(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 110599147 A
(43)申请公布日 2019.12.20

(21)申请号 201910873379.X

(22)申请日 2019.09.17

(71)申请人 福州大学
地址 350002 福建省福州市鼓楼区工业路523号

(72)发明人 杨旸 林鸿瑞 郭文忠 刘西蒙 郑相涵 邹剑

(74)专利代理机构 福州元创专利商标代理有限公司 35100
代理人 钱莉 蔡学俊

(51)Int.Cl.
G06Q 20/12(2012.01)
G06Q 20/36(2012.01)
G06Q 20/38(2012.01)
G06Q 20/40(2012.01)
G06F 16/9535(2019.01)
G06F 21/30(2013.01)
G06F 21/60(2013.01)

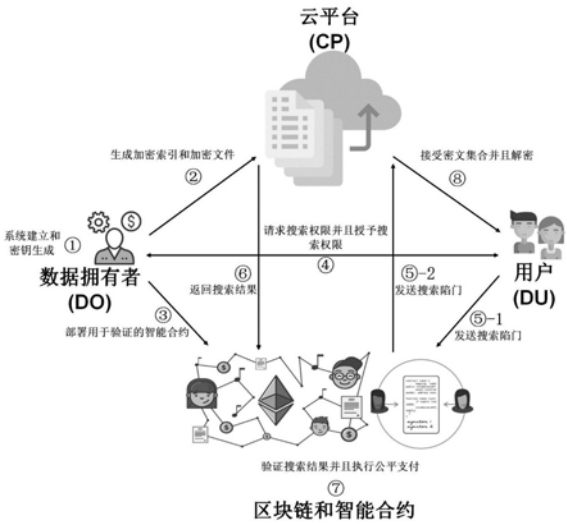
权利要求书2页 说明书14页 附图4页

(54)发明名称

一种基于区块链的密文检索公平支付方法及系统

(57)摘要

本发明涉及一种基于区块链的密文检索公平支付方法及系统，包括数据拥有者、数据用户、云平台、以及部署在区块链上的智能合约；数据拥有者的加密数据通过智能合约授权给一个以上的数据用户进行检索和解密；数据用户当满足授权条件并且在智能合约中存储有足够的搜索费用时能够发起搜索请求；所述智能合约验证云服务器返回的搜索结果的正确性和完整性，验证通过后，所述云服务器将相关度最高的k个搜索结果返回给数据用户。本发明能够进一步解决公平支付的问题。



CN 110599147 A

1.一种基于区块链的密文检索公平支付系统，其特征在于，包括数据拥有者、数据用户、云平台、以及部署在区块链上的智能合约；

数据拥有者的加密数据通过智能合约授权给一个以上的数据用户进行检索和解密；数据用户当满足授权条件并且在智能合约中存储有足够的搜索费用时能够发起搜索请求；所述智能合约验证云服务器返回的搜索结果的正确性和完整性，验证通过后，所述云服务器将相关度最高的k个搜索结果返回给数据用户。

2.根据权利要求1所述的一种基于区块链的密文检索公平支付系统，其特征在于，所述数据拥有者拥有一组要外包给云平台的文件，数据拥有者从文件中提取关键词集合并将其加密成加密索引，同时加密这些文件并将密文和加密索引发送到云平台进行远程存储；数据拥有者能够授权给某个数据用户查询的权利并且赚取用户的查询费用；

所述数据用户在发起搜索请求之前，需要先得到数据拥有者的授权；数据用户通过智能合约将生成的搜索陷门提交给云平台，如果云平台返回的搜索结果通过智能合约的验证，则数据用户向云平台支付服务费，向数据拥有者支付消息费，否则，数据用户不支付任何费用；

所述云平台利用云存储服务来存储数据拥有者的加密索引和加密文件，并向数据用户提供在线搜索服务；所述云平台使用加密索引执行搜索操作，并将正确且完整的前k个最相关的搜索结果返回给数据用户，以赚取服务费；

所述区块链利用智能合约来记录验证数据，从而智能合约能够验证云平台返回的搜索结果的正确性和完整性；数据拥有者和数据用户在区块链上部署一种以上的智能合约来执行包括用户管理、公平支付和搜索在内的功能。

3.根据权利要求1所述的一种基于区块链的密文检索公平支付系统，其特征在于，所述智能合约包括用户管理合约、公平支付合约以及用户接口合约；所述用户管理合约以及公平支付合约由数据拥有者部署到以太坊；智能合约的交互包括以下步骤：

数据用户将价值为fee的以太币存入公平支付合约的押金池中；

数据用户向公平支付合约发出搜索陷门，并附上自己的用户接口合约地址；

公平支付合约调用用户管理合约，检查数据用户是否是授权用户且数据用户在押金池中是否有足够的以太币发起一次搜索操作；如果当前数据用户是授权用户并在押金池中有足够的以太币发起一次搜索操作，则公平支付合约广播搜索陷门，然后云平台接收搜索陷门后执行搜索操作后返回搜索结果；

公平支付合约通过事先存储的验证密钥来验证云平台的搜索结果；

如果公平支付合约中的验证函数输出为true，则分别将信息费和服务费从押金池中转账到数据拥有者和云平台，并调用用户接口合约接收搜索结果；否则，押金池中的搜索费用被退还给数据用户。

4.根据权利要求2所述的一种基于区块链的密文检索公平支付系统，其特征在于，所述数据拥有者从文件中提取关键词集合并将其加密成加密索引，同时加密这些文件并将密文和加密索引发送到云平台进行远程存储具体为：

数据拥有者从明文文档集合$\mathcal{D}$中的每个文档中抽取一个以上的关键词形成总的关键词字典$\mathcal{W}=(w_1,\cdots,w_r)$；采用倒排索引的数据结构实现多关键词排序搜索；将包含搜索关键词

集合W的文件的标识符集合表示为$\mathcal{F}(W)=(F_{\theta_1},F_{\theta_2},\cdots)$，$\mathcal{F}(W)$中的文档标识符按照域加权评分排序；

数据拥有者使用密钥为ek的对称加密算法SEnc将明文文档集合$\mathcal{D}$加密成密文文档集合$\mathcal{C}$；数据拥有者将加密索引设为$\mathcal{EI}=(\mathcal{T},\{\mathcal{EF}(W)\}_{W\subseteq\mathcal{W}})$，最后将$(\mathcal{EI},\mathcal{C})$外包给云平台储存；其中，$\mathcal{EF}(W)$为加密的$\mathcal{F}(W)$，$\mathcal{T}$为查找表，其结构为<key,value>，其中，key域存储伪随机函数的输出，value包含元组<value,proof>，其中，value域储存加密的文件标识符集合的地址，proof域储存多关键词排序搜索结果的验证数据。

5.根据权利要求1所述的一种基于区块链的密文检索公平支付系统，其特征在于，所述数据拥有者通过在智能合约中将当前数据用户标记为非法用户，使该数据用户失去数据拥有者赋予的搜索权限。

6.一种基于权利要求1-5任一项所述的区块链的密文检索公平支付系统的方法，其特征在于，提供数据拥有者、数据用户、云平台，包括以下步骤：

数据拥有者生成系统参数和密钥；

数据拥有者从明文文档中提取关键词集合，并生成相应的加密的关键词索引；数据拥有者使用对称加密算法加密文件，然后将加密索引和密文文档外包给云平台；

数据拥有者在区块链上部署智能合约进行用户管理和公平支付，并将验证操作需要的数据记录在智能合约中以实现公开验证和公平支付；

数据用户请求搜索权限，数据拥有者使用智能合约中的用户管理合约将搜索权限授予数据用户，之后，数据拥有者将搜索密钥授予数据用户；

数据用户为搜索相关功能部署智能合约，数据用户使用搜索密钥生成多关键词搜索陷门，并将其发送到区块链并触发智能合约中的公平支付合约；在数据用户发起搜索请求之前，数据用户需要在智能合约中存入足够的搜索费；如果数据用户是一个授权的用户，并且支付了足够的搜索费用，智能合约将自动在区块链中广播搜索陷门，云平台将接收到该搜索陷门；

