(19)国家知识产权局

(12)发明专利申请

(10)申请公布号 CN 116226160 A
(43)申请公布日 2023.06.06

(21)申请号 202211475316.7
(22)申请日 2022.11.23
(71)申请人 翼方健数(北京)信息科技有限公司
地址 100037 北京市海淀区阜成路73号A座五层507,508,509,510,511,512号
申请人 翼健(上海)信息科技有限公司
(72)发明人 张李军 张浩 王震
(74)专利代理机构 北京华清迪源知识产权代理有限公司 11577
专利代理师 郑兴旺
(51)Int.Cl.
G06F 16/242(2019.01)
G06F 16/22(2019.01)
G06F 21/60(2013.01)
H04L 9/06(2006.01)
H04L 9/08(2006.01)

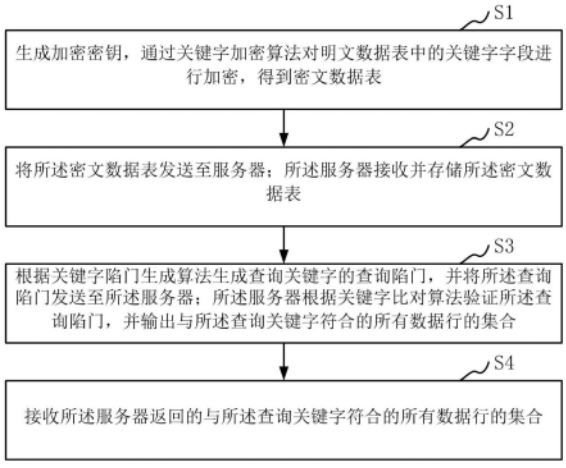
权利要求书2页 说明书8页 附图2页

(54)发明名称

基于密文数据的精确匹配查询方法及装置

(57)摘要

本申请公开了一种基于密文数据的精确匹配查询方法及装置,方法包括:生成加密密钥,通过关键字加密算法对明文数据表中的关键字字段进行加密,得到密文数据表;将密文数据表发送至服务器;服务器接收并存储密文数据表;根据关键字陷门生成算法生成查询关键字的查询陷门,并将查询陷门发送至服务器;服务器根据关键字比对算法验证查询陷门,并输出与查询关键字符合的所有数据行的集合;接收服务器返回的与查询关键字符合的所有数据行的集合。本申请提供基于密文数据的精确匹配查询方法及装置,执行效率高,能有效地阻止攻击者对数据的统计分析以及阻止攻击者进行文件注入攻击恢复出之前查询请求中的关键字明文数据。



CN 116226160 A

1.一种基于密文数据的精确匹配查询方法,其特征在于,包括:

生成加密密钥,通过关键字加密算法对明文数据表中的关键字字段进行加密,得到密文数据表;

将所述密文数据表发送至服务器;所述服务器接收并存储所述密文数据表;

根据关键字陷门生成算法生成查询关键字的查询陷门,并将所述查询陷门发送至所述服务器;所述服务器根据关键字比对算法验证所述查询陷门,并输出与所述查询关键字符合的所有数据行的集合;

接收所述服务器返回的与所述查询关键字符合的所有数据行的集合。

2.根据权利要求1所述的基于密文数据的精确匹配查询方法,其特征在于,所述关键字加密算法为:C_i=Hash(t_i,index),其中,C_i为密文,Hash是哈希算法,index为数据表中该数据记录所在的行号,t_i=HMAC(key,kw_i),HMAC是基于哈希函数的消息认证码算法,key为密钥,kw_i为待加密的关键字。

3.根据权利要求2所述的基于密文数据的精确匹配查询方法,其特征在于,所述Hash为SHA256算法或SM3算法。

4.根据权利要求2所述的基于密文数据的精确匹配查询方法,其特征在于,所述HMAC为SHA256-HMAC。

5.根据权利要求2所述的基于密文数据的精确匹配查询方法,其特征在于,所述密钥的长度为16字节。

6.根据权利要求1所述的基于密文数据的精确匹配查询方法,其特征在于,所述关键字陷门生成算法为:T=(t,id),其中,T为查询陷门,t=HMAC(key,kw),HMAC是基于哈希函数的消息认证码算法,key为密钥,kw为查询关键字,id=PRF(seed,time),seed是PRF的种子,PRF为伪随机算法,time为当前时间,id为请求标识。

