实用的云环境下可验证的身份基匿名保序加密

黄 霖 赵运磊
(复旦大学软件学院 上海 210203)
(复旦大学计算机科学技术学院上海市数据科学重点实验室 上海 210203)

摘 要 保序加密可以使用户在密文上直接比较明文的大小,但是目前已有的保序加密方案都不可以验证密文的完整性和正确性。云服务器可能会传送给用户不完整或不可信的密文,但有时数据拥有方不希望公布自己的身份。因此构建云环境下可验证的身份基匿名保序加密。算法提供身份授权,即所有用户可以进行范围查询,而只有特权用户才可以进行获取数据拥有者身份、验证和解密操作;身份基的方式便于可特权访问多个数据库的用户的私钥管理。使用该算法加密数据后,对数据进行的范围查询效率较高,对 32 bit 和 64 bit 数据的比较操作仅需要 0.28 μs 和 0.42 μs。

关键词 保序加密 揭序加密 范围查询 身份验证 身份匿藏 云计算

中图分类号 TP3 **文献标志码** A DOI:10.3969/j.issn.1000-386x.2022.07.044

VERIFIABLE IDENTITY-BASED ANONYMOUS ORDER-PRESERVING ENCRYPTION IN THE PRACTICAL CLOUD ENVIRONMENT

Huang Lin Zhao Yunlei
(*School of Software, Fudan University, Shanghai 210203, China*)
(*Shanghai Key Laboratory of Data Science, School of Computer Science, Fudan University, Shanghai 210203, China*)

Abstract Order-preserving encryption allows users to directly compare the size of data in cipher texts. However, none of the existing order-preserving encryption schemes can verify the integrity and correctness of the cipher text. And the cloud server may transmit incomplete or untrusted cipher text to the user. The data owner sometimes does not want to publish his or her identity. Therefore, this paper provides a verifiable identity-based anonymous order-preserving encryption in the cloud environment. The algorithm ensured that all accessible users could perform range queries, and only privileged users performed identity acquisition, verification, and decryption operations. The identity-based approach facilitated private key management for users who had privileged access to multiple databases. After using this algorithm to encrypt data, the range query efficiency of data is higher, and the comparison operation for 32-bit and 64 bit data only needs 0.28 μs and 0.42 μs.

Keywords Order-preserving encryption (OPE) Order-revealing encryption (ORE) Range query Identity-verification Identity-hiding Cloud computing

0 引 言

随着大数据技术的日益发展,越来越多的信息存储于云数据库中。虽然云计算解决了传统计算方式存在的许多问题,但是新技术也带来了新挑战:一个重要的问题是云数据的管理不完全可信[1-3]。

针对此问题,可以在创建和更新数据时采用支持数据库密文域处理的加密方式,如此在查询时不需进行解密操作,从而防止好奇的管理员偷窥数据或服务

收稿日期:2020-02-22。国家重点研发计划项目(2017YFB0802000);国家自然科学基金项目(61472084);上海创新行动计划项目(16DZ1100200);上海科学技术发展基金项目(16JC1400801);山东省重点研发计划项目(2017CXG0701,2018CXGC0701)。
黄霖,硕士生,主研领域:保序加密,数字签名。**赵运磊**,教授。

器被攻破时数据明文泄露,有效提高云数据库的安全性。在众多支持密文域处理的加密方式中,保序及揭序加密是其中一条重要的分支。其中,揭序加密(Order-Revealing Encryption, ORE)是保序加密的泛化,而保序加密(Order-Preseving Encryption, OPE)是揭序加密的特例。对于范围查询,保序及揭序加密可以使用户在密文上直接比较明文的大小,保护敏感数据不会被云数据库服务器泄露。

保序加密方案的概念和定义由 Agrawal 等[4]提出。Boldyreva 等[5]提出了关于保序加密的严格安全定义理想安全性(IND-OCPA)。IND-OCPA 安全性是挑战者和敌手之间的博弈,通俗来讲,可进行如下描述:敌手准备两个明文序列,这两个序列具有相同的顺序,但数值是随机的。敌手把这两个序列发送给挑战者。挑战者加密一个序列,并将密文发送给敌手,敌手的目标是猜出挑战者加密的是这两个序列中的哪一个;如果敌手猜出正确结果的概率是 50%,则算法具有 IND-OCPA 安全性。Boneh 等[6]受到了密码学混淆器的启发提出了一种新的能够提供比较密文对应明文大小功能的加密方案,即揭序加密。作为保序加密的泛化,揭序加密中对密文所对应明文的比较并不是直接通过比较密文数值的大小来进行,而是借助了公开的比较函数。所有实现 IND-OCPA 安全性的 ORE 方案需要频繁的客户端与服务器的交互[7-10],而非交互式、无状态的 ORE 方案都需要强大的加密原语,例如多线性映射或不可分辨混淆[6,11],在今天无法有效实现。Chenette 等[12]受到可搜索对称加密文献[13-14]中考虑的基于模拟器的定义启发,给出了一个基于模拟器的关于泄露函数 L 的 ORE 安全性定义,在安全和效率之间提供了一个具体的权衡。拥有泄露的 ORE 的安全和 IND-OCPA 都是在仅仅已知密文模型上的安全定义;如果敌手还知道密文集的背景,可能会采取频率统计攻击。Kerschbaum[15]提出了抵抗频率攻击的理想安全性,即 IND-FAOCPA,并提出了一种满足 IND-FAOCPA 安全性的保序加密算法。但现今满足 IND-FAOCPA 安全性的方案都不具有较强实用性。之后的研究[16-17]声称 Kerschbaum 的定义是不精确的,并描述了对他的方案的攻击,Cash 等[18]提出了新的参数隐藏方案。

