现有WLAN认证方案：

在较长一段时间内设置一个固定的、相对比较简单的口令。

现有方案存在的问题：

口令容易泄露：

1. 口令设置过于简单，很容易被攻击者猜到；
2. 一旦授权了某个用户，这个用户在口令更换之前都知道口令。

目标：

口令泄露可能导致（1）非法用户未经许可接入WLAN，（2）合法用户被诱骗接入非法AP，以及（3）用户传输的数据被未经授权的用户获得。因此口令需要得到保护。

对于访客WLAN，管理员可能只希望赋予访客时间有限的WLAN访问权。

提出：具有有效期的口令，每位用户的口令都只在其被授权的时间段内有效。

方案：

原理：口令按一定的算法产生并且每天变化，用户只能计算在授权期内的口令。一旦过了授权期，用户就无法计算相应的口令。

算法涉及三类参数（设这些参数的长度均为32字节）：秘密参数P[0]和S，以及公开参数O。

算法分为两部分：正向杂凑和逆向杂凑（设杂凑值也为32字节），而后将两者亦或。

正向杂凑递推式：P[i] = H(P[i-1] XOR O[i])

P[1] = H(P[0] XOR O[1]) P[2] = H(P[1] XOR O[2]) ...... P[365] = H(P[364] XOR O[365])

逆向杂凑一般表达式：H^(366-i)(S)

H^365(S) H^364(S) ....... H(S)

两者异或得到每天的口令：

P[1] XOR H^365(S) P[2] XOR H^364(S) ....... P[365] XOR (S)

安全性分析：

1. 攻击者的猜测复杂度分析
2. 假设攻击者没有获得过口令

攻击者有两种攻击方法：

1. 每天猜测口令，猜测复杂度为365\*32字节，得到全部天数的所有口令。
2. 猜测P[0]和S，猜测复杂度为64字节，得到全部天数的所有口令。

假设攻击者找到P’[0]和S’，使得P’[1] XOR H^365(S’)正好等于P[1] XOR H^365(S)，即找到一对P[0]和S的碰撞，使用这对P[0]和S能够计算得到正确的第一天的口令。此种搜寻的复杂度为32字节，但是攻击者无法将这种碰撞推广到其他天数。

这一结论即使在没有公开参数的情况下也成立：

假设攻击者找到P’[0]、S’，使得正好等于第一天的口令：

H(P’[0]) XOR H^365(S’) = H(P[0]) XOR H^365(S)

但是，却无法用这组数来计算第二天的口令，因为：

H^2(P’[0]) XOR H^364(S’) = H^2(P[0]) XOR H^364(S)

这一等式几乎不会成立。

1. 假设攻击者获得过口令：

攻击者猜测秘密参数S，猜测复杂度为32字节，得到剩余天数的所有口令。

1. 和设置固定随机数口令（设口令的长度为32字节）对比：

攻击者从未获得口令，猜测口令，猜测复杂度为32字节。

攻击者曾经获得过口令，不需要再猜测了，猜测复杂度为0。

1. 和每天重新设置随机数口令（设口令的长度为32字节）对比：

攻击者从未获得过口令，每天猜测口令，猜测复杂度为365\*32字节，得到全部天数的所有口令。

攻击者曾经获得过口令，猜测剩余天数的口令，猜测复杂度为剩余天数\*32字节，得到剩余天数的所有口令。

从猜测口令上看，新解决方案与该方案的安全强度一致。

但是如果某位访客被授予不同时间段的访问权， 如在1月1日和12月31日均被授予访问权，而该访客又能获得全年度所有的公开参数，该访客就能得到全年度的所有口令。而每天重新设置随机数口令就不存在此种情形。

从这一点上说，新解决方案的安全强度稍差于每天重新设置随机数口令的方案。

实现：

1. 现有方案的实现：
2. AP端：

WLAN的SSID、认证方式和口令等配置选项以键值对的形式存储于配置文件中。

无线连接服务读取配置文件，根据配置文件确定使用何种认证方式以及认证过程中使用的口令等。

1. 移动终端：

用户输入口令后将口令和对应的SSID同样以键值对的形式存储于配置文件中。

无线连接客户端使用存储的口令连接对应SSID的WLAN。

1. 新方案的实现：
2. AP端：

在现有配置文件和无线连接服务的基础上，增加口令维护程序，专门用于每天口令的更新。口令维护程序每天计算新的口令，并将口令写入无线连接服务的配置文件，这样无线连接程序就会使用新的口令和移动终端握手了。

1. 秘密参数的存放

32字节长的秘密参数P[0]和S存储于配置文件中，均以base64编码的形式存储。

此外，配置文件还需要存储每天新生成的P[i]。

1. 公开参数的存放

365个32字节长的公开参数O存储于另一配置文件中，同样以base64编码的形式存储。

1. 口令维护程序的日常工作

每天零点，口令维护程序首先从存放公开参数的配置文件中读取当天的公开参数O[i]。然后从存放秘密参数的配置文件中读取前一天的P（最后一行记录的数据就是P[i-1]），计算P[i] = H(P[i-1] XOR O[i])。接着从其读取S，计算H^(366-i)(S)。最后计算P[i] XOR H^(366-i)(S)得到今天的口令。

将口令以及当天的公开参数O以base64编码的形式写入无线连接服务的配置文件，供无线连接服务调用。

公开参数通过Beacon帧广播出去，OpenWrt的代码是开源的，可以通过修改OpenWrt的代码实现Beacon帧对公开参数的支持。

为了解决AP系统时间和终端系统时间不同步的问题，Beacon帧广播公开参数的同时广播系统时间。

完成这些工作后，口令维护程序在其自身的配置文件中增加当天的P（最后一行记录的数据变成了P[i]）。

1. 口令维护程序的年初处理

跨年之前，WLAN管理员需要将下一年的秘密参数P[0]和S，以及365个公开参数通过WLAN管理网站以文件的形式写入AP。

每年第一天零点，口令维护程序读取新一年的配置文件：如果该文件不存在，或者文件中存放的数据不合法，口令维护程序使AP宕机；否则口令维护程序使用新一年的配置文件替换上一年的配置文件。

然后进行每天的日常工作。

1. 口令维护程序的初始化处理

AP容易发生断电，AP断电后，口令维护程序重新运行。口令维护程序首先需要读取配置维护程序的最后一行记录，检查是否是当天的P。

如果是当天的P，说明当天的日常工作已经完成，口令维护程序直接启动无线连接服务即可；

如果不是当天的P，说明当天的日常工作尚未完成。如果AP被断电多日，最后一个P可能是若干天之前的P，那么口令维护程序首先需要计算至前一天的P。最后执行口令维护程序的日常工作，得到当天的口令。

1. 管理员分发口令

由于管理员知道所有的秘密参数和公开参数，因此管理员可以获得任意一段时期的口令，并分发给用户。

注：管理员的权限可能太大，因为管理员能够计算得到任意一天的口令。

1. 移动终端：

移动终端使用定制的应用连接无线网。

1. 访客口令的获得及存放

分发给用户的口令包含以下三部分：上一天的P[i]，授权时长，以及授权结束时S进行若干次杂凑运算后的值S’。

由于口令比较复杂，因此将口令存放在二维码中。管理员可以将二维码显示在显示屏上、打印到纸上、或者通过邮件发送到用户的指定邮箱，供用户扫描。

应用收到口令后，将口令存储于本地。

此外，应用需要获得WLAN的SSID，SSID可以在分发口令时一并分发。

1. 获取公开参数

用户启动应用时，应用首先判断终端是否已接入WLAN。

如果终端未接入WLAN，则开始接收Beacon帧，解析SSID。

如果收到指定SSID的Beacon帧时，从Beacon帧中解析系统时间和公开参数O[i]。

如果AP系统时间和终端系统时间差距过大，则拒绝连接这一AP，否则使用AP的系统时间作为当前时间进行授权状态检测。

如果应用未收到指定SSID的Beacon帧，则每隔固定时间继续接收。

1. 授权状态检测

用户以AP的系统时间作为当前时间检测移动终端的授权状态：检测当前时间是否超过授权期，如果超过授权期，则提示用户，不再进行后续操作；

如果还在授权期范围内，再从口令文件中读取口令文件中最后一条P，如果最后一条P不是今天或者昨天的P，说明用户昨天未接入WLAN，应用无法计算今天的口令，则提示用户，不再进行后续操作。

如果最后一条P是昨天的P，说明用户今天未接入过WLAN，进行第4步；

如果最后一条P是今天的P，说明用户今天接入过WLAN，进行第5步。

1. 用户今天未曾接入过WLAN

首先从口令文件中读取上一天的P，计算P[i] = Hash(P[i-1] XOR O[i])，然后从口令文件中读取S’，进行一定次数的杂凑运算，最后与P[i]异或，得到今天的口令。然后以今天的口令接入指定SSID的WLAN。

如果移动终端接入指定SSID的WLAN，说明获得的公开参数正确，口令计算过程无误，增加今天的P至口令文件的最后一行。

1. 用户今天接入过WLAN

首先从口令文件中读取今天的P，然后从口令文件中读取S’，进行一定次数的杂凑运算，最后与P[i]异或，得到今天的口令。然后以今天的口令接入指定SSID的WLAN。

1. 接入后处理

终端接入WLAN后，终端开始每隔固定时间检测连接状态。

如果连接断开，则重复第2步。

附：无线接入流程图：



AP无缝切换

（1）信号覆盖上的无缝，且客户端可以自由且及时的切换。

（2）切换时间较短，不会中断业务。

（3）最核心的是，跨三层漫游时业务不会中断

Fat AP，不同AP SSID相同，口令相同，认证方式相同，加密方式相同，信道错开

无缝切换取决于终端，如果信号太差了，终端可能决定换一个AP，也可能不。

AP可以设置一个阈值，如果信号太差，强制把终端踢下线。

IP肯定要换，导致上层业务中断。

Fat AP，802.11r（快速漫游）协议

家用，WPA-PSK认证

https://www.reddit.com/r/openwrt/comments/515oea/finally\_got\_80211r\_roaming\_working/

http://www.jianshu.com/p/25845e7ed6a4

hostapd和openwrt支持，即AP可以支持，要求终端也支持

AC + Fit AP模式：无线AP的管理集中在AC端，包括SSID和口令等，只要控制AC即控制了所有的AP。

企业级，高成本

https://www.zhihu.com/question/19751226

AP之间共享公开参数

1. 所有的AP都持有一份相同的公开参数表，如果全部使用，则更新公开参数表，管理员上传备用公开参数表，AP根据自己的系统时间更新公开参数。

公开参数表需要定期更新

AP之间的系统时间可能不完全相同导致口令不相同

2. 有一个主AP，生成新一天的公开参数，其他AP（从AP）从主AP获取公开参数：

2.1. 假设现在是第i天伊始，主AP生成第i+1天的公开参数，从AP在新一天伊始向主AP查询公开参数，主AP返回第i-1天、第i天和第i+1天的公开参数，从AP可以根据需要提取相应的公开参数：