7.根据权利要求1所述的基于密文数据的精确匹配查询方法,其特征在于,所述关键字比对算法为:

抽取所述查询陷门的请求标识,验证所述请求标识是否存在;

若所述请求标识存在,则返回错误消息;

若所述请求标识不存在,则计算C'=Hash(t,index),其中,Hash是哈希算法,index为数据表中该数据记录所在的行号;

判断C'是否与密文关键字相同,若相同,则返回True,若不相同,则返回False。

8.一种基于密文数据的精确匹配查询装置,其特征在于,包括:

加密模块,用于生成加密密钥,

关键字加密算法模块,用于通过关键字加密算法对明文数据表中的关键字字段进行加密,得到密文数据表;

发送模块,用于将所述密文数据表发送至服务器;所述服务器接收并存储所述密文数据表;

关键字陷门生成算法模块,用于根据关键字陷门生成算法生成查询关键字的查询陷门,并将所述查询陷门发送至所述服务器;所述服务器根据关键字比对算法验证所述查询陷门,并输出与所述查询关键字符合的所有数据行的集合;

接收模块,用于接收所述服务器返回的与所述查询关键字符合的所有数据行的集合。

9.一种计算机设备，包括存储器和处理器，所述存储器存储有计算机程序，其特征在于，所述处理器执行所述计算机程序时实现权利要求1至7中任一项所述的方法的步骤。

10.一种计算机可读存储介质，其上存储有计算机程序，其特征在于，所述计算机程序被处理器执行时实现权利要求1至7中任一项所述的方法的步骤。

基于密文数据的精确匹配查询方法及装置

技术领域

[0001] 本申请涉及数据查询技术领域，具体涉及一种基于密文数据的精确匹配查询方法及装置。

背景技术

[0002] 数据的精确匹配是数据库查询中最常用的查询方式，精确匹配是指查询时期望在数据库中找到与查询指定的值完全一致的数据(也称为等值查询)。由于网络攻击导致数据库泄露的事件时有发生，企业和个人越来越多地对数据库进行了加密。由于数据库中存储的数据是密文形式，在精确匹配查询时，通常需要将加密数据进行解密后才能判断是否和查询指定值相等。解密后数据变成了明文状态，这就给数据的安全性带来了风险。因此若能直接在密文形式下进行精确匹配查询，则能在更好的安全性保证下实现数据库的查询功能。

[0003] 密文形式的数据精确匹配查询可以归于可搜索加密领域，搜索的数据可以看成是关键词。可搜索加密的核心思想是对关键词进行随机化或模糊化的处理，形成密文形式的索引，从而保护关键字的安全性。对关键字相关联的文件(或关联的信息)进行额外的加密处理，并把密文索引和加密后的关联文件构成一个对应关系。查询时提交关键字(通常为了隐私性，也是密文形式)，通过一个比对算法判断出是否与数据库中某个密文索引相同(此时即进行精确匹配)，相同时则输出其对应的关联文件或信息。

[0004] 在对关键词进行随机化处理时，可搜索加密根据所使用的密钥方式可以分为对称可搜索和公钥可搜索方案。2000年Song等人最早提出了对称可搜索方案，采用全文扫描的方式，虽然可以避免让服务器获取到明文数据信息，但算法效率很低，且不能抵抗统计分析。2016年Xia等人提出了支持文件动态变化的关键词搜索方案，2017年杨旸等人提出了关键词的排序搜索方案以及Kim等人构建了高效更新的前向安全可搜索加密方案等。

