计算机系统应用 ISSN 1003-3254, CODEN CSAOBN
Computer Systems & Applications,2022,31(10):108−115 [doi: 10.15888/j.cnki.csa.008629]

E-mail: csa@iscas.ac.cn
http://www.c-s-a.org.cn
Tel: +86-10-62661041

基于 ElGamal 的同态云端密文存储检索方案①

邵　航, 李子臣, 王东飞

(北京印刷学院 信息工程学院, 北京 102600)
通信作者: 李子臣, E-mail: mir_soh@163.com

摘　要: 云端数据存储的安全性和检索效率是网络空间安全亟待解决的问题之一. 本文提出了一个新的密文检索模型, 并在此基础上利用 ElGamal 同态密码算法和 SM4 分组密码算法, 设计了一种基于混合同态加密的云端密文存储检索方案. 首先, 该检索方案能够在数据上传、检索和下载的过程中, 保证数据的安全, 可用于个人云端 U 盘等应用场景. 其次, 对该方案的正确性和安全性进行分析. 最后, 通过实验的方式对方案的正确性进行了证明. 实验结果表明该方案在保证数据安全的情况下, 检索结果正确, 效率高.

关键词: 个人云盘; 同态加密; ElGamal; SM4; 云端密文检索; 云存储; 隐私保护

引用格式: 邵航,李子臣,王东飞.基于 ElGamal 的同态云端密文存储检索方案.计算机系统应用,2022,31(10):108–115. http://www.c-s-a.org.cn/1003-3254/8629.html

Homomorphic Cloud Ciphertext Storage and Retrieval Scheme Based on ElGamal

SHAO Hang, LI Zi-Chen, WANG Dong-Fei

(School of Information Engineering, Beijing Institute of Graphic Communication, Beijing 102600, China)

Abstract: The security and efficiency of cloud data storage are urgent issues to be solved in cyberspace security. Therefore, a new ciphertext retrieval model is proposed in the study, and on this basis, the ElGamal homomorphic cipher algorithm and SM4 block cipher algorithm are used to design a cloud ciphertext storage and retrieval solution based on hybrid homomorphic encryption. The retrieval solution can ensure data security during data uploading, retrieving, and downloading and can be applied to personal cloud USB drives and other application scenarios. Moreover, the correctness and safety of the scheme are analyzed and proved through experiments. The experimental results reveal that the scheme can assure correct retrieval results with high efficiency while ensuring data security.

Key words: personal cloud disk; homomorphic encryption; ElGamal; SM4; cloud ciphertext retrieval; cloud storage; privacy protection

云存储和云服务器技术的迅猛发展, 将数据搬上云, 已是大势所趋, 这样不仅可以使存储容量实现动态扩容, 方便及时应对如“双 11”购物节等用户数据激增的情况, 还可以降低初创公司前期的投入成本, 实现只需按需按量采购云服务器. 同时, 惠于当前各云厂商推出的个人免费云盘服务, 对于普通用户只需注册就可以使用, 这极大地方便了人们的日常生活. 然而, 数据存储在第三方云端, 无论是对于企业用户, 还是对于个人用户, 安全问题[1] 始终困扰着他们.

与传统存储相比, 目前的云存储的数据都被放在云端, 由云服务器统一计算管理. 但云端存放的数据就可能处在一种不安全的状态[2], 一方面, 有的企业会将全部数据上云, 而这些数据中可能会有很多的秘密信息, 例如一些企业隐私和用户信息等; 另一方面, 用户不

① 基金项目: 国家自然科学基金 (61370188); 北京市教委科研计划 (KM202010015009, KM202110015004); 北京印刷学院博士启动金 (27170120003/020); 北京印刷学院科研创新团队项目 (Eb202101); 北京印刷学院校内学科建设项目 (21090121021); 北京印刷学院重点教改项目 (22150121033/009); 北京印刷学院科研基础研究一般项目 (Ec202201)
收稿时间: 2021-11-07; 修改时间: 2021-12-02, 2021-12-10; 采用时间: 2021-12-21; csa 在线出版时间: 2022-07-15

可能会完全信任提供云存储和云计算的服务商. 无论是企业还是个人用户在使用云存储时都会担心数据的安全性, 所以对于用户不信任和数据安全问题急需解决.

