(19)中华人民共和国国家知识产权局

(12)发明专利申请

(10)申请公布号 CN 109391474 A
(43)申请公布日 2019.02.26

(21)申请号 201811587525.4

(22)申请日 2018.12.25

(71)申请人 武汉思普崚技术有限公司
地址 430070 湖北省武汉市东湖新技术开发区光谷大道77号金融港后台服务中心一期A4栋2层01号

(72)发明人 李洪宇

(74)专利代理机构 北京弘权知识产权代理事务所(普通合伙) 11363
代理人 逯长明 许伟群

(51)Int.Cl.
H04L 9/32(2006.01)
H04L 29/06(2006.01)
H04L 29/08(2006.01)

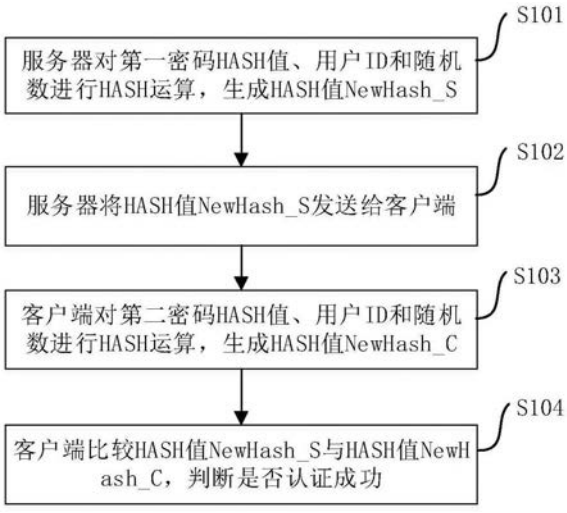
权利要求书1页 说明书5页 附图1页

(54)发明名称

一种非加密链路的安全认证方法及系统

(57)摘要

本申请公开了一种非加密链路的安全认证方法及系统，首先，服务器对第一密码HASH值、用户ID和随机数进行HASH运算，生成HASH值NewHash_S，将HASH值NewHash_S发送给客户端；然后，客户端对第二密码HASH值、用户ID和随机数进行HASH运算，生成HASH值NewHash_C，再比较HASH值NewHash_S与HASH值NewHash_C，判断是否认证成功。本申请的技术方案避免采用HTTPS建立加密链路的方式，在以HTTP方式认证的过程中，不传输密码，只在客户端对密码进行认证，使得客户的密码在认证过程中不存在被监听和泄漏的风险。



CN 109391474 A

1.一种非加密链路的安全认证方法,其特征在于,包括:

服务器对第一密码HASH值、用户ID和随机数进行HASH运算,生成HASH值NewHash_S;

服务器将所述HASH值NewHash_S发送给客户端;

客户端对第二密码HASH值、用户ID和随机数进行HASH运算,生成HASH值NewHash_C;

客户端比较所述HASH值NewHash_S与所述HASH值NewHash_C,判断是否认证成功。

2.根据权利要求1所述的认证方法,其特征在于,在所述服务器利用第一密码HASH值、用户ID和随机数,生成HASH值NewHash_S之前还包括:

服务器从客户端获取用户名、用户ID和随机数;

服务器根据所述用户名查找第一密码HASH值,所述第一密码HASH值为服务器对密码进行HASH运算后的值。

3.根据权利要求1所述的认证方法,其特征在于,还包括:

客户端通过浏览器打开登录页面;

客户端从所述登录页面上获取用户名和密码;

客户端对所述密码进行HASH运算,生成第二密码HASH值。

4.根据权利要求1所述的认证方法,其特征在于,客户端比较所述HASH值NewHash_S与所述HASH值NewHash_C,判断是否认证成功的步骤包括:

判断所述HASH值NewHash_S与所述HASH值NewHash_C是否相同;

如果所述HASH值NewHash_S与所述HASH值NewHash_C相同,则客户端认证成功;

