(19)国家知识产权局

(12)发明专利申请

(10)申请公布号 CN 114741711 A
(43)申请公布日 2022.07.12

(21)申请号 202210355792.9

(22)申请日 2022.04.06

(71)申请人 石家庄铁道大学
地址 050043 河北省石家庄市北二环东路17号

(72)发明人 郑丽娟 吴朋钢 赵美茹 吕亚奇 宫天 封柔昕 杨含玉 赵博远

(74)专利代理机构 河北冀华知识产权代理有限公司 13151
专利代理师 王占华

(51)Int.Cl.
G06F 21/60(2013.01)
H04L 9/08(2006.01)
G06F 16/13(2019.01)
G06F 16/14(2019.01)

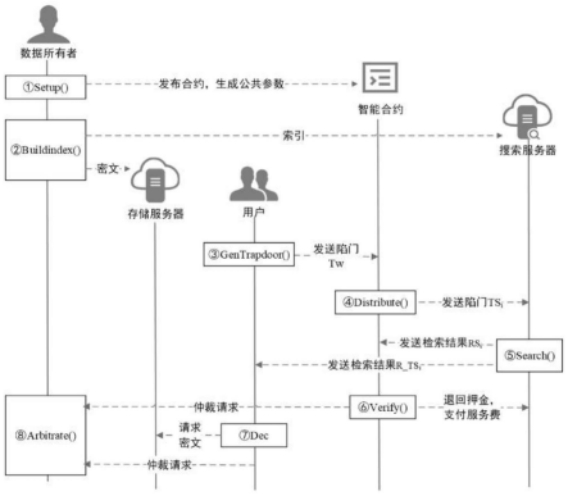
权利要求书4页 说明书10页 附图3页

(54)发明名称

基于区块链的多关键字可搜索加密方法

(57)摘要

本发明公开了一种基于区块链的多关键字可搜索加密方法，包括如下步骤：初始化；索引构建；陷门生成；分发；检索；验证；解密；仲裁。在所述方法中利用分发算法，将检索任务分解后由多个搜索服务器运行，降低了对单一云服务器的依赖，去中心化程度更高。在所述方法中利用仲裁算法和验证算法检测云服务器的恶意行为，验证检索结果，保证了用户与云服务器之间的公平交易。在检索算法中设计了关键字权重机制，能够在检索结果为空时基于权重提供模糊结果。实验分析和对比表明本申请具有较高的执行效率。



CN 114741711 A

1.一种基于区块链的多关键字可搜索加密方法,其特征在于:

初始化:数据所有者生成公共参数和密钥,并创建智能合约,将公共参数和公钥公开,将私钥保密;

索引构建:数据所有者生成密文文件、索引、共享密钥和其他参数,将生成的密文文件发送至存储服务器,将索引发送至搜索服务器,将共享密钥发送给其他用户,将生成的其他参数保密;

陷门生成:用户将需要检索的多个关键字生成检索陷门,并发起检索请求;

分发:智能合约将用户的检索请求分解后发送给各个搜索服务器;

检索:搜索服务器处理检索请求,将得到的检索结果发送给用户和智能合约;

验证:智能合约对检索结果进行验证,对验证失败的结果发起仲裁请求;

解密:用户根据检索结果进行计算,根据计算结果向存储服务器请求对应的密文文件,将密文文件解密后得到对应的明文文件,如果计算或解密失败则发起仲裁请求;

仲裁:数据所有者处理仲裁请求,校验检索结果。

2.如权利要求1所述的基于区块链的多关键字可搜索加密方法,其特征在于,所述初始化包括如下步骤:

1)数据所有者生成两个阶为素数p的乘法循环群G_1和G_2,g为G_1的生成元;生成一个双线性映射$e:G_1\times G_1\rightarrow G_2$;选择伪随机函数$F:\{0,1\}^*\rightarrow\{0,1\}^k$;选择两个密码哈希函数$H_1:\{0,1\}^*\rightarrow G_1$与$H_2:G_2\rightarrow\{0,1\}^k$,k为整数;选择抗碰撞哈希函数$H_3:\{0,1\}^*\rightarrow\{0,1\}^k$,$H_4:\{0,1\}^*\rightarrow\{0,1\}^k$;选择对称加密算法(Enc,Dec);数据所有者公布系统公共参数$pub=\{p,G_1,G_2,e,F,H_1,H_2,H_3,H_4,(Enc,Dec)\}$,并将其发布至区块链;

数据所有者随机选取$s\in Z_p$计算$S_{DO}=g^s$作为私钥,并计算公钥$P_{DO}=g^{\frac{1}{s}}$;用户随机选取$u\in Z_p$,计算$S_u=g^{\frac{1}{u}}$作为私钥,计算用户公钥$P_u=g^u$;数据所有者和用户将私钥保存,将公钥作为公开信息发布至区块链;

2)数据所有者创建智能合约并将其发布至区块链;用户及搜索服务器在方案执行过程中需要调用智能合约以完成整个检索过程;在每次检索完成后,用户可提取智能合约中扣留的押金;智能合约的功能包括:暂存用户服务费和搜索服务器的押金,分解和发送检索任务,验证检索结果。

3.如权利要求1所述的基于区块链的多关键字可搜索加密方法,其特征在于,所述索引构建包括如下步骤:

1)数据所有者将明文文件进行关键字拆分,得到关键字集合$W=\{W_1,w_2,w_3,...\}$;数据所有者使用对称算法加密明文文件得到密文文件,随机选取$x\in Z_p$,对于明文文件M_i,其对应的密文文件$C_i=Enc(M_i,k)$,k为对称密钥,$k=H_2(e(g,g^x))$;密文文件集合为$C=\{C_1,C_2,C_3,...\}$,对于文件C_i,将密文文件排序后按照序号i分配文件编号$FN_i=2^{i-1}$,并计算$H_4(C_i)$,得到$VC=\{[FN_1,H_4(C_1)],[FN_2,H_4(C_2)],...,[FN_i,H_4(C_i)]\}$;数据所有者将所有密文文件C及其编号发送至存储服务器,将VC发布至区块链。

2)数据所有者利用密文文件编号及其对应的关键字,构建关键字-文档编号文件索引I,计算过程如下:

设包含关键字w的密文文件编号集合为$\{FN_1,FN_2,FN_3,...\}$;

2-1)利用关键字w,计算$y=H_3(w)$;

2-2)计算$C_w=e(g^x,g^y)=e(g,g)^{xy}$;

2-3)令$Key=g^{sx}$,计算$b=F(Key)$;

2-4)计算$verify_bit=e(H_1(w),Key) \bmod 2$;

2-5)计算聚合后的文件编号$Nw=FN_1\&FN_2\&FN_3\&...$,并将verify_bit插入到Nw的第b个比特位中,&表示与运算;

关键字w对应的索引为$Iw=[Cw,Nw]$,数据所有者对关键字集合W中的每个关键字均执行以上操作,生成索引$I=\{Iw_1,Iw_2,...\}$,之后将索引上传至各个搜索服务器;同时,数据所有者计算$VN=H_4(Nw||x)$,$Vw=H_4(w||Nw||x)$生成集合$VRs=\{VN_1,VN_2,VN_3,...\}$和$VRu=\{Vw_1,Vw_2,Vw_3,...\}$,用于对有争议的检索结果进行仲裁;

3)数据所有者将共享密钥$Key=g^{sx}$发至各个用户,用户必须利用此密钥才能构造合法的检索陷门;数据所有者将x,VRs,VRu,S_{DO}作为秘密信息保存。

4.如权利要求1所述的基于区块链的多关键字可搜索加密方法,其特征在于,所述陷门生成包括如下步骤:

1)设待检索关键字集合为$W'=\{w_1,w_2,w_3,...\}$,用户需要同时为每个关键字指定权重p,得到集合$W_p=\{[w_1,p_1],[w_2,p_2],[w_3,p_3],....\}$;权重代表各个关键字的检索优先级,主要用于在检索结果为空时计算模糊结果,权重越高的关键字包含在检索结果中的可能性越大;如用户不指定权重,则默认按照顺序分配,$p_i=2^i$,i为关键字序号;

对于关键字$w\in W'$,其陷门的计算过程如下:

1-1)计算$y=H_3(w)$;

1-2)随机选取$r\in Z_p$,计算$t1=e(g^r,S_u)=e(g,g)^{\frac{r}{u}}$,$t2=e(Key\cdot g^y\cdot g^r,P_{DO})=e(g^{rsxy},P_{DO})=e(g,g)^{rxy}$;

1-3)关键字w的检索陷门$tw=[t1,t2]=\left[e(g,g)^{\frac{r}{u}},\ e(g,g)^{rxy}\right]$;

2)对每个待检索关键字执行以上操作后得到陷门集合$Tw=\{[tw_1,p_1],[tw_2,p_2],[tw_3,p_3],...\}$,之后用户将陷门集合Tw及服务费发送至智能合约;由于智能合约在验证检索结果时可能需要数据所有者进行仲裁,用户在发送的服务费需要包含一部分仲裁费用。

5.如权利要求1所述的基于区块链的多关键字可搜索加密方法,其特征在于,所述分发包括如下步骤:

1)合约账户收到用户发送的陷门集合和服务费之后的时间t内,参与检索的搜索服务器将押金发送至合约账户;合约账户收到押金后将在时间t内成功支付押金的搜索服务器按支付顺序排序,向这些服务器分发检索陷门,超时的支付押金的交易将被退回原账户;最后得到参与检索的搜索服务器集合S;

2)为保证检索结果的可靠性,每个陷门都要由2台或以上数量的搜索服务器检索;智能合约将收到的陷门集合Tw分解为多个子集,然后将得到的子集分发给搜索服务器,Tw中的每个陷门至少包含于2个子集中;搜索服务器S_i收到的子集为TS_i。

6.如权利要求1所述的基于区块链的多关键字可搜索加密方法,其特征在于,所述检索包括如下步骤:

1)各搜索服务器收到智能合约发送的陷门集合TS_i后,对集合中每个陷门执行本地检

索，得到检索结果集合RS_i＝{Nw_1,Nw_2,...,Nw_j,j∈(1,m)}，j代表陷门编号，Nw_j代表编号为j的陷门的检索结果；

2)搜索服务器执行本地检索时对于每个陷门$tw' = [t1', t2'] \subseteq TS_i$，利用本地存储的索引Iw'＝[Cw',Nw']，计算等式t1'·Cw'＝t2'是否成立，若成立则输出1，将对应的Nw'加入集合RS_i中；若不成立则输出0，将0加入集合RS_i中，表明该陷门对应的检索结果为空；

3)搜索服务器将检索结果集合RS_i发送至智能合约用于验证；若集合RS_i中检索结果均为0，则表明本次检索结果为空，检索流程结束；若集合RS_i中检索结果中存在非0结果，则继续进行计算；

4)对于陷门集合TS_i，对应包含该集合中所有关键字的检索结果为R_TS_i＝Nw_1&Nw_2&...&Nw_j，参与检索的服务器首先计算R_TS_i，若R_TS_i＞0，表明检索结果不为空，则将检索结果R_TS_i发送给用户；若R_TS_i＝0，则按照关键字权重计算陷门TS_i的子集的检索结果的权重，将权重最大的结果R_SubTS_i作为模糊结果返回给用户。

7.如权利要求1所述的基于区块链的多关键字可搜索加密方法，其特征在于，所述验证包括如下步骤：

1)假设2台搜索服务器对某一关键字的检索结果为Nw_i与Nw_i'，智能合约验证Nw_i＝Nw_i'是否成立，即每个编号相同的陷门对应的检索结果是否相同，若成立则输出1；如果不成立则输出0，并向数据所有者发起仲裁请求，请求中包含Nw_i与Nw_i'；所有结果验证通过后，智能合约退回各个搜索服务器的押金，并按照检索的陷门数量向搜索服务器分发服务费；若验证失败，智能合约保留押金并拒绝支付对应搜索服务器的服务费；

2)智能合约验证完成后将验证结果发送至用户，同时发起一笔交易将押金退回给搜索服务器；经过时间t后，再将服务费发送给搜索服务器；在时间t内，若用户对检索结果有怀疑可向数据所有者发起仲裁请求，该笔支付服务费的交易将被暂停。

8.如权利要求1所述的基于区块链的多关键字可搜索加密方法，其特征在于，所述解密包括如下步骤：

1)参与检索的搜索服务器为n个时，用户收到n个检索结果{R_TS_1,R_TS_2,R_TS_3,...,R_TS_n}，检索结果代表包含相应关键字的密文文件编号集合；将检索结果求交集即可得到R_Tw'，R_Tw'＝R_TS_1&R_TS_2&R_TS_3&...&R_TS_n；

用户计算b＝F(Key)，然后将R_Tw'的第b个比特位删除，得到最终的结果R_Tw；用户将结果R_Tw发送至存储服务器请求对应的文件；存储服务器从R_Tw按比特位拆分出文件编号，查找到对应的文件后将其发送给用户；

2)用户获取密文文件C后首先计算得到对称密钥

$$k = H_2\left(e(P_{DO}, Key)\right) = H_2\left(e\left(g^{\frac{1}{s}}, g^{sx}\right)\right)$$

，之后执行对称解密算法即可获取对应的明文文件M＝Dec(C,k)；根据密文文件C还可以计算H_4(C)'并将其与区块链中存储的H_4(C)进行比较，从而判断密文文件是否被篡改。

9.如权利要求1所述的基于区块链的多关键字可搜索加密方法，其特征在于，所述仲裁包括如下步骤：

1)对于来自智能合约的仲裁请求，数据所有者利用待仲裁的检索结果Nw″，计算H_4(Nw″|

|x)∈VRs是否成立;若成立则输出1,表明检索结果合法;否则输出0,表明检索结果不合法,此时智能合约扣留押金并拒绝向搜索服务器支付服务费;

2)对于来自用户的仲裁请求,数据所有者利用待仲裁的检索结果Nw″和用户提交的原始关键字w”计算H_4(w”||Nw″||x)∈VRu是否成立;若成立则输出1,表明检索结果准确;否则输出0,表明检索结果不准确;数据所有者将对应搜索服务器的信息广播至区块链中,智能合约不再向该服务器支付服务费。

基于区块链的多关键字可搜索加密方法

技术领域

[0001] 本发明涉及区块链技术领域,尤其涉及一种基于区块链的多关键字可搜索加密方法。

背景技术

[0002] 云计算近年来得到了蓬勃发展,大量公司开始向社会提供云服务。云服务种类包括平台即服务、基础设施即服务和软件即服务。用户可以根据自身需要选择购买或租用不同的云服务。越来越多的机构也开始将数据和应用转移到云服务器中。

[0003] 云服务带来更多便利的同时,其存在的安全和隐私泄露风险也不容忽视。个人用户最常用的就是租用云存储服务器存储个人数据。例如苹果公司的iCloud。用户可以将手机中的照片、视频和其他类型的文件存储在iCloud云服务器中。这在节约本地存储空间的同时,用户可以不限设备随时随地查看iCloud中的文件。

