面向智能电网边缘计算的密文多关键字检索方法

许爱东[1] 朱 静[2] 蒋屹新[1] 张宇南[1] 吴 涛[3] 蒋龙生[3]

[1](南方电网科学研究院有限责任公司 广东 广州 510670)

[2](重庆邮电大学通信与信息工程学院 重庆 400065)

[3](重庆邮电大学网络空间安全与信息法学院 重庆 400065)

摘 要 随着我国电网智能化升级的不断深入以及边缘计算技术的兴起,如何对电网边缘端存储的用电数据进行安全防护和按需检索成为实现电网边缘化智能计算的关键问题。对此提出一种新型直接索引结构方案,通过采用哈希 SHA256 算法产生一组码字数组作为记录的索引,搜索时用户输入多关键字并产生对应的陷门,该陷门与索引进行精准匹配,并将所有匹配结果反馈给用户。该方案具有较低的空间复杂度;通过利用智能电表的采样数据,建立数学模型并验证了该方法的有效性。

关键词 智能电网 哈希 密文检索 多关键字 边缘计算

中图分类号 TP309 **文献标志码** A DOI:10.3969/j.issn.1000-386x.2022.07.047

RETRIEVAL METHOD OF MULTI-KEYWORD CIPHER-TEXT FOR EDGE COMPUTING IN SMART GRID

Xu Aidong[1] Zhu Jing[2] Jiang Qixin[1] Zhang Yunan[1] Wu Tao[3] Jiang Longsheng[3]

[1](*SEPRI, China Southern Grid, Co., Ltd., Guangzhou 510670, Guangdong, China*)

[2](*School of Communication and Information Engineering, Chongqing University of Posts and Telecommunications, Chongqing 400065, China*)

[3](*Cyberspace Security and Information Law Institute, Chongqing University of Posts and Telecommunications, Chongqing 400065, China*)

Abstract With the continuous deepening of the intelligent upgrade of China's power grid and the rise of edge computing technology, how to carry out security protection and on-demand retrieval of power consumption data stored at the edge of the grid becomes a key issue for intelligent calculation of grid marginalization. This paper proposes a new direct index structure scheme. The hash SHA256 algorithm was used to generate a set of code-word arrays as the index of the record. In the process of searching, the users input multiple keywords and generated corresponding trapdoors. The trapdoor and the index were precisely matched. All the matching results were fed back to the users. This scheme could achieve lower space complexity. And we used the sampling data of the smart meter and built a mathematical model, which verified the effectiveness of the method.

Keywords Smart grids Hash Cipher-text retrieval Multiple keywords Edge computing

0 引 言

随着我国电网智能化的不断发展,电网会产生大量的采样数据,电网采样数据的采集、传输、保存需要大量的带宽与存储资源,同时中心化存储也可能会造成用户隐私信息的泄露[1]。随着边缘计算的兴起,电网终端可以更好地支撑本地实时智能化业务处理。本地采集的原始数据可以在边缘执行初始分析,只传有用数据到云端,从而减少网络负担,降低传输成本,保证数据的隐私与安全,如图 1 所示。边缘的计算资源配置可以满足小区域数据离线处理与分析,从而保障

收稿日期:2020 - 02 - 23。国家重点研发计划项目(2018YFB0904900,2018YFB0904905);国家自然科学基金项目(61802039,61772098);重庆市教育委员会科学技术研究项目(KJQN201800630)。**许爱东**,教授级高工,主研领域:电力系统、工控系统安全。**朱静**,硕士生。**蒋屹新**,教授级高工。**张宇南**,工程师。**吴涛**,讲师。**蒋龙生**,本科生。

电力数据的安全传输与处理。因此如何对电网边缘端存储的用电数据进行安全防护和按需检索成为实现电网边缘化智能计算的关键问题。

图 1　边缘化智能计算

为了保护数据隐私,Song 等[2]提出第一个基于密文扫描思想的可搜索加密 SWP 方案,该方案对单个关键字的查询需要扫描整个文件,故检索效率较低,且无索引的可搜索对称加密(Searchable Symmetric Encryption,SSE)方案需要研究部门开发专门的加密算法,目前还无法应用到智能电网中。Goh[3]提出了基于 Z-IDX 的 SSE 方案,并使用布隆过滤器作为单个文件的索引结构,然而布隆过滤器存在假阳性,当前智能电网的基础设施是无法接受的这一缺点的。文献[4]提出实现最优搜索的 2 种方案(SSE-1、SSE-2),SSE-1 方案针对选择关键字攻击是安全的,SSE-2 对于自适应选择关键字攻击是安全的,但是对于智能电网反向索引直接性更新困难,故效率过低。上述工作仅能解决单关键字密文检索的问题。