如果所述HASH值NewHash_S与所述HASH值NewHash_C不相同,则客户端认证失败。

5.根据权利要求4所述的认证方法,其特征在于,还包括:如果所述客户端认证失败,则提示用户名或者密码错误。

6.根据权利要求3所述的认证方法,其特征在于,还包括:

服务器配置所述客户端登录页面的访问方式为HTTP方式;

登录页面通过所述HTTP方式将所述用户名、所述用户ID和所述随机数发送给服务器。

7.根据权利要求1-6任一项所述的认证方法,其特征在于,所述用户ID由服务器产生,并发送给客户端;所述随机数由客户端产生,并发送给服务器。

8.根据权利要求7所述的认证方法,其特征在于,所述用户ID与所述随机数都保存在客户端的浏览器缓存中。

9.根据权利要求1所述的认证方法,其特征在于,所述HASH运算采用SHA256算法或者SHA512算法。

10.一种非加密链路的安全认证系统,其特征在于,包括:客户端和与客户端建立通信连接的服务器;

所述服务器,用于对第一密码HASH值、用户ID和随机数进行HASH运算,生成HASH值NewHash_S;将所述HASH值NewHash_S发送给客户端;

所述客户端,用于对第二密码HASH值、用户ID和随机数进行HASH运算,生成HASH值NewHash_C;比较所述HASH值NewHash_S与所述HASH值NewHash_C,判断是否认证成功。

一种非加密链路的安全认证方法及系统

技术领域

[0001] 本申请涉及计算机技术领域，尤其涉及一种非加密链路的安全认证方法及系统。

背景技术

[0002] 现今在WEB认证交互过程中，主要使用HTTP协议进行交互认证，通常采用用户名、密码方式，再加以校验码、短信码等认证方式对认证交互提供安全保证。由于HTTP在传输过程中传输的都是明文信息，存在被监听和泄漏的风险，所以在现今绝大多数认证过程中，都采用了HTTPS方式，对链路进行加密认证，HTTPS是超文本传输安全协议，是以安全为目标的HTTP通道。

[0003] 然而，采用HTTPS的方式又使得处理效率降低，因为用户在登录一个系统时，采用HTTPS方式建立加密隧道，在加密隧道中实现WEB认证交互过程。使用HTTPS方式，对于每个连接都需要建立加密隧道，每个加密隧道建立需要至少4个报文，后续的数据也需要进行加解密，在此过程中CPU需要处理大量加解密数据，所以服务器并发认证的处理效率较低，一般情况会低于HTTP并发的10倍以上。而采用HTTP方式，则存在被监听和泄漏的风险。

[0004] 因此，在HTTP方式的交互认证过程中，如何提供安全保护，使得客户的密码在认证过程中不存在被监听和泄漏的风险，成为本领域技术人员亟待解决的问题。

发明内容

[0005] 本申请提供了一种非加密链路的安全认证方法及系统，在HTTP方式的交互认证过程中，提供安全保护，使得客户用户名、密码在认证过程中不存在被监听和泄漏的风险。

