中国科学院大学
University of Chinese Academy of Sciences

硕士学位论文

动态口令的无线接入系统研究

作者姓名：陈逸恺

指导教师：王琼霄 高级工程师

中国科学院信息工程研究所

学位类别：工程硕士

学科专业：计算机技术

培养单位：中国科学院信息工程研究所

2018 年 6 月

Research on WLAN System of Dynamic Password

A thesis submitted to

University of Chinese Academy of Sciences

in partial fulfillment of the requirement

for the degree of

Master of Engineering

in Computer Technology

By

Chen Yikai

Supervisor： Senior Engineer Wang Qiongxiao

Institute of Information Engineering, Chinese Academy of Sciences

June 2018

中国科学院大学
研究生学位论文原创性声明

本人郑重声明：所呈交的学位论文是本人在王琼霄导师的指导下独立进行研究工作所取得的成果。尽我所知，除文中已经注明引用的内容外，本论文不包含任何其他个人或集体已经发表或撰写过的研究成果。对论文所涉及的研究工作做出贡献的其他个人和集体，均已在文中以明确方式标明或致谢。

作者签名：

日　　期：

中国科学院大学
学位论文授权使用声明

本人完全了解并同意遵守中国科学院有关保存和使用学位论文的规定，即中国科学院有权保留送交学位论文的副本，允许该论文被查阅，可以按照学术研究公开原则和保护知识产权的原则公布该论文的全部或部分内容，可以采用影印、缩印或其他复制手段保存、汇编本学位论文。

涉密及延迟公开的学位论文在解密或延迟期后适用本声明。

作者签名：　　导师签名：

日　　期：　　日　　期：

摘 要

WLAN 以其高速的上网体验和低廉的价格赢得了大众的青睐，人们对 WLAN 的需求日益高涨。为了满足外来人员访问 WLAN 的需求，不少机构部署了访客 WLAN。此类 WLAN 需要实现更细粒度的授权管理，每位来访人员均应只在访问期内拥有 WLAN 访问权：访客来访时被授予 WLAN 访问权，而访问结束时，该访问权应当被及时撤销。同时，WLAN 系统应尽可能减少管理员和用户的负担，降低自身的运维成本。然而，现有的 WLAN 鉴别方式均无法很好地将上述两个目的有机地结合起来。

本设计提出并实现了一种结合物理认证的 WLAN 动态口令更新方法及系统，可应用于大型机构的访客 WLAN，在兼顾不同类型用户 WLAN 访问需求的同时，尽可能降低 WLAN 运维成本，提高 WLAN 管理员和用户的使用体验，同时提高 WLAN 系统的安全性。

WLAN 动态口令方案在现有 WPA-PSK 鉴别协议的基础上，引入动态口令机制。WLAN 口令每隔设定的时间间隔自动更新，移动终端只有与 AP 共享相同的口令才能正常通过 WPA-PSK 协议接入 WLAN。一旦 AP 更新了口令，移动终端必须同步更新口令，才能再次接入 WLAN。更新过程引入物理认证因素，持有移动终端的用户必须通过物理认证、进入受控物理环境，移动终端才能计算出新口令，因而失去物理认证权限的用户则会被拒绝再次接入 WLAN。更新口令需要一随机数，称为物理认证参数，该参数由特定的设备产生并广播到设定的、受物理访问控制保护的物理环境中。AP 口令更新后，移动终端只有获得新的物理认证参数，才能计算得到与 AP 相同的新口令。

WLAN 动态口令系统包含三部分：能够独立更新 WLAN 口令的主 AP，需要和主 AP 交互才能更新 WLAN 口令的从 AP，以及能够根据需要自动更新 WLAN 口令的移动终端。主 AP 每隔设定的时间间隔自动生成和广播物理认证参数，并更新自身口令。主 AP 完成口令更新后，从 AP 即向主 AP 请求新口令以完成自身口令的更新。而移动终端则能根据不同的情形使用当前口令接入 WLAN、计算新口令并使用新口令接入 WLAN、提示用户进入受控物理环境以

更新口令、以及提示用户口令已彻底失效。

本设计将物理认证和 WLAN 认证有机地结合起来，使得来访人员只能在访问期内接入 WLAN，实现了来访人员 WLAN 访问权的动态授予和撤销，同时不增加 WLAN 管理员和用户的操作负担，还提高了 WLAN 系统的安全性。

关键词：动态口令，WLAN 接入，物理认证

Abstract

WLAN is becoming increasingly popular because of its higher speed and cheaper price compared with other wireless technologies. To meet the demand of visitors to access WLAN, many organizations have deployed guest WLANs. Such WLANs require finer grained authorization management. Each visitor should only be allowed to access WLAN during the visit period. Visitors are authorized to access WLAN when they arrive. However, such authorization will be revoked once they leave away. At the same time, WLAN systems should minimize the burden on WLAN administrators and users and reduce WLAN operation and maintenance costs. However, none of the existing WLAN authentication methods well combine the above two goals.

This design proposes and implements a WLAN dynamic password scheme combined with physical access control, which can be applied to large-scale guest WLANs of large organizations. The proposed scheme can satisfy different requirements of different WLAN users. Meanwhile, it can reduce WLAN operation and maintenance costs, improve the experience of WLAN administrators and users, and improve the security of WLAN systems.

The WLAN dynamic password scheme introduces dynamic password mechanism into the WPA-PSK protocol. The WLAN password is automatically updated every specified interval. Mobile devices can access WLAN normally by the WPA-PSK protocol only if they share the same password with the target AP. Once the AP updates the password, mobile devices must update the password synchronously to access WLAN again. A physical authentication factor is introduced to the password updating process. Users must pass the physical authentication and enter a specific location so that their mobile devices can calculate the new password and continue accessing WLAN. However, users who can no longer pass the physical authentication will be rejected to access WLAN again. Updating password requires a random

number, called physical authentication parameter, which is generated by a specific device and broadcast to a specific location which is protected by physical access control. After the password is updated, mobile devices can get a new password that is the same as that of the AP only after acquiring the new physical authentication parameter.

The WLAN dynamic password system consists of three parts: a master AP that can independently update its own WLAN password, several slave APs that needs to interact with the master AP to update their own WLAN passwords, and several mobile devices that can automatically update their own WLAN passwords when needed. The master AP generates and broadcasts physical authentication parameters and updates its own password every specified interval. Once the master AP gets a new password, the slave AP requests the new password to update its own password. While the mobile device will use the current password to access the WLAN, calculate a new password and use the new password to access the WLAN, prompt the user to enter the specific location to update the password, or prompt the user that the password is completely expired according to different situations.

This design combines WLAN authentication with physical access control so that visitors can only access WLAN during their visit period. Visitors' authorization will be dynamically granted and revoked without increasing the operation burden of WLAN administrators and users. It will also improve the security of WLAN systems.

Key Words: Dynamic Password, WLAN Access, Physical Authentication

目 录

图目录

表目录

第 1 章 引言

1.1 背景和意义

近年来，移动互联网领域有了长足发展。首先是移动智能终端如智能手机、平板电脑、笔记本电脑等的性能不断提高，功能更加健全，操作更加简单，价格也越来越亲民，越来越多的人愿意买、也有能力买各类智能移动终端。在国内，智能手机用户群体涵盖了绝大部分人群，智能手机已成为人们日常工作学习生活中不可或缺的物品之一[1]。与此同时，互联网（尤其是移动互联网）上的资源数量也是与日俱增，涵盖工作学习生活的方方面面。借助互联网，人们可以开展社交、订餐、出行、购物、支付、理财、娱乐等各种活动，因此人们对互联网的依赖也正日益增强。在移动智能终端和互联网不断更新换代的同时，无线接入技术也在不停地更新换代：从 2G 到 3G 再到 4G，现在又开始进入 5G 时代，无线访问更快、更稳定、也更安全。尽管如此，有一种无线技术仍然是人们的首选，那就是 WLAN 技术。WLAN 技术一直受到人们青睐的原因是它访问网络足够快、足够稳定、足够安全，同时也足够便宜。人们不需要额外为网络流量付费，能够随意在线观看高清视频、下载大体积文件等。智能移动终端普及和互联网资源的丰富助长了人们对 WLAN 的需求，也刺激了 WLAN 的发展，越来越多的机构都开始部署 WLAN，人们可以在家里、工作单位、饭店、宾馆、甚至机场、火车站、公园、公交车上收到 WLAN 信号[2]。

工作上，随着在线会议、网上办公、以及云端存储等各项概念的提出，不少任务都由线下转移到了线上，许多事务都需要在网上处理，因此人们在工作中对网络访问的需求也越来越大。以我们研究所为例，经常有合作单位员工来我们研究所与我们研究所的员工协同工作，他们需要使用 Microsoft Office Online 来编辑 Word 文档、展示 PowerPoint 幻灯片，需要使用百度网盘来上传和下载办公材料，需要收发电子邮件，需要使用微信和不在研究所的其他人员沟通工作事项，等等。正因为如此，一个单位如果经常有外来人员访问的话，其在部署 WLAN 时就需要考虑来访人员的 WLAN 接入问题[3]。

与机场、火车站等单位不同，像我们研究所这样的单位并非对社会公众开放，

只有单位内部员工以及受邀而来的外部人员才能进入本单位的工作场所。相应地，此类单位部署的访客 WLAN 也仅允许有限的人员访问——单位员工和访客。然而这两类人员访问 WLAN 的需求并不相同：内部人员当然希望能够长久地接入 WLAN，而访客则只需要在访问期内能够接入 WLAN 即可，WLAN 不必要也不应该赋予访客过长的访问权。由于不同访客的来访时间不同，访问期限也不同，他们接入 WLAN 的需求也不尽相同。因此与通常的、供固定人员接入的 WLAN 不同，供访客接入的 WLAN 需要解决以下两个问题：一是实现区分不同类型用户的授权管理——内部员工能够长久地接入 WLAN，而访客只在来访时间段内能够接入 WLAN；二是实现对短期用户的动态授权——访客来访时被授予 WLAN 访问权，此授权在访客离开时应当被及时撤销。

在可预见的未来，人们接入 WLAN 的需求将一直存在并且很可能进一步加强，那些经常有外部人员来访的单位很有部署访客 WLAN 的必要，而部署访客 WLAN 就需要解决上述需求。然而目前很多访客 WLAN 并没有解决上述需求，或者虽然能够解决上述需求，但是 WLAN 的维护成本很高。除此之外，目前单位普遍采用的 WLAN 鉴别方式还存在一定的安全隐患。目前 WLAN 的鉴别方式主要有以下几种：

（1）不鉴别[5]。WLAN 公开，没有口令、后台鉴别服务器或者其他任何方式鉴别移动终端，任何能够收到 WLAN 信号的移动终端都可以接入 WLAN。

公开 WLAN 的好处就是方便，来访人员只要能够收到 WLAN 信号，就能直接接入 WLAN 而不必询问他人，这也减轻了管理员的负担。但是公开 WLAN 的缺点也很明显：它没有对用户做任何访问控制，无论是内部员工还是来访人员都有长期的 WLAN 访问权，无法实现对访客访问 WLAN 的动态授权及撤销。不仅如此，未经授权的非法用户也能接入 WLAN。除此之外，公开 WLAN 上传输的消息不经加密处理，攻击者能够截获明文数据，用户的隐私信息就可能遭到泄露。央视 315 晚会连续两年报道公共免费 WLAN 存在安全隐患，并现场演示了通过公开 WLAN 获取用户的隐私信息：用户接入晚会现场的公开 WLAN 后打开一两个常用应用，浏览一下消费记录，该用户的姓名、手机号、银行卡号和身份证号等信息就在大屏幕上显示了出来[4]。

（2）WPA-PSK 鉴别方式[5]。WLAN 被设置有唯一的口令，用户只有在移

动终端上输入该口令才能接入 WLAN。为接入 WLAN，用户需要向管理员或者知道口令的用户询问口令。

通常小型 WLAN 都会选择此种鉴别方式，因为此种鉴别方式易于部署，且维护成本低，管理员只需要在 AP 端设置一次口令即可。同时，该方式也更加安全：后续通信数据经加密后传输，攻击者在不知道口令的情况下无法得到加密密钥，也就无法获取用户的隐私信息。但是由于所有用户都共享相同的口令，此种鉴别方式无法区别不同类型的用户。且由于口令在较长的一段时间内保持不变，访客在被告知口令后就能不受限制地接入 WLAN 直至管理员更改口令，换言之，WLAN 无法在访客结束访问后及时撤销对其访问 WLAN 的授权。除此之外，口令通常比较简单，容易遭到暴力穷举攻击，同时由于口令长期不变，攻击者有较长的时间猜测口令。一旦攻击者获得了口令，使用口令保护的 WLAN 将和公开 WLAN 没有区别：攻击者能够接入 WLAN，影响合法用户的上网体验；也能够设立钓鱼 AP，欺骗合法用户与之建立连接，窃取用户的隐私信息；还能够解密移动终端与 AP 之间收发的数据，窃取用户的隐私信息等。

（3）WAPI-PSK[5]。此种鉴别方式在用法上与 WPA-PSK 鉴别方式类似，只是在技术上，其采用的数据格式、认证协议和加密算法等与 WPA-PSK 鉴别方式有所不同。

（4）WPA-EAP 鉴别方式[5, 6, 7, 8]。此种鉴别方式由后台鉴别服务器（通常为 RADIUS 服务器）鉴别用户身份，用户只有向后台鉴别服务器证明自己的身份后才能接入 WLAN。其中后台鉴别服务器鉴别用户身份的方式又有如下几种：

- 用户名和口令认证，用户在接入 WLAN 时需要在移动终端输入自己的用户名和口令，后台鉴别服务器根据用户名是否在用户列表中、口令是否正确来决定是否允许用户接入 WLAN；
- 数字证书认证[9]，用户在接入 WLAN 时需要在移动终端提供包含私钥的数字证书，后台鉴别服务器验证数字证书拥有者确实为当前接入用户后方可允许用户接入 WLAN；

此类鉴别方式下，管理员需要为每个有权接入 WLAN 的用户创建账户。同时针对不同的细分鉴别方式，为每个用户账户设置用户名和口令、并将用户名和

口令分发给每个用户，或者要求每个用户提供其用于接入 WLAN 的数字证书。针对来访人员，管理员可能还会为其设置访问期，那么后台鉴别服务器在鉴别用户身份时就还需要额外判断用户的接入时间是否仍在访问期内。来访人员访问结束后，设置了访问期的用户账户自动失效，来访人员不能再凭原有的用户账户接入 WLAN。管理员也可以通过及时删除用户账户来撤销来访人员的 WLAN 访问权。

此类鉴别方式适合大型 WLAN，尽管如此，大多大型 WLAN 并未采取此类鉴别方式。此类鉴别方式能够提供健壮安全的 WLAN：一方面，用户名和口令在 TLS 隧道内传输，用户凭据不会泄露；另一方面，移动终端每次接入 WLAN 时都会和后台鉴别服务器协商产生一对加密后续通信数据用的根密钥，根密钥只有移动终端、AP 和后台鉴别服务器知晓，攻击者无法窃取用户通过 WLAN 传输的隐私信息。同时，此类鉴别方式也能够实现用户访问 WLAN 的细粒度授权：每个用户访问 WLAN 的起止时间都可以不同，可早可晚、可长可短。但是此种鉴别方式是一种完全中心化的方式，十分依赖管理员，管理员的工作负担较重：公开 WLAN 不需要管理员参与，用户就能接入 WLAN，使用口令鉴别的 WLAN 用户既可以向管理员获取口令、也可以向知道口令的用户获取口令，但使用后台鉴别服务器鉴别的 WLAN 必须由管理员为每个用户创建用户账户，用户也只能向管理员获取用户名和口令，管理员的工作量就会比较大。这对那些经常有外来人员访问的 WLAN 而言尤其麻烦，管理员需要频繁地创建和删除用户账户。

为了方便，有的管理员会创建唯一的宾客账户供所有来访人员访问 WLAN 时使用。尽管降低了管理员的工作量，但也无法及时撤销对来访人员访问 WLAN 的授权，因为宾客账户的用户名和口令通常在较长的一段时间内保持不变，与使用口令保护的 WLAN 类似，来访人员在被告知宾客账户的用户名和口令后就能一直接入 WLAN，即使访问早已结束。

同样为了减少管理员管理访客账户的工作量，教育科研机构的访客 WLAN 还可以采用 Eduroam[10]。希望部署 Eduroam WLAN 的机构需要事先加入 Eduroam 联盟，联盟成员机构的员工在访问联盟内其他机构时，可以使用其所属机构的用户名和口令接入 Eduroam WLAN，无需受访机构为这些人员创建用户账户。用户在接入 Eduroam WLAN 时，其输入的用户名和口令会被传递至其所

属机构的鉴别服务器，由其所属机构的鉴别服务器决定是否允许该用户接入WLAN。此种鉴别方式最大的缺点就是使用范围有限，只有加入了 Eduroam 联盟的机构能够部署 Eduroam WLAN，也只有 Eduroam 联盟成员机构的员工能够接入 Eduroam WLAN，其他机构来访人员临时接入 WLAN 的需求并没有得到有效解决。

（5）WAPI-Cert 鉴别方式[5]。此种鉴别方式在用法上与采用数字证书认证的 WPA-EAP 鉴别方式类似，只是在技术上，其采用的数据格式、认证协议和加密算法等与 WPA-EAP 有所不同。