云平台根据监听到的搜索陷门执行搜索操作，并将相关度排名前k的文档标识符返回给智能合约进行验证；

根据数据拥有者提供的验证数据，公平支付合约验证云平台返回的搜索结果的正确性和完整；如果搜索结果是正确的完整的，公平支付合约自动使用数据用户预先支付的搜索费给云平台支付信息费，给数据拥有者支付服务费；

验证通过后，云平台将密文文档发送给数据用户；从云平台接收密文文件后，数据用户对密文文件进行解密。

一种基于区块链的密文检索公平支付方法及系统

技术领域

[0001] 本发明涉及可搜索加密与公平支付技术领域,特别是一种基于区块链的密文检索公平支付方法及系统。

背景技术

[0002] 随着云计算的发展,越来越多的企业和个人利用这一新兴技术,将大量数据和计算任务迁移到云平台上,以节省本地存储和计算资源。在云平台为用户提供远程存储和计算服务的同时,用户数据的隐私问题也逐渐显露出来。云平台可以在没有限制的情况下随时访问和使用用户数据。为了在保证云平台中数据可用性的同时保证安全性,可搜索加密技术成为了云计算的研究热点,该技术可以同时达到数据保密和信息检索的目的。然而,目前大部分的可搜索加密方案只支持单关键词搜索,而且云平台对返回的搜索结果没有进行排序。一个实用的可搜索加密方案应该允许用户搜索包含多个关键词的文档,并返回最相关的文件,以节省网络带宽。

[0003] 然而,目前的可搜索加密方案也面临着新的攻击范式,云服务器可能不诚实地执行搜索操作(以节省计算资源),并向用户发送不正确或不完整的搜索结果。在先支付搜索费后使用搜索服务的业务模型中,即使出现了上述场景,用户也必须向云平台支付服务费。如果业务模式改为先使用服务后支付搜索服务费,不诚实或恶意的用户即使收到正确且完整的搜索结果,也可能对云平台进行诋毁,拒绝支付服务费。为了解决上述问题,目前的可搜索加密方案需要一个权威机构的参与来解决支付问题。但依靠可信第三方的支付方式存在着一定的局限性:需要引入完全信任的一方(如银行)公平地处理支付问题;可信第三方可能没有能力验证搜索结果或其他外包计算操作的正确性;数据拥有者、用户的隐私可能会被可信第三方泄漏。因此,一个实用的可搜索加密方案应确保数据拥有者、用户和云平台之间的公平支付。近年来,针对加密数据设计的可验证可搜索加密方案受到了广泛的研究兴趣,这些方案可以验证搜索结果的正确性和完整性。虽然很多验证技术(如同态MAC或RSA累加器)都可以检测到云平台的不诚实行为,但如果没有可信的第三方,则无法正常工作。为了解决这个问题,Hu等人提出了一种基于区块链的可搜索加密方案:该方案将搜索索引存储在智能合约中,搜索算法由智能合约而不是云平台执行。Chen等人、Wang等人和Wu等人也采用了类似的方法:智能合约的搜索操作始终是可信的,可以返回正确的结果,因此不需要对结果进行验证。为了在区块链中存储大容量的索引,这些方案必须将复杂的可搜索索引划分为数千个块,并存储在数千个区块链交易事务中(由于每个事务的存储容量较低)。而且这些交易必须一个接一个地(而不是以并发的方式)上传到区块链,这将花费大量的时间。这三个方案利用以太坊智能合约执行整个搜索算法,由于智能合约执行成本高,导致了大量时间和以太币开销。因此这些方案可扩展性低、成本高。为了实现可搜索加密的公平支付,Zhang等人利用基于比特币的定时承诺协议设计了一个公平的支付系统,该系统使用比特币的输入和输出脚本来验证搜索结果的完整性和正确性,但是该方案的运行会消耗相当数量的比特币,而比特币的价格过高,且比特币的智能合约并不完整,功能也过于有限。Cai

等人使用以太坊的智能合约设计了一个定时付款协议，以便在可搜索加密方案中公平地实现先使用搜索服务后付款的业务流程。除非用户申请仲裁请求，否则Cai的方案不会执行验证算法。当用户对云平台返回的结果不满意时，用户可以提起仲裁请求，每个仲裁节点接收到仲裁请求后独立执行判断过程，由仲裁节点重新实现关键词搜索算法，以验证搜索结果是否正确。最后，这些单独的仲裁结果汇总到一个仲裁智能合约中。最后，仲裁合约根据所有的仲裁结果做出最终决定，即，云平台是否作弊。可以看出，Cai的方案在仲裁过程中浪费了大量的计算资源。

[0004] 区块链技术的出现引入了一种新的去中心化的支付模式来解决这些问题，它不受任何中央机构的控制。区块链中的智能合约是一种自动执行的合同，其条款(买卖双方之间的约定)被直接写入计算机的代码行。智能合约允许匿名方之间进行可信的交易和协议，而无需中央权威机构、法律体系的参与。因此，区块链和智能合约适合在可搜索加密系统中执行验证操作，以实现云平台，用户和数据拥有者之间公平支付。

[0005] 目前的基于区块链的可搜索加密方案使用区块链内置的付款功能都实现了公平支付，但是，这些方案都不支持多关键词搜索、top-k排序和公开可验证的功能，因此，这些方案不具备实用性。

发明内容

[0006] 有鉴于此，本发明的目的是提出一种基于区块链的密文检索公平支付方法及系统，能够进一步解决公平支付的问题。

[0007] 本发明采用以下方案实现：一种基于区块链的密文检索公平支付系统，包括数据拥有者、数据用户、云平台、以及部署在区块链上的智能合约；

[0008] 数据拥有者的加密数据通过智能合约授权给一个以上的数据用户进行检索和解密；数据用户当满足授权条件并且在智能合约中存储有足够的搜索费用时能够发起搜索请求；所述智能合约验证云服务器返回的搜索结果的正确性和完整性，验证通过后，所述云服务器将相关度最高的k个搜索结果返回给数据用户。

[0009] 进一步地，所述数据拥有者拥有一组要外包给云平台的文件，数据拥有者从文件中提取关键词集合并将其加密成加密索引，同时加密这些文件并将密文和加密索引发送到云平台进行远程存储；数据拥有者能够授权给某个数据用户查询的权利并且赚取用户的查询费用；

[0010] 所述数据用户在发起搜索请求之前，需要先得到数据拥有者的授权；数据用户通过智能合约将生成的搜索陷门提交给云平台，如果云平台返回的搜索结果通过智能合约的验证，则数据用户向云平台支付服务费，向数据拥有者支付消息费，否则，数据用户不支付任何费用；

[0011] 所述云平台利用云存储服务来存储数据拥有者的加密索引和加密文件，并向数据用户提供在线搜索服务；所述云平台使用加密索引执行搜索操作，并将正确且完整的前k个最相关的搜索结果返回给数据用户，以赚取服务费；

[0012] 所述区块链利用智能合约来记录验证数据，从而智能合约能够验证云平台返回的搜索结果的正确性和完整性；数据拥有者和数据用户在区块链上部署一种以上的智能合约来执行包括用户管理、公平支付和搜索在内的功能。

[0013] 进一步地,所述智能合约包括用户管理合约、公平支付合约以及用户接口合约;所述用户管理合约以及公平支付合约由数据拥有者部署到以太坊;智能合约的交互包括以下步骤:

[0014] 数据用户将价值为fee的以太币存入公平支付合约的押金池中;

[0015] 数据用户向公平支付合约发出搜索陷门,并附上自己的用户接口合约地址;

[0016] 公平支付合约调用用户管理合约,检查数据用户是否是授权用户且数据用户在押金池中是否有足够的以太币发起一次搜索操作;如果当前数据用户是授权用户并在押金池中有足够的以太币发起一次搜索操作,则公平支付合约广播搜索陷门,然后云平台接收搜索陷门后执行搜索操作后返回搜索结果;

[0017] 公平支付合约通过事先存储的验证密钥来验证云平台的搜索结果;

[0018] 如果公平支付合约中的验证函数输出为true,则分别将信息费和服务费从押金池中转账到数据拥有者和云平台,并调用用户接口合约接收搜索结果;否则,押金池中的搜索费用被退还给数据用户。