[0005] 为了解决多用户加密的场景，Boneh利用椭圆曲线双线性对提出了公钥可搜索加密。采用公钥来对关键词进行加密，而用对应的私钥来构建密文状态的关键词精确匹配算法。目前几乎所有的公钥可搜索加密都是采用双线性对这一密码学工具来进行设计。双线性对基于椭圆曲线点乘算法进行计算，实现时涉及椭圆曲线方程、双线性对类型选择等多种参数，计算过程复杂，效率较低(根据目前公开的数据表明速度比对称加密至少慢1个数量级以上)。

[0006] 综上，现有的数据精确匹配算法具有以下缺陷：

[0007] (1)数据的加密处理和比对算法效率不高，当数据库中的密文索引数据量较大时，数据的搜索和匹配效率难以满足实际需求；

[0008] (2)现有算法相同的数据(如关键字)得到的密文索引相同，无法抵抗统计分析；攻击者可能利用之前泄露的文件来进行对应明文信息的恢复；

[0009] (3)查询请求中的关键字数据容易受到文件注入攻击，攻击者通过构造包含特定关键字的新文件(或数据库中的数据记录)并结合之前的查询请求能够恢复出查询请求中

的关键字明文。

发明内容

[0010] 为此,本申请提供一种基于密文数据的精确匹配查询方法及装置,以解决现有技术存在的数据精确匹配算法效率低、安全性差的问题。

[0011] 为了实现上述目的,本申请提供如下技术方案:

[0012] 第一方面,一种基于密文数据的精确匹配查询方法,包括:

[0013] 生成加密密钥,通过关键字加密算法对明文数据表中的关键字字段进行加密,得到密文数据表;

[0014] 将所述密文数据表发送至服务器;所述服务器接收并存储所述密文数据表;

[0015] 根据关键字陷门生成算法生成查询关键字的查询陷门,并将所述查询陷门发送至所述服务器;所述服务器根据关键字比对算法验证所述查询陷门,并输出与所述查询关键字符合的所有数据行的集合;

[0016] 接收所述服务器返回的与所述查询关键字符合的所有数据行的集合。

[0017] 作为优选,所述关键字加密算法为:C_i=Hash(t_i,index),其中,C_i为密文,Hash是哈希算法,index为数据表中该数据记录所在的行号,t_i=HMAC(key,kw_i),HMAX是基于哈希函数的消息认证码算法,key为密钥,kw_i为待加密的关键字。

[0018] 作为优选,所述Hash为SHA256算法或SM3算法。

[0019] 作为优选,所述HMAC为SHA256-HMAC。

[0020] 作为优选,所述密钥的长度为16字节。

[0021] 作为优选,所述关键字陷门生成算法为:T=(t,id),其中,T为查询陷门,t=HMAC(key,kw),HMAC是基于哈希函数的消息认证码算法,key为密钥,kw为查询关键字,id=PRF(seed,time),seed是PRF的种子,PRF为伪随机算法,time为当前时间,id为请求标识。

[0022] 作为优选,所述关键字比对算法为:

[0023] 抽取所述查询陷门的请求标识,验证所述请求标识是否存在;

[0024] 若所述请求标识存在,则返回错误消息;

[0025] 若所述请求标识不存在,则计算C'=Hash(t,index),其中,Hash是哈希算法,index为数据表中该数据记录所在的行号;

[0026] 判断C'是否与密文关键字相同,若相同,则返回True,若不相同,则返回False。

[0027] 第二方面,一种基于密文数据的精确匹配查询装置,包括:

[0028] 加密模块,用于生成加密密钥,

[0029] 关键字加密算法模块,用于通过关键字加密算法对明文数据表中的关键字字段进行加密,得到密文数据表;

[0030] 发送模块,用于将所述密文数据表发送至服务器;所述服务器接收并存储所述密文数据表;

[0031] 关键字陷门生成算法模块,用于根据关键字陷门生成算法生成查询关键字的查询陷门,并将所述查询陷门发送至所述服务器;所述服务器根据关键字比对算法验证所述查询陷门,并输出与所述查询关键字符合的所有数据行的集合;