（6）WiFiDog 加 Web 认证网关的鉴别方式[5, 11]。此种鉴别方式也由后台鉴别服务器鉴别用户身份。与 WPA-EAP 鉴别方式不同，此种鉴别方式并非在用户接入 WLAN 之前、而是在用户接入 WLAN 后鉴别用户身份。严格意义上来说，这是一种 WLAN 访问控制方式而非 WLAN 鉴别方式。用户接入 WLAN 后，其能够访问的网络站点受到限制（通常只能访问内网），用户必须在内网的 Web 认证网关上登录后才能访问外网。其中用户登录认证网关的方式又有如下几种：

- 手机号登录[12]，用户首先需要在认证网关输入自己的手机号，而后认证网关会向该手机号发送一条验证码，用户输入收到的验证码后完成登录，登录后用户将获得一段时间的 WLAN 访问权；
- 微信一键登录[13]，用户使用微信提供的单点登录服务登录认证网关，用户在认证网关选择微信一键登录，认证网关会唤起微信应用，用户在微信应用界面同意使用微信单点登录后完成登录（通常认证网关还会要求用户关注指定的公众号），登录后用户将获得一段时间的 WLAN 访问权。用户也可以在接入 WLAN 后直接打开微信应用并关注指定的公众号，在指定的公众号栏目下选择“一键连 Wi-Fi”以登录认证网关；
- 微信好友登录[14]，与上一种登录方式相同，用户也使用微信提供的单点登录服务登录认证网关，但是认证网关会要求登入用户必须是指定微信用户的微信好友，也就是说，管理员需要事先将自己的微信账号和认证网关进行绑定；
- 邀请人邀请登录，用户打开认证网关后，认证网关会展示一个二维码，只有 WLAN 长期用户（即邀请该用户加入 WLAN 的其他用户）扫描该二维码后才能登录；或者由 WLAN 长期用户在认证网关中输入授权码后才能登录。用户

登录后将获得一段时间的 WLAN 访问权；

- 用户名和口令登录[12]，用户在认证网关输入用户名和口令后完成登录。

当然还有其他的登录方式，理论上来说，认证网关可以采用任何 Web 网站可以采用的登录方式。就便捷性而言，除了用户名和口令登录外，上述其他登录方式均不需要管理员创建用户账户，能够有效降低管理员的工作负担。就安全性来说，不同的细分登录方式的安全性不同。总的来说，此类鉴别方式只能实现用户的访问控制，无法为移动终端的无线流量提供更多的安全保障，移动终端无线流量的安全性仍依赖于其接入 WLAN 时采用的鉴别方式。如果移动终端在接入 WLAN 时未被鉴别用户身份，亦即该 WLAN 为公开 WLAN（这也是大多数的情况），通信数据将以明文形式在 WLAN 上传输，毫无安全性可言。其中，手机号登录和微信一键登录在访问控制上也与公开 WLAN 无异，只要有手机号或者微信号的用户均可接入 WLAN，并且可以无限制地接入 WLAN。而后三种登录方式则只有被授权的用户才能接入 WLAN，能够实现临时用户 WLAN 访问权的动态授予和撤销。

（7）MAC 地址过滤。[5]即管理员预先在 AP 上设置允许接入的移动终端的 MAC 地址和/或拒绝接入的移动终端的 MAC 地址，相当于为接入 WLAN 的移动终端建立黑名单和/或白名单，MAC 地址在白名单内或不在黑名单内的移动终端才会被允许接入 WLAN。此种鉴别方式也相当于为每个用户建立了用户账户——需要频繁地修改允许接入 WLAN 的移动终端的 MAC 地址白名单，管理员的工作负担较重。并且 MAC 极易伪造，此种鉴别方式也不具有足够的安全性。

总的来说，现有的鉴别方式或者无法满足不同类型用户细粒度管理的需求，或者无法提供足够的安全性，或者会给管理员和用户造成过重的操作负担。本设计旨在提出并实现一种新的 WLAN 鉴别方案，专门针对大型访客 WLAN，能够兼顾访客 WLAN 不同用户的不同需求，同时尽可能地降低 WLAN 的维护成本，给管理员和用户带来更好的使用体验，同时提高 WLAN 的安全性。

1.2 本设计的主要研究工作

本设计参考了其他学者提出的动态口令算法、结合物理位置的鉴别授权机制、以及基于口令的 WLAN 鉴别协议（802.11i 协议的预共享密钥模式，亦即 WPA-PSK 协议）的安全性增强方案，提出了一种新的、兼顾便捷性和安全性的

访客 WLAN 鉴别方案，即基于物理访问控制的 WLAN 动态口令方案。

方案在原有使用静态口令鉴别移动终端的基础上，引入动态口令机制。WLAN 口令会根据设定的时间间隔自动变化，为接入 WLAN，移动终端和 AP 必须共享相同的 WLAN 口令，因此 AP 口令更新后，移动终端必须同步更新口令才能接入 WLAN。但是只有授权用户的移动终端能够得到变化后的新口令，从而再次接入 WLAN。通过不正当渠道获得口令的用户以及结束访问的访客的移动终端将无法计算出新口令，也就无法再次接入 WLAN。

为了确保只有授权用户能够更新口令，系统引入了物理访问控制。大型机构通常都有部分区域受物理访问控制保护，只有内部员工以及正在访问的访客才有权限进入这些区域。口令在更新过程中引入了只有在设定的、受物理访问控制保护的物理环境（受控物理环境）中才能获得的参数，称之为物理认证参数，该参数是计算新口令的必需参数。为接入 WLAN，合法用户只有通过物理认证、进入受控物理环境，才能获得物理认证参数，也才能计算得到新口令，在 WLAN 口令更新后再次接入 WLAN。

本设计实现了基于物理访问控制的 WLAN 动态口令系统，系统内所有 AP 都会在设定的时间点自动更新口令，无需管理员参与；而移动终端也会根据具体情况使用原口令或者更新口令后使用新口令接入 WLAN，更新过程对用户完全透明。经测试，口令定期更新给用户接入 WLAN 造成的时延可以忽略不计，用户无法察觉到口令过程。除此之外，口令更新对 WLAN 系统内各 AP 的各项性能指标也没有什么太大的影响。

1.3 本文结构

本文由五个章节组成，具体安排如下：第一章是引言，主要介绍访客 WLAN 鉴别方案研究的背景和意义，并简要概括了本设计完成的工作。第二章介绍了国内外相关研究，包括动态口令算法和结合物理位置的鉴别授权机制，另外还介绍了 802.11i 协议的预共享密钥模式及其存在的安全问题和提出的改进方案。第三章和第四章分别介绍了基于物理访问控制的 WLAN 动态口令系统方案及实现情况。第五章总结了本文内容及创新点。

第 2 章 相关研究

2.1 动态口令算法

口令认证是目前互联网上最常用的认证方式之一，口令认证不需要 U 盾、指纹识别器等认证专用设备，不易受其他因素干扰，因意外而导致无法登录的概率低。用户只要记住用户名和口令，并在登录界面上输入用户名和口令即可完成登录，十分方便。但是口令也存在着简单易猜测、易被窃取、存在重放攻击等缺点[15]。为了提高口令认证的安全性，学者提出了动态口令的概念。动态口令是指每隔一小段时间就发生变化、或每次登录都不一样的口令。动态口令生成算法即如何生成某段时间或者某次的口令是动态口令的核心，动态口令生成算法主要有：基于时间或计数器同步的动态口令算法、基于挑战和响应的动态口令算法、基于逆向杂凑链的动态口令算法和基于正向杂凑链的动态口令算法。

2.1.1 基于时间或计数器同步的动态口令算法

此种算法下，认证方和被认证方各自独立地产生相同的口令，登录时，被认证方将口令发送给认证方，认证方比较收到的口令和自己计算出来的口令是否相等，若是则被认证方通过认证。例如工行的电子密码器，能够显示 6 位数字密码，且每隔一小段时间就发生变化，用户输入电子密码器上的 6 位数字密码即可完成转账。再如目前人们付款常用到的支付宝付款码和微信付款码，就是二维码形式的动态口令。付款码每 60 秒更新一次，由商家扫描后发送给第三方支付机构，第三方支付机构以该口令作为用户授权向商家付款的凭据。无论是电子密码器的数字密码还是付款码，被认证方更新口令不需要认证方的参与。口令以某些认证双方共有的动态信息为基础产生，保证口令动态变化。但是认证双方又需要同步这些信息，保证认证双方能够在同一时刻产生相同的口令。动态信息本身可以是公开信息，如当前时间。因此认证双方还需要共享某一密钥，作为产生口令的参数之一，保证只有认证双方能够计算出口令。

RFC 4226 标准提出了一种基于计数器同步的动态口令算法[16]。在共享密钥的前提下，认证双方需要各自维护一台计数器，并通过某种方式同步双方的计数器，保证双方的计数值始终相等。当被认证方需要使用口令登录时，认证方和被

认证方都使用下述公式计算口令：

$$OTP(K,C) = Truncate(HMAC - SHA - 1(K,C)) \quad (2.1)$$

其中 *OTP* 表示动态口令，*K* 表示认证双方共享的密钥，*C* 表示当前计数值，Truncate 函数表示将长字节串转化为可显示、用户可键入的短数字字符串。

RFC 6238 标准在上述标准的基础上，提出了一种基于时间同步的动态口令算法[17]。认证双方无需专门维护一台计数器，而以系统时间为基础计算计数值，计算公式如下：

$$C = (T - T_0) / X \quad (2.2)$$

其中 T 表示当前时间，以 Unix 时间戳表示，T_0 表示起始时间，通常为 0，X 表示时间间隔，标准建议定为 30 秒。

2.1.2 基于挑战和响应的动态口令算法

与基于时间或计数器同步的动态口令算法不同，基于挑战和响应的动态口令算法中，认证双方共享的动态信息由认证方一方产生（称为挑战）并发送给被认证方，认证双方再使用和基于时间或计数器同步的动态口令算法中描述的口令计算公式类似的公式计算得到动态口令。因此，认证双方事先也需要共享一对密钥，确保只有认证双方能够计算出口令。

Steffen Hallsteinsen 和 Ivar Jørstad 提出了一种利用手机短信传输挑战的动态口令算法[18]。认证双方事先需要使用 SPEKE 协议协商产生一对密钥 *Key*，作为后续生成动态口令的基础。当被认证方需要登录时，认证方产生一个随机数作为挑战 *Challenge*，并通过手机短信的形式发送到被认证方的手机。被认证方收到挑战后，按以下公式计算动态口令：

$$OTP = Hash(Challenge \parallel Key) \quad (2.3)$$

为了让被认证方验证短信的真实性，认证方在发送挑战的同时还附带发送了动态口令的消息鉴别码。被认证方生成动态口令后，通过验证消息鉴别码的正确性来验证短信确实由认证方发出。

事实上，目前很多利用手机短信的动态口令方案并没有涉及到这么复杂的操作：动态口令由认证方单独产生并通过手机短信发送给被认证方，被认证方收到口令后即按照传统口令认证方式完成认证，并不需要做额外的计算。换言之，通

过手机短信传输的并非挑战，而是动态口令本身。严格来说，这是一种动态口令发布方式，而不是动态口令生成方式。例如建行客户在使用账号支付进行网上投资消费时，建行会向客户手机发送 6 位随机数验证码，客户输入验证码后就能完成支付。在支付过程中，用户并未参与动态口令（即随机验证码）的产生，动态口令由建行服务器独立产生，生成算法任意，并通过手机短信发送给用户，用户收到验证码后再通过互联网发送给服务器，完成付款。此类方案基于的前提假设是手机短信是安全的，如果手机短信不安全，动态口令将被攻击者截获。

2.1.3 基于逆向杂凑链的动态口令算法

Leslie Lamport 提出了一种基于逆向杂凑链的动态口令算法[19]。被认证方首先产生一个随机数作为种子密钥 *Seed*，种子密钥本地存储，并不发送给认证方。然后将该种子密钥杂凑若干次，得到一条杂凑链：

$$Seed, Hash(Seed), Hash(Hash(Seed)), ..., Hash^{n}(Seed) \quad (2.4)$$

被认证方逆序使用杂凑链中的杂凑值：注册时，被认证方将杂凑链最后一个值 $Hash^{n}(Seed)$发送给认证方，认证方保存该值。第一次登录时，认证方将杂凑链倒数第二个值 $Hash^{n-1}(Seed)$作为本次登录的口令发送给认证方。认证方将收到的口令杂凑后和事先保存的值比对，若相等则认证通过。认证通过后，认证方保存本次登录口令，留作下次认证时使用，以此类推。因此，第 *m* 次登录的动态口令

$$OTP = Hash^{n-m}(Seed) \quad (2.5)$$

由被认证方发送给认证方，而认证方事先保存 $Hash^{n-m+1}(S)$，并通过验证等式

$$Hash(OTP) = Hash^{n-m+1}(Seed) \quad (2.6)$$

是否成立来决定认证是否通过。

但是此种算法存在的问题是杂凑链长度有限，被认证方登录若干次后就会因为用完杂凑链中所有杂凑值而无法继续登录，除非重置杂凑链。António Pinto 和 Ricardo Costat 提出了一种能够自动更新杂凑链的动态口令算法[20]，算法结合了 Lamport 动态口令算法以及基于时间同步的动态口令算法——使用基于时间同步的动态口令算法更新杂凑链，即更新杂凑链的种子密钥。认证双方事先需要共享

一对密钥，在需要时，使用该密钥生成一条新的杂凑链。当杂凑链用尽、需要更新杂凑链时，被认证方首先产生一个随机数 *Nonce1*，然后以 *Nonce1*、当前时间 *Time* 和预共享密钥 *Key* 为输入计算另一个数 *Nonce2*，

$$Nonce2 = Hash(Time, Hash(Key), Nonce1) \quad (2.7)$$

最后将 *Nonce1* 和 *Nonce2* 发送给认证方。认证方收到 *Nonce1* 和 *Nonce2* 后，首先验证 *Nonce2* 的有效性，即验证等式

$$Nonce2 = Hash(Time, Hash(Key), Nonce1) \quad (2.8)$$

是否成立。考虑到网络传输存在时延，即认证方验证 *Nonce2* 的时间晚于被认证方生成 *Nonce2* 的时间，认证方会同时比较以下两个等式是否成立：

$$Nonce2 = Hash(Time - 1, Hash(Key), Nonce1) \quad (2.9)$$

$$Nonce2 = Hash(Time - 2, Hash(Key), Nonce1) \quad (2.10)$$

亦即，认证方允许存在两个口令更新周期的时延。如果三个等式中任意一个等式成立，就认为 *Nonce2* 有效。此时，认证方会产生一随机数 *Random*，并以该随机数和 *Nonce2* 为基础产生杂凑链的种子密钥 *Seed*：

$$Seed = Hash(Random, Nonce2) \quad (2.11)$$

种子密钥一旦产生，杂凑链随即生成。认证方将其产生的随机数以及杂凑链的最后一个值发送给被认证方，有了认证方产生的随机数，加上之前产生并发送给认证方的 *Nonce2*，被认证方也能计算出和认证方相同的种子密钥，生成一条和认证方相同的杂凑链，被认证方通过比较杂凑链的最后一个值是否相等来决定是否更新杂凑链。杂凑链更新后，动态口令的用法和 Lamport 算法一致。

2.1.4 基于正向杂凑链的动态口令算法

Huiyi Liu 和 Yuegong Zhang 提出了一种基于正向杂凑链的动态口令算法[21]。与逆向杂凑链不同，正向杂凑链不需要事先计算出整条杂凑链，而是每次登录时对原口令杂凑若干次作为新的口令，杂凑链可以无限延伸，也就不存在登录次数有限的问题。算法需要两个杂凑函数 $Hash_A$ 和 $Hash_B$，其中 $Hash_A$ 用以生成杂凑链，而 $Hash_B$ 则用以安全传输口令，防止口令在传输过程中泄露，因为对于正向杂凑链而言，一旦链中任意一个口令泄露，攻击者很容易就能计算出后续口令。

认证双方需要共享一对种子密钥，作为杂凑链的起始值。除此之外，被认证方需要维护以下变量：

I_{user}，记录被认证方当前所处的杂凑链的位置，即被认证方对种子密钥杂凑的次数，初始为 0；

I_{user_server}，记录认证方当前所处的杂凑链的位置，即认证方对种子密钥杂凑的次数，初始为 0；

S_{cur_user}，被认证方当前的密钥值，即对种子密钥杂凑 I_{user} 次后的值，该值和 I_{user} 始终保持对应关系，初始值为种子密钥。

而认证方也需要维护类似变量：

I_{server}，记录认证方当前所处的杂凑链的位置，即认证方对种子密钥杂凑的次数，初始为 0；

S_{cur_server}，认证方当前的密钥值，即对种子密钥杂凑 I_{server} 次后的值，该值和 I_{server} 始终保持对应关系，初始值为种子密钥。

注册阶段，认证方和被认证方各自对种子密钥杂凑 m 次（m 为随机数），将 I_{user}、I_{user_server}、I_{server} 增加 m，同时更新 S_{cur_user} 和 S_{cur_server}。

登录阶段，

（1）被认证方首先计算

$$delta = I_{user} - I_{user\_server} \quad \text{.....(2.12)}$$

如果 *delta* 等于 0，被认证方应当将当前密钥杂凑 m 次（m 为随机数），将 I_{user} 增加 m，同时更新 S_{cur_user}。这样能够保证每次登录都会使杂凑链向前延伸，从而产生和上次登录不同的口令。

（2）在 *delta* 不为 0 的前提下，被认证方将 *delta* 发送给认证方。认证方收到 *delta* 后，将当前的密钥值杂凑 *delta* 次，得到的结果应该和被认证方当前的密钥值相同。

（3）然后，认证方发送响应

$$ANS = Hash_B^{delta}(Hash_A^{delta}(S_{cur\_server})) \quad \text{.....(2.13)}$$

