(19)国家知识产权局

(12)发明专利

(10)授权公告号 CN 114866260 B
(45)授权公告日 2022.10.28

(21)申请号 202210782316.5
(22)申请日 2022.07.05
(65)同一申请的已公布的文献号
申请公布号 CN 114866260 A
(43)申请公布日 2022.08.05
(73)专利权人 杭州天谷信息科技有限公司
地址 310012 浙江省杭州市西湖区西斗门路3号天堂软件园D幢19层
(72)发明人 钟一民 陈传义 郭峰 金宏洲 程亮
(74)专利代理机构 杭州裕阳联合专利代理有限公司 33289
专利代理师 王榧
(51)Int.Cl.
*H04L 9/32*(2006.01)

(56)对比文件
CN 113922962 A,2022.01.11
CN 113918899 A,2022.01.11
CN 110086599 A,2019.08.02
CN 110298152 A,2019.10.01
CN 111095327 A,2020.05.01
CN 112446701 A,2021.03.05
US 2017109759 A1,2017.04.20
WO 2019233951 A1,2019.12.12
马晓婷等.基于区块链技术的跨域认证方案.《电子学报》.2018,(第11期),

审查员 裴广坤

权利要求书2页 说明书12页 附图1页

(54)发明名称

一种变色龙哈希分布式身份使用方法和系统

(57)摘要

本发明涉及计算机技术领域中的一种变色龙哈希分布式身份使用方法和系统,包括以下步骤:使用变色龙哈希算法对分布式身份生成可验证凭证,并验证可验证凭证的可靠性;根据可验证凭证生成可验证表述,并验证可验证表述的可靠性;根据分布式身份持有者所提出的声明需求,重新生成可验证凭证,得到更改版可验证凭证,并验证更改版可验证凭证的可靠性,具有降低可验证凭证颁发方工作量的优点,突破了现有可验证凭证结构无法实现对声明内容进行任意排列组合的功能的瓶颈。


CN 114866260 B

1.一种变色龙哈希分布式身份使用方法,其特征在于,包括以下步骤:

使用变色龙哈希算法对分布式身份生成可验证凭证,并验证所述可验证凭证的可靠性,其中,使用变色龙哈希算法对分布式身份生成可验证凭证包括:获取分布式身份持有者所提供的若干个声明;计算分布式身份的普通哈希值和变色龙随机数,并根据所述普通哈希值和变色龙随机数计算第一变色龙哈希值;根据各个所述声明,计算各个所述声明对应的变色龙随机数;使用可验证凭证颁发者私钥对所述第一变色龙哈希值进行签名得到第一签名;

根据所述可验证凭证生成可验证表述,并验证所述可验证表述的可靠性;

根据分布式身份持有者所提出的声明需求,重新生成可验证凭证,得到更改版可验证凭证,并验证所述更改版可验证凭证的可靠性。

2.根据权利要求1所述的一种变色龙哈希分布式身份使用方法,其特征在于,对所述可验证凭证进行可靠性验证,包括以下步骤:

验证所述第一签名的有效性;

根据各个所述声明,计算各个所述声明对应的第二变色龙哈希值;

判断所述第二变色龙哈希值是否与第一变色龙哈希值相等,且所述第一签名有效,若是,则所述可验证凭证可靠,反之,则所述可验证凭证不可靠。

3.根据权利要求2所述的一种变色龙哈希分布式身份使用方法,其特征在于,根据所述可验证凭证生成可验证表述,包括以下步骤:

获取可验证表述的非签名数据,其中,所述非签名数据包括可验证表述元数据和一个或多个的可验证凭证;

使用分布式身份持有者私钥对所述非签名数据进行签名得到第二签名。

4.根据权利要求3所述的一种变色龙哈希分布式身份使用方法,其特征在于,验证所述可验证表述的可靠性,包括以下步骤:

验证所述第一签名和第二签名的有效性;

根据各个所述声明,计算各个所述声明对应的第二变色龙哈希值;

判断所述第二变色龙哈希值是否与第一变色龙哈希值相等,且所述第一签名与第二签名均有效,若是,则所述可验证表述可靠,反之,则所述可验证表述不可靠。

5.根据权利要求1所述的一种变色龙哈希分布式身份使用方法,其特征在于,所述声明需求包括增加声明、删减声明、声明内容修改或声明顺序修改中的任意一种。

6.一种变色龙哈希分布式身份使用系统,其特征在于,包括第一生成检验单元、第二生成检验单元和更新检验单元;

所述第一生成检验单元用于使用变色龙哈希算法对分布式身份生成可验证凭证,并验证所述可验证凭证的可靠性,

其中,所述第一生成检验单元包括第一生成单元,且所述第一生成单元包括获取单元、第一计算单元和第一签名单元,所述获取单元用于获取分布式身份持有者所提供的若干个声明;所述第一计算单元用于计算分布式身份的普通哈希值和变色龙随机数,并根据所述普通哈希值和变色龙随机数计算第一变色龙哈希值,并根据各个所述声明计算各个所述声明对应的变色龙随机数;所述第一签名单元用于使用可验证凭证颁发者私钥对所述第一变色龙哈希值进行签名得到第一签名;