[0032] 接收模块,用于接收所述服务器返回的与所述查询关键字符合的所有数据行的集

合。

[0033] 第三方面，一种计算机设备，包括存储器和处理器，所述存储器存储有计算机程序，所述处理器执行所述计算机程序时实现基于密文数据的精确匹配查询方法的步骤。

[0034] 第四方面，一种计算机可读存储介质，其上存储有计算机程序，其特征在于，所述计算机程序被处理器执行时实现基于密文数据的精确匹配查询方法的步骤。

[0035] 相比现有技术，本申请至少具有以下有益效果：

[0036] 本申请提供了一种基于密文数据的精确匹配查询方法及装置，方法包括：生成加密密钥，通过关键字加密算法对明文数据表中的关键字字段进行加密，得到密文数据表；将密文数据表发送至服务器；服务器接收并存储密文数据表；根据关键字陷门生成算法生成查询关键字的查询陷门，并将查询陷门发送至服务器；服务器根据关键字比对算法验证查询陷门，并输出与查询关键字符合的所有数据行的集合；接收服务器返回的与查询关键字符合的所有数据行的集合。本申请提供基于密文数据的精确匹配查询方法及装置，执行效率高，能有效地阻止攻击者对数据的统计分析以及阻止攻击者进行文件注入攻击恢复出之前查询请求中的关键字明文数据。

附图说明

[0037] 为了更直观地说明现有技术以及本申请，下面给出几个示例性的附图。应当理解，附图中所示的具体形状、构造，通常不应视为实现本申请时的限定条件；例如，本领域技术人员基于本申请揭示的技术构思和示例性的附图，有能力对某些单元(部件)的增/减/归属划分、具体形状、位置关系、连接方式、尺寸比例关系等容易作出常规的调整或进一步的优化。

[0038] 图1为本申请实施例一提供的一种基于密文数据的精确匹配查询方法(以数据拥有者为执行主体)流程图；

[0039] 图2为本申请实施例一提供的一种基于密文数据的精确匹配查询方法(数据拥有者和服务器双方交互)流程图；

[0040] 图3为本申请实施例一提供的明文形式的数据库表；

[0041] 图4为本申请实施例一提供的密文形式的数据库表。

具体实施方式

[0042] 以下结合附图，通过具体实施例对本申请作进一步详述。

[0043] 在本申请的描述中：除非另有说明，“多个”的含义是两个或两个以上。本申请中的术语“第一”、“第二”、“第三”等旨在区别指代的对象，而不具有技术内涵方面的特别意义(例如，不应理解为对重要程度或次序等的强调)。“包括”、“包含”、“具有”等表述方式，同时还意味着“不限于”(某些单元、部件、材料、步骤等)。

[0044] 本申请中所引用的如“上”、“下”、“左”、“右”、“中间”等的用语，通常是为了便于对照附图直观理解，而并非对实际产品中位置关系的绝对限定。在未脱离本申请揭示的技术构思的情况下，这些相对位置关系的改变，当亦视为本申请表述的范畴。

[0045] 实施例一

[0046] 请参阅图1和图2，本实施例提供了一种基于密文数据的精确匹配查询方法，包括：

[0047] S1:生成加密密钥,通过关键字加密算法对明文数据表中的关键字字段进行加密,得到密文数据表;

[0048] 具体的,数据拥有者生成加密密钥key,调用关键字加密算法对数据表中的明文关键字字段的每个数据进行加密,形成整个密文数据表。

[0049] 更具体的,关键字加密算法Encrypt为:

[0050] 设数据库记录中所有的关键字列表为$KW=\{kw_1,kw_2,\cdots,kw_n\}$。

[0051] 算法输入:待加密的关键字kw_i,$i\in[1,n]$,加密密钥key,以及数据表中该数据记录所在的行号index;

[0052] 算法输出:返回kw_i的密文C_i。

[0053] 算法描述:

[0054] 步骤1:计算$t_i=HMAC(key,kw_i)$,其中,HMAC是一个基于哈希函数的消息认证码算法(如SHA256-HMAC算法),key是算法的密钥,密钥的长度为16字节。