给被认证方，被认证方收到认证方的响应后，通过验证等式

$$ANS = Hash_B^{delta}(S_{cur\_user}) \quad \text{.....(2.14)}$$

来验证响应确实由认证方发出。验证通过后，被认证方需要继续将当前密钥杂凑 n 次（n 为随机数），将 I_{user} 增加 n，同时更新 S_{cur_user}。然后计算动态口令 OTP：

$$OTP = Hash_B(S_{cur\_user}) \quad (2.15)$$

（4）被认证方将 n 和 OTP 一同发送给被认证方，被认证方按照下述公式计算 OTP

$$OTP = Hash_B(Hash_A^{\,n} Hash_A^{\,delta}(S_{cur\_server}))) \quad (2.16)$$

通过比对计算得到的 OTP 和收到的 OTP 是否相等来决定被认证方是否通过认证。一旦认证通过，认证方应当将 I_{server} 增加 $delta + n$，同时更新 S_{cur_server}；而被认证方也需要更新 I_{user_server}。

2.1.5 小结

总的来说，现有的动态口令算法都要求认证双方事先共享一对密钥，不同用户与服务器共享不同密钥，即服务器为每个用户创建用户账户。如果需要创建用户账户，已有的 WPA-EAP 认证协议足以提供健壮安全的 WLAN 网络。并且除了基于逆向杂凑链的动态口令算法，用户一旦获得了根密钥就能计算出任意时间的动态口令，无法实现对临时用户的动态授权和撤销。除此之外，基于逆向杂凑链的动态口令算法使用次数有限，需要经常更新种子密钥，对于大型 WLAN 而言，由于涉及用户众多，更新种子密钥是不现实的。而基于正向杂凑链的动态口令算法，其杂凑链能够无线延伸，方便用户长久接入，但是需要认证方和被认证方多次交互，而 AP 和移动终端并没有多次交互以产生口令的条件，最好的方法是 AP 和移动终端双方各自独立地生成动态口令。此外，正向杂凑链存在的问题是一旦链中的任意一个口令泄露，将导致后续口令泄露，利用手机短信或类似的安全信道来传输更新口令所必须的参数（即挑战）或许可以成为防止后续口令泄露的方法之一。

2.2 结合物理位置的认证授权

很多情况下，用户只有处于特定的物理位置时才需要登录或者获取授权，比如通常人们只在工作单位才需要打开办公文件。如果能够将物理位置和认证授权

结合起来，只有当用户在特定的物理环境中时才能通过认证或者获得授权，这一方面对合法用户的影响较小——因为用户本来就只有在特定的物理环境中时才需要登录或获取授权，另一方面又大大提高了安全性——攻击者在特定的物理环境外无法登录或者获得授权。在认证授权过程中引入物理位置的方式主要有以下三种：利用随机数发射器、利用全球定位系统和利用周围环境信息。

Ahren Studer 和 Adrian Perrig 提出了一种结合物理位置的文件加解密系统[22]：文件加密存储，用户只有在特定的物理位置附近时才能解密。系统要求在特定的物理位置上安装一台随机数发射器，不断地向周围发送随机数，用户只有获取到随机数才能计算出文件的解密密钥，从而解密文件。作者针对家用和企业用这两种应用场景分别提出了各自的解密密钥恢复方案，对于家用场景而言，作者假设只有合法用户能够获取到随机数发射器发出的随机数；而对于企业用场景而言，作者假设非法用户也能获取到随机数发射器发出的随机数。

在使用家用方案恢复解密密钥前，随机数发射器需要生成一对 RSA 公私钥对*(N, e, d)*，并将公钥分发给需要恢复解密密钥的设备，同时需要恢复解密密钥的设备需要产生并在本地存储一个密钥 *k*。当需要恢复解密密钥时，设备首先通过安全信道获取到发射器发射的随机数 *m*（*m* 的长度仅为 20 到 30 位）。接下来，设备需要产生一个和 *N* 互素的随机数 *R*（*R* 的长度远远大于 *m*，能够抵抗穷举攻击），并计算

$$b = R^e \cdot k (\mathrm{mod}\, N) \quad (2.17)$$

b 用 *m* 加密后发送给发射器。发射器收到后，使用 *m* 解密得到明文 *b*。发射器对 *b* 签名后将签名结果

$$sigma = b^d (\mathrm{mod}\, N) \quad (2.18)$$

返回给设备。设备根据下述公式计算解密密钥 *K*

$$K = sigma \cdot R^{-1} = b^d \cdot R^{-1} = (R^e \cdot k)^d \cdot R^{-1} = R^{ed} \cdot k^d \cdot R^{-1} = R \cdot k^d \cdot R^{-1} = k^d (\mathrm{mod}\, N) . \quad (2.19)$$

因此，无论 *m* 和 *R* 如何变化，解密密钥都为 $k^d (\mathrm{mod}\, N)$。

对于企业场景而言，作者在假设非法用户也能获取到随机数发射器发出的随机数的同时还假设企业对每台需要恢复解密密钥的设备分配唯一序列号，并维护所有设备的序列号列表，一旦某台设备遗失或被盗，企业会及时地从序列号列表

中删除该台设备的序列号。在使用企业用方案恢复解密密钥前，随机数发射器和需要恢复解密密钥的设备都需要产生并本地存储一个密钥。当需要恢复解密密钥时，设备首先获取到发射器发射的随机数 m（m 同样仅为 20 到 30 位）。接下来，设备将自己的密钥 k 和 m 异或后通过 TLS 安全信道发送给发射器。发射器利用设备的序列号判断设备是否仍在正常使用中，若是，则恢复设备的密钥 k。因为发射器也知道 m，所以只要再次异或 m 就能去掉 m 的影响。然后按照下式计算解密密钥

$$K = HMAC_{Key}(ID \| k) \quad (2.20)$$

其中 Key 表示发射器自己的密钥。

然后将解密密钥 K 通过 TLS 安全信道发送给设备，设备就能解密文件了。

除此之外，有论文提出可以利用全球定位系统获取当前物理位置[23]，也有论文提出可以通过提取环境信息来获取当前物理位置，只要收集到足够多的、与指定物理位置紧密相关的环境信息，如指定物理位置附近 WLAN 的 SSID 等，就能证明用户当前确实在指定位置[24]。

尽管有多种结合物理位置的认证授权方式，目前仍未有一种可被应用于 WLAN 的认证方式。

2.3 802.11i 协议的预共享密钥模式及安全性分析

本设计希望在使用静态口令鉴别移动终端的 802.11i 协议的预共享密钥模式的基础上，引入动态口令机制，实现对不同类型用户的细粒度授权管理。事实上，由于口令简单易猜测、且长期不变，故而暴力穷举攻击被认为是行之有效的攻击方式之一。目前已有多个提高 802.11i 协议的预共享密钥模式的安全性的解决方案，这些方案或多或少都引入了动态口令机制，因此也可以被借鉴到本设计中。接下来，本文先简单介绍下 802.11i 协议的预共享密钥模式，总结下针对该协议的攻击，主要是暴力穷举攻击，最后详细介绍下其他学者提出的 WLAN 动态口令方案。

2.3.1 协议概述

首先 AP 和移动终端需要共享相同的口令，通常 AP 的口令由管理员设置，而移动终端的口令由用户输入。口令通常为 8-20 个 ASCII 字符构成的字符串，

最短 8 字符，最长 63 字符。然后 AP 和移动终端根据下式计算预共享密钥 *PSK*：

$$PSK = PBKDF2(Passphrase, SSID, SSID\ Length, 4096, 256) \quad (2.21)$$

PBKDF2 是一个伪随机数生成函数，其中参数 *Passphrase* 是字符串形式的口令，*SSID* 是字节串形式的 SSID，作为盐值，4096 表示迭代 4096 次，而 256 则指定输出长度为 256 位，即预共享密钥的长度固定为 32 字节。

802.11i 协议的核心是 AP 和移动终端的四次握手，通过四次握手，AP 和移动终端完成双向鉴别，产生后续通信的会话密钥。四次握手的前提是 AP 和移动终端共享相同的成对主密钥 *PMK*，在预共享密钥模式中，

$$PMK = PSK \quad (2.22)$$

四次握手由 AP 发起，首先 AP 产生一个随机数 *ANonce*，发送给移动终端。移动终端收到 *ANonce* 后，也产生一个随机数 *SNonce*。移动终端使用 *PMK*、AP 和移动终端的 MAC 地址、以及两个随机数 *ANonce* 和 *SNonce* 计算成对传输密钥 *PTK*：

$$\begin{aligned} PTK &= PRF - X(PMK, "Pairwise\ \ key\ expansion", Min(AA,\ SPA) \\ &||Max(AA,\ SPA)||Min(\ ANonce,\ SNonce)||M\ ax(ANonce,\ SNonce)) \end{aligned} \quad (2.23)$$

其中 *PRF* 是一个伪随机数生成函数，*X* 指定输出长度，由加密方式（CCMP 和 TKIP）决定，*AA* 表示 AP 的 MAC 地址，*SPA* 是移动终端的 MAC 地址。

PTK 按顺序分割为三部分，分别为用以生成消息鉴别码的密钥确认密钥 KCK、加密组密钥 GTK 的密钥加密密钥 KEK、以及加密后续通信数据的临时密钥 TK，各密钥的长度由加密方式决定。此时移动终端可以将 SNonce 发送给 AP，握手包附带完整性校验码 MIC。AP 收到 SNonce 后也能生成 KCK、KEK 和 TK，使用 KCK 验证 MIC 的正确性，如果不正确，则对移动终端的认证不通过，协议终止；如果正确，则继续协议。AP 发送 GTK 给移动终端，GTK 用 KEK 加密，握手包同样附带完整性校验码 MIC。移动终端收到后也需要验证 MIC 的正确性，如果不正确，则对 AP 的认证不通过，协议终止；如果正确，则安装 PTK 和 GTK，发送确认包给 AP，确认包同样附带完整性校验码 MIC。AP 收到确认包后验证 MIC 的正确性，安装 PTK，四次握手结束，双向认证完成。整个协议如图 2.1 所示[5, 25, 26]：

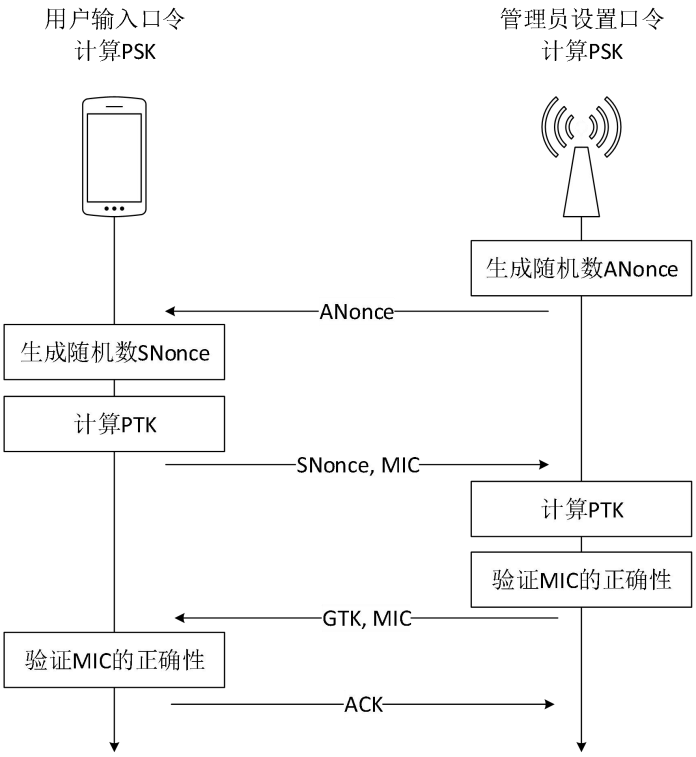

图 2.1 802.11i 协议的预共享密钥模式流程图

Figure 2.1 Flow Chart of the 802.11i Protocol in PSK Mode

2.3.2 攻击类型

攻击者攻击 802.11i 协议的预共享密钥模式主要有三个靶点：口令、四次握手过程和加密方式。尽管有不少针对后两者的攻击，但是能够造成的影响仍然比较有限[27, 28]。802.11i 协议被确立为国际标准这么多年以来，结合 CCMP 加密方式的四次握手协议仍然被认为是安全的认证加密协议。相对的，口令越来越成为 WLAN 安全的短板。因为口令过于简单，并且长时间保持不变，很容易遭到猜测攻击。随着计算能力的提高、计算资源的增多，实施一次穷举攻击所需的时间和金钱代价越来越小，穷举攻击破解 WLAN 逐渐成为现实的可能，被认为是行之有效的攻击方法。穷举攻击即尝试所有可能的口令，直到找到正确的口令为止。考虑到人们通常会设置诸如中文名全拼、手机号、生日等字符串作为口令[29]，攻击者完全可以实施字典攻击[30]，即构建常用口令列表，并逐个测试列表内的潜在口令，直至找到正确口令或者穷举完列表内全部口令为止。通过实施字典攻击，攻击者可以在合理的时间范围内以较大的概率得到正确的口令。

针对 WLAN 口令的穷举攻击主要可以分为两类：在线方式和离线方式。在线方式是指攻击者使用计算机模拟真实用户的行为：猜测口令并尝试接入 WLAN。若计算机成功与 AP 完成四次握手，说明当前测试口令是正确口令，否则尝试其他口令。Omar Nakhila 等人提出了并行在线字典攻击[31]：攻击者创建多台虚拟移动终端，每台移动终端负责一部分口令，各自独立并行地与 AP 进行四次握手，直到其中任意一台移动终端成功接入 WLAN 为止。通过引入并发机制，攻击效率得到了大幅提高（据称是未引入并发机制的攻击效率的 100 倍）。在线方式攻击者需要和 AP 进行四次握手，攻击效率受 AP 并发性能影响，且容易被 AP 检测到，离线方式则不需要和 AP 交互。离线方式是指攻击者捕获合法移动终端和 AP 发出的握手包，使用待测口令按照 802.11i 协议的要求独立地计算握手包的 MIC 值，并将其和握手包附带的 MIC 值进行比对，若两者相同，则说明当前测试口令是正确口令，否则尝试其他口令。离线方式可以再分为两类：主动方式和被动方式。被动方式下，攻击者单纯地侦听 WLAN 网络，直到有新来的移动终端接入 WLAN。由于不确定什么时候会有新移动终端想要接入 WLAN，攻击者可能需要等待较长时间。主动方式下，攻击者会主动让当前已连接的移动终端下线，迫使这些移动终端重新接入 WLAN。主动方式分为三个阶段：解认证、抓包、破解[32]。离线攻击的核心是计算和比对 MIC 值，为了提高计算效率，有的攻击者利用了 GPU 这一高性能的计算器件[33]，有的攻击者通过分布式计算来整合零散的计算资源[34]，有的攻击者充分利用了云端资源[35]，等等。Thomas Roth 宣称他使用亚马逊的 EC2 云计算平台暴力破解了 WLAN 口令，破解代价为每口令 1.68 美元。

2.3.3 解决方案

为了增强口令的安全性，管理员通常会设置复杂的口令并经常更换口令。提升口令复杂度在一定程度上能够降低攻击者猜测到口令的可能性，但是无法避免攻击者通过其他渠道获取口令，同时也增加了分发口令的难度。而频繁更换口令既增加了管理员的工作——管理员需要经常向合法用户分发新口令，也给合法用户带来了不必要的麻烦——经常更换已保存的口令，这对于大型 WLAN 来说是不切实际的。

为了从技术上提高基于口令的 WLAN 认证协议的安全性，有学者在 WLAN

口令中提出引入动态口令机制。Lei Zhang 等人提出了一种被称为 G-WPA-PSK 的 WLAN 动态口令方案[36]。首先 AP 和所有的移动终端约定一个密钥发生器，该密钥发生器能够根据不同的随机数种子生成不同的预共享密钥。每隔一段时间，AP 就会向所有的移动终端发送一个随机数种子，移动终端收到随机数种子后需要返回确认信息给 AP。当收到所有移动终端的确认包后，AP 会重置其与所有移动终端的连接，当移动终端再次与 AP 建立连接时，其使用的预共享密钥就是由之前收到的随机数种子导出的预共享密钥。通过定期更新预共享密钥，WLAN 安全性得到了一定程度的提高。

Xiaona Liao 提出可以在四次握手过程中引入 Diffie-Hellman 密钥协商协议来提高 WLAN 安全性[37]。AP 随机产生大素数 p，整数 a（a 是 p 的一个原根），以及任意整数 m，计算

$$AN1 = a^m \pmod{p} \quad (2.24)$$

并将 *AN1*、a 和 p 代替 *ANonce* 发送给移动终端。移动终端生成任意整数 M，计算

$$SN1 = a^M \pmod{p} \quad (2.25)$$

然后将 *SN1* 和 M 代替 *SNonce* 发送给 AP。之后 AP 和移动终端计算

$$C = a^{Mm} = AN1^M = SN1^m \pmod{p} \quad (2.26)$$

移动终端和 AP 使用 C 代替原来的 ANonce 和 SNonce 计算 PTK。

由于 M 和 m 是保密的，因此原来公开的 ANonce 和 SNonce 也成为了保密的参数，攻击者就无法针对 WLAN 口令实施暴力穷举攻击。并且攻击者即使知道口令也无法计算 PTK，无法解密移动终端和 AP 后续通信过程中传输的数据，WLAN 的安全性得到了一定程度的提高。

Xiaona Liao 同时提出了一种区分用户的 WLAN 口令方案[37]。用户得到口令的同时得到一个专有序列号。在四次握手的过程中，PMK 由预共享密钥和专用序列号异或得到。攻击者在不知道专用序列号的前提下，无法与 AP 完成四次握手，伪装成合法用户接入 WLAN，也无法解密合法用户的移动终端和 AP 间的通信数据。由于不同用户的专用序列号不同，不同用户的 PMK 也不同，合法用户也无法解密其他用户的移动终端和 AP 间的通信数据。此种方式相当于为每个