然而,现有的保序及揭序加密算法缺乏对密文正确性及完整性的验证,缺少对不同身份用户的权限管理,因此本文考虑构造一种具有身份管理功能的顺序揭露签密方法。签密是将数字签名和公钥加密的功能合二为一,既保证了加密内容的完整性和可验证性,又保证了加密消息的私密性,并且比简单地结合签名和加密的效率大为提升。匿签密的概念由 Zhao[19]提出,它可以看作是公钥加密、数字签名和身份隐藏的一种新的整体集成。Wang 等[20]构造了一种基于身份的匿签密。本文在安全性与实用性之间做了权衡,最终提出了一种具有“拥有泄露的 ORE 安全[12]”的便于实现的身份基匿名签密。具体的泄露函数为 $L_f(m_1,m_2,\cdots,m_t)=\{1(m_j<m_t)\parallel ind_{\text{diff}}(m_j,m_t):j\in[t-1]\}$,即算法会泄露明文顺序信息和两个明文的最高不同比特位,其中函数 1 表示当满足输入时,输出为真。

1 系统模型与安全模型

1.1 系统模型

如图 1 所示,数据分享者 A 匿名将自己的可进行范围查询的敏感数据加密并签名,保存在云数据库服务器中。

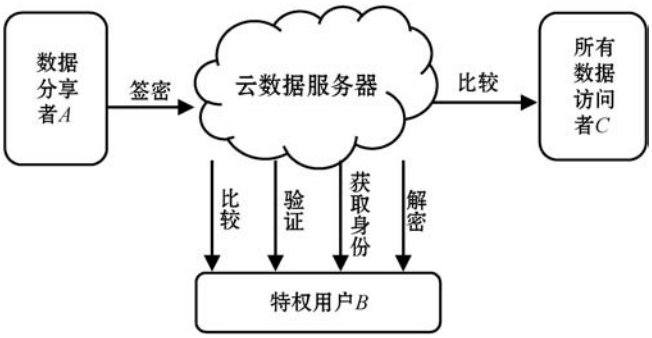

图 1 系统模型与用户权限

所有可以访问这个云服务器的用户 C 都可以直接对密文进行比较(不需要密钥),但没有验证数据正确性、完整性和对密文解密获得明文的权限。

拥有更进一步访问权限的用户 $B\subset C$,除了拥有 C 所拥有的比较权限外,还可以得知数据来自于 A,验证数据正确性和完整性,并且对数据进行解密,得知明文。用户 B 可以是具有权力的政府部门、数据监管者,或者用户 A 指定的其他人,其 ID 固定,可以访问云中不同数据库里的敏感数据。之所以采用身份基签密,而不是传统公钥体系的签密,是因为身份基签密可以一个 ID 对应不同的私钥(不同数据库使用不同的主私钥),而在传统公钥体系中,一个特定的公钥只能生成一个私钥。因而如果 B 拥有多个数据库的特殊访问权限(这与 B 能访问包含敏感信息的明文往往是关联的,这是 B 的身份赋予它的权利),身份基签密更利于私钥管理。

加密签名操作在数据分享者 A 的客户端进行;比较操作在云数据库服务器中进行,在 B 和 C 进行范围

查询时,服务器返回结果给 B 和 C;验证、解密、获取数据拥有者身份都在 B 的客户端进行。

考虑以下场景:A 为某保密单位,公布了一些实验成果或统计结果,其中一些敏感数据使用本文算法加密,公众可以对其进行范围查询,但不能获取其精确的值。C 可以范围查询这些数据,获取例如一组数据最大的几个值,并查看这些值对应的其他信息;而 B 作为特权用户可以获得 A 的身份,并能对它的身份进行验证,还能在范围查询后,获取加密数据的精确数值。A、B、C 与云数据服务器交互的详细流程见图 2 和图 3。

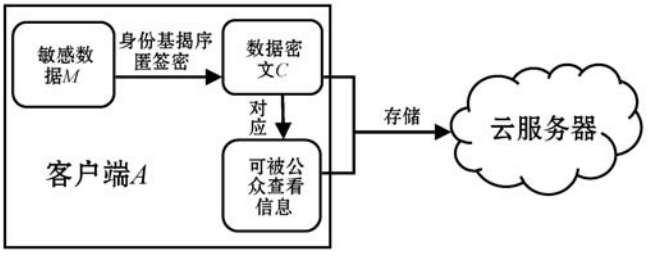

图 2　数据分享者 A 与云服务器交互流程

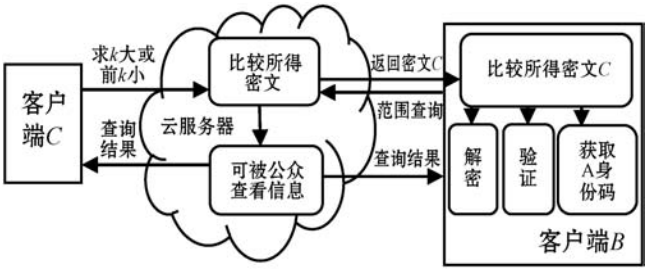

图 3　不同权限数据用户 B 和 C 与云服务器交互流程

数据分享者将敏感数据明文 M 使用身份基揭序匿签密加密签名为密文 C,每个敏感数据密文与表格中可被公众查看相关信息一一对应。之后 A 将敏感数据密文和可被公众查看信息一起匿名存储在云服务器中。

C 向云服务器发出范围查询请求,云服务器对敏感数据密文 C 进行比较,并由筛选出的密文得出它们对应的可被公众查看的相关信息,之后服务器将查询结果(这些相关信息)发送给 C 的客户端。