[0055] 步骤2:计算密文$C_i=Hash(t_i,index)$,其中,Hash是一个安全的哈希算法(如SHA256,SM3等)。

[0056] 步骤3:返回密文结果C_i。

[0057] S2:将密文数据表发送至服务器;服务器接收并存储密文数据表;

[0058] 具体的,数据拥有者将密文数据表发送给服务器,服务器接收并存储该密文数据表。

[0059] S3:根据关键字陷门生成算法生成查询关键字的查询陷门,并将查询陷门发送至服务器;服务器根据关键字比对算法验证查询陷门,并输出与查询关键字符合的所有数据行的集合;

[0060] 具体的,针对查询的关键字kw,数据拥有者调用关键字陷门生成算法生成查询陷门T,将查询陷门T发送给服务器。服务器收到查询陷门T,调用关键字比对算法,验证请求标识是否是新鲜数(即不是之前查询请求的重放),并记录关键字比对算法返回True的所有数据行index的集合S,将集合S返回给数据拥有者。

[0061] 更具体的,关键字陷门生成算法TrapGen为:

[0062] 算法输入:指定的查询关键字kw,密钥key(该密钥与关键字加密算法中的密钥相同)。

[0063] 算法输出:返回关键字kw的查询陷门T。

[0064] 算法描述:

[0065] 步骤1:计算$t=HMAC(key,kw)$,此处的HMAC算法与关键字加密算法中的HMAC算法相同。

[0066] 步骤2:利用伪随机算法PRF生成本次查询的请求标识:$id=PRF(seed,time)$,其中,seed是PRF的种子,time为当前时间作为算法参数,保证每次生成的id不同。

[0067] 步骤3:返回查询陷门$T=(t,id)$。

[0068] 关键字比对算法Compare为:

[0069] 算法输入:查询陷门T,数据记录的密文关键字C,数据记录所在行号index。

[0070] 算法输出:关键字精确匹配查询结果(True/False)或者错误消息Error。

[0071] 算法描述:

[0072] 步骤1:对查询陷门T＝(t,id),抽取出请求标识id,验证该id是否已经出现过(对之前的所有查询请求id进行记录)。

[0073] 若该id不存在,则继续执行下一步;

[0074] 否则若该id已存在,表明该查询请求是重放,拒绝处理,返回错误消息Error:已处理过该请求。

[0075] 步骤2:计算C'＝Hash(t,index),此处的Hash算法与关键字加密算法中的Hash算法相同。

[0076] 步骤3:比对密文C和C'是否相同,相同则返回True,否则返回False。

[0077] S4:接收服务器返回的与查询关键字符合的所有数据行的集合。

[0078] 具体的,数据拥有者接收集合S,精确匹配查询结束(可根据实际的数据处理需求对集合S表示的数据记录做进一步的操作)。

[0079] 下面以一个具体的例子来进一步说明本实施例提供的基于密文数据的精确匹配查询方法。

[0080] 为了简便起见,假设数据库表有三个字段:数据行号index,姓名name以及职业occupation,其中,职业作为要查询的字段,即加密和查询算法中的关键字字段,职业包含teacher,student,doctor,worker,officer,lawyer等6个关键字。

[0081] S1:原始明文形式的数据库如图3所示,加密后的密文数据库如图4所示,其中,姓名字段可采用AES算法进行加密(这里简易地以****来表示密文,仅表明数据记录的其他字段也可以进行加密处理),而关键字字段则采用本申请提供的关键字加密算法Encrypt进行加密。Hash算法选择SHA256算法,HMAC算法选择基于SHA256的消息认证码算法SHA256-HMAC。伪随机函数基于SHA256算法生成,秘密的种子seed＝“this is atest seed”,time为获取到的当前操作系统时间(取值精确到秒)。密钥key的长度为16字节,即安全参数大小为128比特。Key取值为0x010203040506070809 0A0B0C0D0E0F10.

