(19)国家知识产权局

(12)发明专利

(10)授权公告号 CN 113014397 B
(45)授权公告日 2023.08.18

(21)申请号 202110286870.X
(22)申请日 2021.03.17
(65)同一申请的已公布的文献号
申请公布号 CN 113014397 A
(43)申请公布日 2021.06.22
(73)专利权人 杭州师范大学
地址 311121 浙江省杭州市余杭区余杭塘路2318号
(72)发明人 韩金铭 胡斌 梁锡坤 谢琪 陶利民 陶国芳 王圣宝 韩立东
(74)专利代理机构 杭州天勤知识产权代理有限公司 33224
专利代理师 胡红娟
(51)Int.Cl.
H04L 9/32(2006.01)
H04L 9/40(2022.01)
(56)对比文件
CN 108270572 A,2018.07.10
CN 111092720 A,2020.05.01
CN 112486412 A,2021.03.12
US 2018309574 A1,2018.10.25
王彩芬等.基于理想格的用户匿名口令认证密钥协商协议.计算机工程.2018,(第04期),第218-223页.
审查员 马晓晓

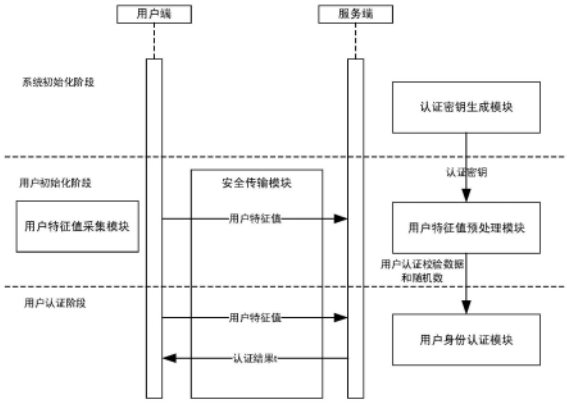
权利要求书2页 说明书6页 附图2页

(54)发明名称

一种快速安全的身份认证方法

(57)摘要

本发明公开了一种快速安全的身份认证方法,包括:接收第一信息和第二信息,所述第一信息为随机数参数和随机数矩阵,所述第二信息为通过用户端获得的第一用户信息;通过对第一信息和第二信息进行处理,获得随机盐值和第一数据;利用Ring-SIS哈希算法,通过第一数据和第一信息获得用户认证校验数据向量z,将所述用户认证校验数据向量z、所述的随机盐值和第一信息作为用户认证信息;接收需验证的用户信息,通过所述的用户认证信息对需验证的用户信息进行验证。本发明公开的认证方法通过将哈希算法与理解格结合,能够安全、高效的进行身份认证。



CN 113014397 B

1.一种快速安全的身份认证方法,其特征在于,包括:

S1:接收第一信息和第二信息,所述第一信息为随机数参数和随机数矩阵,所述第二信息为通过用户端获得的第一用户信息;

S2:通过对第一信息和第二信息进行处理,获得随机盐值和第一数据;

S3:利用环SIS哈希算法,通过第一数据和第一信息获得用户认证校验数据向量z,将所述用户认证校验数据向量z、所述的随机盐值和第一信息作为用户认证信息;

S4:接收需验证的用户信息,通过所述的用户认证信息对需验证的用户信息进行验证;

步骤S3中,通过第一数据和第一信息得出用户认证校验数据向量z的步骤如下:

所述的第一数据为m个n位比特的向量,分别为$x_0,\cdots,x_m$,进行m次循环,每次循环用向量x_i计算向量y_j,向量y_j的元素y_i,i=0,...,n,用8进制形式表示y_i的下标为$y_{i_0+8i_1}$,向量$y_{i_0+8i_1}$与随机参数的关系式如下:

$$y_{i_0+8i_1}=\sum_{k_0=0}^{7}(\omega^{16})^{i_1\cdot k_0}\left(\omega^{(2i_0+1)k_0}\cdot\sum_{k_1=0}^{7}\omega^{8k_1(2i_0+1)}\cdot x_{k_0+8k_1}\right)$$

其中,ω为特征因子,x为第一数据特征元素,$i_0,i_1,k_0,k_1,\in\{0,\cdots,7\}$,得到的标量$y_{i_0+8i_1}$与计算中间量$y'_i$的关系式为:

$$\mathbf{y}'_i=(y_{8i},y_{8i+1},\ldots,y_{8i+7})\in\mathbb{Z}_p^8,i=(0,\ldots,7)$$

