应用于数据库安全保护的加解密引擎系统

王 茜[1],朱志祥[2],史晨昱[1],张 磊[1]
(1. 西安未来国际信息股份有限公司,陕西 西安 710063;
2. 西安邮电大学,陕西 西安 710121)

摘 要:针对系统业务数据安全存储问题,采用加解密引擎服务对应用端发送数据进行加密,根据用户 ID 来识别不同用户的传输命令,利用用户私有的 KEY 进行加密存储。该方案在云计算平台下具有保护用户数据存储安全、隔离数据的功能。当应用端用户查询已加密数据时,加解密引擎服务端根据用户 ID 读取缓存区用户密钥,解密数据返回给应用端明文数据。当加解密系统和第三方应用进行集成时,加解密引擎将载入用户定义的加密算法、加密矢量等信息,按照用户自身的密钥加密敏感数据,可保证用户敏感数据存储安全并隔离不同用户的业务数据。以某市人口库系统集成为例,验证了方案的可行性。

关键词:安全存储;加密存储;密钥;云计算
中图分类号:TP302.1 **文献标识码**:A **文章编号**:1673-629X(2014)01-0143-04
doi:10.3969/j.issn.1673-629X.2014.01.037

Encryption and Decryption Engine System Applying to Database Security and Detection

WANG Qian[1], ZHU Zhi-xiang[2], SHI Chen-yu[1], ZHANG Lei[1]
(1. Xi'an Future International Information Co., Ltd., Xi'an 710063, China;
2. Xi'an University of Posts & Telecommunications, Xi'an 710121, China)

Abstract: In order to implement the secure storage of the system transaction data, the encryption and decryption engine is used to encrypt the data that transmitted from the application terminal. The transfer commands of different users can be identified by the user ID, and the user's private key can be used to achieve encrypted storage. The scheme can protect the storage security of user data and isolate data in cloud computing platform. When the users from application terminal query the encrypted data, the encryption and decryption engine server-side will read the user's private key from the buffer according to the user ID, then the server-side will decrypt the data and return clear text to application terminal. When the encryption and decryption system is integrated with a third-party application, the user-defined encryption algorithm and encryption vector information will be loaded in the encryption and decryption engine, then the sensitive data will be encrypted according to the user's own private key, which can guarantee the secure storage of sensitive data and achieve isolation for business data of different users. Finally, the integration with the population database system of a city is used as an example to verify the feasibility of the scheme.

Key words: secure storage; encrypted storage; encrypted key; cloud computing

0 引 言

信息化应用系统都需要关系型数据库支撑,这些数据库中存储了一些敏感信息,对这些敏感数据添加安全保护措施就是一个最为急迫的需求[1-4];传统的信息系统中,大量的数据存储在关系型数据库中,如银行、电信等行业数据库中存储了大量的个人数据,对这些敏感数据增加安全措施,防止数据泄露是数据安全的重要方向[5-9];同时随着云计算日益发展,现在信息化应用系统越来越多地使用云计算的方式对外提供信息服务,在云计算多租户模式下,不同的用户共用一套业务系统,不同租户之间的数据安全、数据隔离问题就成了一个需要解决的问题[10-12];因此,建立应用于数据库安全保护和数据隔离的加解密引擎系统设计变得极为迫切。

收稿日期:2013-04-09 **修回日期**:2013-07-12 **网络出版时间**:2013-11-12
基金项目:陕西省自然科学基金资助项目(2012JM7017)
作者简介:王 茜(1966-),女,博士,高级工程师,研究方向为电子政务;朱志祥,博士,教授,研究方向为网络与信息安全技术。
网络出版地址:http://www.cnki.net/kcms/detail/61.1450.TP.20131112.1627.003.html

1 系统架构

图 1 为文中提出的加解密引擎系统的总体架构。在该体系架构中,第三方应用首先需要加载加解密引擎统一数据库连接(UDBC)驱动包,所发送的 SQL 请求再通过 Socket 发送到加解密引擎系统中,加解密引擎服务中心经过解析 SQL 请求语句,根据加解密引擎规则管理部分预先读取的规则判断该 SQL 是否需要加密(即使加密算法、密钥转换明文为密文),判断需要加密时通过密钥管理组件中密钥获取接口读取该用户自身的密钥值,结合用户预先添加的加密规则进行加密。当用户请求查询密文数据时,加解密引擎解析 SQL 请求判断是否包含需要解密的字段,如果包含则同样使用密钥访问接口、规则读取接口和加密算法规则逆向解密数据。