[0082] S2:可以看到关键字加密后,相同的关键字的密文是不同的(第2行和第3行的关键字明文相同,但密文不同)。查询时,假设要查询关键字为kw＝“doctor”,则利用关键字陷门生成算法TrapGen生成对应的查询陷门T＝t,id),其中,

[0083] t＝0x84E399586A185271E91300C07B9B2529,

[0084] id＝0x9A53DED28382D6AF02647E58D0FB1300。

[0085] S3:服务器执行关键字比对算法Compare,以查询陷门T和每个密文字段作为输入,经过计算,得到精确匹配的数据行是第2行和第3行,返回结果S＝{2,3}。可以看出这与明文形式的匹配结果相同。匹配的记录输出时,实际中可根据需要输出这两行记录的指定字段作为查询结果(比如只输出姓名)。

[0086] 本实施例提供的基于密文数据的精确匹配查询方法,适用于两种数据安全分享场景:

[0087] 一、数据外包和查询场景。此时数据拥有方利用自己的密钥key将关键字字段进行加密后上传到外包服务器(数据记录的其他字段可根据安全需要采用普通的加密,比如使用AES等加密算法)。查询时,由数据拥有方生成查询陷门并发送给服务器,服务器负责密文数据库的存储和执行比对算法,返回匹配的数据记录给数据拥有方。在这种场景中,将服务器和第三方攻击者都视为攻击者,需要保证数据记录本身和查询的关键字的安全性,即对

应的明文不能被恢复。

[0088] 二、服务器自己加密和查询场景。服务器自己拥有明文数据，为了保证数据明文不被窃取，对数据进行加密并在密文形式的数据上执行查询。此时，服务器针对数据查询方提交的关键字明文进行查询陷门T的计算，然后比对数据库中的关键字字段，返回匹配的数据记录。在这种场景中，也不能让攻击者(此时即不是合法的数据查询方)知晓数据记录和查询的关键字。

[0089] 本实施例中提供的三种算法具有以下特点：

[0090] 一、算法的精确度高

[0091] 关键字比对算法Compare的正确性高。由关键字加密算法Encrypt可知C_i＝Hash(t_i,index)，而根据查询时提交的关键字陷门计算出的C'＝Hash(t,index)。因为index是相同的，根据哈希函数的抗碰撞性，C'＝C_i当且仅当t_i＝t，这是因为当t_i≠t而C'＝C_i的概率可忽略。

[0092] 进一步，根据t_i和t的生成算法HMAC可知，这两个值相等当且仅当数据库里的关键字与查询的关键字相同，即kw_i＝kw。因此关键字比对算法Compare输出True时表示密文字段对应的关键字和查询的关键字相同，False则表示这两个关键字不同，实现了关键字的精确比对，得到了匹配结果。

[0093] 二、算法的安全性高

[0094] (1)对于数据外包和查询场景。此时对服务器而言，由于关键字已被加密，故服务器不知晓关键字密文字段对应的关键字明文。查询时，查询者首先将查询的关键字通过陷门生成算法生成查询陷门。根据哈希函数单向性，这个查询陷门不会泄露被查询的关键字。服务器执行关键字比对算法Compare，只能知晓查询的关键字和密文字段中关键字是否相同，而不会知晓关键字明文信息，这也就保证了查询的关键字的安全性。另外，我们通过引入数据位置index来对关键字进行加密，实现了不同位置的关键字的密文不同(即使是相同的关键字在不同的位置上得到的密文也不同)，这样服务器或第三方攻击者不能通过密文字段来得出关键字是否相同的信息，也不能实施统计分析判断出关键字的分布。由于每次查询请求都使用了查询id来进行标识，攻击者无法利用之前的查询，有效地阻止了攻击者发起文件注入攻击来获取查询请求的关键字明文。

[0095] (2)对于服务器自己加密和查询场景。该场景主要防止攻击者利用网络攻击获取到数据库明文从而造成数据泄露。此场景查询时查询方提交的是关键字明文，为了保护关键字明文，在进行网络传输时，可以利用SSL/TLS等协议来加密查询请求。服务器生成对应的查询陷门，执行密文关键字比对，返回匹配结果。这样第三方攻击者也不能通过数据库密文字段来恢复出关键字明文以及关键字的分布信息。