值得注意的是,在实际的应用中,范围查询分为两类:第一类为求出所有密文中的前 k 大或前 k 小($k \geq 1$)的值,第二类为求出大于或小于某明文值的信息。对于第一类查询,则 C 可以直接进行比较。对于第二类查询,C 需要知道明文所对应密文值才可进行比较,但其不能也不应该知道任一明文所对应密文值,否则其可能发起选择明文攻击,以较快速度获取任意一个密文所对应的明文值,所以应当限定 C 不能进行第二类范围查询。

同样地,具有特殊权限的 B 向云服务器发出范围查询请求时,云服务器也会经历相同的过程,将可被公众查看信息发送给客户端 B。另外,如果 B 提出了请求,云服务器还会将符合范围查询结果的可被公众查看信息也发送给客户端 B。B 可以根据这些密文,在自己的客户端进行解密、验证密文是否真实完整操作,并且能够获取到 A 的身份。在该算法中,B 可求出某明文值对应的密文,并可以进行所有的范围查询;在查询大于或小于某明文值的信息时,B 需要先将此明文值加密为密文再进行问询。

1.2　安全模型

在本方案中,我们假设数据使用者 B 和 C 是经过授权和可信的,所以我们不会考虑数据用户方面的泄露。同时我们认为密钥分发渠道不会引起密钥泄露。此外,我们假设明文分布是稀疏的。但云服务器被认为是诚实但好奇的。换句话说,云服务器会诚实地执行算法流程,但会对存储的数据、查询语句、查询到的数据和其他附加信息的内容感到好奇。

定义 1　拥有泄露的 ORE 的安全(Security of ORE with Leakage)　对于 $q \in \mathbf{N}$,安全参数 $\lambda \in \mathbf{N}$,$A=(A_1,A_2,\cdots,A_q)$ 是一个对手,$S=(S_0,S_1,\cdots,S_q)$ 是一个模拟器,$L(\cdot)$ 是一个泄露函数安全参数,我们称拥有 $L(\cdot)$ 的 $\Pi_{\mathrm{ORE}}=(\mathrm{ORE.\ Setup},\mathrm{ORE.\ Encrypt},\mathrm{ORE.\ Compare})$ 是一个安全的 ORE 方案,如果对于任何多项式规模的对手 $A=(A_1,A_2,\cdots,A_q)$,存在一个多项式规模的模拟器 $S=(S_0,S_1,\cdots,S_q)$,使得表 1 所示实验中的 $REAL_A^{\mathrm{ORE}}(\lambda)$ 和 $SIM_{A,S,L}^{\mathrm{ORE}}(\lambda)$ 的输出在计算上无法区分。

表 1　$REAL_A^{\mathrm{ORE}}(\lambda)$ 和 $SIM_{A,S,L}^{\mathrm{ORE}}(\lambda)$ 实验

$REAL_A^{\mathrm{ORE}}(\lambda)$	$SIM_{A,S,L}^{\mathrm{ORE}}(\lambda)$
1. $sk \leftarrow \mathrm{ORE.\ Setup}(1^\lambda)$	1. $st_S \leftarrow S_0(1^\lambda)$
2. $(m_1,st_A) \leftarrow A_1(1^\lambda)$	2. $(m_1,st_A) \leftarrow A_1(1^\lambda)$
3. $c_1 \leftarrow \mathrm{ORE.\ Encrypt}(sk,m_1)$	3. $(c1,st_S) \leftarrow S_1(st_S,L(m_1))$
4. 对于 $2 \leqslant i \leqslant q$: $(m_i,st_A) \leftarrow A_i(st_A,c_1,c_2,\cdots,c_{i-1})$ $c_i \leftarrow ORE.\ Encrypt(sk,m_i)$	4. 对于 $2 \leqslant i \leqslant q$: $(m_i,st_A) \leftarrow A_i(st_A,c_1,c_2,\cdots,c_{i-1})$ $(ci,st_S) \leftarrow S_i(st_S,L(m_1,m_2,\cdots,m_i))$
5. 输出 $(c_1,c_2,\cdots,c_q)$ 和 st_A	5. 输出 $(c_1,c_2,\cdots,c_q)$ 和 st_A

考虑云服务器知道所有密文集。如上所述,所有实现 IND-OCPA[5] 安全性的非交互式、无状态的 ORE 方案都需要强大的加密原语,在今天无法有效实现,因此作为一个实用的保序加密方案,它的安全需要满足定义 1,即拥有泄露的 ORE 的安全[12],这里的泄露函

数为 $L_f(m_1,m_2,\cdots,m_t)=\{1(m_j<m_t)\parallel ind_{diff}(m_j,m_t):j\in[t-1]\}$。对于定义 1,如果泄露函数为 $L_f(m_1,m_2,\cdots,m_t)=\{1(m_i<m_j):1\leqslant i<j\leqslant t\}$,则这个方案是满足理想安全性 IND-OCPA[2]的。

验证和匿名部分的安全由离散对数[21]、双线性映射[22-25]和成熟的对称加密保证,离散对数满足 DLP 安全,双线性映射满足 DBDH 安全。最终验证和匿名部分满足 DBDH 安全。

定义 2 DLP(Discrete Logarithm Problem) 对于素数 p 阶乘法群 G_1,其生成元为 g,给出一个群中的元素 Y,不存在一个多项式时间内的解法,能够找到一个整数 $x\in\mathbf{Z}_p^*$,使得 $Y=g^x$。

定义 3 DBDH(Decisional Bilinear Diffie-Hellman) 对于素数 p 阶乘法群 G_1,其生成元为 g,对于任意 $(a,b,c)\in\mathbf{Z}_p^*$,给定双线性映射 $\hat{e}$ 不存在一个多项式时间内的解法,使得可以给出 $(g,g^a,g^b,g^c)\in G_1^4$,计算出 $\hat{e}(g,g)^{abc}\in G_T$。