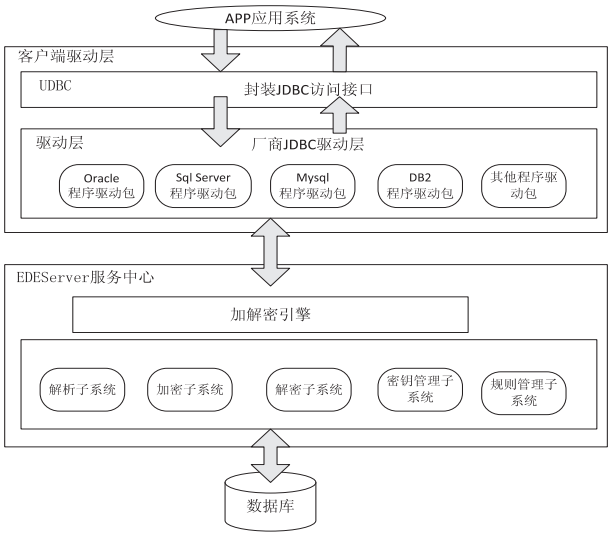

图 1 系统架构图

2 加解密引擎服务

加解密引擎服务实现了对应用端发送命令中需要加密字段的加密,并且根据 USERID 来识别不同用户传输的命令,使用用户私有的 KEY 进行加密存储。并在云计算平台下起了保护用户数据安全存储、隔离数据的功能。当应用端用户查询已加密数据时,加解密引擎服务端根据用户 ID 读取缓存区用户密钥,解密数据返回给应用端明文数据。

2.1 加密组件

加解密引擎服务的核心部分是对符合规则的字段数据按照用户设置的用户标识 ID、加密算法、加密密钥以及加密矢量进行加密运算生成密文,最终将密文推送给数据库系统进行密文存储。如何将加密规则和用户提交的请求命令进行匹配是加密组件应解决的核心问题。为此将用户信息、加密规则、密钥库对象分别进行封装成一系列对象进行处理,在应用端完成提交表名和列解析后剥离出数据库所有字段,依次使用加密规则、用户信息进行过滤,对符合规则的列进行加密;对不符合规则的列将不加密。

通过配置相关属性文件可以将加密组件添加到 EDEServer 中,在实际应用中将这一个组件打成了一个 JAR 包,可以按照项目实际需求进行增加或移除。

加密组件的数据来源于第三方应用系统,用户添加 UDBC(统一数据库连接)将数据库操作命令以字符流传输到加解密引擎服务端后,再由读取流读取数据流并进行序列化后转化为命令请求对象,加密组件以命令请求对象为参数进行构造、过滤、查询、加密处理。将最终的密文以 TCP 的形式通过数据驱动程序发送到数据库中完成密文存储。

另外加密组件使用内存数据库或缓存对象操作规则,通过密钥管理接口实现读取密钥 index、验证密钥有效性、注销密钥等操作。

解析组件通过递归的方式解析出最终不含子查询命令的语句,将列名、表名、数据库名作为参数传递到加密方法中。

加密算法根据所传递的列名、表名、数据库名参数和缓存规则库的规则进行对比,如果不符合缓存规则该类不进行加密,使用 break 语句直接跳出;如果符合规则,加密方法通过调用密钥管理库的密钥读取接口获取该列的密钥,并将该密钥存储到缓存库中,以供解密使用。

2.2 解密组件

加解密引擎服务的另一核心部分是对符合规则的数据按照用户标记 ID、加密算法、密钥以及加密矢量进行解密运算生成明文,最终将明文送到第三方应用系统中,供显示或做进一步业务处理。

