

组织：中国互动出版网 (<http://www.china-pub.com/>)

RFC 文档中文翻译计划 (<http://www.china-pub.com/compters/emook/aboutemook.htm>)

E-mail: ouyang@china-pub.com

译者：王奎 (pywang wangtianzhi@263.net)

译文发布时间：2001-12-28

版权：本翻译文档可以用于非商业用途自由转载，但必须保留本文档的翻译及组织信息。

Network Working Group
Request for Comments: 1334

B. Lloyd
L&A
W. Simpson
Daydreamer
October 1992

PPP 身份验证协议

(RFC1334——PPP Authentication Protocols)

备忘录状态

此 RFC 为 internet community 详细说明了 IAB 标准跟踪协议，并且请求讨论和建议以便改进。请参考 IAB Official Protocol Standards 的当前版本，确保这个协议陈述和状态的标准化。此备忘录的分发不受限制。

摘要

点到点协议 (the Point-to-Point Protocol) 提供了一种在点到点链路上封装网络层协议信息的方法。PPP 也定义了可扩展的链路控制协议(Link Control Protocol)，它 (Link Control Protocol) 使用验证协议磋商在链路上传输网络层协议前验证链路的对端。这个文档定义了两种验证协议：密码验证协议 (the Password Authentication Protocol) 和挑战—握手验证协议 (the Challenge-Handshake Authentication Protocol)。此 RFC 是 IETF(the Internet Engineering Task Force)的 PPP 协议工作组的成果。关于这个备忘录的建议请提交给：ietf-ppp@ucdavis.edu邮件列表。

目录

1. 介绍.....	2
1.1 要求说明书.....	2
1.2 术语.....	2
2. 密码验证协议.....	3
2.1 配置选项格式.....	3
2.2 包格式.....	4
3.1 配置选项格式.....	7
3.2 包格式.....	7
安全考虑.....	10

参考文献.....	10
致谢.....	11
主席地址.....	11
作者地址.....	11
完整版权说明.....	11
致谢.....	12

1. 介绍

PPP 有三个主要的组成部分：

1. 在串行链路上封装数据报 (datagrams) 的方法。
2. 建立, 配置和测试数据链路连接 (the data-link connection) 的 LCP 协议 (Link Control Protocol)。
3. 建立和配置不同网络层协议的一组 NCP 协议 (Network Control Protocol)。

为了在点到点链路 (point-to-point link) 上建立通信, PPP 链路的一端必须在建立阶段 (Establishment phase) 首先发送 LCP 包 (packets) 配置数据链路。在链路建立后, 在进入网络层协议阶段前, PPP 提供一个可选择的验证阶段。

默认的, 身份验证不是强制的。如果希望进行链路的身份验证, 则实现者必须在建立阶段指明身份验证一协议配置选项。

这些协议主要是为通过交换网 (switched circuits) 或者拨号线 (dial-up lines) 连接到 PPP 网络服务器的主机和路由器服务的, 但是也可以被用到专用链路 (dedicated links) 中。服务器在为网络层磋商选择选项时可以对连接的主机或路由器进行身份验证。

此文档定义了 PPP 身份验证协议。链路建立和验证阶段, 和验证协议配置选项定义在 PPP 协议中[1]。

1.1 要求说明书

在本文档中, 用以下几个词来表示说明书的要求, 这些词一般以大写字体书写。

MUST

这个词表示在此说明书中是绝对要求的。

MUST NOT

这个词组表示在此说明书中是绝对禁止的。

SHOULD

此词表示在此说明书中是推荐的。

MAY

此词表示在此说明书中是可选的。

1.2 术语

本文档中, 频繁使用以下术语:

authenticator—验证者:

要求验证的链路端点。验证者说明了在链路建立阶段使用的验证协议。

Peer—点对点链路的另一端:

正在被验证者验证的一端。

Silently discard—静静地丢弃

丢弃 packet 而不进行进一步的处理。执行（这个动作）应该提供记录错误，包括丢弃 packet 的内容，的容量，并且应该在一个统计计数器中记录这一事件。

2. 密码验证协议

密码验证协议（PAP）提供了一种简单的方法，可以使对端（peer）使用 2 次握手建立身份验证。这个方法仅仅在链路初始化时使用。

链路建立阶段完成后，对端不停地发送 Id/Password 对给验证者，一直到验证被响应或者连接终止为止。

PAP 不是一个健壮的身份验证方法。密码在电路上是明文发送的，并且对回送或者重复验证和错误攻击没有保护措施。对端控制着尝试的频率和时间。

包含健壮验证方法（例如 CHAP，下面描述）的任何实现者必须提供商议优先于 PAP 的方法。

这个验证方法最适合用在使用有效的明文密码在远程主机上模拟登陆的地方了。通过这种用法，这个方法向普通用户要登陆远程主机提供了一种安全的类似级别。

实现注意：要限制暴露在 PPP 链路上传输明文密码和避免在整个网络上发送明文密码是可能的。如果远程主机密码是以单向转换值保存的，并且转换函数的算法是在当地主机上完成的，则明文密码应该在和远程主机的转换密码比较前在本地转换。

2.1 配置选项格式

下面是关于 PAP 的验证—协议配置选项格式的总结。各个域由左到右传输。

```

0           1           2           3
0 1 2 3 4 5 6 7 8 9 0 1 2 3 4 5 6 7 8 9 0 1 2 3 4 5 6 7 8 9 0 1
+---+---+---+---+---+---+---+---+---+---+---+---+---+---+---+---+
|   Type   |   Length   |   Authentication-Protocol   |
+---+---+---+---+---+---+---+---+---+---+---+---+---+---+---+---+

```

类型

3

长度

4

验证—协议

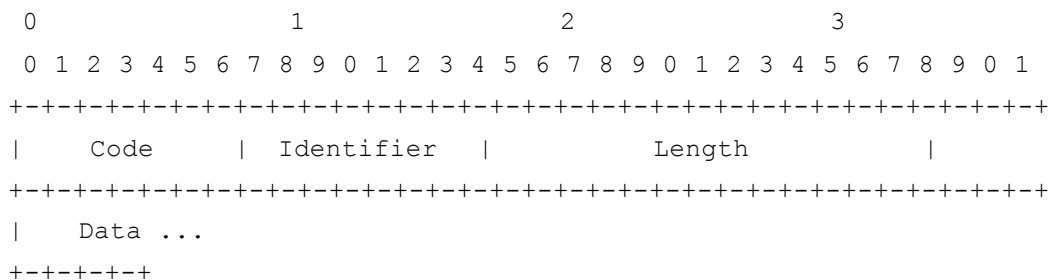
c023（对于 PAP）

数据

没有数据域

2.2 包格式

一个 PAP 包是完全封装在 PPP 数据链路层帧(协议域是 c023 代表 PAP)的信息域中的。下面是 PAP 包格式的总结。各个域由左到右传输。



代码

代码域是一个字节，代表 PAP 包的类型。PAP 代码分配如下：

- | | |
|---|----------------------|
| 1 | Authenticate-Request |
| 2 | Authenticate-Ack |
| 3 | Authenticate-Nak |

标识符

标识符是一个字节，用于匹配请求和响应。

长度

长度域是两个字节，代表 PAP 包的长度，包括代码域，标识符和数据域。超出长度域指定的字节被认为是数据链路层的填料，在接收端应该忽略掉。

数据

数据域是零个或多个字节。数据域的格式由代码域决定。

2.2.1 Authenticate-Request

描述

Authenticate-Request 包用来启动 PAP。在验证阶段链路的一端必须传输代码域为 1（验证—请求）的 PAP 包。直到接收到一个有效的响应包或者可选的重试计数器超时，验证—请求包必须不停地发送。

验证者应该期待对端发送一个 Authenticate-Request 包。一旦接收到 Authenticate-Request 包，必须返回某种验证响应（下面描述）。

实现注意：因为 Authenticate-Request 包可能会丢失，所以在完成验证阶段后验证者必须允许重复的 Authenticate-Request 包。在验证阶段完成（部分信息可能不同）后，在协议阶段必须返回相同的响应代码。在另外的阶段接到的任何 Authenticate-Request 包必须被静静地处理掉。