用户创建了用户账户，维护成本较高。

2.3.4 小结

在现有的 WLAN 动态口令方案中，G-WPA-PSK 的解决方案并不能实现细粒度的用户授权管理，所有知道原口令的移动终端都能够获得新口令，无法通过口令更新筛选掉没有或者失去 WLAN 访问权的非法用户。在四次握手过程中引入 Diffie-Hellman 密钥协商协议并没有改变 WLAN 通过简单静态口令来鉴别用户的本质，仍无法实现细粒度的用户授权管理。除此之外，Diffie-Hellman 密钥协商协议由于涉及到大整数计算，会大大降低接入效率。而在区分用户的 WLAN 口令方案中，认证双方需要事先共享一对专用序列号，不同用户与 AP 共享不同专用序列号，相当于为每个用户创建了用户账户。而且该口令方案仍属于静态口令方案，仅仅是提高了口令复杂度，无法实现对临时用户的动态授权和撤销。因此，尽管能在一定程度上提高安全性，现有的 WLAN 动态口令方案仍然不能实现在不提高管理员和用户操作负担的前提下实现细粒度的用户授权管理。

2.4 本章总结

综上所述，目前已有的 WLAN 动态口令方案并不能在兼顾管理员和用户操作体验的同时实现细粒度的用户授权管理、实现临时用户 WLAN 访问权的动态授予和撤销。而其他的动态口令机制通常要求为每个用户创建用户账户、分配不同的口令，如果需要创建用户账户，已有的 WPA-EAP 认证协议足以提供健壮安全的 WLAN 网络。同时有的动态口令机制需要认证双方多次交互以产生口令，而 AP 和移动终端并没有这样的条件；有的动态口令机制无法通过口令更新筛选掉非法用户；而有的动态口令机制使用次数有限。总之不能直接应用于 AP 和移动终端上。而结合物理认证可以实现认证双方独立地更新口令，同时在口令更新过程中筛选掉失去访问权地用户，并且可以可以无限制地更新口令。但是结合物理位置认证的认证授权方式目前尚未有应用于 WLAN 认证的先例。

第 3 章 基于物理访问控制的 WLAN 动态口令系统方案

3.1 威胁模型

本设计提出的基于物理访问控制的 WLAN 动态口令系统方案有如下假设：

（1）存在受物理访问控制保护的区域，用户只有通过物理认证才能进入这些区域（即受控物理环境）。

根据方案，WLAN 口令按照设定的时间间隔自动更新。WLAN 口令更新后，用户只有进入受控物理环境才能获得新口令，从而再次接入 WLAN。方案针对的是有大量外部人员频繁访问的大型 WLAN，部署此类 WLAN 的机构通常会对其全部或者部分工作场所设置物理访问控制，防止非法人员进入。例如有些机构会在入口处设置闸机，内部员工或外来访客只有刷门禁卡后才能进入；有些机构会在入口处安排门卫，要求人员在进入工作场所时向门卫出示员工卡或访客卡；有些机构会在入口或者其他必经通道的门上设置机械锁或电子锁，员工或访客只有用钥匙开锁、刷门禁卡或者输入口令后才能开门；等等。因此，这一假设是符合实际且容易实现的。为了方便合法用户更新口令，受控物理环境最好是合法用户经常经过的区域，这样合法用户就不必为了更新口令而特地跑去某一位置。

（2）用户愿意安装接入 WLAN 的应用。

移动终端自带的 WLAN 接入应用仅支持静态口令。为了让移动终端能够支持动态口令，在对用户透明的情况下自动更新 WLAN 口令，用户需要安装定制的 WLAN 接入应用。应用的体积很小，相信用户愿意使用移动数据流量下载并安装定制应用以接入访客 WLAN。应用需要操作系统的管理员权限，对于现在的移动操作系统而言，获得管理员权限并非是一件困难的事。

（3）没有除了暴力穷举攻击以外的其他攻击。

方案假设攻击者只能实施暴力穷举攻击，攻击者可以尝试所有可能的 WLAN 口令，同时假设口令更新周期小于攻击者找到正确口令所需的时间，亦即攻击者无法在口令有效期内找到正确口令，暴力穷举攻击会失效。除了暴力穷举攻击外，方案假设攻击者不能实施其他攻击，例如攻击者不能欺骗用户安装恶意应用等。

3.2 基本内容

本设计提出基于物理访问控制的 WLAN 动态口令系统方案的目的是为访客 WLAN 提供更细粒度的授权管理，实现访客 WLAN 访问权的动态授予和撤销，同时提高 WLAN 系统的安全性。方案的核心是在现有基于口令的 802.11i 协议的基础上引入动态口令机制，移动终端只有和 AP 共享相同的 WLAN 口令才能正常接入 WLAN，而 AP 的口令会按照设定的时间间隔自动更新。更新过程引入物理认证因素，用户只有通过物理认证、进入设定的受控物理环境，其持有的移动终端才能与 AP 同步更新 WLAN 口令，从而再次接入 WLAN，失去物理认证权限的用户则会被拒绝再次接入 WLAN。

图 3.1 显示了基于物理访问控制的 WLAN 动态口令系统的整体架构。系统分为三部分：能够独立更新 WLAN 口令的主 AP、需要和主 AP 交互才能更新 WLAN 口令的从 AP、以及能够根据具体情形自动更新 WLAN 口令的移动终端。

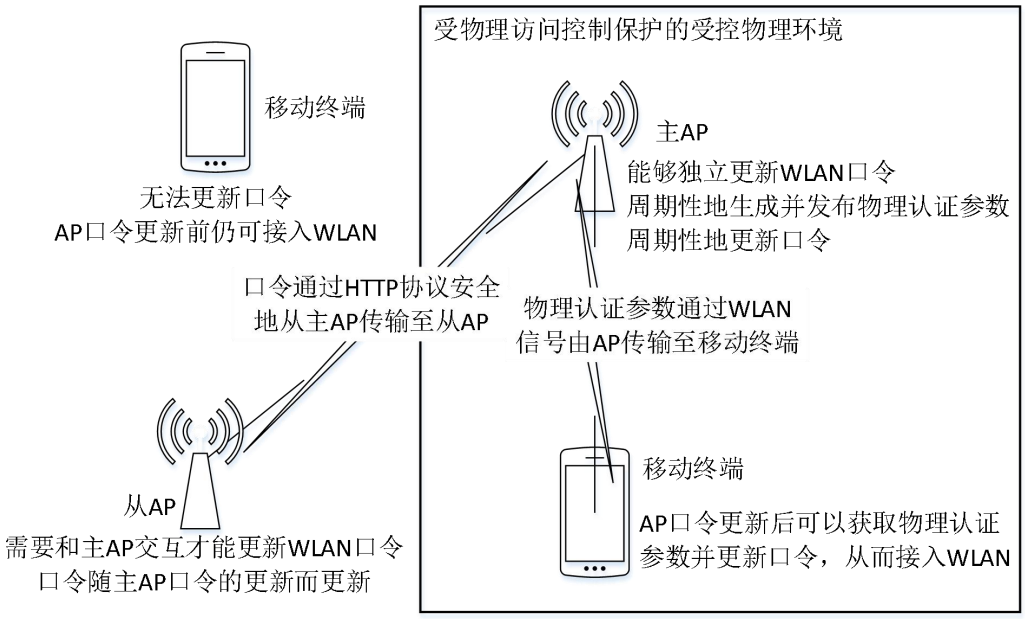

图 3.1 基于物理访问控制的 WLAN 动态口令系统整体架构

Figure 3.1 Overview of the WLAN System with Location Related Dynamic Password

其中，主 AP 会定期生成并在设定的、受物理访问控制保护的物理环境中广播某一随机数，称为物理认证参数，该参数是计算新口令所必需的参数。主 AP

能够自主生成物理认证参数，因而能够独立更新口令。从 AP 则无法自主生成物理认证参数，因而只能通过向主 AP 请求物理认证参数以计算新口令、或者直接请求新口令的方式更新自身口令。而移动终端则会根据自身与 AP 的口令是否一致来决定是使用自身口令接入 WLAN、还是计算新口令并使用新口令接入 WLAN。

图 3.2 的第二个箭头显示的是主 AP 更新 WLAN 口令的工作流程。如图 3.2 所示，管理员需要预先设置好主 AP 的初始口令，此后主 AP 便会按照设定的时间间隔自动更新口令，当需要更新口令时，主 AP 将完成下述工作：

（1）生成物理认证参数；

（2）计算新口令；

（3）将物理认证参数借助 WLAN 信号发布到受控物理环境中；

（4）使用新口令借助 802.11i 协议的预共享密钥模式鉴别移动终端；

（5）将新口令通过安全信道（如专用有线局域网）借助 HTTP 协议传输至从 AP。

主 AP 周期性地执行上述操作，不断更新自身口令。

图 3.2 的第三个箭头显示的是从 AP 更新 WLAN 口令的工作流程。如图 3.2 所示，管理员不需要专门设置从 AP 的初始口令，从 AP 启动时即向主 AP 请求当前口令以鉴别移动终端。此后从 AP 会在自身口令失效时自动更新口令，当需要更新口令时，从 AP 将完成下述工作：

（1）通过安全信道借助 HTTP 协议从主 AP 处获取新口令作为自身口令；

（2）使用新口令借助 802.11i 协议的预共享密钥模式鉴别移动终端。

从 AP 周期性地执行上述操作，始终保持自身口令与主 AP 的口令相同，亦即保证同一 WLAN 系统内的所有 AP 的口令相同。

图 3.2 的第一个箭头显示的是移动终端更新 WLAN 口令的工作流程。如图 3.2 所示，用户需要预先通过带外方式获得初始口令，并将初始口令输入其持有的移动终端。在 AP 口令更新前，移动终端即可使用自身口令接入 WLAN。而当移动终端需要且能够更新口令时，移动终端将完成以下工作：

（1）从主 AP 发出的无线信号中提取出物理认证参数；

（2）使用和主 AP 相同的算法计算得到新口令；

（3）尝试使用新口令借助 802.11i 协议的预共享密钥模式接入 WLAN；

（4）若接入成功则接受新口令作为自身口令，口令更新完成；

（5）否则拒绝新口令作为自身口令，仍保留原口令，口令更新失败，口令状态回滚到更新前状态。

口令更新完成后，在下一次口令更新前，移动终端均可使用新口令接入 WLAN。

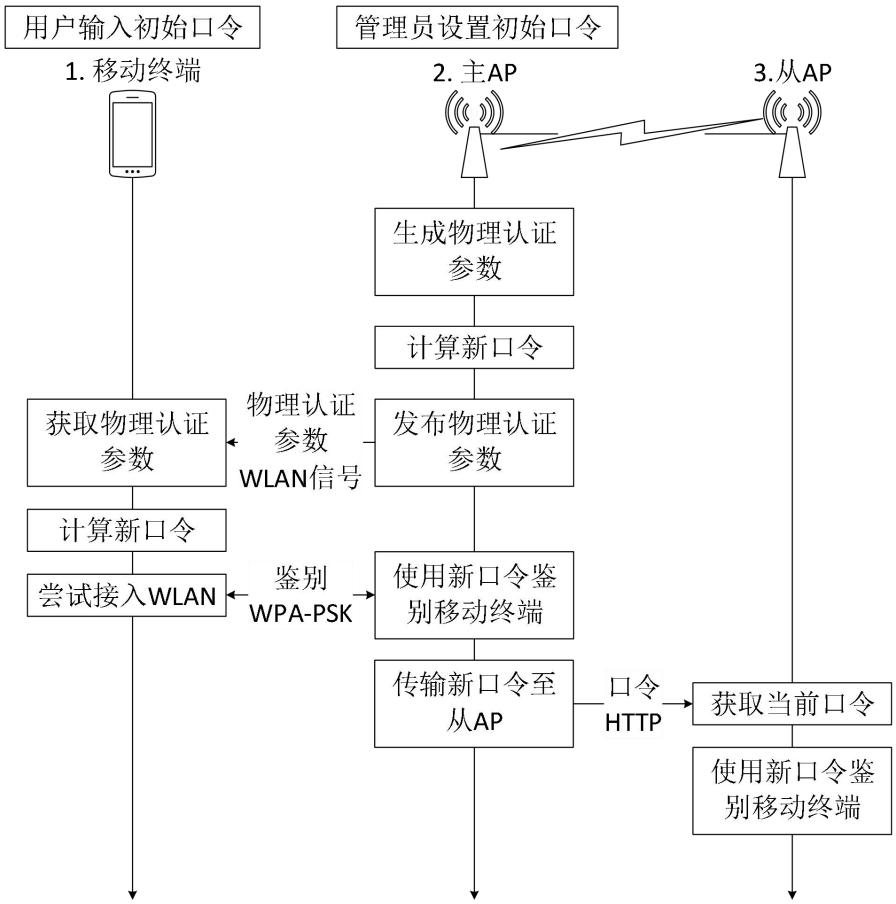

图 3.2 基于物理访问控制的 WLAN 动态口令系统工作流程

Figure 3.2 The Work Flow of the WLAN System with Location Related Dynamic Passwordd

3.3 方案细节

基于物理访问控制的 WLAN 动态口令系统方案的核心包含以下五个部分：初始口令的设置和分发、口令更新周期的设置、物理认证参数的生成和发布、以

及 AP 和移动终端的口令更新。

3.3.1 初始口令的设置和分发

初始口令的设置和分发与静态口令基本相同，口令存储于配置文件中。管理员需要为主 AP 设置初始口令，即将初始口令写入配置文件。主 AP 启动后即从配置文件中读取初始口令，考虑到主 AP 启动时，初始口令就可能已经失效，主 AP 可能需要将口令更新至启动时。从 AP 本地不存储口令，管理员也不需要设置从 AP 的初始口令。从 AP 启动后，直接从主 AP 处获取启动时的口令。用户询问口令时，管理员将 AP 当前时间的口令通过带外方式告知用户，用户也可以将当前时间的口令通过带外方式告知其他用户。移动终端获得当前时间的口令（即移动终端的初始口令）后即将初始口令写入配置文件。有了初始口令，移动终端便可在当前口令更新周期内自由访问 WLAN。

移动终端和 AP 共享相同的口令后，移动终端借助 802.11i 协议的预共享密钥模式接入 WLAN。在 802.11i 协议的预共享密钥模式中，有两个与口令相关的概念：密码短语和预共享密钥。密码短语就是通常人们在接入 WLAN 过程中键入的短文本字符串，ASCII 码格式，长度在 8 至 63 之间。而预共享密钥则是一个 32 字节的随机数，由密码短语经过伪随机数生成函数导出。预共享密钥直接作为后续 AP 和移动终端四次握手所需的成对主密钥。在用户接入 WLAN 的过程中，802.11i 协议既允许用户输入密码短语，也允许用户直接输入预共享密钥。

在本方案中，口令是 32 字节的随机数，作为预共享密钥直接进行 802.11i 四次握手。由于口令太长，不方便用户键入，因此口令不可能按照现有的分发方式分发。考虑到现在的移动终端都配备了摄像头，口令可以以二维码的形式分发。管理员或者知道口令的用户将当前口令以二维码的形式显示在屏幕上或者打印到纸上，移动终端扫描二维码后得到当前口令。鉴于当前二维码的广泛应用，相信这种方式能够得到用户的认可。

3.3.2 口令更新周期的设置

与主 AP 的初始口令相同，口令更新周期由管理员设置，存储于主 AP 的配置文件中。每个口令都附带失效时间，失效时间与对应口令一起存储于配置文件中（初始口令的失效时间由管理员设置）。当口令失效时，主 AP 更新口令，按照口令更新周期确定新口令的失效时间，并将新口令及其失效时间写入配置文件。

同时，如果从 AP 向主 AP 请求当前口令，主 AP 应当将当前口令及其失效时间一同返回从 AP，这样从 AP 就知道当前口令的失效时间，主从 AP 会在相同的时点更新口令。

口令的更新周期不是固定不变的，管理员完全可以根据需要随时调整更新周期。平常管理员可以设置一个较长的更新周期，而在单位遇有重大事项的时候设置一个较短的更新周期。如果管理员需要调整口令更新周期，可以修改主 AP 的配置文件，然后重启主 AP。重启后，主 AP 会从配置文件中读取重启前的口令及其失效时间，以及新的口令更新周期。由于每次更新口令时，主 AP 都会将新口令及其失效时间写入配置文件，因此重启前后主 AP 的口令及其失效时间并不发生变化。此外，从 AP 无论是在主 AP 重启前还是重启后向主 AP 请求口令，都能够得到相同的口令和失效时间，因此重启主 AP 也不会对从 AP 产生任何影响。综上所述，即使主 AP 重启，主从 AP 均可在原有口令的基础上继续更新，并且失效时间也不变，因此对移动终端也不会有任何影响。但是一旦口令失效，主 AP 就会应用新的口令更新周期，整个系统的口令更新周期也就都发生了变化。

3.3.3 物理认证参数的产生和发布

物理认证参数是一串长随机数，由主 AP 按照设定的口令更新周期生成并发布，移动终端只有在设定的受控物理环境中才能获得。AP 每隔固定时间间隔会广播信标帧，告知移动终端 WLAN 的存在。信标帧存在供应商自定义字段，允许 AP 发布自定义内容的数据，物理认证参数可以通过该字段发布。移动终端收到主 AP 发出的、指定 SSID 的 WLAN 信号后，从信标帧的供应商自定义字段中提取出物理认证参数，从而计算新口令。

