运营技术广角

一种基于保形加密的大数据脱敏系统实现及评估

卞超轶[1,2]，朱少敏[1]，周涛[1]

（1. 北京启明星辰信息安全技术有限公司，北京 100193；

2. 北京邮电大学，北京 100876）

摘　要：数据脱敏，是指对数据中包含的一些涉及机密或隐私的敏感信息进行特殊处理，以达到保护私密及隐私信息不被恶意攻击者非法获取的目的。保形加密是众多数据脱敏技术的一种，但其具有保持原始数据格式不变的重要优势，从而在一定程度上对上层应用透明。随着大数据时代的到来以及 Hadoop 平台的广泛应用，传统的基于关系型数据库的数据脱敏技术已不能满足实际的生产需要。针对 Hadoop 大数据平台实现了一种基于保形加密的数据脱敏系统，支持对多种数据存储格式以及纯数字、纯字母或数字—字母混合等多种数据类型敏感数据的加密脱敏处理。然后对 3 种不同的实现方式进行了探讨，并开展了一系列实验对系统的加密脱敏性能进行详细的评估比较。

关键词：大数据；数据脱敏；保形加密；系统；评估

中图分类号：TP309.2　　**文献标识码**：A

doi: 10.11959/j.issn.1000-0801.2017059

Implementation and evaluation of big data desensitization system based on format-preserving encryption

BIAN Chaoyi[1,2], ZHU Shaomin[1], ZHOU Tao[1]

1. Beijing Venus Information Security Technology Incorporated Company, Beijing 100193, China

2. Beijing University of Posts and Telecommunications, Beijing 100876, China

Abstract: Data desensitization is a process that makes some special transformations on sensitive data in order to protect the secrecy and privacy from being acquired by malicious attackers. Format-preserving encryption is one of the techniques of data desensitization, which has the advantage of keeping data format unchanged so that the upper layer applications are not affected. Along with the coming of big data and the wide application of the Hadoop platform, data desensitization techniques for traditional relational database management systems cannot satisfy the need of production. A data desensitization system based on format-preserving encryption for Hadoop platform was implemented, which provided the encryption support for multiple data storage formats and data value types. Moreover, three different sorts of implementations were discussed, and a series of experiments were carried out to evaluate the performance.

Key words: big data, data desensitization, format-preserving encryption, system, evaluation

收稿日期：2016-10-20；**修回日期**：2017-02-24

1 引言

数据脱敏，是指对数据中包含的秘密或隐私信息，如个人身份识别信息、商业机密数据等进行特殊处理，以达到数据变形的效果，使得恶意攻击者无法从经过脱敏处理的数据中直接获取敏感信息，从而实现对机密及隐私的防护。在金融、医疗、电信、电力等诸多行业，数据脱敏都有着非常广泛的应用。例如，在电力系统中，用户用电信息中就包含着很多重要的敏感数据，一旦泄露出去，就可能被不法分子利用来分析用户行为以及电网的组织结构等，因此在存储、传输及共享时必须进行脱敏处理。根据数据脱敏的效果，可以将其分为两大类——可恢复类和不可恢复类。可恢复类指经过脱敏处理的数据可以通过一定的方式恢复成原始数据，以各种加解密算法为代表；不可恢复类则是指经过脱敏处理的数据无法复原，如模糊、掩盖等。不可恢复类脱敏主要用于数据的共享与公开，而可恢复类则同时可用于静态存储和动态传输时数据安全隐私的防护。因此，可恢复类数据脱敏技术具有更加广泛的应用场景。保形加密(format-preserving encryption，FPE)属于可恢复类数据脱敏技术的一种，它的特点是密文与原文具有相同的数据格式，从而具备对上层应用透明的优势。

随着大数据时代的到来，以 Hadoop 为代表的大数据平台被广泛应用，而针对关系型数据库的脱敏技术及产品不能直接沿用至新型的大数据平台。虽然保形加密算法已经较为成熟，但是将其应用于大数据平台的研究和产品还很少见。因此，本文针对 Hadoop 平台实现了基于保形加密的大数据脱敏系统，支持包含 HDFS 文件、HBase 表、Hive 表等多种不同的数据存储格式，能够高效完成对纯数字、纯字母以及数字—字母混合 3 种不同类型数据的脱敏操作。同时，还尝试了几种不同的实现方式，并在实验平台上开展了相应的测试来评估比较加密处理的性能。