如果 Authenticate-Nak 包丢失，和验证者终止链路，则 LCP Terminate-Request 包和 Terminate-Ack 包提供一个可选择的方法表示验证失败。

下面是 **Authenticate-Request** 包格式的总结。各个域由左到右传输。

```

0             1             2             3
0 1 2 3 4 5 6 7 8 9 0 1 2 3 4 5 6 7 8 9 0 1 2 3 4 5 6 7 8 9 0 1
+-----+-----+-----+-----+-----+-----+-----+-----+
|   Code   | Identifier |           Length           |
+-----+-----+-----+-----+-----+-----+
| Peer-ID Length| Peer-Id ...
+-----+-----+-----+-----+
| Passwd-Length | Password ...
+-----+-----+-----+-----+

```

代码

1 **Authenticate-Request**。

标识符

标识符是一个字节，用于匹配请求和回应。每次发送一个 **Authenticate-Request** 包，标识符域必须改变。

Peer-ID-Length

Peer-ID-Length 域是一个字节，代表 **Peer-ID** 域的长度。

Peer-ID

Peer-ID 域是零个或多个字节，代表被验证端的名字。

Passwd-Length

Passwd-Length 域一个字节，代表 **Password** 域的长度。

Password

Password 域是零个或者多个字节，是用来验证的密码。

2.2.2 Authenticate-Ack and Authenticate-Nak

描述

如果在接收的 **Authenticate-Request** 包中的 **Peer-ID/Password** 对是可识别的和可接受的，则验证者必须发送一个代码域是 2 (**Authenticate-Ack**) 的 **PAP** 包。

如果在接收的 **Authenticate-Request** 包中的 **Peer-ID/Password** 对是不可识别的和不可接受的，则验证者必须发送一个代码域是 3 (**Authenticate-Nak**) 的 **PAP** 包，并且应该终止链路。

下面是 **Authenticate-Ack** 包和 **Authenticate-Nak** 包格式的总结。各个域由左到右传输。

```

0             1             2             3
0 1 2 3 4 5 6 7 8 9 0 1 2 3 4 5 6 7 8 9 0 1 2 3 4 5 6 7 8 9 0 1
+-----+-----+-----+-----+-----+-----+-----+-----+
|   Code   | Identifier |           Length           |
+-----+-----+-----+-----+-----+-----+
| Msg-Length | Message ...
+-----+-----+-----+-----+

```

代码

2 Authenticate-Ack;

3 Authenticate-Nak

标识符

标识符域是一个字节，用于匹配请求和回应。此域必须从引起这次回应的 Authenticate-Request 包标识符域复制过来的。

Msg-Length

Msg-Length 域是一个字节，代表 Message 域的长度。

Message

Message 域是零个或者多个字节，并且它的内容依靠于实现者。它是可读的，不得影响协议的操作。建议在 Message 中包含可显示的 ASCII 字符（32-126）。扩展字符集的机制是进一步研究的主题。

3 Challenge-Handshake Authentication Protocol

CHAP 用于使用 3 次握手周期性的验证对端。在链路建立初始化时这样做，也可以在链路建立后任何时间重复验证。

在链路建立完成后，验证者向对端发送一个“challenge”信息。对端使用一个“one-way-hash”函数计算出的值响应这个信息。验证者使用自己计算的 hash 值校验响应值。如果两个值匹配，则验证是承认得，否则连接应该终止。

CHAP 通过使用递增的标识符和可变得挑战值提供了防止回送攻击的保护。使用重复挑战目的是任一个攻击的暴露时间。验证者控制着挑战的频率和时间。这种验证方法依靠只有验证者和对端知道的秘密（secret）。这个秘密（secret）不在链路上传播。这种方法最可能用的地方是相同的秘密容易访问链路的两端。

实现注意：CHAP 要求秘密是明文形式的。为了避免在网络的其他链路上发送秘密，建议在中心服务器上检查 challenge 和 response 值，而不是在每一个网络访问服务器上检查。另外，秘密应该以可逆转的加密形式发送到服务器上。