[0019] 进一步地,所述数据拥有者从文件中提取关键词集合并将其加密成加密索引,同时加密这些文件并将密文和加密索引发送到云平台进行远程存储具体为:

[0020] 数据拥有者从明文文档集合 $\mathcal{D}$ 中的每个文档中抽取一个以上的关键词形成总的关键词字典 $\mathcal{W}=(w_1,\cdots,w_z)$;采用倒排索引的数据结构实现多关键词排序搜索;将包含搜索关键词集合W的文件的标识符集合表示为 $\mathcal{F}(W)=(F_{\theta_1},F_{\theta_2},\cdots)$,$\mathcal{F}(W)$ 中的文档标识符按照域加权评分排序;

[0021] 数据拥有者使用密钥为ek的对称加密算法SEnc将明文文档集合 $\mathcal{D}$ 加密成密文文档集合 $\mathcal{C}$;数据拥有者将加密索引设为 $\mathcal{EI}=(\mathcal{T},\{\mathcal{EF}(W)\}_{W\subseteq\mathcal{W}})$,最后将 $(\mathcal{EI},\mathcal{C})$ 外包给云平台储存;其中,$\mathcal{EF}(W)$ 为加密的 $\mathcal{F}(W)$,$\mathcal{T}$ 为查找表,其结构为<key,value>,其中,key域存储伪随机函数的输出,value包含元组<value,proof>,其中,value域储存加密的文件标识符集合的地址,proof域储存多关键词排序搜索结果的验证数据。

[0022] 进一步地,所述数据拥有者通过在智能合约中将当前数据用户标记为非法用户,使该数据用户失去数据拥有者赋予的搜索权限。

[0023] 本发明还提供了一种基于上文所述的区块链的密文检索公平支付系统的方法,提供数据拥有者、数据用户、云平台,包括以下步骤:

[0024] 数据拥有者生成系统参数和密钥;

[0025] 数据拥有者从明文文档中提取关键词集合,并生成相应的加密的关键词索引;数据拥有者使用对称加密算法加密文件,然后将加密索引和密文文档外包给云平台;

[0026] 数据拥有者在区块链上部署智能合约进行用户管理和公平支付,并将验证操作需要的数据记录在智能合约中以实现公开验证和公平支付;

[0027] 数据用户请求搜索权限,数据拥有者使用智能合约中的用户管理合约将搜索权限授予数据用户,之后,数据拥有者将搜索密钥授予数据用户;

[0028] 数据用户为搜索相关功能部署智能合约,数据用户使用搜索密钥生成多关键词搜索陷门,并将其发送到区块链并触发智能合约中的公平支付合约;在数据用户发起搜索请

求之前，数据用户需要在智能合约中存入足够的搜索费；如果数据用户是一个授权的用户，并且支付了足够的搜索费用，智能合约将自动在区块链中广播搜索陷门，云平台将接收到该搜索陷门；

[0029] 云平台根据监听到的搜索陷门执行搜索操作，并将相关度排名前k的文档标识符返回给智能合约进行验证；

[0030] 根据数据拥有者提供的验证数据，公平支付合约验证云平台返回的搜索结果的正确性和完整；如果搜索结果是正确的完整的，公平支付合约自动使用数据用户预先支付的搜索费给云平台支付信息费，给数据拥有者支付服务费；

[0031] 验证通过后，云平台将密文文档发送给数据用户；从云平台接收密文文件后，数据用户对密文文件进行解密。

[0032] 与现有技术相比，本发明有以下有益效果：

[0033] 1、本发明能够实现高效的公平支付检索：本发明设计了一个可验证的多关键检索系统来实现(基于域加权评分)top-k排名搜索，其中只有最相关的个加密文件才会被返回给用户。同时提出采用多关键词倒排索引数据结构，并给出了一个高效的查找表。本发明的搜索效率随着关键词数量而不是文档总数的增加而增加。

[0034] 2、本发明能够实现灵活的系统扩展：本发明中一个数据拥有者对应任意多个用户，不需要在系统建立阶段确定用户的总数量和身份，因此本发明可以随时在系统中添加新的用户。而且，系统中公共参数的数量并不随着用户的数量的增加而线性增长。无论系统支持多少个用户，都不会带来额外的通信和存储开销。在云计算平台中，这个特点对于不断增加的用户数量是非常重要的。

[0035] 3、本发明能够实现高效可验证的搜索：云平台储存用户的文档且执行用户的搜索任务，区块链执行用户的验证操作并自动实现公平支付，此过程中无需任何第三方的参与，用户只需运行轻量级的对称解密算法来完成最终的解密运算。

[0036] 4、本发明能够实现安全的密文检索机制：在不需要可信的密钥生成中心的情况下，数据拥有者完全有权管理数据的搜索权限。当某个用户想检索数据拥有者的数据时，用户需要向数据拥有者申请搜索密钥，同时还需要数据拥有者将该用户的身份添加到智能合约的合法用户列表中。即使用户为了利益出卖自己的搜索密钥给其他用户，其他用户仍然无法执行检索操作，只有同时拥有搜索密钥和智能合约中的合法身份，用户才可发起搜索请求。

[0037] 5、本发明拥有高效的用户召回机制：一旦数据拥有者想撤回某个用户的搜索权限，数据拥有者只要调用用户管理智能合约标记该用户为非法用户，该召回机制具有高效性。

[0038] 6、本发明具有去中心化的优点：为了消除中心化系统中可信第三方为了利益偏袒一方的作弊行为，本发明设计了基于区块链技术的可搜索加密验证算法来解决搜索结果可验证问题。数据拥有者通过上传验证密钥到智能合约中，使得智能合约具备了验证云平台返回的搜索结果的能力，任意一方都不能改变智能合约的验证结果。因此，该检索系统的验证操作和公平支付协议不依赖任何可信第三方，从而实现了完全去中心化的公平支付检索系统。

附图说明

[0039] 图1为本发明实施例的系统原理示意图。

[0040] 图2为本发明实施例的只能合约工作流程示意图。

[0041] 图3为本发明实施例的公平支付合约(FPC)的代码框架。

[0042] 图4为本发明实施例的用户管理合约(UMC)的代码框架。

[0043] 图5为本发明实施例的用户接口合约(UIC)的代码框架。

具体实施方式

[0044] 下面结合附图及实施例对本发明做进一步说明。

[0045] 应该指出,以下详细说明都是示例性的,旨在对本申请提供进一步的说明。除非另有指明,本文使用的所有技术和科学术语具有与本申请所属技术领域的普通技术人员通常理解的相同含义。

[0046] 需要注意的是,这里所使用的术语仅是为了描述具体实施方式,而非意图限制根据本申请的示例性实施方式。如在这里所使用的,除非上下文另外明确指出,否则单数形式也意图包括复数形式,此外,还应当理解的是,当在本说明书中使用术语“包含”和/或“包括”时,其指明存在特征、步骤、操作、器件、组件和/或它们的组合。

[0047] 如图1以及图2所示,本实施例提供了一种基于区块链的密文检索公平支付系统,包括数据拥有者(DO)、数据用户(DU)、云平台(CP)、以及部署在区块链上的智能合约;

[0048] 数据拥有者的加密数据通过智能合约授权给一个以上的数据用户进行检索和解密;数据用户当满足授权条件并且在智能合约中存储有足够的搜索费用时能够发起搜索请求;所述智能合约验证云服务器返回的搜索结果的正确性和完整性,验证通过后,所述云服务器将相关度最高的k个搜索结果返回给数据用户。

[0049] 本实施例的符号变量说明如下表所示。

[0050]