所述第二生成检验单元用于根据所述可验证凭证生成可验证表述，并验证所述可验证表述的可靠性；

所述更新检验单元用于根据分布式身份持有者所提出的声明需求，重新生成可验证凭证，得到更改版可验证凭证，并验证所述更改版可验证凭证的可靠性。

7.根据权利要求6所述的一种变色龙哈希分布式身份使用系统，其特征在于，所述第一生成检验单元还包括第一验证单元，且所述第一验证单元包括第一检验单元、第二计算单元和第一判定单元；

所述第一检验单元用于验证所述第一签名的有效性；

所述第二计算单元用于根据各个所述声明，计算各个所述声明对应的第二变色龙哈希值；

所述第一判定单元用于判断所述第二变色龙哈希值是否与第一变色龙哈希值相等，且所述第一签名有效，若是，则所述可验证凭证可靠，反之，则所述可验证凭证不可靠。

8.根据权利要求7所述的一种变色龙哈希分布式身份使用系统，其特征在于，所述第二生成检验单元包括第二生成单元，且所述第二生成单元包括数据获取单元和第二签名单元；

所述数据获取单元用于获取可验证表述的非签名数据，其中，所述非签名数据包括可验证表述元数据和一个或多个的可验证凭证；

所述第二签名单元用于使用分布式身份持有者私钥对所述非签名数据进行签名得到第二签名。

9.根据权利要求8所述的一种变色龙哈希分布式身份使用系统，其特征在于，所述第二生成检验单元还包括第二验证单元，且所述第二验证单元包括第二检验单元、第三计算单元和第二判定单元；

所述第二检验单元用于验证所述第一签名和第二签名的有效性；

所述第三计算单元用于根据各个所述声明，计算各个所述声明对应的第二变色龙哈希值；

所述第二判定单元用于判断所述第二变色龙哈希值是否与第一变色龙哈希值相等，且所述第一签名与第二签名均有效，若是，则所述可验证表述可靠，反之，则所述可验证表述不可靠。

# 一种变色龙哈希分布式身份使用方法和系统

## 技术领域

[0001] 本发明涉及计算机技术领域,具体涉及一种变色龙哈希分布式身份使用方法和系统。

## 背景技术

[0002] 现有DID(分布式身份)使用时,可验证凭证结构中的Merkle树相关内容使得可验证凭证的数据结构显得较为复杂,因为Merkle树中每个数据都和前后的其他数据有关联,如果可验证凭证进行选择性披露,必须同时提供非披露数据的哈希值以便验证方正确计算Merkle树的树根并用于数字签名的验证,因此该方法验证数字签名时还需要额外计算Merkle树的信息;在根据可验证凭证出示可验证表述时,由于Merkle树相关内容严格控制了声明内容的顺序,现有可验证凭证结构无法实现对声明内容进行任意排列组合的功能,无法满足一些需要对可验证凭证更换声明顺序的场景的隐私保护需求,例如每个可验证凭证的验证者对声明的关注度是不同的,最优做法应该是每次验证前将声明按照对方的关注程度进行排序后再将排序后的声明发给对方,以便对方验证并比对声明信息,而目前的Merkle树结构无法满足此功能。

[0003] 另一方面,现有DID使用时,可验证凭证结构中的Merkle树相关内容使得可验证凭证的数据结构发生任何改变后均需要重新计算可验证凭证的数字签名及Merkle树相关的内容,增大了可验证凭证颁发方的负荷。

## 发明内容

[0004] 本发明针对现有技术中的缺点,提供了一种变色龙哈希分布式身份使用方法和系统,具有降低可验证凭证颁发方工作量的优点,突破了现有可验证凭证结构无法实现对声明内容进行任意排列组合的功能的瓶颈。

[0005] 为了解决上述技术问题,本发明通过下述技术方案得以解决:

[0006] 一种变色龙哈希分布式身份使用方法,包括以下步骤:

[0007] 使用变色龙哈希算法对分布式身份生成可验证凭证,并对所述可验证凭证进行可靠性验证;

[0008] 根据所述可验证凭证生成可验证表述,并验证所述可验证表述的可靠性;

[0009] 根据分布式身份持有者所提出的声明需求,重新生成可验证凭证,得到更改版可验证凭证,并验证所述更改版可验证凭证的可靠性。

[0010] 可选地,使用变色龙哈希算法对分布式身份生成可验证凭证,包括以下步骤:

[0011] 获取分布式身份持有者所提供的若干个声明;

[0012] 计算分布式身份的普通哈希值和变色龙随机数,并根据所述普通哈希值和变色龙随机数计算第一变色龙哈希值;

[0013] 根据各个所述声明,计算各个所述声明对应的变色龙随机数;

