(19)中华人民共和国国家知识产权局

(12)发明专利

(10)授权公告号 CN 1716953 B
(45)授权公告日 2010.09.15

(21)申请号 200410069510.0

(22)申请日 2004.06.28

(73)专利权人 华为技术有限公司
地址 518129 广东省深圳市龙岗区坂田华为总部科研中心 F1-18 楼知识产权部

(72)发明人 周思义

(74)专利代理机构 北京凯特来知识产权代理有限公司 11260
代理人 郑立明

(51)Int.Cl.
H04L 29/06(2006.01)
H04L 9/32(2006.01)

(56)对比文件
CN 1423882 A,2003.06.11, 全文.
WO 03/09181 A1,2003.11.06, 全文.
CN 1483265 A,2004.03.17, 全文.
EP 1267548 A2,2002.12.18, 说明书第 2 页第 0006 段.

审查员 贺秀莲

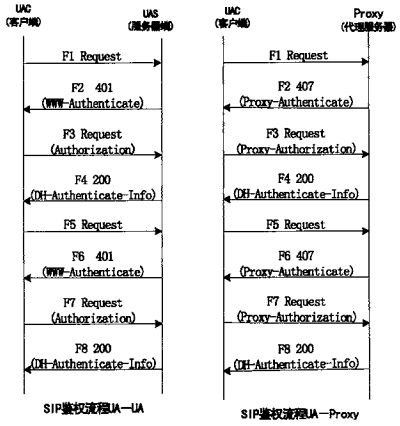
权利要求书 2 页 说明书 10 页 附图 3 页

(54)发明名称

会话初始协议认证的方法

(57)摘要

本发明公开了一种会话初始协议认证的方法,该方法包括:客户端发送不带认证信息的请求消息到服务器端,请求接入;服务器端收到所述请求消息后回送带有服务器端的认证交换信息及服务器端 DH 认证响应信息的响应消息;客户端对收到的响应消息进行认证,认证通过后,发送带有客户端认证信息的请求消息到服务器端;服务器端根据收到的请求消息对用户进行认证,并回送包含服务器端认证信息的响应消息;用户根据收到的包含服务器端认证信息的响应消息验证服务器端的合法性。利用本发明,可以有效地提高 SIP 认证的安全性。



CN 1716953 B

1. 一种会话初始协议认证的方法，其特征在于，所述方法包括：

A、客户端发送不带认证信息的请求消息到服务器端，请求接入；

B、所述服务器端收到所述请求消息后回送带有服务器端的认证交换信息及服务器端 Diffie-Hellman 认证响应信息的响应消息；

C、所述客户端根据收到的响应消息对所述服务器端进行认证，当所述服务器端的认证通过后，发送带有客户端认证信息的请求消息到服务器端；

D、所述服务器端根据收到的带有客户端认证信息的请求消息对所述客户端进行认证，并回送包含服务器端认证信息的响应消息；

E、所述客户端根据收到的所述包含服务器端认证信息的响应消息验证所述服务器端的合法性。

2. 根据权利要求 1 所述的会话初始协议认证的方法，其特征在于，所述步骤 B 之前，还包括：服务器端根据用户名和初始密码生成所述服务器端 Diffie-Hellman 认证响应信息。

3. 根据权利要求 1 所述的会话初始协议认证的方法，其特征在于，所述步骤 C 发送的请求消息中的客户端认证信息包括：客户端 Diffie-Hellman 认证响应信息；或者，客户端的认证交换信息和客户端 Diffie-Hellman 认证响应信息。

4. 根据权利要求 3 所述的会话初始协议认证的方法，其特征在于，所述步骤 C 中发送带有客户端认证信息的请求消息到服务器端之前，还包括：

C1、所述客户端根据所述服务器端的认证交换信息及本端的认证交换信息获取共享密钥；

C2、根据所述共享密钥生成所述客户端 Diffie-Hellman 认证响应信息。

5. 根据权利要求 3 所述的会话初始协议认证的方法，其特征在于，所述步骤 D 中所述服务器端根据收到的带有客户端认证信息的请求消息对所述客户端进行认证包括：

当所述带有客户端认证信息的请求消息的头域中不包括所述客户端的认证交换信息时，所述服务器端使用用户名和初始密码对收到的请求消息进行认证；

当所述带有客户端认证信息的请求消息的头域中包括所述客户端的认证交换信息时，所述服务器端根据所述客户端的认证交换信息以及本端的认证交换信息获取共享密钥，根据所述共享密钥对收到的请求消息进行认证。

6. 根据权利要求 3 所述的会话初始协议认证的方法，其特征在于，所述步骤 D 中回送包含服务器端认证信息的响应消息之前，还包括：

当所述带有客户端认证信息的请求消息的头域中不包括所述客户端的认证交换信息时，根据所述用户名和初始密码生成所述服务器端 Diffie-Hellman 认证响应信息；

当所述带有客户端认证信息的请求消息的头域中包括所述客户端的认证交换信息时，所述服务器端根据所述客户端的认证交换信息以及本端的认证交换信息获取共享密钥，根据所述共享密钥生成服务器端 Diffie-Hellman 认证响应信息。

7. 根据权利要求 1 或 3 所述的会话初始协议认证的方法，其特征在于，所述包含服务器端认证信息的响应消息包括：可选的 Diffie-Hellman 认证信息头域。

8. 根据权利要求 7 所述的会话初始协议认证的方法，其特征在于，所述 Diffie-Hellman 认证信息头域包括：服务器端的认证信息。

9. 根据权利要求 6 所述的会话初始协议认证的方法，其特征在于，所述方法还包括：在

所述客户端和所述服务器端都获取共享密钥后，进行消息交互时，只使用所述共享密钥对交互的消息进行加密。

10. 根据权利要求 1 所述的会话初始协议认证的方法，其特征在于，所述服务器端包括：代理服务器、背靠背服务器、重定向服务器和注册服务器。

会话初始协议认证的方法

技术领域

[0001] 本发明涉及网络安全技术领域，具体涉及一种会话初始协议认证的方法。

背景技术

[0002] 随着互联网及下一代网络的发展，其方便的接入、逐步提高的接入速度、易于扩展的特性、以及丰富的业务功能，受到了运营商及用户的欢迎，但与此同时其安全性方面也逐步受到人们的关注。SIP（会话初始协议）协议作为下一代网络的核心协议在安全性方面也面临着同样的问题，接入认证是解决这一问题的方式之一，已有的 SIP 协议 (RFC3261) 提供了基本的接入认证方式，即所谓的 degist（摘要）认证。