[0096] 三、算法的效率高、可扩展性强

[0097] (1)算法的高效性。本实施例提供的三种算法都是基于对称密码原语(哈希算法，HMAC算法，伪随机函数)，算法的执行效率很高。实际测试时，在CentOS系统，主频2.0G的Intel Xeon CPU上单线程都能达到100万次/秒的查询速度。此外，可以看出针对每次查询，陷门只需要计算一次。关键字比对算法仅与陷门和数据位置index有关，因此算法可以采用多线程方式并性化地执行，显著提高查询的计算效率。

[0098] (2)算法的可扩展性。从关键字加密算法可以看出，算法可以支持关键字和数据记

录的灵活扩展,满足实际中关键字和数据的增加(或删除)等动态变化时的场景需求。

[0099] 本实施例提供的基于密文数据的精确匹配查询方法,首先将要查询的数据列字段使用关键字加密算法Encrypt进行加密。为了直观,本实施例将这列数据字段称为关键字字段。查询时,对指定的查询关键字明文生成密文形式的查询陷门T,将查询陷门T提交给加密数据库存储服务器,服务器执行关键字比对算法Compare,比对查询陷门T和关键字字段,输出匹配的数据记录(实际中也可根据需要输出匹配记录行的指定字段)。执行效率高,能有效地阻止攻击者对数据的统计分析以及阻止攻击者进行文件注入攻击恢复出之前查询请求中的关键字明文数据。

[0100] 实施例二

[0101] 本实施例提供了一种基于密文数据的精确匹配查询装置,包括:

[0102] 加密模块,用于生成加密密钥,

[0103] 关键字加密算法模块,用于通过关键字加密算法对明文数据表中的关键字字段进行加密,得到密文数据表;

[0104] 发送模块,用于将所述密文数据表发送至服务器;所述服务器接收并存储所述密文数据表;

[0105] 关键字陷门生成算法模块,用于根据关键字陷门生成算法生成查询关键字的查询陷门,并将所述查询陷门发送至所述服务器;所述服务器根据关键字比对算法验证所述查询陷门,并输出与所述查询关键字符合的所有数据行的集合;

[0106] 接收模块,用于接收所述服务器返回的与所述查询关键字符合的所有数据行的集合。

[0107] 关于基于密文数据的精确匹配查询装置的具体限定可以参见上文中对于基于密文数据的精确匹配查询方法的限定,在此不再赘述。

[0108] 实施例三

[0109] 本实施例提供了一种计算机设备,包括存储器和处理器,所述存储器存储有计算机程序,所述处理器执行所述计算机程序时实现基于密文数据的精确匹配查询方法的步骤。

[0110] 实施例四

[0111] 本实施例提供了一种计算机可读存储介质,其上存储有计算机程序,所述计算机程序被处理器执行时实现基于密文数据的精确匹配查询方法的步骤。

[0112] 综上,本申请提供的基于密文数据的精确匹配查询方法及装置具有以下优势:

[0113] (1)算法执行效率高并且可以大规模地并性化处理查询比对,算法具备良好的可扩展性,非常适合于数据库的密文等值查询以及可搜索加密中的关键字查询等需求场景;

[0114] (2)算法通过引入数据位置index作为加密参数,以及设置查询请求的标识id,有效地阻止攻击者针对数据密文字段的统计分析和文件注入攻击,保证了原始明文字段和查询时关键字数据的安全性;

[0115] (3)算法主要基于哈希的消息认证码进行设计,密文形式的数据比对过程中无需解密,因此整个比对环节没有明文出现,从而保证了数据明文不会泄露。

[0116] 以上实施例的各技术特征可以进行任意的组合(只要这些技术特征的组合不存在矛盾),为使描述简洁,未对上述实施例中的各个技术特征所有可能的组合都进行描述;这

些未明确写出的实施例,也都应当认为是本说明书记载的范围。

[0117] 上文中通过一般性说明及具体实施例对本申请作了较为具体和详细的描述。应当理解,基于本申请的技术构思,还可以对这些具体实施例作出若干常规的调整或进一步的创新;但只要未脱离本申请的技术构思,这些常规的调整或进一步的创新得到的技术方案也同样落入本申请的权利要求保护范围。