首先针对 Hadoop 大数据平台设计了一种保形加密机制，能够将 Hadoop 平台上多种存储形式和数据类型的敏感数据进行脱敏处理，并达到保留数据格式不变的效果；然后采用了多种不同的实现方式达到同样的数据脱敏效果，包括简单的单机处理模式、ETL(extract-transform-load，抽取—转换—加载)工具模式、Spark 并行处理模式，可以适用于不同的场景，满足不同的需要；最后在实际的 Hadoop 集群上开展一系列实验对多种实现方式及数据规模进行了详细的性能评估，比较了不同场景下的性能差别，同时也验证了系统用于实际生产环境下大数据脱敏的可行性。

2 研究背景及相关工作

对研究背景及相关工作进行具体的描述，主要包含对保形加密与 Hadoop 大数据平台的介绍。

2.1 保形加密

保形加密(也称为保留格式的加密)是一类特殊的对称加密机制，它最主要的特点就是保证密文的格式与加密前的明文格式完全相同，例如，对由 16 位数字组成的银行卡号进行加密后仍为 16 位数字，从而具有无需更改数据库范式以及对上层应用透明的优势。保形加密可用于数据的掩盖，并可通过调节加密的位数来实现不同的访问控制粒度。

学术界在保形加密领域的研究关注已经持续了 10 多年。2002 年，Black 和 Rogaway 首次从密码学的角度对保形加密进行了研究[1]，关注于整数域上的保形加密问题，并提出了 3 种构造加密机制的方法：Prefix、Cycle-walking 及 Generalized-Feistel。这 3 种方法中均利用了分组加密算法来产生伪随机置换，因为虽然真随机置换是一种理想的保形加密机制，但对于数域较大的场景预先生成并记忆随机置换表在实际中是不可行的。研究证明了保形加密的安全性与构造中所使用的分组加密算法的安全性相同。后续研究提出了一系列的加密算法及模型，其中比较典型的有 FFSEM[2]、FFX[3]、RtE[4]、BPS[5]等。在这些算法及模型中，Feistel 网络得到最为广泛的采用，因为它具有可证明的安全性，得到了更多认可。Feistel 网络是分组加密算法(如 DES)中经常采用的对称加解密结构，包含多轮的迭代过程，其中每一轮都需要一个伪随机数值作为输入，通常用 AES 来产生。

美国国家标准与技术研究院 (National Institute of Standards and Technology，NIST)针对保形加密发布了相关的标准草案——SP800-38G[6]，并给出了 3 种具体的加密算法：FF1、FF2 及 FF3。这些算法的主体流程是类似的，其核心均为一个 Feistel 网络结构，如图 1 所示。图 1 中绘出了 3 轮迭代过程的示意：在每一轮中数据被划分成两段——A_i 及 B_i，B_i 在经过函数 F_K 变换后再与 A_i 相加得到下一轮的 B_{i+1}，而下一轮的 A_{i+1} 则为本轮的 B_i。其中，函数 F_K 中包含了 AES 的加密运算，K 表示加密密钥。F_K 函数还需要 3个额外的输入——基数 n、tweak 值 T 以及当前迭代轮数。标

2017059-2

准草案中给出的 3 种不同算法主要在于 F_K 函数的不同形式以及迭代轮数。

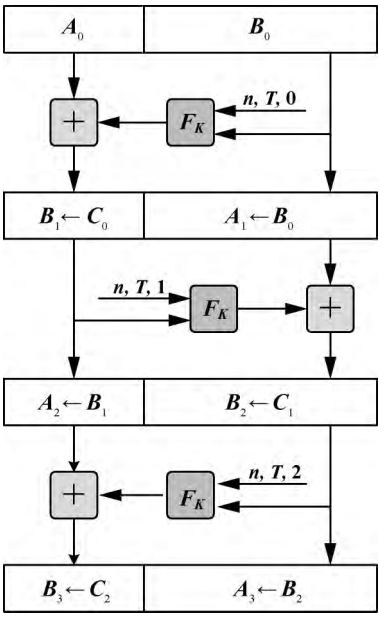

图 1　Feistel 网络结构示意

国内的研究学者在保形加密方面也开展了相应的工作[7-10]，主要是利用 Feistel 网络来设计构造新的加密算法，将算法的适用性范围扩展到任意分组长度、任意字符集以及变长编码字符集（如同时包含中英文字符的数据），从而可以对更多类型的数据进行加密。