[0004] 但由于这些信息都是明文存储在云服务器中的,数据一旦泄露,将对用户的隐私造成严重侵害。通过将数据加密后上传至云服务器能够保护数据安全,但对数据的管理和使用带来了不便,特别是用户使用较频繁的检索操作。

[0005] 为了实现在密文上的关键字搜索,学者们提出了可搜索加密。现有可搜索加密方案的研究中,对算法的表达能力的提升是重要的研究方向。其中多关键字检索是研究的热点,学者探索了支持逻辑连接的关键字检索方案,或者利用kNN(K-NearestNeighbor)、Word2vec等技术来计算和评估关键字与密文文档的相似程度,实现多关键字检索。学者们还提出了更方便的模糊检索方案。

[0006] 然而除模糊检索外,现有的多关键字可搜索加密方案只能提供精确结果并在此基础上实现对检索结果的排序。当用户检索的多个关键字没有对应的文档时检索结果为空,用户体验较差。

[0007] 大量可搜索加密方案中假设云服务器是诚实的,在实际应用中云服务可能是不诚实的或恶意的,其提供的检索结果不可靠。因此引入对检索结果的验证机制,从而防范云服务器恶意行为也是可搜索加密方案的重要研究方向,其中基于区块链的可搜索加密方案是研究的热点。但这些可搜索加密方案中云服务器为逻辑上的单一节点,中心化程度较高,而且不能杜绝云服务器的恶意行为。例如,当云服务器恶意返回空结果时,可能会使最终的检索结果为空。

[0008] 目前大量可搜索加密方案中均假设服务器是诚实的,但在实际应用中,云服务器可能会由出于成本考虑,在方案运行过程中出现不诚实的行为。例如返回不完整的检索结果以减少流量消耗,也可能返回随机结果或空结果以节约计算资源。这对用户与云服务器之间的交易公平存在很大损害,需要引入对检索结果的验证机制。

[0009] 区块链技术的应用能够有效解决这一问题。区块链中的数据不可篡改,且可验证可追溯。区块链能够作为可信方对检索结果进行验证,从而保证交易公平。

[0010] 现有方案中通过区块链建立的验证机制在一定程度上保证了交易公平,但这些方

案中云服务器依然是逻辑上的单一节点，去中心化程度不够高。当云服务器出现恶意行为，只能通过扣除押金的方式进行惩罚，在方案的运行中依然需要依赖有过恶意行为的云服务器。

发明内容

[0011] 本发明所要解决的技术问题是如何提供一种能够提高检索效率、保证交易公平以及具有较高灵活性的基于区块链的多关键字可搜索加密方法。

[0012] 为解决上述技术问题，本发明所采取的技术方案是：一种基于区块链的多关键字可搜索加密方法，其特征在于：

[0013] 初始化：数据所有者生成公共参数和密钥，并创建智能合约，将公共参数和公钥公开，将私钥保密；

[0014] 索引构建：数据所有者生成密文文件、索引、共享密钥和其他参数，将生成的密文文件发送至存储服务器，将索引发送至搜索服务器，将共享密钥发送给其他用户，将生成的其他参数保密；

[0015] 陷门生成：用户将需要检索的多个关键字生成检索陷门，并发起检索请求；

[0016] 分发：智能合约将用户的检索请求分解后发送给各个搜索服务器；

[0017] 检索：搜索服务器处理检索请求，将得到的检索结果发送给用户和智能合约；

[0018] 验证：智能合约对检索结果进行验证，对验证失败的结果发起仲裁请求；

[0019] 解密：用户根据检索结果进行计算，根据计算结果向存储服务器请求对应的密文文件，将密文文件解密后得到对应的明文文件，如果计算或解密失败则发起仲裁请求；

[0020] 仲裁：数据所有者处理仲裁请求，校验检索结果。

[0021] 采用上述技术方案所产生的有益效果在于：1)本申请所述方法使用去中心化的多服务器架构，将数据存储和数据检索分别交由多个云服务器。在此基础上设计了分发算法，能够将包含多关键字的检索任务分解后分发给多个搜索服务器执行，提高了检索效率，同时实现了负载均衡。

[0022] 2)本申请所述方法中设计了验证算法和仲裁算法，能够验证检索结果，防范云服务器的恶意行为，保证交易公平。在检索算法中引入关键字权重机制，能够在检索结果为空时将权重最高的模糊结果返回给用户，灵活性更强。

[0023] 3)在真实数据集上对本申请进行了对比实验，证明了本申请具有较高的效率。

附图说明

[0024] 下面结合附图和具体实施方式对本发明作进一步详细的说明。

[0025] 图1是本发明实施例所述方法的模型图；

[0026] 图2是本发明实施例所述方法的流程图；

[0027] 图3是本发明实施例中索引构建阶段耗时对比图；

[0028] 图4是本发明实施例中陷门生成阶段耗时对比图；

[0029] 图5是本发明实施例中关键字检索阶段耗时对比图；

[0030] 图6是本发明实施例中搜索服务器数量对所述方法总耗时的影响图。

具体实施方式

[0031] 下面结合本发明实施例中的附图，对本发明实施例中的技术方案进行清楚、完整地描述，显然，所描述的实施例仅仅是本发明的一部分实施例，而不是全部的实施例。基于本发明中的实施例，本领域普通技术人员在没有做出创造性劳动前提下所获得的所有其他实施例，都属于本发明保护的范围。

[0032] 在下面的描述中阐述了很多具体细节以便于充分理解本发明，但是本发明还可以采用其他不同于在此描述的其它方式来实施，本领域技术人员可以在不违背本发明内涵的情况下做类似推广，因此本发明不受下面公开的具体实施例的限制。

[0033] 本申请实施例中公开了一个基于区块链的多关键字可搜索加密方法，如图1所示：

[0034] 本申请实施例的参与方包括数据所有者(DO)、存储服务器(S_Store)、搜索服务器(S_Search)、用户(DU)，均作为区块链节点。数据所有者负责对文档进行加密生成密文数据并建立索引。存储服务器负责存储数据所有者生成的密文数据。搜索服务器负责存储索引数据，并根据用户上传的检索陷门计算和检索对应的索引数据。用户根据自身检索需要计算生成检索陷门，并通过区块链将其发送给搜索服务器，获取对应的索引数据。用户根据索引数据向存储服务器请求对应的密文数据。

[0035] 本申请包括以下8个多项式时间算法：

[0036] 1)初始化算法Setup(λ)→(pub,P_{DO},S_{DO},P_u,S_u)。算法以安全参数λ为输入，数据所有者生成公共参数pub，并将智能合约部署在区块链上。数据所有者和其他用户生成公私钥对，将私钥S_{DO}和S_u保密，将公钥P_{DO}和P_u公开；

[0037] 2)索引构建算法Buildindex(pub,W,M,S_{DO})→(C,I,Key,VC,VRs,VRu)。算法以公开参数pub，明文关键字集合W，明文文件集合M和数据所有者私钥S_{DO}为输入，输出密文文件集合C，密文索引I，共享密钥Key以及用于结果校验的集合VC,VRs,VRu。算法由数据所有者执行，对于集合C与索引I分别发送至存储服务器和搜索服务器，将搜索密钥Key发送至其他用户，将生成的集合VC公开在区块链上，集合VRs和VRu保密。

[0038] 3)陷门生成算法GenTrapdoor(Key,W',S_u,P_{DO})→Tw。算法以搜索密钥Key，待检索关键字集合W'，用户私钥S_u和数据所有者公钥P_{DO}为输入，生成陷门集合Tw后将其发送至智能合约。算法由用户执行，在执行时可为每个关键字指定权重。