[0006] 一方面，本申请提供了一种非加密链路的安全认证方法，包括：

[0007] 服务器对第一密码HASH值、用户ID和随机数进行HASH运算，生成HASH值NewHash_S；

[0008] 服务器将所述HASH值NewHash_S发送给客户端；

[0009] 客户端对第二密码HASH值、用户ID和随机数进行HASH运算，生成HASH值NewHash_C；

[0010] 客户端比较所述HASH值NewHash_S与所述HASH值NewHash_C，判断是否认证成功。

[0011] 结合第一方面，在所述服务器利用第一密码HASH值、用户ID和随机数，生成HASH值NewHash_S之前还包括：

[0012] 服务器从客户端获取用户名、用户ID和随机数；

[0013] 服务器根据所述用户名查找第一密码HASH值，所述第一密码HASH值为服务器对密码进行HASH运算后的值。

[0014] 结合第一方面，还包括：

[0015] 客户端通过浏览器打开登录页面；

[0016] 客户端从所述登录页面上获取用户名和密码；

[0017] 客户端对所述密码进行HASH运算，生成第二密码HASH值。

[0018] 结合第一方面，客户端比较所述HASH值NewHash_S与所述HASH值NewHash_C，判断是否认证成功的步骤包括：

[0019] 判断所述HASH值NewHash_S与所述HASH值NewHash_C是否相同；

[0020] 如果所述HASH值NewHash_S与所述HASH值NewHash_C相同，则客户端认证成功；

[0021] 如果所述HASH值NewHash_S与所述HASH值NewHash_C不相同，则客户端认证失败。

[0022] 结合第一方面，还包括：如果所述客户端认证失败，则提示用户名或者密码错误。

[0023] 结合第一方面，还包括：

[0024] 服务器配置所述客户端登录页面的访问方式为HTTP方式；

[0025] 登录页面通过所述HTTP方式将所述用户名、所述用户ID和所述随机数发送给服务器。

[0026] 结合第一方面，所述用户ID由服务器产生，并发送给客户端；所述随机数由客户端产生，并发送给服务器。

[0027] 结合第一方面，所述用户ID与所述随机数都保存在客户端的浏览器缓存中。

[0028] 结合第一方面，所述HASH运算采用SHA256算法或者SHA512算法。

[0029] 第二方面，本申请实施例还提供了一种非加密链路的安全认证系统，包括：客户端和与客户端建立通信连接的服务器；

[0030] 所述服务器，用于对第一密码HASH值、用户ID和随机数进行HASH运算，生成HASH值NewHash_S；将所述HASH值NewHash_S发送给客户端；

[0031] 所述客户端，用于对第二密码HASH值、用户ID和随机数进行HASH运算，生成HASH值NewHash_C；比较所述HASH值NewHash_S与所述HASH值NewHash_C，判断是否认证成功。

[0032] 由以上技术方案可知，本申请提供了一种非加密链路的安全认证方法及系统，首先，服务器对第一密码HASH值、用户ID和随机数进行HASH运算，生成HASH值NewHash_S，将HASH值NewHash_S发送给客户端；然后，客户端对第二密码HASH值、用户ID和随机数进行HASH运算，生成HASH值NewHash_C，再比较HASH值NewHash_S与HASH值NewHash_C，判断是否认证成功。本申请的技术方案避免采用HTTPS建立加密链路的方式，在以HTTP方式认证的过程中，不传输密码，只在客户端对密码进行认证，使得客户的密码在认证过程中不存在被监听和泄漏的风险。

附图说明

[0033] 为了更清楚地说明本申请的技术方案，下面将对实施案例中所需要使用的附图作简单地介绍，显而易见地，对于本领域普通技术人员而言，在不付出创造性劳动性的前提下，还可以根据这些附图获得其他的附图。

[0034] 图1为本申请提供的一种非加密链路的安全认证方法的流程图；

[0035] 图2为本申请提供的一种非加密链路的安全认证系统的示意图。

具体实施方式

[0036] 为了使本技术领域的人员更好地理解本申请中的技术方案，下面将结合附图，对本申请实施例中的技术方案进行清楚、完整地描述。

[0037] 参见图1，本申请提供了一种非加密链路的安全认证方法，包括：

[0038] 步骤101，服务器对第一密码HASH值、用户ID和随机数进行HASH运算，生成HASH值NewHash_S。目前为了提高用户登录信息的安全性，多数服务器都不会直接保存用户名对应的密码，而是对密码进行一个HASH运算，得到一个密码HASH值保存在服务器中。HASH算法是一种不可逆的加密算法，因为对密码进行运算的过程中不需要使用密钥，明文密码直接加密处理成密文，加密后的密码HASH值无法被破解，只有重新输入密码并再次经过同样的HASH运算得到相同的密码HASH值才能算是解密成功。HASH算法是把任意长度的输入(又叫做预映射pre-image)通过散列算法变换成固定长度的输出，该输出就是散列值。这种转换是一种压缩映射，也就是，散列值的空间通常远小于输入的空间，不同的输入可能会散列成相同的输出，所以不可能从散列值来确定唯一的输入值。因此，服务器只保存密码HASH值，能够保证密码不会被反向破解，保证密码的安全。