将保形加密应用于数据脱敏在传统的关系型数据库上已经相对成熟，但在大数据平台方面的工作还很少见，仅有个别最新产品（如 HP security voltage[11]）提供了相关功能以支持 NIST 标准草案中的保形加密算法。本文工作尝试弥补这一方面的不足，开发实现了面向 Hadoop 大数据平台的保形加密系统，并评估比较了不同实现方式的加密性能，为在实际生产环境中应用提供重要参考价值。

2.2　Hadoop 大数据平台

Hadoop[12]是由 Apache 软件基金会负责开发及维护的开源软件框架，主要目标是针对大数据的分布式存储及分布式处理。Hadoop 的核心由分布式存储组件 HDFS 与运算处理组件 MapReduce 组成。

HDFS 是一种分布式的文件系统，它将文件分块并分布式地存储到多个数据节点（datanode）上，由元数据节点（namenode）负责管理文件系统的命名空间并存储所有文件及文件夹的元数据信息。由于分布式的特性，HDFS 能够提供高吞吐量的数据访问，从而适合大规模数据集上的应用。同时，HDFS 提供多文件副本的冗余存储及数据校验，具有高容错性的特点。

MapReduce 是一种用于大规模数据集的并行运算模型，它由 map（映射）与 reduce（化简）两步组成，通过多个 mappers 并行地处理键值对，从而映射成新的键值对，再将这一中间结果输出到相应的 reducers 并发地进行化简运算处理以得到最终结果。MapReduce 最大的特点是充分利用分布式计算以提高大规模数据集的计算处理效率。

在 HDFS 文件存储管理及 MapReduce 运算处理支持的基础上，Hadoop 平台上发展出丰富的组件及多种数据管理访问方式，除了基本的 HDFS 文件外，广泛使用的还包括列式存储的 HBase 和用类似关系型数据库中以表结构存储、SQL 查询语言访问管理的 Hive 等。

然而，MapReduce 也存在一些缺点，其中在性能方面的一个重要不足是其需要将每步处理的中间结果通过硬盘进行中转，从而带来大量的硬盘 I/O 开销。针对此问题，UC Berkeley（美国加利福尼亚大学伯克利分校）的研究者开发了 Spark[13]通用并行计算框架及平台。Spark 在存储方面沿用 HDFS，主要是重新实现了分布式计算部分，将中间计算结果通过内存中转，从而大幅提升了计算处理的效率。

本文工作面向 Hadoop 大数据平台，支持 Hadoop 平台上的多种数据存储管理方式，并且探讨了多种不同的系统实现方式，其中包含利用相对更高效的 Spark 并行计算框架以提升加密效率。

3　保形加密大数据脱敏系统

本节对保形加密大数据脱敏系统进行具体描述，并对一些重要的实现细节给出说明。

3.1　概述

本文共尝试了 3 种不同的系统实现方式，分别是简单单机模式、ETL 工具模式以及 Spark 并行模式。这 3 种模式均是面向 Hadoop 大数据平台上存储的数据，区别主要在于核心的计算流程。简单来说，单机模式是先将数据从 Hadoop 平台上导出，然后再使用单机程序进行数据加密操作；ETL 工具模式是利用支持 Hadoop 平台的 ETL 工具作为媒介，形成“导出—加密—输出”的流水线操作；Spark 并行模式则是直接使用 Spark 并行计算框架进行开发，将加密操作以 Spark 作业的方式提交到集群上运行。在这些实现方式中，采用的保形加密算法是经过简单修改的

