Multi-keyword Cipher-text Retrieval Method for Smart Grid Edge Computing*

Jing Zhu1, *, Tao Wu2, a, Jintao Li1, Yanbin Liu3, Qixin Jiang4, 5

1 School of Communication and Information Engineering, Chongqing University of Posts and Telecommunications, China
2 School of Cybersecurity and Information Law, Chongqing University of Posts and Telecommunications, Chongqing China
3 School of Computer Science, Chongqing University of Posts and Telecommunications, Chongqing China
4 Guangdong Provincial Key Laboratory of Power System Network Security, Guangzhou China
5 Electric Power Institute, CSG, Guangzhou China

*Corresponding author e-mail: zjcqhn@163.com, awutao@cqupt.edu.cn

Abstract. With the continuous deepening of China's power grid intelligent upgrading and the rise of edge computing technology, power grid terminals can carry out data storage and intelligent calculation at the edge of the power grid. However, the generated data by users may cause the leakage of user privacy information. The power grid edge system can retrieve the user's electricity data on demand, so as to improve the level of privacy protection and reduce the demand of transmission and computing resources through the localized storage of data. Therefore, how to carry out privacy protection and on-demand retrieval is a key problem to realize the intelligent power grid edge. In this paper, a multi-keyword cipher-text retrieval scheme (MCR-SGEC) for smart grid edge computing is proposed, which can achieve accurate matching of index and trap door, and has low spatial complexity. At the same time, on this basis, a multi-keyword fuzzy cipher-text retrieval method for smart grid edge computing (MFSE-SGEC) is proposed, which allows users to have spelling errors, which makes the user experience better.

1. Introduction

With the continuous development of China's power grid intelligence, the power grid will produce a large number of sampling data. The collection, transmission and storage of power grid sampling data need a lot of bandwidth and storage resources. At the same time, centralized storage may also cause the leakage of user privacy information [1]. With the rise of edge computing, power grid terminals can better support local real-time intelligent business processing. The original data collected locally can perform initial analysis at the edge and only transfer the useful data to the cloud, so as to reduce the network burden, reduce the transmission cost, and ensure the privacy and security of the data, as shown in Fig. 1. The edge computing resource allocation can meet the off-line processing and analysis of small area data, so as to ensure the safe transmission and processing of power data [2]. Therefore, how to carry out security protection and on-demand retrieval of the power consumption data stored at...
the edge of the power grid has become the key problem to realize the power grid edge intelligent computing.

![Diagram of Marginal Intelligent Computing]

In order to protect user data privacy, searchable encryption (SE) technology [3] came into being. The traditional SE method encrypts and uploads the data before outsourcing to the cloud server, but this method reduces the availability of the data to a certain extent, and users need to download all encrypted data, decrypt them one by one, and then query plaintext. This method not only wastes storage and bandwidth resources, but also causes the problem of privacy information disclosure. In view of the threat to the privacy data security of smart grid users, two searchable encryption methods based on keyword and hidden vector encryption are proposed [1]. The searchable attributes are hidden at the same time of data encryption. When the query conditions of the inquirer are met, the query keywords become the keyword threshold, thus realizing the protection of user privacy information. A searchable encryption method for smart grid (SEM-SG) based on MD5 [4] analyzes the reasons why the current typical SSE method is not suitable for smart grid scenarios. This method takes the original data encrypted by pseudo-random function as index, and matches trapdoor and index accurately, which has high practicability.

In view of [4], which only supports single keyword retrieval, it caused the search efficiency is lower, and only focus on the precise search of keywords. Especially, when the search user has spelling errors, the method can’t return the relevant search results. In practical application, multi-keyword fuzzy search method has attracted more attention because of its improvement in search accuracy, fault tolerance and user experience.

In this paper, a multi keyword cipher-text retrieval (MCR-SGEC) method is proposed for edge computing of smart grid, so as to achieve efficient, secure and multi key searchable encryption function. Specifically, AES and SHA-256 algorithm are combined to realize the precise matching between multi keyword trapdoor and record set index, and the search result list is returned to the user. In addition, a fuzzy cipher-text retrieval (MFSE-SGEC) method for smart grid edge computing is proposed. By selecting keywords from the original power data, Simhash and N-gram algorithm are used to calculate each keyword, and the keyword index is obtained. The server calculates the trapdoor of keywords input by users, and finds the location of keywords according to Hamming distance, so as to realize fuzzy search. The main work of this paper is as follows:

(1) Review and analyze the ideas of current typical SSE schemes, and discusses why these schemes are not suitable for power data;

(2) A cipher-text index structure for smart grid edge computing is proposed. SHA-256 algorithm and pseudo-random function are combined to construct cipher-text index to realize efficient and secure retrieval;

(3) A multi-keyword fuzzy cipher-text retrieval method for smart grid edge computing is proposed. Simhash and n-gram algorithm are combined to construct cipher-text index to realize multi-keyword fuzzy efficient retrieval.