得到n/8个8维向量,最后用这n/8个8维向量按序拼接,得到n维向量y_j,所有向量y_j形成n×m矩阵Y;根据矩阵A和Y利用如下公式计算得到用户认证校验数据向量z,$\mathbf{z}\in\mathbb{Z}_p^n$:

$$z_i=\sum_{j=0}^{15}a_{i,j}\cdot y_{i,j}(mod\ p)$$

其中,z_i为哈希值向量元素,α为随机数矩阵A元素,y为计算中间量拼接矩阵Y元素。

2.根据权利要求1所述的快速安全的身份认证方法,其特征在于,步骤S1中,所述的随机数参数为随机数矩阵行n、随机数矩阵列m、随机数矩阵模组p和特征因子ω,随机数长度l为(m×n)/8,所述的随机数矩阵A为m*n,通过所述的随机数参数生成随机数矩阵。

3.根据权利要求1所述的快速安全的身份认证方法,其特征在于,步骤S1中,所述的第二信息为通过用户端,经过安全协议,获得的第一用户信息。

4.根据权利要求1或3所述的快速安全的身份认证方法,其特征在于,步骤S1中,所述的第一用户信息包括用户的指纹、面部特征、用户输入的口令中的一种或多种。

5.根据权利要求2所述的快速安全的身份认证方法,其特征在于,步骤S2中,所述的通过对第一信息和第二信息进行处理的步骤如下:

根据第二信息的长度l_{pwd}与所述的随机数长度l的差值生成随机盐值,然后将第二信息和随机盐值拼接得到第一数据。

6.根据权利要求5所述的快速安全的身份认证方法,其特征在于,所述的随机盐值的长度$l_1\geqslant16$。

7.根据权利要求1所述的快速安全的身份认证方法,其特征在于,步骤S4中,所述的接

收需验证的信息，通过所述的用户认证信息对需验证的用户信息进行验证，具体步骤如下：

S41：接收需验证的信息，所述的需验证的信息为通过用户端发送，经过安全协议处理后的信息；

S42：通过需验证的信息，获得所述需验证的信息的随机盐值，将所述的需验证的用户信息的随机盐值和第一信息拼接得到第二数据；

S43：如果第二数据的长度不满足第一信息的随机数长度l，则认为认证失败，如果满足，利用环SIS哈希算法，通过第二数据和第一信息获得用户认证校验数据向量z′，如果与所述的用户认证校验数据向量z相同，则认证成功，否则认证失败。

一种快速安全的身份认证方法

技术领域

[0001] 本发明属于数据加密和隐私保护技术领域,涉及一种快速安全的身份认证方法。

背景技术

[0002] 随着互联网的迅速发展与普及,Web技术日新月异,使得诸如电子商务、网上银行等系统被广泛的应用,与此同时,网民自身的数据所面临的完全问题也日益突出,例如,拥有全球最大的中文IT技术社区的CSDN网站由于对数据的保护存在安全漏洞问题,使得黑客对其的攻击得逞,造成了600多万用户的账户和密码泄漏,这给网民带来了较大的经济损失.由此可见,保护网民自身的数据安全,使之不被窃取、不被篡改或破坏的重要性.解决这个问题的有效策略就是使用数据加密技术9.01.因此,研究出良好的数据加密技术并设计出可行的数据加密算法对于网民数据的安全起着极其重要的作用,

[0003] 目前基于用户口令或生物特征进行身份认证的实现过程中,通常采用哈希加盐的方式对用户口令或特征值进行处理,并将处理结果保存在服务端。这种处理过程中主要存在以下安全性风险:

[0004] (1)使用不安全的哈希算法,如MD5、SHA1等,SHA1算法实现了对数据的哈希运算,可以解决在数据库中以明文的形式存储用户密码的潜在安全隐患问题,但是如果直接采用哈希算法对用户的密码进行加密,加之人们习惯使用自己的姓名拼音的首字母、出生年月日期、手机号码、家庭电话号码等好记的数字或字母的组合设置密码,则攻击者能够借助字典、彩虹表较容易地获取到使用特定密码的所有用户的账号信息.因此,这种策略在面对字典、彩虹表攻击时就显得有些脆弱了,并没有达到更高的安全性。这些哈希算法已经从理论上被证明是不安全的,使用这些算法处理用户口令或特征性会大大增加安全风险;

[0005] (2)盐值安全性不够,如使用软件方式生成的盐值,随机性不高;或者盐值长度不够(如只使用4个字节长度的盐值等)导致盐值容易被攻击者猜到。此外,绝大部分实现方式使用的都是固定长度的盐值,也会增加盐值被攻击者猜中的风险;