NIST 标准草案中的 FF1 算法[6]。

3.2 保形加密算法

为了同时支持纯数字、纯字母及数字—字母混合这3种类型数据的加密操作，对 NIST 标准草案的 FF1 算法[6]进行了简单修改。在给定分组加密密钥 K、基数 n 及 tweak 值 T 时，FF1 算法能够对明文 P 进行保形加密，默认 n=10 以下字符集对应数字 0~9，再往上增长则依次对应英文字母 a~z，如 16 进制对应的字符集为{0,1,⋯,9,a,b,⋯,f}。该算法给出了在 FF1 算法基础上进行简单修改后的保形加密算法整体流程的伪代码描述。

输入　明文 P、FF1 加密算法 F、分组加密密钥 K、基数 n、tweak 值 T

输出　密文 C

(1)判断基数 n 是否不大于 10，或者等于 36

(2)　如果是，则 $C=F(n,K,T,P)$，返回

(3)　如果不是，则再判断 n 是否等于 26

(4)　　如果是，则

(5)　　P^*=Map(P)

(6)　　$C^*=F(n,K,T,P^*)$

(7)　　C=InverseMap(C^*)，返回

(8)　　如果不是，返回基数设置异常错误

算法通过对基数 n 的设置来调节所支持的字符集，n 的取值范围是{1,2,3,⋯,10,26,36}(其中 n=1 表示字符集只包含一个数字 0，没有意义)。举例来说，n=10 表示加密数域是十进制数，也就是字符集为数字(即 0~9)；n=36 表示加密数域是 36 进制数，从而支持字符集为数字及英文字母混合(即 0~9、a~z)；而为了支持纯英文字母字符集(即 a~z)的加密，可令 n=26，此时原本对应的字符集为 0~9、a~p，所以需要在加密前及加密后附加进行一次额外的映射操作(第(5)行和第(7)行)，将其转换成 a~z。

算法是不区分大小写字母的，但可对其进行进一步的扩充，也就是说可以将同时包含数字及大小写字母的字符集看作 62 进制数域，再进行相应的字符映射即可；类似地，还可以继续扩充以支持更大的字符集，如全体 ASCII 字符。为了简单而不失代表性，本文只实现了以上算法，进一步的扩展支持工作将在后续研究中完成。

FF1 算法是一种对称加密算法，其解密过程与加密过程是相同的，因此基于其的算法也是如此，这里就不再介绍算法的解密部分，而在接下来的具体实现方式描述以及之后的实验评估部分也将略去对解密操作的说明。

3.3 简单单机模式

简单单机模式是 3 种模式中最为简单、直接的系统实现方式，其思路是将存储在 Hadoop 平台上的数据先导出保存到本地，再使用实现的保形加密算法对存储在本地文件中的数据进行加密操作，从而完成数据脱敏过程。根据存储管理方式的不同，使用了对应的 Hadoop 编程接口以支持 HDFS 文件、HBase 表及 Hive 表数据的导出。然后在单机上应用实现的算法对数据进行逐条加密。

简单单机模式的优点是简单、直观，并且可以脱机处理(数据导出后不需要再连接大数据平台)，但缺点也很明显——效率低，因为只使用了单机对数据进行串行式的逐条加密处理，既没有利用大数据平台分布式的特点，也没有在加密方面进行并行处理。

3.4 ETL 工具模式

为了提高加密的效率，可以利用 ETL 工具来实现流水线式处理以及并行度的提升。选用开源的 ETL 工具——Pentaho Data Integration(Kettle)[14]，将保形加密以转换插件方式提供，从而直接支持 Hadoop 平台上的多种数据格式。Kettle 是一款跨平台开源 ETL 工具，它使得用户可以直接通过使用可视化工具的拖拽来完成数据的导入、导出及基础转换操作，支持 Cloudera 版本(CDH)[15]、Hortonworks 版本(HDP)[16]等多种 Hadoop 发行版本。所使用的 Kettle 的版本号是 6.0.0.0-353。

保形加密插件的开发主要包含两大部分，即加密算法和交互界面。加密算法即上述的算法，而交互界面的作用主要是与用户进行交互，提供一些参数配置功能，包括明文列名、密文列名、密钥配置(指定密钥或随机密钥)、tweak 配置(指定 tweak 值或随机 tweak 值)及基数等。基于 Kettle 实现的保形加密系统主界面以及保形加密转换插件配置界面如图 2 所示，这也正是 ETL 工具模式的另一个优点——良好的用户交互图形界面。相对而言，其他两种模式的系统实现仅能通过终端参数指定来进行简单的交互。

