一种支持多用户的公平密文检索方案

崔永杰[1,2],彭长根[1,2,3],丁红发[2,3],许德权[1,2]
(1. 贵州大学 计算机科学与技术学院,贵州 贵阳 550025;
2. 贵州省公共大数据重点实验室(贵州大学),贵州 贵阳 550025;
3. 贵州大学 密码学与数据安全研究所,贵州 贵阳 550025)

摘 要:大量用户将私密数据外包到云服务器上以缓解本地存储压力。为了保障数据的安全性,用户上传至云服务器前通常会执行数据加密处理,最后利用可搜索加密技术实现对密文数据的安全有效检索。然而,现有的方案通信模式通常是一对一的以及存在用户与云服务器间搜索交易的不公平性问题,即用户成功支付服务费后,云服务器并没有向用户返回正确且完整的检索结果。针对上述问题并考虑该场景下支持多用户检索的情况,提出一种基于国密算法的多用户公平可搜索加密方案。利用 CP-ABE 对属性私钥指定树形访问结构,实现密文数据的细粒度访问控制;然后,结合加解密效率高的 SM4 分组算法对数据集进行处理生成密文;最后根据区块链的公平机制以及智能合约自动执行的特点解决云服务器与用户之间的交易公平性问题,并且交易可追踪且不可逆转的特性使方案不需额外验证,从而减少计算开销。实验结果表明,所述方案在安全索引生成阶段耗时均处于毫秒级,对比传统方案具有优势;而且方案内的数据集增多,密文检索的效率并不随着线性增长,具有一定的稳定性。

关键词:可搜索加密;属性加密;智能合约;公平交易;国密算法
中图分类号:TP309 **文献标识码**:A **文章编号**:1673-629X(2022)10-0100-08
doi:10.3969/j.issn.1673-629X.2022.10.017

A Fair Searchable Encryption Scheme Supporting Multiple Users

CUI Yong-jie[1,2],PENG Chang-gen[1,2,3],DING Hong-fa[2,3],XU De-quan[1,2]
(1. School of Computer Science and Technology,Guizhou University,Guiyang 550025,China;
2. Guizhou Province Key Laboratory of Public Big Data (Guizhou University),Guiyang 550025,China;
3. Institute of Cryptography and Data Security,Guizhou University,Guiyang 550025,China)

Abstract:A large number of users outsource private data to cloud servers to relieve local storage pressure. In order to ensure the security of data,users usually perform data encryption before uploading to the cloud server,and finally use searchable encryption technology to achieve the secure and effective retrieval of ciphertext data. However,the communication mode of the existing solutions is usually one-to-one and there is a problem of unfairness in the search transaction between the user and the cloud server,that is,after the user successfully pays the service fee,the cloud server does not return the correct and complete search result to the user. Aiming at the above problems and considering the support of multi-user retrieval in this scenario,a multi-user fair searchable encryption scheme based on the national secret algorithm is proposed. First,we use CP-ABE to specify a tree access structure for the attribute private key. Then,combined with SM4 grouping algorithm with high encryption and decryption efficiency,the data set is processed to generate ciphertext. Finally,according to the fairness mechanism of the blockchain and feature of automatic execution of smart contracts,the problem of fairness of transactions between cloud servers and users is solved,which guarantees data security based on the traceable and irreversible characteristics of transactions while supporting result verification. Experiment shows that the proposed scheme consumes milliseconds in the security index generation stage,which has advantages compared with traditional schemes. The data set in the scheme increases,and the efficiency of ciphertext retrieval does not increase linearly with certain stability.

Key words:searchable encryption;attribute encryption;smart contract;fair transaction;national secret algorithm

收稿日期:2021-11-05 **修回日期**:2022-03-09
基金项目:国家自然科学基金项目(U1836205);贵州省科技计划基金项目(黔科合平台人才[2020]5017);贵州省教育厅自然科学项目(黔教合 KY 字[2021]140)
作者简介:崔永杰(1997-),男,硕士研究生,研究方向为密码学、可搜索加密;通讯作者:彭长根(1963-),男,博士生导师,二级教授,CCF 高级会员(48309S),研究方向为密码学。

0　引　言

云存储服务使得大量用户将本地私有数据外包存储至云端存储,并可与特定数据用户共享数据,但这也使用户失去了对数据的物理控制。因此,用户通常会对外包数据执行加密操作,然而,密文数据阻碍了数据的共享使用,使得云计算失去了它本身的优势。为了提高云环境下密文数据的灵活性,即解决加密数据如何检索的问题,Song 等人[1]提出了可搜索加密技术(Searchable Encryption,SE)。该技术支持用户在密文数据中执行关键词检索方案。

2004 年,文献[2]提出了公钥可搜索加密的概念,该方案利用双线性空间的基于身份加密算法、指数运算,适用于复杂的加密环境。文献[3]在 IND-CKA 的基础上提出了可搜索加密的自适应语义安全模型的定义,同时,该方案中论证了自适应语义安全即是查询过程中密文不可区分性;并在安全模型的基础上构建了两个可搜索加密方案,一个是非自适应语义安全的,一个是自适应语义安全的。但该方案都只支持单关键字搜索,且存在检索效率低的问题。文献[4]引入了编辑距离,提出了一种基于公钥的模糊可搜索加密方案。通过控制编辑距离小于某个阈值实现模糊检索,并引入了通识符来进一步减少计算开销。