如果主AP已进入第i天，而从AP还在第i-1天，并且从AP需要第i-1天的公开参数，它可以提取第i-1天的公开参数；

如果主AP已进入第i天，且从AP也进入第i天，需要更新公开参数，它可以提取第i天的公开参数；

如果主AP未进入第i+1天，但从AP已进入第i+1天，需要更新公开参数，它可以提取第i+1天的公开参数。

AP之间的系统时间可能不完全相同导致口令不相同

存在单点失效的情形

2.2. 主AP在新一天的伊始更新公开参数，从AP在新一天伊始（或者提前一小段时间）向主AP查询新一天的公开参数，主AP返回当前时间和公开参数，如果从AP没有获取到新一天的公开参数，则过一段时间后继续向主AP请求，直到获得新一天的公开参数。

主AP更新公开参数和从AP请求存在一定的时间差

新一天伊始网络流量会比较大

如果从AP没有及时向主AP获取新一天的公开参数，从AP的口令会滞后更新于主AP

存在单点失效的情形

2.3. 主AP在新一天的伊始更新公开参数，从AP在新一天伊始（或者提前一小段时间）向主AP查询新一天的公开参数，主AP判断是否很快就要更新公开参数，或者刚更新完公开参数：如果很快就要更新公开参数，则暂时先不返回，等更新公开参数后再返回新的公开参数（考虑到如果长时间不返回消息，连接可能断开，如果从AP请求的时间过早，则主AP需要先返回当前时间的公开参数，从AP收到响应后继续向主AP请求新的公开参数）；如果刚更新完公开参数，则立即返回当前的公开参数；如果两者都不是，则立即返回当前的公开参数。

新一天伊始的网络流量会比较大。

如果从AP没有及时向主AP获取新一天的公开参数，从AP的口令会滞后更新于主AP

新一天伊始网络流量会比较大

存在单点失效的情形

2.4. 主AP在新一天的伊始更新公开参数，从AP在新一天伊始（或者提前一小段时间）和主AP建立长链接（不断发送心跳包），一旦有从AP与主AP建立连接，主AP首先要返回当前时间的公开参数。如果当前时间的公开参数正是主AP需要的，从AP断开与主AP的长连接。如果主AP更新了公开参数，则通知所有与其建立了连接的从AP新的公开参数，从AP进入新一天，并断开与主AP的长连接。

如果从AP没有及时向主AP获取新一天的公开参数，从AP的口令会滞后更新于主AP

新一天伊始网络流量会比较大

存在单点失效的情形

主AP：

从文件中读取初始口令和公开参数；

涉及文件的读，判断是否需要更新初始口令

生成新的公开参数；

涉及随机数生成，需要考虑多线程读写公开参数（和初始口令）

分发公开参数（和初始口令）给从AP；

涉及接收、处理并响应HTTP请求，涉及多线程，需要考虑多线程读写公开参数（和初始口令）

广播公开参数；

涉及自定义Wi-Fi管理帧

计算新口令；

使用新口令重新启动无线连接程序；

涉及调用其它程序

更新初始口令和公开参数。

涉及文件的写，（需要考虑多线程读写初始口令）

从AP：

从主AP获取新的公开参数（和初始口令），但不广播公开参数；

涉及发出HTTP请求，处理HTTP响应，判断公开参数（和初始口令）是否有效

（从文件中读取初始口令）；

涉及文件的读，判断初始口令是否已经过期

计算新口令；

使用新口令重新启动无线连接程序；

涉及调用其他程序

（更新初始口令）。

涉及文件的写

移动终端：

通过带外方式获得初始口令并存储在本地；

涉及二维码的读取和解析

从文件中读取初始口令；

涉及文件的读，判断初始口令是否已经过期

从AP获取公开参数；

涉及解析Wi-Fi管理帧中的自定义数据，判断公开参数是否有效

计算新口令；

使用新口令接入无线网；

涉及调用其他程序

更新初始口令。

涉及文件的写

场景阐述

关于结合物理控制的WIFI接入

1. 针对的环境。大公司建筑，其中也有门卫等物理控制（有些房间是需要有门禁才能够进入的）。大公司将WIFI信号布满全部建筑，不可避免，有些信号会在“有物理控制”的房间之外。

在大面积范围，会使用多个WIFI AP。

1. 需求：
   1. 给访客和员工WIFI使用，必须有Password控制；否则会有人非法使用和窃听。
   2. 对于访客，不希望使用创建用户名/口令的方式。否则，需要维护不同的用户名/口令；每天增加、每天删除，IT管理工作量太大。
   3. 使用尽可能简化的方案，注重使用上的便利和快捷。
2. 方案：
   1. 使用我们之前讨论的方案。每天的口令是Pi = Oi XOR Hash(Pi-1)
   2. 在核心的几个房间，有门禁监控的房间，AP有广播公开参数Oi。保证在没有门禁监控的地方，不可能获得Oi；**在单位进出的必经之路。**注意，在其他的一些地方，仍然有AP信号，但是这些AP不会广播Oi。
3. 过程：
   1. 访客到特定的房间，自动获得了Oi。
   2. 由于我们假定他是访客，所以有人接待，该员工会把当前的Pi告诉访客。合法员工打开自己的客户端，就可以看到当前的Pi。
   3. 访客只要能够每天进入门禁（假设有成功的门禁物理控制），就能够自动获得当天Password。过了几天、进不去公司，Password自动失效了。
   4. 获得了Password的访问，想要作为窃听攻击者，在公司旁边窃听。也只能窃听当天，因为没有能力获得Oi。
4. 效果：
   1. WIFI的口令是不断变化的，访客不可能在事后在公司旁边窃听、非法接入网络。
   2. 任何访客（甚至合法员工）只需要获得一次口令，就能够长期地合法使用WIFI。有足够的便利性。
   3. 结合门禁物理控制，自然地使得访问的Password失效。同时，保证WIFI信号是遍及所有工作空间的。
   4. WIFI的接入控制是非常简单的。只有1个口令，没有创建用户/删除用户的必要，也没有IT管理人员的工作量。直接由员工告诉访客当天的Password，操作上非常简单。
   5. 方案参数可调，可以要求每天来；也可以松散一些，隔一二天来等等（参考我们之前的方案）。

时变Wi-Fi口令

你需要做的是：

1. 理清楚现有的想法；做什么？为什么这么做？有什么逻辑。

2. 理清楚还需要做些什么工作。

Here is a brief summary of our discussions so far:

Topic: Adding temporal constraints (including time limit and continuity) to Wi-Fi passwords

Objective: one entry of password (with a simple app on user’s device), temporal access for a period of time with certain continuity requirements.

Assumption: No conspire/collusion attacks. No user accounts created (usability requirement).

Security: To break the temporal constraints, an attacker needs to break the hash functions. No conspire/collusion attacks can be addressed without creating user accounts.

we need to finish two things:

1. the scheme, => security and usability.

2. a prototype system with 802.11 Wi-Fi.

你先把我们周一中午讨论的内容总结一下，邮件发给之前转给你等我邮件中的里liyingjiu老师还有我和林老师（英文，算是对之前邮件讨论的接续）

匿名性、前向安全、后向安全。

yjli@smu.edu.sg

linjingqiang@iie.ac.cn

wangqiongxiao@iie.ac.cn

I sorted out the proposed solutions.

The main idea of the existing solutions is to establish a functional relationship between Wi-Fi password and time. So Wi-Fi password can be calculated from a specific formula. However, users can only calculate Wi-Fi passwords for a specific period of time, and thereby they can get Wi-Fi access during this time. So the key of the proposed solutions is the expression of the function.

There are two parameters in the proposed solutions:

One is the public parameter O. It ensures users’ physical presence. That is, users must connect to Wi-Fi everyday. But it remains uncertain that whether it can bring more security. And in some that scenarios, it is unnecessary to ask user to connect to Wi-Fi everyday.

The other is the private parameter S. It determines the deadline for Wi-Fi access. After deadline, user can’t connect to Wi-Fi.

Now we have three solutions.

In the original solution, the function is:

P[i] = Hash(P[i-1]) XOR O[i].

This function only contains the public parameter O. So it can only guarantee the continuity of Wi-Fi access.

In the second solution, the function is:

P[i] = Hash(P[i-1]) XOR O[i],

P’[i] = P[i] XOR Hash^(366-i)S.

This function contains both two parameters. So it can guarantee both the continuity and the deadline for Wi-Fi access.

In the third solution, the function is:

P[i] = Hash(P[i-1]),

P’[i] = P[i] XOR Hash^(366-i)S.

Or it can be rewritten as:

P[i] = Hash^(i)P[0] XOR Hash^(366-i)S

This function only contains private parameter S. So it can only guarantee the deadline for Wi-Fi access.

However, there is conspiracy attack in the second and third solution. Conspirators can calculate more Wi-Fi passwords than what they can calculate separately. This means that, conspirators can have longer time period of Wi-Fi access than the sum period they originally have. Also, this may do harm to forward security.

As long as Wi-Fi password is calculated out, it will be used as the pre-shared key of WPA/WPA2-PSK authentication. But we don’t know whether creating accounts can do it better. Maybe using WPA/WPA2-PSK without creating accounts have prominent advantages in some scenarios.

我整理了现有的方案。

现有方案的核心思想是在Wi-Fi口令和时间之间建立函数关系，Wi-Fi口令能够根据特定的公式计算得到。用户只能计算出特定时间段的Wi-Fi口令，从而获得这段时间的Wi-Fi访问权。因而解决方案的关键是Wi-Fi口令对时间的函数表达式。

除了时间，现有方案还涉及到以下参数：

公开参数O，这保证了用户登录的连续性。用户必须保证每天登陆一次Wi-Fi。

除了保证用户每天登陆Wi-Fi，该参数是否能带来更多的安全性？

在某些情况下，要求用户每天登陆可能是不必要的。

保密参数S，这确定了用户登录的截止日。过了截止日，用户不能再登陆Wi-Fi。

现在有三个子方案：

第一个方案的计算公式为：

该方案只有公开参数O，所以只能保证用户访问的开始时间和连续性，但是不能保证访问的结束时间。

第二个方案的计算公式为：

该方案包括公开参数O和保密参数S，所以能够保证用户访问的开始时间、结束时间和连续性。

第三个方案的计算公式为：

该方案只有保密参数S，所以只能保证用户访问的开始时间和结束时间，但是不能保证访问的连续性。

该方案存在共谋攻击，并且合谋能够获得的Wi-Fi访问期将超过合谋者各自的Wi-Fi访问期的并集。

无论采取何种计算公式，生成的口令最后都将作为WPA/WPA2-PSK的预共享密钥。

现在存疑的是使用不创建用户的WPA/WPA2-PSK是否比创建用户的WPA/WPA2-EAP更具有优越性。

需求：在不创建用户的前提下，授予不同用户以不同时间段的Wi-Fi访问权。

思路：在Wi-Fi口令中添加时间约束，Wi-Fi口令随着时间的变化而变化。用户只能知道特定时间段的Wi-Fi口令，从而获得该段时间内的Wi-Fi访问权。

安全假设：没有合谋攻击。

方案1：

每天更新一个公开参数O，该参数是一个随机数，每天发生变化。设第i-1天的口令是P[i-1]，第i天的公开参数是O[i]，则第i天的口令P[i]由下式决定：

P[i] = Hash(P[i-1]) XOR O[i]。

授权方法：如果希望授权某一用户访问Wi-Fi（当天是第i天），则只需要告知该用户前一天的口令P[i-1]，该用户就能访问当天以及后续各天的Wi-Fi。而只要授权用户有一天没有登陆Wi-Fi，那么这个用户就彻底失去了Wi-Fi的访问权。