ETL 工具模式对保形加密的效率有两方面的提升。第一个方面是“数据导出—加密脱敏—结果存储”这条流水线的形成，即数据源源不断地从 Hadoop 大数据平台流出并进入保形加密模块进行脱敏处理，然后再紧接着输出到指定位置。第二个方面则是其支持并行处理：在 Kettle 的单机运行模式上，可以设置保形加密转换步骤的并发数，从而利用多核心处理器的并发处理优势；Kettle 还可以组

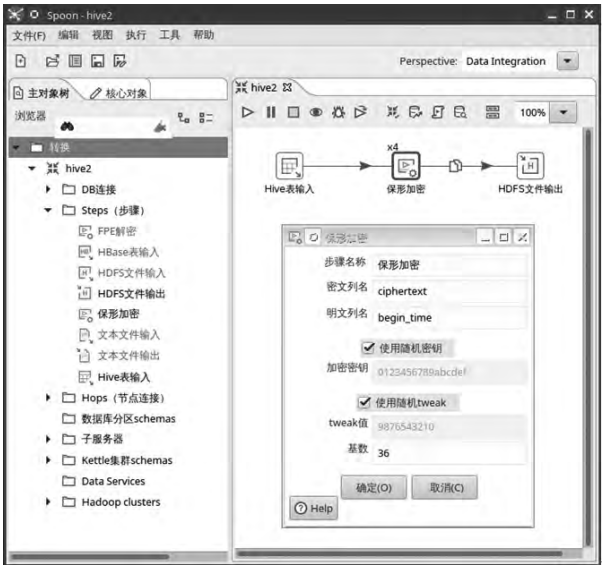

图 2　基于 Kettle 的 ETL 工具模式系统实现界面

织成集群模式，通过将数据分发到集群的各个主机上，实现多主机并行处理的效果。

3.5　Spark 并行模式

为了进一步利用 Hadoop 平台在分布式存储及并行计算方面的优势，可以直接基于并行计算框架实现保形加密操作。相较于 MapReduce，Spark 利用内存计算避免了低效的硬盘输入/输出操作，从而具有更高的计算效率。因此，基于 Spark 实现的并行模式将为大数据规模的保形加密效率带来实质性的提升。

具体地，使用 Spark 的 Scala 编程接口实现对 Hadoop 平台上大数据的加密脱敏处理过程为：首先读取 Hadoop 平台上存储的数据(HDFS 文件、HBase 表、Hive 表等)形成 Spark 的数据抽象——弹性分布式数据集 (resilient distributed dataset，RDD)，然后再应用 Spark 提供的并行计算编程接口，在分布式集群上对所有 RDD 执行并行的加密操作。

在此种模式下，保形加密操作是以 Spark 作业的方式呈现的，通过将其提交到分布式计算平台上执行来完成对数据的脱敏处理。因此，针对 Spark 的参数调优对于此模式的运行效率有一定影响。

4　实验评估

为了评估所实现的保形加密大数据脱敏系统的性能，并比较第 2 节所提到的 3 种模式的差别，在实际的 Hadoop 大数据平台上开展了一系列实验，本节将描述这些实验内容及评估比较的结果。

4.1　实验环境

采用的 Hadoop 平台是由 3 台戴尔 PowerEdge R720 服务器组成的小集群，采用的 Hadoop 版本是 CDH 5.4。在保形加密系统的简单单机模式和 ETL 工具模式中连接 Hadoop 集群的主机是一台联想 ThinkPad T440p 笔记本电脑。服务器与笔记本电脑的 CPU 及内存的具体参数见表 1。

表 1　CPU 及内存参数

	戴尔 PowerEdge R720	联想 ThinkPad T440p
CPU	Intel Xeon CPU E5-2603 v2 @ 1.80 GHz，8 核心	Intel Core i5-4210M CPU @ 2.60 GHz，4 核心
内存	64 GB DDR3 1 600 MHz	4 GB DDR3 1 600 MHz