符号	描述
$\mathcal{D}$	明文文档集合 $\mathcal{D}=\{D_1,D_2,\cdots,D_n\}$
$\mathcal{C}$	密文文档集合 $\mathcal{C}=\{C_1,C_2,\cdots,C_n\}$
$\mathcal{F}$	文档标识符集合 $\mathcal{F}=\{F_1,F_2,\cdots,F_n\}$
$\mathcal{W}$	$\mathcal{D}$ 的关键词字典
W	搜索关键词集合
$token$	W 的搜索陷门
$\mathcal{D}(W)$	包含 W 的明文文档集合
$\mathcal{C}(W)$	包含 W 的密文文档集合
$\mathcal{F}(W)$	包含 W 的文档标识符集合
$\mathcal{EF}(W)$	加密的 $\mathcal{F}(W)$
$\lvert\mathcal{F}(W)\rvert$	$\mathcal{F}(W)$ 包含的元素数量
$\mathcal{D}_k(W)$	$\mathcal{D}(W)$ 中相关度最高的前 k 个明文文档
$\mathcal{C}_k(W)$	$\mathcal{C}(W)$ 中相关度最高的前 k 个密文文档
$\mathcal{F}_k(W)$	$\mathcal{F}(W)$ 中相关度最高的前 k 个文档标识符

[0051]

$F_j(W)$	$\mathcal{F}(W)$ 中第 j 个文档标识符
$EF_j(W)$	加密的 $F_j(W)$
$\mathcal{EI}/proof$	加密索引/加密索引的验证数据
$\mathcal{T}$	查找表
$SEnc/SDec$	对称加密算法/解密算法
ek/sk	对称加密/搜索密钥
γ_{κ_1}	密钥为 k_1 的伪随机函数(PRF)
μ_{κ_2}	密钥为 k_2 的消息验证码函数(MAC)
$a \parallel b$	字符串 a 和 b 联接
FPC	公平支付合约
UMC	用户管理合约
UIC	用户接口合约

[0052] 在本实施例中，所述数据拥有者拥有一组要外包给云平台的文件，数据拥有者从文件中提取关键词集合并将其加密成加密索引，同时加密这些文件并将密文和加密索引发送到云平台进行远程存储；数据拥有者能够授权给某个数据用户查询的权利并且赚取用户的查询费用；

[0053] 所述数据用户在发起搜索请求之前，需要先得到数据拥有者的授权；数据用户通

过智能合约将生成的搜索陷门提交给云平台，如果云平台返回的搜索结果通过智能合约的验证，则数据用户向云平台支付服务费，向数据拥有者支付消息费，否则，数据用户不支付任何费用；

[0054] 所述云平台利用云存储服务来存储数据拥有者的加密索引和加密文件，并向数据用户提供在线搜索服务；所述云平台使用加密索引执行搜索操作，并将正确且完整的前k个最相关的搜索结果返回给数据用户，以赚取服务费；

[0055] 所述区块链利用智能合约来记录验证数据，从而智能合约能够验证云平台返回的搜索结果的正确性和完整性；数据拥有者和数据用户在区块链上部署一种以上的智能合约来执行包括用户管理、公平支付和搜索在内的功能。

[0056] 在本实施例中，所述智能合约包括用户管理合约、公平支付合约以及用户接口合约；所述用户管理合约以及公平支付合约由数据拥有者部署到以太坊；智能合约的交互包括以下步骤：

[0057] 数据用户将价值为fee的以太币存入公平支付合约的押金池中；

[0058] 数据用户向公平支付合约发出搜索陷门，并附上自己的用户接口合约地址；

[0059] 公平支付合约调用用户管理合约，检查数据用户是否是授权用户且数据用户在押金池中是否有足够的以太币发起一次搜索操作；如果当前数据用户是授权用户并在押金池中有足够的以太币发起一次搜索操作，则公平支付合约广播搜索陷门，然后云平台接收搜索陷门后执行搜索操作后返回搜索结果；

[0060] 公平支付合约通过事先存储的验证密钥来验证云平台的搜索结果；

[0061] 如果公平支付合约中的验证函数输出为true，则分别将信息费和服务费从押金池中转账到数据拥有者和云平台，并调用用户接口合约接收搜索结果；否则，押金池中的搜索费用被退还给数据用户。

[0062] 在本实施例中，所述数据拥有者从文件中提取关键词集合并将其加密成加密索引，同时加密这些文件并将密文和加密索引发送到云平台进行远程存储具体为：

[0063] 数据拥有者从明文文档集合$\mathcal{D}$中的每个文档中抽取一个以上的关键词形成总的关键词字典$\mathcal{W}=(w_1,\cdots,w_r)$；采用倒排索引的数据结构实现多关键词排序搜索；将包含搜索关键词集合W的文件的标识符集合表示为$\mathcal{F}(W)=(F_{\theta_1},F_{\theta_2},\cdots)$，$\mathcal{F}(W)$中的文档标识符按照域加权评分排序；

[0064] 数据拥有者使用密钥为ek的对称加密算法SEnc将明文文档集合$\mathcal{D}$加密成密文文档集合$\mathcal{C}$；数据拥有者将加密索引设为$\mathcal{EI}=(\mathcal{T},\{\mathcal{EF}(W)\}_{W\subseteq\mathcal{W}})$，最后将$(\mathcal{EI},\mathcal{C})$外包给云平台储存；其中，$\mathcal{EF}(W)$为加密的$\mathcal{F}(W)$，$\mathcal{T}$为查找表，其结构为<key,value>，其中，key域存储伪随机函数的输出，value包含元组<value,proof>，其中，value域储存加密的文件标识符集合的地址，proof域储存多关键词排序搜索结果的验证数据。

[0065] 在本实施例中，所述数据拥有者通过在智能合约中将当前数据用户标记为非法用户，使该数据用户失去数据拥有者赋予的搜索权限。

[0066] 本实施例还提供了一种基于上文所述的区块链的密文检索公平支付系统的方法，提供数据拥有者、数据用户、云平台，包括以下步骤：

[0067] 数据拥有者生成系统参数和密钥；

[0068] 数据拥有者从明文文档中提取关键词集合，并生成相应的加密的关键词索引；数据拥有者使用对称加密算法加密文件，然后将加密索引和密文文档外包给云平台；

[0069] 数据拥有者在区块链上部署智能合约进行用户管理和公平支付，并将验证操作需要的数据记录在智能合约中以实现公开验证和公平支付；

[0070] 数据用户请求搜索权限，数据拥有者使用智能合约中的用户管理合约将搜索权限授予数据用户，之后，数据拥有者将搜索密钥授予数据用户；

[0071] 数据用户为搜索相关功能部署智能合约，数据用户使用搜索密钥生成多关键词搜索陷门，并将其发送到区块链并触发智能合约中的公平支付合约；在数据用户发起搜索请求之前，数据用户需要在智能合约中存入足够的搜索费(包括消息费和服务费)；如果数据用户是一个授权的用户，并且支付了足够的搜索费用，智能合约将自动在区块链中广播搜索陷门，云平台将接收到该搜索陷门；

[0072] 云平台根据监听到的搜索陷门执行搜索操作，并将相关度排名前k的文档标识符返回给智能合约进行验证；

[0073] 根据数据拥有者提供的验证数据，公平支付合约验证云平台返回的搜索结果的正确性和完整；如果搜索结果是正确的完整的，公平支付合约自动使用数据用户预先支付的搜索费给云平台支付信息费，给数据拥有者支付服务费(按预定义的分配比例)；否则，数据用户的搜索费将被退回到他自己的账户；

[0074] 验证通过后，云平台将密文文档发送给数据用户；从云平台接收密文文件后，数据用户对密文文件进行解密。

[0075] 特别的，本实施例针对上述系统与方法对几个关键的步骤进行详细的描述。

[0076] 在系统建立的阶段，即图1中的(1)，输入安全参数λ，DO选取伪随机函数(PRF) $\gamma_{\kappa_1}:\{0,1\}^{\lambda}\times\{0,1\}^{*}\rightarrow\{0,1\}^{d}$ 和消息验证码函数(MAC) $\mu_{\kappa_2}:\{0,1\}^{\lambda}\times\{0,1\}^{*}\rightarrow\{0,1\}^{l}$，其中d是文档标识符的长度，λ是标准MAC函数(例如基于SHA256的HMAC)。DO选择密钥空间为 $\mathcal{EK}$ 的对称加密/解密算法对SEnc/SDec。DO设置公开参数为 $PP=(\gamma_{\kappa_1},\mu_{\kappa_2},SEnc/SDec)$。