为了满足用户的查询需求,多关键字密文检索技术应运而生,文献[5]引入 K 最近邻(K-NN)技术实现移动云计算环境下的高效多关键字安全排名搜索系统(EMRS),且能保证搜索结果的准确性。现在更多的研究将 SSE 方案应用到实际问题中,文献[6]提出将可搜索加密方案应用于加密电子健康数据上,可以有效和安全地搜索电子病历。文献[7]提出对智能家居模型的可搜索加密方案,实现对智能家居数据的高效管理及有效保护。

上述方法仅适用于文本文档,但对海量的电力数据来说,还存在一些不足,鉴于此文献[8]提出了一种实用的 SSE 方案,该方案通过牺牲少量信息公开来实现更高的空间效率,但该方案仅支持单关键字检索,故检索效率较低。为此本方案提出多关键字的适合电力数据的密文检索方案,本方案通过将 AES 算法与 SHA-256 算法相结合,实现多关键字陷门与记录集合索引的精确匹配,并将搜索结果列表返回给用户。

本文的主要贡献体现在 3 个方面:

(1) 回顾分析了当前典型的 SSE 方案的思想,并讨论这些方案为什么不适用于电力数据;

(2) 提出了一种面向智能电网边缘计算的密文索引结构,将 SHA-256 算法与伪随机函数结合构建密文索引,实现高效安全检索;

(3) 解决电力系统下密文检索的多关键字检索问题,适用于智能电网边缘计算下的密文检索场景。

1　定义与背景

1.1　系统模型

本文的系统模型中涉及 3 个实体,分别为数据所有者、数据检索者和云服务器,如图 2 所示。

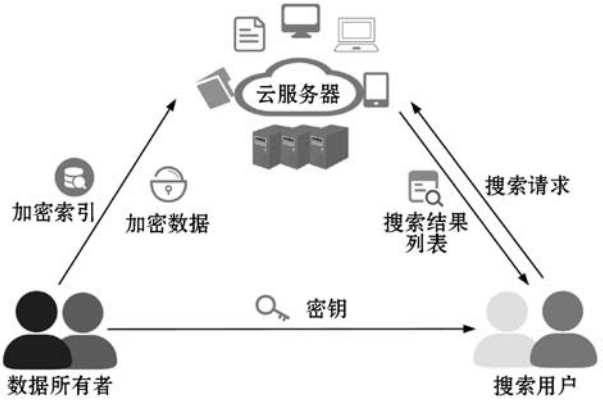


图 2　系统框架

数据所有者:负责搜集数据,建立索引,并把数据和索引加密外包到云服务器上。除此之外,数据所有者还需要给需要查询数据的使用者合理授权。

数据检索者:将生成的关键字的陷门发送给服务器,且只有得到数据所有者授权的情况下才能访问数据。

云服务器:提供了密文检索所需的大量存储空间和计算资源,云服务器一旦接到数据检索者的合法请求,匹配所有包含该搜索关键字的列表,并返回给数据检索者。

1.2　问题定义

本方案假设数据所有者是可信任的,但云服务器是诚实且好奇的[9-11],这表明云服务器会执行操作者的所有命令,同时也会尝试学习和分析服务器上的数据和索引结构,比如可能存在篡改加密数据、勾结恶意用户并发送错误的搜索结果、删除一部分加密数据,使存在更多的存储空间并以此来赚取更多的利益等不诚实的行为[12]。