[0039] 4)分发算法Distribute(Tw)→({TS_1,TS_2,TS_3,…})。算法以用户发送的陷门集合Tw为输入，智能合约执行该算法输出分解后的陷门集合{TS_1,TS_2,TS_3,…}，其中$TS_i \subseteq Tw, i = 1,2,3 \ldots$。

[0040] 5)检索算法Search(TS_i,I,P_u)→(RS_i,R_TS_i)。算法以陷门集合TS_i，密文索引I，用户公钥P_u为输入，各个搜索服务器执行该算法输出检索结果RS_i和R_TS_i。

[0041] 6)验证算法Verify(RS_i)→{0,1}。算法由智能合约执行，算法以各个搜索服务器发送的检索结果RS_i为输入，输出验证结果，并根据验证结果发送押金和服务费。

[0042] 7)解密算法Dec(R_TS_i,P_{DO},Key)→M。算法由用户执行，以检索结果R_TS_i，数据所有者公钥P_{DO}，搜索密钥Key为输入，得到检索结果对应的明文文件M。

[0043] 8)仲裁算法Arbitrate(w",Nw",x)→{0,1}。算法由数据所有者执行，以待仲裁的检索结果Nw"，待仲裁的关键字w"(可选)，秘密信息x为输入，输出验证结果。

[0044] 如图2所示，本发明实施例公开了一种基于区块链的多关键字可搜索加密方法，所

述方法中包括的8个多项式时间算法的具体内容及执行流程如下所述：

[0045] 初始化算法：Setup(λ)→(pub,P_{DO},S_{Do},P_u,S_u)。

[0046] 本申请首先需要执行初始化算法Setup()，对应于图2中步骤①。该算法的执行过程如下：

[0047] 1)数据所有者生成两个阶为素数p的乘法循环群G_1和G_2，g为G_1的生成元。生成一个双线性映射e:$G_1\times G_1\rightarrow G_2$。选择伪随机函数F：$\{0,1\}^*\rightarrow\{0,1\}^k$。选择两个密码哈希函数$H_1$:$\{0,1\}^*\rightarrow G_1$与$H_2$:$G_2\rightarrow\{0,1\}^k$，k为整数。选择抗碰撞哈希函数$H_3$:$\{0,1\}^*\rightarrow\{0,1\}^k$，$H_4$:$\{0,1\}^*\rightarrow\{0,1\}^k$。选择对称加密算法(Enc,Dec)。数据所有者公布系统公共参数pub＝{p,G_1,G_2,e,F,H_1,H_2,H_3,H_4,(Enc,Dec)}，并将其发布至区块链。

[0048] 数据所有者随机选取$s\in Z_p$计算$S_{DO}=g^s$作为私钥，并计算公钥$P_{DO}=g^{\frac{1}{s}}$。用户随机选取$u\in Z_p$，计算$S_u=g^{\frac{1}{u}}$作为私钥，计算用户公钥$P_u=g^u$。数据所有者和用户将私钥保存，将公钥作为公开信息发布至区块链。

[0049] 2)数据所有者创建智能合约并将其发布至区块链。用户及搜索服务器在方案执行过程中需要调用智能合约以完成整个检索过程。智能合约的功能包括：暂存用户服务费和搜索服务器的押金；分解和发送检索任务；验证检索结果。在每次检索完成后，用户可提取智能合约中扣留的押金。

[0050] 索引构建算法：Buildindex(pub,W,M,S_{DO})→(C,I,Key,VC,VRs,VRu)。

[0051] 初始化完成后，数据所有者需要执行Buildindex()算法，对应于图2中步骤②。该算法执行过程如下：

[0052] 1)数据所有者将明文文件进行关键字拆分，得到关键字集合W＝{w_1,w_2,w_3,…}。数据所有者使用对称算法加密明文文件得到密文文件。随机选取$x\in Z_p$，对于明文文件M_i，其密文文件C_i＝Enc(M_i,k)，k为对称密钥，k＝$H_2(e(g,g^x))$。密文文件集合为C＝{C_1,C_2,C_3,…}，对于文件C_i，将密文文件排序后按照序号i分配文件编号$FN_i=2^{i-1}$，并计算$H_4(C_i)$，得到集合VC＝{[FN_1,$H_4(C_1)$],[FN_2,$H_4(C_2)$],…,[FN_i,$H_4(C_i)$]}。数据所有者将所有密文文件C及其编号发送至存储服务器，将VC发布至区块链。此时密文文件的哈希值均锚定在区块链上，可以用于校验检索结果。

[0053] 2)数据所有者利用密文文件编号及其对应的关键字，构建关键字-文档编号文件索引I，计算过程如下：

[0054] 设包含关键字w的密文文件编号集合为{FN_1,FN_2,FN_3,…}。

[0055] 2-1)利用关键字w，计算y＝H_3(w)。

[0056] 2-2)计算$C_w=e(g^x,g^y)=e(g,g)^{xy}$。

[0057] 2-3)令$Key=g^{sx}$，计算b＝F(Key)。

[0058] 2-4)计算verify_bit＝e(H_1(w),Key)mod 2。

[0059] 2-5)计算聚合后的文件编号Nw＝FN_1&FN_2&FN_3&…，并将verify_bit插入到Nw的第b个比特位中，&表示与运算。

[0060] 关键字w对应的索引为Iw＝[Cw,Nw]。数据所有者对关键字集合W中的每个关键字均执行以上操作，生成索引I＝{Iw_1,Iw_2,…}，之后将索引上传至各个搜索服务器。同时，数据所有者计算VN＝H_4(Nw||x)，Vw＝H_4(w||Nw||x)生成集合VRs＝{VN_1,VN_2,VN_3,…}和VRu＝

$\{Vw_1,Vw_2,Vw_3,\cdots\}$,用于对有争议的检索结果进行仲裁。

[0061] 3)数据所有者将共享密钥$Key=g^{sx}$发至各个用户,用户必须利用此密钥才能构造合法的检索陷门。数据所有者将x,VRs,VRu,S_{DO}作为秘密信息保存。

[0062] 陷门生成算法:$GenTrapdoor(Key,W',S_u,P_{DO})\rightarrow Tw$。

[0063] 用户需要执行GenTrapdoor()算法将需要检索的关键字生成检索陷门,对应于图2中步骤③。该算法的执行过程如下:

[0064] 1)设待检索关键字集合为$W'=\{w_1,w_2,w_3,\cdots\}$,用户需要同时为每个关键字指定权重p,得到集合$W_p=\{[w_1,p_1],[w_2,p_2],[w_3,p_3],\cdots\}$。权重代表各个关键字的检索优先级,主要用于在检索结果为空时计算模糊结果,权重越高的关键字包含在检索结果中的可能性越大。如用户不指定权重,则默认按照顺序分配,$p_i=2^i$,i为关键字序号。

[0065] 对于关键字$w\in W'$,其陷门的计算过程如下:

[0066] 1-1)计算$y=H_3(w)$。

[0067] 1-2)随机选取$r\in Z_p$,计算$t1=e(g^r,S_u)=e(g,g)^{\frac{r}{u}}$,$t2=e(Key\cdot g^y\cdot g^r,P_{DO})=e(g^{rsxy},P_{DO})=e(g,g)^{rxy}$。

[0068] 1-3)关键字w的检索陷门$tw=[t1,t2]=\left[e(g,g)^{\frac{r}{u}},\ e(g,g)^{rxy}\right]$。

[0069] 2)对每个待检索关键字执行以上操作后得到陷门集合$Tw=\{[tw_1,p_1],[tw_2,p_2],[tw_3,p_3],\cdots\}$,之后用户将陷门集合Tw及服务费发送至智能合约。由于智能合约在验证检索结果时可能需要数据所有者进行仲裁,用户在发送的服务费需要包含一部分仲裁费用。当仲裁费用不足时,合约的验证功能会受到一定的影响。