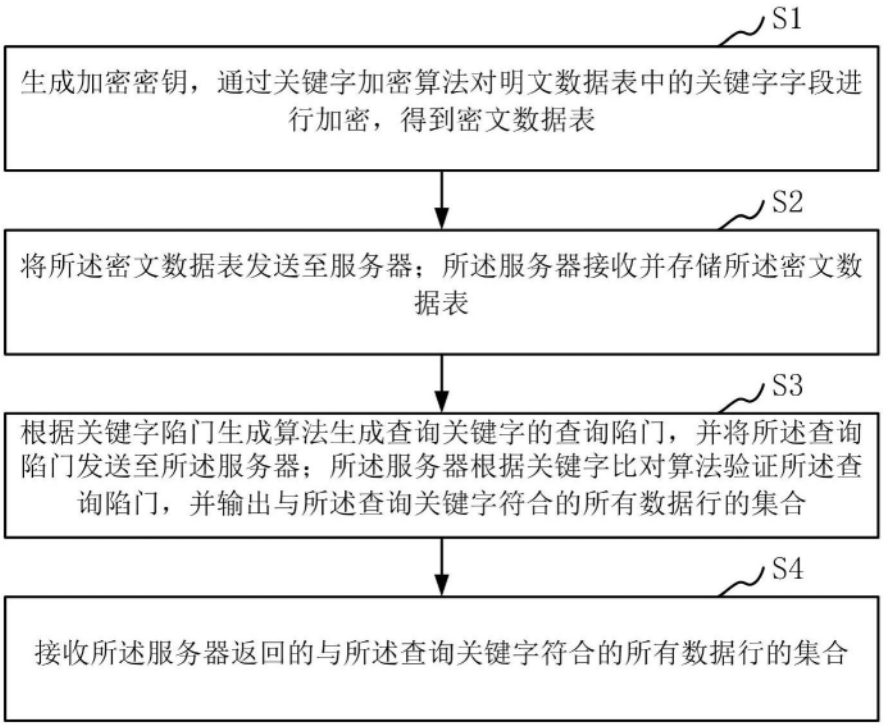

S1
生成加密密钥，通过关键字加密算法对明文数据表中的关键字字段进行加密，得到密文数据表
S2
将所述密文数据表发送至服务器；所述服务器接收并存储所述密文数据表
S3
根据关键字陷门生成算法生成查询关键字的查询陷门，并将所述查询陷门发送至所述服务器；所述服务器根据关键字比对算法验证所述查询陷门，并输出与所述查询关键字符合的所有数据行的集合
S4
接收所述服务器返回的与所述查询关键字符合的所有数据行的集合

图1


数据拥有者
服务器
步骤1：对明文数据表关键字字段加密，形成密文数据表
发送密文数据表
步骤2：接收并存储密文数据表
步骤3：对查询关键字kw，生成查询陷门T
发送查询陷门T
步骤4：接收查询陷门T，执行算法3得到返回为True的所有数据行index的集合S
步骤6：接收查询结果S并根据实际需求进行进一步处理。
发送查询结果S
步骤5：服务器将查询结果S返回给数据拥有者

图2

数据行号index	姓名name	职业occupation
1	James	student
2	Elan	doctor
3	Jerry	doctor
4	Alice	worker
5	Bob	lawyer
6	Tayler	teacher
7	Sherry	officer
8	Foster	worker

图3

数据行号index	姓名name	职业occupation（**关键字字段**）
1	****	D6A7504DD2A2E9C39CEFC4FD1C246176
2	****	384091DE81E72E70F4672F7606F1A090
3	****	6833EE77DE2A56EF1BA70E1049E4A615
4	****	16C919E9B5520624E74672A4A685036A
5	****	829E67817AFF59753CDF69EF07F083F5
6	****	99D353904B87CD090A164B2F48506289
7	****	B0AD0EEBC257DF0470644EF79448485C
8	****	4A0CE415F7C98FAA46C5FAEBE2F0C4AF

图4