[0003] SIP 协议具有简单、扩展性好及与 Internet 应用紧密结合的特点，仅用 3 条消息 (INVITE、BYE 和 ACK) 和 4 个头域 (To、From、Call-ID 和 Cseq) 就能实现简单的 Internet 电话。SIP 中有客户机和服务器之分。客户机是指为了向服务器发送请求而与服务器建立连接的应用程序。B2B 用户代理 (Back to Back User Agent) 和代理 (Proxy) 中含有客户机。服务器是用于向客户机发出的请求提供服务并回送应答的应用程序。共有四类基本服务器：

[0004] 1. B2B 用户代理服务器：当接到 SIP 请求时它联系用户，并代表用户返回响应。

[0005] 2. 代理服务器：代表其它客户机发起请求，既充当服务器又充当客户机的媒介程序。在转发请求之前，它可以改写原请求消息中的内容。

[0006] 3. 重定向服务器：它接收 SIP 请求，并把请求中的原地址映射成零个或多个新地址，返回给客户机。

[0007] 4. 注册服务器：它接收客户机的注册请求，完成用户地址的注册。用户终端程序往往需要包括用户代理客户机和用户代理服务器。

[0008] SIP 的认证过程是一个类似于 HTTP(HyperText Transfer Protocol) 的无状态的基于 Challenge（问询）的机制 (RFC2617)，基本思路是认证的双方共享用户名和初始密码。在认证的过程中，认证方向被认证方发送 Challenge，被认证方在收到 Challenge 后，将用户名和初始密码经过加密，形成一个字符串，传递给认证方；认证方将自己知道的用户名和密码通过同样的方式进行加密，得到一个字符串，通过比较该字符串和被认证方传递的字符串是否一致来判断用户的密码是否正确。

[0009] 在 SIP 中采用 Digest Scheme（摘要机制）的认证方式，具体流程如图 1 所示。

[0010] 对于 UAS（服务器端），如果需要认证 UAC（客户端），则必须发送 401 Unauthorized 响应，401 Unauthorized 响应表示客户试图未经授权访问受密码保护的资源或者客户。401 响应中必须携带 WWW-Authenticate 头域，UAC 据此显示用户名字 / 密码对话框，然后在填写合适的 Authorization 头后再次发出请求，在 Authorization 头域中携带认证信息。注册服务器和重定向服务器也可以使用 401 响应来进行认证 UAC。

[0011] 对于 Proxy（代理服务器）而言，如果要认证 UAC，则必须采用 407Proxy Authentication Required 响应，407 Proxy Authentication Required 类似于 401，表示客

户必须先经过代理服务器的授权，并必须在其中携带 Proxy-Authenticate 头域。UAC 可以再次发起请求，在 Proxy-Authorization 头域中携带认证信息。

[0012] 当 UAC 因为收到 401 或者 407 响应而重新发起请求时，一般应该使用和上一个请示求相同的 Call-ID，From 头域和 To 头域，但是 Cseq 头域中的序数必须加一，即有相同 Call-ID 的请求必须拥有递增的 Cseq 号。

[0013] 该认证方式只提供最基本的接入认证功能，在网络安全方面存在以下缺陷：

[0014] 1、RFC3261 中的基本 Digest 认证机制只能对 UAC 的 Request 消息发起认证，对 401 或者 407 响应则没有相应的认证机制，所以很容易导致对 UAC 发起 Plain Text（明文）攻击。

[0015] 2、由于在 RFC3261 Digest 所有的认证（只有对 UAC 的 Request 消息发起的认证）过程中都使用了初始密钥，所以容易被监听，分析 (Authorization 和 Proxy-Authorization) 头域，而得出初始密钥，容易导致字典攻击。