4.2　实验分析

对第 2 节所述的 3 种模式实现的系统都开展了相应的实验进行性能评估。列出的所有实验数据均是在同样的系统环境下 10 次独立重复实验的平均结果。

首先，测试了一些不同参数设置的影响。以 ETL 工具模式为例，测试了不同并发数、不同数据规模等场景下保形加密的性能。

不同并发数设置下 ETL 工具模式在 Thinkpad 笔记本电脑上单机执行时的保形加密性能比较结果如图 3 所示，其中使用的数据规模是 10M 条(即 10^7 条)数据。由图 3 可知，将并发数设置成计算机所具有的 CPU 核心数 (本例中为 4)时达到的性能最高——处理速度约为 1.6 万条/s。

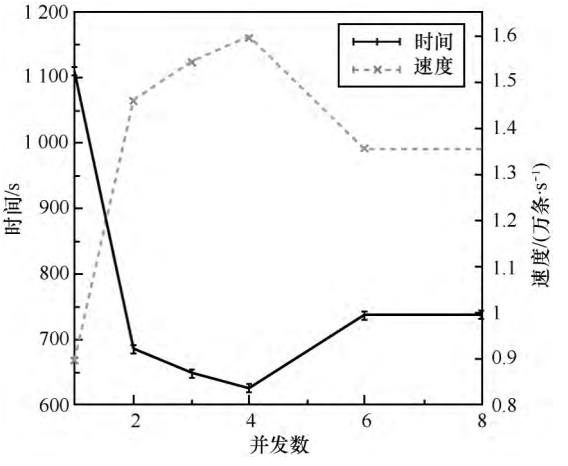

图 3　保形加密性能比较结果(不同并发数设置)

不同数据规模下 ETL 工具模式在 Thinkpad 笔记本电脑上单机执行的处理性能对比结果如图 4 所示，其中，并发数设定为 4，数据规模由最小的 1 万(10^4)条一直增大到 1 亿(10^8)条。图中的 x 轴(数据条数)和 $y1$ 轴(即左侧的 y 轴，

时间)均为对数尺度。由图 4 可知,完成加密脱敏处理的时间随数据规模的增大而基本呈线性增长趋势,处理的速度在数据规模达到 10^6 之后维持稳定。数据规模较小时处理速度较慢,其原因可能是初始连接 Hadoop 集群读取数据到保形加密的流水线启动期间执行相对较慢。

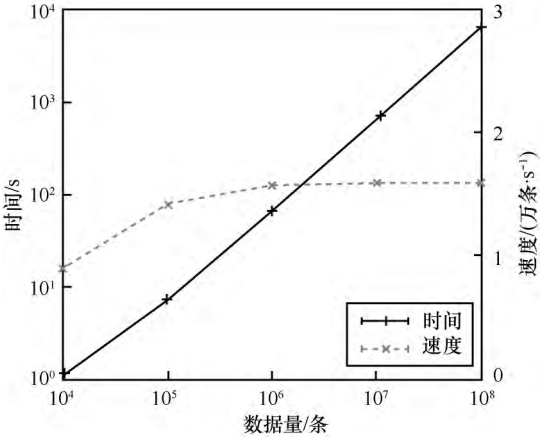

图 4 保形加密性能比较结果(不同规模设置)

同时,还开展了对不同类型数据(即基数设置不同)的加密性能评估比较,结果证明加密性能基本相同,即对纯数字、纯字母或数字—字母这 3 种类型的数据具有相同的加密性能,此处略去相关的结果。

类似地,对于简单单机模式和 Spark 并行模式也评估了一些不同参数设置下的性能变化。由于篇幅的限制,这里不再一一给出。

然后,开展实验来测评 3 种模式对相同规模数据进行加密脱敏处理的性能差别。实验中其他参数设置均为最优(如 ETL 工具中保形加密的并发数、Spark 作业提交的相关参数等)。3 种模式在不同数据规模下的加密处理速度对比见表 2。由表 2 可知,在这 3 种模式中,Spark 并行模式的加密处理速度最快,而且随着数据规模的增大,其处理速度还会有所提升,其主要原因是在数据规模较小时 Spark 的并行优势还没有得到充分发挥。相比较来看,另外两种模式的加密处理速度在不同数据规模下基本维持稳定。总体来看,ETL 工具模式的加密处理速度约为简单单机模式的 3 倍,而 Spark 并行模式的处理速度在大数据规模下(100M,即 1 亿条数据)更能达到简单单机模式的 16 倍之多。