CHAP 算法要求秘密的长度必须至少一个字节。秘密至少应该和选择好的密码一样大小和不可猜。比较好的是秘密应该至少是选择的哈希算法的哈希值的长度（对于 MD5 是 16 个字节）。这样保证了足够大的范围使得秘密提供了防止穷尽搜索攻击的保护措施。

选择 one-way 哈希算法使得要想从已知的 challenge 和 response 值得出秘密的计算是不可行的。

Challenge 值应该符合两个标准：唯一性和不可预测性。每一个 challenge 值应该是唯一的，因为使用与相同秘密联系的 challenge 值的副本可以让攻击者利用前一个截获得响应包响应。由于希望可以使用相同的秘密在不同区域中验证服务器，challenge 应该具有全局和暂时的唯一性。每一个 challenge 值也应该是不可预测的，否则攻击者欺骗对端响应一个可预测的未来 challenge，然后用这个响应伪装成对端欺骗验证者。尽管象 CHAP 这样的协议不能够防止实时的窃听攻击，但是使用唯一的和不可预测的 challenge 可以防止一定范围的能动攻击。

关于唯一性来源和产生分歧概率的讨论包含在 Magic-Number 配置选项中。

3.1 配置选项格式

下面是 Challenge-Handshake 验证协议使用的 Authentication-Protocol 配置选项格式的总结。各个域由左到右传输。

```

0           1           2           3
0 1 2 3 4 5 6 7 8 9 0 1 2 3 4 5 6 7 8 9 0 1 2 3 4 5 6 7 8 9 0 1
+-----+-----+-----+-----+
|   Type   |   Length   |   Authentication-Protocol   |
+-----+-----+-----+-----+
| Algorithm |
+-----+-----+-----+

```

类型

3

长度

5

Authentication-Protocol

c223 Challenge-Protocol Authentication Protocol

算法

算法域是一个字节，代表所使用的 one-way 哈希算法。CHAP 算法域的最新值在最近的“Assigned Numbers” RFC[2]中有详细说明。当前的值分配如下：

0-4 unused(保留)

5 MD5[3]

3.2 包格式

CHAP 包封装在 PPP 数据联络层帧的信息域中，它的协议域是 c223。下面是 CHAP 包格式的总结。各个域由左到右传输。

```

0           1           2           3
0 1 2 3 4 5 6 7 8 9 0 1 2 3 4 5 6 7 8 9 0 1 2 3 4 5 6 7 8 9 0 1
+-----+-----+-----+-----+
|   Code   | Identifier |           Length           |
+-----+-----+-----+-----+
| Data ...
+-----+-----+

```

代码

代码域是一个字节，代表 CHAP 包的类型。CHAP 代码分配如下：

1 Challenge

2 esponse

3 Success

4 Failure

标识符

标识符是一个字节，用于匹配 challenge,response 和 replies。

长度

长度域是两个字节，代表 CHAP 包的长度，包括 Code,Identifier,Length 和 Data。超出这个长度的字节应该被认为是链路层的填料，在接收端应该被忽略。

数据

数据域是零个或者多个字节。它的格式由 code 域决定。

3.2.1 Challenge 和 Response

描述

Challenge 包用来启动 CHAP。验证者必须发送一个代码域是 1 的 CHAP 包。一直到接收到有效的响应包或者重试计数器超时，必须不停发送 Challenge 包。

在网络层协议阶段的任一个时间也可以发送 Challenge 包确保连接没有改变。

在验证阶段和网络层协议阶段对端应该期待 Challenge 包。无论何时接到 Challenge 包，对端必须发送一个代码域是 2 的 CHAP 包。

无论何时验证者接到一个 Response 包，则它比较 Response 值和自己计算的期待值是否相同。在比较的基础上，验证者必须发送一个 Success 或者 Failure 包。

实现注意：因为 Success 包有可能丢失，验证者必须在完成验证阶段后允许重复的 Response 包。为了防止名字和秘密的泄漏，在验证阶段后，任何具有当前 Challenge 标识符的 Response 包必须返回相同的响应代码。在其他阶段接到的任何 Response 包必须被静静地丢掉。

