各，发造成严重的影响。比如家长控制模式下，小孩子的客户端MAC地址变动位领导早上好，我是智能与服务中心AI组的林夏娜。我本科毕业于中山大学，专业为通信工程，硕士毕业于中山大学的电子与通信专业。我的课题是

下面我将通过以下方面来介绍我的课题工作。

首先简单介绍我的课题背景。

MAC地址是设备唯一标识符，并且无法轻易改变，因此经常被用来实现对用户的追踪。

为了保护用户隐私，IOS和安卓系统等开始推出随机MAC地址功能，通过在不同的SSID等情况下生成不同的随机地址来实现隐藏用户真实地址的目的。但是，对于需要根据MAC地址来进行规则管理的路由器功能而言，MAC地址随机变动时，将导致管理规则失效。从而对路由器功能造成严重的冲击。比如家长控制模式下，如果小孩子的设备MAC地址变动，家长控制的功能便会因为没有识别出小孩设备而失效。为了解决这个问题，需要寻求一种不通过MAC地址也能对设备端进行唯一标识的办法。

这便是本课题需要解决的问题。本课题首先调研了现有的客户端行为技术，然后通过抓包分析客户端的联网行为来提取相关特征，在结合现有技术和抓包分析的基础上实现一种随机MAC地址客户端重识别的算法，即对于开启了随机MAC的设备要进行正确的识别，并且验证所提算法的有效性。

首先，通过调研相关资料，我发现现有的技术大概可分为四种。第一钟是基于硬件信息来对客户端进行重识别，主要通过在设备端上的APP或者WEP上植入相关获取代码，获取设备的运营商、内存、设备ID等相关信息，无法在路由器上实现。第二种是通过网络嗅探获取主机在IP层和TCP层上的IP包长、端口号等来进行操作系统类型的识别，识别的精度比较低，无法对相同操作系统的设备进行区分。第三钟是通过无线抓包分析，可获得设备支持的速率、扩展容量、供应商等信息。最后是基于用户行为识别的办法，通过网络抓包统计和分析DNS、HTTP中的User Agent、Cookie等信息来分辨设备。

为了验证现有技术在课题背景下是否有可用性和挖掘新的特征，我通过搭建如图所示的抓包拓扑来分析客户端的联网行为，并且模拟路由器在正常场景下可能出现的基本情况，比如客户端与路由器的首次连接、客户端忘记密码后再重新连接、客户端连接路由器后不做任何操作或者开启了部分应用等等。

对采集的数据集进行分析，可以发现大部分情况下，客户端在联网之前，需要通过DHCP协议来申请相应的IP地址。DHCP的原理如图所示，首先是发现阶段，客户端广播一个discover包，收到该包的服务器可以恢复offer包告知客户端，接着客户端发送请求包给服务器以申请IP地址，服务器确认后客户端才可以使用IP地址进行联网通信。从客户端发出的请求报文中可以发现，客户端在DHCP协议中发送相关的可选字段来通知服务器该设备的一些信息。比如主机名和供应商信息。因为客户端的主机名是可以进行修改的，因此该字段是可变的。在参数请求列表中，客户端告知服务器它需要得知哪些信息。而不同客户端的可选字段和参数请求列表的顺序和数量具有一定的差异性。对数据集中的特征值进行去重和清洗后发现，在29个安卓客户端中，Iplen的可能取值只有18个。IOS则只有一个，也就是说，26个IOS设备，其发送到IP LEN只有328这一个值，非常的一致，没有一点独特性。如果每个客户端发出的值都不一样，那说明这个值具有较好的独特性，可以用来标识该客户端。如果很多客户端的值都一致，那么通过这个个特征没办法排除掉那些相似的设备。取特征中的值的个数除以总的设备个数粗略估计哪些特征是可用的。因为host name是可以认为改变的，可靠性比较低，因此先排除它作为可靠的标识。从图中可以发现，安卓中的IP包长和供应商信息独特性较好，而IOS只有请求列表具有较小的差异性外，其他均是一致的。Windows也是IP包长和供应商信息。接下来是

具体分析方法是对特征值进行去重后得到其不重复的个数，比如说26个IOS设备的TTL值均为328，那么去重后相当于不重复的TTL个数只有1个，如果26个IOS设备的TTL值都没有重复，即相当于TTL取值个数有26个。也就是说，不重复的个数越多，说明不同设备在该特征上的信息越独特，那么这个信息就越能代表该设备。将每个特征值去重后的个数除以客户端总量，便可得到其独特性。从表格上可以发现，主机名的独特性是最好的。但是用户随时可以在手机上修改设备的主机名称，因此主机名的稳定性比较差。排除主机名后，可选项和供应商信息在安卓和Windows上的独特性都比较好，在IOS上几乎没有独特性。如果几十个客户端所发出的TTL均为255，去重后TTL字段只有一个取值，一点差别都没有。如果5个客户端发出了5个不同的TTL，那么说明TTL具有100%的独特性。如果其中2个客户端的所发出的TTL相同，那么5个客户端的TTL取值只有4个，则TTL的独特性会下降。