[0077] 在密钥生成阶段，当DO想要分享自己拥有的文档时，DO输入安全参数λ运行密钥生成算法KeyGen生成加密密钥，搜索密钥sk和验证密钥vk。如图1所示的工作流程(1)。具体算法为：

[0078] $\mathrm{KeyGen}(1^{\lambda})\rightarrow(\mathrm{ek},\mathrm{sk},\mathrm{vk})$：输入安全参数λ，DO随机选择密钥 $\kappa_1,\kappa_2\in_R\{0,1\}^{\lambda}$ 和对称加密密钥 $ek\in_R\mathcal{EK}$。定义 $\mathrm{sk}=\kappa_1,\mathrm{vk}=\kappa_2$。

[0079] 在加密阶段，DO的加密文件可以被许多用户搜索到。该阶段中，DO从明文文档集合 $\mathcal{D}$ 中抽取关键词字典 $\mathcal{W}$ 并利用sk构建加密索引 $\mathcal{EI}$。DO使用ek将明文文档集合 $\mathcal{D}$ 加密成密文文档集合 $\mathcal{C}$，DO使用vk为加密索引 $\mathcal{EI}$ 产生验证数据proof。上述操作完成之后，DO部署用户管理合约和公平支付合约到区块链。如图1所示的工作流程(2)-(3)。

[0080] DO从明文文档集合 $\mathcal{D}$ 中的每个文档中抽取若干关键词形成总的关键词字典 $\mathcal{W}=(w_1,\cdots,w_\tau)$。首先，本发明采用倒排索引的数据结构以实现多关键词排序搜索。下表为支持三个关键词的倒排索引结构示例。假设三个关键词集合表示为 $\mathcal{W}=(w_i,w_j,w_k)$，关键词按词典顺序排列。如果DU想查询少于三个关键词，则需要对搜索关键词集合W进行扩展：将包含

一个关键词的集合(wi)扩展为(wi,wi,wi);将包含两个关键词的集合(wi,wj)扩展到(wi,wj,wj)。

[0081]

W	$\mathcal{F}(W)$		W	$\mathcal{F}(W)$
(w_1,w_1,w_1)	$(F_1,F_2,F_3,\cdots)$		(w_1,w_2,w_3)	$(F_2,F_6,F_7,\cdots)$
…	…		…	…
(w_t,w_t,w_t)	$(F_1,F_5,F_{10},\cdots)$		(w_1,w_2,w_t)	$(F_{11},F_{14},F_{15},\cdots)$
(w_1,w_2,w_2)	$(F_2,F_3,F_8,\cdots)$		(w_1,w_3,w_4)	$(F_6,F_{10},F_{11},\cdots)$
…	…		…	…
(w_1,w_t,w_t)	$(F_1,F_5,F_8,\cdots)$		(w_1,w_3,w_t)	$(F_1,F_2,F_4,\cdots)$
(w_2,w_3,w_3) …	$(F_9,F_{12},F_{15},\cdots)$ …		(w_2,w_3,w_4) …	$(F_4,F_{22},F_{31},\cdots)$ …
(w_2,w_t,w_t)	$(F_4,F_{16},F_{18},\cdots)$		(w_2,w_3,w_t)	$(F_7,F_{11},F_{14},\cdots)$

[0082]

…	…		…	…
(w_{t-1},w_t,w_t)	$(F_2,F_4,F_{11},\cdots)$		(w_{t-2},w_{t-1},w_t)	$(F_5,F_6,F_8,\cdots)$

[0083] 本发明将包含搜索关键词集合W的文件的标识符集合表示为 $\mathcal{F}(W)=(F_{\theta_1},F_{\theta_2},\cdots)$，$\mathcal{F}(W)$ 中的文档标识符按照域加权评分 $S_{W,F}=\sum_{\alpha=1}^{3}S_{w_\alpha,F}$ 排序。

[0084] 基于上述倒排索引,DO利用搜索密钥sk＝κ_1和验证密钥vk＝κ_2来构建包含验证数据Proof的加密索引。加密索引由查找表 $\mathcal{T}$ 和加密的文件标识符集合 $\mathcal{EF}(W)$ 组成。查找表 $\mathcal{T}$ 的结构可以表示为＜key,value＞。其中key域储存伪随机函数 γ_κ 的输出,value域包含元组＜value,proof＞,其中value域储存加密的文件标识符集合的地址,proof域储存多关键词排序搜索结果的验证数据。

[0085] 详细的构造如下:对倒排索引中的每个关键词集合W,DO计算 $\gamma_{\kappa_1}(W)$ 并设置 $\mathcal{T}\left[\gamma_{\kappa_1}(W)\right].value=address(\mathcal{F}_k(W))\oplus\gamma_{\kappa_1}(W\|0)$，$\mathcal{T}\left[\gamma_{\kappa_1}(W)\right].proof=\mu_{\kappa_2}(\gamma_{\kappa_1}(W)\|F_1(W)\|\cdots\|F_k(W))$，其中 $\mathcal{F}_k(W)=(F_1(W),\cdots,F_k(W))$ 是前k个域加权评分最高的文档标识符集合。记号 $address(\mathcal{F}_k(W))$ 记作集合 $\mathcal{F}_k(W)$ 的地址。如果包含关键词W的文档的数量为β且β＜k，则

$\mathcal{T}\left[\gamma_{\kappa_1}(W)\right].proof=\mu_{\kappa_2}(\gamma_{\kappa_1}(W)\|F_1(W)\|\cdots\|F_\beta(W))$ 且 $\mathcal{F}_k(W)=(F_1(W),\cdots,F_\beta(W))$。集合 $\mathcal{F}_k(W)$ 中的每个元素被加密成 $\mathcal{EF}(W)=(EF_1(W),\cdots,EF_k(W))=(F_1(W)\oplus\gamma_{\kappa_1}(W\|1),\cdots,F_k(W)\oplus\gamma_{\kappa_1}(W\|1))$。DO使用密钥为ek的对称加密算法SEnc将明文文档集合 $\mathcal{D}$ 加密成密文文档集合 $\mathcal{C}$。DO将加密索引设为 $\mathcal{EI}=(\mathcal{T},\{\mathcal{EF}(W)\}_{W\subseteq\mathcal{W}})$，最后将 $(\mathcal{EI},\mathcal{C})$ 外包给云平台储存。

[0086] 接着，DO部署公平支付合约(FPC)到以太坊中并且将验证密钥vk＝κ_2记录到FPC中。FPC是本实施例中核心的组件，它负责检查每个发起搜索请求的DU是否是一个授权用户，FPC记录、广播搜索陷门，验证CP的搜索结果，最终实现公平支付。在FPC被部署之后，DO部署用户管理合约(UMC)来注册授权用户。FPC和UMC的代码结构如图3和图4所示。

[0087] 在陷门生成阶段，DU在区块链上部署了搜索相关的智能合约，并向DO请求搜索权限。如果DO允许，DO将搜索密钥sk授予DU(如图1所示的工作流程4)。DU使用搜索密钥sk将多关键词集合W生成多关键词搜索陷门并将其上传到FPC(如图1所示的工作流程5-1)。FPC检查搜索陷门的有效性，如果陷门合法则将搜索陷门发送给CP进行处理(如图1所示的工作流程5-2)。

[0088] 陷门生成算法由DU执行。当DU第一次向CP请求搜索服务时，他首先向DO请求搜索权限。如果请求被允许，DO将搜索密钥sk授予DU，并在用户管理合约(UMC)中的授权用户集合中添加DU的以太坊地址。DU使用搜索密钥sk生成多关键词搜索陷门token。接着，DU部署用户接口合约(UIC)并且存款一笔以太币到FPC的押金池中(与他自己的账户相关联)。具体来说，DU产生多关键词搜索陷门 $token=(\gamma_{\kappa_1}(W),\gamma_{\kappa_1}(W\|0),\gamma_{\kappa_1}(W\|1))$。DU调用FPC的initRequest()函数上传陷门到FPC。收到搜索陷门之后，FPC调用UMC检查DU是否是一个授权用户。如果DU是一个授权用户，且DU在FPC的押金池中有足够的以太币。则FPC抛出以太坊事件token以通知CP执行搜索操作。UIC用来接收来自FPC的被验证过的搜索结果。UIC的代码结构如图5所示。