安全性考虑：

该方案能够保证用户访问的开始时间以及访问的连续性，但是不能保证访问的结束时间。访问的开始时间必须是当前时间。

如果一个用户因为一次偶然的机会知道了Wi-Fi的口令，并且这个用户能够保持每天登陆一次Wi-Fi，那么这个用户将一直拥有Wi-Fi的访问权。

为了保证授权期间经过后用户将无法获得Wi-Fi的访问权，需要引入一个参数，授权期间经过后，用户无法再计算出这个参数。

方案2：

每年更新一个保密参数S，该参数也是一个随机数，但是每年发生变化，同时每天更新一个公开参数O。我们仍然能够计算：

P[i] = Hash(P[i-1]) XOR O[i]。

但是，第i天的口令的计算表达式改为：

P’[i] = Hash(366-i)(S) XOR P[i]。

授权方法：

如果希望授权某一用户访问Wi-Fi（当天是第i天），授权期为x天，则只需要告知该用户前一天的P[i-1]以及S1 = Hash(366-i-(x-1))(S)，用户可以计算得到后续各天的口令。

第i天：Hash(x-1)(S1) XOR O[i] XOR Hash(P[i-1]) = Hash(366-i)(S) XOR P[i]

第i+1天：Hash(x-2)(S1) XOR O[i+1] XOR Hash(P[i]) = Hash(366-i-1)(S) XOR P[i+1]

第i+(x-1)天：Hash0(S1) XOR O[i+(x-1)] XOR Hash(P[i+(x-2)]) = Hash(366-i-(x-1))(S) XOR P[i+(x-1)]

可用性考虑：

能否实现任意确定访问的开始时间。

安全性考虑：

该方案能够保证访问的开始时间、结束时间和连续性。访问的开始时间必须是当前时间。

能不能从公开参数O上考虑，公开参数可以通过加密递增序列得到。

方案3：

每年更新两个保密参数S1和S2。

第i天的Wi-Fi口令由下式决定：

P[i] = Hash(i)(S1) XOR Hash(366-i)(S2)。

授权方法：

如果希望授权某一用户访问Wi-Fi，授权期从第i天开始，至第j天结束，则只需要告知该用户A = Hash(i)(S1)和B = Hash(366-j)(S2)，用户可以计算得到各天的口令：

第i天：Hash0(A) XOR Hash(j-i)(B) = Hash(i)(S1) XOR Hash(366-i)(S2)

第i+1天：Hash1(A) XOR Hash(j-i-1)(B) = Hash(i+1)(S1) XOR Hash(366-i-1)(S2)

第j-1天：Hash(j-i-1)(A) XOR Hash1(B) = Hash(j-1)(S1) XOR Hash(366-j+1)(S2)

第j天：Hash(j-i)(A) XOR Hash0(B) = Hash(j)(S1) XOR Hash(366-j)(S2)

安全性考虑：

该方案能够保证用户访问的开始时间和结束时间，但是不能保证访问的连续性。访问的开始时间可以任意确定。

存在合谋攻击，并且合谋能够获得的访问期限将超过所有合谋者被授权的访问期限的并集。

可用性考虑

保证访问的开始时间和连续性是否是必要的？

输入Hash值并不方便：Hash值以二维码的方式显示。

来宾扫描下载应用链接的二维码，下载并安装应用。然后邀请者生成Wi-Fi口令的二维码信息，让来宾用应用扫描二维码，连接Wi-Fi。

需要完成的工作：

1、查阅相关文献；

在文献情报中心中搜索了time evolving password / time constrain password / time limit password / Wi-Fi password time / temporal constraints Wi-Fi password 都没有找到相关文献（目标不相关，思路不相关）。应该属于创新开创性的成果。

2、考虑安全性、可用性，确定最终方案；

3、查看相关协议，提出整体方案，考虑实现的技术细节；

方案：

用户以扫描二维码的方式获得S和有效期，应用程序存储，并计算当日的S。

在无线设备和AP建立Association后，AP在WPA2-PSK的第一次握手协议中添加当日的O[i]以及AP的系统时间（精确到日即可），在EAPoL-Key frame的Key Data中，无线设备在收到O[i]后即可计算当日的P[i]，其他过程与协议规定的一致。

每日0点，无线访问接入点主动发起WPA2-PSK的握手协议，无线设备收到第一个握手包后根据AP时间和O[i]重新计算P[i]，响应AP的鉴别请求。对不能通过鉴别请求的无线设备，AP应当断开其连接。

难点：能否向数据报文中添加数据？能否提取出相应的数据？

协议过程、帧格式，如何在协议中添加鉴别数据；

（1）基础知识

802.1x协议详解

http://wenku.baidu.com/link?url=VdItclyU0nPAkGH3H3WT1Q7pSlbq-BnEAvCo-p\_IrZJUH2DLLP1AiZrth4cYUWicRSpP\_kAjZgWBtg-55hHtxQFaBAPjafWmOK1KXz-Q-tK

EAPoL – Extensible Authentication Protocol over LAN

https://www.vocal.com/secure-communication/eapol-extensible-authentication-protocol-over-lan/

802.1x技术介绍

http://www.h3c.com.cn/Products\_\_\_Technology/Technology/Security\_Encrypt/Other\_technology/Technology\_recommend/200812/624138\_30003\_0.htm

WPA-PSK无线网络破解原理及过程

http://sec.chinabyte.com/339/13254339.shtml

WEP,WPA-PSK,WPA2-PSK握手深入分析1

http://blog.csdn.net/ivan\_/article/details/39205371

Wi-Fi 四次握手 协议过程

http://blog.sina.com.cn/s/blog\_602c72c50101isuh.html

关于成对密钥与组密钥的结构 协议过程

http://bbs.chinaunix.net/thread-3645998-1-1.html

关于WPA/WPA2 4次握手 帧格式

http://www.cnblogs.com/chenyuejun/p/4480481.html

详解EAPOL的报文格式(内含截图) 帧格式

http://wenku.baidu.com/link?url=yiLOGYh6W2bxcEJEOkiFspNNRyezLKa5FvgzU8g\_GDVvlguqrRhGU2Rg0VJIpkfGWS5tk5CDroM7bcwwDBi95jf1V86BoKR3iU-eUevcNPa

802.11 Sniffer Capture Analysis - WPA/WPA2 with PSK or EAP 帧格式

https://supportforums.cisco.com/document/100611/80211-sniffer-capture-analysis-wpawpa2-psk-or-eap

无线接入过程三个阶段

http://blog.csdn.net/myarrow/article/details/7930131

Wi-Fi网络接入原理（上）——扫描Scanning

http://blog.csdn.net/peixiuhui/article/details/45674881

Wi-Fi网络接入原理（中）——认证Authentication

http://blog.csdn.net/peixiuhui/article/details/45674893

802.11管理功能–用户接入过程

802.11帧格式 帧格式

http://blog.csdn.net/myarrow/article/details/9306961

802.11协议数据包类型

http://www.jianshu.com/p/6cc4ea0dc0bc

802.11概述及帧结构分析

http://blog.csdn.net/lizzywu/article/details/5010938

802.11抓包分析 帧格式

http://wenku.baidu.com/link?url=HJTVyBF3wgmcaVkjq1yeREFKN22TpQ-pbiv378JRAmf\_3T1kya6fBsFCbi92XUU4J2Ei4IXB-cR5PeSvj1a-EobgiwhOoTOXfJO833H6Ee\_