为了保证数据安全, 我们可以先将数据加密后再上传到云端, 但当存储的数据越来越多时, 对于数据的检索又成为一大问题. 传统方式是先将数据解密后再检索, 但这样的检索效率非常低, 无法满足实际需求. 所以我们需要设计出一种无需解密就能检索的方案, 而同态加密可以实现密文间的计算[3,4], 所以使用同态加密技术, 这样既能保证数据的安全性也能提升检索效率. 同态加密的概念是由 Riverst 等人[5] 于 1978 年首次提出, 同年 Riverst 等人[6] 又提出了基于大整数分解难题的 RSA 公钥加密算法, 该算法具有乘法同态性; 1999 年, Paillier 提出了基于合数阶剩余类的 Paillier 公钥密码算法[7], 该算法具有加法同态性和乘法同态性; 2009 年 Gentry 构造出首个全同态加密方案[8], 该方案基于理想格, 之后 Gentry 等人又在 2010 年和 2013 年分别提出 DGHV 方案[9] 和 GSW13 方案[10], 前者基于近似最大公因子问题 (approximate greatest common divisor, AGCD)[11], 后者基于 LWE (learning with error) 问题; 2012 年以后, 文献 [12,13] 中提出 BGV12 方案和 Bra12 方案, 前者通过模交换和密钥交换技术实现无需 bootstrapping 就能建立层次型同态加密方案 (Leveled-FHE) 方案, 后者无需使用模交换就可建立 Leveled-FHE 方案. 但全同态加密算法的效率很低[14], IBM 在其开源库 HElib 中尝试使用了 BGV 方案设计了一个基于全同态加密的密文检索实验[15], 实验结果表明, 目前将全同态方案直接用于密文检索会大大限制检索效率, 实用性较弱. 此外国内外学者也做出了很多研究, 文献 [16] 提出一个基于 LWE 和 AGCD 问题的新型密文同态加密方式, 根据逐个计算密文相似度, 进而排序选出相似度最大者即为检索结果, 但其中密文排序增加了云端计算量; 文献 [17] 利用加法同态性提出一种在密态数据库上的可搜索加密方案; 文献 [18] 以整数向量加密技术为基础, 通过建立向量空间模型, 进而在密文下计算检索向量与文件向量的相似度进行检索, 但在建立空间向量模型和计算相似度时会增加计算量; 文献 [19] 给出了一种基于改进的同态加密算法的全文密文检索方案, 但也需要排序、查找后才能达到检索的目的;文献 [20] 提出了一个基于新型同态密文检索方案 CRSHE, 但同样需要通过排序反映文档与关键词之间的相关性去实现检索. 从上面可以看出同态加密技术在云存储中实现密文检索大有可为, 但大都需要一些额外的计算, 例如排序、查找、计算相似度等, 增加系统开销.

而本文针数据存储的安全和密文检索的高效需求, 首先设计一个新的密文检索模型, 在此基础上提出一种混合加密技术, 即使用成熟的 ElGamal 算法和安全的国密 SM4 算法设计了一种高效的云端同态密文检索方案, 并给出了该方案的具体流程. 其次, 通过理论证明和实验仿真的方式分析了方案的正确性与安全性. 最后, 对实验数据进行分析, 实验数据表明, 在保证检索结果正确的前提下, 能有效提高检索效率.

1 预备知识

1.1 检索模型

在该方案设计的检索模型如图 1 所示, 主要参与方: 用户、云服务器、可信第三方. 方案的主要功能分为用户录入数据 (1) 和用户检索数据 (2–7).

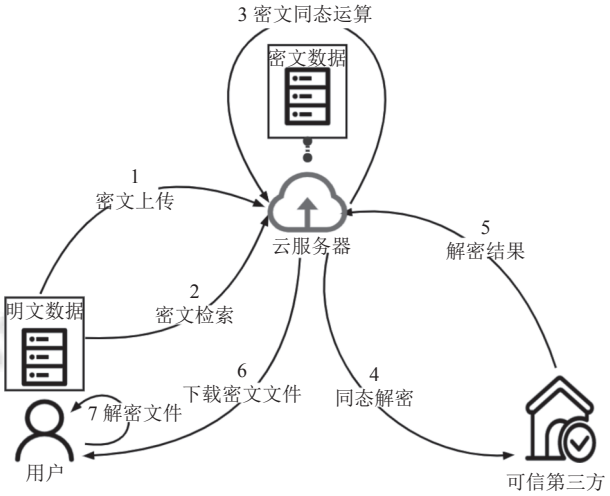

图 1 密文检索系统模型

(1) 用户

用户首先将加密数据上传至云服务器, 当需要检索时, 上传检索密文关键词, 下载所需加密文档, 然后解密得到所需文件.

(2) 云服务器

云服务器作为存储和计算数据的平台, 在密文数据中检索计算, 将计算结果发送给可信第三方, 解密得到检索序号, 根据检索序号返回最终的加密文档给用户.

(3) 可信第三方

可信第三方首先产生同态加密的公私钥, 并公布公钥; 然后将云服务器发送过来的密文通过私钥解密, 筛选出检索序号, 并发送给云服务器.