[0070] 分发算法:$Distribute(Tw)\rightarrow(\{TS_1,TS_2,TS_3,\cdots\})$。

[0071] 用户将陷门及服务费发送至智能合约,智能合约执行Distribute()算法分解检索任务,并将子任务发送至各个搜索服务器节点,对应于图2中步骤④。该算法的执行过程如下:

[0072] 1)合约账户收到用户发送的陷门集合和服务费之后的时间t内,参与检索的搜索服务器将押金发送至合约账户。合约账户收到押金后将在时间t内成功支付押金的搜索服务器按支付顺序排序,向这些服务器分发用户陷门,超时的支付押金的交易将被退回原账户。最后得到参与检索的搜索服务器集合S。

[0073] 2)为保证检索结果的可靠性,每个陷门都要由2台或以上数量的搜索服务器检索;以2台搜索服务器为例,智能合约利用搜索服务器集合S和收到的检索陷门Tw执行如下算法Algorithm 1,其中n代表云服务器数量,m代表Tw中陷门数量,TS_i表示云服务器S_i收到的陷门集合,$TS_i\subseteq Tw$;

Algorithm 1：DistributeTrapdoor

Input: Tw
Optput: TS_i

[0074]
```
1. if n<2 then
2.   return False
3. else
4. for i=1 to i=m do
5.     TS[i % n] = TS[i % n] + Tw[i]
6.     TS[(i + 1) % n] = TS[(i + 1) % n] + Tw[i]
7.   end for
8. end if
```

[0075] 智能合约执行完毕后，将TS_i分发给对应的搜索服务器。

[0076] 检索算法：$Search(TS_i, I, P_u) \rightarrow (RS_i, R\_TS_i)$。

[0077] 搜索服务器收到智能合约发送的检索任务后执行Search()算法，将算法执行的结果发送给智能合约及用户，对应于图2中步骤⑤。该算法的执行过程如下：

[0078] 1)各搜索服务器收到智能合约发送的陷门集合TS_i后，对集合中每个陷门执行本地检索，得到检索结果集合$RS_i = \{Nw_1, Nw_2, \cdots, Nw_j, j \in (1,m)\}$，j代表陷门编号，$Nw_j$代表编号为j的陷门的检索结果。

[0079] 2)搜索服务器执行本地检索时对于每个陷门$tw' = [t1', t2'] \subseteq TS_i$，利用本地存储的索引$Iw' = [Cw', Nw']$，计算等式$t1' \cdot Cw' = t2'$是否成立，若成立则输出1，将对应的Nw'加入集合$RS_i$中。若不成立则输出0，将0加入集合$RS_i$中，表明该陷门对应的检索结果为空。

[0080] 3)搜索服务器将检索结果集合RS_i发送至智能合约用于验证。若集合RS_i中检索结果均为0，则表明本次检索结果为空，检索流程结束。若集合RS_i中检索结果中存在非0结果，则继续进行计算。

[0081] 4)对于陷门集合TS_i，对应包含该集合中所有关键字的检索结果为$R_TS_i = Nw_1 \& Nw_2 \& \cdots \& Nw_j$。参与检索的服务器首先计算$R_TS_i$，若$R_TS_i > 0$，表明检索结果不为空，则将检索结果$R_TS_i$发送给用户。若$R_TS_i = 0$，则按照关键字权重计算陷门$TS_i$的子集的检索结果的权重，将权重最大的结果$R_SubTS_i$作为模糊结果返回给用户。$R_SubTS_i$的计算过程如下Algorithm 2和Algorithm 3所示。

Algorithm 2：calculateR_TSi

Input：RSi
Output: R_TSi

```
1. turn RSi into binaryformat
2. s = number of items in RSi
3. for i=1 to i=s do
4. R_TSi = R_TSi&RSi[i]
5. end for
6. turn R_TSi into decimal format
7. if R_TSi != 0
8.    return R_TSi
9. else
10. execute Algorithm 3
```

[0082]

[0083] Algorithm 3：calculate max subset R_SubTSi

Input：RSi, weight = $[p_1, p_2, p_3, \ldots]$
Output: R_SubTSi

```
1. for i= 0 to i=s do
2.     result[i] = RSi[i]
3.     P[i] = weight[j]
4.     for j = i+1 to j=s do
5.         result[i] = temp
6.         var = temp &RSi[j]
7.         if var != 0
8.             result[i] = var
9.             P[i] = P[i] + weight[j]
10.    end for
11. end for
12. k = the index of max(P)
13. R_SubTSi = result[k]
14. return R_SubTSi
```

[0084]

[0085] 验证算法：Verify(RS_i)→{0,1}。

[0086] 智能合约执行Verify()算法对搜索服务器发送的检索结果RS_i进行交叉验证，对应于图2中步骤⑥。该算法的执行过程如下：

[0087] 1)假设2台搜索服务器对某一关键字的检索结果为Nw_i与Nw_i'，智能合约验证Nw_i＝Nw_i'是否成立，即每个编号相同的陷门对应的检索结果是否相同，若成立则输出1。如果不成立则输出0，并向数据所有者发起仲裁请求，请求中包含Nw_i与Nw_i'。所有结果验证通过后，智能合约退回各个搜索服务器的押金，并按照检索的陷门数量向搜索服务器分发服务费。若验证失败，智能合约保留押金并拒绝支付对应搜索服务器的服务费。

[0088] 2)智能合约验证完成后将验证结果发送至用户，同时发起一笔交易将押金退回给搜索服务器。经过时间t后，再将服务费发送给搜索服务器。在时间t内，若用户对检索结果有怀疑可向数据所有者发起仲裁请求，该笔支付服务费的交易将被暂停。

[0089] 解密算法:Dec(R_TS$_i$,P$_{DO}$,Key)→M。

[0090] 用户收到搜索服务器发送的检索结果后,需要执行Dec()算法得到对应的明文文件,对应于图2中步骤⑦。该算法的执行过程如下:

[0091] 1)参与检索的搜索服务器为n个时,用户收到n个检索结果{R_TS$_1$,R_TS$_2$,R_TS$_3$,…,R_TS$_n$},检索结果代表包含相应关键字的密文文件编号集合。将检索结果求交集即可得到R_Tw',R_Tw'=R_TS$_1$&R_TS$_2$&R_TS$_3$&…&R_TS$_n$。

[0092] 用户计算b=F(Key),然后将R_Tw'的第b个比特位删除,得到最终的结果R_Tw。用户将结果R_Tw发送至存储服务器请求对应的文件。存储服务器从R_Tw按比特位拆分出文件编号,查找到对应的文件后将其发送给用户。

[0093] 2)用户获取密文文件C后首先计算得到对称密钥 $k = H_2(e(P_{DO}, Key)) = H_2\left(e\left(g^{\frac{1}{s}}, g^{sx}\right)\right)$,之后执行对称解密算法即可获取对应的明文文件M=Dec(C,k)。根据密文文件C还可以计算H_4(C)'并将其与区块链中存储的H_4(C)进行比较,从而判断密文文件是否被篡改。

[0094] 以上为正常解密流程。此外,用户还可以对检索结果进行进一步的逻辑运算。例如,当检索关键字为{a,b,c,d},检索结果r=ra&rb&rc&rd,可计算包含关键字a,b,d的同时不含关键字c的结果r'=(r)&(!rc)。这些逻辑运算条件也可包含在检索请求中。

[0095] 如果计算检索结果R_Tw时出现R_Tw=0的情况,此时表明不存在同时包含所有关键字的文件,可将n个检索结果中权重最大的结果作为模糊结果,也可根据需要进一步在此基础上利用区块链中的单关键字检索结果RS$_i$进行逻辑运算,计算相应的检索结果。