如果 Failure 包丢失和验证者终止链路，则 LCP 的 Terminate-Request 包和 Terminate-Ack 包提供了另一种代表验证失败的方法。

下面是 Challenge 和 Response 包格式的总结。各个域由左到右传输。

```

0          1          2          3
0 1 2 3 4 5 6 7 8 9 0 1 2 3 4 5 6 7 8 9 0 1 2 3 4 5 6 7 8 9 0 1
+---+---+---+---+---+---+---+---+---+---+---+---+---+---+---+---+
|   Code   | Identifier |           Length           |
+---+---+---+---+---+---+---+---+---+---+---+---+---+---+---+---+
| Value-Size | Value ...
+---+---+---+---+---+---+---+---+---+---+---+---+---+---+---+---+
| Name ...
+---+---+---+---+---+---+---+---+---+---+---+---+---+---+---+---+

```

代码

1 Challenge

2 Response

标识符

标识符是一个字节，每发送一个 Challenge 包标识符必须改变。Response 包的标识符必须从引起这个响应的 Challenge 包的标识符复制过来的。

Value-Size

此域是一个字节，代表 Value 域的长度。

Value

Value 域是一个或多个字节。最重要的字节先传输。

Challenge Value 是一个可变的字节流。上面讲述了 Challenge Value 唯一性的重要性以及它和秘密的关系。每次发送 Challenge 包必须改变 Challenge Value。Challenge Value 的长度依靠于产生字节所使用的方法，独立于所用的哈希算法。

Response Value 是在字节流上用单向哈希算法计算得出的，字节流包含 Identifier,后面是 secret,再后面是 Challenge Value。Response Value 的长度依靠于所用的哈希算法（对于 MD5 是 16 个字节）。

名字

名字域是一个或多个字节，代表发送包的系统的标识。对这个域的内容没有限制。例如，它可以是 ASCII 字符串或者是 ASN.1 语法中的全局唯一标识。名字不应该是以 NULL 或者 CR/LF 结尾的。大小由长度域决定。

因为 CHAP 可以验证许多不同的系统，所以名字域的内容可以用作在秘密数据库查询秘密的关键字。这也可以在每个系统上支持更多的 Name/Secret 对。

3.2.2 Success 和 Failure

描述

如果在 Response 包中的 Value 等于期待的值，则验证者必须发送一个代码域是 3 (Success) 的 CHAP 包。

如果在 Response 包中的 Value 不等于期待的值，则验证者必须发送一个代码域是 4 (Failure) 的 CHAP 包，并且应该终止链路。

下面是 Success 和 Failure 包格式的总结。各个域由左到右传输。

```

0           1           2           3
0 1 2 3 4 5 6 7 8 9 0 1 2 3 4 5 6 7 8 9 0 1 2 3 4 5 6 7 8 9 0 1
+---+---+---+---+---+---+---+---+---+---+---+---+---+---+---+---+
|   Code   | Identifier |           Length           |
+---+---+---+---+---+---+---+---+---+---+---+---+---+---+---+---+
| Message ...
+---+---+---+---+---+---+---+---+---+---+---+---+---+---+

```

代码

3 Success

4 Failure

标识符

标识符是一个字节，用于匹配 request 和 replies。标识符必须从引起这个响应的 Response 包的标识符复制过来的。

Message

Message 域是零个或者多个字节，它的内容依靠实现者。它被设计成可读的，不得影响协议操作。建议在 Message 中包含可显示的 ASCII 字符（32—126）。扩展字符集的机制是进一步研究的主题。大小由长度域决定。

安全考虑

安全问题是此备忘录的主要话题。

PPP 中的验证协议的交互操作很大程度上依靠于实现者。在文档中通篇使用 SHOULD 表明了这点。

例如，一旦验证失败，有些实现者并不终止链路。相反，实现者限制网络层的通信量的类型构造子网，这样反过来允许用户有机会更新秘密或者发邮件给网络管理员说明问题。

对于验证失败没有重试机制。然而，LCP 状态机可以在任何时候重新磋商验证协议，这样就允许了一个新的重试。建议任何用来为验证失败的计数器在成功验证前或者终止失败的链路前不要重置。