然而,上述方案存在检索效率以及通信模式是一对一的问题,更加适合单用户场景下的检索;为了满足支持多用户场景下的可搜索加密方案,文献[5]提出了密文策略属性基加密系统(CP-ABE)这一概念。该方案中,用户的私钥是由密钥生成中心(KGC)根据系统公共参数的主密钥和用户的属性计算得出,命名为属性私钥。密文对应访问结构,只有属性集合里面的属性满足访问控制才可以成功解密密文,以此实现多用户场景下的可搜索加密方案。2014 年,文献[6]首次定义了基于属性的可搜索加密算法,扩大了信息的共享性,并减少了存储开销。

但是,上述方案存在恶意云服务器的问题,没有诚实地向用户返回准确的检索结果,存在交易不公平的问题,不能实现文献[7]给出的公平性定义。

因此,文献[8]提出了一种基于区块链的公平 SSE 方案,减少云服务器可能会返回错误的结果,保证用户可以得到正确的检索结果,利用区块链公平公正的特性溯源以验证其检索结果。安全性和性能分析表明,该方案在语义上是安全可行的。文献[9]提出了区块链上基于云辅助的属性基可搜索加密方案,利用了区块链的公开透明机制实现安全搜索及其不可篡改性确保关键字的完整性,但还是存在检索效率低的问题。

为了支持多用户场景下安全高效的公平检索,该文将属性加密机制,国密系列算法 SM3、SM4 以及区块链的智能合约技术相结合,提出一种支持多用户的公平国密可搜索加密算法。主要贡献如下:

(1)将国密算法与传统可搜索加密方案结合,SM3 杂凑算法可避免高概率的局部碰撞,生成更具安全性的 256 比特的索引。而 SM4 分组算法的优势在于加解密算法结构相同,可提升数据集的加解密效率;再引入属性加密,利用 CP-ABE 对共享密文数据生成树形访问结构,只要用户群体的属性私钥满足访问结构,就能获取解密密钥,打破了传统通信模式一对一的局限性。

(2)引入智能合约自动执行合约内容的特点,通过部署四个合约函数在虚拟的区块链环境中,以实现公平性协议。该协议解决了云环境下检索外包密文数据时,用户与云服务器之间的交易不公平性问题,且满足四级公平性。

(3)由实验结果可知,安全索引生成耗时均处于毫秒级,对比现有的同类方案具有一定优势,总体耗时呈现为线性增长的曲线。在检索阶段,数据集的增多并不影响云服务器检索到包含关键字的密文集所用的时间,具有稳定性。

1　相关知识

1.1　国密算法

(1)SM3 杂凑算法[10]首先需确定初始值以及常量 T,随后定义布尔函数为:

$$FF_j(X,Y,Z)=\begin{cases}X\oplus Y\oplus Z, & 0\leqslant j\leqslant 15\\ (X\cap Y)\cup(X\cap Z)\cup(Y\cap Z), & 16\leqslant j\leqslant 63\end{cases}$$

$$GG_jf(x)=\begin{cases}X\oplus Y\oplus Z, & 0\leqslant j\leqslant 15\\ (X\cap Y)\cup(\neg X\cap Z), & 16\leqslant j\leqslant 63\end{cases}$$

置换函数定义为:

$$P_0(X)=X\oplus(X<<<9)\oplus(X<<<17)$$

$$P_1(X)=X\oplus(X<<<15)\oplus(X<<<23)$$

然后对长度为 $l(l<2^{64})$ 比特的消息 m 进行填充操作,填充后的消息 m' 的比特长度为 512 的倍数。最后将消息进行迭代压缩,生成杂凑值,长度为 256 比特。

(2)SM4 分组密码算法[11]是一个迭代分组密码算法,由加解密算法和密钥扩展算法组成。首先生成密钥及密钥参量,加密密钥长度为 128 位。轮密钥由加密密钥生成,表示为:

$$(\mathrm{rk}_0,\mathrm{rk}_1,\cdots,\mathrm{rk}_{31})$$

其中,$\mathrm{rk}_i(i=0,1,\cdots,31)$ 为 32 比特。

二是加密算法,输入明文 (X_0,X_1,X_2,X_3) 经历 32 次迭代运算和一次反序变换 R 组成,分为四组加密,最

后输出密文 (Y_0, Y_1, Y_2, Y_3)。

迭代过程为:

$$X_{i+4} = F(X_i, X_{i+1}, X_{i+2}, X_{i+3}, \mathrm{rk}_i) = X_i \oplus T(X_{i+1} \oplus X_{i+2} \oplus X_{i+3} \oplus \mathrm{rk}_i)$$

而且该解密算法结构与加密变换相同,不同的只是轮密钥的使用顺序,与之相反,所以 SM4 分组算法的加解密效率对比 AES 算法更高。

1.2 属性加密

定义 1 访问结构(Access Structure):设定集合 $\{P_1, P_2, \cdots, P_n\}$ 为实体集,假设 $\forall B, C$,有 $B \in A$ 且 $B \subseteq C$,那么一定有 $C \in A$。从而确定 $A \subseteq 2^P$ 集合是单调的,其中属于 A 的集合被称为授权集合。

定义 2 访问树:设定 γ 为访问树,x 为节点,n_x 为节点 x 的子节点数量,k_x 为节点 x 的阈值。树中的每个非叶子节点称为门限,由子节点和 k_x 表示。当门限为“或”门时,有 $0 < k_x \leqslant n_x, k_x = 1$;当门限为“与”门时,有 $0 < k_x \leqslant n_x, k_x = n_x$。