[0039] 常见的HASH算法有很多，例如MD4、MD5、SHA1等等，本申请实施例中优选的采用SHA256算法或者SHA512算法，相比于其他算法来说，SHA256算法和SHA512算法安全强度比较高。SHA256算法的哈希值大小为256位，SHA512算法的哈希值大小为512位。

[0040] 值得说明的是，本申请中对密码进行HASH运算和对第一密码HASH值、用户ID以及随机数进行HASH运算时的两个HASH算法并不相同。对第一密码HASH值、用户ID和随机数进行HASH运算，利用随机数的随机性，保证本次HASH加密得到的NewHash_S是随机的，表示并不是每一次的得到NewHash_S都是相同的，也进一步保证了用户ID的安全。

[0041] 本申请中用户的ID是由服务器根据用户客户端的使用情况产生的，并且通过服务器还可以对客户端的登录页面进行配置，配置好以后，服务器再将用户ID与登录页面一同发送给客户端，由客户端进行保存。上述随机数由客户端产生，具体的可以设置随机器生成器产生，随机数顾名思义就是随机产生的一些数字，这些数字之间不具有任何关联性，保证数字之间的随机性，然后每一次的认证过程都会产生一个新的随机数进行HASH运算，保证每一次认证过程得到的NewHash_S都是不同的，减少使用相同的NewHash_S用户ID会被破解的风险。用户ID与客户端产生的随机数不仅保存在客户端内地的浏览器缓存中，还会传送给服务器进行处理。

[0042] 本申请中由服务器将登录页面配置成HTTP的访问方式，这种访问方式无需建立如HTTPS方式中的加密链路，提高服务器的处理效率。当用户在登录页面输入用户名和密码之后，密码会直接保存在客户端，而客户端的用户ID、用户名以及随机数会被登录页面一同发送给服务器。

[0043] 进一步的，在步骤101之前，还包括：服务器从客户端获取用户名、用户ID和随机数。服务器根据所述用户名查找第一密码HASH值，所述第一密码HASH值为服务器对密码进行HASH运算后的值。根据以上步骤可知，每一个密码都有与之对应的用户名，那么每一个与密码对应的密码HASH值也会有一个与之对应的用户名，所以当服务器从客户端获取用户名之后，就可以根据用户名直接找到与之对应的密码HASH值。服务器通常会预先保存与每一个对应的用户名和密码，但是在将密码进行了HASH运算后，会将密码删除或者进行别的保密处理，总之在服务器端不会直接保存密码。

[0044] 步骤102，服务器将所述HASH值NewHash_S发送给客户端。本申请中的安全认证方法通过服务器和客户端两部分共同实现，因为本申请的技术方案中不需要传输密码，只将密码保存在客户端，所以在服务器得到了NewHash_S之后，还需要将NewHash_S发送回客户

端进行处理，完成在客户端内对用户进行认证的操作。

[0045] 步骤103，客户端对第二密码HASH值、用户ID和随机数进行HASH运算，生成HASH值NewHash_C。第二密码HSAH值由客户端根据用户输入的密码进行计算生成，具体过程如下：客户端通过客户端内的浏览器打开登录页面；从所述登录页面上获取用户输入的用户名和密码；对所述密码进行HASH运算，生成第二密码HASH值。本步骤中生成密码HASH值的算法与上述内容中服务器生成密码HASH值的算法相同，保持双端运算的一致性，保证相同的密码进过相同的运算后，得到的结果是相同的。生成了第二密码HASH值以后，客户端还要利用与步骤101相同的HASH算法，如SHA256算法或者SHA512算法，对第二HASH值、用户ID和随机数进行HASH运算，生成客户端的HASH值NewHash_C，客户端产生NewHash_C的步骤与步骤101相同的，目的在于使服务器与客户端的全部运算过程相同，避免相同的密码进行不同的运算产生不同的结果，进而影响认证结果。