1.2 ElGamal 同态加密算法

ElGamal 算法[21] 是国际上公认的公钥密码体制, 其加密算法是基于 Diffile-Hellman 密钥交换算法[22], 是由 Taher ElGamal 在 1985 年提出, 其安全性基于计算有限域上的离散对数难题, 相比于 RSA 算法, ElGamal 算法能抵抗重放攻击. 该算法由参数设置、密钥生成、加密、解密和同态乘法 5 部分组成:

(1) 参数设置

随机选择一个大素数p, 构造一个模p的有限域Z_p, g是Z_p上的生成元, 且$g \in Z_p$.

(2) 密钥生成

随机选取$X \in [1, p-1]$, 计算$Y = g^X \bmod p$, 私钥$SK = \{X\}$, 公钥$PK = \{p, g, Y\}$.

(3) 加密

发送者对于明文消息m, 随机生成一个秘密数$k \in [1, p-1]$, 使用公钥对明文消息加密得到密文: $E(m) = \{\gamma = g^k \bmod p, \beta = mY^k \bmod p\}$, 其中, $E(\cdot)$表示加密算法.

(4) 解密

接收者收到密文消息$\{c_1, c_2\}$后, 利用私钥解密得到明文$D(E(m)) = \beta(\gamma^X)^{-1} \bmod p$, 其中, $D(\cdot)$表示解密算法.

(5) 同态乘法

若对于两个明文消息 m_1, m_2, 加密后的密文分别为: $E(m_1) = \{\gamma_1 = g^{k_1} \bmod p, \beta_1 = m_1 Y^{k_1} \bmod p\}$和$E(m_2) = \{\gamma_2 = g^{k_2} \bmod p, \beta_2 = m_2 Y^{k_2} \bmod p\}$, 则$E(m_1)E(m_2) = \{\gamma_1\gamma_2, \beta_1\beta_2\} = \left\{g^{k_1+k_2} \bmod p, m_1 m_2 Y^{k_1+k_2} \bmod p\right\}$, 且$D(E(m_1)E(m_2)) = m_1 m_2 Y^{k_1+k_2}\left[\left(g^{k_1+k_2}\right)^X\right]^{-1} \bmod p = m_1 m_2$, 因此 ElGamal 算法具有乘法同态性.

2 方案设计

2.1 整体方案设计

如图 2 所示, 是该方案中云端密文检索系统的整体框架, 同态计算和求逆在云服务器中进行实现.

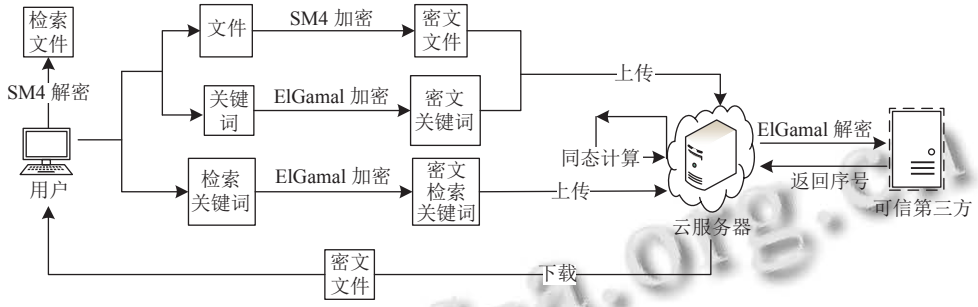

图 2 密文检索系统整体框架

(1) 初始化

可信第三方生成 ElGamal 的公私钥对, 并将公钥公开; 用户在客户端生成 SM4 算法的密钥.

(2) 录入数据

用户使用同态公钥将关键词加密, 使用 SM4 密钥将文件内容加密, 然后将加密后的关键词和加密后的文档一起上传至云服务器存储, 即录入数据.

(3) 检索数据

用户使用同态公钥加密检索关键词, 再向云服务器提交检索请求, 即检索数据.

(4) 同态计算

云服务器接收到检索请求后, 首先将密文检索关键词先求逆, 然后逐个与密文关键词相乘, 最后将结计算结果发送给可信第三方; 可信第三方收到后使用同态私钥进行解密, 返回给云服务器一个结果; 云服务器根据结果, 返回给用户相应的检索结果.

(5) 解密数据

用户在客户端收到云服务器发送的加密文件后, 用 SM4 密钥解密, 得到检索关键词所对应的明文文件.

2.2 系统具体结构

下面具体介绍整个方案的流程结构, 整个方案主要可以分成两部分, 第 1 部分是录入数据, 第 2 部分是检索数据, 且每个角色都有不同的功能, 图 3 是方案检索成功的详细序列图.