设定 r 为 γ 的根节点,γ_x 为访问树中以 x 为根的子树。满足访问树:$\gamma_x(\tau) = 1$,代表属性集合 τ 满足访问树 γ_x。

1.3 智能合约

智能合约(Smart Contract)是一套以数字形式定义的承诺,包括合约参与方可以在上面执行这些承诺的协议。

由于区块链的特性,所有操作在以太坊中都是透明和可靠的,这意味着理论上可以使用以太坊智能合约来执行任何计算任务。所以利用部署的合约函数自动执行的特点以及区块链公开透明的性质可以达成系统内的公平性。

2 系统模型与定义

2.1 系统模型

所述方案的系统模型包含五个部分,分别是授权中心(Authorization Center)。数据拥有者(Data Owner)。云服务器(Cloud Service Provider)。智能合约(Smart Contract)以及数据使用者(Data User),如图 1 所示。

AC:授权中心负责管理系统中各个属性,根据用户的属性为其生成属性密钥。

DO:数据拥有者使用 SM4 分组加密算法加密共享数据,生成关键词索引并将其上传至云服务器。

CSP:云服务器是诚实且好奇的半可信实体,负责执行关键词检索操作以及密文存储。

SC:智能合约是提前部署在系统内部,自动执行合约内容,主要负责收取或发放押金。传递陷门。传递用户所需的密文集合以及验证。

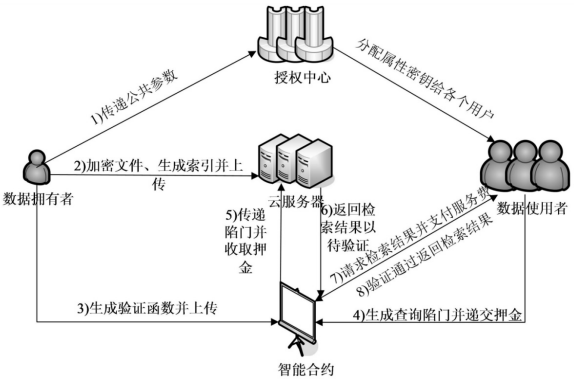

图 1 系统模型

DU:用户通过生成查询陷门,再传送给云服务器以查询相关数据。用户可以从 AC 处获得属性密钥,如果用户的属性密钥满足密文的访问结构,则可以获取解密密钥。

所用的符号以及函数定义如表 1 所示。

表 1 系统参数

参数	含义
D	明文数据集
C	密文数据集
I	安全索引
T_ω	查询陷门函数
C_ω	包含关键字 ω 的密文数据集
A	访问结构
Transport( )	传输待验证集至本地合约账户的函数
UserCost( )	收取用户押金的合约函数
ServiceCost( )	收取云服务器押金的合约函数
Verify( )	验证密文并传递货币的合约函数

2.2 算法定义

所述方案结合国密算法 SM3 和 SM4、属性加密、可搜索加密方案[12]以及智能合约,提出了一种多项式概率算法。即 Π = (Setup, Enc, Trapdoor, Search, SCVerify, Des)。

(1) $\mathrm{Setup}(1^\lambda) \rightarrow (\mathrm{PK}, s)$。输入安全参数 λ,输出系统公共参数 PK 以及主密钥 s。

$\mathrm{KeyGen}(s, A) \rightarrow \mathrm{Priv}_A$。执行用户密钥生成算法,授权中心根据访问结构 A 和主密钥 s 生成用户的私钥 Priv_A。

(2) $\mathrm{Enc}(\mathrm{PK}, D, \omega, \tau, K_1) \rightarrow (C, V(C_i))$。输入系统公共参数 PK,明文 D,经过属性 τ 加密的关键词 ω 以及对称密钥 K_1。输出密文 C 以及待验证值集 $V(C_i)$。

(3) $\mathrm{TrapDoor}(\omega, \mathrm{Priv}_A, \mathrm{UserCost}) \rightarrow T_\omega$。输入关键词集合 ω、用户的属性私钥 Priv_A 以及合约函数 UserDeposit( ),输出查询陷门 T_ω。

(4) $\mathrm{Search}(I, T_\omega, \mathrm{PK}, \mathrm{ServiceCost}) \rightarrow (C_\omega,$

$p(C_\omega)$)。输入安全索引 I、查询陷门 T_ω、公共参数 PK 以及合约函数 ServiceCost( ),输出检索结果 C_ω 以及证明值集合 $p(C_\omega)$ 返回至智能合约。CSP 还需验证密文中的关键词是否满足陷门中的搜索策略以及检查 DU 的属性集 τ 是否满足密文中的访问策略。

(5) $\text{SCVerify}(C_\omega, V(C_i), p(C_\omega)) \rightarrow m$。输入检索的密文结果 C_ω、用户支付检索服务器费 $ Fee、待验证函数 $V(C_i)$ 以及证据集函数 $p(C_\omega)$,智能合约进行验证:

$$m = \begin{cases} 0, V(C_i) \neq p(C_\omega) \\ 1, V(C_i) = p(C_\omega) \\ \varnothing \end{cases}$$

智能合约输出 $m = 1$,代表验证成功;如果输出为 0,则代表验证失败,根据部署的合约函数回退费用;如果输出 $\varnothing$,代表用户并未与智能合约交互请求检索结果,没收押金给予云服务器,以此实现用户-云服务器间的公平交易。