本方案假设智能电网数据具有相同的数据结构,并包含同等数量的属性集合。例如,电力数据应包含年份、月份、时间、电器类型、用电类型、电表电量、功率等属性。

本方案假设存储在云服务器中的智能电网数据的记录总数要远远大于每个记录中的关键字总数,即数据集数量要远远大于关键字数量。

1.3　符号定义

本方案的相关符号定义如表 1 所示。

表 1 符号参数

符号	含义
n	数据检索者设置的属性个数($n \leqslant N$)
m	数据检索者输入的关键字个数($m \leqslant N$)
N	一条记录中的关键词总数
$w_{i,j}$	记录 R_j 中的第 j 个关键词($1 \leqslant j \leqslant n$, $1 \leqslant i \leqslant 2^d$)
$\{0,1\}^n$	n 位数字的集合
$K \xleftarrow{R} \{0,1\}^n$	表示从集合 $\{0,1\}^n$ 中均匀采样 K 个元素
$F=\{R_1,R_2,\cdots,R_{2d}\}$	表示一个集合记录
$h:\{0,1\}^* \rightarrow \{0,1\}^r$	哈希函数,将任意长度映射成 r 位哈希值
$f:\{0,1\}^k \times \{0,1\}^r \rightarrow \{0,1\}^r$	伪随机函数,将 r 位数字映射到另一个带有 k 位键的 r 位数字

2 数据检索算法

2.1 基于 SHA-256 的索引框架

该索引框架主要由 *Keygen*(s)、*BuildIndex*(R_i,K)、*Trapdoor*(K,w)、*Search*(T_w,I_i)4 个算法组成。

Keygen(s)密钥生成函数,s 为安全参数。

BuildIndex(R_i,K):索引构建函数,数据所有者输入密钥 K 和记录 R,输出记录 R 的索引 I_R。

Trapdoor(K,w):陷门生成函数,由数据检索者输入密钥 K 和想要搜索的关键字 $w_1,w_2,\cdots,w_m$,输出关键字 w 的陷门 T_{w_m}。

Search(T_w,I_i):用户查询函数。数据检索者输入关键字 $w_1,w_2,\cdots,w_m$ 的陷门 T_{w_m}和索引 I_R 文件,服务器执行匹配与查询,若 $w_1,w_2,\cdots,w_m \in R$ 则输出 1,否则输出 0。

2.2 MD5 算法

MD5 将任意长度的“字节串”转化为一个 128 bit 的散列值,且加密过程不可逆,即无法将一个 MD5 值转换回原始的字符串,以防止被篡改[13-14]。MD5 的计算速度要比其他哈希算法的计算速度要快,但 MD5 比较容易发生碰撞,且于 2005 年已经被中国密码学家王小云教授攻破,故安全性较低。

2.3 SHA-256 算法

SHA-256 算法是一个 MD 结构迭代哈希函数,该算法从分组长度是 512 位的多重分组信息中创建一个 256 位的消息摘要,其过程是不可逆的[15-16]。SHA-256 的安全性要比 MD5 高很多,且自 MD5 被破解后,SHA-256 成为当前最流行的安全加密算法。下面是常用哈希算法的特性对比如表 2 所示[17-18]。

表 2 常用哈希算法的特性对比

哈希算法	摘要	分组	步骤	运行	碰撞阈值	安全性
MD5	128	512	80	较快	$>2^{64}$	较低
SHA-1	160	512	64	一般	$>2^{80}$	一般
SHA-256	256	512	64	较慢	$>2^{128}$	较高
SHA-512	512	1 024	80	慢	$>2^{256}$	高

3 基于 SHA-256 的密文检索方案

根据系统模型,本文遵循一般的 SSE 方案[19-20]。我们假设智能网格数据是一个记录集合,每个记录包含 N 种关键字。考虑到集合中并非所有的关键字都是搜索必需的,可能数据检索者只想通过几个特定类型的关键字查询记录。因此,该方案设定数据检索者需要查询的关键字为 N 中的前 n 种属性。

(1) 在确定数据检索者需要查询的 n 种属性后,数据所有者运行 Keygen 函数获得一个 k 位的密钥 K 并将其保密。

(2) 构建记录 R_i 的索引 I_{R_i}。数据所有者通过 BuildIndex 函数调用 Trapdoor 函数,对于记录中的已确定的 n 种属性,计算 $T_{w_{i,j}}=f_K(h(w_{i,j}))$ 和码字 $X_{i,j}=\hat{f}_{T_{w_{i,j}}}(h(id(R_i)))$,其中 $\hat{f}$ 是另外一个伪随机函数,即 $\hat{f}:\{0,1\}^r \times \{0,1\}^r \rightarrow \{0,1\}^r$。对于计算得到的每一个码字 $X_{i,j}$进行随机排序,得到一个有 n 个属性的数组 I_{R_i}。然后,数据所有者将索引 I_{R_i}附加到加密的记录 R_i 上,上传到云服务器。其索引构建过程的伪代码如算法 1 所示。