https://mrncciew.com/2014/10/27/cwap-802-11-probe-requestresponse/

如何对AP和无线网卡编程。

http://www.aichengxu.com/aspnet/46603.htm

PSK？服务器？

4、在设备上实现。

另外，你要把Wi-Fi登录的各种应用场景好好分析一下，以便提出我们具体能解决什么问题

在宾馆住宿、去其他单位办事（开会、考试）：不定期使用

工作单位、学校：长期使用

家：长期独占使用

机场、火车站、饭店：短期使用

给熟人使用

WPA2-PSK

WPA2-Enterprise

微信好友连Wi-Fi

手机号和接收到的验证码构成用户名和口令。

eduroam

1. Wi-Fi使用场景

* 公园、景区、商场、市场、体育馆、公交车、车站、火车站、机场、医院、酒店、餐厅、咖啡厅、网吧、政府部署的覆盖整个行政区域的

人员流动性非常高

非特定用户

谁都可以访问，口令分发困难

* 医院住院部、学校（临时访问）、会议中心、宾馆、租房（短期）

人员流动性中等

特定用户（住院病人、临时访问人员、住户）访问且访问时长不同

* 工作单位、住宅

人员流动性低

特定用户长期访问Wi-Fi

1. 目前的认证方式及存在的问题

* 不认证（点击一键登录后登录，接收验证码短信后登录）

所有人都可以访问，蹭网现象不可避免，会降低授权用户的上网体验；

* 身份认证，使用身份证的末六位作为口令

身份信息可能泄露；

需要维护后台服务器和用户数据库。

* 短信认证，向指定手机号发送用户名和口令短信

短信网关可能出问题；

支付短信费用；

需要维护后台服务器和用户数据库。

* 设置固定口令

不考虑人员流动；

口令会被非授权人员知晓；

口令过于简单，通常是联系电话，很容易被猜到；

服务员需要经常接受顾客咨询。

* 无线漫游——针对学校

成立联盟；

部署工作量大。

1. 认证渠道

* Web Portal认证、微信认证

相当于开放

* 开放

数据传输未加密

* 802.1x

中等程度的通信保密性

* 802.1x+EAP

通信完全保密

1. 威胁模型

面向宾馆、住院部、会议中心等的Wi-Fi认证：

特点：人员流动性中等，可以授权特定的用户访问，且不同用户需要的访问时间不同。

无线接入点直接鉴别用户，没有后台鉴别服务器参与；

不需要用户数据库，不创建用户账户；

授予特定的人特定的Wi-Fi访问权限；

不同的人拥有不同时长的Wi-Fi访问权限；

不考虑共谋攻击。

1. 解决方案

扫描二维码，下载客户端；

获得口令二维码，口令包含用户的授权截止日及截止日的Wi-Fi口令；

客户端扫描并存储口令信息；

客户端接收路由器的系统时间和当天的公开参数，计算当天的口令，接入无线网；

经过一天，无线路由器重新计算Wi-Fi口令、生成新一天的公开参数，并要求所有终端重新登录；

客户端重新接收路由器的系统时间和新一天的公开参数，计算新一天的口令，接入无线网。

年前，人工重置新一年的保密参数。

无线接入点的工作：

保存前一天的口令；产生并广播当天的公开参数；计算今天的口令；鉴别移动终端。

移动终端客户端的工作：

保存前一天的口令；接受当天的公开参数并计算当天的口令；接入无线网。

1. 优势

低成本，不需要维护用户数据库和后台服务器；

不同用户拥有不同时间的Wi-Fi访问权限；

防止猜测口令攻击。

新的方案：

看上去需要创建用户

需要应用，其目的是管理口令

存储多次杂凑后的口令

目的：为了防止离线攻击

问题：延长对合法用户的鉴别时间，为什么我们的方案能够解决这一问题

O\_i作为每次会话的盐值：

优势：用户很容易就能发现账号被别人登录过

问题：O\_i怎么作为盐值

口令怎么生成、服务器怎么存储用户的口令

即使发现被登录了，有些重要信息也泄露了

为了计算下一次的口令，服务器必须存储本次或下次的口令

上面两者冲突了

同态杂凑

额外的秘密信息

优势：需要不断地突破服务器

问题：直接拿到种子秘密可以吗

挑战：一个设备一个账号。

口令泄露的后果

非法接入点，造成用户数据被攻击者截获

未经授权接入无线网，拖慢网速，影响合法用户上网体验

授予不同用户不同时长的Wi-Fi访问权限

用户的口令泄露造成的不良影响有限

为了计算整个期间的预共享密钥，攻击者必须穷举两个种子密钥

攻击者可以减少杂凑运算次数，但是攻击者能够计算的期限也相应缩短了

攻击者必须先获得预共享密钥，然后才能获得口令

攻击者获得了当天的预共享密钥对其知晓种子密钥没有任何帮助

需要考虑的安全性有用户访问的匿名性、前向安全。

时变Wi-Fi口令解决方案

1. 背景

随着智能手机的普及，人们对无线网络（WLAN）的需求越来越大，同时随着无线技术的发展，无线接入点（AP）成本的下降，WLAN的覆盖范围也在不断扩大。覆盖了WLAN的公共场所也越来越多，比如公交车、火车站、饭店、宾馆等。同时很多企业也部署了来宾WLAN，供来访的宾客使用。以宾馆为例，现在绝大多数宾馆提供WLAN接入服务，方便宾客在下榻期间通过WLAN访问互联网。宾馆WLAN的认证方式从可以分为以下几种：

（1）不认证，WLAN处于开放状态，任何人都可以接入AP。

（2）预共享密钥认证，通常家用AP都会采用此种认证方式，给AP设置一个固定的口令，只有正确输入该口令，宾客才能接入WLAN。

（3）用户名和口令认证，即在宾客办理入住手续后为宾客生成特定的用户名和口令，通常生成的用户名和口令只在宾客下榻期间有效，宾客办理退宿手续后，该用户名和口令也就失效了。

（4）身份信息认证，以住户独一无二的身份信息如身份证号作为认证信息，住户在接入WLAN时服务器要求住户提供身份信息，只有身份信息对应的住户确实在宾馆下榻，服务器才会允许住户接入WLAN。

（5）微信（微博、QQ）认证，即要求用户关注宾馆的微信公众号才能接入互联网，但是由于任何人都可以关注该宾馆的微信公众号，因此此类认证从本质上说也是不认证。

（6）手机认证，如果是向用户任意指定的手机号发送验证码，并要求用户正确输入其收到的验证码，用户才能接入互联网，此类认证从本质上说是不认证，因为只要是用户合法输入的手机号，服务器都会向其发送验证码。如果用户不能任意指定手机号，而必须使用其在下榻时注册的手机号，那么这种方式则类似于身份信息认证。

对于后四者而言，宾馆的AP通常使用的是网页登录。AP本身是开放的，即不认证的，住户首先与AP建立连接，但是住户对网络的访问却是受限的，住户只能访问登录网站。AP在与住户建立连接后会将住户重定向至登录网页，要求用户输入认证信息，登录服务器通过身份鉴别后才会允许住户访问互联网。

上述认证的方式都针对WLAN对用户的认证方式而言的，其中只有预共享密钥认证用户才对WLAN进行了认证，而其他认证用户均未认证用户的合法性。

然而这些认证方式都存在一定程度的安全威胁，由于WLAN的普及，这些安全隐患也显得越来越重要。

1. 安全威胁

无线安全标准802.11i规定了三种认证方式：不认证、预共享密钥（PSK）认证和可扩展鉴别协议（EAP）认证。绝大部分来宾WLAN采用前两种认证方式。不认证的WLAN没有安全性可言，移动终端和AP相互不鉴别对方身份，两者之间传输的数据既不加密也没有完整性保护。对于PSK认证，移动终端和AP通过预先设置的共享密钥相互鉴别对方身份，同时使用PSK派生的密钥对传输的数据进行加密和签名。但是如果PSK被恶意人员知晓，恶意人员就能伪装成合法用户接入WLAN，也能伪造AP与合法用户建立无线连接，还能解密AP和合法移动终端之间传输的数据，这样的WLAN的安全性和开放WLAN的安全性相同。

针对WLAN存在以下类型的攻击：

（1）蹭网攻击：窃取合法用户的认证凭据，伪装成合法用户接入WLAN；

（2）非法AP攻击：伪装成合法的AP，欺骗合法用户与其建立连接；

（3）机密性攻击：截获AP和移动终端之间传输的数据，获取其中的隐私信息；

（4）完整性攻击：篡改数据包或者发出非法的数据包，致使移动终端和AP之间的连接断开或者作出错误判断；

（5）拒绝服务攻击：使得AP不能接受合法用户的接入请求。

本文主要针对前两种攻击：蹭网攻击和非法AP攻击。蹭网攻击的原因是AP未鉴别用户身份，而非法AP攻击则是由于用户未对AP进行识别。对于不认证的WLAN而言，用户和WLAN相互不认证，任何用户都能接入，所以蹭网在所难免。非法用户的蹭网占据了WLAN的带宽，影响了合法用户的上网体验。考虑到用户通过SSID选择接入的AP，恶意人员也能通过将其控制的AP的SSID命名为目标AP的SSID以将其控制的AP伪装成合法AP。通常用户在能收到多个相同SSID的WLAN情况下，用户会选择信号更强的WLAN。如果非法AP离合法用户更近，非法AP的信号更强，用户就很有可能与非法建立连接。2012年2月，有黑客自爆在星巴克和麦当劳，将其自身笔记本电脑伪装成合法的AP，诱骗在星巴克和麦当劳就餐的顾客与其建立无线连接，并使用抓包软件窃取顾客的个人信息和网银、支付宝密码。因为对不认证的AP而言，用户只能根据SSID选择想要连接的WLAN，大大降低了非法AP的搭建难度。同样，对于网页版的用户名和口令认证与身份认证，尽管AP会鉴别用户的身份，但用户却无法鉴别AP的合法性，恶意用户只需要在搭建非法AP的同时部署界面与合法AP类似的网站就可以达到欺骗合法用户的目的。

对于预共享密钥认证方式的WLAN而言，在移动终端接入AP的过程中，两者以PSK为依据、经过四次握手相互鉴别对方的身份，只有两者持有相同的PSK，握手才能成功。因而如果攻击者不知道WLAN的预共享密钥，那么攻击者就无法接入WLAN，也无法伪装成合法的AP欺骗合法用户与之建立连接。但是预共享密钥认证存在下述问题：

首先，来宾WLAN 的口令通常设置的比较简单，例如8个8，或者企业的联系电话等，攻击者很容易才到来宾WLAN的口令。当然这种问题的解决方式也比较简单，设置一个相对比较复杂的口令，并定期更换。

其次，如果攻击者通过某些方式获得了来宾WLAN的口令，例如在宾馆下榻一晚，攻击者在此后很长的一段时间内都知道了宾馆WLAN的口令，因为宾馆不会不断地更换口令。也就是说用户一旦通过某些合法的方式获得了来宾WLAN口令的访问权，即使该用户之后结束了对某一单位的访问，亦即失去了WLAN的访问权，该用户仍然能访问WLAN。换言之，WLAN口令的长期性和用户授权访问WLAN的暂时性存在矛盾。

因此需要一种机制平衡上述矛盾，同时提供AP和用户的双向鉴别。

1. 解决方案
2. 原理

为了解决上述安全威胁，我们提出了以下方案：

每年更新每天更新一个公开参数O，该参数是一个随机数，每天发生变化。设第i-1天的口令是P[i-1]，第i天的公开参数是O[i]，我们可以计算

P[i] = Hash(P[i-1]) XOR O[i]。

另外，在计算每年第一天的口令时，不使用上一年最后一天的口令，而是使用新生成的随机数作为前一天的口令。同时每年更新一个保密参数S，该参数也是一个随机数，但是每年发生变化，我们可以计算

P’[i] = Hash(366-i)(S) XOR P[i]，

以及

PSK = Hash(P’[i] XOR O[i])。

如果希望授权某一用户访问WLAN（当天是第i天），授权期为x天，则只需要告知该用户前一天的P[i-1]以及S1 = Hash(366-i-(x-1))(S)，用户可以计算得到后续各天的口令：

第i天：Hash(x-1)(S1) XOR O[i] XOR Hash(P[i-1]) = Hash(366-i)(S) XOR P[i]

第i+1天：Hash(x-2)(S1) XOR O[i+1] XOR Hash(P[i]) = Hash(366-i-1)(S) XOR P[i+1]

……

第i+(x-1)天：Hash0(S1) XOR O[i+(x-1)] XOR Hash(P[i+(x-2)]) = Hash(366-i-(x-1))(S) XOR P[i+(x-1)]

首先，该方案采用了PSK认证的方式，该方式提供AP和移动终端双向鉴别，在不知道口令的前提下，攻击者无法蹭网，也无法伪装成合法AP欺骗合法用户与之建立连接。

其次，该方案中的口令是不断变化的，不断变化的口令增加了攻击者猜测口令的难度。一方面口令并非永久有效，一旦经过了授权期，用户就无法计算新一天的口令，也就无法接入WLAN。假设用户想要计算授权期过后的第一天的口令

Hash(366-i-x)(S) XOR P[i+x]

= Hash-1(S1) XOR O[i+x] XOR Hash(P[i+(x-1)])

其中P[i+(x-1)]是用户已知的，O[i+x]是公开的，用户可以获得，但是用户无法计算逆杂凑函数Hash-1(S1)，因此用户在经过授权期之后无法计算新一天的口令。