[0016] **发明内容**

[0017] 本发明的目的是提供一种会话初始协议认证的方法，以提高网络接入的安全性。

[0018] 本发明的目的是通过以下技术方案实现的：

[0019] 一种会话初始协议认证的方法，包括：

[0020] A、客户端发送不带认证信息的请求消息到服务器端，请求接入；

[0021] B、所述服务器端收到所述请求消息后回送带有服务器端的认证交换信息及服务器端 Diffie-Hellman DH 认证响应信息的响应消息；

[0022] C、所述客户端根据收到的响应消息对所述服务器端进行认证，当所述服务器端的认证通过后，发送带有客户端认证信息的请求消息到服务器端；

[0023] D、所述服务器端根据收到的带有客户端认证信息的请求消息对所述客户端进行认证，并回送包含服务器端认证信息的响应消息；

[0024] E、所述客户端根据收到的所述包含服务器端认证信息的响应消息验证所述服务器端的合法性。

[0025] 所述步骤 B 之前，还包括：服务器端根据用户名和初始密码生成所述服务器端 DH 认证响应信息。

[0026] 所述步骤 C 发送的请求消息中的客户端认证信息包括：客户端 DH 认证响应信息；或者，客户端的认证交换信息和客户端 DH 认证响应信息。

[0027] 所述步骤 C 中发送带有客户端认证信息的请求消息到服务器端之前，还包括：

[0028] C1、所述客户端根据所述服务器端的认证交换信息及本端的认证交换信息获取共享密钥；

[0029] C2、根据所述共享密钥生成所述客户端 DH 认证响应信息，其中所述客户端 DH 认证响应信息为所述客户端认证信息。

[0030] 所述步骤 D 中所述服务器端根据收到的带有客户端认证信息的请求消息对所述客户端进行认证包括：

[0031] 当所述带有客户端认证信息的请求消息的头域中不包括所述用户的认证交换信息时，所述服务器端使用用户名和初始密码对收到的请求消息进行认证；

[0032] 当所述带有客户端认证信息的请求消息的头域中包括所述用户的认证交换信息

时，所述服务器端根据所述用户的认证交换信息以及本端的认证交换信息获取共享密钥，根据所述共享密钥对收到的请求消息进行认证。

[0033] 所述步骤 D 中回送包含服务器端认证信息的响应消息之前，还包括：

[0034] 当所述带有客户端认证信息的请求消息的头域中不包括所述用户的认证交换信息时，根据所述用户名和初始密码生成所述服务器端 DH 认证响应信息；

[0035] 当所述带有客户端认证信息的请求消息的头域中包括所述用户的认证交换信息时，所述服务器端根据所述用户的认证交换信息以及本端的认证交换信息获取共享密钥，根据所述共享密钥生成服务器端 DH 认证响应信息。

[0036] 所述包含服务器端认证信息的响应消息包括：可选的 DH 认证信息头域。

[0037] 所述 DH 认证信息头域包括：服务器端的验证信息。

[0038] 所述方法还包括：在所述客户端和所述服务器端都获取共享密钥后，进行消息交互时，只使用所述共享密钥对交互的消息进行加密。

[0039] 所述服务器端包括：代理服务器、背靠背服务器、重定向服务器和注册服务器。

[0040] 由以上本发明提供的技术方案可以看出，本发明在现有 SIP 基本认证基础上，引入 DH(Diffie-Hellman) 算法，并对 SIP 头域及字段进行扩展，使初始密钥只在第一次交互中使用，在其他的认证过程中使用共享密钥，使初始密钥得到了充分的保护，可以有效地防止字典攻击；同时，当任何一方重启后或者希望更换共享密码时，也可以重新启用校验初始密码的方式，通过使用认证次数计数器，有效地防止向 UAS 或者 Proxy 的 replay（重发）攻击。利用本发明，可以大大提高网络的安全性。

[0041] **附图说明**

[0042] 图 1 是现有技术中 SIP 的认证流程；

[0043] 图 2 是本发明会话初始协议认证的方法的流程图；

[0044] 图 3 是本发明方法中用户接入时的消息流程图。

[0045] **具体实施方式**

[0046] 本发明的核心在于在现有 SIP 基本认证基础上，引入 DH 算法，并对 SIP 头域及字段进行扩展，不仅对用户的请求消息发起认证，而且对服务器端的响应消息也提供相应的认证机制，以便有效地防止对用户发起的 PlainText；同时，在本发明方法中，初始密码只在第一次交互中使用，后续的认证过程都通过共享密码来加密，以便有效地防止字典攻击，而且，当任何一方重启后或者希望更换共享密码时，也可以重新启用校验密码的方式，以便有效地防止 replay 攻击和兼容异常情况。

[0047] 为了使本技术领域的人员更好地理解本发明，下面结合附图和实施方式对本发明作进一步的详细说明。

[0048] 参照图 2，图 2 是本发明方法的详细流程，包括以下步骤：

[0049] 步骤 201：客户端发送不带认证信息的请求消息到服务器端，请求接入。所述服务器端包括：代理服务器、背靠背服务器、重定向服务器和注册服务器。

[0050] 步骤 202：服务器端收到所述请求消息后根据用户名和初始密码生成服务器端 DH 认证响应信息。

[0051] 步骤 203：向客户端回送带有服务器端的认证交换信息及服务器端 DH 认证响应信息的响应消息。

[0052] 步骤 204 :客户端根据服务器端的认证交换信息及本端的认证交换信息获取共享密钥。

[0053] 步骤 205 :根据所述共享密钥生成客户端 DH 认证响应信息。

[0054] 步骤 206 :客户端对收到的响应消息进行认证,认证通过后,发送带有客户端认证信息的请求消息到服务器端。所述带有客户端认证信息的请求消息包括 :可选的客户端的认证交换信息、客户端 DH 认证响应信息。

[0055] 步骤 207 :服务器端根据收到的请求消息对用户进行认证,并回送包含服务器端认证信息的响应消息。所述包含服务器端认证信息的响应消息包括 :可选的 DH 认证信息头域。所述 DH 认证信息头域包括 :服务器端的认证信息。

[0056] 服务器端收到的请求消息有两种情况 :该消息的头域中不包括客户端的认证交换信息 ;该消息的头域中包括客户端的认证交换信息。

[0057] 当该消息的头域中不包括客户端的认证交换信息时,服务器端使用用户名和初始密码对收到的请求消息进行认证并根据所述用户名和初始密码生成所述服务器端 DH 认证响应信息。

[0058] 当该消息的头域中不包括客户端的认证交换信息时,服务器端根据用户的认证交换信息以及本端的认证交换信息获取共享密钥,根据所述共享密钥对收到的请求消息进行认证,并根据所述共享密钥生成服务器端 DH 认证响应信息。

[0059] 步骤 208 :用户根据收到的包含服务器端认证信息的响应消息验证所述服务器端的合法性。

[0060] 上述过程结束后,也就是说在客户端和服务器端都获取共享密钥后,进行消息交互时,只使用所述共享密钥对所述消息进行加密。

[0061] 参照图 3,图 3 示出了本发明方法中用户接入时的消息流程 :

[0062] 对于 UAS(服务器端),如果需要认证 UAC(客户端),则必须发送 401 响应,并必须在其中携带 WWW-Authenticate 头域,在该头域中包含 UAS 的认证,以防止 Middle-In-Man 攻击。UAC 可以再次发起请求,在 Authorization 头域中携带认证信息。Registrars(注册服务器) 和 Redirectserver(重定向服务器) 也可以使用 401 响应来进行认证 UAC。