[0089] 在搜索阶段，CP利用加密索引 $\mathcal{EI}$ 和搜索陷门token，CP输出前k的最相关的搜索结果集合 $\mathcal{C}$ (如图1所示的工作流程6)。

[0090] CP捕捉到FPC抛出的事件之后，CP将此事件解析成元组 $(\gamma_{\kappa_1}(W),\gamma_{\kappa_1}(W\|0),\gamma_{\kappa_1}(W\|1))$ 并用此元组执行搜索操作。在查找表 $\mathcal{T}$ 中，CP使用 $\gamma_{\kappa_1}(W)$ 搜索 $\mathcal{T}[\gamma_{\kappa_1}(W)].value$ 和 $\mathcal{T}[\gamma_{\kappa_1}(W)].proof$。对于每个 $F_j(W)\in\mathcal{F}_k(W)$，CP通过计算 $EF_j(W)\oplus\gamma_{\kappa_1}(W\|1)$ 恢复出文件标识符$F_j(W)$。接着CP发送 $\mathcal{F}_k(W)$ 和 $\mathcal{T}\left[\gamma_{\kappa_1}(W)\right].proof$ 给FPC进行接下来的验证过程。

[0091] 在验证阶段，智能合约利用保存在智能合约上的验证密钥vk，验证数据Proof，搜索陷门token，搜索结果 $\mathcal{F}_k(W)$，智能合约验证结果的正确性和完整性。如果搜索结果是有效的，公平支付合约FPC输出1并将信息/服务费用转到DO/CP的以太坊地址。否则，合约输出0并将搜索费返回给DU(如图1所示的工作流程7)。$Verify(vk,proof,token,\mathcal{F}_k(W))\rightarrow 1/0$。该过程由公平支付合约(FPC)独立运行。FPC收到集合 $\mathcal{F}_k(W)$ 后，FPC验证标识符集合 $\mathcal{F}_k(W)$ 的正确性和完整性。假设FPC从CP收到的验证数据为Proof，从DU得到的搜索陷门为 $token=(\gamma_{\kappa_1}(W),\gamma_{\kappa_1}(W\|0),\gamma_{\kappa_1}(W\|1))$，FPC重新计算 $Proof'=\mu_{\kappa_2}(\gamma_{\kappa_1}(W)\|F_1(W)\|F_2(W),\cdots,F_k(W))$ 并

验证Proof＝Proof′是否成立。如果上式成立,FPC按照预定义的分配比例从押金池转搜索费用给DO和CP(分别作为信息费和服务费),并将搜索结果发送到UIC智能合约。否则,FPC将搜索费转回DU自己的账户。

[0092] 在解密阶段,本阶段输入密文集合$\mathcal{C}_k(W)$和对称加密密钥ek,DU恢复明文集合$D_k(W)$。如图1所示的工作流程(8)。:DU得到CP返回的搜索结果$\mathcal{C}_k(W)$,用对称密钥ek解密密文文档得到$\mathcal{D}_k(W) \leftarrow SDec(\mathcal{C}_k(W))$。

[0093] 较佳的,本实施例将FPC账户拥有的以太币的总量记为deposit pool。本实施例利用智能合约来验证来自CP的搜索结果,智能合约将保证搜索结果的完整性和正确性。本实施例中的智能合约交互流程如图2所示,包括以下步骤:

[0094] (1)DO与CP协商搜索费用fee和搜索费用的分配比例proportion。然后,DO部署FPC和UMC到以太坊,DU部署UIC到以太坊。

[0095] (2)DU将价值为fee的以太币存入FPC的押金池中。

[0096] (3)DU向FPC发出搜索陷门,并附上自己的UIC地址。

[0097] (4)FPC调用UMC,检查DU是否是授权用户且DU在押金池中是否有足够的以太币发起一次搜索操作。

[0098] (5)如果(4)中的条件都被满足了,FPC广播搜索陷门,然后CP接收陷门,CP执行搜索操作后返回搜索结果。

[0099] (6)FPC通过存储在FPC中的验证密钥来验证CP的搜索结果。

[0100] (7)如果FPC中的验证函数输出为true,则分别将信息费和服务费从押金池deposit pool中转账到DO和CP,并调用UIC接收搜索结果。(DU的搜索费按预定义的分配比例分为服务费和信息费)。

[0101] (8)否则,押金池中的搜索费用将被退还给DU。

[0102] 其中,DO部署用户管理合约(UMC)来管理授权用户列表userList,它将用户以太坊地址映射到布尔值(“1”表示已授权用户地址,“0”表示撤销的用户地址)。DO能够通过调用UMC中的addUser/removeUser函数来添加/删除用户,该函数只能由DO执行。FPC调用verifyUser函数来进行用户身份验证。UMC的代码框架如图4所示。

[0103] 其中,当DO和CP协商好搜索费用(信息费和服务费的总和)和信息费和服务费的分配比例时。DO部署公平支付合约(FPC),FPC验证DU提交的搜索陷门的搜索结果:一旦CP提供错误的搜索结果或没有提供完整的搜索结果,搜索结果将被FPC拒绝,CP将得不到任何费用。一旦CP提供的搜索结果被FPC验证是完整且正确的,FPC会根据分配比例从押金池中转账信息费给DO,转账服务费给CP。因此,CP不能故意返回部分或错误的搜索结果以节省计算资源。相反,如果CP提供正确的搜索结果,押金池中的自动支付将被触发。因此,DU不能中断付款过程,因为DU的押金在CP提供了正确的搜索结果后会自动从他的FPC的押金池中扣除。FPC的代码框架如图3所示。FPC提供了以下三个接口:

[0104] deposit()→balance value:DU调用此函数从他的外部账户中转一定的以太币到FPC的押金池中。当FPC收到DU的押金时,它将更新这个用户的账户余额。

[0105] initRequest(token,address)→Ethereum event:DU调用此函数来请求搜索服务。initRequest函数将通过调用UMC中的verifyUser函数来检查调用方DU的有效性。如果

DU的地址是UMC中授权用户集合userList中的一个元素，并且DU在押金池中有足够的以太币，initRequest函数发出与这个陷门相关的以太坊事件。CP监听FPC发出的事件。CP接收并将事件解析为元组(userAddr,token)，该元组用作搜索函数的输入。搜索操作完成后，CP调用FPC中的verifyResultFromCP函数对结果进行验证并获得服务费。

[0106] verifyResultFromCP(userAddr,identifiers,proof)→Boolean:该函数由CP调用，如果CP的搜索结果被验证是完整且正确的，FPC转账总量为fee×proportion的以太币给DO，转账总量为fee×(1-proportion)的以太币给CP。否则，搜索费fee将退还给DU。最后，该函数调用与userAddr关联的UIC的receiveResults函数来保存搜索结果。在图3中，本发明假设CP和DO平分搜索费用，即，搜索费的分配比例proportion为1:1。

[0107] 较佳的，在以太坊的点对点网络中，服务器可以通过运行JavaScript的web3.js库来监听以太坊发出的事件，这使得跟踪事务变得很容易。如果CP使用事件来返回搜索结果，则可能存在安全风险。每个监听区块链的人都可以在不使用任何身份验证机制的情况下得到一些搜索结果。为了解决这个问题，本实施例引入了DU部署的用户接口合约(UIC)。一旦搜索结果通过完整性和正确性的验证，FPC将调用UIC来记录搜索结果。只有UIC的创造者(数据拥有者)有权调用receiveResults函数接收FPC发送的正确无误的搜索结果;DU还可以调用getSearchResults函数得到存储在UIC上搜索结果。UIC的代码框架如图5所示。