生成物理认证参数的 AP 只能有一台，确保在一个口令更新周期内只有唯一一个物理认证参数。但是发布物理认证参数的 AP 可以有多台，只要他们能够从生成物理认证参数的 AP 处获得物理认证参数。大型 WLAN 通常需要部署多个 AP，确保 WLAN 信号对办公场所的全方位覆盖，让所有用户都能接入 WLAN，给用户提供更好的上网体验。有些 AP 的信号甚至会超出办公场所，如果这些 AP 也发布物理认证参数，那么非法用户即使不能通过物理认证，也将能够获得物理认证参数从而更新口令。因此，WLAN 必须指定其中一台或多台信号覆盖范围在受控物理环境内的 AP 发布物理认证参数，确保只有在受控物理环境内的

移动终端才能收到携带物理认证参数的 WLAN 信号，从而确保只有能够通过物理认证的合法用户才能更新口令，在 AP 口令更新后再次接入 WLAN。

当然位于受控物理环境外的移动终端也可能收到属于同一 WLAN 的信号，但是这些信号由属于同一 WLAN 的其他 AP 产生，这些 AP 不会广播物理认证参数，移动终端只能凭借已保存的口令接入 WLAN，但无法计算新口令。此时若 AP 的口令更新，移动终端将无法接入 WLAN。

3.3.4 AP 口令更新

AP 更新口令是一个周期性的行为。在设定的口令更新时点，主 AP 需要生成一个随机数作为新的物理认证参数并发布。物理认证参数一旦更新，主 AP 随即按以下算法更新口令：设原来的口令为 *P[i - 1]*，新的物理认证参数为 *O[i]*，则新口令

$$P[i] = Format(Hash(P[i-1]\ XOR\ O[i])) \quad \text{(3.1)}$$

其中，*XOR* 表示异或运算，*Hash* 表示安全杂凑函数，如 SM3、SHA256 等，*Format* 表示格式化杂凑值，将杂凑值转为格式符合要求的口令，如果选择杂凑值为 32 字节的杂凑函数，如 SM3、SHA256，则可以直接接受杂凑值作为新口令；如果杂凑值并非 32 字节，则可以通过截取前 32 字节或者剩余字节补 0 的方式作为新口令。

主 AP 生成新口令后，需要立即将新物理认证参数及新口令写入配置文件。这样，主 AP 即使因为计划或意外重启，也能继续广播重启前的物理认证参数，继续使用重启前的口令鉴别移动终端，继续向从 AP 传输重启前的口令，继续在重启前口令的基础上更新口令，保证主 AP 口令更新过程的连续性。

主 AP 更新口令后，从 AP 应当立即同步更新口令，使 WLAN 系统内的所有 AP 在相同时点使用相同口令鉴别移动终端。如果各 AP 间的口令更新有较长的一段时延，当移动终端从一个已更新口令的 AP 切换到一个尚未更新口令的 AP 时就会因为口令不一致而无法接入。考虑到主从 AP 的系统时间可能有所差别，从 AP 可以在当前口令失效前一小段时间就向主 AP 请求物理认证参数或口令，一旦发现主 AP 的物理认证参数或口令更新，从 AP 随即更新口令，这样主从 AP 就可以在尽可能短的时间间隔内全部完成口令更新。由于从 AP 启动时从主 AP 处获得启动时的口令，更新口令时也从主 AP 处获得新口令，从 AP 始终保持和

主 AP 口令的一致性。即使因为计划或意外重启，从 AP 的口令更新过程也不会受到任何影响。

口令应当通过安全信道由主 AP 传输至各从 AP。为确保只有从 AP 能够通过安全信道更新口令，新口令在传输前需要加密且经完整性保护，只有 WLAN 系统内的 AP 拥有解密密钥和完整性校验密钥，从而获得明文的口令。

AP 一旦获得新口令，即使用新口令鉴别移动终端，此时移动终端只有和 AP 同步更新口令才能接入 WLAN。

3.3.5 移动终端口令更新

移动终端每隔固定时间间隔会扫描附近的 WLAN，更新附近 WLAN 的列表。移动终端收到指定 SSID 的 WLAN 信号后，首先判断是否能够继续使用当前已有的口令接入 WLAN。如果当前时间仍在当前口令的有效期内，即当前口令仍有效，移动终端就能直接使用当前口令接入 WLAN。如果发现当前口令已失效，但口令仅更新过一次，即当前时间已进入下一口令更新周期，移动终端就会试图从 WLAN 信标帧中提取物理认证参数。只有解析得到新的物理认证参数，才能使用和 AP 相同的算法计算出新口令，从而再次接入 WLAN；如果无法获取新的物理认证参数，移动终端需要提醒用户通过物理认证，进入受控物理环境。如果发现当前口令不仅已失效，而且已更新过不止一次，移动终端将无法将口令更新至当前时间，只能提醒用户无法接入 WLAN。在更新口令过程中，移动终端如果使用新口令成功接入 WLAN，那么应当立即将新口令写入配置文件。这样，移动终端即使因为计划或意外重启，也能继续使用新口令接入 WLAN，继续在新口令的基础上更新口令，保证口令更新过程的连续性。

在接入 WLAN 的过程中，移动终端需要通过某种方式确定：以自己的口令为基准，AP 的口令是否更新过，如果是，更新过几次。如果移动终端的口令和 AP 的口令相同，移动终端能够直接接入 WLAN；如果较之移动终端的口令，AP 的口令仅更新过一次，移动终端尚能够通过获取物理认证参数、计算新口令，从而使用新口令来接入 WLAN；其他情况下，除非从管理员或者知道口令的用户处重新获取初始口令，移动终端将无法再次接入 WLAN。

考虑到移动终端和 AP 间的系统时间可能不同步，并且口令的更新周期不固定，方案通过引入序列号来解决移动终端和 AP 的口令同步问题，WLAN 系统为

每个口令分配一个序列号，序列号从零开始，每次更新加一。在 WLAN 系统初始化过程中，管理员为主 AP 设置初始口令，并分配序列号为零。此后主 AP 每次更新口令，都将序列号加一。在口令分发过程中，序列号连同口令一起被发送给需要接入 WLAN 的用户。移动终端每次更新口令，也将序列号加一。移动终端通过序列号来判断自身口令是否和 AP 口令一致，是否需要更新自身口令。WLAN 系统内所有 AP 都广播自身口令的序列号，和物理认证参数一样，序列号也通过 WLAN 信标帧的供应商自定义字段广播。移动终端收到 WLAN 信号后，首先提取序列号，将之和自身口令的序列号比较，如果相同，说明移动终端和 AP 的口令相同，移动终端可以直接接入 WLAN；如果 AP 广播的序列号比移动终端自身口令的序列号大一，移动终端可以通过更新口令来接入 WLAN；否则移动终端将无法通过更新口令来接入 WLAN。

这种方式并没有移动终端系统时间的参与，所以即使移动终端和 AP 的系统时间不同步，也不影响移动终端接入 WLAN。如果 AP 的系统时间早于移动终端的系统时间，AP 更新口令后，发布的序列号会加一，移动终端就知道 AP 更新了口令，它也跟着更新口令；如果 AP 的系统时间晚于移动终端的系统时间，AP 迟迟不更新口令，但是 AP 发布的序列号也没变，移动终端就知道 AP 没有更新口令，它也没必要更新口令。

3.4 安全性分析

引入动态口令机制能够提高 WLAN 口令的安全性：

一方面，WLAN 口令由原来的 8 至 20 个 ASCII 字符构成的字符串变为 32 字节的随机数，口令空间大幅度扩张，攻击者通过实施各种攻击方式，例如暴力穷举攻击，以获取口令所需的时间也就大幅度延长。

另一方面，由于口令每隔一段时间就会变化，攻击者攻击口令的时间窗口就变小了。攻击者可能还未来得及猜到正确口令，口令就已经发生了变化。并且对于攻击者而言，知道旧口令无助于其得到新口令，因为在计算新口令的过程中引入了完全随机的物理认证参数，而该参数是攻击者无法获得的，并且可能比新口令更复杂。即使攻击者知道旧口令，并且想利用知道的旧口令，也需要穷举物理认证参数，而穷举物理认证参数的时间复杂度反而会超过穷举新口令的时间复杂度。因此对于攻击者而言，即使知道了旧口令，也会选择直接穷举新口令，犹如

不知道旧口令一样。换言之，一旦 WLAN 系统更新了口令，攻击者的攻击行为就必须从头开始，而之前的攻击成果则付之东流。只要口令更新间隔小于攻击者成功实施一次穷举攻击所需的最短时间，口令就能免受穷举攻击的威胁。

只要攻击者不知道 WLAN 口令，就不能通过合法 AP 的认证，从而接入 WLAN，影响合法用户的上网体验，也不能伪装成合法 AP 欺骗移动终端与之建立连接，截获用户的网络流量，或者捕捉并解密移动终端和 AP 的通信数据，获取用户的隐私信息。退一步讲，即使攻击者通过其他方式获得了某一更新周期的 WLAN 口令，其所能造成的消极影响也是有限的。一旦 WLAN 口令更新，攻击者就不能再实施恶意行为。

与此同时，方案本身并不会造成 WLAN 口令泄露：

首先，口令的分发过程不会造成口令泄露，因为移动终端仍采用和原来相同的带外方式获取初始口令。

其次，口令的存储方式不会造成口令的泄露。因为口令的存储方式与原来一样，都存储于本地文件中。AP 的口令由管理员设置且只有管理员可以访问，而移动终端的口令只有获得了管理员权限的应用可访问。

再次，认证过程不会造成口令的泄露。因为方案仅在原有基于口令的 WLAN 认证协议的基础上加入了动态口令机制，并没有改变认证协议本身。

最后，口令更新过程也不会造成口令泄露。在口令更新的过程中，认证双方无需在公开信道上多次交互，认证双方各自独立地计算出相同的口令，移动终端仅需在单位内部设定的、受物理访问控制保护的、攻击者不可达的物理环境中获得物理认证参数即可更新口令。

此外，攻击者无法伪装成主 AP 欺骗移动终端接受非法的新口令致使移动终端无法接入合法 AP。攻击者当然可以设立恶意的主 AP 并随意广播物理认证参数，并且移动终端无法直接验证物理认证参数的正确性，不能确信物理认证参数一定来自主 AP，且在传输过程中没有出错或被篡改。因此，移动终端在接收到物理认证参数后，只能姑且认为物理认证参数是正确的，然后计算新口令。之后移动终端需要尝试使用新口令接入 WLAN：如果移动终端成功接入 WLAN，证明新口令和收到的物理认证参数确实是正确的，移动终端才能更新口令，同时将新口令写入配置文件；如果移动终端无法完成和 AP 的四次握手，说明新口令是

错误的，此时移动终端应当清除新口令，好像从未获得过此次物理认证参数一样，继续等待更新口令。由于攻击者并不知道旧口令，也就无法得到和移动终端相同的新口令，移动终端和 AP 的四次握手就无法正常完成，移动终端最终也会拒绝这一新口令。因此攻击者就不能在不知道口令的情况下伪装成主 AP 发布错误的物理认证参数，诱使移动终端产生错误的口令，致使移动终端无法接入合法 AP。

第 4 章 基于物理访问控制的 WLAN 动态口令系统实现

4.1 开发环境和语言

本设计在办公用台式计算机上实现了基于物理访问控制的 WLAN 动态口令系统。计算机搭载四核、3.2GHz 主频、三级 6M 缓存的处理器，搭配 4G 1600MHz DDR3 内存，运行 64 位 Linux 操作系统。计算机还同时配备了两张网卡：一张网卡运行于主控模式，作为主 AP 或从 AP；另一张网卡运行于受控模式，作为移动终端。本系统的开发语言为 C/C++。

Linux 系统对无线网卡的管理结构是分级层次结构，从应用程序到无线网卡中间需要经历多层，每层执行不同的功能，其大致框架如图 4.1 所示[38, 39]。

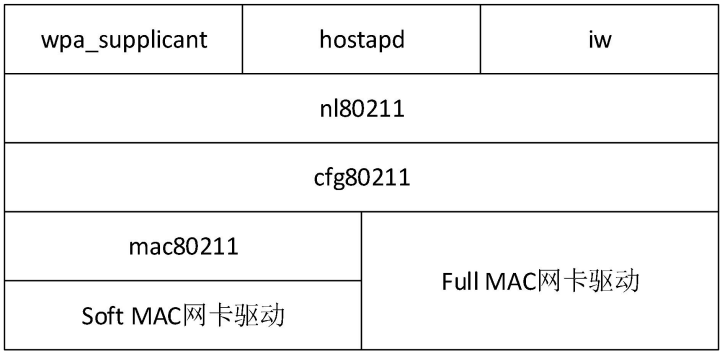

图 4.1 Linux 系统无线网卡管理栈

Figure 4.1 Management Stack of Wireless Card in Linux

无线网卡包含两部分：硬件和无线网卡驱动，操作系统无法直接控制网卡硬件，由无线网卡驱动负责信息在操作系统和网卡硬件之间的上传下达。无线网卡驱动有一个很重要的功能：管理物理层的 MAC 状态机，典型的 MAC 状态有：信标、探测、认证、解认证、关联、解关联、重关联，对应移动终端接入 WLAN 过程中的不同阶段。AP 需要定期广播信标帧，宣告 WLAN 的存在。对于不广播的 WLAN，移动终端需要发出探测请求帧，AP 返回探测响应帧。当移动终端需要接入时，移动终端先发出认证请求帧，AP 返回认证响应帧，然后移动终端发出关联请求帧，AP 返回关联响应帧，移动终端和 AP 就建立起连接了。如果移

动终端需要从一个 AP 转移到同一 WLAN 系统内的另一个 AP，移动终端就需要发出重关联请求帧，AP 返回重关联响应帧。另外移动终端也可以通过发出解认证、解关联请求帧来解除认证状态、断开连接，AP 返回相应的响应帧。因此，移动终端和 AP 的连接过程就是一次次的 MAC 状态转移过程，无线网卡需要维护 MAC 状态的转移。无线网卡驱动中管理 MAC 状态机的部分被称为 MAC 层管理实体 MLME，根据 MLME 由硬件完成还是软件完成，可以将网卡驱动分为软 MAC 和全 MAC。全 MAC 的 MLME 由硬件管理，执行效率高，但是兼容性差，不利于更新，现在全 MAC 驱动几乎没有了。而软 MAC 的 MLME 由软件管理，mac80211 是 Linux 的一个内核子系统，专门负责 MLME。mac80211 为驱动开发人员开发无线网卡驱动提供了框架，由无线网卡驱动适配 mac80211，再由 mac80211 适配上层代码，这样，凡是软 MAC 驱动的无线网卡都能在 Linux 上正常运行。由于 MLME 由软件完成，软 MAC 驱动的无线网卡的硬件成本较低，同时升级驱动可以直接通过系统更新实现，除此之外，可以对无线网卡有更加精细的控制——mac80211 允许软件产生和解析 WLAN 数据帧。因此，软 MAC 驱动是当前发展的主流。

从 mac80211 往上，就是 cfg80211 和 nl80211，两者共同实现对无线设备的配置管理。其中 cfg80211 运行于内核空间，而 nl80211 运行于用户空间，因此 nl80211 在 cfg80211 之上，而 cfg80211 又在 mac80211 之上。cfg80211 提供了一系列配置 WLAN 的 API，而 nl80211 则允许用户在用户态调用这些 API（通过和 cfg80211 交互）。nl80211 通过 netlink 机制向 cfg80211 发送特定的消息来调用 cfg80211 的 API、控制无线网卡行为。netlink 套接字是 Linux 系统中一种特殊的进程间通信，专用于用户空间和内核空间的通信。进程间传输的是一个又一个的数据帧，用户空间程序生成预先定义好结构的数据帧，发送给内核空间程序。同时由于采用了消息机制，cfg80211 和 nl80211 之间的通信不再是单向的了，而可以实现全双工通信。

WLAN 应用程序属于用户空间程序，在 nl80211 之上，通过调用 nl80211 提供的接口来操作无线网卡，例如向信标帧中加入自定义的数据、让无线网卡扫描附近的 WLAN、让移动终端接入指定的 WLAN 等。nl80211 提供了 nl80211.h 头文件，文件中定义了所有暴露给用户空间的 API 函数索引（不是函数本身），以

及这些函数所需的参数格式和定义[40]。WLAN 应用程序通过 netlink 机制，将 API 函数索引及对应参数封装到 netlink 数据帧中，发送给 cfg80211。cfg80211 解析数据帧中内容，就知道需要调用哪个函数、以及该函数的参数，完成无线网卡功能调用。Linux 系统详细的 WLAN 架构如图 4.2 所示[41]：