[0063] 对于 Proxy(代理服务器),如果需要认证 UAC,则必须发送 407 响应,并必须在其中携带 Proxy-Authenticate 头域,在该头域中包含 Proxy 的认证,以防止 Middle-In-Man 攻击。UAC 可以再次发起请求,在 Proxy-Authorization 头域中携带认证信息。

[0064] 当 UAC 因为收到 401 或者 407 响应而重新发起请求时,一般应该使用相同的 Call-ID,From 头域和 To 头域,但是 CSeq 头域中的序数必须加一。

[0065] 但是一个 Server(UAS 或者 Proxy) 不能向 ACK(确认客户端已经接收到对 INVITE 的最终响应) 请求和 CANCEL 请求发起认证。对于 UAC 而言,比较好的方式是在 ACK 消息中包含一个通过认证的认证信息,该认证信息包括 Authorization 和 Proxy-Authorization 头域,它是在和 ACK 对应的 INVITE 消息中携带的并已经通过 UAS 或 Proxy 认证。

[0066] 为了防止 Plain Text 攻击,UAS 或者 Proxy 应在认证的成功响应 (200) 中包含新增头域 DH-Authentication-Info,在该新增头域中包含 UAS 或者 Proxy 的验证信息,UAC 通过该信息验证 UAS 或者 Proxy 的合法性。

[0067] 这可以是一个可选的特性,如果 UAC 和 UAS 或者 Proxy 间配置了必须包含

DH-Authentication-Info,则可以实现 UAC 验证其接入的服务器的合法性。如果在 200 响应中没有包含 DH-Authentication-Info 或者验证失败,且在 UAC 和 UAS 或者 Proxy 间配置了必须要包含 DH-Authentication-Info,则 UAC 可以认为这是某个恶意的服务器接收了消息,可以选择拒绝服务器。

[0068] 在上述认证过程中,只在初始的认证过程中使用初始密钥 Ki,而在以后的认证过程中都使用共享密钥 Ks,对于图 2 中的消息而言,F2-F4 所使用的密钥是 Ki,而 F6-F8 使用的密钥是 Ks。由于 Ki 只出现一次,所以可以有效地防止 Plain Text 和字典攻击。

[0069] 下面详细说明本发明中的 SIP 头域中的扩展参数及新增的 SIP 头域:

[0070] 本技术领域人员知道,在 RFC3261 中定义了四个头域,分别为:WWW-Authenticate,Proxy-Authenticate,Authorization,Proxy-Authorization,本发明即是在此基础上,通过对这四个头域中参数的扩展实现 DH-Digest 认证。

[0071] 本发明中定义的头域如下:

[0072] 1.WWW-Authenticate =" WWW-Authenticate" HCOLON challenge

[0073] 2.Proxy-Authenticate =" Proxy-Authenticate" HCOLON challenge

[0074] 其中,challenge = (" Digest" |LWS digest-cln*(COMMA digest-cln))

[0075] /dh-challenge/other-challenge

[0076] dh-challenge = (" DH-Digest" | LWS digest-cln*(COMMA digest-cln))

[0077] other-challenge = auth-scheme LWS auth-param*(COMMA auth-param)

[0078] digest-cln = realm/domain/nonce/opaque/stale/algorithm

[0079] /qop-options/dh-b/dh-response-auth/auth-param

[0080] realm =" realm" EQUAL realm-value

[0081] realm-value = quoted-string

[0082] domain =" domain" EQUAL LDQUOT URI*(1*SP URI)RDQUOT

[0083] URI = absoluteURI/abs-path

[0084] nonce =" nonce" EQUAL nonce-value

[0085] nonce-value = quoted-string

[0086] opaque =" opaque" EQUAL quoted-string

[0087] stale =" stale" EQUAL(" true" /" false")

[0088] algorithm =" algorithm" EQUAL(" MD5" /" MD5-sess" /token)

[0089] qop-options =" qop" EQUAL LDQUOT qop-value

[0090] *(" , " qop-value)RDQUOT

[0091] qop-value =" auth" /" auth-int" /token

[0092] dh-b =" DH-B" EQUAL dh-b-value

[0093] dh-b-value = quoted-string

[0094] dh-response-auth =" DH-Rspauth" EQUAL dh-response-digest

[0095] dh-response-digest = LDQUOT 32LHEX RDQUOT

[0096] 其中,带下划线部分为头域中新增的参数。

[0097] 对上述 WWW-Authenticate 和 Proxy-Authenticate 头域中的参数说明如下:

[0098] realm:realm-value 必须是一个全局唯一的字符串,并且全部由可显示的字符组

成,用来呈现给用户,指示用户输入用户名及密码。

[0099] Domain :由双引号包含的一个或多个URI 列表,表明在这些domain 域中,可以使用同样的认证信息。该参数对 Proxy-Authenticate 头域无意义。

[0100] Nonce :nonce 是由 server 提供的一串以 16 进制或 base64 表示的随机字符串。

[0101] Opaque :opaque 是由 server 提供的一串以 16 进制或 base64 表示的随机字符串,客户端应不作任何改变,返回给 server。

[0102] Stale :该参数是一个标志,有 TRUE 和 FALSE 两个值,用来指示前一个请求,由于 nonce 过期而导致的认证失败。当客户端收到的 401/407 中,该参数值为 TRUE 时,只需使用新的 nonce,重新计算一次摘要即可,不需再要求用户输入用户名及密码。只有当 server 收到的 request 中 nonce 是过期的,但是该过期的 nonce 对应的摘要正确时(也就是说用户名和密码是正确的),才可将该参数设置为 TRUE。

[0103] Algorithm :用于指示两边计算摘要的算法,当没有该参数时,缺省为 MD5 算法。

[0104] qop-options :为了兼容 RFC2069 而引入的任选参数,用于指示 server 所能支持的" quality of protection ",可以带多个值,目前有两个取值:" auth"、" auth-int"(取值不同在加密算法上稍有不同)。具体使用方法参见后面的摘要计算方法。