[0014] 使用可验证凭证颁发者私钥对所述第一变色龙哈希值进行签名得到第一签名。

[0015] 可选地，对所述可验证凭证进行可靠性验证，包括以下步骤：

[0016] 验证所述第一签名的有效性；

[0017] 根据各个所述声明，计算各个所述声明对应的第二变色龙哈希值；

[0018] 判断所述第二变色龙哈希值是否与第一变色龙哈希值相等，且所述第一签名有效，若是，则所述可验证凭证可靠，反之，则所述可验证凭证不可靠。

[0019] 可选地，根据所述可验证凭证生成可验证表述，包括以下步骤：

[0020] 获取可验证表述的非签名数据，其中，所述非签名数据包括可验证表述元数据和一个或多个的可验证凭证；

[0021] 使用分布式身份持有者私钥对所述非签名数据进行签名得到第二签名。

[0022] 可选地，验证所述可验证表述的可靠性，包括以下步骤：

[0023] 验证所述第一签名和第二签名的有效性；

[0024] 根据各个所述声明，计算各个所述声明对应的第二变色龙哈希值；

[0025] 判断所述第二变色龙哈希值是否与第一变色龙哈希值相等，且所述第一签名与第二签名均有效，若是，则所述可验证表述可靠，反之，则所述可验证表述不可靠。

[0026] 可选地，所述声明需求包括增加声明、删减声明、声明内容修改或声明顺序修改中的任意一种。

[0027] 一种变色龙哈希分布式身份使用系统，包括第一生成检验单元、第二生成检验单元和更新检验单元；

[0028] 所述第一生成检验单元用于使用变色龙哈希算法对分布式身份生成可验证凭证，并对所述可验证凭证进行可靠性验证；

[0029] 所述第二生成检验单元用于根据所述可验证凭证生成可验证表述，并验证所述可验证表述的可靠性；

[0030] 所述更新检验单元用于根据分布式身份持有者所提出的声明需求，重新生成可验证凭证，得到更改版可验证凭证，并验证所述更改版可验证凭证的可靠性。

[0031] 可选地，所述第一生成检验单元包括第一生成单元，且所述第一生成单元包括获取单元、第一计算单元和第一签名单元；

[0032] 所述获取单元用于获取分布式身份持有者所提供的若干个声明；

[0033] 所述第一计算单元用于计算分布式身份的普通哈希值和变色龙随机数，并根据所述普通哈希值和变色龙随机数计算第一变色龙哈希值，并根据各个所述声明计算各个所述声明对应的变色龙随机数；

[0034] 所述第一签名单元用于使用可验证凭证颁发者私钥对所述第一变色龙哈希值进行签名得到第一签名。

[0035] 可选地，所述第一生成检验单元还包括第一验证单元，且所述第一验证单元包括第一检验单元、第二计算单元和第一判定单元；

[0036] 所述第一检验单元用于验证所述第一签名的有效性；

[0037] 所述第二计算单元用于根据各个所述声明，计算各个所述声明对应的第二变色龙哈希值；

[0038] 所述第一判定单元用于判断所述第二变色龙哈希值是否与第一变色龙哈希值相等，且所述第一签名有效，若是，则所述可验证凭证可靠，反之，则所述可验证凭证不可靠。

[0039] 可选地，所述第二生成检验单元包括第二生成单元，且所述第二生成单元包括数据获取单元和第二签名单元；

[0040] 所述数据获取单元用于获取可验证表述的非签名数据，其中，所述非签名数据包括可验证表述元数据和一个或多个的可验证凭证；

[0041] 所述第二签名单元用于使用分布式身份持有者私钥对所述非签名数据进行签名得到第二签名。

[0042] 可选地，所述第二生成检验单元还包括第二验证单元，且所述第二验证单元包括第二检验单元、第三计算单元和第二判定单元；

[0043] 所述第二检验单元用于验证所述第一签名和第二签名的有效性；

[0044] 所述第三计算单元用于根据各个所述声明，计算各个所述声明对应的第二变色龙哈希值；

[0045] 所述第二判定单元用于判断所述第二变色龙哈希值是否与第一变色龙哈希值相等，且所述第一签名与第二签名均有效，若是，则所述可验证表述可靠，反之，则所述可验证表述不可靠。

[0046] 采用本发明提供的技术方案，与现有技术相比，具有如下有益效果：

[0047] 1、本发明中的DID使用时，改进后的可验证凭证结构去掉了Merkle树相关的部分，增加了变色龙随机数，可验证凭证的数据结构变得简单，并且可以保证所有内容不可篡改；在根据可验证凭证出示可验证表述时，由于没有Merkle树相关内容对声明内容顺序的严格控制，改进后的可验证凭证结构可以对声明内容进行任意取舍和任意排列组合，且仍能验证通过，满足DID持有者对隐私保护的需求的同时满足更多应用场景下对声明重新排序的需求；

[0048] 2、本发明中的DID使用时，因新增、删除、修改、交换声明条目而需要可验证凭证颁发方重新生成可验证凭证时，改进后的可验证凭证结构使得可验证凭证颁发方无需重新计算可验证凭证的数字签名及Merkle树相关的内容，减轻了可验证凭证颁发方的负荷。

## 附图说明

[0049] 为了更清楚地说明本发明实施例或现有技术中的技术方案，下面将对实施例或现有技术描述中所需要使用的附图作简单的介绍，显而易见地，下面描述中的附图仅仅是本发明的一些实施例，对于本领域普通技术人员来讲，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。

[0050] 图1为本实施例一提出的角色之间的关系图。

## 具体实施方式

[0051] 下面结合实施例对本发明做进一步的详细说明，以下实施例是对本发明的解释而本发明并不局限于以下实施例。

[0052] 实施例中的专有名词解释说明：

[0053] DID是一种特定格式的字符串，用来代表一个实体的数字身份。DID的标识格式为：did:example:123456789abcdefghijk，其中，前缀did是固定的，表示这个字符串是一个did标识字符串；中间的example被称为DID方法，就是用来表示这个DID标识是用哪一套方案

