time identification. Journal of Communications, 2015, 10(2): 117-123

[20] Gope P, Hwang T. A realistic lightweight authentication protocol preserving strong anonymity for securing RFID system. Computers & Security, 2015, 55: 271-280

[21] Xu D, Sakurai T, Vu H L. An analytical model of MAC access delay in IEEE 802.11e EDCA//Proceedings of the Wireless Communications and Networking Conference. Las Vegas, USA, 2006: 1938-1943

[22] Chatzimisios P, Boucouvalas A C, Vitsas V. Packet delay analysis of IEEE 802.11 MAC protocol. Electronics Letters, 2003, 39(18): 1358-1359

[23] Mun H, Chan Y Y, Han K, et al. Enhancement of anonymous authentication scheme in wireless sensor network//Proceedings of the International Conference for Internet Technology and Secured Transactions. London, UK, 2010: 1-4

ZHONG Cheng, born in 1994, M. S. candidate. His main research interests include network and information security, privacy protection.

LI Xing-Hua, born in 1978, Ph. D., professor, Ph. D. supervisor. His research interests include network and information security, privacy protection, cloud computing and security protocol formal methodology.

SONG Yuan-Yuan, born in 1990, M. S. candidate. His main research interests include network and information security, privacy protection.

MA Jian-Feng, born in 1963, Ph. D., professor. His research interests include information security, coding theory, and cryptograph.

Background

This research is supported by the National Natural Science Foundation of China (U1708262, 61672413).

What this paper focuses on is to achieve anonymous authentication using the shared-key. Most previous anonymous authentication schemes are based on the asymmetric key. In those schemes, users can employ the public key of the authentication server to encrypt the identity information, or make use of mathematical methods such as elliptic curves to achieve anonymous authentication. Nevertheless, tedious calculation in those schemes leads to serious resource consumption of user, especially does not apply to some limited computing power devices. And authentication schemes based on the shared keys often processes user's shared key with Hash function or CRC, The result acting as the unique message to determine the tag's identity is sent to the server. Upon receiving the message, the server has to try every user's shared key and validate the validity of the message to determine the real user and complete the authentication. But those schemes have been proved to have some other issue.

This paper first prove that Existing anonymous authentication schemes based on the shared keys is vulnerable to the association analysis for the authentication time, which will finally lead to the disclosure of user identity information. And a lightweight anonymous authentication scheme for wireless networks is proposed in this paper. It's based on shared keys, with a user grouping mechanism. Overall, the proposed method considers the security features, mutual authentication, forward and backward security and security against the desynchronization attack, etc. In addition, the scheme only needs a small amount of calculation. This work is particularly valuable because it provides a new idea for anonymous authentication.