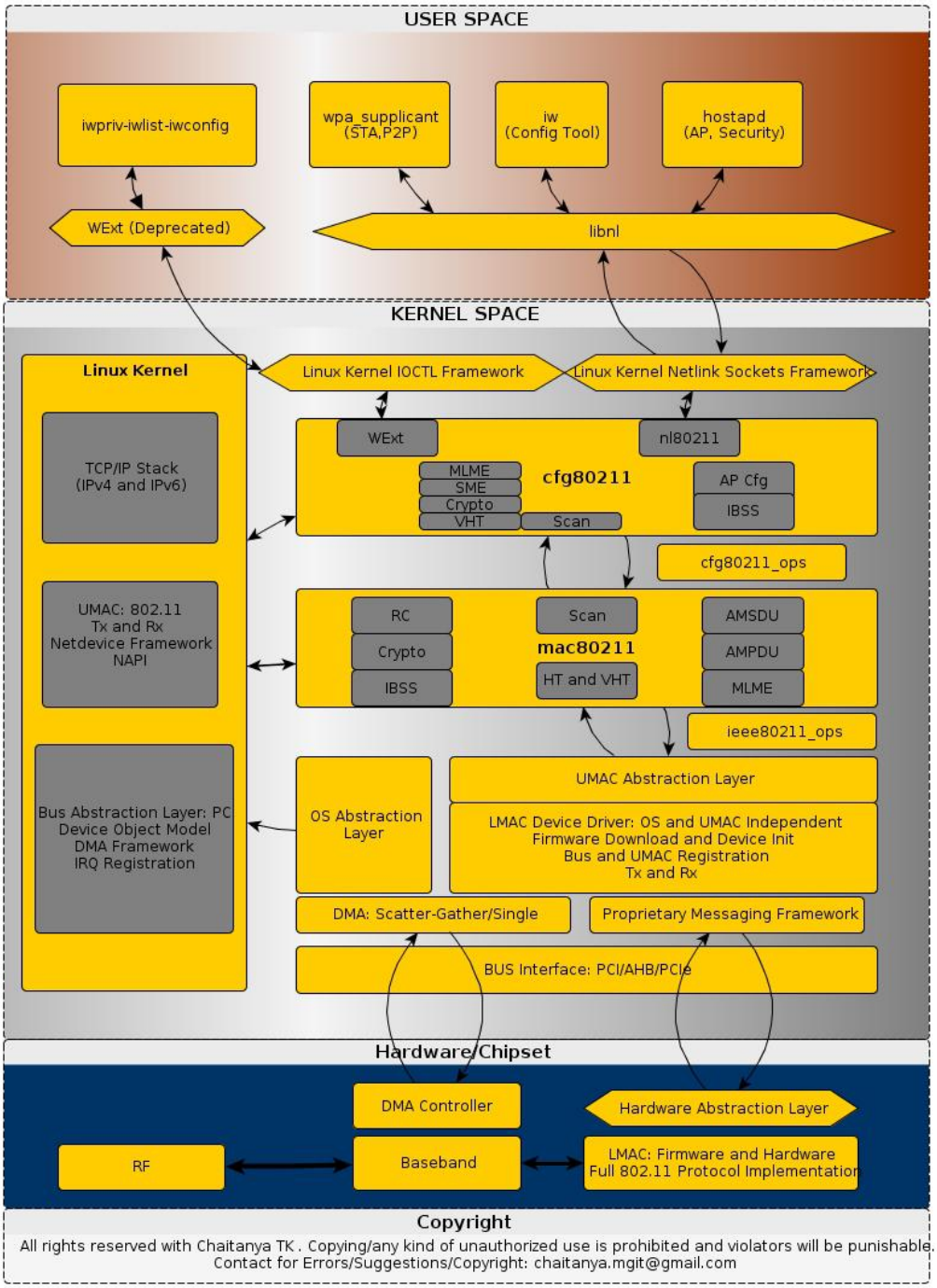

图 4.2 Linux 系统管理无线网卡的详细结构

Figure 4.2 The Detail Structure to Manage Wireless Card in Linux

hostapd 和 wpa_supplicant 分别是 Linux 系统提供的 WLAN 认证程序和客户端程序，能够运行在基于 Linux 内核的服务器操作系统、桌面操作系统、以及嵌入式操作系统上，实现无线设备之间的认证、关联和数据传输[42]。两者都以守护进程的方式运行在用户空间，同时支持 802.11i 协议的预共享密钥模式和 802.1X 认证服务器模式，以及 CCMP 和 TKIP 加密方式，还支持 802.11r、802.11w 等其他相关协议。其中，hostapd 能让无线网卡工作于主控模式，即 AP 模式，负责处理无线设备的接入请求，亦即用无线网卡开 WLAN 热点。而 wpa_supplicant 能让无线网卡处于受控模式，即移动终端模式，让无线网卡接入指定的 WLAN。除此之外，hostapd 还可以让计算机成为 RADIUS 服务器。

hostapd 用以开启 WLAN 热点，管理员使用配置文件来指定 WLAN 的各项参数，例如 SSID、口令、预共享密钥等，包括向信标帧中添加自定义数据也可以使用配置文件完成。配置文件的主要参数包括：

表 4.1 hostapd 配置文件主要参数及意义

Table 4.1 Main Parameters and their Meanings of hostapd Profile

参数名	意义
interface	用以开启 WLAN 热点的无线网卡名称
driver	无线网卡驱动类型，通常为 nl80211
ssid	SSID
auth_algs	认证阶段的认证算法：如果采用 802.11i 协议认证，则认证阶段的认证算法为空，即不认证，取值为 1
wpa	指定使用 WPA 认证：2 表示采用 WPA2 认证协议，即 802.11i 协议
wpa_key_mgmt	密钥协商算法：WPA-PSK 表示采用预共享密钥认证
rsn_pairwise	加密方式：采用 CCMP 加密方式可以得到更好的安全性
wpa_passphrase	密码短语
wpa_psk	预共享密钥
vendor_specific	供应商自定义数据
ctrl_interface	控制接口路径

配置文件路径以主函数参数的形式提供给 hostapd。hostapd 每次启动时都会解析主函数参数，读取配置文件，并据此设定 WLAN 的各项参数，开启 WLAN

热点。除此之外，hostapd 还提供控制接口，接受来自其他程序的控制命令。hostapd_cli 是与 hostapd 一同提供、用于控制 hostapd 的应用程序，它能够显示当前已接入的移动终端列表、断开指定一台或者全部已接入的移动终端的连接、解除指定一台或者全部已接入的移动终端的认证状态等。hostap_cli 和 hostapd 的工作模式属于客户端服务器模式，hostapd 始终处于运行状态，侦听控制接口，如果 hostapd_cli 发起任务请求，hostapd 就返回执行响应。两者通过 Unix 域套接字完成信息交互，ctrl_interface 即套接字文件路径，hostapd_cli 只有与 hostapd 建立套接字连接后才能控制 hostapd。

wpa_supplicant 用以接入 WLAN，WLAN 用户同样可以使用配置文件来指定希望接入的 WLAN 的各项参数，让无线网卡接入指定 WLAN。配置文件的主要参数包括：

表 4.2 wpa_supplicant 配置文件主要参数及意义

Table 4.2 Main Parameters and their Meanings of wpa_supplicant Profile

参数名	意义
ctrl_interface	控制接口路径
network	一个 WLAN 的各项参数

其中 network 可以有多个，每个 network 都代表一个希望接入的 WLAN。移动终端可以保存多个 WLAN，并根据需要接入其中任意一个。network 的主要参数包括：

表 4.3 network 的主要参数及意义

Table 4.2 Main Parameters and their Meanings of network Parameter

参数名	意义
ssid	名称
auth_alg	认证阶段的认证算法：如果采用 802.11i 协议认证，则认证阶段的认证算法为空，即不认证，取值为 OPEN
key_mgmt	密钥协商算法：WPA-PSK 表示采用预共享密钥认证
pairwise	加密方式：采用 CCMP 加密方式可以得到更好的安全性
psk	带双引号表示口令，不带双引号表示预共享密钥

wpa_supplicant 与 hostapd 略微不同的是，用以接入 WLAN 的无线网卡名称和驱动类型并非在配置文件中指定，而是与配置文件路径一起以主函数参数的形式提供给 wpa_suplicant。wpa_supplicant 每次启动时都会解析主函数参数，读取配置文件，而后让无线网卡扫描附近的 WLAN。如果无线网卡收到已保存的 WLAN 列表内某个 WLAN 的信号，wpa_supplicant 就会让无线网卡接入该 WLAN。如果能够收到多个 WLAN 的信号，wpa_supplicant 会综合考虑信号强度、优先级等各种因素选定其中一个 WLAN 接入。除了配置文件，wpa_supplicant 同样提供控制接口，接受来自其他程序的控制命令。wpa_cli 和 wpa_gui 是与 wpa_supplicant 一同提供的、用于控制 wpa_supplicant 的应用程序，其中 wpa_cli 是命令行程序，而 wpa_gui 则是图形界面程序。两者功能相同，能够显示附近 WLAN 的列表、查看当前无线网卡的 WLAN 接入状态、创建一个希望接入的 WLAN 并设置其各项参数、接入指定的 WLAN、断开与当前已接入的 WLAN 的连接等。

wpa_cli/wpa_gui 和 wpa_supplicant 的工作模式也属于客户端服务器模式，两者同样通过 Unix 域套接字完成信息交互。相比于配置文件，wpa_cli/wpa_gui 能够提供更细粒度的管理控制——能够按需地接入或者断开指定的 WLAN。

wpa_supplicant 提供了 wpa_ctrl.h 头文件，文件中定义了和 wpa_supplicant 建立连接、发送请求、断开连接等一系列函数，其中发送请求的函数声明如下：

int wpa_ctrl_request(struct wpa_ctrl *ctrl, const char *cmd, size_t cmd_len, char *reply, size_t *reply_len, void (*msg_cb)(char *msg, size_t len));

在这个函数中，参数 ctrl 表示与 wpa_supplicant 建立的套接字的句柄；cmd 是命令名称，文本字符串形式，不同的字符串会让 wpa_supplicant 执行不同的操作，例如 SCAN_RESULTS 是获取附近 WLAN 的列表（wpa_supplicant 会让无线网卡定时地扫描附近的 WLAN 并将附近 WLAN 的列表保存在缓冲区中）、SET_NETWORK 是设置希望接入的 WLAN 的各项参数、STAUTS 是显示当前的 WLAN 接入状态、等等；reply 是 wpa_supplicant 的执行结果；len 和 reply_len 分别是 cmd 和 reply 的长度；msg_cb 是回调函数，由 wpa_supplicant 执行，可以为空。wpa_ctrl_request 函数会将 cmd 命令通过 ctrl 套接字发送给 wpa_supplicant，然后接收来自 wpa_supplicant 的响应并填充 reply 缓冲区。wpa_supplicant 有专门的函数来处理 wpa_ctrl_request 的请求，此类函数被定义在文件 ctrl_iface.c 中，

增加或修改此类函数就能让 wpa_supplicant 处理开发者自定义的命令、返回开发者想要的结果。wpa_cli 就是通过调用 wpa_ctrl_request 函数来实现用户需求的：用户调用 wpa_cli 并将自己的需求以主函数参数的形式提供给 wpa_cli，wpa_cli 解析主函数参数，调用 wpa_ctrl_request 函数，根据用户的需求设置不同的 cmd 参数，最后将执行结果显示在终端上。

4.2 系统实现

基于物理访问控制的 WLAN 动态口令系统包含主 AP、从 AP 和移动终端三部分。其中：

- 主 AP 需要定期生成并在指定的受控物理环境中发布物理认证参数，同时定期更新鉴别移动终端所用的口令；
- 从 AP 需要定期更新鉴别移动终端所用的口令；
- 移动终端需要在收到指定 SSID 的 WLAN 信号后，根据具体情况使用自身口令接入 WLAN、更新口令并使用新口令接入 WLAN、提醒用户进入受控物理环境以更新 WLAN 口令、提醒用户 WLAN 口令已失效等。

下文将详细介绍这三个部分是如何实现的。

4.2.1 主 AP

如前所述，借助 hostapd，无线网卡可以开启 WLAN 热点。其中，WLAN 热点广播的自定义参数由 hostapd 配置文件的 vendor_specific 参数指定，而鉴别移动终端所用的预共享密钥则由 wpa_psk 参数指定。

为了实现主 AP 口令的自动更新，本设计实现了口令更新程序。口令更新程序会定期生成新的物理认证参数并计算新口令，而后修改 hostapd 配置文件并重启 hostapd，从而使无线网卡发布新的物理认证参数、使用新口令鉴别移动终端。

运行口令更新程序需要配置文件和口令文件。其中配置文件主要存储拟开启 WLAN 热点的无线网卡的名称、拟开启 WLAN 热点的 SSID、口令文件路径、口令更新周期等配置信息。配置信息以键值对的形式存储于配置文件中，配置文件路径则以主函数参数的形式提供给口令更新程序。而口令文件主要存储初始口令及其序列号和失效时间、以及产生初始口令的物理认证参数等口令相关的信息。口令和物理认证参数都是 32 字节的伪随机数，十六进制编码后存储。口令更新

程序会周期性地更新口令文件。

除此之外，主 AP 还需要 Web 服务器用以传输受保护的口令给从 AP。口令更新程序会定期修改 Web 网页文件，让 Web 服务器返回新的受保护的口令给从 AP。其中，加密口令使用的密码算法为 SM4 对称密码算法，而生成消息鉴别码的算法则采用 HMAC-SM3 消息鉴别码算法。密钥由 WLAN 系统内的 AP 共享，十六进制编码后存储在各自口令更新程序的配置文件中。

在启动口令更新程序前，管理员需要为主 AP 设置初始口令：管理员生成 32 字节的伪随机数作为初始口令，并设定其失效时间，初始口令的序列号为零，初始物理认证参数也可以设为全零，并将这些信息写入口令文件。

口令更新程序启动后以守护进程的方式运行在后台，周期性地更新口令。程序分为两部分：初始化和口令的周期性更新。口令更新程序启动后首先需要完成初始化工作：

（1）读取配置信息和初始口令。程序从指定路径读取配置文件，解析其中的口令文件路径、口令更新周期等配置信息，而后从口令文件路径读取口令文件，解析其中的初始口令及其序列号和失效时间等相关信息。

（2）更新初始口令至当前时间。程序获取系统时间作为当前时间，判断初始口令是否已失效，即当前时间是否已超过初始口令的失效时间。如果当前时间已经进入后续的口令更新周期，程序就会逐次地更新口令，直至将口令所属的更新周期推进至当前时间。在这种情况下，口令更新程序就需要将当前时间的口令及相关信息写入口令文件。

（3）重置 hostapd 配置文件相关参数后重启 hostapd，开启 WLAN 热点，让 WLAN 热点使用当前口令鉴别移动终端，并在受控物理环境中广播当前物理认证参数和当前口令的序列号。

（4）重置 Web 网页文件，传输受保护的当前口令给从 AP。

初始化工作完成后，程序就会进入睡眠状态，直至当前口令失效。当口令失效时，口令更新程序便会醒来并更新口令：

（1）生成新物理认证参数。程序自主生成 32 字节的伪随机数作为新物理认证参数。

（2）计算新口令。程序以原口令和新物理认证参数为输入计算新口令（其

中的安全杂凑函数采用 SM3 杂凑算法），将口令所属的更新周期向前推进一个周期——口令的序列号加一，口令的失效时间为原口令的失效时间加上口令更新周期。

（3）更新口令文件，即新口令及相关信息写入口令文件。

（4）修改 hostapd 配置文件相关参数后重启 hostapd，开启 WLAN 热点，让无线网卡广播新的物理认证参数及新口令的序列号，并使用新口令鉴别移动终端。

（5）修改 Web 网页文件，传输受保护的新口令给从 AP。

完成上述工作后程序会再次进入睡眠状态，等待下一次更新口令，就这样周而复始，循环往复。图 4.3 展示了主 AP 口令更新程序的的工作流程。

程序实现了一个专为主 AP 更新 WLAN 口令的类，类成员变量和成员函数分别如表 4.4 和表 4.5 所示。

表 4.4 主 AP 口令更新类的成员变量

Table 4.4 Member Variables of C++ Class for the Master AP Updating Passwords

名称	类型	含义
specific_ssid	std::string	指定 SSID
interface_name	std::string	无线网卡名称
password_file_path	std::string	口令文件路径
current_password	struct passwd	当前口令
update_interval	uint32_t	口令更新周期
physical_parameter	unsigned char[32]	当前物理认证参数
secret_key	unsigned char[32]	向从 AP 安全传输当前口令的密钥

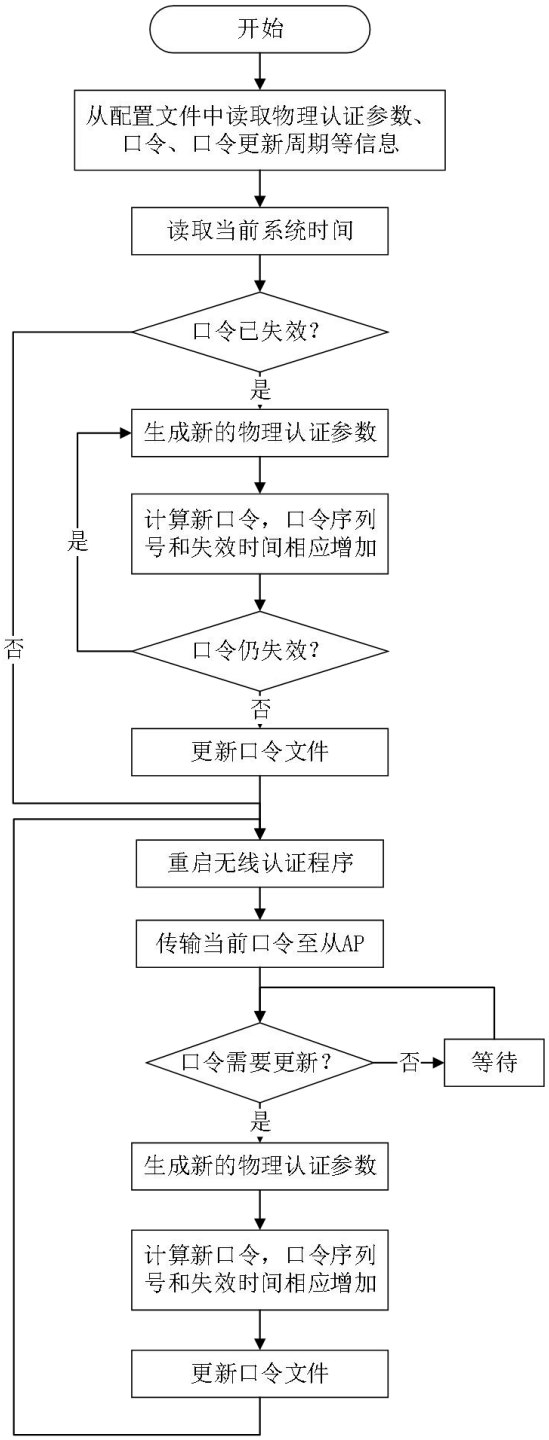

图 4.3 主 AP 口令更新程序的工作流程

Figure 4.3 The Work Flow of the Master AP Updating Passwords

表 4.5 主 AP 口令更新类的成员函数

Table 4.5 Member Functionsof C++ Class for the Master AP Updating Passwords

名称	含义
handle_res init(const std::string&, std::map<std::string, std::string>&);	解析配置文件。从初始口令文件中读取 WLAN 口令
void generate_physical_parameter(void);	生成伪随机数作为物理认证参数
void calculate_password(void);	计算新口令
handle_res update_password_file(void);	更新初始口令文件
handle_res restart_hostapd(void);	启动 hostapd
handle_res transmit_new_password_to_slave_aps(void);	向各从 AP 传输新口令
handle_res main(std::string& conf_file_path);	主程序