[0108] 特别的，本实施例中涉及到域加权评分，词频是用来评估文档中某个关键词重要性的参数。但是，一个文档有不同的区域(例如标题、摘要和正文)，并且出现在不同区域中的关键词具有不同的重要性。例如，标题中的关键词比摘要中的关键词更重要，与其他区域相比，正文中的关键词的重要性最低。本实施例采用域加权评分计算相关分数。假设有一组文档，每个文档都有t个区域，这些区域的权重分别为$g_1,\cdots,g_t\in[0,1]$，使得$\sum_{i=1}^{t}g_i=1$。对于$1\leqslant i\leqslant t$，设s_i为关键词w与文件F的第i个区域匹配(或不匹配)的布尔值，则域加权得分定义为$S_{w,F}=\sum_{i=1}^{t}g_i s_i$。对于关键词集$W=(w_1,\cdots,w_m)$，域加权评分记为$S_{w,F}=\sum_{j=1}^{m}S_{w_j,F}$。

[0109] 特别的，本实施例中涉及到倒排索引。倒排索引是一种高效的信息检索数据结构，用于加速搜索过程，它存储了从关键词到一组文档(包含关键词)的映射。倒排索引的一个例子如下表所示，其中第一行表示包含关键词w_1的文件的标识符为F_1,F_2,F_3等等。

[0110]

KEYWORD	FILES
W_1	$F_1,F_2,F_3,\cdots$
W_2	$F_2,F_3,F_7,F_9,\cdots$
···	···
W_m	$F_1,F_5,F_{10},\cdots$

[0111] 特别的，智能合约实际上是一种数字化的法律合同，该法律合同由计算机执行的程序表示。智能合约可以在不需要可信的第三方(TTP)的情况下，在参与者之间建立起信任

关系。由于缺乏可编程的数字系统，直到比特币和以太坊平台的出现，智能合约才由概念第一次转为了现实。比特币的脚本语言是智能合约的第一个不完善的版本，它缺乏图灵完备性和高可伸缩性。与比特币相比，以太坊被称为可编程的区块链。以太坊不像比特币那样预先定义一组脚本内容，而是允许用户根据实际需要编写复杂的智能合约。以太坊平台允许外部用户调用合约账户的智能合约来实现特定的功能。外部账户和合约帐户都被一个20字节的十六进制字符串所标识，例如0xca35b7d915458ef540ade6068dfe2f44e8fa733c。以太坊智能合约以字节码的格式存储在以太坊区块链上，并在以太坊虚拟机(EVM)中执行。一个智能合约可能包含多个函数。因此，智能合约调用者需要一个应用程序二进制接口(ABI)来指定要调用合约中的哪个函数以及输出的格式。在以太坊中，用户可以利用私钥控制自己的外部账户，例如将以太币汇款到另一个地址。本实施例使用智能合约作为公平的仲裁者，验证CP提供的搜索结果的完整性和正确性，保证数据拥有者、云平台和用户之间的公平支付。

[0112] 现有的检索系统普通存在在线公平支付问题：如果用户先支付检索费后获取服务，云平台可能为了节省计算资源而不返回正确的搜索结果；如果用户先获取服务后付费，则在云平台返回正确的搜索结果后，用户可能故意不支付服务费，存在极大的作弊隐患。本实施例设计了一个基于区块链的可验证多关键词排序检索系统，该系统利用智能合约来验证搜索结果的正确性和完整性。该发明利用智能合约对搜索结果的自动化验证功能，实现了云平台、数据拥有者和用户之间的自动化公平支付。云服务器根据搜索请求返回相关度最高的个文档。本发明实现了多用户的安全数据共享，数据拥有者的加密数据可以通过智能合约授权给多个用户进行安全检索和解密。本发明可以防止任何用户和云平台在本检索系统中存在的作弊行为，确保使用本检索系统的所有参与者都不会产生经济损失。

[0113] 本领域内的技术人员应明白，本申请的实施例可提供为方法、系统、或计算机程序产品。因此，本申请可采用完全硬件实施例、完全软件实施例、或结合软件和硬件方面的实施例的形式。而且，本申请可采用在一个或多个其中包含有计算机可用程序代码的计算机可用存储介质(包括但不限于磁盘存储器、CD-ROM、光学存储器等)上实施的计算机程序产品的形式。

[0114] 本申请是参照根据本申请实施例的方法、设备(系统)、和计算机程序产品的流程图和/或方框图来描述的。应理解可由计算机程序指令实现流程图和/或方框图中的每一流程和/或方框、以及流程图和/或方框图中的流程和/或方框的结合。可提供这些计算机程序指令到通用计算机、专用计算机、嵌入式处理机或其他可编程数据处理设备的处理器以产生一个机器，使得通过计算机或其他可编程数据处理设备的处理器执行的指令产生用于实现在流程图一个流程或多个流程和/或方框图一个方框或多个方框中指定的功能的装置。

[0115] 这些计算机程序指令也可存储在能引导计算机或其他可编程数据处理设备以特定方式工作的计算机可读存储器中，使得存储在该计算机可读存储器中的指令产生包括指令装置的制造品，该指令装置实现在流程图一个流程或多个流程和/或方框图一个方框或多个方框中指定的功能。

[0116] 这些计算机程序指令也可装载到计算机或其他可编程数据处理设备上，使得在计算机或其他可编程设备上执行一系列操作步骤以产生计算机实现的处理，从而在计算机或其他可编程设备上执行的指令提供用于实现在流程图一个流程或多个流程和/或方框图一

个方框或多个方框中指定的功能的步骤。

[0117] 以上所述，仅是本发明的较佳实施例而已，并非是对本发明作其它形式的限制，任何熟悉本专业的技术人员可能利用上述揭示的技术内容加以变更或改型为等同变化的等效实施例。但是凡是未脱离本发明技术方案内容，依据本发明的技术实质对以上实施例所作的任何简单修改、等同变化与改型，仍属于本发明技术方案的保护范围。