算法 1 *BuildIndex*(R_i,K)

输入:file collection R, key K, file stored keyword type list。

输出:file encrypted index I_{R_i}。

1. Allocate an n elements empty array I_{R_i}, and set all elements to 0
2. **for** each keyword type $w_{i,j}$ in R_i **do**
3. compute $T_{w_{i,j}}=f_K(h(w_{i,j}))$
4. compute $X_{i,j}=\hat{f}_{T_{w_{i,j}}}(h(id(R_i)))$
5. random rank $X_{i,j}$, update $X_{i,j}$
6. set $I_{R_i}=X_{i,j}$
7. **end for**

(3) 生成多关键字 $w_1,w_2,\cdots,w_m$ 的陷门 T_{w_m}。该方案选定记录中的前 10 个属性记为一个码字数组,作为记录 R 的索引 I_R。数据检索者输入关键字 w_1,$w_2,\cdots,w_m$ 后,通过 Trapdoor 函数计算 $w_1,w_2,\cdots,w_m$ 的散列值 $h(w_1),h(w_2),\cdots,h(w_m)$,其中 $h:\{0,1\}^* \rightarrow \{0,1\}^r$,然后计算并输出陷门 $T_{w_m}=f_K(h(w_1),$

$h(w_2),\cdots,h(w_m))$,其中 f 为伪随机函数,即 $f:\{0,1\}^k\times\{0,1\}^r\rightarrow\{0,1\}^r$。

(4) 搜索包含多关键字 $w_1,w_2,\cdots,w_m$ 的数据列表,数据检索者输入要查询的多关键字 $w_1,w_2,\cdots,w_m$,并计算陷门 $T_{w_m}=f_K(h(w_1),h(w_2),\cdots,h(w_m))$,然后将 T_{w_m} 发送到云服务器。云服务器接收到 T_{w_m} 后,对每条密文记录 $R_i(1\leqslant i\leqslant 2^d)$ 计算码字 $X_{w_1,w_2,\cdots,w_m}=\hat{f}_{T_{w_m}}(h(id(R_i)))$,并检查索引 I_{R_i} 中是否包含码字 $X_{w_1,w_2,\cdots,w_m}$。如果索引中包含码字 $X_{w_1,w_2,\cdots,w_m}$,云服务器将 $id(R_i)$ 列表返回给数据检索者,否则,将空列表返回给数据检索者。其搜索过程的伪代码如算法 2 所示。

算法 2　$Search(T_w,I_i)$

输入:file encrypted index I_{R_i}, trapdoor T_{w_m}。

输出:search result list。

1. **for** row in range (I_{R_i}. shape[0]) **do**
2. Compute $X_{w_m}=\hat{f}_{T_{w_m}}(h(id(R_i)))$
3. $X=set(X_{w_m})$ and $I=set(I_{R_i})$
4. Determines whether set I. shape contains set X
5. **if** $X\subseteq I$. shape **then**
6. return list $id(R_i)$
7. **else**
8. return empty list
9. **end if**
10. **end for**

一个实际的密文检索方案应该能在数据更新时处理索引更新。本文是建立在直接索引的基础上对数据进行处理,故索引是动态的且易于更新。数据所有者只需要重新运行 BuildIndex 函数,就可获得数据更新之后的新索引,并将索引重新存储到云服务器中。

4　实验结果分析

本文所述文献的索引类型、搜索复杂度性能对比如表 3 所示[21-22]。本文旨在提高密文检索的效率与安全性,使攻击者无法从索引中取得任何有关明文记录的信息。由于码字是随机插入的,故用户搜索复杂度为 $O(2^d\cdot n)$,而数据检索者设置的属性个数 n 在实际应用中认为是非常小的常数,故该搜索方案有效。

表 3　SSE 方案特性对比

SSE 方案	索引	更新	搜索复杂度
SWP[2]	无	简单	$O(2^d\cdot N)$
Z-IDX[3]	直接	简单	$O(2^d)$
SSE-1[4]	倒序	困难	$O(1)$
EMRS[5]	直接	困难	$O(2^d)$
本文方案	直接	简单	$O(2^d\cdot n)$