再次，如果攻击者如果想要猜测口令，攻击者可以猜测当天的口令，口令是随机数的杂凑值，其穷举范围比普通的口令要大得多。退一步讲，即使攻击者成功猜测了当天的口令，攻击者也只是知道了当天的口令，但无法知道之后以及之前的口令。所以相对于随机数口令，攻击者穷举口令的成本与之相当，但是攻击者能够获得收益却远小于随机数口令。当然攻击者可以试图猜测种子密钥S，由于当天的密钥是种子密钥的若干次杂凑后的结果，多次杂凑运算会增加攻击者穷举的时间。如果攻击者想要知道更多天数的口令，他每一次猜测都需要进行更多次数的杂凑运算，攻击的成本机会上升。而攻击者如果降低攻击的成本，减少攻击的计算复杂度，攻击者能够获得的收益也将下降。

最后，即使攻击者通过某些手段获得了合法用户的口令，其能够造成的影响也是有限的，因为合法用户的口令也是有有效期的，攻击者只能知道合法用户授权期内的口令，而对其他时间段的口令一无所知。

1. 实现

疑问

P[i] = Hash(P[i-1]) XOR O[i]中公开参数的作用。

PSK = Hash(P’[i] XOR O[i])公开参数类似于盐，对口令的影响。

与每年更新一个很复杂的口令的区别是否足够明显

任务计划

|  |  |  |
| --- | --- | --- |
| 解析配置文件 | 6月13日 |  |
| Base64编码、SM3杂凑 | 6月14日 |  |
| 无线接入点口令生成逻辑 | 6月16日 |  |
| 向广播帧中添加供应商自定义参数 | 8月3日 | hostapd |
| 读取广播帧中的供应商自定义参数 | 9月5日 | iw |
| 接入指定SSID的无线网 | 9月5日 | wpa\_supplicant |
| 移动终端口令生成逻辑 |  |  |
| 整合现有代码 |  |  |
|  |  |  |
| 交叉编译 | 7月6日 |  |
|  |  |  |
| 移植到安卓手机上 |  |  |
| 调用摄像头，读取并解析二维码 |  |  |
| 生成并展示二维码 |  |  |

遇到的问题和解决方法

netlink/netlink.h没有那个文件或目录

Sudo apt-get install libnl-3-dev

cd /usr/include

sudo ln -s libnl3/netlink netlink

查看网卡的MAC地址

ifconfig -a

禁用指定MAC地址的无线网卡

sudo vim /etc/NetworkManager/NetworkManager.conf

[keyfile]

unmanaged-devices=mac:xx:xx:xx:xx:xx:xx

sudo service network-manager restart

将无线网卡转为监视模式

sudo ifconfig wlp3s0 down

sudo iwconfig wlp3s0 mode monitor

sudo ifconfig wlp3s0 up

hostapd开启无线热点

sudo hostapd -d hostapd.conf

iw扫描无线网

sudo iw dev wlp3s0 scan -u

wpa\_supplicant接入指定SSID的无线网

sudo wpa\_supplicant -i wlp3s0 -c wpa\_supplicant.conf

对……未定义的引用

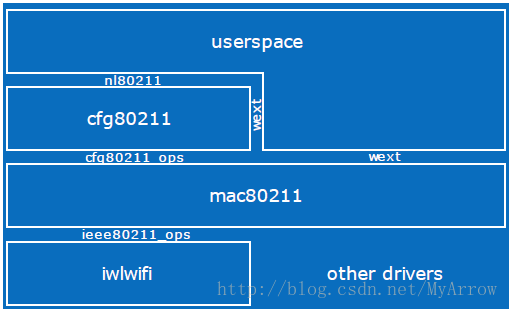
$(pkg-config --cflags --libs libnl-3.0 libnl-genl-3.0)

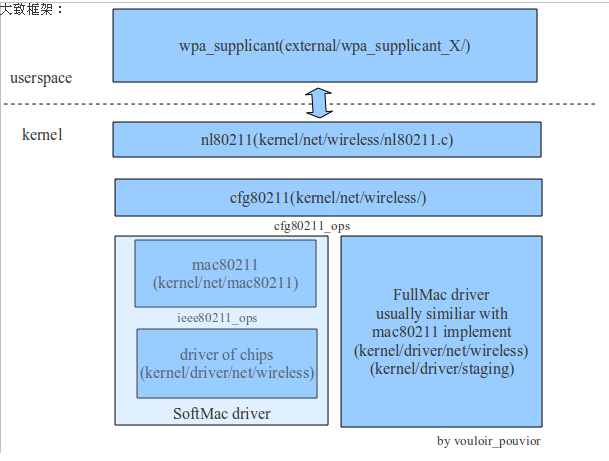
--cflags 表示编译时需要；

--libs 表示链接时需要。

参考资料

OpenWrt Wireless





<http://www.linuxwireless.org/en/developers/Documentation/Glossary/>

<http://www.linuxwireless.org/en/developers/Documentation/mac80211/>

<https://wireless.wiki.kernel.org/en/developers/Documentation/mac80211>

<http://blog.csdn.net/wsclinux/article/details/50608399>

<http://blog.csdn.net/liuxd3000/article/details/23761663>

<http://blog.csdn.net/hui523hui523hui523/article/details/38493725>

<http://blog.csdn.net/kris_fei/article/details/76901145>

<http://blog.chinaunix.net/uid-27167114-id-3783603.html>

<http://blog.csdn.net/robertsong2004/article/details/38897603>

<http://blog.csdn.net/u012845519/article/details/19400077>

<http://blog.csdn.net/an_zhenwei/article/details/16975363>

MLME Stands for Media Access Control (MAC) Sublayer Management Entity. MLME is the management entity where the Physical layer (PHY) MAC state machines reside. Examples of states a MLME may assist in reaching:

Authenticate

Deauthenticate

Associate

Disassociate

Reassociate

Beacon

Probe

Timing Synchronization Function (TSF)

MLME: 即MAC(Media Access Control ) Layer Management Entity，它管理物理层MAC状态机。

FullMAC is a term used to describe a type of wireless card where the MLME is managed in hardware. You would not use mac80211 to write a FullMAC wireless driver.

FullMAC：其MLME由硬件管理，当写FullMAC无线驱动时，不需要使用mac80211。

mac80211 is a framework which driver developers can use to write drivers for SoftMAC wireless devices.

SoftMAC devices allow for a finer control of the hardware, allowing for 802.11 frame management to be done in software for them, for both parsing and generation of 802.11 wireless frames. Most 802.11 devices today tend to be of this type, FullMAC devices have become scarce.

mac80211 implements the cfg80211 callbacks for SoftMAC devices, mac80211 then depends on cfg80211 for both registration to the networking subsystem and for configuration. Configuration is handled by cfg80211 both through nl80211 and wireless extensions.

In mac80211 the MLME is done in the kernel for station mode (STA) and in userspace for AP mode (hostapd).

If you have new userspace utilities which support nl80211 you do not need wireless-extensions to support a mac80211 device.

mac80211: 是一个驱动开发者可用于为SoftMAC无线设备写驱动的框架 （Kernel态）。

<http://www.cnblogs.com/rohens-hbg/p/5407495.html>

<http://patchwork.ozlabs.org/patch/471839/>

<http://patchwork.ozlabs.org/patch/471841/>

One Time Password

[1] rfc6238-TOTP: Time-Based One-Time Password Algorithm

[2] rfc4226-HOTP: An HMAC-Based One-Time Password Algorithm

[3] One-time Password for Biometric Systems: Disposable Feature Templates

Hash-chain Based Authentication Protocol

基于杂凑链的鉴别协议 Lamport鉴权协议

次数有限，杂凑链如何更新；双向鉴别；鉴别同时提供机密性、完整性和非否认性

[1] Hash-Chain-Based Authentication for IoT, 2016, Universidad de Salamanca

如何更新杂凑链，使用共享密钥生成杂凑链的种子，杂凑链的末端作为请求的令牌。

[2] An Efficient Authentication Infrastructure of WLAN Based on Self-Updating Hash Chain, 2007

自更新杂凑链技术解决杂凑链长度有限的问题。

[3] 通用可组合的自更新 Hash 链认证模型，2011

[4] Hash chains authentication protocol of non-repudiation in mobile payment, 2011, 计算机应用研究

鉴别同时提供机密性、完整性和非否认性，使用杂凑链的倒数第二个值作为请求口令，并使用该值对请求签名，使用杂凑链的末端加密。

[5] A Hash-Chain Based Method for Full or Partial Authentication of Communication in a Real-Time Wireless Environment, 2010

传输实时数据流时鉴别数据来源，保护数据完整性和非否认性。

[6] A countable and time-bound password-based user authentication scheme for the applications of electronic commerce, 2009, Information Sciences

口令鉴别，种子口令多次模平方，每次登录成功后更新口令，新口令是旧口令的模平方根。

口令有次数和有效期限制，服务器不用保存用户口令。

[7] 对一种基于口令和时间限制的用户身份认证协议的改进，2011，四川理工学院学报（自然科学版）

口令鉴别，模平方改为杂凑运算。

口令有次数和有效期限制。

[8] Hash Chain based Strong Password Authentication Scheme, 2016

[9] A Hash-Chain Based Authentication Scheme for Fast Handover in Wireless Network, 2006

移动终端在不同无线接入点之间移转时鉴别状态的移转，提高效率、增强安全性、同时降低管理负担。

[10] A Novel Hash Chain Construction for Simple and Efficient Authentication, 2016

两个用户协商获得共享密钥：只有用到杂凑函数、只需要知道对方的ID、只有初始化阶段需要通信。

[11] A Probabilistic Signature Scheme to Provide Authentication Through a Hash Chain, 2006

二维杂凑链，可以用在鉴别协议中。

[12] Comments on a One-Way Hash Chain based Authentication for FMIPv6, 2010

对一种新的鉴别协议的讨论。

[13] A Novel Sender Authentication Scheme Based on Hash Chain for Vehicular Ad-Hoc Networks, 2011

新的鉴别协议，鉴别车载自组织网络中的汽车。

[14] A Random Increasing Sequence Hash Chain and Smart Card-Based Remote User Authentication Scheme, 2013

新的远程用户鉴别协议。

Wi-Fi Password Security

Parallel active dictionary attack on WPA2-PSK Wi-Fi networks

并行线上攻击

基于WPS的线上攻击

[1] Michelle Gong, Brian Hart, Shiwen Mao; ”Advanced Wireless LAN Technologies: IEEE 802.11AC and Beyond”, SIGMOBILE Mobile Computing and Communications Review, ACM,(2015), Volume 18 Issue 4: pp 48-52.

用WLAN的人越来越多

[5] L. Zhang, J. Yu, R. Zong, J. Chang, and J. Xue, “Prevention research of cracking WPA-PSK key based on GPU,” in Proceedings of the 2nd International Conference on Consumer Electronics, Communications and Networks (CECNet), 2012, pp. 1965–1969.