[0096] 仲裁算法:Arbitrate(w",Nw",x)→{0,1}。

[0097] 数据所有者执行Arbitrate()算法处理由智能合约或用户发起的仲裁请求,对应于图2中步骤⑧。该算法的执行过程如下:

[0098] 1)对于来自智能合约的仲裁请求,数据所有者利用待仲裁的检索结果Nw",计算H_4(Nw"||x)∈VRs是否成立。若成立则输出1,表明检索结果合法。否则输出0,表明检索结果不合法,此时智能合约扣留押金并拒绝向搜索服务器支付服务费。

[0099] 2)对于来自用户的仲裁请求,数据所有者利用待仲裁的检索结果Nw"和用户提交的原始关键字w"计算H_4(w"||Nw"||x)∈VRu是否成立。若成立则输出1,表明检索结果准确。否则输出0,表明检索结果不准确。数据所有者将对应搜索服务器的信息广播至区块链中,智能合约不再向该服务器支付服务费。

[0100] 本申请在可搜索加密中引入区块链技术,提出了基于区块链的多关键字可搜索加密方法。通过方法中设计的分发算法和检索算法,使得多个搜索服务器同时参与检索过程,实现多关键字检索,降低了对单一的云服务器节点的依赖。

[0101] 从安全性方面将本申请与文献1[Xu P,Tang S,Xu P,et al.Practical multi-keyword and boolean search over encrypted e-mail in cloud server[J].IEEE Transactions on Services Computing,2019,14(6):1877-1889.]以及文献2[Yang X,Chen G,Wang M,et al.Multi-keyword certificateless searchable public key authenticated encryption scheme based on blockchain[J].IEEE Access,2020,8:158765-158777]的方案进行了对比,如表1所示。

[0102] 本申请与文献[1]、文献[2]都基于双线性配对构造了非对称可搜索加密方案，并实现了多关键字检索。本申请与文献[2]的方案均引入了区块链，同时利用智能合约实现了对检索结果的验证，保证了交易公平。两种方案都能够保证陷门的不可区分性安全，抵御内部和外部用户的关键字猜测攻击。文献[2]的方案将索引存储在区块链中，而由于区块链中的数据是透明的，索引存在一定的安全风险。本申请在运行过程中只需要在区块链中公开陷门，安全性更强。

[0103] 而文献[1]的方案基于传统的云服务器架构，该方案只包含了用户与云服务器之间的交互过程。该方案基于诚实的云服务器模型构建，缺少可信方对检索结果的验证，无法保证交易公平，也不能抵御内部关键字猜测攻击。

[0104] 表1安全性对比

[0105]

现有技术	泄露信息	云服务器模型	陷门不可区分性安全	抵御内部关键字猜测攻击	检索结果的完整性验证
文献[1]	无	诚实	Yes	No	No
文献[2]	索引、陷门	恶意	Yes	Yes	Yes
本申请	陷门	恶意	Yes	Yes	Yes

[0106] 从计算量方面将本申请与文献[1][2]的方案进行了对比，分析了各个算法在索引构建阶段、陷门生成阶段和检索阶段的计算量，如表2所示。

[0107] n表示明文文件的数量，m表示关键字的数量，k表示查询中的关键字数量，r表示查询结果对应的文件的数量(r<<n)。

[0108] T_M表示一个乘法运算的计算量，T_H表示一个哈希运算的计算量，T_E表示一次幂运算的计算量，T_P表示一次双线性配对运算的计算量，其中$T_H<T_M<T_E<T_P$。

[0109] 在索引构建阶段，由于索引结构的不同，文献[1]的方案的计算量主要受到文件数量的影响。本申请和文献[2]的方案受到关键字数量的影响较大，计算量相近。

[0110] 在陷门生成阶段，本申请和文献[1][2]的方案都与检索的关键字数量相关，文献[2]使用幂运算最少，计算量最小。本申请在此阶段涉及双线性配对运算，计算量最多。

[0111] 在检索阶段，文献[1]的方案受到关键字数量和文件数量的影响，计算量最高。文献[2]不受其他参数的影响，计算量最小。本申请的计算量适中。

[0112] 综上，本申请在以上三个阶段的总计算量适中，但本申请在保证安全的同时灵活性更强。

[0113] 表2计算量对比

[0114]

现有技术	索引构建阶段	陷门生成阶段	检索阶段
文献[1]	$(2T_E+T_P)\cdot n$	$3k\cdot T_E$	$4T_P+2(k-1)\cdot T_P\cdot r$
文献[2]	$mT_M+mT_H+(2m+3)T_E$	$(k+1)\cdot T_M+kT_H+(2k+3)\cdot T_E$	T_M+3T_P
本申请	$m\cdot(3T_E+2T_P+3T_H)$	$k\cdot(T_H+3T_E+2\cdot T_P)$	$m\cdot T_M$

[0115] 实验分析：

[0116] 通过搭建实验环境对本申请的执行效率进行了分析。实验环境为64位Windows10操作系统，CPU为AMD Ryzen7 4800H 2.90GHz，内存16GB。实验在本地虚拟机VMware Ubuntu16.04虚拟机中进行，编程语言为python，版本为python 3.5。

[0117] 使用SHA256算法作为本申请中的哈希函数，使用AES对称加密算法加密明文文件。本文使用的数据集为公开的Enron Email数据集，在实验中选取了其中3000个文件作为原始明文文档，并提取了其中1200个关键字用于实验分析。将本申请与文献[1][2]中的方案，在索引构建、陷门生成、搜索三个阶段的耗时进行了对比。实验过程中关键字数量从200递增至1200，步长为200。每次实验均重复50次并取平均值作为实验结果。

[0118] 在索引构建阶段的耗时如下图3所示。横坐标为关键字数量，纵坐标表示基于对应数量的关键字为3000个明文文档计算索引的耗时。文献[1]的方案在索引构建阶段主要受到文件数量的影响，几乎不受到关键字数量的影响。本申请与文献[2]的方案在这一阶段的耗时相近，如图3所示。

[0119] 在陷门生成阶段的耗时如图4所示。横坐标为关键字数量，纵坐标表示计算对应数量关键字的检索陷门所消耗的时间。三种方案均受到关键字数量的影响。文献[2]涉及的计算量最小，耗时最低。本方案与文献[1]的方案在此阶段的耗时相近，但由于本方案涉及双线性对运算，当关键字数量较多时耗时略高于文献[1]的方案。

[0120] 在检索阶段的耗时如图5所示，横坐标为检索请求中包含的关键字数量，纵坐标为完成对应检索请求的耗时。由于本申请支持使用多个搜索服务器执行检索过程，在图5中分别给出了本申请在S_Search＝1和S_Search＝4的情况下的耗时。其中S_Search表示云服务器。本方案与文献[1]的方案的耗时与关键字数量线性相关。文献[1]的耗时几乎不受关键字数量的影响。与文献[1][2]的方案相比，本方案在此阶段的耗时适中。

[0121] 最后，测试了搜索服务器数量对本方案的总耗时的影响，如下图6所示。总耗时统计了在200个关键字和3000个明文文档数量的情况下，本申请在索引构建阶段、陷门生成阶段和检索阶段的总时间。每次检索均随机选取20个关键字构建检索陷门并分发给参与检索的搜索服务器。可以看出，通过提高搜索服务器的数量可以提高本申请的运行效率。