考虑本算法是使用了公钥(身份信息相当于公钥)的保序加密。一般的保序加密不使用公钥加密,因为使用公钥加密模式的保序加密方法容易招致选择明文攻击——敌手可选择明文并得知其加密密文,同时敌手又知道已有密文的顺序,从而可通过不断获取已知明文的密文,与已有密文进行大小对比,进而得知已有密文所对应明文大小。而本算法可以抵抗选择明文攻击。

除此以外,本文还会考虑当敌手已知所有密文集和密文集中每个密文出现的频率时的安全性。我们将计算每个明文出现的频率可以与明文集中其他的多少个明文出现的频率混淆。

2 算法描述

本文所提出的可验证的揭序加密算法包含 ORE_INIT、ORE_GEN、ORE_ENC、ORE_COMP、ORE_DEC(非必须)、ORE_CHECK(非必须)六个步骤。

2.1 ORE_INIT:初始化

1) 输入一个安全参数 k,生成两个 q 阶循环群 G_1 和 G_2,g 为 G_1 生成元,规定一个双线性对:$\hat{e}:G_1\times G_1\rightarrow G_2$,$1_{G_2}$ 为群 G_2 单位元。

2) 私钥生成器(Private Key Generator,PKG)生成一个随机数 $s\in\mathbf{Z}_q^*$ 作为系统主密钥。

3) 选择一个哈希函数 $h:\{0,1\}^*\rightarrow G_1$,一个密钥导出函数 $KDF:\{0,1\}^*\rightarrow\{0,1\}^n$,一个对称加密函数 Enc,一个伪随机函数 $F:\{0,1\}^*\rightarrow\mathbf{Z}_M$。

系统公开参数,记为:$SysPar=\{n,q,g,\hat{e},G_1,G_2,1_{G_2},h,Enc,F,KDF\}$,其中 n 为数据长度安全参数。

2.2 ORE_GEN:私钥生成

对于身份为 ID_A 的数据拥有者 A,PKG 生成私钥 $SK_{ID}^A=h(ID_A)^s$,并分发给 A 秘密保管。

A 愿意将数据库的数据大小信息分享给将要进行范围查询的任何人,并且指定身份为 ID_B 的 B 可以对数据进行正确性完整性验证和解密,PKG 生成私钥 $SK_{ID}^B=h(ID_B)^s$,并分发给 B 秘密保管。

PKG 不负责验证 B 的身份,仅仅负责由用户 ID 生成他的私钥 SK_{ID};若 B 的身份 ID_B 为误,则他不可以正确地解密和验证数据。在实际应用中,为了防止 PKG 在此处遭到 DDoS(分布式拒绝服务)攻击,可采取同网络 IP 短时间内申请生成私钥次数限制等措施。

值得注意的是,这里的 ID_A 和 ID_B 起到了 PK_A 和 PK_B 的作用,但因为需要身份匿藏,所以 ID_A 不公开;若需要验证 B 的身份,则 ID_B 也不公开。

2.3 ORE_ENC:身份基揭序匿签密生成

数据编码部分参考了 Chenette 等[12]提出的方法,并进行修改,使签密满足一定的频率隐藏特性。令 $M\in\{0,1\}^*$ 为明文信息,设 $a_1a_2\cdots a_n$ 为明文 M 的二进制编码形式,对于二元组 (i,M),定义下面的编码:$E(i,M)=(a_1a_2\cdots a_{i-1}\parallel 1^{n-i})$。其中,整数 n 为消息的比特长度,$i\in[n]$。

数据拥有者 A 的加密签名具体步骤如下:

1) 数据拥有者 A 选取一个随机数 $x\in\mathbf{Z}_q^*$,计算 $X=(h(ID_A))^x$,$PS=\hat{e}(SK_A,h(ID_B))^x$。值得注意的是,在同一列数据中(即需要比较的数据),x 值应相同;在更新数据时可以采用之前的 x,或定期更新 x 并更新所有数据。

2) 若 $PS\neq 1_{G_2}$,则计算 $K=KDF(PS)$;否则回到 1),重新选取随机数 x。

3) 若 $M\in\{0,1\}^*$ 最后一位为 0,则对于最后一个 1 后的每一个 0 对应的明文位数 $i\in[n]$,$c_i=F(K,E(i,m))-r$,r 为随机数并且 $r\in\mathbf{Z}_q^*$;若 $c_i\notin\mathbf{Z}_c$ 则重复此步骤重新选取 r;若 $M\in\{0,1\}^*$ 最后一位为 1,则对于最后一个 0 后的每一个 1 对应的明文位数 $i\in[n]$,$c_i=F(K,E(i,m))+r$,r 为随机数并且 $r\in\mathbf{Z}_q^*$;若 $c_i\notin\mathbf{Z}_c$ 则重复此步骤重新选取 r;对于其他情况的每一位 $i\in[n]$,$c_i=F(K,E(i,m))+a_i$。

4) 将密文串连接成为 $C_{pre}=c_1c_2\cdots c_n$,计算 $H=h(C_{pre})$,$C_{head}=Enc_K(H\mid\mid ID_A\mid\mid x)$,$C=C_{head}\parallel C_{pre}$。

对于长度为 n 的二进制字符串,算法复杂度为

$O(n)$,需要 n 次对长度为 n 的二进制字符串进行 $F(\cdot)$ 运算。如需进一步提高加密效率,可引入并行计算。

2.4　ORE_COMP:揭序匿签密比较

1) 从 C 中获取 $C_{\text{pre}}=c_1c_2\cdots c_n$。

2) 对于两个匿签密串 C 和 C',从 $i=1(i\in[n])$ 开始,分别比较 c_i 和 c'_i 大小,直到 c_i 与 c'_i 不相等。若 $c_i>c'_i$,则 $M_i\geqslant M'_i$;若 $c_i<c'_i$,则 $M_i\leqslant M'_i$。