为了更清晰地展示 3 种模式的加密效率差别,不同数据规模下 3 种模式的总执行时间对比如图 5 所示,图例中“simple”表示简单单机模式,“ETL”表示 ETL 工具模式,“Spark”表示 Spark 并行模式。

表 2 3 种模式加密处理速度对比

数据量/条	处理速度/(条·s⁻¹)		
	简单单机模式	ETL 工具模式	Spark 并行模式
10^6	4 847.5	15 923.6	23 888.9
10^7	5 050.2	15 969.3	68 426.5
10^8	4 967.0	15 860.4	83 151.0

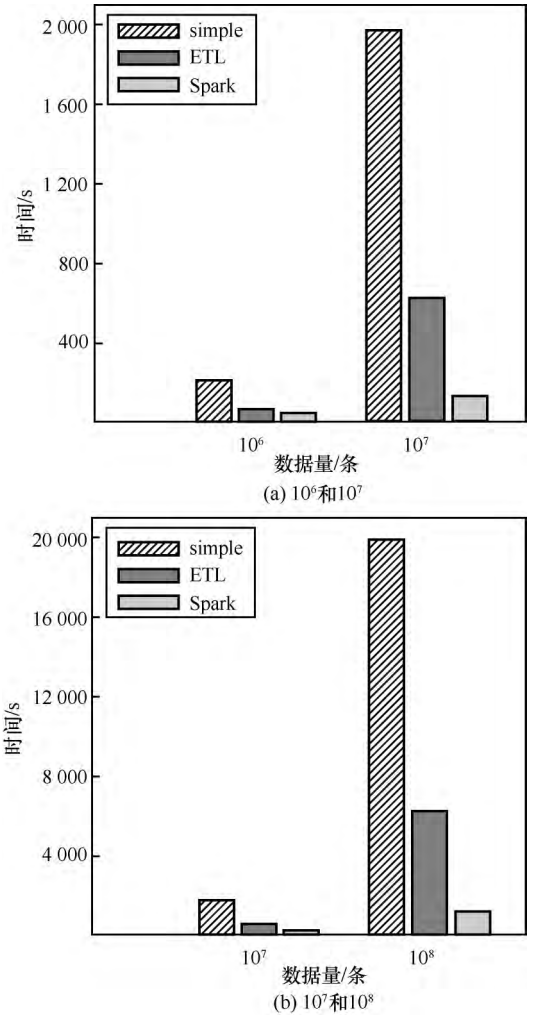

图 5 3 种模式总执行时间比较

从这一结果中也能得出,基于保形加密的大数据脱敏系统性能可以满足实际生产需要的结论。具体来说,对于 10^8 规模的数据(即 1 亿条),使用 ETL 工具模式处理仅需要花费约 105 min,而使用 Spark 并行模式更是只需要花费约 20 min,而且这只是在一台配备四核处理器的笔记本电脑单机 ETL 模式以及仅由 3 台服务器组成的 Spark 集群下的测量结果。若在实际生产环境中采用集群模式的 ETL 或者更大规模的 Spark 集群,则必然能取得更高的加密脱敏效率。

5 结束语

针对大数据环境下的脱敏问题设计了一种面向 Hadoop 平台的基于保形加密的解决方案，并完成了具体的系统实现工作。该系统支持 Hadoop 平台下的多种数据存储格式，如 HDFS 文件、HBase 表、Hive 表等，可以对纯数字、纯字母及数字—字母混合等多种类型的敏感数据完成保形加密的脱敏操作。给出了 3 种不同的实现方式，即简单单机模式、ETL 工具模式及 Spark 并行模式，它们有着各自的优/缺点和适用场景。在实际的 Hadoop 平台上，开展了一系列实验来评测 3 种实现方式的系统性能，结果验证了系统在实际生产环境中的可行性，也对 3 种模式的使用选择有重要的指导意义。在后续研究工作中，将尝试扩展算法使其支持更多种的数据类型，并深入测试 ETL 工具模式中使用集群工作模式的效果以及 Spark 参数调优的具体影响。

参考文献：

[1] BLACK J, ROGAWAY P. Ciphers with arbitrary finite domains [M]. Berlin Heidelberg: Springer, 2002.