(方法)来进行定义和操作的,DID方法可以自定义;最后的长字符串部分是在该DID方法下的唯一标识字符串。

[0054] DID文档是一种对DID身份进行存证的文档,一般是对DID相关信息进行关联,尤其是建立DID与其公钥的关联,然后把DID标识作为Key,把DID文档作为Value存储到区块链中,利用区块链不可篡改、共享数据的特点,使得DID验证者能快速访问获取DID持有者的公钥。

[0055] VC是Verifiable Claims 或 Verifiable Credentials的简称,可译为可验证声明或可验证凭证,是一个 DID 给另一个 DID 的某些属性做背书而发出的描述性声明,并附加自己的数字签名,用以证明这些属性的真实性,是一种对应于DID应用场景的数字证书。

[0056] VP是Verifiable Presentation的简称,可译为可验证表述,是VC持有者向验证者表明自己身份的数据。一般情况下,可以直接出示VC全文,但是在某些情况下,出于隐私保护的需要,我们并不需要出示完整的VC内容,只希望选择性披露某些属性,此时可以在VP中对披露的VC属性以明文示出,对不披露的VC属性以哈希值示出,VP的接收方可以根据明文及哈希值的VC属性计算Merkle树的根后进行数字签名的验证并得到通过。

[0057] 本申请如有对DID、DID文档、VC、VP及相关概念及结构的描述不够详尽之处,可参考业界的DID标准或实际实现方式。例如,2019年W3C(World Wide Web Consortium)发布DID的首个公开工作草案:“Decentralized Identifiers (DIDs) v1.0”。

[0058] 实施例一

[0059] 如图1所示,一种变色龙哈希分布式身份使用方法,包括以下步骤:使用变色龙哈希算法对分布式身份生成可验证凭证,并对可验证凭证进行可靠性验证;具体地,使用变色龙哈希算法对分布式身份生成可验证凭证,包括以下步骤:获取分布式身份持有者所提供的若干个声明;计算分布式身份的普通哈希值和变色龙随机数,并根据普通哈希值和变色龙随机数计算第一变色龙哈希值;使用可验证凭证颁发者私钥对第一变色龙哈希值进行签名得到第一签名。

[0060] 更进一步地,分布式身份持有者向可验证凭证生成者提交若干个声明,在本实施例中可将若干个声明记为claim1~claimN,而一个声明为一个信息条目,例如“姓名:XX”、“年龄:XX”、“住址:XX省XX市XX区XX路XX号”等均可以包含在声明中的信息条目中,可验证凭证声明的具体内容以及对应的证明可以记为{ claim1, r1;claim2, r2;……;claimi, ri ;……;claimN, rN },设定1≤i≤N,则ri的计算方式具体如下:

[0061] 首先将分布式身份作为变色龙消息,计算普通哈希值MD=H(DID),选择变色龙随机数为RD=H(metadata),其中metadata为可验证凭证的元数据,包含固定不变的信息,如DID、发行目的、发行时间、有效期等。对MD、RD进行变色龙哈希生成第一变色龙哈希值CHD=CH(MD,RD,PKS),并使用可验证凭证颁发者私钥SKS对第一变色龙哈希值CHD进行签名得到第一签名SIGD1,其中可验证凭证颁发者的公钥标记为PKS,变色龙哈希使用的公钥为PKS,使得私钥作为变色龙哈希的陷门密钥,然后计算新的变色龙消息claimi的普通哈希MCi=H(claimi),可验证凭证颁发者利用SKS、MD、RD、MCi求解变色龙随机数,得到MCi对应的变色龙随机数为ri。根据变色龙哈希的特性,可验证凭证颁发者以外的任何成员均无法计算得到ri。

[0062] 由此可见,可验证凭证结构中原有的Merkle树相关内容使得可验证凭证的数据结构显得较为复杂,改进后的可验证凭证结构去掉了Merkle树相关的部分,增加了变色龙随机数,可验证凭证的数据结构也变得更为简单。

[0063] 可验证凭证生成后,由可验证凭证颁发者发送给分布式身份持有者,并对可验证凭证进行可靠性验证,具体包括以下步骤:验证第一签名的有效性;根据各个声明,计算各个声明对应的第二变色龙哈希值;判断第二变色龙哈希值是否与第一变色龙哈希值相等,且第一签名有效,若是,则可验证凭证可靠,反之,则可验证凭证不可靠。

[0064] 进一步地,首先使用可验证凭证颁发者的公钥对第一签名进行验证,如若通过则表示可验证凭证颁发者对第一变色龙哈希值的签名有效,且分布式身份、元数据(metadata)均未被篡改,因为如果两者被篡改,将导致MD或变色龙随机数RD发生变化,MD或变色龙随机数RD的变化又将导致第一变色龙哈希值CHD的变化,第一变色龙哈希值CHD的变化将最终导致第一签名无法验证通过,即第一签名无效;反之,则可以得知分布式身份、metadata均未被篡改。