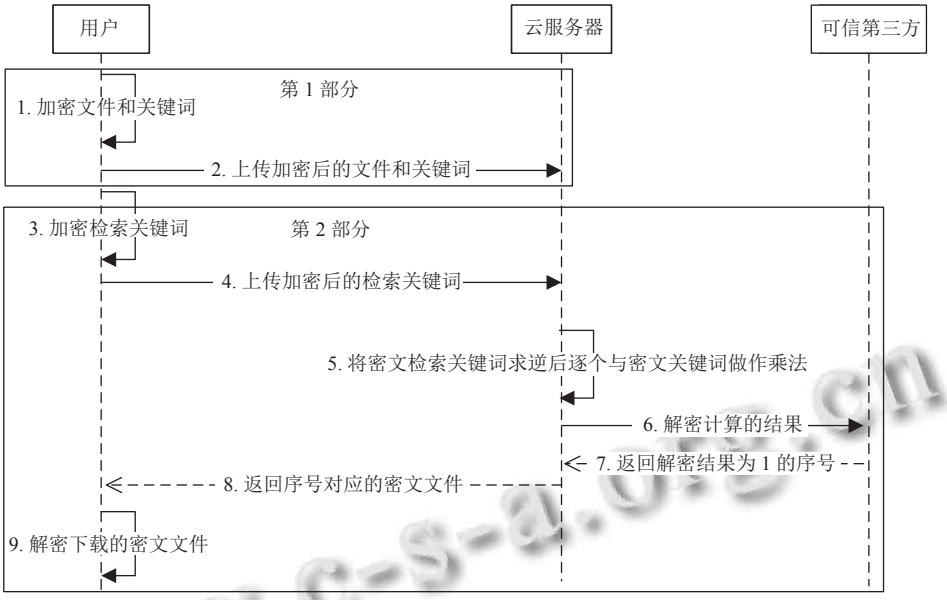

图 3 方案序列图

下面分别详细介绍这两部分.

(1) 用户录入数据

用户待上传 n 个明文文件 ($1 \leqslant i \leqslant n$), 如表 1 所示.

表 1 待上传的明文数据

关键词1	…	关键词j	…	关键词m	原文件
M_{11}	…	M_{1j}	…	M_{1m}	F_1
⋮	⋱	⋮	⋱	⋮	⋮
M_{i1}	…	M_{ij}	…	M_{im}	F_i
⋮	⋱	⋮	⋱	⋮	⋮
M_{n1}	…	M_{nj}	…	M_{nm}	F_n

本方案支持多关键词检索, 设关键词个数为 m 个 ($1 \leqslant j \leqslant n$), 但关键词数量增加会增加同态计算的次数, 引起时间复杂度的增加, 因此在保证检索的效果和减少时间开销的前提下, 我们应当控制关键词数量, 所以在下面的实验中, 我们取 m=2.

用户生成的密钥有: 用于 ElGamal 算法加解密公私钥对{PK, SK}, 以及 SM4 分组密码算法的密钥{K}.

该方案采用的混合加密, 关键词用 ElGamal 算法加密, 文件内容使用国密 SM4 算法加密, 然后合并一起上传服务器. 每个用户拥有自己上传文件的密钥, 可信第三方拥有所有加密关键词的密钥, 具体如表 2 所示.

表 2 角色和密钥分配

角色	公钥PK	私钥SK	密钥K
用户	√	—	√
可信第三方	√	√	—

用户上传文件流程如图 4 所示.

1) 可信第三方生成同态加密公私钥

随机取一个较大的素数 p, 构造一个模 p 的有限域Z_p, g 是Z_p中的生成元, 随机取$X \in Z_p$, 计算出$Y = g^X \bmod p$, 得到同态加密的公私钥对$\{SK = \{X\}, PK = \{p, g, Y\}\}$, 且公布公钥 PK.

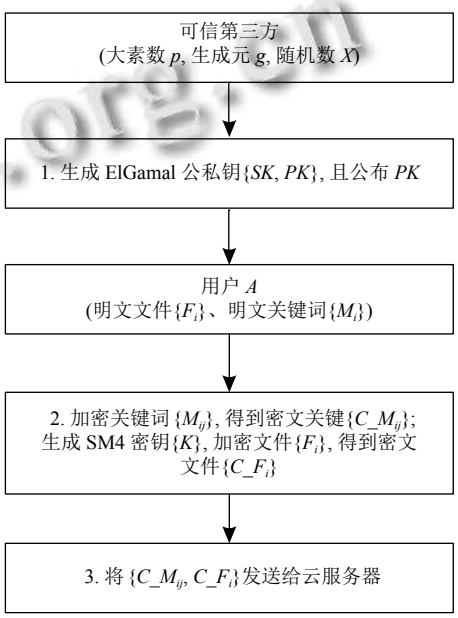

图 4 文件上传

2) 用户生成对称加密的密钥并加密数据

在客户端取随机数$k \in Z_p$, 使用公钥 PK 对关键词

M_{ij}加密, 计算得$\{\gamma_{ij} = g^k \bmod p, \beta_{ij} = M_{ij}Y^k \bmod p\}$, 生成对应的密文关键词$C_M_{ij}$. 其中文件关键词$M_{ij}$在加密前需要通过 Unicode 编码为十六进制字符串并转为整数形式; 在客户端生成 128 位的随机数作为 SM4 密钥 K 并保存在本地, 并使用密钥 K 对文件加密得到密文文件C_F_i.