[0006] (3)现有哈希算法不能对抗量子攻击,在量子计算环境下,即使使用穷举法等暴力攻击也能被攻破,无法满足此类环境的安全性需求。

[0007] 另外,由于现有哈希算法不支持并行实现,在某些极端情况下可能会因为处理速度不够快而导致身份认证请求响应不够及时的问题。

发明内容

[0008] 本发明提供一种快速安全的身份认证方法,所述认证方法通过将哈希算法与理想格结合,能够安全、高效的进行身份认证。

[0009] 一种快速安全的身份认证方法,包括:

[0010] S1:接收第一信息和第二信息,所述第一信息为随机数参数和随机数矩阵,所述第二信息为通过用户端获得的第一用户信息;

[0011] S2:通过对第一信息和第二信息进行处理,获得随机盐值和第一数据;

[0012] S3:利用环SIS(Ring-SIS)哈希算法,通过第一数据和第一信息获得用户认证校验数据向量z,将所述用户认证校验数据向量z、所述的随机盐值和第一信息作为用户认证信息;

[0013] S4:接收需验证的用户信息,通过所述的用户认证信息对需验证的用户信息进行验证。

[0014] 本发明提供的方法中,由于随机盐值长度可变,能够增加攻击者的猜测难度,提高了系统的安全性,由于所选取的参数和矩阵由随机数组成,相当于对用户信息进行了加密,进一步提高了安全性,并且本发明对用户信息处理过程为多个并行向量处理过程,提高了计算效率,减少了身份认证响应时间。

[0015] 步骤S1中,所述的随机数参数为随机数矩阵行n、随机数矩阵列m、随机数矩阵模组p和特征因子ω,随机数长度l为(m×n)/8,所述的随机数矩阵A为m*n,通过所述的随机数参数生成随机数矩阵。

[0016] 矩阵A是格基(lattice base),哈希过程是对格基的线性组合,成功猜中该哈希值的过程能规约为解决环SIS问题,基于格通过Ring-SIS(环SIS)哈希算法实现用户信息加盐方式基于格SVP问题,解决环SIS问题能规约为解决格的SVP问题,能够抗量子攻击,在量子环境下具有安全性。

[0017] 步骤S1中,所述的第二信息为通过用户端,经过安全协议,获得的第一用户信息。用户信息经过安全协议,使得用户信息具有的保密性,完整性和可用性。

[0018] 步骤S1中,所述的第一用户信息包括用户的指纹、面部特征、用户输入的口令中的一种或多种。

[0019] 步骤S2中,所述的通过处理第一信息和第二信息,具体步骤为根据第二信息的长度l_{pwd}与所述的随机数长度l的差值生成随机盐值,然后将第二信息和随机盐值拼接得到第一数据。生成随机盐值和第一数据的方法能够增加用户特征值的随机性,降低特征值被猜中的风险。

[0020] 所述的随机盐值长度$l_1 \geq 16$,进一步的,$16 \leq l_1 \leq 112$。较长的随机盐值长度能够增加随机性,减少攻击者猜中几率。

[0021] 步骤S3中,通过第一数据和第一信息得出用户认证校验数据向量z的步骤如下:

[0022] 所述的第一数据为m个n位比特的向量,分别为$x_0,\ldots,x_m$,进行m次循环,每次循环用向量x_i计算向量y_j,向量y_j的元素y_i,(i=0,...,n),用8进制形式表示y_i的下标为$y_{i_0+8i_1}$,向量$y_{i_0+8i_1}$与随机参数的关系式如下:

[0023] $$y_{i_0+8i_1} = \sum_{k_0=0}^{7} (\omega^{16})^{i_1 \cdot k_0} \left(\omega^{(2i_0+1)k_0} \cdot \sum_{k_1=0}^{7} \omega^{8k_1(2i_0+1)} \cdot x_{k_0+8k_1} \right)$$

[0024] 其中,ω为特征因子,x为的第一数据特征元素,$i_0,i_1,k_0,k_1,\in\{0,\ldots,7\}$,得到的标量$y_{i_0+8i_1}$与计算中间量$y'_i$的关系式为:

[0025] $$\mathbf{y}'_i = (y_{8i}, y_{8i+1}, \ldots, y_{8i+7}) \in \mathbb{Z}_p^8, i = (0,\ldots,7)$$

[0026] 得到n/8个8维向量,最后用这n/8个8维向量按序拼接,得到n维向量y_j,所有向量y_j形成n×m矩阵Y;根据矩阵A和Y利用如下公式计算得到用户认证校验数据向量$z(\mathbf{z} \in \mathbb{Z}_p^n)$:

[0027] $$z_i = \sum_{j=0}^{15} a_{i,j} \cdot y_{i,j} (mod\ p)$$

[0028] 其中，z_i为哈希值向量元素，α为随机数矩阵A元素，y为计算中间量拼接矩阵Y元素。

[0029] 计算标量y的过程相互独立，计算标量y的过程可以并行处理，提升计算速度，减少响应时间。

[0030] 步骤S4中，所述的接收需验证的用户信息，通过所述的用户认证信息对需验证的用户信息进行验证，具体步骤如下：

[0031] S41：接收需验证的用户信息，所述的需验证的用户信息为通过用户端发送，经过安全协议处理后的信息；

[0032] S42：通过需验证的用户信息，获得所述需验证的用户信息的随机盐值，将所述的需验证的用户信息的随机盐值和第一信息拼接得到第二数据；

[0033] S43：如果第二数据的长度不满足第一信息的长度参数l，则认为认证失败，如果满足，利用Ring-SIS哈希算法，通过第二数据和第一信息获得用户认证校验数据向量z′，如果与所述的用户认证校验数据向量z相同，则认证成功，否则认证失败。

[0034] 与现有技术相比，本发明的有益效果体现为：

[0035] (1)基于理想格(Ideal Lattice)的Ring-SIS技术实现的用户口令/设备特征值加盐处理方式基于格SVP问题能抗量子攻击，在量子计算环境下具有高安全性；

[0036] (2)随机盐值长度可变的处理方式提高了攻击者猜测攻击的难度，提高了系统安全性；

[0037] (3)除了随机盐值以外，在用户口令或设备特征值处理过程中加入了随机数矩阵，在随机数矩阵存储和计算过程安全性得到保证的情况下，相当于对口令或特征值进行了加密处理，进一步提高了系统安全性；

[0038] (4)本方法的口令/特征值处理过程是多个不相关向量处理过程，所以易于通过并行计算处理实现，提高了计算的并发度，减少了身份认证的响应时间。

附图说明

[0039] 图1为本发明实施例的方案结构图；

[0040] 图2为本发明实施控的制流程图。

具体实施方式

[0041] 在系统初始化阶段，对参数进行初始化，服务端生成的参数主要有随机数矩阵行n＝64，随机数矩阵列m＝16，随机数矩阵模组p＝257，以及特征因子ω＝42，和m*n的随机数矩阵，其中每个元素都是模p的随机整数，本发明提供方法的流程图如图1所示：

[0042] 所述认证密钥生成模块，在系统初始化阶段，对参数进行初始化，同时为认证过程生成数组，将参数和数组作为认证密钥保存在服务端；

[0043] 所述用户特征值采集模块，在用户初始化阶段，用户端通过相应的外部设备采集用户的指纹、面部特征等能够鉴别用户身份的特征信息，或用户输入的口令，并将采集到的

信息通过安全传输模块传递给服务端；

[0044] 所述安全传输模块，即用户将身份特征信息经过安全协议，在保证信息的保密性，完整性和可用性的前提下传输到服务端的用户特征值预处理模块；

[0045] 所述用户特征值预处理模块，即用户将特征值或认证口令通过安全传输模块发送到服务端，服务端根据特征值或认证口令的长度生成随机盐值，然后将特征值或口令和随机盐值进行拼接得到数据instr，作为用户特征值矩阵，之后利用Ring-SIS hash算法将用户特征值矩阵、认证密钥作为参数计算出用户认证校验数据向量z，随机盐值作为计算用户认证校验数据的辅助数据。保存向量z，随机盐值以及认证密钥作为用户认证信息；

[0046] 所述用户身份认证模块，即在用户身份认证阶段，用户端将用户特征信息(特征值或口令)通过安全传输模块发送到服务端，服务端找到该用户的随机盐值和认证密钥，将用户特征信息与随机盐值拼接得到待认证数据instr，如果待认证数据instr长度不满足认证密钥中的长度参数，则认证失败；否则使用用户特征值预处理模块中的Ring-SIS hash算法得到用户认证校验数据向量z′与服务端存放的用户认证校验数据向量z进行比较，如果一致，则认证成功，否则认证失败，具体方法步骤如图2所示：