其中，成员函数 main 是此类对外提供服务的接口，调用该函数即可实现 WLAN 口令的自动更新，该函数的伪代码如下：

```
init(conf_file, conf) // conf_file 表示配置文件路径，conf 表示键值对形式的配置信息
now = time(0)
while true do
    while now >= current_password.expired_time do
        generate_physical_parameter()
        calculate_password()
    end
    restart_hostapd()
    transmit_new_password_to_slave_aps()
    sleep(current_password.expired_time - now)
end
```

4.2.2 从 AP

为了实现从 AP 口令的自动更新，本设计同样实现了口令更新程序。口令更新程序会定期向从 AP 请求新口令，而后修改 hostapd 配置文件并重启 hostapd，

从而使无线网卡使用新口令鉴别移动终端。

运行口令更新程序需要配置文件。配置文件主要存储拟开启 WLAN 热点的无线网卡的名称、拟开启的 WLAN 热点的 SSID、主 AP 的 IP 地址和端口（主 AP 设置静态 IP，端口为主 AP 的 Web 服务器的监听端口）、解密从主 AP 处获得的口令及验证口令完整性的密钥等配置信息。配置信息同样以键值对的形式存储于配置文件中，配置文件路径也以主函数参数的形式提供给口令更新程序。

管理员不需要设置初始口令便可启动口令更新程序。口令更新程序启动后同样以守护进程的方式运行在后台，周期性地更新口令。程序分为两部分：初始化和口令的周期性更新。口令更新程序启动后需要完成初始化工作：

（1）从指定路径读取配置文件，获取主 AP IP 等配置信息。

（2）请求当前口令。向主 AP 请求当前时间的口令及其序列号和失效时间——与主 AP 建立网络套接字连接，通过套接字发送 HTTP 协议的 GET 请求给主 AP，然后接收来自该套接字的响应。收到来自主 AP 的响应后，口令更新程序需要先解密以得到明文口令，并验证口令的完整性。

（3）重置 hostapd 配置文件相关参数后重启 hostapd，开启 WLAN 热点，让 WLAN 热点使用当前口令鉴别移动终端，并在受控物理环境中广播当前口令的序列号。

初始化工作完成后，程序就会睡眠状态，直至当前口令失效。当口令快要失效时，口令更新程序便会醒来并更新口令：

（1）请求新口令。程序向主 AP 轮询当前时间的口令及其序列号和失效时间，根据返回口令的序列号判断主 AP 的口令是否已经更新。一旦主 AP 更新完口令，就会返回新口令及其序列号和失效时间，口令更新程序发现返回口令的序列号大于自己当前口令的序列号就知道主 AP 的口令更新了。

（2）修改 hostapd 配置文件相关参数后重启 hostapd，开启 WLAN 热点，让无线网卡广播新口令的序列号，并使用新口令鉴别移动终端。

完成上述工作后程序会再次进入睡眠状态，等待下一次更新口令，就这样周而复始，循环往复。图 4.4 展示了从 AP 口令更新程序的工作流程。

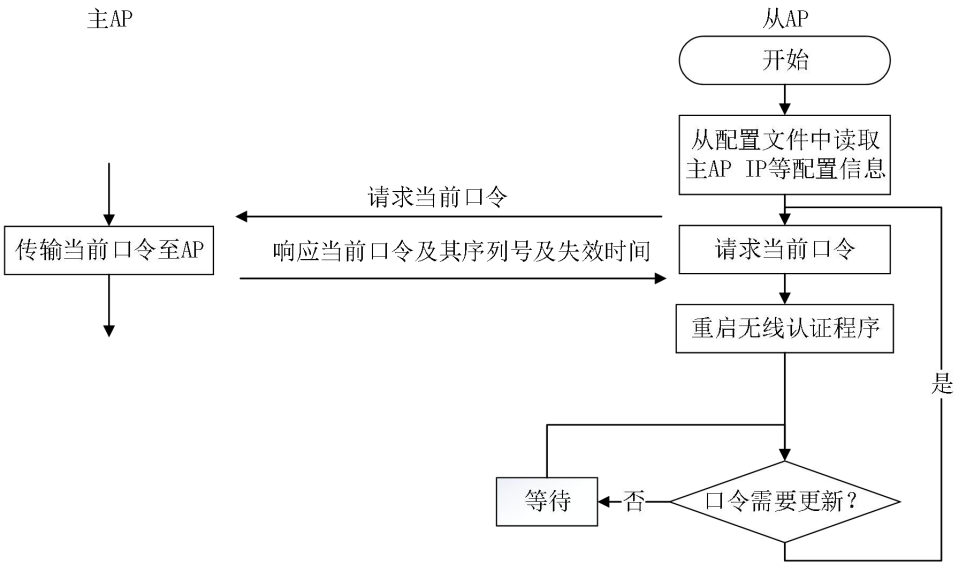

图 4.4 从 AP 口令更新程序的工作流程

Figure 4.4 The Work Flow of the Slave AP Updating Passwords

程序实现了一个专为从 AP 更新口令的类，类成员变量和函数分别如表 4.6 和表 4.7 所示：

表 4.6 从 AP 口令更新类的成员变量

Table 4.6 Member Variables of C++ Class for the Slave AP Updating Passwords

名称	类型	含义
specific_ssid	std::string	指定 SSID
interface_name	std::string	无线网卡名称
specific_ssid	std::string	指定 SSID
current_password	struct passwd	当前口令
master_ap_addr	sockaddr_in	主 AP 的 IP 地址
secret_key	unsigned char [32]	从主 AP 安全传输当前口令的密钥

表 4.7 从 AP 口令更新类的成员函数

Table 4.7 Member Functionsof C++ Class for the Slave AP Updating Passwords

名称	含义
handle_res init(const std::string& conf_file_path, std::map<std::string, std::string>& conf);	解析配置文件
handle_res get_current_password(void);	获取当前口令
handle_res restart_hostapd(void);	启动 hostapd
handle_res main(std::string& conf_file_path);	主程序

其中，成员函数 main 是此类对外提供服务的接口，调用该函数即可实现 WLAN 口令的自动更新，该函数的伪代码如下：

```
init(conf_file_path, conf) // conf_file 表示配置文件路径，conf 表示键值对形式的配置信息
now = time(0)
while true do
    cur_sn = current_password.serial_number
    do get_current_password() until cur_sn != current_password.serial_number
    restart_hostapd()
    sleep(current_password.expired_time - now - 10)
end
```

4.2.3 移动终端

如前所述，借助 wpa_supplicant，无线网卡可以接入指定 WLAN。其中，附近 WLAN 列表的获取、拟接入 WLAN 的 SSID 和预共享密钥的设置都可以通过调用 wpa_supplicant 提供的 wpa_ctrl_request 函数接口并根据需要设置 cmd 参数实现。例如，获取附近 WLAN 的列表可以通过 SCAN_RESULTS 命令实现，同时修改 wpa_supplicant 对 SCAN_RESULTS 请求的响应函数，在返回结果中增加物理认证参数和 AP 当前口令的序列号，因为 SCAN_RESULTS 命令原生的返回结果并不包含开发者自定义的参数（开发者自定义的参数被过滤掉了）；设置拟接入 WLAN 的 SSID 和预共享密钥可以通过 SET_NETWORK 命令实现；验证当前 WLAN 接入状态可以通过 STATUS 命令实现；等等。

为了实现移动终端口令的自动更新，本设计同样实现了口令更新程序。口令更新程序会定期获取附近 WLAN 列表，当发现指定 SSID 的 WLAN 时，解析包含在信标帧中的 AP 口令的序列号及物理认证参数（如果有的话），在需要时更新移动终端的口令，使移动终端和 AP 共享相同的口令从而让移动终端接入 WLAN。

口令更新程序没有配置文件，但有口令文件。口令文件路径、拟接入的 WLAN 的 SSID、WLAN 客户端程序的控制接口等配置信息全部以主函数参数的形式提供给口令更新程序。口令文件主要存储初始口令及其序列号等口令相关的信息。

在启动口令更新程序前，用户需要向管理员或者其他知道口令的用户索取 WLAN 系统正在使用的口令（作为初始口令）及其序列号，然后将上述信息写入口令文件。除此之外，wpa_supplicant 需要事先运行在后台。

口令更新程序启动后以守护进程的方式运行在后台，当需要时自动更新口令。口令更新程序启动后首先也需要完成初始化操作：

（1）解析主函数参数，从中获取口令文件路径等配置信息.

（2）从口令文件中读取初始口令及其序列号等相关信息，作为程序自身的口令。

（3）和 wpa_supplicant 建立 Unix 域套接字连接，将指定 SSID 的 WLAN 的预共享密钥设置为初始口令。

接下来，口令更新程序会周期性地获取附近 WLAN 的列表，如果能够收到指定 SSID 的 WLAN 信号，则从信标帧中提取 AP 当前口令的序列号和物理认证参数，比较 AP 当前口令的序列号和程序自身口令的序列号的大小：

- 如果两者相同，程序不需要更新口令，继续使用原口令接入 WLAN；
- 如果 AP 当前口令的序列号比程序自身口令的序列号大一，并且程序成功从信标帧中提取出了物理认证参数，就能使用和主 AP 相同的算法计算新口令。然后将指定 SSID 的口令设置为新口令，而后尝试接入指定 SSID 的 WLAN。如果移动终端成功使用新口令接入 WLAN，口令更新程序就更新自身的口令，并将新口令写入口令文件。这样即使因计划或意外重启，程序也能在现有口令的基础上更新；

- 如果 AP 当前口令的序列号比程序自身口令的序列号大一，但是程序不能从信标帧中提取出物理认证参数，也就无法更新口令。程序会提示用户通过物理认证、进入受控物理环境以更新口令；
- 如果 AP 当前口令的序列号比程序自身的序列号大很多，程序无法更新口令，就会提示用户口令已永久失效。

图 4.5 展示了移动终端口令更新程序的工作流程。程序主函数的伪代码如下：

```
get initial password from configuration files
connect to wpa_supplicant
set initial password for the specific ssid
while true do
    parse beacon frame
    if found beacon frame with specific ssid then
        if sn == ap_sn then
            join in WLAN
        else if sn + 1 == ap_sn && found physical parameter then
            calculate temporary new password
            set temporary new password for the specific ssid
            try join in WLAN
            if success then
                update passwords
            else
                rollback
        else if sn +1 == ap_sn
            request users to enter the constrained location to update passwords
        else
            inform users that it is impossible to update passwords
    else
        do nothing
end
```

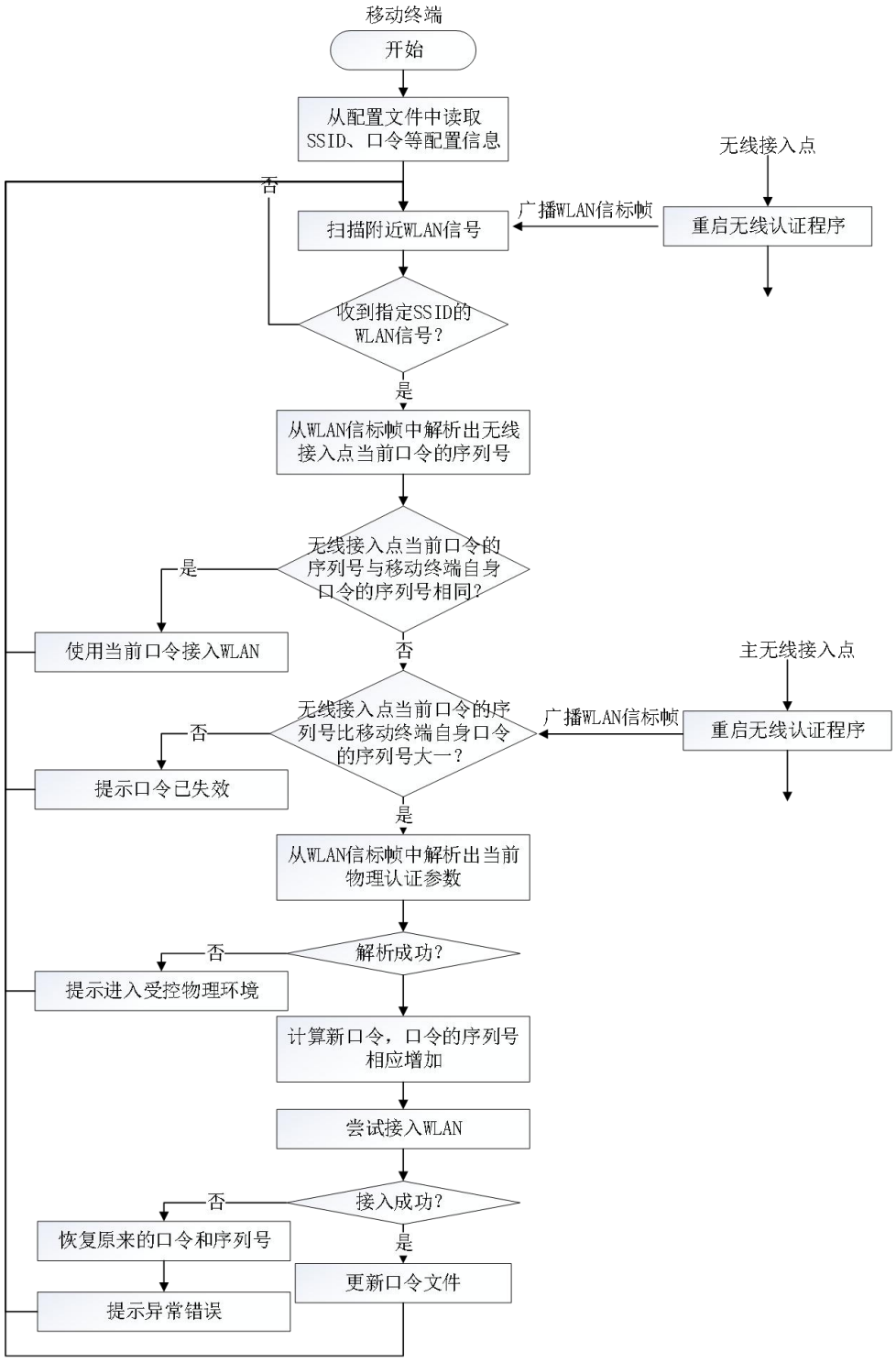

图 4.5 移动终端口令更新程序的工作流程

Figure 4.5 The Work Flow of the Mobile Device Updating Passwords

4.3 系统测试

引入动态口令机制会对移动终端接入 WLAN 的性能造成一定程度的影响，例如移动终端在接入 WLAN 时需要判断口令是否过期，如果是则需要计算新口令，这在一定程度上会延长移动终端接入 WLAN 的时间。另外，如果 WLAN 口令更新时有移动终端正通过 WLAN 访问互联网，该移动终端会经历断线后重连的过程，而这也会对用户的上网体验造成影响，WLAN 系统应当尽可能缩短由 WLAN 口令更新造成的用户无法访问互联网的时间。除此之外，为了方便移动终端在同一 WLAN 系统内不同 AP 间的漫游，主从 AP 需要在尽可能短的时间内完成口令的更新，从 AP 更新时延也属于系统性能的一部分。本设计测试了上述移动终端接入时延、移动终端断网重连时延、从 AP 更新时延等指标，具体如下。

4.3.1 移动终端接入时延

移动终端接入时延是指移动终端在未接入 WLAN 时从收到指定 SSID 的 WLAN 信号时起至成功接入 WLAN 所需的平均时间。由于 WLAN 口令每隔设定的时间间隔会自动更新，移动终端接入口令时存在不更新口令、直接使用自身当前口令接入 WLAN 和更新口令后接入 WLAN 两种情形，实验对这两种情形分别进行了测试。另外，实验还测试了静态口令机制下，移动终端接入 WLAN 所需的平均时间以作为参照。除此之外，如果移动终端需要更新口令，却无法从信标帧中解析出物理认证参数，移动终端将无法更新口令；亦或者相对于移动终端自身口令而言，AP 的口令已多次更新，移动终端也无法更新口令，此时，移动终端均无法接入 WLAN，实验也测试了移动终端从收到指定 SSID 的 WLAN 信号至提示无法更新口令所需的平均时间。实验以扫描得到附近 WLAN 列表为起点，以成功接入 WLAN 或提示无法接入 WLAN 为终点，计算这两个时间节点之间的时间间隔。针对以上不同情形，实验分别测试了 20 次接入时延，并计算了平均数，测试结果如表 4.8 所示。