不要求验证是双向的或者在两个方向使用相同的协议。在任一个方向上使用不同的协议是完全可以接受的。当然，这依靠于在磋商时指定的协议。

在实践中，在每个 PPP 服务器上有一个数据库，它联合验证信息的用户名字。不期望使用多个方法验证特殊的命名用户。这样使用户容易受到攻击。作为代替的，对于每一个命名用户有一个准确的方法用来验证用户名。如果一个用户在不同的环境下需要使用不同的验证方法，那么应该采用截然不同的用户名，每一个准确代表一个验证方法。

密码和其他的秘密应该保存在各自的端点以至于对它们的访问尽可能的受到限制。理想的，只能是为了完成验证而需要访问的过程可以访问秘密。

应该使用一种机制分发秘密，这种机制能够限制处理秘密实体的数目。理想的，没有通过验证的人不会再得到秘密的内容。使用 SNMP 安全协议[4]可以实现这个目标，但是这样的机制不在这个规范的范围內。

目前正在研究和试验其他的分发机制。SNMP 安全文档很好的概括了对网络的威胁。

参考文献

- [1] Simpson, W., "The Point-to-Point Protocol (PPP)", [RFC 1331](#), Daydreamer, May 1992.
- [2] Reynolds, J., and J. Postel, "Assigned Numbers", [RFC 1340](#), USC/Information Sciences Institute, July 1992.
- [3] Rivest, R., and S. Dusse, "The MD5 Message-Digest Algorithm", MIT Laboratory for Computer Science and RSA Data Security, Inc. RFC 1321, April 1992.

- [4] Galvin, J., McCloghrrie, K., and J. Davin, "SNMP Security Protocols", Trusted Information Systems, Inc., Hughes LAN Systems, Inc., MIT Laboratory for Computer Science, [RFC 1352](#), July 1992.

致谢

此文档的一些内容来自 RFC1172, 它是由 Drew Perkins of Carnegie Mellon University 和 Russ Hobby of the University of California at Davis 共同制定的。

特别感谢 Dave Balenson, Steve Crocker and, James Galvin, 和 Steve Kent, 感谢他们的广泛的解释和建议。

主席地址

可以通过现任主席联系工作组。

Brian Lloyd
Lloyd & Associates
3420 Sudbury Road
Cameron Park, California 95682

Phone: (916) 676-1147

EEmail: brian@lloyd.com

作者地址

关于此备忘录的问题可以直接联系:

William Allen Simpson
Daydreamer
Computer Systems Consulting Services
P O Box 6205
East Lansing, MI 48826-6205

EEmail: Bill.Simpson@um.cc.umich.edu

完整版权说明

Copyright (C) The Internet Society (2001). All Rights Reserved.

只要在所有复本与推导性工作中包含上面的版权声明和这段文字，就可以全部地或者部分地且没有任何限制地复制这篇文档与其译本以及把它提供给其它文档，同样也可以准备、复制、出版与发行推导性工作，而且需要评述此推导性工作否则就得解释它，或者辅助此推导性工作的实现。然而，此文档本身不可以做任何修改，诸如删除版权声明或者因特网社区或其它因特网组织的涉及，除了当需要开发因特网标准的目的时之外且在此种情况下必须遵循在因特网标准过程中定义的版权程序，或者除了当要求把它译成其它语言（即不是英文）的目的时之外。

在上面准予的有限的许可是永久性的，而且因特网社区或者它的继承者或指派者都不会废除它。

在此包含的这篇文档与信息是基于“AS IS”而提供的，并且因特网社区与因特网工程任务组织声明了所有的授权、表达或含义，且包含对任何授权的限制，比如这里信息的使用不会违反任何授权或者出于特殊目的的商品性或適切性的任何含蓄授权。

致谢

感谢因特网社区当前为 RFC 编辑提供了功能基金。