对于长度为 n 的二进制字符串,算法复杂度为 $O(n)$,仅仅需要 n 次数据比较。

2.5　ORE_CHECK:揭序匿签密验证

非必要步骤。在传统的保序/揭序加密体系中,仅仅需要 2.1 节至 2.4 节这四步即可。但需要验证密文正确性和完整性时,可进行此步操作。

1) 数据访问者 B 计算 $PS=\hat{e}(X,SK_B)$。若 $PS\neq 1_{G_2}$,则计算 $K=KDF(PS)$;否则匿签密无效。验证和解密步骤仅需计算一次 K;短时间内一列数据的 K 都相等(在 x 和 X 尚未更新时),因此每次验证一列中多个数据时只需计算一次 K。

2) 从 C 中获取 $C_{\text{pre}}=c_1c_2\cdots c_n$,使用 K 解密 C_{head} 得到 H,ID_A,x。

3) 验证 $H=h(C_{\text{pre}})$,验证 ID_A 合法,并且 $X=(h(ID_A))^x$,则说明匿签密来自于用户 A 并且是完整真实的;否则匿签密 C 无效。

2.6　ORE_DEC:揭序匿签密解密

非必要步骤。一般的保序或揭序加密定义中不包含解密算法,因为明文信息可通过对所有密文的二分查找获得(需要知道加密密钥和加密方法),但如此就需要多轮客户端与服务器端的通信和对密文的比较。本算法可对密文直接进行解密,解密过程如下:

1) 从 C 中获取 $C_{\text{pre}}=c_1c_2\cdots c_n$。

2) 数据访问者 B 计算 $PS=\hat{e}(X,SK_B)$。若 $PS\neq 1_{G_2}$,则计算 $K=KDF(PS)$;否则匿签密无效。验证和解密步骤仅需计算一次 K;一列数据的 K 都相等(在 x 和 X 尚未更新时),因此每次解密一列中多个数据时只需计算一次 K。

3) 已知 $i=1$ 时,$E(1,M)=1^{n-1}$。从 $i=1(i\in[n])$ 开始,计算 $E(i,M)=(a_1a_2\cdots a_{i-1}\parallel 1^{n-i})$,$c'_i=F(K,E(i,m))$,比较 c'_i 与 c_i 大小,逐比特恢复明文。

若 $c'_i=c_i$,则 $M_i=0$;

若 $c'_i=c_i-1$,则 $M_i=1$;

若 $c'_i<c_i-1$,则对于第 i 位及之后的每一位(用 j 表示),$M_j=1$;

若 $c'_i<c_i$,则对于第 i 位及之后的每一位(用 j 表示),$M_j=0$。

算法需要遍历一遍密文,对于明文长度为 n 的二进制字符串,算法复杂度为 $O(n)$,需要 n 次对长度为 n 的二进制字符串进行 $F(\cdot)$ 运算。

3　性能评估

3.1　正确性

(1) 比较的正确性。对于相同的输入,F 的输出是相同的,因此对于相同前缀的比特位,F 的结果是相同的。因而,若两个字符串某 i 位前面的比特位相同,则 F 的结果也是相同的,$F(K,E(i,m))+a_i$ 可比较在前缀相同时第 i 位的大小,依次比较每一位编码的结果即可比较两数的大小。

而结尾为连续的 0 或连续的 1 时,若比较到这些 0 或者这些 1 的位数,则这数一定小于或等于,或者大于或等于另一个数。那么对于这些 0 或 1,在 $F(K,E(i,m))$ 的基础上在密文大小范围内减一个随机数或加一个随机数,若两数不同则不会影响大小比较的结果,若两数相同则会随机输出一个数,这有利于抗频率统计攻击。

(2) 解密验证及获取数据分享方身份的正确性。由双线性映射的双线性性质可得,$PS=\hat{e}(SK_A,h(ID_B))^x=\hat{e}(h(ID_A)^s,h(ID_B))^x=\hat{e}(h(ID_A)^x,h(ID_B)^s)=\hat{e}(X,SK_B)$,因此数据分享方和数据用户最终算得的 K 相同。所以对于数据分享方使用对称加密方法加密的结果,数据用户可以解密得到正确的值。对于解密来说,数据用户可以通过重新进行运算 F 获得对明文编码伪随机后的结果,之后通过与密文的比较来获得明文中的二进制比特位;对于验证来说,数据用户获得的分享方 ID 和 x 以及对整个密文的哈希值 H 都是正确的,从而由离散对数和哈希运算的单向性,以及 DBDH 问题的困难性可以保证验证结果的正确。

此外,普通用户因为没有特权用户的私钥,所以得不到数据分享方加密使用的 K,因此他们只能对数据进行范围查询,而不能执行解密验证以及获取分享方身份的操作。

3.2　安全性

一般的保序加密不使用公钥加密,因为公钥加密容易招致选择明文攻击——敌手可选择明文并得知其加密密文,同时敌手又知道已有密文的顺序,从而可通过不断获取已知明文的密文,与已有密文进行大小对比,进而得知已有密文所对应明文大小。而在本文算法中,加密方在密钥生成时既使用了公钥(ID 信息),

又使用了私钥。在单纯的签密中,假设敌手拥有数据拥有者 A 的 ID_A,选择一个明文 M 试图进行加密从而获取密文 C',但因为其不具有 A 的私钥 SK_A(私钥保存在客户端,云数据库的好奇的敌手并不能获得),所以其不能产生 A 对数据 M 进行加密产生的结果 C,因此其不能够进一步将 C' 与数据库已有密文比较大小,从而不能进行选择明文攻击。