[0046] 步骤104，客户端比较所述HASH值NewHash_S与所述HASH值NewHash_C，判断是否认证成功。具体的认证过程如下：判断所述HASH值NewHash_S与所述HASH值NewHash_C是否相同；如果所述HASH值NewHash_S与所述HASH值NewHash_C相同，则客户端认证成功；如果所述HASH值NewHash_S与所述HASH值NewHash_C不相同，则客户端认证失败。如果所述客户端认证失败，则提示用户名或者密码错误。另外，在步骤101之前，如果服务器根据用户名没有找到对应的密码HASH值，也会反馈给客户端，提示用户名或者密码错误。本申请中由于不直接传输密码，所以在客户端和服务器中都不会对密码进行直接的验证，而是验证相同操作之后获得的HASH值，如果HASH值相同，那么密码必然也是相同的，因此可以认证成功。

[0047] 值得说明的是，本申请的技术方案中，如果客户端用户认证失败，那么提示给用户的信息是“用户名或者密码错误”，而不是单纯地提示“密码错误”，如果只提示密码错误，那么代表该用户就是存在的，可能会产生对该用户进行暴力破解的攻击行为。

[0048] 由以上技术方案可知，本申请提供了一种本申请提供了一种非加密链路的安全认证方法，首先，服务器对第一密码HASH值、用户ID和随机数进行HASH运算，生成HASH值NewHash_S，将HASH值NewHash_S发送给客户端；然后，客户端对第二密码HASH值、用户ID和随机数进行HASH运算，生成HASH值NewHash_C，再比较HASH值NewHash_S与HASH值NewHash_C，判断是否认证成功。本申请的技术方案避免采用HTTPS建立加密链路的方式，在以HTTP方式认证的过程中，不传输密码，只在客户端对密码进行认证，使得客户的密码在认证过程中不存在被监听和泄漏的风险。

[0049] 参见图2，本申请还提供了一种非加密链路的安全认证系统，包括：客户端21和与客户端21建立通信连接的服务器22；

[0050] 所述服务器22，用于对第一密码HASH值、用户ID和随机数进行HASH运算，生成HASH值NewHash_S；将所述HASH值NewHash_S发送给客户端21；

[0051] 所述客户端21，用于对第二密码HASH值、用户ID和随机数进行HASH运算，生成HASH值NewHash_C；比较所述HASH值NewHash_S与所述HASH值NewHash_C，判断是否认证成功。

[0052] 所述服务器22还用于，从客户端21获取用户名、用户ID和随机数；根据所述用户名查找第一密码HASH值，所述第一密码HASH值为服务器对密码进行HASH运算后的值。服务器22配置所述客户端21登录页面的访问方式为HTTP方式，产生用户ID，并发送给客户端21。

[0053] 所述客户端21还用于，通过浏览器打开登录页面；从所述登录页面上获取用户名

和密码;对所述密码进行HASH运算,生成第二密码HASH值。判断所述HASH值NewHash_S与所述HASH值NewHash_C是否相同;如果所述HASH值NewHash_S与所述HASH值NewHash_C相同,则客户端认证成功;如果所述HASH值NewHash_S与所述HASH值NewHash_C不相同,则客户端认证失败。如果认证失败,则提示用户名或者密码错误。产生随机数,并发送给服务器22。

[0054] 所述登录页面通过所述HTTP方式将所述用户名、所述用户ID和所述随机数发送给服务器22。

[0055] 由以上技术方案可知,本申请提供了一种非加密链路的安全认证系统,首先,服务器对第一密码HASH值、用户ID和随机数进行HASH运算,生成HASH值NewHash_S,将HASH值NewHash_S发送给客户端;然后,客户端对第二密码HASH值、用户ID和随机数进行HASH运算,生成HASH值NewHash_C,再比较HASH值NewHash_S与HASH值NewHash_C,判断是否认证成功。本申请的技术方案避免采用HTTPS建立加密链路的方式,在以HTTP方式认证的过程中,不传输密码,只在客户端对密码进行认证,使得客户的密码在认证过程中不存在被监听和泄漏的风险。