如何将加密规则和用户提交的请求命令进行绑定是解密组件应解决的核心问题。为此系统将用户信息、加密规则、密钥库封装成一系列对象后进行处理,在应用端完成提交表名和列解析后剥离出数据库所有字段,依次使用加密规则、用户信息进行过滤,如果符合加密规则则对该列进行解密;如果不符合规则,则该列不进行解密。

在并发访问加解密引擎系统时,大数据量密文的解密返回可能需要消耗较长时间,应用界面将处于等待状态,这样会使得用户体验效果较差,这个问题可以通过部署多实例进行处理,或者利用单实例多线程进行处理。

同样,通过配置相关属性文件也可将解密组件添加到 EDEServer 中,在实际应用中将这一个组件打成一个 JAR 包,可按照项目实际需求进行增加或移除。

用户通过数据库操作命令将数据库存储的密文读

取到加解密引擎服务中,解密组件将命令请求对象作为参数进行构造、过滤、查询、解密处理,将最终的明文以 TCP 形式、通过数据驱动程序发送到第三方应用系统中。

2.3　用户数据隔离保护

加解密引擎系统根据用户 ID 获取用户自身密钥来加密业务数据,不同用户所使用的密钥不同,使得用户使用应用系统产生的业务数据最终都是密文存储,即使使用自身密钥也无法进行解密,从而防止了内部用户窃取、访问数据。

另外加解密引擎系统增加了用户管理模块,支持信用的注册、审核、操作审计等功能,对用户实现分权限、分角色、分组管理,实现了对业务数据访问控制的安全,阻挡了非法用户对应用系统的操作。

3　技术实现

在标准的 JDBC 接口基础上,加解密引擎系统对原有的 JDBC 访问接口和类进行了二次封装和重写,这一部分在加解密系统中称为 UDBC(统一数据库连接),UDBC 和驱动层程序工作在不同层面(如图 1 所示),应用程序添加不同的数据驱动使用 UDBC 与数据库系统进行通讯。在第三应用需要添加 UDBC 的驱动 URL、驱动 JAR 即可,UDBC 屏蔽了不同数据库驱动地址、SQL 请求的差异性。

1)统一数据库连接模块是基于 JDBC 驱动的数据库连接驱动的二次开发,包括继承了一些驱动类、实现了 Connection 等接口,通过加载该驱动,客户端将数据发送至引擎模块,经过加解密处理后最终入库。

2)加解密引擎服务主要是对 SQL 表达式的解析采用递归算法,逐层对 SQL 语句是否符合加解密规则进行判断,如果符合加密或者解密规则进入加密或解密流程。解析过程采用迭代方式将返回字段与规则中定义的字段进行对比,以判断是否需要进行解密处理。