[0105] dh-b :为了实现 DH Digest 而特别引入的一个参数,代表了 UAS 或者 Proxy 的 DH 交换数 ;通过此参数,UAC 可以用来计算出共享密钥。

[0106] 当 scheme 为 DH-Digest,而且 WWW-Authentication 和 Proxy-Authentication 中不包含此 dh-b 时,则意味着 UAS 或者 Proxy 已经将 db-b 在以前的消息中发送给 UAC 了,当前的这次认证可以直接采用共享密钥,而不需要采用初始密钥(如果 UAC 不记得共享密钥,如重启后,也可以使用初始密钥进行认证)。当其中包含 dh-b 时,则表示 UAS 或者 Proxy 希望重新发起一次共享密钥的协商。

[0107] dh-response-auth :为了防止恶意服务器发起的 401 或者 407 响应,UAC 需要对 UAS 或者 Proxy 发起的 401 或者 407 响应进行认证。UAS 或者 Proxy 在发送 401 或者 407 响应时,必须采用初始密钥(当 dh-b 参数存在时)或者采用共享密钥(当不存在 dh-b 参数时)进行认证。

[0108] auth-param :该参数是为了将来扩展引入的。

[0109] 3.Proxy-Authorization =" Proxy-Authorization" HCOLON credentials

[0110] 4.Authorization =" Authorization" HCOLON credentials

[0111] 其中,credentials = (" Digest" LWS digest-response)

[0112] /dh-digest-response/other-response

[0113] digest-response = dig-resp*(COMMA dig-resp)

[0114] dh-digest-response = dig-resp*(COMMA dig-resp)

[0115] dig-resp = username/realm/nonce/digest-uri

[0116] /dresponse/algorithm/cnonce

[0117] /opaque/message-qop

[0118] /nonce-count/dh-a/auth-param

[0119] username =" username" EQUAL username-value

[0120] username-value = quoted-string

[0121] digest-uri =" uri" EQUAL LDQUOT digest-uri-value RDQUOT

[0122] digest-uri-value = rquest-uri ;Equal to request-uri as specified

[0123] by HTTP/1.1

[0124] message-qop =" qop" EQUAL qop-value

[0125] cnonce =" cnonce" EQUAL cnonce-value

[0126] cnonce-value = nonce-value

[0127] nonce-count =" nc" EQUAL nc-value

[0128] nc-value = 8LHEX

[0129] dresponse =" response" EQUAL request-digest

[0130] request-digest = LDQUOT 32LHEX RDQUOT

[0131] dh-a =" DH-A" EQUAL dh-a-value

[0132] dh-a-value = quoted-string

[0133] auth-param = auth-param-name EQUAL(token/quoted-string)

[0134] auth-param-name = token

[0135] other-response = auth-scheme LWS auth-param*(COMMA auth-param)

[0136] auth-scheme = token

[0137] 其中,带下划线部分为头域中新增的参数。

[0138] 对上述 Authorization 和 Proxy-Authorization 头域中的参数说明如下 :

[0139] response :一个 128 比特的,由 32 个 16 进制数表示的字符串,它是由后面的公式计算而来的。

[0140] username :在指定的 realm 范围内的用户名。

[0141] digest-uri :与最初的 Request-Line 的 Request-URI 相同,之所以不直接使用请求消息中的 Request-URI,是因为中间的 proxy 可能会修改 Request-URI。

[0142] message-qop :为了兼容 RFC2069 而引入的任选参数,用于指示客户端所能支持的“quality of protection(保护质量)”,只能带一个值,且只能是 server 过来的 qop 中的一个值。它将影响对摘要的计算。WWW-Authenticate 或 Proxy-Authenticate 头域中如果包含了 qop 参数,那么在 Authorization 或 Proxy-Authorization 头域中必须带此参数。

[0143] cnonce :如果 WWW-Authenticate 或 Proxy-Authenticate 头域中带了 qop 参数,那么在 Authorization 或 Proxy-Authorization 头域中必须带此参数,否则不需要带此参数。该参数由客户端给出,用于避免 plain text 攻击、提供 message integrity protection、以及提供相互的认证。

[0144] cnonce-count :如果 WWW-Authenticate 或 Proxy-Authenticate 头域中带了 qop 参数,那么在 Authorization 或 Proxy-Authorization 头域中必须带此参数,否则不需要带此参数。该参数是一个 16 进制的计算器,用于计数客户端发出的包含 nonce 的请求消息个数。例如,对于 server 给定的一个 nonce,客户端发出的第一个请求消息,该参数为“nc = 00000001”,server 会保存自己的一份 nc 拷贝,这样 server 能对客户端发过来的该参数的值与自己保存的值进行比较,这样就可以判断是否受到了 replay 攻击。

[0145] dh-a :为了实现 DH Digest 而特别引入的一个参数,代表了 UAC 的 DH 交换数 ;通过

此参数，UAS 或者 Proxy 可以用来计算出共享密钥。

[0146] 当 Authorization 和 Proxy-Authorization 头域中的 scheme 为“DH-Digest”，此时如果头域中包含了 dh-a 参数，则当前的这次认证是使用初始密钥进行加密的，此时不管 Challenge(WWW-Authentication 和 Proxy-Authentication) 中是否包含了 dh-b 参数，都应该使用初始密钥和 dh-a 的算法进行验证（这种情况可能发生在 UAC 重启后，发送新的请求，但是 UAS 或者 Proxy 并不知道 UAC 重启了，而仍然希望 UAC 使用共享密钥来进行验证，此时 UAC 可以忽略 UAS 或者 Proxy 希望通过共享密钥来加密的请求，而通过初始密钥来实现验证）；如果不包含 dh-a 参数，则意味着在以前的消息中 UAC 已经将 dh-a 发送给 UAS 或者 Proxy 了，当前的这次认证是使用共享密钥加密的。

[0147] 根据 Authentication-Info 的 ABNF 定义不能扩展参数，所以在本发明中还新增加了一个头域 DH-Authentication-Info。