4.1　安全分析

1) 数据保密性。在本文中,外包的电力数据被传统的对称加密 AES 算法加密。文献[23]中已经证明 AES 加密算法是安全的,任何实体没有密钥都无法恢复加密数据,故加密数据的保密性可以实现。

2) 索引的隐私保护。该方案用 SHA-256 算法对码字进行哈希处理,并用伪随机函数将码字随机写入索引数组中,可以保证攻击者无法从索引中学到原始的关键字,即保证索引的确定性和查询的保密性。

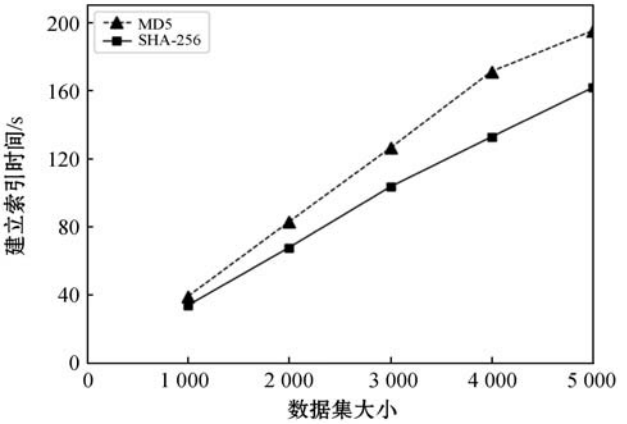
3) 陷门的不可链接性。云服务器能够通过分析关键字陷门来推断识别关键字,鉴于此本文的码字是由引入记录的标识符来构建的,即记录中的相同关键字在索引中具有不同的码字。因此实现了陷门的不可链接性。

4.2　实验数据

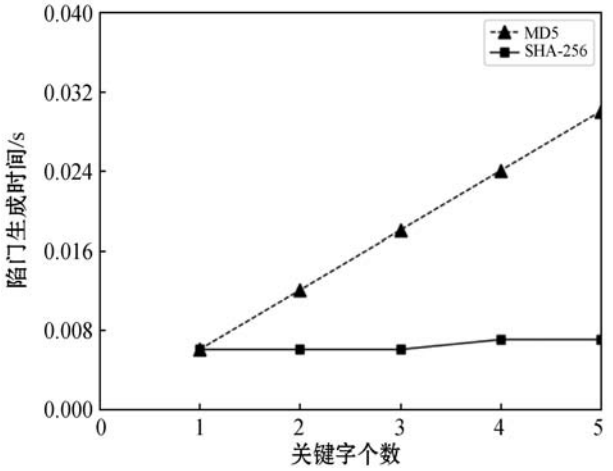
本实验的测试数据采用美国能源信息管理局(EIA)提供的 AIM 数据,AIM 数据表示为 2016 年度美国电力公司每月统计的电力消耗等 20 多个属性,本实验采用该数据集验证本文方案的有效性。

4.3　性能分析

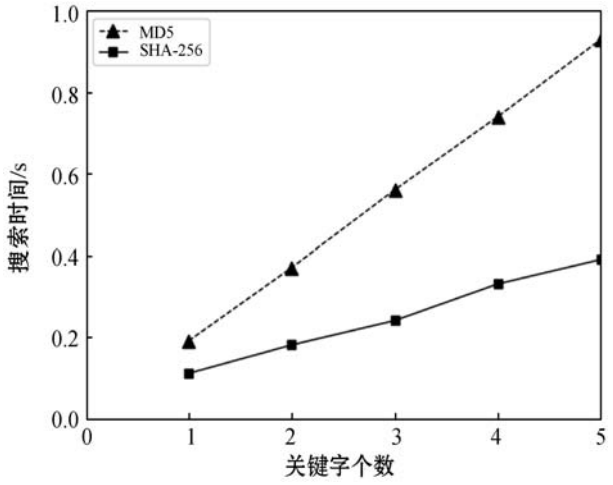
为了验证本文方案有更高的搜索效率,对改进方案与未改进方案进行对比实验,主要从索引构建时间、陷门生成、搜索效率进行比较,其实现结果如图 3 所示。可以看出,图 3(a)中未改进方案构建索引时间比改进后所需时间要稍长一点。图 3(b)中未改进方案关键字陷门生成时间呈线性,而改进后的方法近乎平行。图 3(c)中未改进方案的搜索时间要比改进后搜索时间明显长很多,因此,改进后的方案在索引建立、陷门生成、搜索效率都有了很大的提升。