[0047] 随机数矩阵A＝

[0048] {62 67 69 45 79 55 44 6c 4e 78 68 53 77 41 77 71 4a 45 61 51 42 4d
7530 38 37 75 6f 67 79 59 4d 38 6f 64 70 36 66 35 42 7a 6e 6e 73 54 63 6f 6f
6953 53 4a 34 79 49 39 31 76 51 49 6e 48 4c 71 6a 34 65 61 59 43 45 54 5a
3641 6c 30 66 4a 32 63 5a 59 73 55 51 53 43 6f 56 61 44 73 73 58 50 75 36
4964 66 52 5a 7a 35 76 44 39 68 6a 35 49 64 48 47 7a 62 75 4d 32 68 47 48
6749 65 5a 66 79 47 4b 44 71 66 52 43 43 38 64 55 4b 77 78 4c 41 38 4d 68
344e 70 39 44 37 4c 50 4d 32 71 66 69 6d 4e 78 64 66 7a 6e 6c 65 75 57 50
6141 4b 50 6b 72 31 34 4c 4b 4e 57 42 54 63 75 55 76 32 39 72 4d 58 6c 37
4e53 42 43 48 67 6a 72 67 68 65 52 67 75 65 78 38 47 38 6f 4a 6d 50 51 69
6d6a 65 39 62 4a 4e 6a 32 45 52 4a 4e 42 74 56 6e 41 44 58 53 6e 70 54 70 74
6e37 5a 66 72 51 50 30 71 58 6d 65 4b 7a 6e 50 38 37 4f 38 49 58 77 4a 52
434b 75 41 41 4b 6f 32 61 54 30 50 44 67 4c 49 70 6e 44 38 6e 58 7a 36 42
325a 55 47 75 46 37 37 70 75 39 6c 38 36 33 32 6c 62 69 63 30 32 6f 56 32
7232 62 4a 4e 6a 32 45 52 4a 4e 42 74 56 6e 41 44 58 53 6e 70 54 70 74 6e 37
5a66 72 51 50 30 71 58 6d 65 4b 7a 6e 50 38 37 4f 38 49 58 77 4a 52 43 4b
7541 41 4b 6f 32 61 54 30 50 44 67 4c 49 70 6e 44 38 6e 58 7a 36 42 32 5a
5547 75 46 37 37 70 75 39 6c 38 36 33 32 6c 62 69 63 30 32 6f 56 32 72 32
624a 4e 6a 32 45 52 4a 4e 42 74 56 6e 41 44 58 53 6e 70 54 70 74 6e 37 5a 66
7251 50 30 71 58 6d 65 4b 7a 6e 50 38 37 4f 38 49 58 77 4a 52 43 4b 75 41
414b 6f 32 61 54 30 50 44 67 4c 49 70 6e 44 38 6e 58 7a 36 42 32 5a 55 47
7546 37 37 70 75 39 6c 38 36 33 32 6c 62 69 63 30 32 6f 56 32 72 32 62 4a
4e6a 32 45 52 4a 4e 42 74 56 6e 41 44 58 53 6e 70 54 70 74 6e 37 5a 66 72
5150 30 71 58 6d 65 4b 7a 6e 50 38 37 4f 38 49 58 77 4a 52 43 4b 75 41 41 4b
6f32 61 54 30 50 44 67 4c 49 70 6e 44 38 6e 58 7a 36 42 32 5a 55 47 75 46

3737 70 75 39 6c 38 36 33 32 6c 62 69 63 30 32 6f 56 32 72 32 68 52 63 44
5163 36 43 67 30 30 73 71 6d 62 35 67 68 51 37 43 67 7a 44 45 59 70 65 77
6e55 63 33 77 39 6e 6c 5a 4e 43 66 4c 47 4e 67 61 53 79 49 58 62 66 46 57
617a 43 49 69 52 41 51 34 53 77 38 67 77 43 6d 6e 35 34 6a 77 54 61 54 50
746e 58 6d 46 57 62 43 69 63 36 4e 46 71 4a 38 30 69 79 6b 70 34 44 46 39
7a30 44 55 65 71 34 74 38 58 72 71 4c 73 6f 7a 70 32 35 78 72 38 4c 46 43
764a 67 75 6f 64 61 78 5a 62 58 70 72 66 44 77 4b 38 58 5a 57 4b 53 52 49
554b 30 49 49 53 4c 59 77 71 38 67 53 4d 44 59 44 4d 37 4e 45 72 4b 32 67
4579 42 51 72 49 4a 6b 6b 38 47 62 64 77 47 51 41 4d 34 65 55 38 4d 74 7a
6a7a 75 46 51 4f 71 30 30 74 31 4a 46 72 39 52 6c 50 42 6a 4b 56 66 65 36 6f
4b35 4a 44 68 6e 68 57 31 79 46 35 4f 63 65 67 62 79 76 47 52 6d 78 64 69
4771 71 4a 64 6e 34 34 5a 39 6c 39 46 66 72 4c 4f 43 41 42 68 31 62 31 63 4b
4e49 54 62 47 32 7a 76 55 64 63 69 65 36 4b 43 59 6c 63 6f 6e 31 41 50 35
6330 47 49 39 46 34 50 52 67 4b 4a 53 4e 5a 65 6c 46 6a 43 71 4c 35 64 45 6d
67 69 55 54 67 47 33 68 4b 63 30 7a 77 47 78 79 48 6c 6d 69 6f 45 30 4e 72 37
4b 47 6b 51 6e 45 76 74 43 75 51 47 4a 51 56 57 31 75 33 31 62 55 6c 6c 31 6f
42 58 63 41 77 79 44 39 78 4d 65 35 63 76 61 42 41}，