(6) $\text{Des}(C_\omega, \text{Priv}_A) \rightarrow D_\omega$。输入 C_ω 以及用户属性私钥 Priv_A,输出解密数据 D_ω 至用户。

2.3　安全定义

根据文献[3]的自适应语义安全提出以下安全定义。所述方案在挑战者与敌手之间进行攻击游戏,敌手可以遍历检索陷门和返回的密文结果,继而进行查询攻击。根据分析攻击结果来验证方案的安全性。详细定义如下:

(1)查询模式 (α):给定关键词集合 $(\omega_1, \omega_2, \cdots, \omega_n)$,如果 $\omega_i = \omega_j$,则 $\alpha(i,j) = 1$,否则 $\alpha(i,j) = 0$

(2)访问模式 (A_P):给定包含 N 个查询陷门的访问模式集合为:

$\{A_P(T_1) = D(\omega_1), \cdots, A_P(T_N) = D(\omega_N)\}$

(3)遍历记录 (H):设定 $H = (D, W)$ 为查询的历史记录,D 是明文数据,W 是 N 个关键字集合 $\{\omega_1, \omega_2, \cdots, \omega_n\}$。

(4)轨迹 $\eta(H)$:定义为

$\{(\text{ID}(C_1), \cdots, \text{ID}(C_n)), (|C_1|, \cdots, |C_N|), \alpha, A_P(H)\}$。$\text{ID}(C_i)$ 是密文 C_i 的地址,$|C_i|$ 是 C_i 的大小。

(5)视图 $v(H)$:定义为

$\{(\text{ID}(C_1), \cdots, \text{ID}(C_N)), T, C, I\}$

其中 T 是由 H 生成的查询陷门。

验证实验 $\text{Test}^{A_P,N}$ 过程如下:

(1)系统建立,挑战者执行 Setup( ),输出公共参数 PK 以及主私钥 s,发送公共参数给敌手。

(2)挑战者执行 KeyGen 算法生成密钥组,验证其不可区分性。

(3)挑战者随机抽取参数 $M \in \{0,1\}$,输入密钥以及明文,执行 Enc 算法,将生成的索引 I_M 和密文 C_M 传递给敌手。

(4)敌手根据接收到的 I_M 和 C_M 执行查询攻击,如果敌手输出的 $M' = M$,那么结果返回 1,否则为 0。

定义 3:设 $\xi = (\text{Setup}, \text{Enc}, \text{Des}, \text{TrapDoor}, \text{Search})$ 是一个可搜索加密方案,$\text{Test}^{A_P,N}$ 为密文模型安全实验游戏。如果攻击游戏中对于每一个概率多项式敌手都存在一个可忽略函数 $\Im(n)$,且满足:

$$|\Pr[\text{Test}^{A_P,N} = 1] \leqslant \frac{1}{2} + \Im(n)|$$

那么可证所述方案是满足自适应语义安全的。

2.4　公平性协议

所述方案根据智能合约可在没有第三方的前提下自动执行合约内容的特点,再结合区块链公开透明的特性提出了用户-云服务器间的公平性协议。该协议包含四个合约函数,以此达成用户-云服务器间的公平交易。

首先,智能合约需要接收如下数据:(1)数据拥有者根据密文集 C 生成的待验证集 $V(C_i)$;(2)智能合约接收云服务器的搜索结果 C_ω 和证明集 $p(C_\omega)$,因此,编写合约函数 Transport( ) 接收数据以待验证,保障双方期望。

然后智能合约需要分别收取用户以及云服务器的押金;收取用户押金,并接收用户生成的查询陷门至智能合约存储,因此编写合约函数 UserCost( ) 收取用户押金为了防止用户提前终止服务;收取云服务器押金,并发送查询陷门发送给云服务器,因此编写合约函数 ServiceCost( ) 收取云服务器押金为了防止出现恶意云服务器。

最后,智能合约收取用户服务费 $ Fee,执行验证操作:

$$m = \begin{cases} 0, V(C_i) \neq p(C_\omega) \\ 1, V(C_i) = p(C_\omega) \\ \varnothing \end{cases}$$

当智能合约输出 $m = 1$,验证成功,根据得到的 $m = 1$,合约函数自动执行,传递搜索结果和用户押金至用户,传递服务费 $ Fee 以及云服务器押金至云服务器。

当智能合约输出 $m = 0$,验证失败,说明用户没有得到期待的数据条目,为保障公平性,根据 $b = 0$,合约函数自动执行,返还服务费 $ Fee 以及用户押金,并将云服务器押金补偿给用户。

当智能合约输出 $\varnothing$,说明用户并未请求交互,给予服务费 $ Fee,部署的函数自动执行,为补偿云服务器损失,返还两者押金至云服务器,满足文献[7]中定义的四级公平性,即系统内能提供公平性,系统通过补

偿受害方的损失恢复公平性,编写此合约函数为 Verify( ) 。

3 方案构造

结合属性密码、可搜索加密、国密算法以及智能合约,提出了一种基于国密算法的多用户公平可搜索加密方案。通过以下六个步骤构成:

(1)初始化阶段。

$\text{Setup}(1^{\lambda}) \rightarrow (\text{PK}, s)$

输入安全参数 λ ,设 $(G_1, +), (G_2, +), (G_T, \bullet)$ 是三个循环群, G_1, G_2, G_T 的阶均为素数 N , g 是 G_1 的生成元, $g_2 \in G_2$。当存在随机数 $y, z \in [1, N-1]$,有同态映射 $\psi: g^z = g_0, g^y = g_1$。双线性对满足 $e: G_1 \times G_2 \rightarrow G_T$。双线性对选取性能更好的 R-ate 对[13]。从 SM3 杂凑算法选取 $H_1: \{0,1\}^{\lambda} \rightarrow G_1$; $H_2: \{0,1\}^{*} \rightarrow G_2$ 伪随机函数 $F: \{0,1\}^{K} \times \{0,1\}^{*} \rightarrow \{0,1\}^{l}$。令 τ 为属性集合(包含 n 个元素),每一个属性对应 Z_p^* 中的一个元素,输出系统公共参数 $\text{PK} = \{G_1, G_2, G_T, e, g_1, g_2, g_0, \tau, H_1, H_2, F\}$ 。

用户密钥生成阶段: $\text{KeyGen}(s, A) \rightarrow \text{Priv}_A$,授权中心根据系统主密钥 s 以及访问结构 A 生成用户的属性私钥 Priv_A 。具体过程如下:

首先为访问树 γ 中的每个节点 x 选取一个随机概率多项式 q_x ;由于从根节点 r 开始选取,给定 q_x 的阶为 d_x ,且该阶数与当前节点的阈值 k_x 存在特定关系: $d_x = k_x - 1$;对于根节点 r 还存在 $q_r(0) = y$,后续 q_r 随机选取 d_r 。除根节点外的其他节点 x 也存在关系: $q_x(0) = q_{\text{parent}(x)}(\text{index}(x))$,其 d_x 仍随机选取。 $q_{\text{parent}(x)}$ 表示访问树 γ 中节点 x 的父节点。由此可以确定密钥,其中 $h(X)$ 为 n 阶多项式:

$\text{Priv}_x = (D_x, R_x)$

$D_x = g_2^{q_x(0)} \cdot T(i)^{r_x}, T(i) = g_2^{X^n} g^{h(X)}, i = \text{att}(x)$

$R_x = g^{r_x}$

(2)加密阶段。

$\text{Enc}(\text{PK}, D, \omega, \tau, K_1) \rightarrow (C, V(C_i))$

首先选取随机数 $j \in Z_p$ 计算 t ,随后输入公共参数 PK 。明文 D 以及对称密钥 K_1 使用 SM4 分组加密算法,并用属性集 τ 标记,得到密文如下:

$t = e(H_1(\omega), g_0^j)$

$C = (\tau, C' = H_2(t) \cdot e(g_1, g_2)^j, C'' = g^j, \{C_i = T(i)^j\}_{i \in \tau})$

然后 DO 调用 SM3 杂凑算法,输入系统主私钥 s 对密文集合的每一个密文都进行处理生成对应的消息验证码集合:

$\text{Mac}_{C_i} = H_1(s, C_i), i = (1, 2, \cdots, n)$

最后 DO 对每一个标识符进行处理,存放到一个待验证集合内: $\text{Mac}_{C_i} \rightarrow V(C_i)$ 。算法如下所示:

Algorihm1 $V( )$

Input: C

Output: $V(C)$

1 Select Keys for KGC //从密钥生成中心选取密钥

2 for each $i = 0; n$ do

3 Mac $C_i \leftarrow H_1(C_i)$ //为每一个密文生成唯一标识符

4 end for

5 Client initializes $V( ) \leftarrow \{\}$ //初始化集合

6 for each $i = 0; n$ do

7 Mac$C_i \rightarrow V(C_i)$ //存放标识符至集合内

8 end for

9 Client sends $V(C_i)$ to Smart //发送至智能合约

该集合通过合约函数传递给智能合约本地账户,每次传递的信息都会被区块记录,无法篡改,等待在验证阶段和该密文所对应的证据集进行对比。

索引生成 $\text{IndexGen}(\text{Priv}_A, \omega, \text{PK}) \rightarrow I$ 。

为了生成索引 I ,DO 首先从明文数据集 D 中提取关键字集合 $W = \{\omega_1, \omega_2, \cdots, \omega_n\}$ 。对每一个关键字 $\omega_i \in W$,建立一个数组 $\text{DB}(\omega_i)$ 。该数组如果第 j 个文档包含关键词 ω_i ,那么 $\text{DB}(\omega_i)[j] = 1$;否则 $\text{DB}(\omega_i)[j] = 0$。接下来计算:

$I = (I_1, I_2, I_3)$

$I_1 = e(g_0, g_1)^s \text{Priv}_A$, $I_2 = g^s$, $I_3 = H_1(\omega)^s$

最后数据拥有者输出:

$I = (e(g_0, g_1)^s \text{Priv}_A, g^s, H_1(\omega)^s)$

数据拥有者安全上传密文集 C 、索引 I 至云服务器,调用合约函数 Transport( ) 上传 $V(C)$ 至智能合约中内部账户以待后续验证。

(3)陷门函数生成阶段。

$\text{TrapDoor}(\omega, \text{Priv}_x, \text{UserCost}( )) \rightarrow T_\omega$

查询陷门由用户生成。输入关键字集合 ω 、节点 x 的私钥 Priv_x 计算得到陷门函数 T_ω 。首先选取随机数 $z \in [1, N-1]$,计算得到:

$T_\omega = [H_1^z(\omega), D_x, R_x]$

用户发送查询请求时,合约函数接收 T_ω ,并支付押金给智能合约,该合约函数命名为 $\text{UserCost}(T_\omega)$,是为防止用户中途终止检索协议。

(4)检索阶段。

$\text{Search}(I, T_\omega, \text{ServiceCost}) \rightarrow (C_\omega, p(C_\omega))$