(a) 索引建立时间对比



(b) 陷门生成时间对比



(c) 用户搜索时间对比

图 3 索引构建、陷门生成、查询效率对比

本文在实验室电力边缘计算安全防护平台上测试其可行性。硬件设备如图 4 所示,采用一台具有超高性能、低功耗性能的超级计算机模块——Jetson TX2;软件界面展示如图 5 所示。

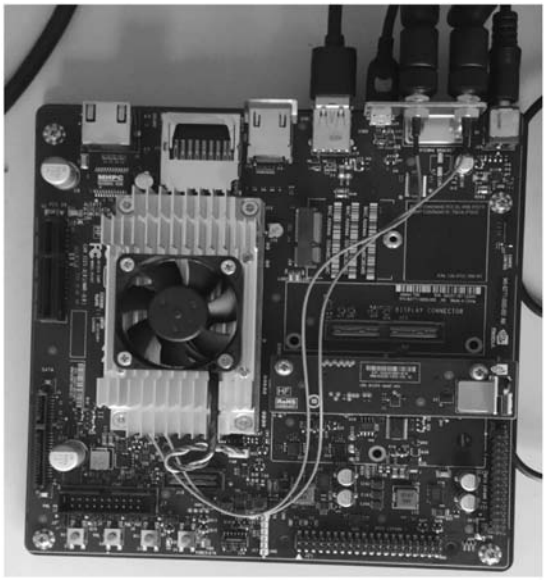


图 4 硬件平台——Jetson TX2

图 5 软件平台之建立索引

5 结 语

本文回顾比较了现有的比较典型的几种 SSE 方案,并分析了这些方案不适合智能电网应用的原因。在此基础上,提出一种基于哈希函数的智能电网数据密文检索框架、索引结构和相应的算法,并基于电力数据建立了密文检索实验平台验证了本文方案的有效性。

实验结果表明,本文方案支持密文多关键字检索,且具有更高的安全性、查询效率、更新方便等优点。在未来的工作中,我们将在本文方案的基础上进一步扩展,考虑转发信息数据可能会被泄露的问题,进一步提高本文方案的安全性。

参 考 文 献

[1] 温蜜,李婧,殷脂. 智能电网中数据的可搜索加密机制[J]. 上海电力学院学报,2013,29(6):513-517,526.

[2] Song X, Wagner D, Perrig A. Practical techniques for searches on encrypted data[C]//2000 IEEE Symposium on Security and Privacy,2000:44-55.

[3] Goh E J. Secure indexes[J]. IACR Cryptology ePrint Archive, 2003,216:1-19.

[4] 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.

[5] Wang C. Secure ranked keyword search over encrypted cloud data[C]//2010 International Conference on Distributed Computing Systems,2010:253-262.

[6] 李晓蓉,宋子夜,任婧怡,等. 云计算中基于属性的可搜索加密电子病历系统[J]. 计算机科学,2017,44(S2):342-347.

(下转第 342 页)

[13] Wang H, Ruan J, Wang G, et al. Deep learning-based interval state estimation of AC smart grids against sparse cyber attacks[J]. IEEE Transactions on Industrial Informatics, 2018, 14(11):4766 - 4778.

[14] Ozay M, Esnaola I, Vural F T Y, et al. Machine learning methods for attack detection in the smart grid[J]. IEEE Transactions on Neural Networks and Learning Systems, 2015,27(8):1773 - 1786.

[15] Esmalifalak M, Liu L, Nguyen N, et al. Detecting stealthy false data injection using machine learning in smart grid[J]. IEEE Systems Journal,2014,11(3):1644 - 1652.

[16] Niu X, Sun J. Dynamic detection of false data injection attack in smart grid using deep learning[EB]. arXiv:1808.01094, 2018.

[17] James J Q, Hou Y, Li V O K. Online false data injection attack detection with wavelet transform and deep neural networks[J]. IEEE Transactions on Industrial Informatics, 2018,14(7):3271 - 3280.

[18] Wang Y, Chen D, Zhang C, et al. Wide and recurrent neural networks for detection of false data injection in smart grids [C]//International Conference on Wireless Algorithms, Systems, and Applications,2019:335 - 345.