[0122] 本申请提出了基于区块链的多关键字可搜索加密方法，该申请将密文数据存储在存储服务器中，将索引存储在搜索服务器中。在此基础上设计了多关键字任务分发算法和检索算法，使得多个搜索服务器同时参与方法运行，提高了方法的去中心化程度。设计了仲裁机制和基于智能合约的验证机制，能够防范云服务器的恶意行为，保证用户与云服务器之间的交易公平。同时还设计了关键字权重机制，能够在多关键字检索结果为空时返回模糊结果给用户，具有更高的灵活性。通过实验分析，证明了本方法具有较高的效率。

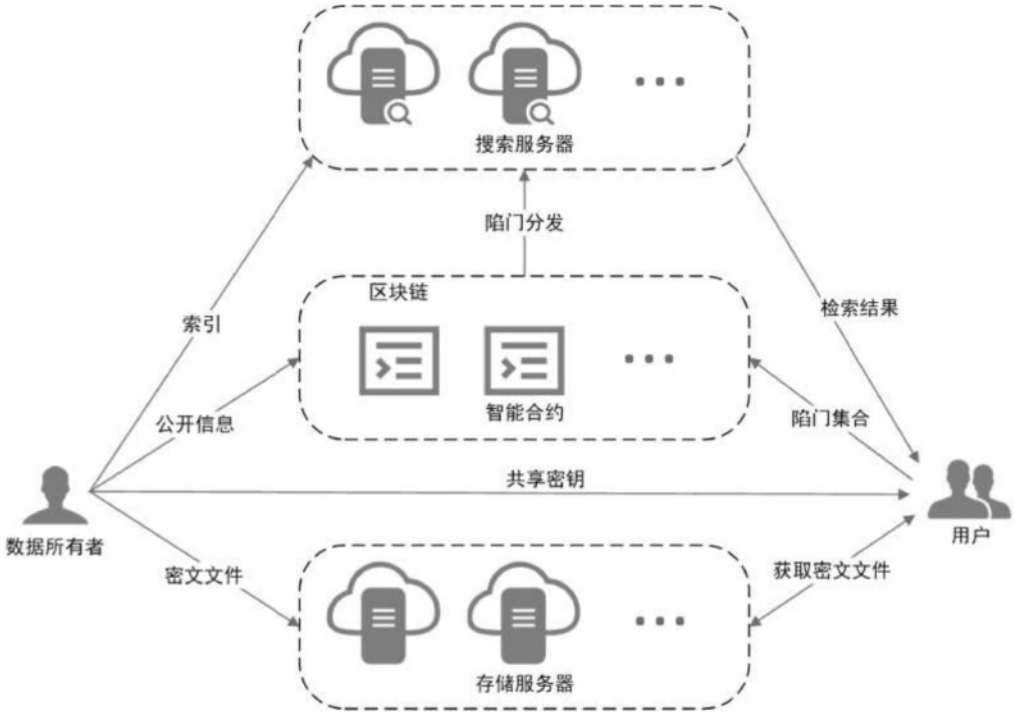

搜索服务器
陷门分发
区块链
索引
检索结果
智能合约
公开信息
陷门集合
共享密钥
数据所有者
用户
密文文件
获取密文文件
存储服务器

图1

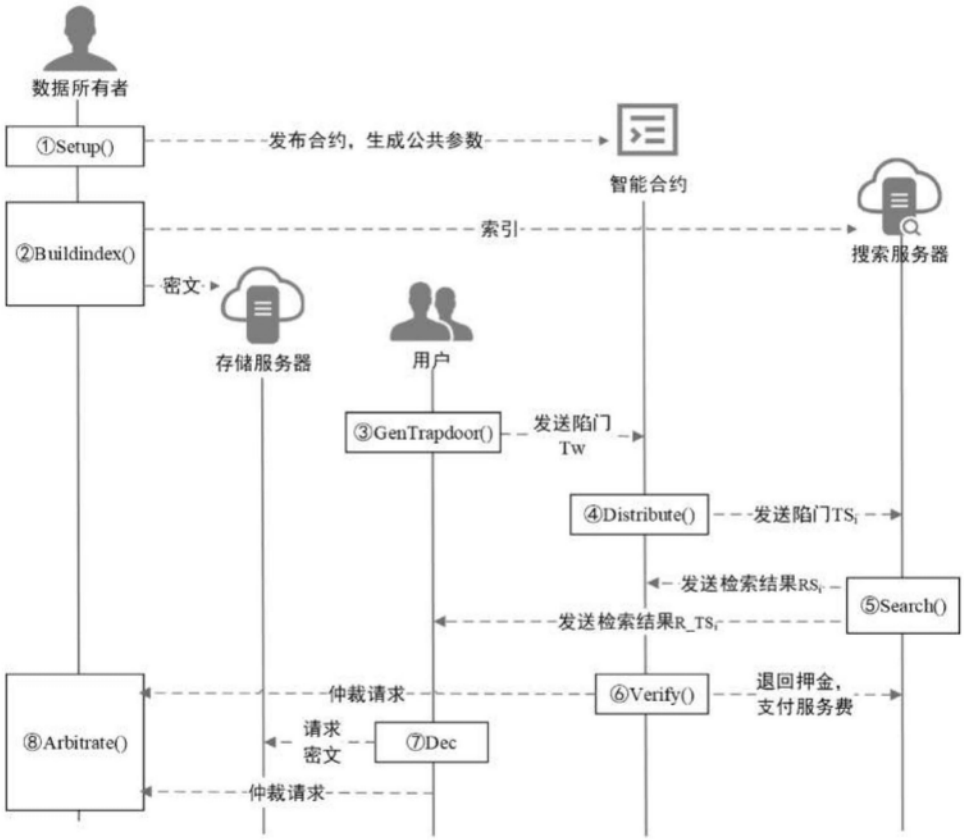

数据所有者
①Setup()
发布合约，生成公共参数
智能合约
②Buildindex()
索引
搜索服务器
密文
存储服务器
用户
③GenTrapdoor()
发送陷门
Tw
④Distribute()
发送陷门TSi
发送检索结果RSi
⑤Search()
发送检索结果R_TSi
⑥Verify()
仲裁请求
退回押金，
支付服务费
⑧Arbitrate()
请求
密文
⑦Dec
仲裁请求