用WLAN的人越来越多

[5] Andrew Gin, Ray Hunt; ”Performance analysis of evolving wireless IEEE 802.11 security architectures”, The 8th International Conference on Mobile Technology, Applications and Systems,(2008).

更换所有移动终端的口令不切实际

对WPA口令的攻击

[2] Halil Bulbul, Ihsan Batmaz, Mesut Ozel; ”Wireless network security: comparison of WEP (Wired Equivalent Privacy) mechanism, WPA (WiFi Protected Access) and RSN (Robust Security Network) security protocols”,The 1st international conference on Forensic applications and techniques in telecommunications, information, and multimedia and workshop, (2008): pp 1-8.

穷举口令攻击；少于20个字符的口令是易受攻击的

[8] I. P. Mavridis, A.-I. E. Androulakis, A. B. alkias, Ph. Mylonas; ”Real-life paradigms of wireless network security attacks ”,Panhellenic Conference on Informatics, (2011): pp 112-116.

使用线下猜测的方式

Crack WPA-PSK Password

A Method for Cracking the Password of WPA2-PSK Based on SA and HMM

The security analysis of WPA encryption in wireless network

Survey on Security Scheme and Attacking Methods of WPA/WPA2

[3] Liu Yong-lei, Jin Zhi-gang, Wang Ying, Survey on security scheme and attacking methods of WPA/WPA2, IEEE 6th International conference on Wireless Communication Networking and Mobile Computing, Chengdu:IEEE,2010, 1-4.

暴力破解；改进的暴力破解；--其他的一些破解方式

Wifi security cracked

使用亚马逊云破解口令的代价为每个口令1.68美元

WPA/WPA2协议安全性研究

Hash Table字典攻击是唯一有效的办法；速度慢，一般只能达到每秒500密钥；用分布式算法和GPU提高破解速度

现有解决方案的不足

WPA Security

Security issues and solutions of WPA encrypted public wireless Local Area Network

伪装成AP

DH算法 可选通信模式

The security analysis of WPA encryption in wireless network

[3] Dongsheng Yin, et al. A Research into The Latent Danger of WLAN [J].Computer Science and Education .2011: 1085-1090.

[4] Hui Wang, et al. Study on Secure Encryption Protocol of WALN[J]. Communications Technology. 2011,44(237),118.

破解时间长；使用GPU破解；定期更换密钥、增加密钥复杂度、使用企业级加密

Defense of WPAWPA2-PSK Brute Forcer

[4] Liu Yong-lei, Jin Zhi-gang, Chen Zhe, et al, Design and Implementation of High-speed Brute Forcer for wpa/wpa2-psk, Computer Engineering, 2011, 37(10):125-127.

分布式猜测攻击

WPA Security

Profiling and Mitigating Brute Force Attack in Home Wireless LAN

易受拒绝服务攻击和暴力破解攻击；

[3] Laishun Z, Minglei Z, Yuanbo G. 2010. “A Client Puzzle Based Defense Mechanism to Resist DoS Attacks in WLAN”. International Forum on Technology and Applications.

让AP重新鉴别移动终端

The Cryptanalysis of WPA & WPA2 in the Rule-Based Brute Force Attack, An Advanced and Efficient Method

[7] K. Tran, “GPU - accelerated WPA PSK cracking solutions,” Minnesota State University, 2010.

GPU加速破解

Distributed method for cracking WPA/WPA2-PSK on multi-core CPU and GPU architecture

Prevention research of cracking WPA-PSK key based on GPU

使用GPU提高破解速度；G-WPA-PSK

攻击产生的影响

Location Related

Dynamic Password

<http://las.ac.cn/>

**dynamic password**

Secure authentication with dynamic password

IEMCON 2016

使用动态口令增强3D口令

Dynamic password token based on SM3 algorithm

ICALIP 2016

提出了一种基于SM3的动态口令算法

Differential Power Analysis on Dynamic Password Token Based on SM3 Algorithm, and Countermeasures

CIS 2015

对基于SM3的动态口令算法的差分能量攻击

A dynamic password-based user authentication scheme for hierarchical wireless sensor networks

JOURNAL OF NETWORK AND COMPUTER APPLICATIONS

使用动态口令提高无线传感器网络节点之间的传输效率

Improvement of Kerberos protocol based on dynamic password and "One-time public key"

ICNC 2014

使用动态公钥提高Kerberos协议的安全性

Security Analysis and Improvement for Kerberos Based on Dynamic Password and Diffie-Hellman Algorithm

International Conference on Emerging Intelligent Data and Web Technologies 2013

对使用动态公钥提高Kerberos协议的安全性的分析

Effects of User Habituation in Keystroke Dynamics on Password Security Policy

International Symposium on High-Assurance Systems Engineering 2011

用户习惯对口令安全的影响

DPASS - Dynamic password authentication and security system using grid analysis

International Conference on Electronics Computer Technology  2011

基于网格分析的动态口令机制

A dynamic password chip design based on Logistic chaotic algorithm

International Conference on Communication Software and Networks  2011

基于逻辑混沌的动态口令

1. 基于某些算法提出一些动态口令

2. 使用动态口令提高特定系统的安全性

**wifi password**

Flexible Communication: A Secure and Trust-Based Free Wi-Fi Password Sharing Service

International Conference on Trust, Security and Privacy in Computing and Communications 2011

基于社交网络的Wi-Fi口令共享服务

1. 几乎没有什么论文跟Wi-Fi口令相关

**location-related password**

Usability and security evaluation of GeoPass a geographic location-password scheme

基于地理位置（地图定位）的口令

Mobile user location-specific encryption (MULE) using your office as your password

使用地理位置作为加解密密钥

SPAM: A Secure Password Authentication Mechanism for Seamless Handover in Proxy Mobile IPv6 Networks

IEEE Systems Journal

PMIPv6网络中无缝切换时的口令鉴别机制

TrustOTP Transforming Smartphones into Secure One-Time Password Tokens

基于硬件的动态口令

When CSI Meets Public WiFi Inferring Your Mobile Phone Password via WiFi Signals

使用Wi-Fi作为侧信道窃取隐私信息

Monte Carlo Strength Evaluation Fast and Reliable Password Checking

评估口令被攻破的可能性

SAuth protecting user accounts from password database leaks

攻击者必须同时攻破多个服务的口令才能登录这些服务

Graphical passwords Learning from the first twelve years

图片口令

Testing metrics for password creation policies by attacking large sets of revealed passwords

使用熵来度量口令安全性

**location-based password**

Design of a time and location based One-Time Password authentication scheme

International Wireless Communications and Mobile Computing Conference 2011

基于时间和地理定位的动态口令

GeoMoB - A geo location based browser for secured mobile banking

ICoAC 2016

地理定位认证

Location-enhanced authentication using the IoT because you cannot be in two places at once

Annual Conference on computer security applications 2016

基于地理定位的物联网认证

My password is here! An investigation into visuo-spatial authentication mechanisms

Interacting with Computers

基于地理定位的图形认证

Designing challenge questions for location‐based authentication systems: a real‐life study

Human-centric Computing and Information Sciences

根据手机的地理定位生成不同的口令重置问题

LRAP: A Location-Based Remote Client Authentication Protocol for Mobile Environments

International Euromicro Conference on Parallel 2011

针对位置信息容易被伪造，对基于地理定位的认证的增强

Exploring the Potential of GeoPass: A Geographic Location-Password Scheme

INTERACTING WITH COMPUTERS

验证了基于地理定位的认证方式的有效性

1. 一般是基于地理定位的

**location-based authentication**

A Location-Based Authentication System Leveraging Smartphones

International Conference on Mobile Data Management 2014

根据用户手机记录的地理位置移动轨迹来提出登录挑战问题，正确回答后登录成功

Geo-location based QR-Code authentication scheme to defeat active real-time phishing attack

基于地理定位的二维码认证方案

A Secure Condition-Based Location Authentication Protocol for Mobile Devices

CMCSN 2016

使用附近的移动终端协助认证

Collaborative Bluetooth-based location authentication on smart phones

PERVASIVE AND MOBILE COMPUTING

使用蓝牙来获取附近附近的移动终端信息来定位，以实现认证

Location Authentication based on Wireless Access Point Information to Prevent Wormhole Attack in Samsung Pay

Advances in Electrical and Computer Engineering

收集一些无线AP的信息，比较是否相同来确定地理位置

Location-based authentication system using space dependent information

annual international conference on mobile systems, applications, and services 2013

使用地理定位不好，用其他的信息来代替定位，同时又是基于地理位置的

Easy attendance: location-based authentication for students integrated with moodle

RoEduNet International Conference 2013

使用NFC来确定地理位置

1. 使用定理定位

2. 搜集其他地理位置相关的信息来确定地理位置

**time-based password**

A secure cloud storage system combining Time-based One Time Password and Automatic Blocker Protocol

International Computer Engineering Conference 2015

利用了TOTP标准

Integration of Deffie Hellman Key Exchange Encryption and Advanced Encryption Standard Algorithm for Securing SMS based One

基于DH和AES的动态口令算法

One time password authentication scheme based on elliptic curves for Internet of Things (IoT)

基于椭圆曲线的动态口令算法

Security Attack Safe Mobile and Cloud-based One-time Password Tokens Using Rubbing Encryption Algorithm

基于Rubbing加密算法的动态口令算法

HMAC based one tıme password generator

基于HMAC的动态口令算法

An Identity-Based One-Time Password Scheme with Anonymous Authentication

基于双线性对的智能卡的动态口令算法，临时身份，匿名登录

**one time password**

CCF参考文献

IEEE Transactions on Dependable and Secure Computing

Zero-Sum Password Cracking Game: A Large-Scale Empirical Study on the Crackability, Correlation, and Security of Passwords

评估了1.45亿口令的安全性、评估了商业口令策略的有效性

Security Analysis of Password-Authenticated Key Retrieval

基于口令的密钥检索协议的安全性分析

Achieving Flatness: Selecting the Honeywords from Existing User Passwords

改进诱饵口令机制

Password Extraction via Reconstructed Wireless Mouse Trajectory

从鼠标移动轨迹中抽提口令

IEEE Transactions on Information Forensics and Security

DPPG: A Dynamic Password Policy Generation System

静态的口令检查机制会提高攻击成功的概率，提出了一种动态的口令策略生成系统

Personal Information in Passwords and Its Security Implications

研究了口令中携带隐私信息的情况

Someone in Your Contact List: Cued Recall-Based Textual Passwords

回忆口令的机制

Zipf’s Law in Passwords

口令的潜在规律

A New Multimodal Approach for Password Strength Estimation

没有密码强度指标本身比任何一个密码强度指标更好

Regional Patterns and Vulnerability Analysis of Chinese Web Passwords

中文口令的脆弱性分析

Cracking More Password Hashes With Patterns

总结一些用户常用口令的模式，提高字典攻击的效率

Next Gen PCFG Password Cracking

提高字典攻击的效率

Learning Multi-Boosted HMMs for Lip-Password Based Speaker Verification

提出了一种新的口令方案，基于唇部运动，说对口令，唇部运动特征还要匹配

Captcha as Graphical Passwords—A New Security Primitive Based on Hard AI Problems