生成私钥 SK 和加密生成 K 的过程中的安全由离散对数问题的困难性和 DBDH 问题的困难性保障,满足定义 2 和定义 3。假设敌手得到了 X,并且知晓 g,但是由 DLP 可知他无法直接由 $\log_g X$ 获得 x,从而篡改验证部分的信息。在生成 K 的过程中,哈希运算将身份码映射到群 G_1,可以表示为 $h(ID_B)=g^{y_B}(y_B\in \mathbf{Z}_q^*)$,$h(ID_A)=g^{y_A}(y_A\in \mathbf{Z}_q^*)$,进而私钥 $sk_A=(h(ID_A))^s=g^{y_A\cdot s}$。假设敌手可以知晓 X、ID_B,获取到 g^A,则由 DBDH 安全性可知其无法在多项式时间内计算出 $PK=e(sk_A,h(ID_B))^x=e(X,sk_B)=e(g,g)^{y_A\cdot y_B\cdot s\cdot x}=BDH(X,h(ID_B),g^s)$ 以及后续的 $K=KDF(PK)$。因此,由 DLP 和 DBDH 问题的困难性可得可验证性和匿名性部分是安全的。

(1) 下面考虑在云服务器已知密文情况下,拥有泄露的 ORE 的密文安全性。因为范围查询不需要密钥,因此好奇的云服务器不会得到密钥。如定义 1,定义 $A=(A_1,A_2,\cdots,A_q)(q=poly(\lambda))$ 是 ORE 安全游戏的有效敌手(λ 为安全参数)。在 $REAL_A^{\mathrm{ORE}}(\lambda)$ 中,一个随机函数 $f\xleftarrow{R}\mathrm{Funs}[([n]\times\{0,1\}^{n-1}),\mathbf{Z}_M]$ 被选择,在 ORE_ENC 中使用。在 F 的 PRF 安全性下,存在模拟器 $S=(S_0,S_1,\cdots,S_q)$,$REAL_A^{\mathrm{ORE}}(\lambda)$ 和 $SIM_{A,S,L_f}^{\mathrm{ORE}}(\lambda)$ 的输出分布在计算上是不可区分的。下面描述模拟器 $S=(S_0,S_1,\cdots,S_q)$。

首先,S_0 初始化空查找表 $L:[q]\times[n]\to\mathbf{Z}_M$,令 $st_S=L$。接下来,对于每一个 $t\in[q]$,当敌手输出一个问询 m_t,输入 $st_S=L$ 和 $L_f(m_1,m_2,\cdots,m_t)$ 并调用模拟算法 S_t。泄露函数 $L_f(m_1,m_2,\cdots,m_t)=\{1(m_j<m_t)\parallel ind_{\mathrm{diff}}(m_j,m_t):j\in[t-1]\}$,其中 $ind_{\mathrm{diff}}(m_j,m_t)$ 是 m_j 和 m_t 第一个不同比特的索引。

设 $(ct_1,ct_2,\cdots,ct_q)$ 为 $REAL_A^{\mathrm{ORE}}(\lambda)$ 中密文输出的联合分布,设 $(\overline{ct}_1,\overline{ct}_2,\cdots,\overline{ct}_q)$ 是模拟器 $SIM_{A,S,L_f}^{\mathrm{ORE}}(\lambda)$ 中密文输出的联合分布。对于任何 $j\in[t]$,将密文 ct_j 写作 $(u_{j,1},u_{j,2},\cdots,u_{j,n})$,$\overline{ct}_j$ 写作 $(\overline{u}_{j,1},\overline{u}_{j,2},\cdots,\overline{u}_{j,n})$。另外,对于 $j\in[t]$,将 m_j 的第 s 位记作 $b_{j,s}$。

对于每一个 $s\in[n]$,有以下三种情况:

存在一个 $j\in[t-1]$,使得 $ind_{\mathrm{diff}}(m_j,m_t)>s$。如果有多个 j 使得 $ind_{\mathrm{diff}}(m_j,m_t)>s$,将 j 设为其中的最小值。那么模拟器设为 $\overline{u}_s=L(j,s)$(或 $L(j,s)$ + 随机整数 r,若 m_j 的第 s 位包含在它末尾的连续 1 或 0 子串)。

对于每个 $l\in[t-1]$,$ind_{\mathrm{diff}}(m_l,m_t)\leqslant s$,并且存在一个 $j\in[t-1]$ 使得 $ind_{\mathrm{diff}}(m_j,m_t)=s$。如果有多个 $j\in[t-1]$ 使得 $ind_{\mathrm{diff}}(m_j,m_t)=s$,将 j 设为其中的最小值。那么,模拟器设置 $\overline{u}_s=L(j,s)-(1-2\cdot 1(m_j<m_t))(\mathrm{mod}\ M)$(或 $L(j,s)-(1-2\cdot 1(m_j<m_t))$ + 随机整数 $r(\mathrm{mod}\ M)$,若 m_j 第 s 位包含在它最后的连续 1 或 0 子串)。

对于每个 $l\in[t-1]$,$ind_{\mathrm{diff}}(m_l,m_t)<s$。在这种情况下,模拟器采样 $y\xleftarrow{R}\mathbf{Z}_M$,设置 $\overline{u}_s=y$。

对于每个 $s\in[n]$,模拟器添加映射 $(t,s)\mapsto\overline{u}_s$ 到 L。最终,模拟器 S_t 输出密文 $\overline{ct}_t=(\overline{u}_1,\overline{u}_2,\cdots,\overline{u}_n)$,并且更新状态 $st_S=L$。

以上构造的模拟器可以使得 $REAL_A^{\mathrm{ORE}}(\lambda)$ 和 $SIM_{A,S,L_f}^{\mathrm{ORE}}(\lambda)$ 的输出分布在计算上是不可区分的。附录展示了相关的证明。