3)解密部分提供密钥对的提取、摘要加密、时间戳等服务,与密钥服务之间的通讯以及数据加密或解密算法,算法包括了 MD5、Base64 等。

4)作为校验加密数据,加解密引擎系统可以配置明文的校验数据库。

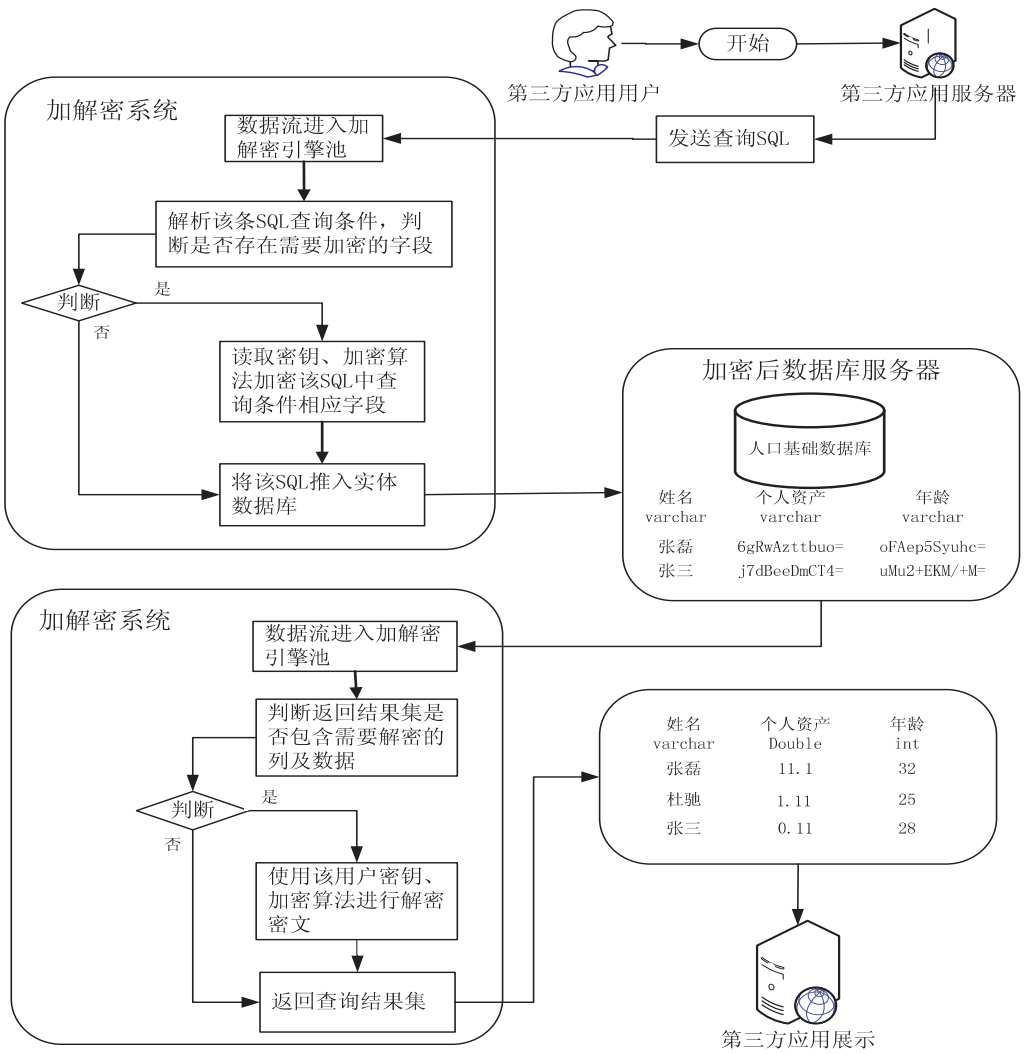

图 2　案例实现过程示意

4 应用案例

采用某个市级人口库系统作为集成对象,将加解密引擎的统一数据库系统连接驱动包添加到该人口库系统中,修改该应用的数据库驱动 URL、数据库主机信息等,使得人口库系统能够适配上加解密引擎系统,然后再配置加解密引擎系统的数据库连接部分,使加解密引擎系统能够与数据库系统进行适配,应用案例实现过程如图 2 所示。

5 结束语

加解密系统和第三方应用进行简易的集成后,加解密引擎将载入用户定义的加密算法、加密矢量等信息,并按照用户自身的密钥加密敏感数据,最终将密文数据存储到原有的数据库中,保证用户敏感数据的存储安全,由于不同用户所使用的 key 不相同,从而隔离了不同用户的业务数据。

参考文献:

[1] 怀艾芹. 基于 DBMS 外层的网络数据库加密系统的研究与设计[J]. 电脑开发与应用,2011,24(4):29-31.

[2] 陈 睿. 密文数据库系统的密钥管理与加脱密引擎[J]. 计算机与现代化,2009(7):57-59.

[3] 徐江峰,马 瑶. 一种基于动态密钥的数据库加密方案[J]. 微计算机信息,2009,25(12-3):12-13.

[4] 徐永青,徐丽珍. 数据库系统安全风险及安全策略研究[J]. 电脑知识与技术,2005(32):7-9.

[5] 侯有利. 数据库加密中的二级密钥设计[J]. 通信技术,2011,44(5):52-53.

[6] 陈铁英,陈 华,刘 瑜. 基于三层次的数据库加密应用系统[J]. 华中科技大学学报(自然科学版),2005,33(7):41-43.

[7] 叶利春. 一种应用于网络考试系统的数据库加密策略[J]. 网络安全技术与应用,2007(7):82-83.