3) 用户上传密文数据.

将C_M_{ij}和C_F_i拼接一起发送给云服务器存储. 此时云服务器中的存储内容如表 3 所示.

表 3　云端中的密文数据

关键词1	…	关键词j	…	关键词m	文件
C_M_{11}	…	C_M_{1j}	…	C_M_{1m}	C_F_1
⋮	⋱	⋮	⋱	⋮	⋮
C_M_{i1}	…	C_M_{ij}	…	C_M_{im}	C_F_i
⋮	⋱	⋮	⋱	⋮	⋮
C_M_{n1}	…	C_M_{nj}	…	C_M_{nm}	C_F_n

(2) 用户检索数据

用户检索数据的流程如图 5 所示, 该部分包含 5 个阶段.

1) 加密检索关键词

假设密文数据库中的密文文件有 n 个, 用户先将检索关键词 Q 通过 Unicode 编码为十六进制字符串, 并转为整数形式得到Q^*, 这时使用之前生成的公钥 PK 对Q^*进行同态加密, 计算得到$\{\gamma_Q = g^k \bmod p, \beta_Q = (Q^*)Y^k \bmod p\}$,即生成检索关键词的密文$C_Q^* = \{\gamma_Q, \beta_Q\}$.

2) 同态计算

云服务器收到密文检索关键词C_Q^*后, 求逆得到$(C_Q^*)^{-1}$, 然后逐个与密文关键词$C_M_{ij} = \{\gamma_{ij}, \beta_{ij}\}$做同态乘法得到$(C_Q)^*{}_{ij} = \{c1_{ij}, c2_{ij}\}$, 具体运算如式 (1):

$$(C\_Q)^*{}_{ij} = C\_M_{ij} \times (C\_Q^*)^{-1} = \{c1_{ij}, c2_{ij}\} = \{\gamma_{ij} \times \gamma_Q, \beta_{ij} \times \beta_Q\} \tag{1}$$

3) 可信第三方解密

云服务器将同态计算的结果$(C_Q)^*{}_{ij}$发送给可信第三方, 可信第三方使用私钥 SK 将每个$(C_Q)^*{}_{ij}$解密为Q^*, 具体运算如式 (2). 若存在Q^*为 1, 则返回$s = i$, $(1 \leqslant s \leqslant n)$; 若不存在, 则返回 $s = -1$, 表示未检索到结果.

$$Q^*{}_{ij} = D((C\_Q)^*{}_{ij}) = \delta_{ij}(\gamma_{ij}{}^X)^{-1} \bmod p \tag{2}$$

4) 云服务器返回检索结果

云服务器根据收到可信第三方返回的解密结果来判断是否将密文文件发送给用户: 若返回的是一个非 0 的结果s $(1 \leqslant s \leqslant n)$, 则返回序号为 s 所对应的密文文件C_F_s; 若返回值为 0, 则表示未检索到结果, 向用户发送“查询失败”的信息.

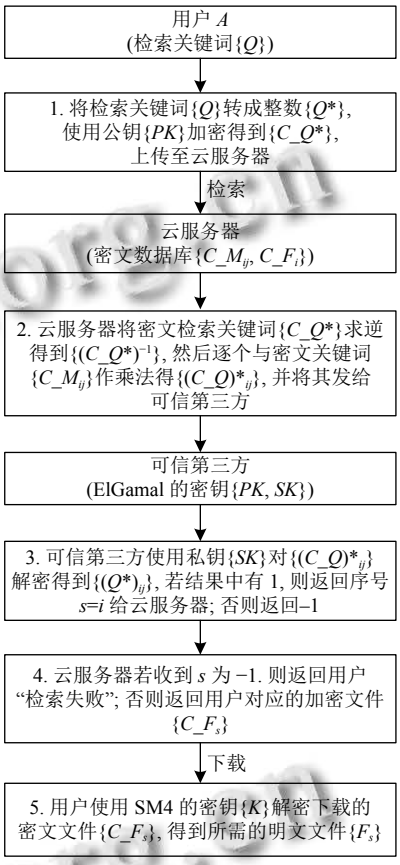

图 5　用户检索文件

5) 用户解密

用户从云服务器下载到密文文件C_F_s $(1 \leqslant s \leqslant n)$, 利用 SM4 的密钥 K 对文件解密, 得到最终检索关键词所对应的原文件F_s.