(2) 如已知每个明文出现的频率,因为存在随机数,本算法加密每个相同的明文,在可忽略情况下会产生不同的密文,从而改变明文的分布。考虑以下这种攻击:敌手统计所有密文中相同位数的前缀出现的次数,可能可以得知某些明文出现的频率之和。明文字符串末尾为 0 连 i 个 1 或 1 连 i 个 0 的数的频率为 $1/2^{i+1}$,它可以与周围 2^i-1 个明文的频率混淆。若字符串长度为 n,则每个明文期望可以与它周围 $\sum_{i=o}^{n-1}(2^i-1)/2^{i+1}=n/2-1+1/2^n$ 个明文的频率混淆,n 较小时混淆的效果较好。

3.3 时间空间性能

对于长度为 n 的二进制字符串,加密操作需要一次双线性映射和两次幂运算,并 n 次对长度为 n 的二进制字符串进行伪随机运算和一次对称加密;比较操作需要 n 次对 $\mathbf{Z}_M$ 范围内的整数进行比较;解密和验证首先需要一次双线性映射和两次幂运算,然后解密操作需要 n 次对长度为 n 的二进制字符串进行伪随机运算,验证操作需要一次对称加密方式的解密。

为了使性能数据更加明显,在 Intel i5、8 GB 内存、Ubuntu 系统下进行测试,对单个数据进行各种操作,所得时间性能如表 2 所示。可以看出算法适合应用于实际,尤其是它的比较操作很快,适合应用于需要经常进行范围查询的场景。因为生成密钥 K 耗时占总耗时比例较大,而它与明文长度无关,所以 32 bit 和 64 bit

数据的加密、解密、验证速度差距不大。

表 2　时间性能测试

明文长度/bit	步骤	数据组数	总耗时/s	平均耗时/μs
32	密钥生成	100 000	33.7	337
	加密	100 000	66.2	662
	比较	**2 000 000**	**0.56**	**0.28**
	解密	200 000	25.8	258
	验证	200 000	41.9	419
64	密钥生成	100 000	33.9	339
	加密	200 000	66.6	666
	比较	**2 000 000**	**0.83**	**0.42**
	解密	200 000	26.2	262
	验证	200 000	42.4	424

值得注意的是,表 2 实验测试的是插入单个数据时的加密操作耗时和解密验证单个数据时的耗时。而如果插入多组数据,或解密验证多组数据,因为需要很多重复的双线性映射和幂模操作,所以如果只进行一次这些操作并记录结果,则平均处理每组数据的耗时会大大降低。以 32 bit 长度数据为例,图 4 以测试的数据组数为横坐标,以每组数据平均耗时为纵坐标,展示了对多组数据加密、解密和验证的耗时。对多组数据加解密操作的耗时,随数据组数增加,逐渐趋近于 1.92 μs 左右;而对多组数据验证操作的耗时,随数据组数增加,逐渐趋近于 161.1 μs 左右。这有利于快速创建数据库和解密获得已授权的大量敏感信息。

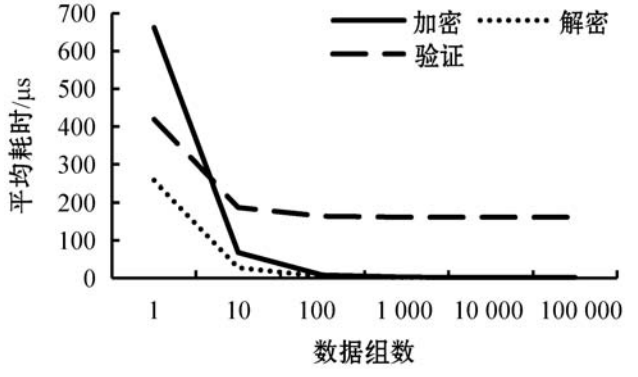

图 4　加解密多组数据时每组数据平均耗时

保序揭序加密协议,在从明文映射到密文时,都需要空间扩展。对于 N 个相同长度的数据,假设数据拥有方 ID 长度为 32 bit,大整数 x 长度为 256 bit,哈希运算得到的数据长度为 128 bit,每个明文验证所需扩展为 512 bit(假设使用的对称加密方式需要块对齐),本算法所需的密文空间如表 3 所示。

表 3　空间性能扩展

数据类型	单个明文所占空间大小/bit	N 个密文所占空间大小/bit	空间扩展倍数
整型	32	1 536N	48
长整型	64	4 608N	72
单精度浮点型	32	1 536N	48
双精度浮点型	64	4 608N	72
字符串	n	$(n^2+512)N$	$(n^2+512)/n$

4　结　语

在众多属性揭示加密算法中,保序及揭序加密是一条重要的分支,然而现有的保序及揭序加密算法缺乏对密文正确性及完整性的验证,缺少对不同身份用户的权限管理,而满足理想安全性的算法实用性较差。因此本文提出一种具有拥有泄露的 ORE 安全的便于实现的具有较优解密效率的身份基匿名签密,在比较时最终会泄露两个明文的最高不同比特位。在此算法的系统模型中,所有用户都可以进行比较操作,而特权用户可以进行身份发现、验证正确完整性和解密操作。算法采用了身份基的加密方式,这有利于对多个数据库的私钥管理。在算法效率方面,对 32 bit 数据间的一次比较操作约耗时 0.28 μs,适用于需要频繁进行范围查询的场景。

本文算法具有一定的频率隐藏特性,但是频率隐藏特性较弱,后续可做更多补充。如何提高算法效率,减少服务器端与客户端之间的通信轮次,减少除顺序信息以外的信息的泄露,是保序及揭序加密算法领域需要持续不断考虑的问题。

参　考　文　献

[1] 王国峰,刘川意,潘鹤中,等. 云计算模式内部威胁综述[J]. 计算机学报,2017,40(2):296 - 316.