[0148] 5.DH-Authentication-Info = " DH-Authentication-Info " HCOLON dh-ainfo*(COMMA dh-ainfo)

[0149] 其中，

[0150] dh-ainfo = nextnonce/message-qop/response-auth

[0151] /cnonce/nonce-count/dh-a

[0152] nextnonce =" nextnonce" EQUAL nonce-value

[0153] response-auth =" rspauth" EQUAL response-digest

[0154] response-digest = LDQUOT*LHEX RDQUOT

[0155] 对上述新增头域 DH-Authentication-Info 中的参数说明如下：

[0156] UAS 或者 Proxy 可以利用该头域：

[0157] A、改变 nonce，客户端收到带 nextnonce 参数的该头域后，如果要发送下一个请求，则应使用新的 nonce 进行计算，否则如果客户端仍使用老的 nonce 计算摘要，server 会要求重新认证，并且带指示 TRUE 的 stale 参数。此时不需要包含 dh-a 参数。

[0158] B、UAS 或 Proxy 向 UAC 认证自己，需要携带 response-digest 参数；此时如果包含了 dh-a 参数，则此 response-digest 是采用初始密钥进行加密的；如果没有携带 dh-a 参数，则此 response-digest 是采用的共享密钥进行加密的。

[0159] Nextnonce：该参数是 server 给出的客户端下一次发送请求消息时用的新 nonce。

[0160] message-qop：该参数应与客户端发来的 qop 参数取相同的值，指示 server 计算 response 摘要时的" quality of protection"。

[0161] 为了防止 replay 攻击，本发明方法规定在上述头域 WWW-Authenticate，Proxy-Authenticate，Authorization，Proxy-Authorization 使用中必须携带 qop 参数。

[0162] 上述参数中涉及的加密算法如下：

[0163] 1、dh-response-digest(DH 响应摘要）的算法如下：

[0164] 因为在 WWW-Authenticate 和 Proxy-Authenticate 中的 qop 是一个选项，可以包含 auth 或者 auth-int 等多种选择。所以在此强制规定，qop-value 为只使用 auth。

[0165] dh-response-digest = <" >MD5(MD5(A1)，unq(nonce-value)

[0166] " :" unq(qop-value)

[0167] " :" MD5(A2)

[0168])<" >

[0169] 其中,A1 的算法如下:

[0170] 如果" algorithm" 指示" MD5" 或没有带该参数,并且没有 dh-b 参数,则 A 1 = unq(username-value)" :" unq(realm-value)" :" shared-key

[0171] 其中,shared-key =<shared key calculated by dh-a and dh-b>

[0172] 如果" algorithm" 指示" MD5" 或没有带该参数,并且有 dh-b 参数,则 A1 = unq(username-value)" :" unq(realm-value)" :" passwd" :" unq(dh-b)

[0173] 其中,passwd =<user' s password>

[0174] dh-b =<dh-b-value>

[0175] 如果" algorithm" 指示" MD5-sess",并且没有 dh-b 参数,则

[0176] A 1 = MD5(unq(username-value)" :" unq(realm-value)" :" shared-key)" :" unq(nonce-value))

[0177] 如果" algorithm" 指示" MD5" 或没有带该参数,并且有 dh-b 参数,则 A1 = MD5(unq(username-value)" :" unq(realm-value)" :" passwd" :" unq(dh-b))

[0178] A2 的算法如下:

[0179] A2 = Method " :" digest-uri-value

[0180] 2、request-digest(请求摘要)的算法如下:

[0181] 如果" qop" 值为" auth" 或" auth-int",则

[0182] request-digest =<" ><MD5(MD5(A1),unq(nonce-value)

[0183] " :" nc-value

[0184] " :" unq(cnonce-value)

[0185] " :" unq(qop-value)

[0186] " :" MD5(A2)

[0187])<" >

[0188] 其中,A1 的算法如下:

[0189] 如果" algorithm" 指示" MD5" 或没有带该参数,并且没有 dh-a 参数,则 A1 = unq(username-value)" :" unq(realm-value)" :" shared-key

[0190] 其中,shared-key =<shared key calculated by dh-a and dh-b>

[0191] 如果" algorithm" 指示" MD5" 或没有带该参数,并且有 dh-a 参数,则 A1 = unq(username-value)" :" unq(realm-value)" :" passwd" :" dh-a

[0192] 其中,passwd =<user' s password>

[0193] dh-a =<dh-a-value>

[0194] 如果" algorithm" 指示" MD5-sess",并且没有 dh-a 参数,则

[0195] A1 = MD5(unq(username-value)" :" unq(realm-value)

[0196] " :" shared-key)" :" unq(nonce-value)" :" unq(cnonce-value)

[0197] 如果" algorithm" 指示" MD5" 或没有带该参数,并且有 dh-a 参数,则

[0198] A1 = MD5(unq(username-value)" :" unq(realm-value)

[0199] " :" passwd" :" dh-a)" :" unq(nonce-value)" :" unq(cnonce-value)

[0200] A2 的算法如下：

[0201] 如果" qop" 指示" auth" ,则 A2 = Method" :" digest-uri-value

[0202] 如果" qop" 指示" auth-int" ,则 A2 = Method " :" digest-uri-value " :" MD5(entity-body)

[0203] 如果 SIP 的消息体为空,则在 RFC2617 中定义的 A2 中使用的 H(entity-body) 采用如下的定义：

[0204] H(entity-body) = MD5(" ") =" d41d8cd98f00b204e9800998ecf8427e"

[0205] 3、response-digest(响应摘要) 的算法：

[0206] response-digest 的算法与前面的 request-digest 算法相似,区别在于 A2 的计算：

[0207] 如果 Authorization 和 Proxy-Authorization 头域中的" qop" 指示" auth" ,则 A2 =" :" digest-uri-value

[0208] 如果 Authorization 和 Proxy-Authorization 头域中的" qop " 指示" auth-int" ,则 A2 =" :" digest-uri-value" :" MD5(entity-body)