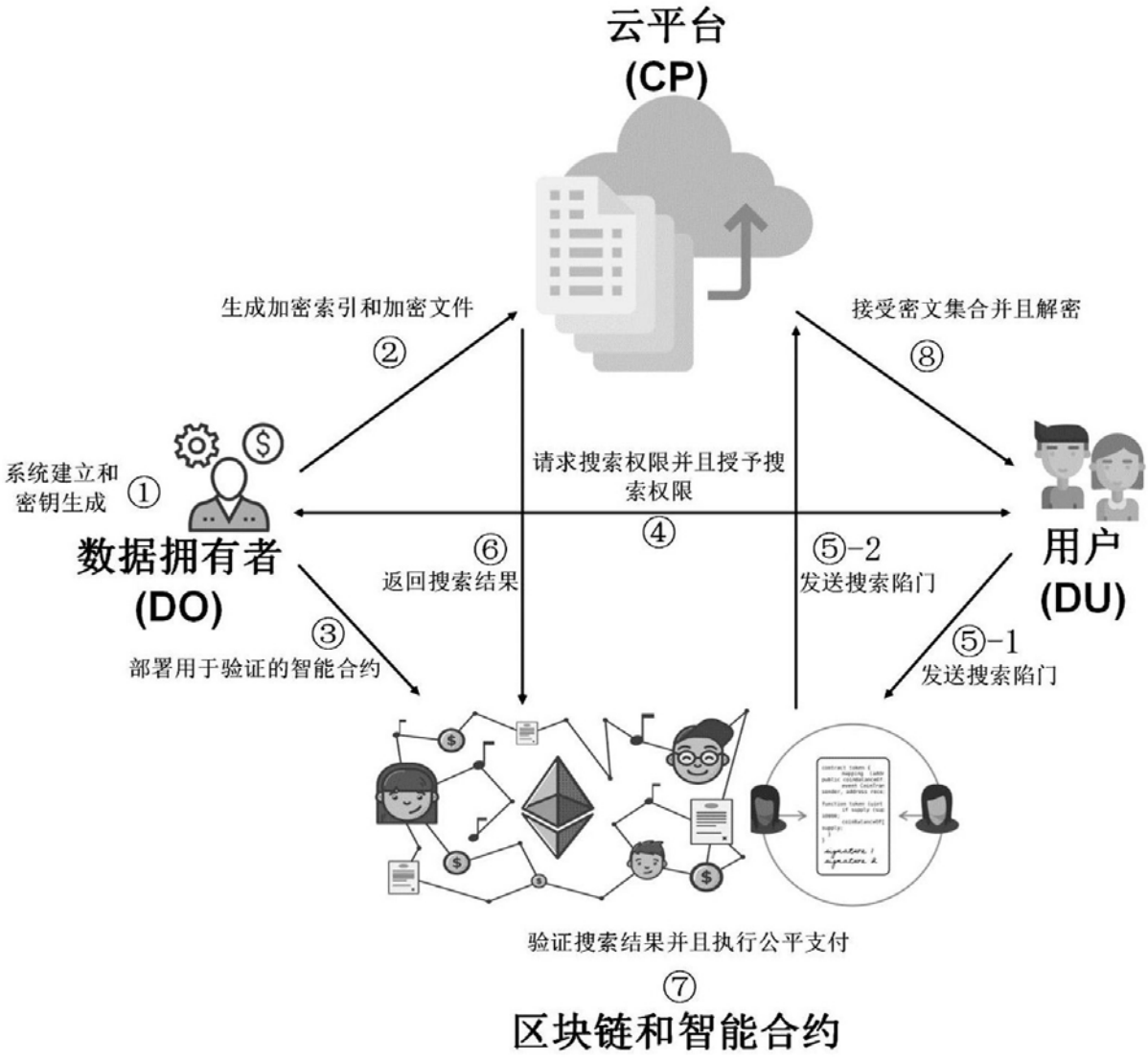

云平台
(CP)
生成加密索引和加密文件
②
接受密文集合并且解密
⑧
系统建立和
密钥生成 ①
请求搜索权限并且授予搜
索权限
④
数据拥有者
(DO)
⑥
返回搜索结果
⑤-2
发送搜索陷门
用户
(DU)
③
部署用于验证的智能合约
⑤-1
发送搜索陷门
验证搜索结果并且执行公平支付
⑦
区块链和智能合约

图1

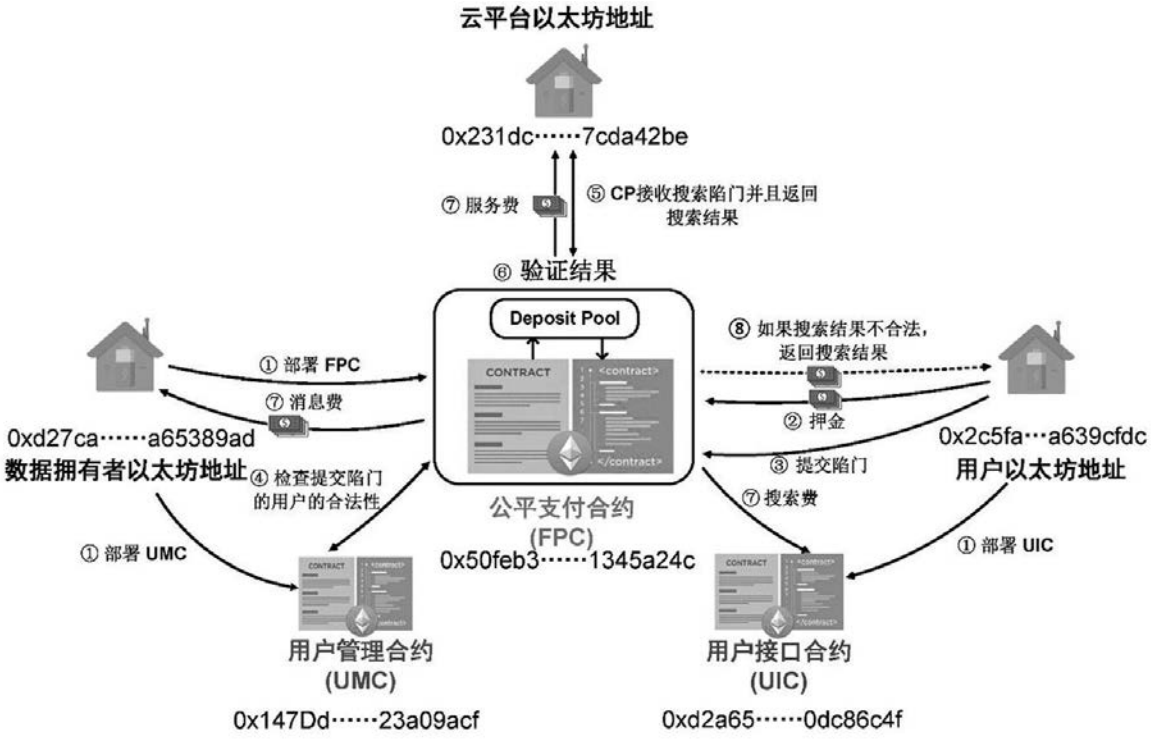

云平台以太坊地址
0x231dc……7cda42be
⑦ 服务费
⑤ CP接收搜索陷门并且返回搜索结果
⑥ 验证结果
Deposit Pool
CONTRACT
<contract>
</contract>
① 部署 FPC
⑦ 消息费
0xd27ca……a65389ad
数据拥有者以太坊地址
④ 检查提交陷门的用户的合法性
① 部署 UMC
公平支付合约
(FPC)
0x50feb3……1345a24c
⑧ 如果搜索结果不合法，返回搜索结果
② 押金
③ 提交陷门
⑦ 搜索费
0x2c5fa…a639cfdc
用户以太坊地址
① 部署 UIC
用户管理合约
(UMC)
0x147Dd……23a09acf
用户接口合约
(UIC)
0xd2a65……0dc86c4f

图2

```
contract fairPayment{
  struct  dataUser{
    bytes32[] token;
    uint256 deposit_pool;
    address UIC;
  }
  address payable DO;
  address payable CP;
  mapping(address=>dataUser) private addrToDU;
  bytes32 private key;
  uint256 constant fee;
  event Token(bytes32 token, uint32 flag, address userAddr);
  constructor(bytes32 verifykey){
    key = verifykey;
  }
  function deposit() public returns(uint256){
    if(msg.value != 0){
      addrToDU[msg.sender].deposit_pool += msg.value;
    }
    return addrToDU[msg.sender].deposit_pool;
  }
  function initRequest(bytes32 token, address UIC){
    //invoke verifyUser function in UMC
    //assert the addrToDU[msg.sender].balance
    //in deposit pool is greater than search fee;
     ...
addrToDU[msg.sender].token[length++] = token;
      addrToDU[msg.sender].UIC = UIC;
      uint32 flag = addrToDU[msg.sender].token.length;
      emit Token(token, flag, msg.sender);
  }
  function verifyResultFromCP(userAddr,flag,ids,proof) public returns(bool){
    assert(msg.sender == CP);
    bytes32 token=addrToDU[userAddr].token[flag];
    //prevent the token from being used
    //more than once (relay attack)
    asser(token!=null);
    Tag=Hmac(key,addrToDU[userAddr].token[flag],ids);
    if(proof == Tag){
      DO.transfer(fee/2);
      CP.transfer(fee/2);
      addrToDU[userAddr].deposit_pool -= fee;
      addrToDU[userAddr].token[flag] = null;
    }else{
      userAddr.transfer(fee);
      addrToDU[userAddr].deposit_pool -= fee;
      addrToDU[userAddr].token[flag] = null;
    }
    //invoke receiveResults function
    //from addrToDU[userAddr].UIC
    ...
  }
}
```

图3

```
Contract userManagement{
  Address DO;
  Address FPC;
  mapping(address=>bool) private userList;
  function addUser(address userAddr){
    if(msg.sender==DO){
    //add the userAddr into the userList.
   ...
    }
  }
  function removeUser(address userAddr){
    if(msg.sender==DO){
    //remove the userAddr from userList.
   ...
    }
  }
```

图4

```
contract userInterface{
    address DU;
    address FPC;
    mapping(bytes32=>string[]) resultHistory;
    function receiveResults(bytes32 token,ids){
        if(msg.sender==FPC){
            //save the identifiers into resultHistory;
        …
        }
    }
    function getSearchResults(bytes32 token) returns(string[]){
        if(msg.sender==DU){
            //return results from resultHistory.
        …
        }
    }
```

图5