[0049] 认证密钥包括n、m、p、ω以及A。

[0050] 对于每个需要认证的用户，采集该用户的口令或生物特征值(指纹、人脸特征值等，长度为l_{pwd}，l_{pwd}<=112)并通过安全通道将该特征值或口令发送至服务端，服务端为该用户生成随机盐值，然后将特征值或口令62 75 34 4a 31 6e 57 4a 34 7a 64 6b 37 50 56 6a 54 67 69 74 58 45 70 30 6c 34 52 67 75 59 53 37 4e 7a 54 68 54 33 4e 7a 57 6a 66 5a 36 52 55 49 78 70 32 48 34 33 77 52和随机盐值

[0051] 6b 77 6b 68 6e 48 4d 55 61 4d 6b 4d 67 36 53 56 33 46 6b 57 48 66 49 72 74 47 34 73 32 30 61 38 4f 32 44 44 6b 55 58 64 56 72 57 53 35 76 453 1 6f 4d 51 67 6a 4a 4d 68 78 57 49 32 57 6b 6f 47 44 64 6c 78 59 77 4f 4e进行拼接得到数据instr：

[0052] 6b 77 6b 68 6e 48 4d 55 61 4d 6b 4d 67 36 53 56 33 46 6b 57 48 66 49 72 74 47 34 73 32 30 61 38 4f 32 44 44 6b 55 58 64 56 72 57 53 35 76 45 31 6f 4d 51 67 6a 4a 4d 68 78 57 49 32 57 6b 6f 47 44 64 6c 78 59 77 4f 4e62 75 34 4a 31 6e 57 4a 34 7a 64 6b 37 50 56 6a 54 67 69 74 58 45 70 30 6c 34 52 67 75 59 53 37 4e 7a 54 68 54 33 4e 7a 57 6a 66 5a 36 52 55 49 78 70 32 48 34 33 77 52找到认证密钥进行Ring-SIS hash运算：

[0053] 将instr看成16个64位比特的向量，分别为$x_0,...,x_{15}$，进行16次循环，每次循环用向量x_i计算向量y_j，向量y_j的元素为y_i，(i=0,...,63)，用8进制形式表示y_i的下标，则y_i可以表示为：$y_{i_0+8i_1}$。计算：

[0054] $$y_{i_0+8i_1}=\sum_{k_0=0}^{7}(\omega^{16})^{i_1\cdot k_0}\left(\omega^{(2i_0+1)k_0}\cdot\sum_{k_1=0}^{7}\omega^{8k_1(2i_0+1)}\cdot x_{k_0+8k_1}\right)$$

[0055] 其中i_0,i_1,k_0,k_1,∈{0,...,7},将计算得到的所有标量$y_{i_0+8i_1}$以下公式计算:

[0056] $$\mathbf{y}_i^{'} = (y_{8i}, y_{8i+1}, \cdots, y_{8i+7}) \in \mathbb{Z}_{257}^{8}, i = (0,\ldots,7)$$

[0057] 得到8个8维向量,最后用这8个8维向量按序拼接,得到64维向量yj,所有向量yj形成64×16矩阵Y;根据矩阵A和Y利用以下公式计算得到向量z $(\mathbf{z} \in \mathbb{Z}_{257}^{64})$,

[0058] $$z_i = \sum_{j=0}^{15} a_{i,j} \cdot y_{i,j} (mod\ 257)$$

[0059] 得到用户认证校验数据向量z。

[0060] 478c 11d33e 69126a 1879e1 b2 bd d87536d9 b6869721a0 abce 1c 2fcf 22d470de b81212433760590b 50c647c6 e54c 66eb b999ac 740a4991e1768f 7fb474b6 da a287