[0065] 另一方面,还需要验证各个声明的变色龙哈希值,即第二变色龙哈希值CH(MCi,ri,PKS),具体地,验证CH(MCi,ri,PKS)是否等于CHD,如相等,则说明MCi均未被可验证凭证颁发者以外的成员修改,而可验证凭证颁发者被信任为不会随意对MCi进行修改,因此分布式身份持有者信任MCi,因为,如果MCi被可验证凭证颁发者以外的成员篡改,由于可验证凭证颁发者以外的成员无法得到对应的ri,将导致第二变色龙哈希值CH(MCi,ri,PKS)发生变化,第二变色龙哈希值CH(MCi,ri,PKS)的变化将导致其无法等于第一变色龙哈希值CHD;反之,可以得知MCi未被可验证凭证颁发者以外的成员篡改。可验证凭证颁发者被信任为不会随意对MCi进行修改的理由在于,可验证凭证颁发者一般为权威性机构,安全保障措施完备,其私钥一般得到更完善的保护,例如由多个管理员共同控制;另外,可验证凭证颁发者还存在更完善的审计机制。

[0066] 待第一签名以及各个声明的变色龙哈希值全部验证步骤通过后,则分布式身份持有者信任整个可验证凭证,该可验证凭证可靠,同时,改进后的可验证凭证结构可以保证所有内容不可篡改。

[0067] 验证完成可验证凭证后,分布式身份持有者可根据可验证凭证生成可验证表述,并验证可验证表述的可靠性;具体地,根据可验证凭证生成可验证表述,包括以下步骤:获取可验证表述的非签名数据,其中,非签名数据包括可验证表述元数据和一个或多个的可验证凭证;使用分布式身份持有者私钥对非签名数据进行签名得到第二签名。

[0068] 对于分布式身份持有者而言,为保证身份信息的安全性,通常需要将一部分敏感信息进行隐藏,此时分布式持有者需要生成可验证表述,在本实施例中设定验证凭证声明的具体内容以及对应的证明可以记为{ claim1, r1;claim2, r2;……;claimi, ri },claimi以后的声明为敏感信息,此时需要使用分布式身份持有者私钥对非签名数据进行签名得到第二签名,从而使得根据可验证凭证出示可验证表述时,实现了敏感信息的隐藏,即,改进后的可验证凭证可以对声明内容进行任意取舍和任意排列组合,满足分布式身份持有者对隐私保护的需求。

[0069] 同时,可验证表述也需要验证其可靠性,具体包括以下步骤:验证第一签名和第二签名的有效性;根据各个声明,计算各个声明对应的第二变色龙哈希值;判断第二变色龙哈

希值是否与第一变色龙哈希值相等，且第一签名与第二签名均有效，若是，则可验证表述可靠，反之，则可验证表述不可靠，需要说明的是，验证第二签名的有效性与验证第一签名的有效性的方法相同，在此不做重复赘述，除此之外，由于是对每条声明进行验证，因此即使分布式身份对可验证表述中的声明顺序进行任意排列组合，均能通过验证。

[0070] 另一方面，在验证第二签名是否有效时，使用分布式身份持有者的公钥对第二签名进行验证，通过后表明分布式身份持有者对可验证表述的签名有效，且可验证表述各部分均未被篡改，因此分布式身份验证者可信任该可验证表述，并且上述全部验证通过后，分布式身份验证者信任整个可验证表述，并由此可知，在根据可验证凭证出示可验证表述时，改进后的可验证凭证结构对声明内容进行任意排列组合后仍能验证通过。

[0071] 进一步地，分布式身份持有者在向可验证凭证颁发者提供声明时，会存在与上一次所提供的声明数量增加、数量减少、声明内容修改或声明顺序修改中的一种或多种情况，此时需根据分布式身份持有者所提出的声明需求，重新生成可验证凭证，得到更改版可验证凭证，并验证更改版可验证凭证的可靠性，其中，声明需求包括增加声明、删减声明、声明内容修改或声明顺序修改中的任意一种。

[0072] 此时若声明需求为增加声明，则分布式身份持有者生成改进的可验证凭证，具体如下：

[0073] {metadata

[0074] // VC声明的具体内容

[0075] claim1, r1

[0076] claim2,r2

[0077] ……

[0078] Claimi, ri

[0079] claim(i+1), r(i+1)

[0080] ……

[0081] claimN, rN

[0082] // 对本VC的数字签名

[0083] proof};

[0084] 其中，根据claim(i+1)求得r(i+1)的方法与上述步骤相同，该可验证凭证在验证时的方法也与前述的可验证凭证方法相同，并且，现有技术中，由于可验证凭证结构中原有的Merkle树相关内容，会使可验证凭证的数据结构发生任何改变后均需要重新计算可验证凭证的数字签名及Merkle树相关的内容，增加了可验证凭证颁发者的负荷；而本实施例则可因新增声明条目而需要可验证凭证颁发者重新生成可验证凭证，改进后的可验证凭证结构仅需要可验证凭证颁发者计算该条目对应的变色龙随机数，无需重新计算可验证凭证的数字签名及Merkle树相关的内容，减轻了可验证凭证颁发者的工作量。

[0085] 若声明需求为删减声明，则分布式身份持有者生成改进的可验证凭证，具体如下：

[0086] {metadata

[0087] // VC声明的具体内容

[0088] claim1, r1

[0089] claim2,r2

[0090] ……

[0091] claim(i-1),r(i-1)