[0056] 本申请可用于众多通用或专用的计算系统环境或配置中。例如:个人计算机、服务器计算机、手持设备或便携式设备、平板型设备、多处理器系统、基于微处理器的系统、置顶盒、可编程的消费电子设备、网络PC、小型计算机、大型计算机、包括以上任何系统或设备的分布式计算环境等等。

[0057] 本申请可以在由计算机执行的计算机可执行指令的一般上下文中描述,例如程序模块。一般地,程序模块包括执行特定任务或实现特定抽象数据类型的例程、程序、对象、组件、数据结构等等。也可以在分布式计算环境中实践本申请,在这些分布式计算环境中,由通过通信网络而被连接的远程处理设备来执行任务。在分布式计算环境中,程序模块可以位于包括存储设备在内的本地和远程计算机存储介质中。

[0058] 本领域技术人员在考虑说明书及实践这里公开的申请后,将容易想到本申请的其它实施方案。本申请旨在涵盖本申请的任何变型、用途或者适应性变化,这些变型、用途或者适应性变化遵循本申请的一般性原理并包括本申请未公开的本技术领域中的公知常识或惯用技术手段。说明书和实施例仅被视为示例性的,本申请的真正范围和精神由下面的权利要求指出。

[0059] 应当理解的是,本申请并不局限于上面已经描述并在附图中示出的精确结构,并且可以在不脱离其范围进行各种修改和改变。本申请的范围仅由所附的权利要求来限制。

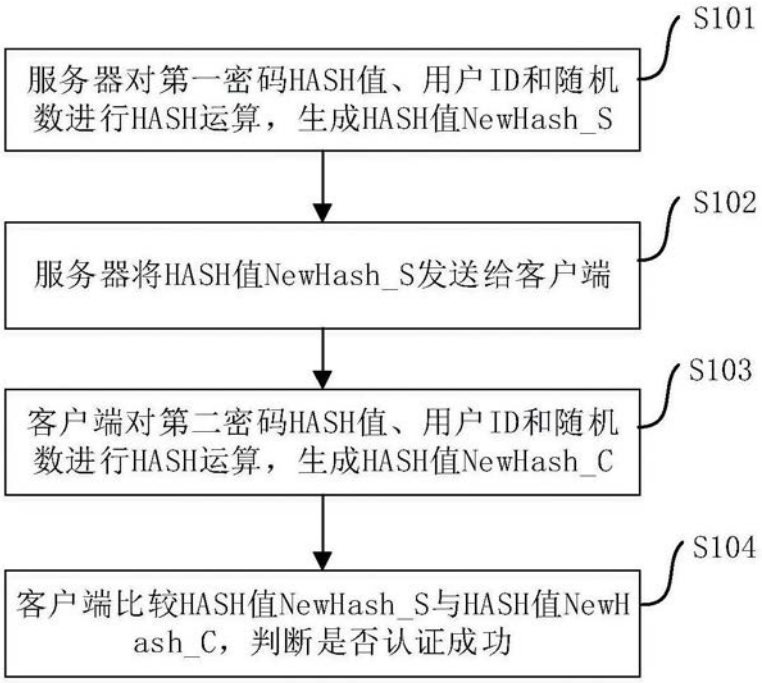

S101
服务器对第一密码HASH值、用户ID和随机数进行HASH运算，生成HASH值NewHash_S
S102
服务器将HASH值NewHash_S发送给客户端
S103
客户端对第二密码HASH值、用户ID和随机数进行HASH运算，生成HASH值NewHash_C
S104
客户端比较HASH值NewHash_S与HASH值NewHash_C，判断是否认证成功

图1

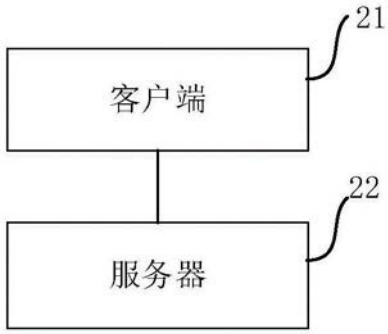

21
客户端
22
服务器

图2