[0209] 如果 SIP 的消息体为空,则在 RFC2617 中定义的 A2 中使用的 H(entity-body) 采用如下的定义：

[0210] H(entity-body) = MD5(" ") =" d41d8cd98f00b204e9800998ecf8427e"

[0211] 虽然通过实施例描绘了本发明,本领域普通技术人员知道,本发明有许多变形和变化而不脱离本发明的精神,希望所附的权利要求包括这些变形和变化而不脱离本发明的精神。

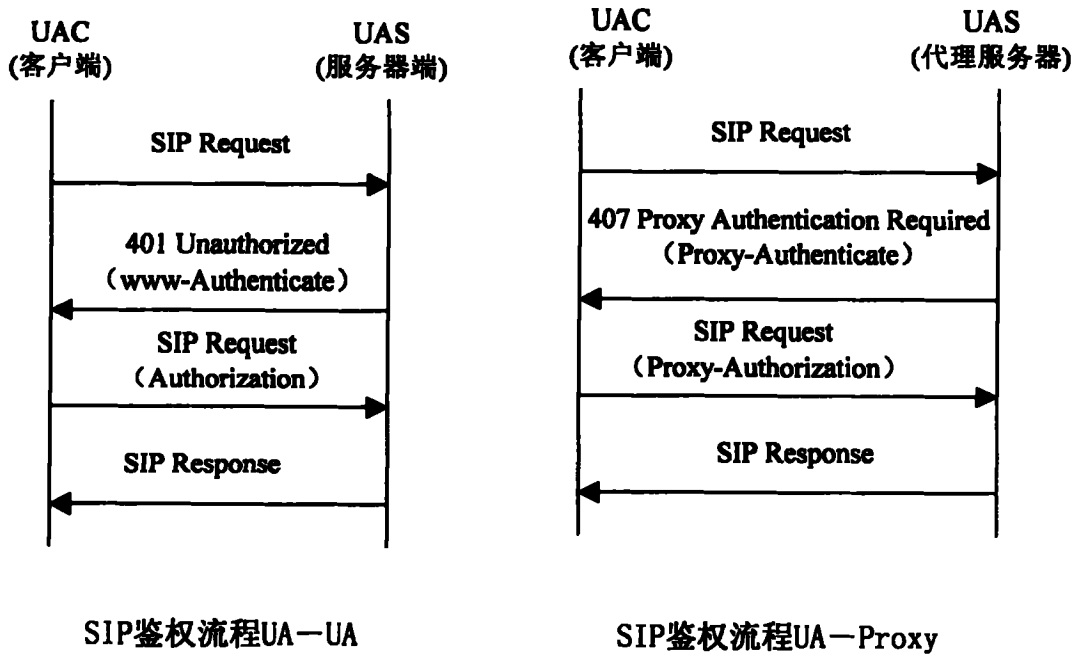

UAC
(客户端)
UAS
(服务器端)
SIP Request
401 Unauthorized
(www-Authenticate)
SIP Request
(Authorization)
SIP Response
SIP鉴权流程UA－UA
UAC
(客户端)
UAS
(代理服务器)
SIP Request
407 Proxy Authentication Required
(Proxy-Authenticate)
SIP Request
(Proxy-Authorization)
SIP Response
SIP鉴权流程UA－Proxy