[0061] 保存向量z,随机盐值以及认证密钥作为用户认证信息。

[0062] 认证过程中,客户端采集用户口令或生物特征值(指纹、人脸特征值等),并通过安全的通道把口令或特征值发送至服务端,服务端找到用户的随机盐值以及认证密钥,拼接得到待认证instr,如果待认证数据instr长度不满足认证密钥中的长度参数,则认证失败;否则用方案中的Ring-SIS hash代入参数计算得到当前认证向量z′,如果z′和服务端存放的认证向量z相同,则认证成功,否则本次认证失败。

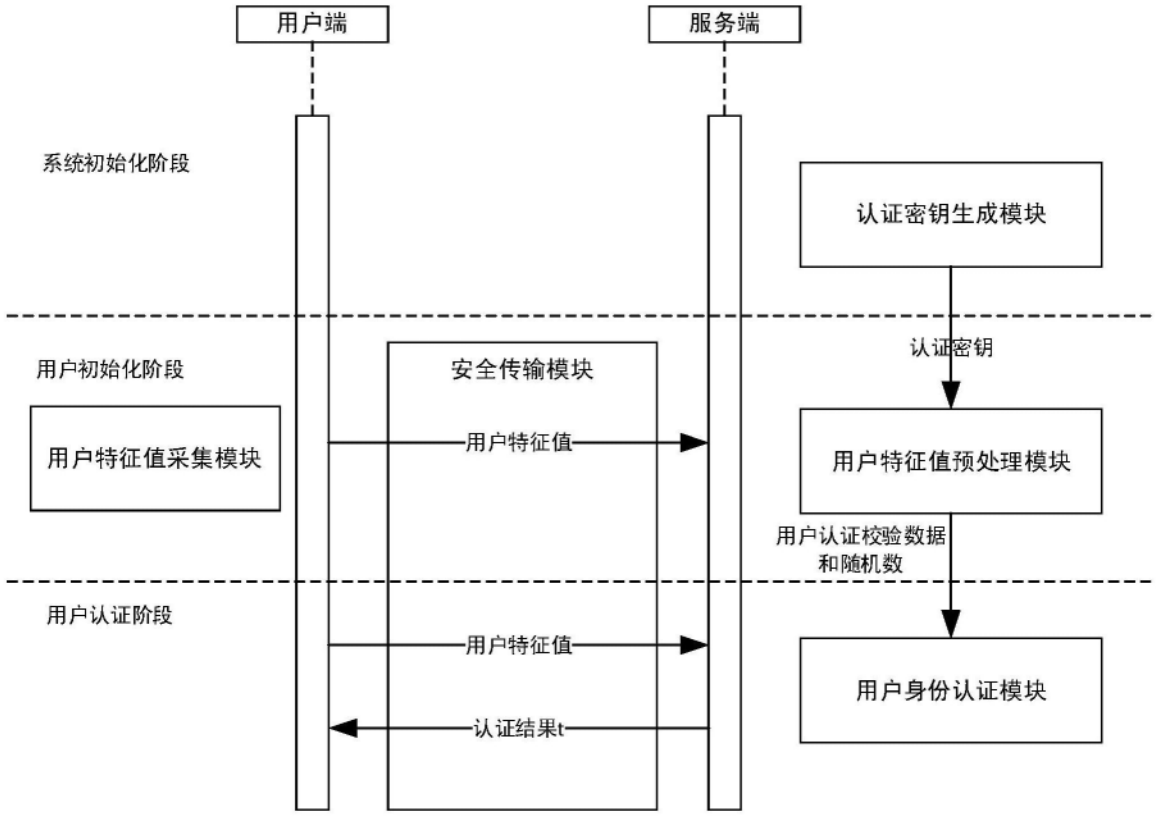

用户端
服务端
系统初始化阶段
认证密钥生成模块
认证密钥
用户初始化阶段
安全传输模块
用户特征值采集模块
用户特征值
用户特征值预处理模块
用户认证校验数据
和随机数
用户认证阶段
用户特征值
用户身份认证模块
认证结果t