3　系统实现与安全性分析

3.1　具体实现

本节通过一个具体的案例来验证本方案, 加密原文件可以是图书内容, 图书数量 n=4, 关键词个数 m=2, 关键词 $M1$ 和 $M2$ 分别是书名和作者, 实验环境为 Intel(R) Core i5-6200U @ 2.30 GHz 双核 16 GB 内存, Microsoft Visual Studio Community 2019.

(1) 假设用户有待加密上传的文件如表 4, 以明文形式展示.

表 4 明文数据

序号	书名	作者	文件内容
1	围城	钱钟书	F_1
2	史记	司马迁	F_2
3	狂人日记	鲁迅	F_3
4	三体	刘慈欣	F_4

上面的明文数据使用混合加密, 即关键词{书名、作者}使用 ElGamal 算法加密, 文件内容使用 SM4 算法加密.

(2) 将关键词用 Unicode 编码为十六进制字符串, 并转为整数形式, 见表 5.

将关键词 (整数形式) 使用 ElGamal 加密, 参数设置: (p,g)(40049372667947663039, 11743527543478996065), (X,Y)=(14450894889756208713, 25589559154825890551) 将文件内容使用 SM4 加密, 然后上传云服务器, 密文数据库如表 6.

表 5 关键词

序号	关键词1 (书名)		关键词2 (作者)	
	十六进制	整数	十六进制	整数
1	56F457CE	1458853838	94B1949F4E66	163490423590502
2	53F28BB0	1408404400	53F89A6C8FC1	92327207800769
3	72C24EBA65E58BB0	8269258428286078000	9C818FC5	2625736645
4	4E094F53	1309232979	521861486B23	90264664828707

表 6 密文数据

序号	书名$\{c_1, c_2\}$	作者$\{c_1, c_2\}$	文件内容
1	{3211599817415563915 3, 28842211535672190855}	{2869979953114052186, 24648727176285130096}	$E(F_1)$
2	{23951723667595728740, 6348022646948245992}	{39774446219191511186, 38862678895494170626}	$E(F_2)$
3	{20223093317074115372, 21621911918880284578}	{20757153970644744369, 26076132915728676331}	$E(F_3)$
4	{26192419111552605170, 66611204033859945448}	{19447094721901965537, 22844493954254118518}	$E(F_4)$

(3) 若用户输入检索关键词 Q=“史记”, 先用 Unicode 编码为十六进制字符串, 并转为整数形式Q^*, 然后使用 ElGamal 加密$(Q^*)^{-1}$得到检索关键词的逆元密文$C_Q^* = \{\gamma_Q, \delta_Q\}$, 结果如表 7 所示.

表 7 检索关键词

检索关键词	十六进制	十进制	检索关键词的密文$(\{c_1, c_2\})$
“史记”	53F28BB0	1408404400	{24914191202926788728, 21280176959620948015}

(4) 云服务器收到检索关键词的密文C_Q^*后, 将密文检索关键词求逆得$(C_Q^*)^{-1}$= {37747856122936643469, 13243315324064048566}, 然后逐个与密文数据库中的密文关键词C_M_{ij}做乘法得到$(C_Q)^*_{ij}(i, j \in Z,\ 1 \leqslant i \leqslant 4$且$1 \leqslant j \leqslant 2)$, 如表 8 所示.

表 8 同态乘法运算密文相乘结果$\left(C_M_{ij} \times (C_Q^*)^{-1}\right)$

序号	关键词1	关键词2
1	{11216929236222963946, 23128607310697629708}	{33593146621890140288, 37977864468152489955}
2	{18283956446679924477, 24618066340401802334}	{1366231225410332531, 29051288589689848885}
3	{33896692461687507920, 4109244469603721049}	{34946155032641737178, 22886193455501072227}
4	{10896494947855593771, 3808468191905602827}	{25731605953598699893, 1282281649647820192}

云服务器将计算的结果发送给可信第三方, 可信第三方收到消息后, 使用私钥 SK 逐个解密得到Q^*_{ij} $(i, j \in Z, 1 \leqslant i \leqslant 4$且$1 \leqslant j \leqslant 2)$, 如表 9. 这里的$Q^*_{21} = 1$, 所以将 s=2 返回给服务器.

表 9 第三方解密结果$(C_Q)^*_{ij}$的解密结果

序号	关键词1	关键词2
1	7401699183511153602	7502181278316145252
2	1	26605886965884883259
3	4887862233434361770	33231225570961283692
4	3067354324817914607	23143437231061775197

(5) 云服务器收到可信第三方发来的 s=2, 将$E(F_2)$发送给用户, 即用户从云服务器下载到$E(F_2)$, 最后使用 SM4 密钥解密得到原文件F_2.