[19] Vaswani A, Shazeer N, Parmar N, et al. Attention is all you need[C]//Advances in Neural Information Processing Systems, 2017:5998 - 6008.

[20] Yu Z, Chin W L. Blind false data injection attack using PCA approximation method in smart grid[J]. IEEE Transactions on Smart Grid,2015,6(3):1219 - 1226.

[21] Bahdanau D, Cho K, Bengio Y. Neural machine translation by jointly learning to align and translate[EB]. arXiv:1409.0473,2014.

[22] Devlin J, Chang M W, Lee K, et al. Bert: Pre-training of deep bidirectional transformers for language understanding [EB]. arXiv:1810.04805,2018.

[23] Kingma D P, Ba J. Adam: A method for stochastic optimization[EB]. arXiv:1412.6980,2014.

(上接第 314 页)

[7] 何青,张小琳,贾梦蕾,等.基于可搜索加密的云计算智慧家居系统研究[J].无线互联科技,2017(5):53 - 54,92.

[8] Li J N, Niu X Y, Sun J S. A practical searchable symmetric encryption scheme for smart grid data[C]//ICC 2019 IEEE International Conference on Communications,2019:1 - 6.

[9] Sun W, Wang B, Cao N, et al. Privacy-preserving multi-keyword text search in the cloud supporting similarity-based ranking[J]. IEEE Parallel and Distributed Technology Systems and Applications, 2013, 25(11):3025 - 3035.

[10] Yu J, Peng L, Zhu Y, et al. Toward secure multikeyword Top-k retrieval over encrypted cloud data[J]. IEEE Transactions on Dependable & Secure Computing, 2013, 10(4): 239 - 250.

[11] Cao N, Wang C, Li M, et al. Privacy-Preserving Multi-Keyword ranked search over encrypted cloud data[J]. IEEE Transactions on Parallel and Distributed Systems,2014, 25(1):222 - 233.

[12] Sun J, Ren L, Wang S, et al. Multi-keyword searchable and data verifiable attribute-based encryption scheme for cloud storage[J]. IEEE Access,2019,7:66655 - 66667.

[13] Krishnamoorthy S. Efficient mining of high utility itemsets with multiple minimum utility thresholds[J]. Engineering Applications of Artificial Intelligence, 2018, 69(1):112 - 126.

[14] Quyen H T L, Tuong L, Bay V, et al. An efficient and effective algorithm for mining top-rank-k frequent patterns[J]. Expert Systems with Applications,2015,42(1):156 - 164.

[15] Tuong L, Bay V, Sung W B. Efficient algorithms for mining top-rank-k erasable patterns using pruning strategies and the subsume concept[J]. Engineering Applications of Artificial Intelligence,2018,68(2):1 - 9.

[16] Zhang C, Wen K, Zheng Y. Maximal frequent itemset mining algorithm based on B-list[J]. Computer Application Research, 2019,36(2):351 - 354.

[17] Deng Z H, Lv S L. PrePost + : An efficient N-lists-based algorithm for mining frequent itemsets via Children—Parent Equivalence pruning[J]. Expert Systems with Applications, 2015,42(13):5424 - 5432.

[18] Dam T L, Li K, Fournierviger P, et al. An efficient algorithm for mining top-rank-k frequent patterns[J]. Applied Intelligence,2016,45(1):96 - 111.

[19] Zhao X, Zhang X, Wang P, et al. A weighted frequent itemset mining algorithm for intelligent decision in smart systems[J]. IEEE Access,2018,6:29271 - 29282.

[20] Lee G, Yun U, Ryu K H. Mining frequent weighted itemsets without storing transaction IDs and generating candidates [J]. International Journal of Uncertainty, Fuzziness and Knowledge-Based Systems,2017,25(1):111 - 144.

[21] Lan G C, Hong T P, Lee H Y, et al. Tightening upper bounds for mining weighted frequent itemsets[J]. Intelligent Data Analysis,2015,19(2):413 - 429.

[22] Nguyen H, Vo B, Nguyen M, et al. An efficient algorithm for mining frequent weighted itemsets using interval word segments[J]. Applied Intelligence,2016,45(4):1008 - 1020.

[23] Li X, Du T, Liu B. Fast algorithm for mining frequent patterns based on B-list[J]. Computer Application Research, 2017,37(8):2357 - 2361,2367.