从测试结果来看，移动终端的接入时延均在 1 秒以内，引入动态口令机制对移动终端的接入时延并无明显影响。

表 4.8 移动终端接入时延测试结果（单位为毫秒）

Table 4.8 Test Results of Access Delay of Mobile Devices

静态口令	无需更新口令	需要更新口令	无法更新口令
			口令已彻底失效：0.0905
594.421	600.694	630.052	需要物理认证参数以更新口令：0.0912

4.3.2 移动终端断网重连时延

移动终端断网重连时延是指移动终端在正常接入 WLAN 时，由于 AP 更新口令、重启 hostapd，导致移动终端与 AP 的连接断开，然后移动终端重新扫描附近 WLAN 列表，从主 AP 广播的信标帧中解析出新的物理认证参数并更新口令，从而接入 WLAN 所需的时间间隔。时延以 AP 开始重启 hostapd 为起点，以移动终端成功接入 WLAN 为终点，计算这两个时间节点之间的时间间隔（AP 和移动终端为同一台计算机上两张不同的网卡，它们共享相同的系统时间）。经测试，本系统的断网重连时延为 3 至 5 秒。经分析，时延主要集中在无线连接程序 wpa_supplicant 启动扫描附近 WLAN 至获得附近 WLAN 列表这段时间内。

4.3.3 从 AP 更新时延

从 AP 更新时延是指主从 AP 更新口令的时间差。同一 WLAN 系统内的全部 AP 应当在尽可能短的时间间隔内完成口令更新。实验以主 AP 开始重启 hostapd 为起点，以从 AP 开始重启 hostapd 为终点，计算了这两个时间节点之间的时间间隔（主从 AP 为同一台计算机上两张不同的网卡，它们共享相同的系统时间）。实验测试了 20 次更新时延，并计算了平均数——为 597.2 毫秒。从测试结果来看，全部 AP 能够在不到 1 秒的时间间隔内完成口令的同步更新，移动终端在不同 AP 间漫游时能够实现正常切换。除此之外，实验还测试了从 AP 更新口令对主从 AP 网络流量、消耗内存、占用 CPU 等性能指标的影响，测试结果如表 4.9 所示。

测试结果表明，更新口令对主 AP 的影响以及对从 AP 的网络流量的影响均微乎其微，但是会对从 AP 的内存和 CPU 等系统资源产生一定程度的影响，但是由于更新时间较短，影响也非常有限。

表 4.9 从 AP 更新时延测试结果

Table 4.9 Test Results of Update Delay of Slave APs

	网络流量（KB/s）	消耗内存（MB）	占用 CPU（%）
主 AP 静息状态下	0-0.2	11.9	0-0.1
主 AP 更新口令时	0.1-0.2	11.9	0-0.1
从 AP 静息状态下	0-0.2	12.3	0-0.1
从 AP 更新口令时	0.1-0.2	20.5	0.4-0.5

第 5 章 结论和创新点

本设计针对有大量外来人员频繁访问的访客 WLAN，研究提出并实现了基于物理访问控制的 WLAN 动态口令系统。WLAN 口令按照设定的时间间隔自动更新，移动终端为了接入 WLAN，必须和 AP 共享相同的口令。一旦 AP 更新了口令，移动终端也必须同步更新，才能按照 802.11i 协议的预共享密钥模式接入 WLAN。WLAN 口令更新后，用户必须通过物理认证，进入单位内部设定的受控物理环境，其持有的移动终端才能更新口令，从而再次接入 WLAN。为实现上述目的，方案引入了只有在受控物理环境中才能获得的物理认证参数，该参数是计算新口令的必需参数。在设定的更新时点，指定的主 AP 生成物理认证参数并将其发布到设定的受控物理环境中，系统内的所有 AP 均以当前口令和新生成的物理认证参数为基础，更新自身口令并使用新口令鉴别移动终端。移动终端得知 AP 口令更新后，通知用户进入受控物理环境，获取物理认证参数并以与 AP 相同的算法计算得到与 AP 相同的口令，从而以新口令接入 WLAN。

本设计的主要创新点有：

（1）将无线认证和物理认证结合，用户是否能够持续接入 WLAN 取决于用户是否能够经常进入受控物理环境，管理员无需为用户作专门的无线访问控制。

用户通过带外方式获得初始口令，获得一段时间的 WLAN 访问权。WLAN 口令更新后，用户只有通过物理认证、进入受控物理环境，才能和 AP 同步更新口令，从而再次获得一段时间的 WLAN 访问权。用户要想在获得初始口令后持续地接入 WLAN，必须经常性地通过物理认证。用户一旦无法在一个口令更新周期内通过物理认证，就失去了 WLAN 访问权。与此相应，用户获得初始口令后，只要能够经常性地通过物理认证、进入受控物理环境，自然能够持续地接入 WLAN，管理员也无法阻止这样的用户接入 WLAN。

（2）实现细粒度、差异化的 WLAN 用户授权管理，能够解决访客 WLAN 的特殊需求。

一方面，可以解决不同类型用户不同的 WLAN 访问需求：机构内部员工可以持续地通过物理认证，也就能够持续地接入 WLAN，而来访人员在结束访问

后就不能通过物理认证，也就失去了 WLAN 访问权；另一方面，可以实现临时用户访问 WLAN 的动态授权和撤销：来访人员在访问期内需要接入 WLAN 时可以向邀请者索取 WLAN 口令，从而获得 WLAN 访问权，结束访问后，由于访客无法再进入受控物理环境，其 WLAN 访问权也就被自动撤销了。

（3）实现方式简单，不增加管理员和用户的操作复杂度，不会给管理员和用户造成额外负担，不降低用户体验。

WLAN 口令设置和分发方式与静态口令完全相同，用户既可以向管理员询问口令，也可以向其他知道口令的用户询问口令，管理员不必面对所有需要接入 WLAN 的人员，用户也不必特地找管理员索要口令。同时，管理员只需要为一个 AP 设置一次初始口令即可，不需要为每个用户创建用户账户并设置有效期，不需要维护每天都在变化的用户列表。除此之外，WLAN 口令自动更新，对管理员和用户均透明，无需用户额外操作，用户无法感知。

（4）增强了基于口令的 WLAN 鉴别协议的安全性。

WLAN 口令复杂度大幅度提高，同时 WLAN 口令不断更新，攻击者没有足够的时间通过穷举攻击找到正确的口令，穷举攻击失效。对于攻击者而言，新旧口令是两个完全不相关的随机数。即使攻击者在足够长的时间内找到了某个口令，也不能将其更新至当前时间从而接入 WLAN。WLAN 口令更新后，攻击者只能从头再来。即使攻击者通过某种渠道获得了口令，其能够造成的消极影响也是有限的，等口令更新后，攻击者就不能再对 WLAN 实施非法行为了。

（5）几乎不降低 WLAN 接入的性能，对用户的使用体验影响小。

引入动态口令机制几乎不影响移动终端的 WLAN 接入时延，在移动终端接入 WLAN 的过程中，用户感受不到移动终端与 AP 进行口令同步的过程。除此之外，在用户正常访问 WLAN 过程中，AP 口令更新使移动终端断网重连给用户带来的影响也较小。移动终端断网重连的时延较小，相信用户可以忍受短时间的网络连接断开。

参考文献

[1] 北京日报报业集团. 北京晨报[N/OL]. 2016, 6369. 北京: 北京晨报社, 2016[2018-03-01]. http://bjcb.morningpost.com.cn/page/1/2016-01/26/B01/20160126B01_pdf.pdf.

[2] Gong M X, Hart B, Mao S. Advanced Wireless LAN Technologies:IEEE 802.11AC and Beyond[J]. Acm Sigmobile Mobile Computing & Communications Review, 2015, 18(4):48-52.

[3] 深智通智能系统. 单位访客管理系统[EB/OL]. [2018-03-01]. http://www.sohu.com/a/218010677_100089839.

[4] 央视网. 消费预警：免费 WIFI 很危险[EB/OL]. [2018-03-01]. http://tv.cntv.cn/video/VSET100158570946/fd4f11798123462ba2c76de9fe65d424.

[5] 华为技术有限公司. WLAN 认证和加密技术白皮书[EB/OL]. [2018-03-01]. http://e.huawei.com/zh/material/onLineView?materialid=6616fe33144845ce9c319c825c518cd8.

[6] 佚名. 全面解析 802.1x 认证原理[EB/OL]. [2018-03-01]. http://blog.csdn.net/lycb_gz/article/details/17523603.

[7] 王少波, 庄重. WLAN 认证方案研究[J]. 数据通信, 2013(3):22-25.

[8] 张志峰. EAP-TTLS 认证方式在 WLAN 中的应用研究[D]. 武汉理工大学, 2006.

[9] 王丽霞. 基于 802.1x/EAP 的 WLAN 安全认证分析与应用研究[J]. 气象科技, 2010, 38(3):347-352.

[10] 陈逸恺, 蔡权伟, 王琼霄. Eduroam 中密码算法的应用分析[J]. 信息安全研究, 2017, 3(6):501-508.

[11] 佚名. wifidog 认证[EB/OL]. [2018-03-01]. http://blog.csdn.net/huithe/article/details/27195903.

[12] 佚名. 浅谈 WLAN 运营中 Portal 认证安全性[J]. 计算机与网络, 2013, 39(2):52-52.

[13] 微信. 微信连 Wi-Fi[EB/OL]. [2018-03-01]. http://wifi.weixin.qq.com/.

[14] 微信. 微信好友 wifi[EB/OL]. [2018-03-01]. http://iot.weixin.qq.com/wiki/document-10_1.html.

[15] 王平, 汪定, 黄欣沂. 口令安全研究进展[J]. 计算机研究与发展, 2016, 53(10):2173-218

8.

[16] IETF. RFC4226 - HOTP An HMAC-Based One-Time Password Algorithm[S/OL]. [2018-03-01]. https://tools.ietf.org/pdf/rfc4226.pdf.

[17] IETF. RFC6238 - TOTP Time-Based One-Time Password Algorithm[S/OL]. [2018-03-01]. https://tools.ietf.org/pdf/rfc6238.pdf.

[18] Hallsteinsen S, Jorstad I, Thanh D V. Using the mobile phone as a security token for unified authentication[C]// International Conference on Systems and Networks Communications. IEEE, 2007:68-68.

[19] Lamport L. Password authentication with insecure communication[J]. Communications of the Acm, 1981, 24(24):770-772.

[20] Pinto A, Costa R. Hash-chain-based authentication for IoT[J]. Adcaij Advances in Distributed Computing & Artificial Intelligence Journal, 2016, 5:págs. 43-57.

[21] Liu H, Zhang Y. An improved one-time password authentication scheme[C]// IEEE International Conference on Communication Technology. IEEE, 2014:1-5.

[22] Studer A, Perrig A. Mobile user location-specific encryption (MULE): using your office as your password[C]// ACM Conference on Wireless Network Security. ACM, 2010: 151-162.

[23] Ryu G, Seo C, Choi D, et al. Location Authentication based on Wireless Access Point Information to Prevent Wormhole Attack in Samsung Pay[J]. Advances in Electrical & Computer Engineering, 2017, 17(3):71-76.

[24] Hsieh W B, Leu J S. Design of a time and location based One-Time Password authentication scheme[C]// Wireless Communications and Mobile Computing Conference. IEEE, 2011:201-206.

[25] H3C 测试中心. 网络之路第十三期：WLAN 专题[EB/OL]. [2018-03-01]. http://www.valleytalk.org/wp-content/uploads/2013/09/wlan.pdf.

[26] IEEE. 802.11i-2004[S].

[27] Liu Y, Jin Z, Wang Y. Survey on Security Scheme and Attacking Methods of WPA/WPA2[C]// International Conference on Wireless Communications NETWORKING and Mobile Computing. IEEE, 2010:1-4.

[28] Vanhoef M, Piessens F. Key Reinstallation Attacks: Forcing Nonce Reuse in WPA2[C]

// The ACM Conference on Computer and Communications Security. ACM, 2017.

[29] Chen C M, Chang T H. The Cryptanalysis of WPA & WPA2 in the Rule-Based Brut e Force Attack, an Advanced and Efficient Method[C]// Asia Joint Conference on Info rmation Security. IEEE Computer Society, 2015:37-41.

[30] 白珅, 王轶骏, 薛质. WPA/WPA2 协议安全性研究[J]. 信息安全与通信保密, 2012(1):1 06-108.

[31] Nakhila O, Attiah A, Jinz Y, et al. Parallel active dictionary attack on WPA2-PSK W i-Fi networks[C]// Military Communications Conference, Milcom 2015 -. IEEE, 2015:6 65-670.

[32] Hafiz M Y M, Ali F H M. Profiling and mitigating brute force attack in home wirel ess LAN[C]// International Conference on Computational Science and Technology. IEE E, 2015:1-6.

[33] Yong-Lei L, Zhi-Gang J. Distributed method for cracking WPA/WPA2-PSK on multi-c ore CPU and GPU architecture[J]. International Journal of Communication Systems, 2 015, 28(4):723-742.[2018-03-01]. https://www.deepdyve.com/lp/wiley/distributed-method-f or-cracking-wpa-wpa2-psk-on-multi-core-cpu-and-gpu-we0CEyXR5C.

[34] 刘永磊, 金志刚, 陈喆,等. wpa/wpa2-psk 高速暴力破解器的设计和实现[J]. 计算机工程, 2011, 37(10):125-127.

[35] Fahmida Y. Rashid. Amazon EC2 Used to Crack Password Encryption on Wireless N etworks[EB/OL]. [2018-03-01]. http://www.eweek.com/security/amazon-ec2-used-to-crack-password-encryption-on-wireless-networks.

[36] Zhang L, Yu J, Zong R, et al. Prevention research of cracking WPA-PSK key based on GPU[C]// International Conference on Consumer Electronics, Communications and Networks. IEEE, 2012:1965-1959.

[37] Liao X, Meng S, Lu K. Security issues and solutions of WPA encrypted public wirel ess Local Area Network[C]// International Conference on Multimedia Technology. IEE E, 2011:3655-3657.

[38] 佚名. Linux Wireless 基础知识（cfg80211 mac80211 nl80211）[EB/OL]. [2018-03-01]. http://blog.csdn.net/liuxd3000/article/details/23761663.

[39] Anonymous. Linux Wireless[EB/OL]. [2018-03-01]. https://wireless.wiki.kernel.org/en/de

velopers/documentation.

[40] 佚名. Linux Wireless Netlink Socket & nl80211[EB/OL]. [2018-03-01]. http://blog.csdn.net/Vichyssoise/article/details/78182779?locationNum=1&fps=1.

[41] 佚名. Linux 中的无线架构[EB/OL]. [2018-03-01]. http://blog.csdn.net/robertsong2004/article/details/38897603.

[42] Jouni Malinen. hostapd and wpa_supplicant[EB/OL]. [2018-03-01]. http://w1.fi/.

致 谢

三年前我满怀着对研究生生涯的憧憬踏进了这所学校，内心洋溢着喜悦之情。三年的学习与科研生涯，既有快乐，也有忧伤，更有对能否顺利完成学业的惶恐与紧张。如今，我已然完成硕士毕业论文，再过不久就将离开国科大、离开信工所了，我的学生生涯就此画上一个圆满的句号，而接下来迎接我的将是一段完全不同的人生旅程。

在过去的三年，我衷心感谢我的导师王琼霄老师。我的毕业论文，是在王老师的精心指点下完成成的。从论题的选择，到研究工作的开展，再到论文的完成，王老师都给了我很多提示和帮助，让我一步一步地完成这篇毕业论文。不仅是毕业设计，我在这三年中取得的科研成果都离不开王老师的悉心指点，让我学会了如何去开展一项科研任务。王老师会时常询问我的科研进展，从中我能够切实地感受到王老师对我的重视与希冀，让我更有动力去完成一项又一项的科研任务。王老师，谢谢您。

然后我要感谢蔡权伟老师、王平建老师和曹洪瑾老师。我曾和曹老师一起参与王老师和蔡老师负责的科研项目，从中收获了很多开发经验，学到了很多编程知识和技巧。另外，蔡老师还为我的论文提供很诸多指点。感谢所有给我指点迷津的老师们。

接下来，我也要感谢和我一起学习、一起参与科研工作的小伙伴们，你们在工作、学习和生活上均给了我很多帮助。特别感谢王明月同学，帮我解决了很多我无法解决的难题。感谢所有这三年中曾经相遇以及现在仍然陪伴在我身边的人们。

三年的研究生生涯即将结束，同时十九年的学生生涯也将落幕，最后我用最后一句话总结，并借此祝福未来的旅程：

长风破浪会有时，直挂云帆济沧海。

陈逸恺

2018 年 6 月

作者简历及攻读学位期间发表的学术论文与研究成果

作者简历：

2011 年 9 月——2015 年 6 月，在南开大学计算机与控制工程学院计算机科学与信息安全系获得学士学位。

2015 年 9 月——2018 年 6 月，在中国科学院信息工程研究所研究所攻读硕士学位。

获奖情况：

2016 年 6 月，获中国科学院大学三好学生荣誉称号

2018 年 2 月，获信息工程研究所所长优秀奖

已发表（或正式接受）的学术论文：

陈逸恺, 蔡权伟, 王琼霄. Eduroam 中密码算法的应用分析[J]. 信息安全研究, 2017, 3(6):501-508.

申请或已获得的专利：

王琼霄, 林璟锵, 陈逸恺, 曹洪瑾. 一种结合物理认证因素的 Wi-Fi 口令动态更新方法及系统: 中国, 201711221952.6[P]. 2017-12-29.

参加的研究项目及获奖情况：

参与统一认证项目，参与完成支持 OIDC 协议的身份认证网关的软件著作权