3.2 安全性分析

在该方案中, 用户首先要将加密后的文件和关键词上传至云端, 然后从云端检索出关键词对应的加密文件, 解密即可得到检索的文件, 所以本方案的安全性可分为数据存储安全性和检索模型安全性.

(1) 数据存储安全性

关键词是采用 ElGamal 算法加密的, 相比于 RSA 算法, ElGamal 算法能抵抗重放攻击, 另外根据计算有限域上的离散对数困难, 攻击者很难根据公钥 PK 去计

算或推导出私钥 *SK*, 这就使得用户在密文检索过程中, 攻击者就算得到公钥 *PK*, 也不能作为云服务器和可信第三方之间的中间者去解密云服务器发送的同态计算结果, 这就保证了用户在检索过程中检索数据不可篡改.

再者就是文件采用的是国密 SM4 分组算法. 加解密过程均由用户在客户端完成, 云服务器无法获知其密钥 *K*. SM4 保证了文件的安全性[23], 可以抵抗穷举攻击、差分攻击、线性攻击等攻击手段, 具有较高的安全性, 使得攻击者即使获得加密文件, 也无法作为用户和云服务器之间的中间者解密出原文件.

(2) 检索模型安全性

密文检索过程满足乘法同态性. 方案中同态加密的公私钥均由可信第三方生成, 用户将关键词和文件加密上传至云端, 都是以密文形式存储. 用户将加密的检索关键词上传至云端, 利用同态加密的性质, 将加密的检索关键词与云端中存储的密文关键词做乘法同态运算, 再利用可信第三方解密来求出检索号, 以此完成密文检索. 由于整个过程均是在密文下进行的, 所以说云端是无法获知任何有关密钥和明文数据的, 且只有用户才能获得明文数据, 这就保证了检索过程中的数据是安全的, 所以说检索模型是安全的.

3.3 性能分析

(1) 效率分析

在本方案中的密文检索只使用了乘法同态, 也就是只用部分同态来实现, 当然也可以使用全同态来实现密文检索, 但目前全同态效率比较低, 难以广泛使用. 以下面的例子为例, 来证明本文使用部分同态比全同态效率更高. 如表 10, 加密 1 000 数字 1 000 次, 取 10 次试验的平均时间; 将 1 000 和 2 000 的密文相乘, 取 10 次试验的平均耗时, 明显可以看出使用 ElGamal 在加密和乘法同态运算上速度更快. 另外表 11 展示与其他方案的对比, 可以看出本方案更加轻量高效 这里 BGV 算法和 CKKS 算法的测试程序采用的是 IBM 的开源库 fhe-toolkit-linux[15]; BFV 算法的测试程序采用的是微软的开源库 SEAL[24].

表 10 测试时间 (ms)

运算	BGV	BFV	CKKS	ElGamal (本文)
加密	705.642	722.216	48.9107	**1.046**
乘法	372.587	5.875	0.37967	**0.01000**

(2) 精确度分析

本方案针对个人用户就是在云端构建单用户的密文数据, 进而在云端进行安全检索, 因为一个文件可以有多个关键词, 所以用户在检索时, 既可以实现单个关键词检索, 也可以实现多关键词检索.

单个关键词检索时, 检索结果只有两种: 找到文件或者是未检索到; 多关键词检索时, 若输入的是一个文件对应的多个关键词, 那么只要有一个匹配上, 则检索成功, 若输入的是多个文件对用的关键词, 则返回多个匹配上的文件, 进而实现多文件检索, 这样效率会更加高, 其精确度和效率远高于逐个关键词检索.

表 11 方案对比

文献	检索方法
[16]	计算密文相似度、排序
[17]	构建密态数据库
[18]	构建空间向量模型, 计算密文相似度
[19]	相似度排序、查找
[20]	相似度排序
本文	**乘法同态, 判断为1, 返回序号**

4 总结与展望

本文利用同态加密的性质, 设计出新的密文检索模型, 再结合安全的国密算法, 提出了一个基于同态加密的云端密文存储检索方案. 该方案能够在保证数据安全的前提下进行数据检索, 即检索过程中云端无法获知任何有关密钥信息和明文数据. 与其他方案相比, 具有轻量级和高效性, 可以应用于小型个人云端 U 盘的场景, 有较好的实用价值. 经实验数据分析表明, 本文方案检索结果正确、安全性好、效率高、实用性强. 在下一步的研究中, 将尝试设计一种高效的全同态加密算法, 并将其应用在云端密文检索中, 使之具有更好的安全性.