图2

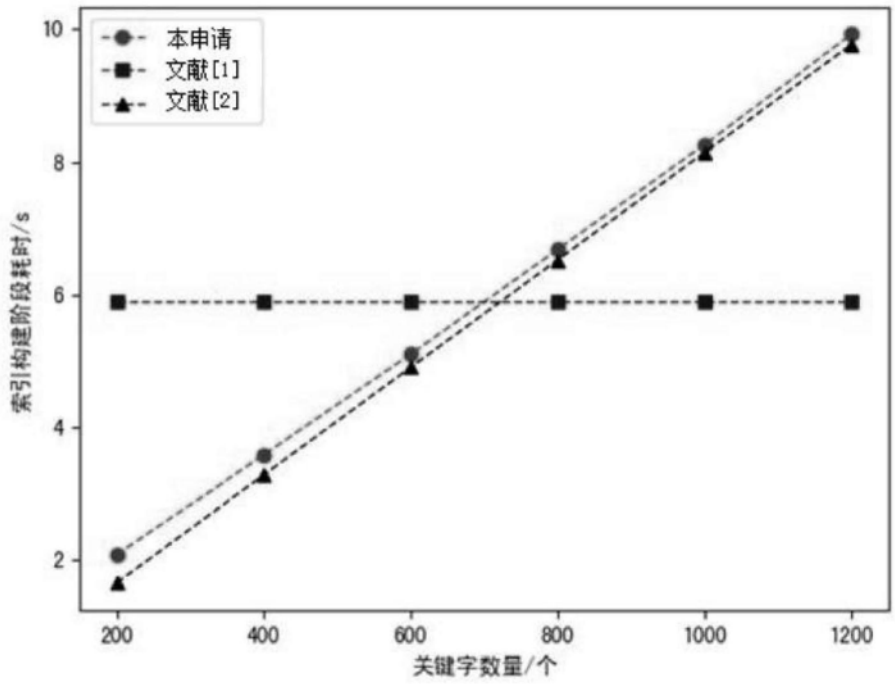

本申请
文献[1]
文献[2]
索引构建阶段耗时/s
关键字数量/个
2
4
6
8
10
200
400
600
800
1000
1200

图3

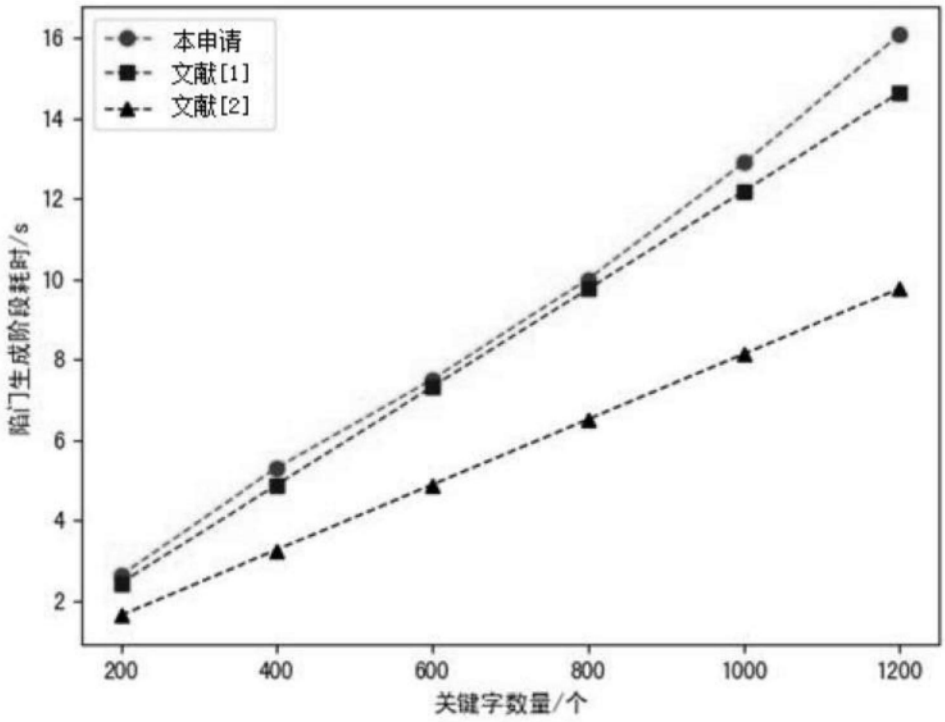

本申请
文献[1]
文献[2]
陷门生成阶段耗时/s
关键字数量/个
2
4
6
8
10
12
14
16
200
400
600
800
1000
1200

图4

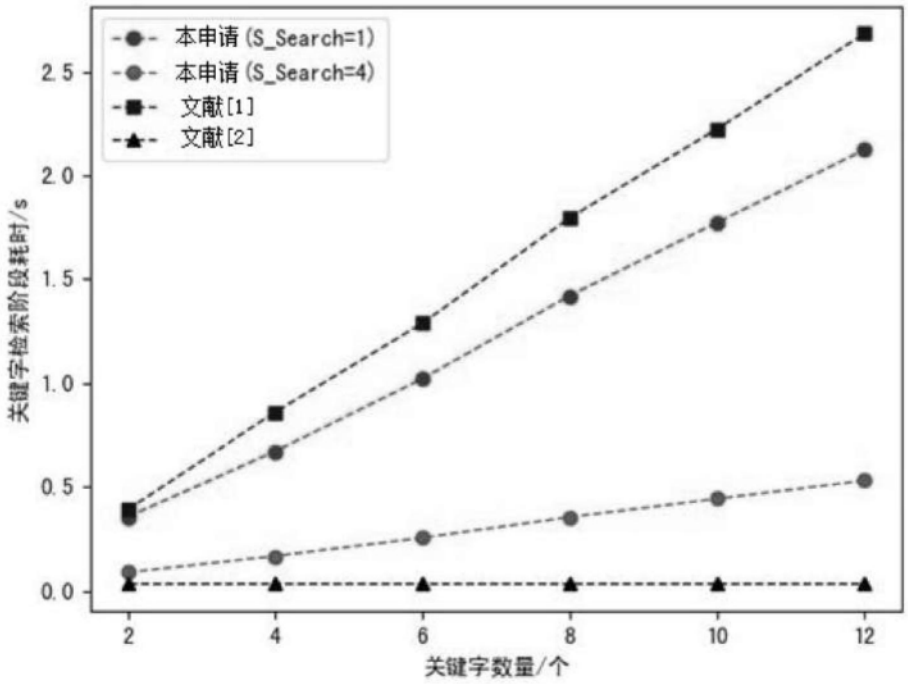

本申请(S_Search=1)
本申请(S_Search=4)
文献[1]
文献[2]
关键字检索阶段耗时/s
关键字数量/个
0.0
0.5
1.0
1.5
2.0
2.5
2
4
6
8
10
12

图5

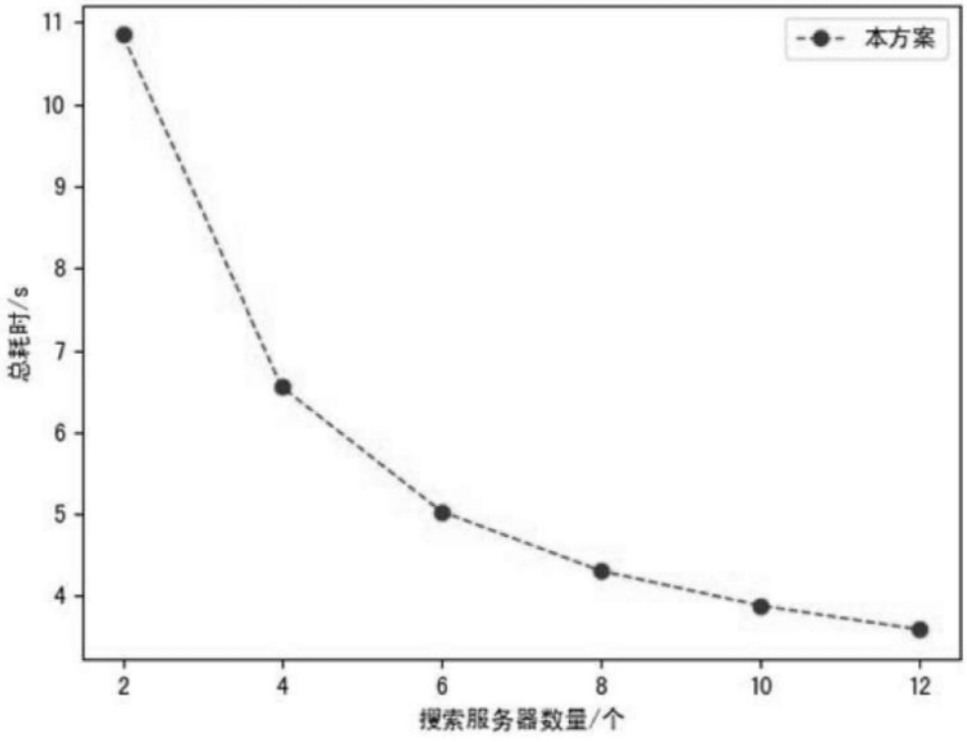

本方案
总耗时/s
搜索服务器数量/个
4
5
6
7
8
9
10
11
2
4
6
8
10
12

图6