图 1

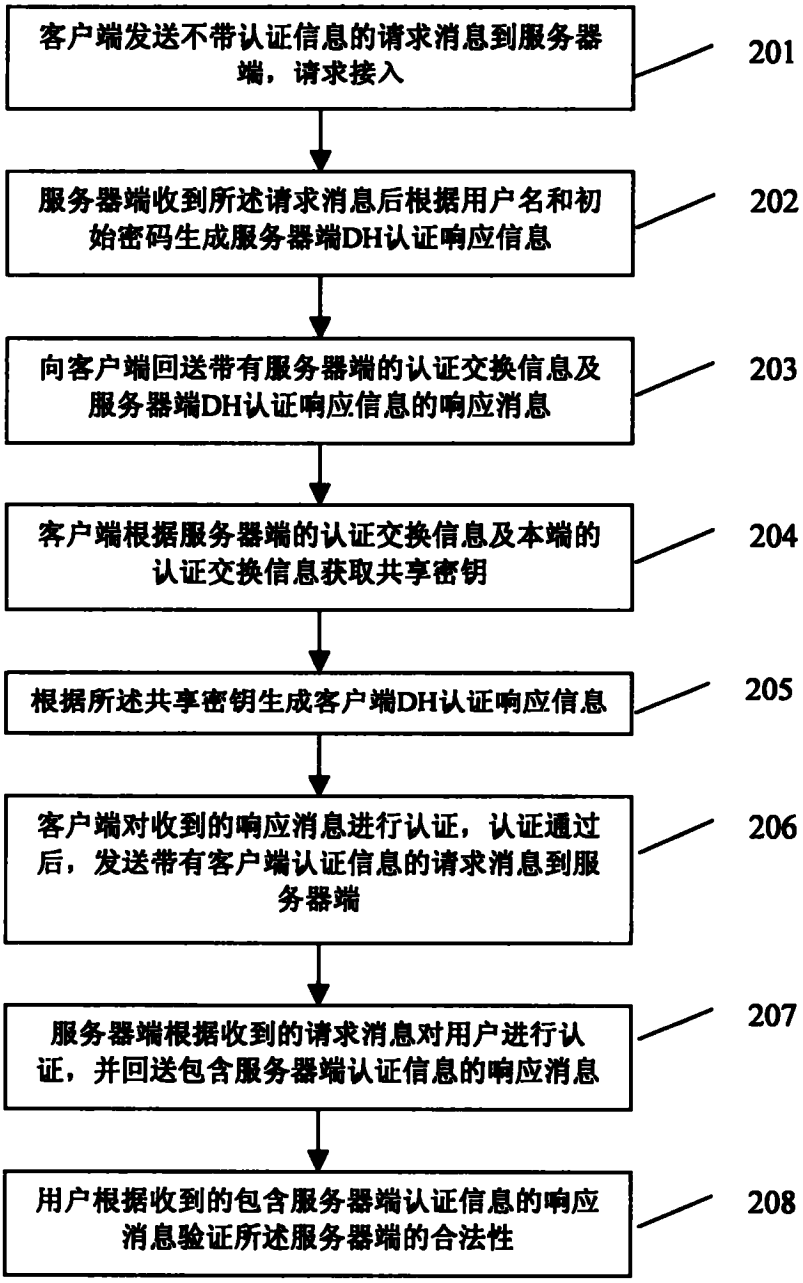

客户端发送不带认证信息的请求消息到服务器端，请求接入
201
服务器端收到所述请求消息后根据用户名和初始密码生成服务器端DH认证响应信息
202
向客户端回送带有服务器端的认证交换信息及服务器端DH认证响应信息的响应消息
203
客户端根据服务器端的认证交换信息及本端的认证交换信息获取共享密钥
204
根据所述共享密钥生成客户端DH认证响应信息
205
客户端对收到的响应消息进行认证，认证通过后，发送带有客户端认证信息的请求消息到服务器端
206
服务器端根据收到的请求消息对用户进行认证，并回送包含服务器端认证信息的响应消息
207
用户根据收到的包含服务器端认证信息的响应消息验证所述服务器端的合法性
208

图 2

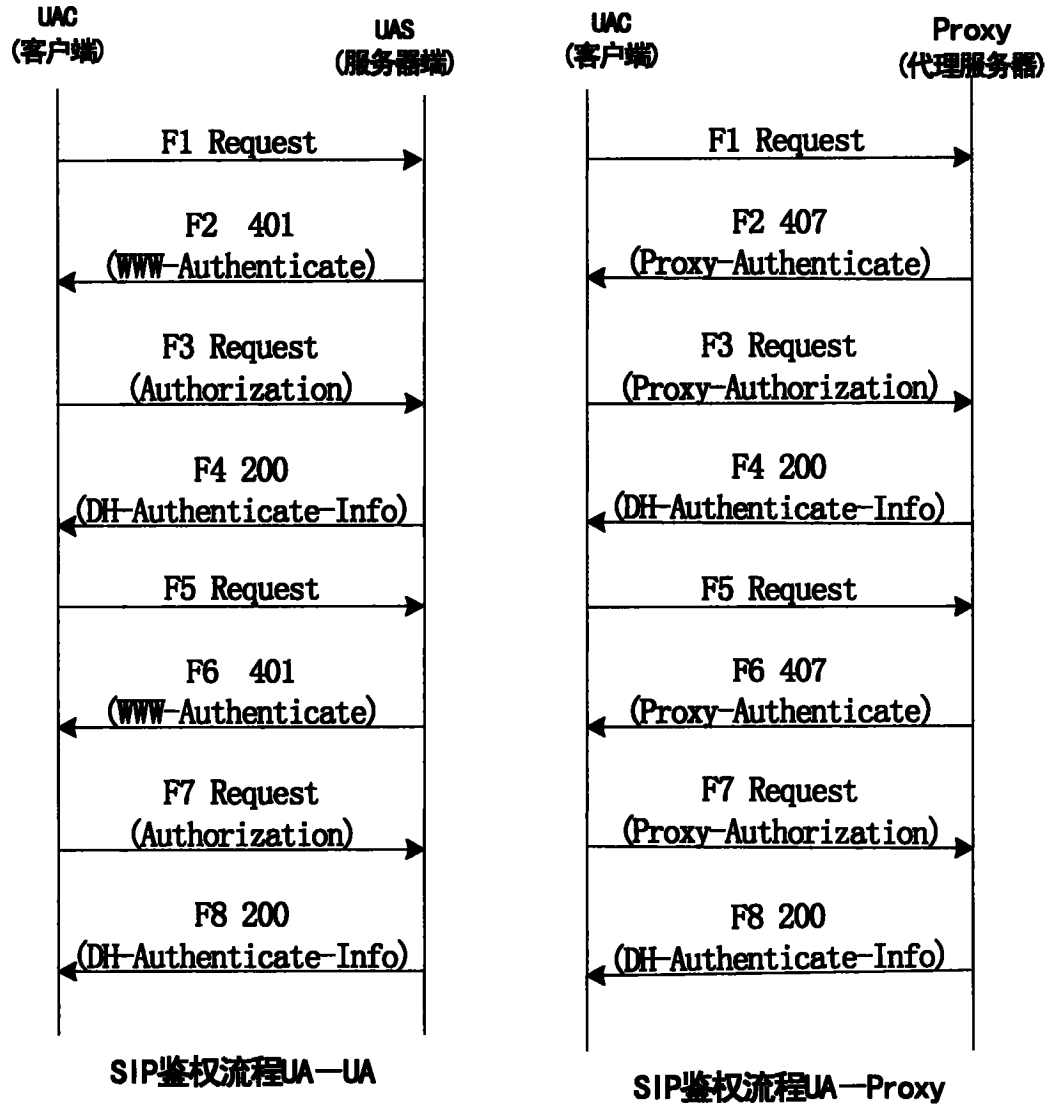

UAC
(客户端)
UAS
(服务器端)
F1 Request
F2 401
(WWW-Authenticate)
F3 Request
(Authorization)
F4 200
(DH-Authenticate-Info)
F5 Request
F6 401
(WWW-Authenticate)
F7 Request
(Authorization)
F8 200
(DH-Authenticate-Info)
SIP鉴权流程UA—UA
UAC
(客户端)
Proxy
(代理服务器)
F1 Request
F2 407
(Proxy-Authenticate)
F3 Request
(Proxy-Authorization)
F4 200
(DH-Authenticate-Info)
F5 Request
F6 407
(Proxy-Authenticate)
F7 Request
(Proxy-Authorization)
F8 200
(DH-Authenticate-Info)
SIP鉴权流程UA—Proxy

图 3