基于困难人工智能问题的图形口令机制

Journal of Cryptology

Round-Optimal Password-Based Authenticated Key Exchange

构建具有最佳圆复杂性的基于口令的密钥交换协议的通用框架

ACM Transactions on Information and System Security

Inhibiting and Detecting Offline Password Cracking Using ErsatzPasswords

保护在服务器中存储的口令，用硬件相关的设备来存储口令的杂凑值，这样即使口令泄露了攻击者也无法恢复

A Large-Scale Evaluation of High-Impact Password Strength Meters

评估口令检测算法的有效性

Designing Password Policies for Strength and Usability

评估了现有口令策略的有效性，提出可以在增强口令策略的同时增强口令策略的可用性

Computers & Security

Towards enhancing click-draw based graphical passwords using multi-touch behaviours on smartphones

结合多点触摸的图形口令机制

EvoPass: Evolvable graphical password against shoulder-surfing attacks

定期地更改口令以必面shoulder-surfing攻击

Time evolving graphical password for securing mobile devices

时变的图形口令方案

User practice in password security: An empirical study of real-life passwords in the wild

评估随着人们安全意识的提高，口令是否发生了变化

Access control lists in password capability environments

使用口令做访问控制

Leakage-resilient password entry: Challenges, design, and evaluation

防止口令泄露的方案

ACM Conference on Computer and Communications Security

UbiKiMa: ubiquitous authentication using a smartphone, migrating from passwords to strong cryptography

提出并实现了一种基于手机的口令鉴别协议和系统

GOTCHA password hackers!

攻击者需要和计算机交互才能实施穷举攻击，防范离线穷举攻击

The password allocation problem: strategies for reusing passwords effectively

如果用户重用口令（多个网站共用一个口令），少量的几个口令如何在多个网站之间分配才是合理的

Passwords and interfaces: towards creating stronger passwords by using mobile phone handsets

由于界面布局问题，PC键盘口令比手机口令强，提出一种新的手机界面布局方式，提高手机口令安全性

测试结果

|  |  |  |  |  |
| --- | --- | --- | --- | --- |
| 轮询间隔 | CPU% | 内存M | 网络KB/s | 时延ms |
| 静息状态 | 0 | 12.3 | 0.3 | - |
| 500ms | 0.1 | 12.3 | 0.4 | 758 1604  1234 2121  706 1654  1179 1674  659 1209 |
| 400ms | 0.1 | 12.3 | 0.5 | 884 1664  1359 2085  836 1523  1307 2148  779 1386 |
| 300ms | 0.1 | 12.3 | 0.4 | 779 1359  1271 1976  743 1519 |
| 100ms | 0.1 | 12.3 | 0.5 | 1467 1980  943 1997  1411 1991 |
| 50ms | 0.1 | 12.3 | 0.5 | 1243 1760  715 1780  1187 1659 |
| 10ms | 0.6 | 12.3 | 0.6 | 699 1183  1183 1658  655 1136 |
| 5ms | 1.3 | 12.3 | 0.5 | 651 1118  1119 1591  591 1065 |
| 1ms | 3.5 | 12.3 | 0.8 | 1155 1618  1046 1634  518 995 |
| 0 | 9.5 | 12.3 | 0.7 | 1082 1558  558 1573  1038 1506  507 1522  983 1450 |

AP重启 0.602 0.604 0.6 0.606

AP重启完毕至移动终端收到解认证信号 0.042 0.044 0.045 0.04

移动终端收到解认证信号至开始扫描 0.1 0.1 0.1 0.101

扫描附近WLAN信号 3.096=1.286+0.535+1.275 1.286 1.27 1.289

重新接入WLAN 0.182 0.176 0.184 0.189

合计 4.022 2.21 2.199 2.225

1ms 10ms 100ms 300ms 500ms 1s

3.6%-4.1%/1.6%-2.1%/5.3-6.0% 0.6%-0.9%/0.3%-0.4% 0.1%-0.3%/0.0%-0.1% 0.0%-0.1%/0.0%-0.1%/0.0%-0.3% 0.0%/0.0%/0.0% 0.0%/0.0%/0.0%

12.4-12.7MB/11.2-11.6MB 13.3MB/12.3MB 13.5MB/11.7MB 13.5MB/12.2MB 13.5MB/12.2MB 13.5MB/12.2MB

1ms 3.6%-4.1%/1.6%-2.1%/5.3-6.0%

AP重启 0.639 0.611 0.616 0.603

AP重启完毕至移动终端收到解认证信号 0.041 0.037 0.037 0.049

移动终端收到解认证信号至开始扫描 0.1 0.1 0.1 0.1

扫描附近WLAN信号 1.466 1.453 1.469 1.452

移动终端发现AP口令已更新 3.255 2.54 2.913 3.309 6(+)

重新接入WLAN 0.2 0.2 0.21 0.2

合计 5.701 4.941 5.345 5.713

重新接入WLAN 4.163 0.198 2.692 0.2 0.202 0.197

10ms 0.6%-0.9%/0.3%-0.4%

AP重启 0.609 0.621 0.614 0.603 0.615

AP重启完毕至移动终端收到解认证信号 0.4 0.039 0.14 0.048 0.036

移动终端收到解认证信号至开始扫描 0.1 0.1 0.101 0.1 0.1

扫描附近WLAN信号 1.437 1.453 1.452 1.469 1.457

移动终端发现AP口令已更新 3.317 3.077 2.889 3.36 2.768 10(+)

重新接入WLAN 2.724=2.526+0.199-0.001 0.2 0.21 0.212 0.205

合计 8.587 5.49 5.406 5.792 5.181

重新接入WLAN 0.202 0.199 0.204 0.203 2.679 0.204 0.204 0.198 2.692 0.207

100ms 0.1%-0.3%/0.0%-0.1%

AP重启 0.601 0.626 0.603 0.624 0.607 0.608

AP重启完毕至移动终端收到解认证信号 0.043 0.037 0.052 0.033 0.041 0.044

移动终端收到解认证信号至开始扫描 0.101 0.1 0.101 0.1 0.1 0.102

扫描附近WLAN信号 1.445 1.477 1.464 1.457 1.446 1.444

移动终端发现AP口令已更新 2.58 2.927 3.32s 2.723 3.483 2.871 14(+)

重新接入WLAN 0.201 2.698=1+1.493+0.205 2.708=1+1.497+0.211 0.205 0.204 0.203

合计 4.971 7.865 8.248 5.142 5.881 5.272

重新接入WLAN 0.205 0.206 2.699 2.681 2.682 2.687 0.2 0.219 0.203 2.685 2.686 0.199 0.202 0.201

150ms

AP重启 0.622

AP重启完毕至移动终端收到解认证信号 0.033

移动终端收到解认证信号至开始扫描 0.1

扫描附近WLAN信号 1.442

移动终端发现AP口令已更新 3.293

重新接入WLAN 0.199

合计 5.689

175ms

AP重启 0.612

AP重启完毕至移动终端收到解认证信号 0.04

移动终端收到解认证信号至开始扫描 0.101

扫描附近WLAN信号 1.447

移动终端发现AP口令已更新 3.064

重新接入WLAN 0.215

合计 5.479

200ms

AP重启 0.618 0.611 0.607 0.616 0.614

AP重启完毕至移动终端收到解认证信号 0.036 0.04 0.052 0.04 0.04

移动终端收到解认证信号至开始扫描 0.1 0.101 0.101 0.101 0.101

扫描附近WLAN信号 1.445 1.428 1.436 1.442 1.429

移动终端发现AP口令已更新 0.189 0.003 0.187 0.183 0.196 13(-)2(++)

重新接入WLAN 0.188 0.191 2.684=1+1.48+0.204 0.21 0.206

合计 2.576 2.374 5.067 2.592 2.586

重新接入WLAN 0.197 2.691 2.686 0.206 2.685 2.693 0.23 0.193 0.175 0.202 0.196 0.198 0.207 0.182 0.211

300ms 0.0%-0.1%/0.0%-0.1%/0.0%-0.3%

AP重启 0.611 0.611 0.603 0.723 0.619

AP重启完毕至移动终端收到解认证信号 0.053 0.04 0.049 0.014 0.037

移动终端收到解认证信号至开始扫描 0.1 0.101 0.1 0.1 0.101

扫描附近WLAN信号 1.433 1.428 1.437 1.452 1.46

移动终端发现AP口令已更新 0.18 0.196 0.189 0.174 0.167 14(-)1(++)

重新接入WLAN 0.216 2.696=1+1.49+0.206 0.212 2.834=1+1.49+0.344 0.219

合计 2.593 5.072 2.59 5.297 2.603

重新接入WLAN 0.202 0.213 0.183 2.687 0.183 0.21 0.192 0.213 0.208 2.69 0.183 2.694 0.195 4.979 0.218

400ms 0.0%-0.1%/0.0%-0.1%/0.0%-0.2%

AP重启 0.615 0.611 0.614 0.616 0.615 0.62

AP重启完毕至移动终端收到解认证信号 0.037 0.04 0.041 0.049 0.037 0.032

移动终端收到解认证信号至开始扫描 0.1 0.101 0.1 0.1 0.1 0.1

扫描附近WLAN信号 1.438 1.44 1.427 1.441 1.436 1.442

移动终端发现AP口令已更新 0.39 0.384 0.001 0.378 0.389 0.388 14(-)1(++)

重新接入WLAN 2.684=1+1.482+0.202 2.693=1+1.488+0.205 0.183 2.687=1+1.487+0.2 0.237 2.698=1+1.49+0.208

合计 5.264 5.269 2.366 5.271 2.814 5.28

重新接入WLAN 2.703 2.702 2.689 0.187 0.214 2.692 0.203 2.689 0.197 0.214 2.725 2.707 0.205 0.209 2.673

500ms 0.0%/0.0%/0.0%

AP重启 0.607 0.607 0.604 0.62 0.611 0.606 0.62 0.611 0.611

AP重启完毕至移动终端收到解认证信号 0.044 0.045 0.044 0.036 0.044 0.053 0.032 0.044 0.044

移动终端收到解认证信号至开始扫描 0.101 0.101 0.1 0.1 0.102 0.102 0.101 0.102 0.101

扫描附近WLAN信号 1.433 1.432 1.437 1.438 1.44 1.44 1.43 1.428 1.422

移动终端发现AP口令已更新 0.398 0.39 0.389 0.384 0.385 0.376 0.397 0.394 0.401 10(-)1(++)

重新接入WLAN 0.201 0.202 0.2 0.206 0.174 0.204 2.756=1+1.515+0.241 2.722=1+1.516+0.206 2.69=1+1.49+0.2