先检查用户的MAC地址是否在信息池中，存在的话说明该用户暂未改变MAC地址。直接将该MAC地址的状态更新为在线，并检测用户主机名是否发生改变并将改动同步到用户信息池里。如果信息池中没有找到，说明该设备要么离线用户表中使用了随机MAC地址的某一个用户，要么是首次连接该路由器的新增用户。将此刻的离线用户表作为该设备的待匹配表，然后对待匹配表中的用户进行DHCP特征信息的匹配，并从中滤除特征不匹配的用户。DHCP特征匹配完毕后的待匹配表为空，说明该用户为新增用户，将其信息新增到信息池中。只剩下一个用户是特征匹配上的时，它依然有可能是特征刚好匹配的新增用户。

我所发现的第一个特征信息是关于DHCP协议的。客户端通过DHCP协议中的四步交互向服务器申请IP地址。在此过程中所发出的DHCP Request帧中，客户端根据自身的发包设置发送选择可选项。不同设备在发送哪些可选项及其发送顺序上具有一定的差异性，部分客户端会在Host Name字段和Vendor字段填写设备的主机名和供应商信息。此外，不同客户端向服务器发送的请求列表和封装的IP层长度也有差别。

为了分析这些差别是否足以对客户端进行区分，对每种特征均进行了独特性的统计。这里的独特性指的是该类特征的可能取值个数与客户端总数的比值。比如26个IOS所发出的TTL只会是0或1时，那么TTL在IOS上的独特性则为2种可能取值除以总数26 等于1/13。可以发现，主机名的独特性是最好的，但由于用户可自行修改设备名称，因此其稳定性较差。排除主机名称外，IP层长度、供应商信息和请求列表在安卓和Windows上的独特性较好。但各项特征在IOS上的独特性都很差。

接下来设计了一种利用这些特征独特性进行客户端识别的算法框架。该框架主要维护一个信息池和三个用户表。路由器在用户首次关联的时候将其MAC地址及对应的DHCP特征信息收集到用户信息池里。当有设备发送DHCP包关联路由器时，首先在信息池中匹配该设备MAC地址信息并进行特征更新。如果MAC地址匹配失败，则客户端只能是使用了随机MAC的某一个离线用户或者是首次连接该路由器的新增用户，在离线表中排除滤除DHCP特征不匹配的用户后生成了待匹配用户表。若无待匹配用户则判断为新增用户。当信息表中只有一个用户的DHCP特征是完全匹配时，一种侧重召回率的策略是为了不漏检而将其判为使用了随机MAC的用户，另一种侧重精准率的策略则认为无法排除该设备依然有是特征匹配的新增用户的可能性而判为新增用户。

从算法结果上看，侧重精准率由于完全无法排除新增用户的可能性导致了没有用户会被标识出来。而侧重召回率即查全率的策略中，安卓的识别准确率为64%，Windows为25.8%，但是IOS只有9%。这与前面的独特性分析结论相吻合，DHCP特征在IOS系统上的独特性很差，其识别率便很低。

为了解决IOS识别率极低和进一步提高其他系统识别率，本文在上一节的基础上结合了HTTP信息。通过TCP建立连接后，客户端和服务器之间可通过HTTP GET/POST Response等报文进行信息传输。从客户端发出的GET和POST报文的User Agent特征中，部分数据帧不携带主机信息，也有将主机信息填入User Agent中的数据帧。从第二和三个User Agent中可得知该主机的系统版本、客户端品牌及型号，而这些信息能够直接用于识别客户端。

对于User Agent特征，我采用了正则化匹配的方式将主机的品牌、型号和系统版本进行提取。其中，系统版本可能会因为升级而发生变化。整个算法流程也是围绕对用户信息池的维护和更新、在线用户的管理及待匹配用户表的逐渐滤除来设计。由于客户端品牌是不会发生改变的，因此在匹配特征时基于它更高的权重，并降低可能发生改变的系统版本的权值，该框架能够滤除品牌及型号不一致的用户和找到匹配分值最大的用户。

最终结果显示，DHCP结合User Agent能够有效提升客户端识别准确率。安卓客户端的识别率从64.3%提升到了89.5，在侧重召回率的策略下，IOS客户端识别率也由原来的9%提升到了25.4%。总体识别率从37%提升到了55.8%。优化后的IOS和Windows客户端的识别率依然很低。

IOS和Windows均无法通过DHCP和HTTP特征取得较高的识别率，再次引进了一个新的特征DHCPv6实现算法的进一步优化。DHCPv6具有与DHCP类似的四步交互，不过DHCPv6同时支持两步快速交互。在IOS系统上随机MAC后，在所采集的数据集上，作为DHCPv6标识符的DUID特征值并没有随着MAC地址的改变而改变。

对该特征进行独特性分析，29个安卓客户端仅抓到一个DHCPv6报文，而IOS系统和Windows系统的特征独特比分别为100%和77.8%。这说明该特征在这两个系统上具有极高的区分度。

于是设计了结合DHCPv6的算法框架。在维护信息表的同时，该框架只对DUID值一致的设备端给出识别，以免误过滤待匹配用户。

结合了DHCPv6之后，IOS和Windows在侧重精准率和召回率的策略下的识别率均得到了大幅度的提升，极高的独特比使得IOS识别率从25%提升到99.7%，windows设备识别率也从25.8%提升到95.2%。总体识别率最高可达95%，但安卓系统的提升极为有限。结合全部算法效果来看，DHCP和UA的结合能够有效实现安卓设备的识别，而DHCPv6则是识别IOS和Windows设备的最有效特征。只有三者结合才能完成重识别任务。

最后对课题进行总结和展望。课题在结合