搜索操作由云服务器完成。云服务器在搜索阶段也向智能合约支付押金,从而得到由智能合约发送的陷门函数。该合约函数命名为 ServiceCost( ) 。云服务器验证密文属性是否满足访问策略:只需

$$C' / \frac{e(D_x, C'')}{e(R_x, C_i)} = \frac{H_2(t) \cdot e(g_1, g_2)^j}{e(g_1, g_2)^j} = H_2(t)$$

成立,则执行搜索请求。

输入陷门 T_ω 以及索引 I,得到密文检索结果 $C_\omega = \{C_{\omega_1}, C_{\omega_2}, \cdots, C_{\omega_j}\}$。对 C_ω 执行计算得到:

$\mathrm{Mac}_{C_i} \rightarrow p(C_{\omega_i}), i = (1,2,\cdots,j)$

算法如下所示:

Algorihm2 $p(\ )$

Input: T_ω, I, C

Output: $p(C)$

1 Server computes $C_\omega \leftarrow \mathrm{Search}(C, I, T_\omega)$ //计算得到检索结果

2 Server Select Keys for KGC //从密钥生成中心选取密钥 2

3 for each $i = 0:n$ do

4 Mac $C_\omega \leftarrow H_l(C_\omega)$ //计算检索结果中的每一位密文标识符

5 end for

6 Serverinitializes $pf \leftarrow \{\}$ //初始化集合

7 for each $i = 0:n$ do

8 Mac $C_{\omega i} \rightarrow p(C_{\omega i})$ //存放标识符至集合内

9 end for

10 Server sends$p(C)$ to Smart //发送至智能合约

命名为证明集合,最后将两者返回到智能合约以待验证。

(5)智能合约验证阶段。

$\mathrm{SCverify}(C_\omega, V(C_i), p(C_\omega)) \rightarrow m$

云服务器完成搜索阶段后,将检索结果返回至智能合约。然后,用户与智能合约交互请求检索结果,用户将服务费 $ Fee 暂存于智能合约。此刻,由于智能合约调用合约函数 Transport( ),得到待验证集 $V(C_i)$。需要验证的密文集合 C_ω 以及证明集 $p(C_\omega)$。执行验证算法 $\mathrm{Verify}(C) \rightarrow m$。

输出的 m 值如下:

$$m = \begin{cases} 0, V(C_i) \neq p(C_\omega) \\ 1, V(C_i) = p(C_\omega) \\ \varnothing \end{cases}$$

如果验证结果 $m = 1$,验证通过,代表返回给用户的密文文档集合正确,传递正确检索结果至用户。智能合约将服务费与服务器的押金转移至服务器的账户,将用户的押金与检索结果返回给用户;若不正确,智能合约将两者的押金转移至用户的账户;若用户未执行交互请求,智能合约通过 Rdeposit( )将两者的押金转移至服务器的账户。以此实现用户-服务器的公平交易环境。

算法如下所示。

Algorithm 3 Verify( )

Input: $V(C_i), p(C_{\omega i})$, $ Fee

Output: b

1 if msg. value< $ Fee //判断是否收到交互请求

2 then return $m = 0$

3 else send $ Fee to Smart

4 for $i = 0:n$ do

5 if $V(C_i) = p(C_{\omega i})$ //验证密文对应的标识符

6 then return $m = 1$

7 else return $m = 0$

8 end if

9 end for

10 end if

(6)用户解密阶段。

$\mathrm{Des}(C_\omega, d_A) \rightarrow D_\omega$

数据使用者 DU 发送交互请求后,得到由智能合约返回的对应密文文档 $C_\omega = \{C_1, C_2, \cdots, C_n\}$,再输入用户私钥 d_A,执行解密算法:

$D_\omega = \mathrm{Des}_{d_A}(C_\omega)(1 \leqslant \omega \leqslant n)$

得到明文文档 $D_\omega = \{D_1, D_2, \cdots, D_n\}$。

4　方案分析

4.1　正确性分析

计算得到:

$$C' / \frac{e(D_x, C'')}{e(R_x, C_i)} = H_2(e(T_\omega, C'')) = H_2(t)$$

输入陷门以及 C'' 计算:

$$H_2(e(T_\omega, C'')) = H_2(e(H_1^z(\omega), g^j)) = H_2(e(H_1(\omega), g^{jz})) = H_2(e(H_1(\omega), g_0^j))$$

又因为 $t = e(H_1(\omega), g_0^j)$,所以可得:

$$C' / \frac{e(D_x, C'')}{e(R_x, C_i)} = H_2(e(T_\omega, C'')) = H_2(t)$$

由此说明陷门与密文属性对应相同的关键词,满足访问树,是正确的。

4.2　安全性分析

定理 1:假设 F 是伪随机函数。H_1 是抗碰撞散列函数以及 SM4 分组算法也是抗选择明文攻击安全的,那么由安全定义可知,本方案是满足自适应语义安全的。

证明:设 B 为挑战者,A 为概率多项式敌手。A 通过获取密钥。索引以及密文集合推测出 B 的视图以赢得游戏。因此,要证明方案是满足自适应安全的,只需满足:

$$\left| \Pr[\mathrm{Test}^{A_r, N} = 1] \leqslant 1/2 + \Im(n) \right|$$

敌手给 B 发送两个同等大小的文档集合,B 生成模拟密钥 priv_A^*。模拟索引 I^* 和模拟密文集合 C^* 及证明过程如下:

(1) B 运行 KeyGen 算法生成用户密钥 Priv_A,而密钥生成过程是在 $\{0,1\}^\lambda$ 随机选取的,故敌手从两个

随机字符串中区分密钥的概率是可忽略的,因此,存在一个可忽略函数 $\Im_1(n)$ 满足:

$$|\Pr[\mathrm{KeyGen}(s,A)\rightarrow \mathrm{Priv}_A]-\Pr[\mathrm{Random}(s,A)\rightarrow(\mathrm{Priv}_{A_r})]|\leqslant \Im_1(n)$$

(2) B 执行索引生成算法 $\mathrm{IndexGen}(\mathrm{Priv}_A,\omega)\rightarrow I$,索引生成过程采用伪随机函数 F,模拟 I^* 使用随机字符串 $\{0,1\}^{\lambda}$ 取代相同长度输出的 $F_{K_2}(\omega_i)$。由伪随机函数的安全性可得,敌手 A 无法在不知道 Priv_A 的前提下区分伪随机函数 F 和随机字符串,因此,存在一个可忽略函数 $\Im_2(n)$ 满足:

$$|\Pr[\mathrm{IndexGen}(\mathrm{Priv}_A,\omega)\rightarrow I]-\Pr[\mathrm{Random}(\mathrm{Priv}_A,\omega)\rightarrow(I_r)]|\leqslant \Im_2(n)$$

(3)因为 SM4 加密算法已知是密文模型安全的,所以模拟密文 C^* 和实际密文 C 是不可区分的,因此存在一个可忽略的函数 $\Im_3(n)$ 满足:

$$|\Pr[\mathrm{Enc}(K_1,D)\rightarrow C]-\Pr[\mathrm{Random}(K_1,D)\rightarrow(C_r)]|\leqslant \Im_3(n)$$

综上所述,可以计算得出:

$$|\Pr[\mathrm{Test}^{A,N}=1]\leqslant 1/2+\Im_1(n)+\Im_2(n)+\Im_3(n)|$$

也就是满足:

$$|\Pr[\mathrm{Test}^{A,N}=1]\leqslant 1/2+\Im(n)$$

因此可知,给定任意敌手 A,模拟输出与实际输出是不可区分的,那么猜测出 B 的视图概率也是不大于 1/2 的,所以,本方案是满足自适应性语义安全的。

4.3 功能分析

本节通过分析对比文献[12]、文献[14]的方案以及文中方案,得到的结果如表 2 所示。

文献[12]实现了大型数据库上的动态可搜索加密方案,支持多关键词查询且通过遗忘交叉标记验证结果,不能保证方案的公平性;文献[14]利用 CP-ABE 实现特定用户群组的访问,然后根据区块链公平公正可溯源的特性,解决了恶意云服务器返回错误检索结果或者不返回检索结果的问题,但检索效率仍存在一定优化空间。

该方案采用的国密 SM3 杂凑算法以及 SM4 分组算法,在加解密方面有着更快的速度,生成的杂凑值也是 256 比特,更加安全,并利用更高效的 R-ate 双线性对。再结合属性加密,通过访问策略来实现多用户场景下的密文检索;其次,引入智能合约,通过其自动执行合约函数的特性,确保服务器和用户的公平交易环境;最后,通过验证对比,确保检索结果的正确性以及安全性。

表 2 各方案功能对比

方案	查询关键字类型	多用户查询	智能合约	验证结果	交易的公平性
文献[12]方案	支持多关键词查询	×	×	√	×
文献[14]方案	支持单关键词查询	√	√	√	×
文中方案	支持单关键词查询	√	√	√	√

5 实验分析

所述方案采用 Enron Email Dataset 作为实验数据集,在安全索引生成阶段以及关键词检索阶段进行仿真实验。明文加解密使用 SM4 分组算法,杂凑算法使用 SM3 算法,并且通过属性加密实现多用户场景下的密文检索,双线性对选取效率更高的 R-ate 对,合约函数编写通过 Solidity 语言完成。运行环境为 Windows 系统,Intel(R) Core(TM) i7-8570H,8G 内存。区块链环境通过 Ganache 模拟,合约函数部署在 Remix 平台上,Ganache 初始化生成多个虚拟合约地址分别作为数据拥有者。云服务器以及用户的地址,每个账户拥有 100 ETH(虚拟货币)。

5.1 性能分析

所述方案将根据各个阶段计算量开销进行性能分析,具体符号含义如表 3 所示。

依据索引生成阶段计算开销。陷门生成阶段计算开销以及检索阶段计算开销三个方面进行分析对比。根据双线性对的特性,R-ate 对运算效率大于传统双线性对,即计算开销上 $T_{rp}<T_p$;根据文献[10]可知,SM3 杂凑算法的效率以及安全性优于传统 SHA-256 算法,即计算开销上 $T_{sh}<T_h$。如表 4 所示,文中方案对比其他方案在索引生成阶段。陷门生成阶段以及检索阶段均有一定优势,计算开销综合最小。

表 3 计算开销符号介绍

计算名称	含义
T_p	一次对称的双线性对运算时间
T_{rp}	一次 R-ate 双线性对运算时间
T_e	一次指数运算的时间
T_m	一次乘法运算的时间
T_{sh}	一次 SM3 杂凑算法运算时间
T_h	一次 SHA-256 哈希运算时间