[0092] claim(i+1), r(i+1)

[0093] ……

[0094] claimN, rN

[0095] // 对本VC的数字签名

[0096] proof};

[0097] 该可验证凭证在验证时的方法也与前述的可验证凭证方法相同,并且由本步骤可见:可验证凭证结构中原有的Merkle树相关内容使得可验证凭证的数据结构发生任何改变后均需要重新计算可验证凭证的数字签名及Merkle树相关的内容;因删除声明条目而需要分布式身份重新生成可验证凭证时,改进后的可验证凭证结构仅需要分布式身份直接删除该条目,无需重新计算可验证凭证的数字签名及Merkle树相关的内容,减轻了分布式身份的负荷。

[0098] 若声明需求为声明内容修改,即分布式身份持有者向分布式身份颁布者提出在可验证凭证中修改一个已有的声明:将claimi改为claimi',则分布式身份持有者生成改进的可验证凭证,具体如下:

[0099] {metadata

[0100] // VC声明的具体内容

[0101] claim1, r1

[0102] claim2,r2

[0103] ……

[0104] Claimi', ri'

[0105] ……

[0106] claimN, rN

[0107] // 对本VC的数字签名

[0108] proof};

[0109] 其中,根据claimi'求得ri'的方法与上述步骤相同,该可验证凭证在验证时的方法也与前述的可验证凭证方法相同,并且由本步骤可见:可验证凭证结构中原有的Merkle树相关内容使得可验证凭证的数据结构发生任何改变后均需要重新计算可验证凭证的数字签名及Merkle树相关的内容;因修改声明条目而需要分布式身份重新生成可验证凭证时,改进后的可验证凭证结构仅需要分布式身份计算该条目对应的变色龙随机数,无需重新计算可验证凭证的数字签名及Merkle树相关的内容,减轻了分布式身份的负荷。

[0110] 若声明需求为声明顺序修改,即分布式身份持有者向分布式身份颁布者提出在可验证凭证中交换两个已有的声明的顺序:将claimi与claim(i+1),则分布式身份持有者生成改进的可验证凭证,具体如下:

[0111] {metadata

[0112] // VC声明的具体内容

[0113] claim1, r1

[0114] claim2, r2

[0115] ……

[0116] claim(i+1)，r(i+1)

[0117] claimi,ri

[0118] ……

[0119] claimN，rN

[0120] // 对本VC的数字签名

[0121] proof}；

[0122] 该可验证凭证在验证时的方法也与前述的可验证凭证方法相同，并且由本步骤可见：可验证凭证结构中原有的Merkle树相关内容使得可验证凭证的数据结构发生任何改变后均需要重新计算可验证凭证的数字签名及Merkle树相关的内容；因交换声明条目的顺序而需要分布式身份重新生成可验证凭证时，改进后的可验证凭证结构仅需要分布式身份直接交换指定条目，无需重新计算可验证凭证的数字签名及Merkle树相关的内容，减轻了分布式身份的负荷。

[0123] 实施例二

[0124] 一种变色龙哈希分布式身份使用系统，包括第一生成检验单元、第二生成检验单元和更新检验单元；第一生成检验单元用于使用变色龙哈希算法对分布式身份生成可验证凭证，并对可验证凭证进行可靠性验证；其中，第一生成检验单元包括第一生成单元，且第一生成单元包括获取单元、第一计算单元和第一签名单元；获取单元用于获取分布式身份持有者所提供的若干个声明；第一计算单元用于计算分布式身份的普通哈希值和变色龙随机数，并根据普通哈希值和变色龙随机数计算第一变色龙哈希值，并根据各个所述声明计算各个所述声明对应的变色龙随机数；第一签名单元用于使用可验证凭证颁发者私钥对第一变色龙哈希值进行签名得到第一签名。

[0125] 更进一步地，分布式身份持有者向可验证凭证生成者提交若干个声明，在本实施例中可将若干个声明记为claim1~claimN，而一个声明为一个信息条目，例如“姓名:XX”、“年龄:XX”、“住址:XX省XX市XX区XX路XX号”等均可以包含在声明中的信息条目中，可验证凭证声明的具体内容以及对应的证明可以记为{ claim1，r1;claim2，r2;……;claimi，ri ;……;claimN，rN }，设定1≤i≤N，则ri的计算方式具体如下：

[0126] 首先将分布式身份作为变色龙消息，计算普通哈希值MD=H(DID)，选择变色龙随机数为RD=H(metadata)，进行变色龙哈希为第一变色龙哈希值CHD=CH(MD,RD,PKS)，并使用可验证凭证颁发者私钥SKS对第一变色龙哈希值CHD进行签名得到第一签名SIGD1，其中可验证凭证颁发者的公钥标记为PKS，变色龙哈希使用的公钥为PKS，使得私钥作为变色龙哈希的陷门密钥，然后计算新的变色龙消息claimi的普通哈希MCi=H(claimi)，可验证凭证颁发者利用SKS、MD、RD、MCi求解变色龙随机数，得到MCi对应的变色龙随机数为ri。根据变色龙哈希的特性，可验证凭证颁发者以外的任何成员均无法计算得到ri。