参考文献

1 冯朝胜, 秦志光, 袁丁. 云数据安全存储技术. 计算机学报, 2015, 38(1): 150–163. [doi: 10.3724/SP.J.1016.2015.00150]

2 黄保华, 黄丕荣, 赵伟宏, 等. 云存储中支持属性撤销的多关键词可搜索加密方案. 计算机工程, 2021, 47(11): 29–36. [doi: 10.19678/j.issn.1000-3428.0061050]

3 杨亚涛, 赵阳, 张卷美, 等. 同态密码理论与应用进展. 电子与信息学报, 2021, 43(2): 475–487. [doi: 10.11999/JEIT191019]

4 刘钦菊, 路献辉, 李杰, 等. 全同态加密自举技术的研究现状及发展趋势. 密码学报, 2021, 8(5): 795–807. [doi: 10.

13868/j.cnki.jcr.000477]
5 Rivest RL, Adleman L, Dertouzos ML. On data banks and privacy homomorphisms. Foundations of Secure Computation, 1978, 4(11): 169–179.
6 Rivest RL, Shamir A, Adleman L. A method for obtaining digital signatures and public-key cryptosystems. Communications of the ACM, 1978, 21(2): 120–126. [doi: 10.1145/359340.359342]
7 Paillier P. Public-key cryptosystems based on composite degree residuosity classes. International Conference on the Theory and Applications of Cryptographic Techniques. Berlin: Springer, 1999. 223–238. [doi: 10.1007/3-540-48910-X_16]
8 Gentry C. Fully homomorphic encryption using ideal lattices. Proceedings of the 41st Annual ACM Symposium on Theory of Computing. Bethesda: ACM, 2009. 169–178.
9 van Dijk M, Gentry C, Halevi S, *et al*. Fully homomorphic encryption over the integers. International Conference on the Theory and Applications of Cryptographic Techniques. Monaco and Nice: Springer, 2010. 24–43.
10 Gentry C, Sahai A, Waters B. Homomorphic encryption from learning with errors: Conceptually-simpler, asymptotically-faster, attribute-based. Annual Cryptology Conference. Santa Barbara: Springer, 2013. 75–92.
11 Howgrave-Graham N. Approximate integer common divisors. International Cryptography and Lattices Conference. Providence: Springer, 2001. 51–66.
12 Kiran K. (Leveled) Fully homomorphic encryption without bootstrapping. Computing Reviews, 2015, 56(10): 613–613.
13 Brakerski Z. Fully homomorphic encryption without modulus switching from classical GapSVP. Advances in Cryptology—CRYPTO 2012. Berlin: Springer, 2012.
14 陈智罡, 宋新霞, 郑梦策, 等. 全同态加密文献计量分析研究. 计算机工程与应用, 2022, 58(4): 40–51. [doi: 10.3778/j.issn.1002-8331.2107-0038]
15 IBM. FHE-toolkit-linux, 2020. https://github.com/IBM/fhe-toolkit-linux. [2021-10-01].
16 刘家森, 王绪安, 王涵, 等. 云服务器中基于同态加密的关键词检索方案. 科学技术与工程, 2021, 21(8): 3180–3185. [doi: 10.3969/j.issn.1671-1815.2021.08.029]
17 孙僖泽, 周福才, 李宇溪, 等. 基于可搜索加密机制的数据库加密方案. 计算机学报, 2021, 44(4): 806–819. [doi: 10.11897/SP.J.1016.2021.00806]
18 韩邦, 李子臣, 汤永利. 基于同态加密的全文检索方案设计与实现. 计算机工程与应用, 2020, 56(21): 103–107. [doi: 10.3778/j.issn.1002-8331.1909-0049]
19 程帅, 姚寒冰. 基于同态加密的密文全文检索技术的研究. 计算机科学, 2015, 42(S1): 413–416.
20 付伟, 李墨洮, 赵华容, 等. CRSHE: 基于同态加密的新型密文检索方案. 计算机工程与科学, 2018, 40(9): 1540–1545. [doi: 10.3969/j.issn.1007-130X.2018.09.003]
21 ElGamal T. A public key cryptosystem and a signature scheme based on discrete logarithms. IEEE Transactions on Information Theory, 1985, 31(4): 469–472. [doi: 10.1109/TIT.1985.1057074]
22 Boneh D. The decision Diffie-Hellman problem. International Algorithmic Number Theory Symposium. Portland: Springer, 1998. 48–63.
23 李子臣. 密码学: 基础理论与应用. 北京: 电子工业出版社, 2019: 66–77.
24 Microsoft. Microsoft SEAL, 2020. https://github.com/microSoft/SEAL. [2021-10-01].

(校对责编: 牛欣悦)