表 4　各阶段计算开销对比

方案	生成索引阶段	陷门生成阶段	检索阶段
文献[15]	T_p +3 T_e + T_m + T_h	3 T_e +2 T_m + T_h	3 T_p + T_m +2 T_e
文献[9]	2 T_p +3 T_e + T_h	4 T_e +3 T_m + T_h	2 T_p + T_m +2 T_e
文中方案	T_{rp} +3 T_e + T_m + T_{sh}	3 T_e + T_m + T_{sh}	2 T_{rp} + T_m +2 T_e

5.2　实验结果

所述方案通过对比 Du 的方案[15]以及 Yan[16]的方案安全索引生成耗时;文中索引生成时间与关键词数量成正比。随机输入 Enron 数据集五个,对应数据集内可提取的关键词数量分别为 25,149,212,525,667,生成索引所耗时长为 72.00 ms,125.99 ms,142.99 ms,240.00 ms,294.00 ms。如图 2 所示,所述方案生成索引耗时处于毫秒级,较现有方案,具有一定时间开销上的优势。

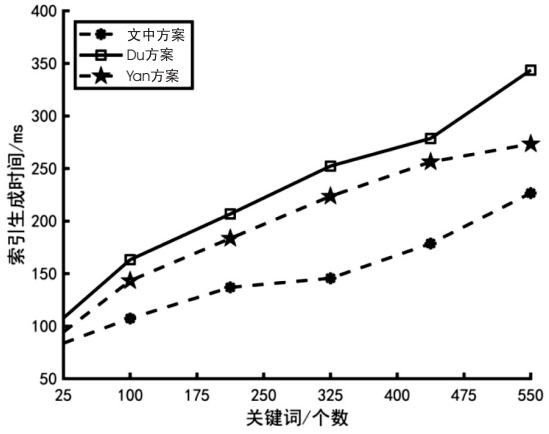

图 2　索引生成耗时

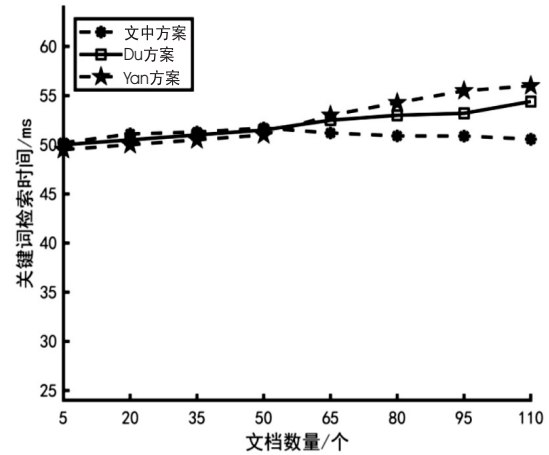

图 3　检索所耗时长

所述方案对比现有方案 Du 的方案[15]以及 Yan[16]的方案,结果如图 3 所示。所述方案以用户检索单个关键词“equitable”为例,得到结果:不同的文档数量下检索所耗时间在 50 ms ~ 51 ms 间,并非线性增长关系;具体如下:分别是文本数量为 5 时,检索时间为 50.2 ms;文本数量为 10 时,检索时间为 51.4 ms;文本数量为 20 时,检索时间为 51.1 ms;文本数量为 40 时,检索时间为 50.9 ms;文本数量为 50 时,检索时间为 50.7 ms;文本数量为 100 时,检索时间为 51.3 ms。由此可见,所述方案中邮件文本的数量增多对方案检索耗时影响不大,具有实用性;在实际应用场景中可以减缓数据库存储过多文本对检索时长的影响。

6　结束语

结合国密算法、属性加密以及智能合约技术,提出了一种支持多用户的公平国密可搜索加密方案,利用属性私钥满足访问结构以获取解密密钥的特性,实现支持多用户的可搜索加密方案,并完成了密钥生成、加解密效率、索引生成以及检索效率上的优化。同时,结合智能合约交易可追踪且不可逆转的特性,实现检索结果的可验证。并且其自动执行合约内容的特点可使双方诚实地按照合约规则执行,以此确保用户与云服务器之间的交易公平性。下一步将考虑实现连接关键词检索,以提高方案的实用性。

参考文献:

[1] SONG D, WAGNER D, PERRIG A, et al. Practical techniques for searches on encrypted data[C]//Proceeding 2000 IEEE symposium on security and privacy. Piscataway:IEEE, 2000:44-55.

[2] BONEH D, DI CRESCENZO G, OSTROVSKY R, et al. Public key encryption with keyword search[C]//International conference on the theory and applications of cryptographic techniques. Interlaken:Springer,2004:506-522.

[3] CURTMOLA R, GARAY J A, KAMARA S, et al. Searchable symmetric encryption:improved definitions and efficient constructions[J]. Journal of Computer Security, 2011, 19(5):895-934.

[4] WANG C, REN K, YU S, et al. Achieving usable and privacy-assured similarity search over outsourced cloud data[C]//2012 Proceedings IEEE INFOCOM. Los Alamitos: IEEE,2012:451-459.

[5] BETHENCOURT J, SAHAI A, WATERS B. Ciphertext-policy attribute-based encryption[C]//2007 IEEE symposium on security and privacy. Washington:IEEE,2007:321-334.

[6] 李　双,徐茂智. 基于属性的可搜索加密方案[J]. 计算机学报,2014,37(5):1017-1024.

[7] PAGNIA H, VOGT H, GÄRTNER F C. Fair exchange[J]. Computer Journal,2003,46(1):55-75.

(下转第 113 页)