[8] 吴兴惠,周玉萍,明秀君. 基于混合密码算法的数据库加密技术研究[J]. 海南师范大学学报(自然科学版),2010(2):161-164.

[9] 李 欣,张秉儒. 在数据库加密系统中的几个关键问题[J]. 网络安全技术与应用,2007(10):86-88.

[10] Paradies M. An efficient blocking technique for reference matching using MapReduce[J]. Datenbank spektrum,2011,11:47-49.

[11] Yang Hailong, Luan Zhongzhi, Li Wenjun, et al. MapReduce workload modeling with statistical approach[J]. Journal of grid computing,2012,10:279-310.

[12] Sauer C, Härder T. Compilation of query languages into MapReduce[J]. Datenbank spektrum,2013,13:5-15.

(上接第 142 页)

3 结束语

文中以公平硬币抛掷游戏为实例,研究了如何通过比特承诺将这类问题安全公平地应用在网络中。文中以 BCDR 系统作为比特承诺的载体设计了公平硬币抛掷协议,同时证明了协议的可行性以及安全性。在游戏协议中,操作简单,数据传输量少,两个参与者必须忠诚地执行协议。文中只针对类似硬币抛掷游戏这类游戏网络化的实现方法,在后续的研究中将致力于更多游戏的网络实现方案研究。

参考文献:

[1] 李顺东. 现代密码学:理论、方法与研究前沿[M]. 北京:科学出版社,2008.

[2] Blum M. Coin flipping by telephone protocol for solving impossible problems[C]//Proc of ACM SIGACT news. New York, USA:[s. n.],1983:23-27.

[3] Blum M, Feldman P, Micali S. Non-interactive zero-knowledge and its applications[C]//Proc of the twentieth annual ACM symposium on theory of computing. New York, USA:[s. n.],1988:103-112.

[4] 佘 堃,沈 仟,周明天. 背包问题在硬币抛掷协议上的研究[J]. 电子科技大学学报,2003,32(4):417-419.

[5] 杨 威,黄刘生,王启研. 基于椭圆曲线的三方比特承诺[J]. 电子与信息学报,2009,31(5):1049-1053.

[6] 郑 东,张 彤,陈克非,等. 基于比特承诺的电子彩票方案[J]. 电子学报,2010,28(10):141-142.

[7] Pallier P. Public-key cryptosystems based on composite degree residue classes[C]//Proc of cryptology-EUROCRYPT'99. Prague, Czech Republic:[s. n.],1999:223-238.

[8] Nisan N, Rosen A. Algorithmic mechanism design[C]//Proc of the thirty-first annual ACM symposium on theory of computing. New York, USA:[s. n.],1999:129-140.

[9] 刘小梅,田彦涛,杨 茂. 基于博弈论的多机器人任务分配算法[J]. 吉林大学学报(信息科学版),2010,28(3):256-263.

[10] 向永红,张春霞,张建军. 计算理论研究的核心问题与方向[J]. 计算机与现代化,2000(1):10-15.

[11] 苏金树,张博锋,徐 昕. 基于机器学习的文本分类技术研究进展[J]. 软件学报,2006,17(9):1848-1859.

[12] 邵立松,窦文华. 自相似网络通信量模型研究综述[J]. 电子与信息学报,2005,27(10):1671-1676.

[13] Andrews M, Dinitz M. Maximizing capacity in arbitrary wireless networks in the SINR model:Complexity and game theory[C]//Proc of 28th IEEE INFOCOM. Rio de Janeiro, Brazil:[s. n.],2009:1332-1340.

应用于数据库安全保护的加解密引擎系统

作者： 王茜， 朱志祥， 史晨昱， 张磊， WANG Qian， ZHU Zhi-xiang， SHI Chen-yu， ZHANG Lei

作者单位： 王茜,史晨昱,张磊,WANG Qian,SHI Chen-yu,ZHANG Lei(西安未来国际信息股份有限公司,陕西 西安,710063)， 朱志祥,ZHU Zhi-xiang(西安邮电大学,陕西 西安,710121)

刊名： 计算机技术与发展

ISTIC

英文刊名： Computer Technology and Development

年，卷(期)： 2014(1)