[2] 郭晶晶,苗美霞,王剑锋. 保序加密技术研究与进展[J]. 密码学报,2018,5(2):180 - 195.

[3] Rittinghouse J W, Ransome J F. Cloud computing: Implementation, management, and security[M]. CRC Press, 2017.

[4] Agrawal R, Kiernan J, Srikant R, et al. Order preserving encryption for numeric data[C]//2004 ACM SIGMOD International Conference on Management of Data, 2004: 563 - 574.

[5] Boldyreva A, Chenette N, Lee Y, et al. Order-preserving symmetric encryption[C]//28th Annual International Conference on Advances in Cryptology, 2009: 224 - 241.

[6] Boneh D, Lewi K, Raykova M, et al. Semantically secure order-revealing encryption: Multi-input functional encryption without obfuscation[C]//Annual International Conference on the Theory and Applications of Cryptographic Techniques, 2015:563 - 594.

[7] Popa R A, Li F H, Zeldovich N. An ideal-security protocol for order-preserving encoding[C]//IEEE Symposium on Security and Privacy(SP),2013:463 - 477.

[8] Kerschbaum F, Schröpfer A. Optimal average-complexity ideal-security order-preserving encryption[C]//2014 ACM SIGSAC Conference on Computer and Communications Security,2014:275 - 286.

[9] Wang X, Zhao Y. Order-revealing encryption: File-injection attack and forward security[C]//European Symposium on Research in Computer Security,2018:101 - 121.

[10] Ahmed S, Zaman A, Zhang Z, et al. Semi-order preserving encryption technique for numeric database[J]. International Journal of Networking and Computing, 2019, 9(1):111 - 129.

[11] Goldwasser S, Gordon S D, Goyal V, et al. Multi-input functional encryption[C]//Annual International Conference on the Theory and Applications of Cryptographic Techniques, 2014:578 - 602.

[12] Chenette N, Lewi K, Weis S A, et al. Practical order-revealing encryption with limited leakage[C]//International Conference on Fast Software Encryption,2016: 474 - 493.

[13] Chang Y C, Mitzenmacher M. Privacy preserving keyword searches on remote encrypted data[C]//International Conference on Applied Cryptography and Network Security, 2005:442 - 455.

[14] Curtmola R, Garay J, Kamara S, et al. Searchable symmetric encryption: Improved definitions and efficient constructions[J]. Journal of Computer Security, 2011, 19(5): 895 - 934.

[15] Kerschbaum F. Frequency-hiding order-preserving encryption[C]//22nd ACM SIGSAC Conference on Computer and Communications Security,2015:656 - 667.

[16] Grubbs P, Sekniqi K, Bindschaedler V, et al. Leakage-abuse attacks against order-revealing encryption[C]//2017 IEEE Symposium on Security and Privacy (SP), 2017: 655 - 672.

[17] Maffei M, Reinert M, Schröder D. On the security of frequency-hiding order-preserving encryption[C]//International Conference on Cryptology and Network Security, 2017: 51 - 70.

[18] Cash D, Liu F H, O'Neill A, et al. Parameter-hiding order revealing encryption[C]//International Conference on the Theory and Application of Cryptology and Information Security,2018:181 - 210.

[19] Zhao Y. Identity-concealed authenticated encryption and key exchange[C]//2016 ACM SIGSAC Conference on Computer and Communications Security,2016:1464 - 1479.

[20] Wang H, Zhao Y. Identity-Based Higncryption[J]. IACR Cryptology ePrint Archive,2019,2019:106.

[21] Diffie W, Hellman M. New directions in cryptography[J]. IEEE Transactions on Information Theory, 1976, 22(6): 644 - 654.

[22] Joux A. A one round protocol for tripartite Diffie-Hellman[C]//International Algorithmic Number Theory Symposium, 2000:385 - 393.

[23] Boneh D, Franklin M. Identity-based encryption from the Weil pairing[C]//Annual International Cryptology Conference, 2001:213 - 229.

[24] Galbraith S D, Paterson K G, Smart N P. Pairings for cryptographers[J]. Discrete Applied Mathematics, 2008, 156(16):3113 - 3121.

[25] Freeman D, Scott M, Teske E. A taxonomy of pairing-friendly elliptic curves[J]. Journal of Cryptology, 2010,23(2):224 - 280.

(上接第 286 页)

[6] Gupta M, Patwa F, Sandhu R. Object-tagged RBAC model for the Hadoop ecosystem[C]//IFIP Annual Conference on Data and Applications Security and Privacy. Springer,2017: 63 - 81.

[7] 苏秋月,陈兴蜀,罗永刚. 大数据环境下多源异构数据的访问控制模型[J]. 网络与信息安全学报,2019,5(1): 78 - 86.

[8] 苏秋月. 大数据平台上基于属性的角色访问控制模型[J]. 现代计算机(专业版),2019(3):21 - 24.

[9] 王于丁,杨家海. 一种基于角色和属性的云计算数据访问控制模型[J]. 清华大学学报(自然科学版),2017,57(11):1150 - 1158.

[10] Gupta M, Patwa F, Sandhu R. An attribute-based access control model for secure big data processing in Hadoop ecosystem[C]//Proceedings of the Third ACM Workshop on Attribute-Based Access Control. ACM, 2018: 13 - 24.

[11] 陈垚坤,刘文丽. 一种适用于 Hadoop 平台的基于属性访问控制模型[J]. 河南师范大学学报(自然科学版), 2016,44(5):146 - 153.

[12] 刘敖迪,杜学绘,王娜,等. 基于区块链的大数据访问控制机制[J]. 软件学报,2019,30(9):2636 - 2654.

[13] Badger L, Sterne D F, Sherman D L, et al. A domain and type enforcement UNIX prototype[J]. Computing Systems, 1996, 9(1): 47 - 83.