[2] SPIES T. Feistel finite set encryption mode[J/OL]. NIST Proposed Encryption Mode, 2008:1-10. (2008-01-24) [2016-07-01]. https://static.aminer.org/pdf/PDF/000/217/259/about_feistel_schemes_with_six_or_more_ rounds.pdf.

[3] BELLARE M, RISTENPART T, ROGAWAY P, et al. Format-preserving encryption[C]//Selected Areas in Cryptography, March 4-9,2009, Berlin, Germany. Berlin Heidelberg:Springer, 2009: 295-312.

[4] BELLARE M, ROGAWAY P, SPIES T. The FFX mode of operation for format-preserving encryption [J]. Unpublished Nist Proposal, 2010, 136(9):633.

[5] BRIER E, PEYRIN T, STERN J. BPS: a format-preserving encryption proposal[J/OL]. NIST submission, 2010: 1-11. (2010-04-04) [2016-07-01]. http://csrc.nist.gov/groups/ST/toolkit/BCM/documents/ proposedmodes/bps/bps-spec.pdf.

[6] DWORKIN M. Recommendation for block cipher modes of operation: methods for format-preserving encryption[J]. NIST Special Publication, 2013(800): 38.

[7] 刘哲理, 贾春福, 李经纬. 保留格式加密模型研究[J]. 通信学报, 2011, 32(6): 184-190.
LIU Z L, JIA C F, LI J W. Research on the format-preserving encryption modes[J]. Journal on Communications, 2011, 32(6): 184-190.

[8] 刘哲理, 贾春福, 李经纬. 保留格式加密技术研究[J]. 软件学报, 2012, 23(1): 152-170.
LIU Z L, JIA C F, LI J W. Research on the format-preserving encryption techniques[J]. Journal of Software, 2012, 23(1): 152-170.

[9] 李敏, 贾春福, 李经纬, 等. 变长编码字符型数据的保留格式加密[J]. 吉林大学学报:工学版, 2012, 42(5): 1257-1261.
LI M, JIA C F, LI J W, et al. Format-preserving encryption for variable-length encoding character data [J]. Journal of Jilin University: Engineering and Technology Edition, 2012, 42(5): 1257-1261.

[10] 李经纬, 贾春福, 刘哲理, 等. 基于 k- 分割 Feistel 网络的 FPE 方案[J]. 通信学报, 2012, 33(4): 62-68.
LI J W, JIA C F, LIU Z L, et al. FPE scheme based on k-splits feistel network [J]. Journal on Communications, 2012, 33(4): 62-68.

[11] HP. HP security voltage[EB/OL]. (2015-02-09)[2016-03-01]. https://saas.hpe.com/en-us/software/voltage-data-encryption-security.

[12] Apache Software Foundation. Apache Hadoop[EB/OL]. (2011-12-10)[2016-07-01]. http://hadoop.apache.org/.

[13] Apache Software Foundation. Apache Spark[EB/OL]. (2014-05-30)[2016-07-01]. http://spark.apache.org/.

[14] Pentaho. Data integration - Kettle[EB/OL]. (2009-05-14)[2016-07-01]. http://community.pentaho.com/projects/data-integration/.

[15] Cloudera. Cloudera CDH[EB/OL]. (2012-10-12)[2016-07-01]. http://www.cloudera.com/products/apache-hadoop/key-cdh-components.html.

[16] Hortonworks. HORTONW0RKS data platform (HDP)[EB/OL]. (2012-11-30)[2016-07-01]. http://hortonworks.com/products/data-center/hdp/.

[作者简介]

卞超轶(1987-)，男，北京启明星辰信息安全技术有限公司高级研究员，启明星辰博士后工作站——北京邮电大学博士后流动站联合培养博士后，主要研究方向为大数据自身安全、大数据安全分析等。

朱少敏(1983-)，男，北京启明星辰信息安全技术有限公司前线技术专家团成员，主要研究方向为电力系统信息安全、多媒体信息处理等。

周涛(1979-)，男，博士，北京启明星辰信息安全技术有限公司教授级高级工程师，主要研究方向为大数据安全分析、事件关联分析、入侵检测等。