合计 2.784 2.777 2.774 2.784 2.756 2.781 5.336 5.301 5.269

重新接入WLAN 2.698 0.206 0.2 2.684 0.212 2.692 0.204 2.685 2.696 0.211 2.686

1s 0.0%/0.0%/0.0%

AP重启 0.606 0.6 0.608 0.615 0.612

AP重启完毕至移动终端收到解认证信号 0.057 0.048 0.044 0.041 0.044

移动终端收到解认证信号至开始扫描 0.1 0.102 0.1 0.1 0.1

扫描附近WLAN信号 1.453 1.46 1.486 1.433 1.457

移动终端发现AP口令已更新 0.357 0.364 3.678 0.839 0.364 9(-)5(+)1(++)

重新接入WLAN 0.209 2.688=1+1.49+0.198 2.696=1+1.49+0.206 2.769=1+1.48+0.289 0.197

合计 2.782 5.262 8.612 5.797 2.774

重新接入WLAN 2.687 0.176 0.206 2.676 1.805 2.684 0.184 0.216 0.203 2.678 0.213 0.183 0.197 2.689 1.625

现有WLAN认证方案：

在较长一段时间内设置一个固定的、相对比较简单的口令。

现有方案存在的问题：

口令容易泄露：

1. 口令设置过于简单，很容易被攻击者猜到；
2. 一旦授权了某个用户，这个用户在口令更换之前都知道口令。

目标：

口令泄露可能导致（1）非法用户未经许可接入WLAN，（2）合法用户被诱骗接入非法AP，以及（3）用户传输的数据被未经授权的用户获得。因此口令需要得到保护。

对于访客WLAN，管理员可能只希望赋予访客时间有限的WLAN访问权。

提出：具有有效期的口令，每位用户的口令都只在其被授权的时间段内有效。

方案：

原理：口令按一定的算法产生并且每天变化，用户只能计算在授权期内的口令。一旦过了授权期，用户就无法计算相应的口令。

算法涉及三类参数（设这些参数的长度均为32字节）：秘密参数P[0]和S，以及公开参数O。

算法分为两部分：正向杂凑和逆向杂凑（设杂凑值也为32字节），而后将两者亦或。

正向杂凑递推式：P[i] = H(P[i-1] XOR O[i])

P[1] = H(P[0] XOR O[1]) P[2] = H(P[1] XOR O[2]) ...... P[365] = H(P[364] XOR O[365])

逆向杂凑一般表达式：H^(366-i)(S)

H^365(S) H^364(S) ....... H(S)

两者异或得到每天的口令：

P[1] XOR H^365(S) P[2] XOR H^364(S) ....... P[365] XOR H(S)

安全性分析：

1. 攻击者的猜测复杂度分析
2. 假设攻击者没有获得过口令

攻击者有两种攻击方法：

1. 每天猜测口令，猜测复杂度为365\*32字节，得到全部天数的所有口令。
2. 猜测P[0]和S，猜测复杂度为64字节，得到全部天数的所有口令。

假设攻击者找到P’[0]和S’，使得P’[1] XOR H^365(S’)正好等于P[1] XOR H^365(S)，即找到一对P[0]和S的碰撞，使用这对P[0]和S能够计算得到正确的第一天的口令。此种搜寻的复杂度为32字节，但是攻击者无法将这种碰撞推广到其他天数。

这一结论即使在没有公开参数的情况下也成立：

假设攻击者找到P’[0]、S’，使得正好等于第一天的口令：

H(P’[0]) XOR H^365(S’) = H(P[0]) XOR H^365(S)

但是，却无法用这组数来计算第二天的口令，因为：

H^2(P’[0]) XOR H^364(S’) = H^2(P[0]) XOR H^364(S)

这一等式几乎不会成立。

1. 假设攻击者获得过口令：

攻击者猜测秘密参数S，猜测复杂度为32字节，得到剩余天数的所有口令。

1. 和设置固定随机数口令（设口令的长度为32字节）对比：

攻击者从未获得口令，猜测口令，猜测复杂度为32字节。

攻击者曾经获得过口令，不需要再猜测了，猜测复杂度为0。

1. 和每天重新设置随机数口令（设口令的长度为32字节）对比：

攻击者从未获得过口令，每天猜测口令，猜测复杂度为365\*32字节，得到全部天数的所有口令。

攻击者曾经获得过口令，猜测剩余天数的口令，猜测复杂度为剩余天数\*32字节，得到剩余天数的所有口令。

从猜测口令上看，新解决方案与该方案的安全强度一致。

但是如果某位访客被授予不同时间段的访问权， 如在1月1日和12月31日均被授予访问权，而该访客又能获得全年度所有的公开参数，该访客就能得到全年度的所有口令。而每天重新设置随机数口令就不存在此种情形。

从这一点上说，新解决方案的安全强度稍差于每天重新设置随机数口令的方案。

实现：

1. 现有方案的实现：
2. AP端：

WLAN的SSID、认证方式和口令等配置选项以键值对的形式存储于配置文件中。

无线连接服务读取配置文件，根据配置文件确定使用何种认证方式以及认证过程中使用的口令等。

1. 移动终端：

用户输入口令后将口令和对应的SSID同样以键值对的形式存储于配置文件中。

无线连接客户端使用存储的口令连接对应SSID的WLAN。

1. 新方案的实现：
2. AP端：

在现有配置文件和无线连接服务的基础上，增加口令维护程序，专门用于每天口令的更新。口令维护程序每天计算新的口令，并将口令写入无线连接服务的配置文件，这样无线连接程序就会使用新的口令和移动终端握手了。

1. 秘密参数的存放

32字节长的秘密参数P[0]和S存储于配置文件中，均以base64编码的形式存储。

此外，配置文件还需要存储每天新生成的P[i]。

1. 公开参数的存放

365个32字节长的公开参数O存储于另一配置文件中，同样以base64编码的形式存储。

1. 口令维护程序的日常工作

每天零点，口令维护程序首先从存放公开参数的配置文件中读取当天的公开参数O[i]。然后从存放秘密参数的配置文件中读取前一天的P（最后一行记录的数据就是P[i-1]），计算P[i] = H(P[i-1] XOR O[i])。接着从其读取S，计算H^(366-i)(S)。最后计算P[i] XOR H^(366-i)(S)得到今天的口令。

将口令以及当天的公开参数O以base64编码的形式写入无线连接服务的配置文件，供无线连接服务调用。

公开参数通过Beacon帧广播出去，OpenWrt的代码是开源的，可以通过修改OpenWrt的代码实现Beacon帧对公开参数的支持。

为了解决AP系统时间和终端系统时间不同步的问题，Beacon帧广播公开参数的同时广播系统时间。

完成这些工作后，口令维护程序在其自身的配置文件中增加当天的P（最后一行记录的数据变成了P[i]）。

1. 口令维护程序的年初处理

跨年之前，WLAN管理员需要将下一年的秘密参数P[0]和S，以及365个公开参数通过WLAN管理网站以文件的形式写入AP。

每年第一天零点，口令维护程序读取新一年的配置文件：如果该文件不存在，或者文件中存放的数据不合法，口令维护程序使AP宕机；否则口令维护程序使用新一年的配置文件替换上一年的配置文件。

然后进行每天的日常工作。

1. 口令维护程序的初始化处理

AP容易发生断电，AP断电后，口令维护程序重新运行。口令维护程序首先需要读取配置维护程序的最后一行记录，检查是否是当天的P。

如果是当天的P，说明当天的日常工作已经完成，口令维护程序直接启动无线连接服务即可；

如果不是当天的P，说明当天的日常工作尚未完成。如果AP被断电多日，最后一个P可能是若干天之前的P，那么口令维护程序首先需要计算至前一天的P。最后执行口令维护程序的日常工作，得到当天的口令。

1. 管理员分发口令

由于管理员知道所有的秘密参数和公开参数，因此管理员可以获得任意一段时期的口令，并分发给用户。

注：管理员的权限可能太大，因为管理员能够计算得到任意一天的口令。

1. 移动终端：

移动终端使用定制的应用连接无线网。

1. 访客口令的获得及存放

分发给用户的口令包含以下三部分：上一天的P[i]，授权时长，以及授权结束时S进行若干次杂凑运算后的值S’。

由于口令比较复杂，因此将口令存放在二维码中。管理员可以将二维码显示在显示屏上、打印到纸上、或者通过邮件发送到用户的指定邮箱，供用户扫描。

应用收到口令后，将口令存储于本地。

此外，应用需要获得WLAN的SSID，SSID可以在分发口令时一并分发。

1. 获取公开参数

用户启动应用时，应用首先判断终端是否已接入WLAN。

如果终端未接入WLAN，则开始接收Beacon帧，解析SSID。

如果收到指定SSID的Beacon帧时，从Beacon帧中解析系统时间和公开参数O[i]。

如果AP系统时间和终端系统时间差距过大，则拒绝连接这一AP，否则使用AP的系统时间作为当前时间进行授权状态检测。

如果应用未收到指定SSID的Beacon帧，则每隔固定时间继续接收。

1. 授权状态检测

用户以AP的系统时间作为当前时间检测移动终端的授权状态：检测当前时间是否超过授权期，如果超过授权期，则提示用户，不再进行后续操作；

如果还在授权期范围内，再从口令文件中读取口令文件中最后一条P，如果最后一条P不是今天或者昨天的P，说明用户昨天未接入WLAN，应用无法计算今天的口令，则提示用户，不再进行后续操作。

如果最后一条P是昨天的P，说明用户今天未接入过WLAN，进行第4步；

如果最后一条P是今天的P，说明用户今天接入过WLAN，进行第5步。

1. 用户今天未曾接入过WLAN

首先从口令文件中读取上一天的P，计算P[i] = Hash(P[i-1] XOR O[i])，然后从口令文件中读取S’，进行一定次数的杂凑运算，最后与P[i]异或，得到今天的口令。然后以今天的口令接入指定SSID的WLAN。

假设用户在第101天被授权接入WLAN，授权期为10天，那么它将收到P[99]，10和S’ = H^256(S)，他可按如下公式计算口令：

第一天：

P[101] = H(P[100] XOR O[101])

Password = P[101] XOR H^9(S’) = P[101] XOR H^265(S)

第二天：

P[102] = H(P[101] XOR O[102])

Password = P[102] XOR H^8(S’) = P[102] XOR H^264(S)

……

第十天：

P[110] = H(P[109] XOR O[110])

Password = P[110] XOR S’ = P[110] XOR H^256(S)

如果移动终端接入指定SSID的WLAN，说明获得的公开参数正确，口令计算过程无误，增加今天的P至口令文件的最后一行。

1. 用户今天接入过WLAN

首先从口令文件中读取今天的P，然后从口令文件中读取S’，进行一定次数的杂凑运算，最后与P[i]异或，得到今天的口令。然后以今天的口令接入指定SSID的WLAN。

1. 接入后处理

终端接入WLAN后，终端开始每隔固定时间检测连接状态。

如果连接断开，则重复第2步。

附：无线接入流程图：