图1

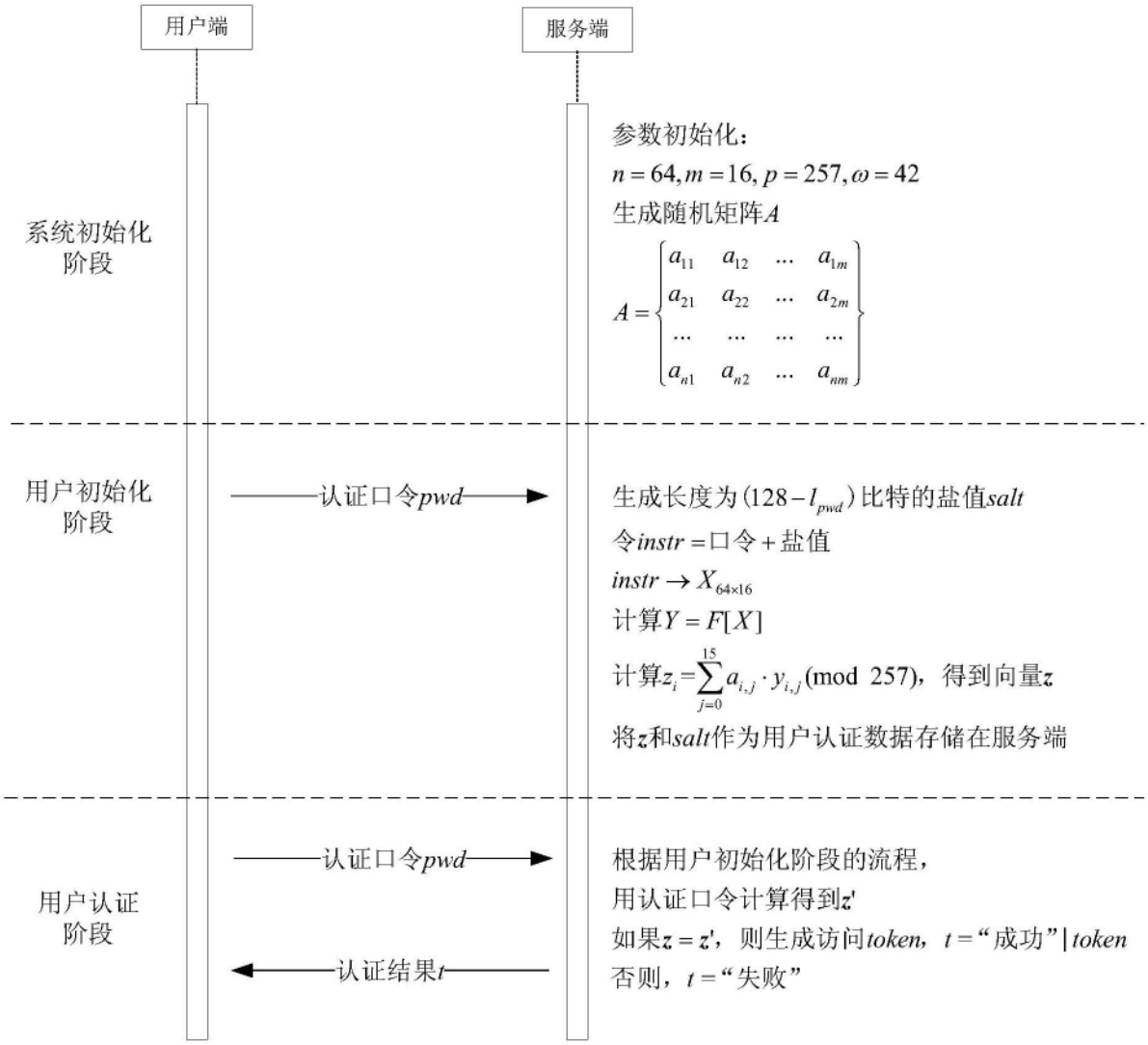

用户端
服务端
系统初始化阶段
参数初始化：
$n=64, m=16, p=257, \omega=42$
生成随机矩阵A
$A=\begin{Bmatrix} a_{11} & a_{12} & \cdots & a_{1m} \\ a_{21} & a_{22} & \cdots & a_{2m} \\ \cdots & \cdots & \cdots & \cdots \\ a_{n1} & a_{n2} & \cdots & a_{nm} \end{Bmatrix}$
用户初始化阶段
认证口令pwd
生成长度为$(128-l_{pwd})$比特的盐值$salt$
令$instr$=口令+盐值
$instr \rightarrow X_{64\times16}$
计算$Y=F[X]$
计算$z_i=\sum_{j=0}^{15} a_{i,j} \cdot y_{i,j} \pmod{257}$，得到向量$z$
将z和$salt$作为用户认证数据存储在服务端
用户认证阶段
认证口令pwd
根据用户初始化阶段的流程，
用认证口令计算得到z'
如果$z=z'$，则生成访问$token$，t="成功"|$token$
否则，t="失败"
认证结果t

图2