[0127] 由此可见，可验证凭证结构中原有的Merkle树相关内容使得可验证凭证的数据结构显得较为复杂，改进后的可验证凭证结构去掉了Merkle树相关的部分，增加了变色龙随机数，可验证凭证的数据结构也变得更为简单。

[0128] 可验证凭证生成后，由可验证凭证颁发者发送给分布式身份持有者，并对可验证凭证进行可靠性验证，因此第一生成检验单元还包括第一验证单元，且第一验证单元包括

第一检验单元、第二计算单元和第一判定单元;第一检验单元用于验证第一签名的有效性;第二计算单元用于根据各个声明,计算各个声明对应的第二变色龙哈希值;第一判定单元用于判断第二变色龙哈希值是否与第一变色龙哈希值相等,且第一签名有效,若是,则可验证凭证可靠,反之,则可验证凭证不可靠。

[0129] 进一步地,首先使用可验证凭证颁发者的公钥对第一签名进行验证,如若通过则表示可验证凭证颁发者对第一变色龙哈希值的签名有效,且分布式身份、元数据(metadata)均未被篡改,因为如果两者被篡改,将导致MD或变色龙随机数RD发生变化,MD或变色龙随机数RD的变化又将导致第一变色龙哈希值CHD的变化,第一变色龙哈希值CHD的变化将最终导致第一签名无法验证通过,即第一签名无效;反之,则可以得知分布式身份、metadata均未被篡改。

[0130] 另一方面,还需要验证各个声明的变色龙哈希值,即第二变色龙哈希值CH(MCi,ri,PKS),具体地,验证CH(MCi,ri,PKS)是否等于CHD,如相等,则说明MCi均未被可验证凭证颁发者以外的成员修改,而可验证凭证颁发者被信任为不会随意对MCi进行修改,因此分布式身份持有者信任MCi,因为,如果MCi被可验证凭证颁发者以外的成员篡改,由于可验证凭证颁发者以外的成员无法得到对应的ri,将导致第二变色龙哈希值CH(MCi,ri,PKS)发生变化,第二变色龙哈希值CH(MCi,ri,PKS)的变化将导致其无法等于第一变色龙哈希值CHD;反之,可以得知MCi未被可验证凭证颁发者以外的成员篡改。

[0131] 待第一签名以及各个声明的变色龙哈希值全部验证步骤通过后,则分布式身份持有者信任整个可验证凭证,该可验证凭证可靠,同时,改进后的可验证凭证结构可以保证所有内容不可篡改。

[0132] 验证完成可验证凭证后,第二生成检验单元根据可验证凭证生成可验证表述,并验证可验证表述的可靠性;其中,第二生成检验单元包括第二生成单元,且第二生成单元包括数据获取单元和第二签名单元;数据获取单元用于获取可验证表述的非签名数据,其中非签名数据包括可验证表述元数据和一个或多个的可验证凭证;第二签名单元用于使用分布式身份持有者私钥对非签名数据进行签名得到第二签名。

[0133] 对于分布式身份持有者而言,为保证身份信息的安全性,通常需要将一部分敏感信息进行隐藏,此时分布式持有者需要生成可验证表述,在本实施例中设定验证凭证声明的具体内容以及对应的证明可以记为{ claim1, r1;claim2, r2;……;claimi, ri },claimi以后的声明为敏感信息,此时需要使用分布式身份持有者私钥对非签名数据进行签名得到第二签名,从而使得根据可验证凭证出示可验证表述时,实现了敏感信息的隐藏,即,改进后的可验证凭证可以对声明内容进行任意取舍和任意排列组合,满足分布式身份持有者对隐私保护的需求。

[0134] 同时,可验证表述也需要验证其可靠性,因此第二生成检验单元还包括第二验证单元,且第二验证单元包括第二检验单元、第三计算单元和第二判定单元;第二检验单元用于验证第一签名和第二签名的有效性;第三计算单元用于根据各个声明,计算各个声明对应的第二变色龙哈希值;第二判定单元用于判断第二变色龙哈希值是否与第一变色龙哈希值相等,且第一签名与第二签名均有效,若是,则可验证表述可靠,反之,则可验证表述不可靠,需要说明的是,验证第二签名的有效性与验证第一签名的有效性的方法相同,在此不做重复赘述,除此之外,由于是对每条声明进行验证,因此即使分布式身份对可验证表述中的

声明顺序进行任意排列组合，均能通过验证。

[0135] 另一方面，在验证第二签名是否有效时，使用分布式身份持有者的公钥对第二签名进行验证，通过后表明分布式身份持有者对可验证表述的签名有效，且可验证表述各部分均未被篡改，因此分布式身份验证者可信任该可验证表述，并且上述全部验证通过后，分布式身份验证者信任整个可验证表述，并由此可知，在根据可验证凭证出示可验证表述时，改进后的可验证凭证结构对声明内容进行任意排列组合后仍能验证通过。

[0136] 进一步地，分布式身份持有者在向可验证凭证颁发者提供声明时，会存在与上一次所提供的声明数量增加、数量减少、声明内容修改或声明顺序修改中的一种或多种情况，此时需更新检验单元根据分布式身份持有者所提出的声明需求，重新生成可验证凭证，得到更改版可验证凭证，并验证更改版可验证凭证的可靠性，其中，声明需求包括增加声明、删减声明、声明内容修改或声明顺序修改中的任意一种。

[0137] 此时若声明需求为增加声明，则分布式身份持有者生成改进的可验证凭证，具体如下：

[0138] {metadata

[0139] // VC声明的具体内容

[0140] claim1, r1

[0141] claim2,r2

[0142] ……

[0143] Claimi, ri

[0144] claim(i+1), r(i+1)

[0145] ……

[0146] claimN, rN

[0147] // 对本VC的数字签名

[0148] proof};

[0149] 其中，根据claim(i+1)求得r(i+1)的方法与上述步骤相同，该可验证凭证在验证时的方法也与前述的可验证凭证方法相同，并且，现有技术中，由于可验证凭证结构中原有的Merkle树相关内容，会使可验证凭证的数据结构发生任何改变后均需要重新计算可验证凭证的数字签名及Merkle树相关的内容，增加了可验证凭证颁发者的负荷；而本实施例则可因新增声明条目而需要可验证凭证颁发者重新生成可验证凭证，改进后的可验证凭证结构仅需要可验证凭证颁发者计算该条目对应的变色龙随机数，无需重新计算可验证凭证的数字签名及Merkle树相关的内容，减轻了可验证凭证颁发者的工作量。

[0150] 若声明需求为删减声明，则分布式身份持有者生成改进的可验证凭证，具体如下：

[0151] {metadata

[0152] // VC声明的具体内容

[0153] claim1, r1

[0154] claim2,r2

[0155] ……

[0156] claim(i-1),r(i-1)

[0157] claim(i+1), r(i+1)

[0158] ……

[0159] claimN, rN

[0160] // 对本VC的数字签名

[0161] proof};

[0162] 该可验证凭证在验证时的方法也与前述的可验证凭证方法相同,并且由本步骤可见:可验证凭证结构中原有的Merkle树相关内容使得可验证凭证的数据结构发生任何改变后均需要重新计算可验证凭证的数字签名及Merkle树相关的内容;因删除声明条目而需要分布式身份重新生成可验证凭证时,改进后的可验证凭证结构仅需要分布式身份直接删除该条目,无需重新计算可验证凭证的数字签名及Merkle树相关的内容,减轻了分布式身份的负荷。

[0163] 若声明需求为声明内容修改,即分布式身份持有者向分布式身份颁布者提出在可验证凭证中修改一个已有的声明:将claimi改为claimi',则分布式身份持有者生成改进的可验证凭证,具体如下:

[0164] {metadata

[0165] // VC声明的具体内容

[0166] claim1, r1

[0167] claim2,r2

[0168] ……

[0169] Claimi', ri'

[0170] ……

[0171] claimN, rN

[0172] // 对本VC的数字签名

[0173] proof};

[0174] 其中,根据claimi'求得ri'的方法与上述步骤相同,该可验证凭证在验证时的方法也与前述的可验证凭证方法相同,并且由本步骤可见:可验证凭证结构中原有的Merkle树相关内容使得可验证凭证的数据结构发生任何改变后均需要重新计算可验证凭证的数字签名及Merkle树相关的内容;因修改声明条目而需要分布式身份重新生成可验证凭证时,改进后的可验证凭证结构仅需要分布式身份计算该条目对应的变色龙随机数,无需重新计算可验证凭证的数字签名及Merkle树相关的内容,减轻了分布式身份的负荷。

[0175] 若声明需求为声明顺序修改,即分布式身份持有者向分布式身份颁布者提出在可验证凭证中交换两个已有的声明的顺序:将claimi与claim(i+1),则分布式身份持有者生成改进的可验证凭证,具体如下:

[0176] {metadata

[0177] // VC声明的具体内容

[0178] claim1, r1

[0179] claim2, r2

[0180] ……

[0181] claim(i+1), r(i+1)

[0182] claimi,ri

[0183] ……

[0184] claimN, rN

[0185] // 对本VC的数字签名

[0186] proof}；

[0187] 该可验证凭证在验证时的方法也与前述的可验证凭证方法相同，并且由本步骤可见：可验证凭证结构中原有的Merkle树相关内容使得可验证凭证的数据结构发生任何改变后均需要重新计算可验证凭证的数字签名及Merkle树相关的内容；因交换声明条目的顺序而需要分布式身份重新生成可验证凭证时，改进后的可验证凭证结构仅需要分布式身份直接交换指定条目，无需重新计算可验证凭证的数字签名及Merkle树相关的内容，减轻了分布式身份的负荷。

[0188] 描述于本申请实施例中所涉及到的单元或模块可以通过软件的方式实现，也可以通过硬件的方式来实现。所描述的单元或模块也可以设置在处理器中，例如，各所述单元可以是设置在计算机或移动智能设备中的软件程序，也可以是单独配置的硬件装置。其中，这些单元或模块的名称在某种情况下并不构成对该单元或模块本身的限定。

[0189] 以上描述仅为本申请的较佳实施例以及对所运用技术原理的说明。本领域技术人员应当理解，本申请中所涉及的发明范围，并不限于上述技术特征的特定组合而成的技术方案，同时也应涵盖在不脱离本申请构思的情况下，由上述技术特征或其等同特征进行任意组合而形成的其它技术方案。例如上述特征与本申请中公开的(但不限于)具有类似功能的技术特征进行互相替换而形成的技术方案。


图1