2. Background and definition

2.1. System Model

The searchable encryption system framework in this paper involves three entities: data owner, data retriever and cloud server (as shown Fig. 2).
Fig. 2. Searchable encryption system framework

Data Owner (DO): responsible for collecting data, building index, and outsourcing data and index encryption to cloud server. In addition, data owners also need to give reasonable authorization to users who need to query data.

Data Retriever (DR): sends trapdoors of generated keywords to the server, and can access the data only when authorized by the data owner.

Edge Server (ES): it provides a large amount of storage space and computing resources required for cipher-text retrieval. Once the edge server receives the legitimate request from the data retriever, it will match all lists containing the search keywords and return it to the data retriever.

2.2. Problem Definition

This scheme assumes that the data owner is trustworthy, but the cloud server is honest and curious [5-7]. This shows that the cloud server will execute all the commands of the operator, and will also try to learn and analyze the data and index structure on the server, such as tampering with encrypted data, colluding with malicious users and sending wrong search results, and deleting some encrypted data. It is dishonest behavior such as making more storage space and earning more profits [8].

This scheme assumes that smart grid data has the same data structure and contains the same number of attribute sets. For example, the power data should include the year, month, time, electrical appliance type, electricity consumption type, electricity meter, power and other attributes.

2.3. Symbols Definition

The definition of relevant symbols in this scheme is shown in Table 1.

| Symbol | Description |
|--------|-------------|
| $n$    | the number of attributes set by the data retriever $(n \leq N)$ |
| $m$    | The number of keywords entered by data Retriever $(m \leq N)$ |
| $N$    | The total number of keywords in a record |
| $w_{i,j}$ | Record the $j$th keyword in $R_j$, $(1 \leq i \leq n, 1 \leq j \leq 2^n)$ |
| $\{0,1\}^*$ | A collection of $n$ digits |
| $K \leftarrow \{0,1\}^*$ | Represents uniformly sample $K$ elements from the set $\{0,1\}^*$ |
| $F = \{R_1, \ldots, R_k\}$ | Represents a collection record |
| $h : \{0,1\}^* \rightarrow \{0,1\}$ | Hash function, map any length to $r$-bit hash |
| $f : \{0,1\}^k \times \{0,1\} \rightarrow \{0,1\}$ | Pseudo-random function that maps $r$ digits to another $r$ digit with a $k$-bit key |
3. preliminaries

3.1. MD5 Algorithm
MD5 converts an arbitrary length "string" into a 128 bit hash value, and the encryption process is irreversible, that is, an MD5 value cannot be converted back to the original string to prevent tampering [9]. The computing speed of MD5 is faster than that of other hash algorithms, but MD5 is prone to collision, and it has been broken down by Wang in 2005 [10], so its security is lower.

3.2. SHA-256 Algorithm
SHA-256 algorithm is an MD structure iterative hash function. The algorithm creates a 256 bit message digest from multiple packet information with 512 bit packet length. The process is irreversible [11,12]. Sha256 is much more secure than MD5. Since MD5 was cracked, sha256 has become the most popular encryption algorithm. The following is a comparison of the characteristics of common hash algorithms, as shown in Table 2 [13,14].

3.3. Simhash Algorithm
The main idea of Simhash algorithm is to map the document features to a 64 bit long fingerprint. Since similar documents can produce similar fingerprints, the similarity between documents can be measured by hamming distance between fingerprints. Simhash algorithm has five main processes: word segmentation, hash, weighting, merging and dimension reduction.

3.4. N-gram Algorithm
A probabilistic grammar based on Markov model. It infers the structural relationship of sentences by statistical data of the probability that N symbols appear simultaneously in a natural language symbol string. When \( n = 2 \), it is called binary grammar. N-gram algorithm is widely used in text mining and natural language processing tasks.

3.5. Hamming Distance
Hamming distance is expressed as the number of bits with different values in corresponding bits between any two codewords in a code set.

\[
d(x, y) = \sum x_i \oplus y_i
\]  

where \( i = 0,1,...,n - 1 \), \( x \) and \( y \) are n-bit codes and \( \oplus \) is XOR. Therefore, the smaller the distance between Han and Ming, the higher the similarity between the two groups of code words. Therefore, Hamming distance is often used to measure the similarity between strings.

4. MCR-SGEC scheme

4.1. MCR-SGEC Framework
\[\text{Keygen}(s) \rightarrow K:\] Key generation function, where \( s \) is the security parameter.
5

*Trapdoor*(\(K,w\)) \(\rightarrow T_w^m\): The trapdoor generating function is that the data retriever inputs the key \(K\) and the keyword \(w_1,\ldots,w_m\) to be searched, and outputs the trapdoor \(T_w^m\) of the keyword \(w_1,\ldots,w_m\).

*BuildIndex*(\(R_i,K\)) \(\rightarrow I_R^i\): The cipher-text index building function, the data owner inputs the key \(K\) and record \(R_i\), and outputs the index \(I_R^i\) of record \(R_i\).

*Search*(\(T_w^m, I_R^i\)) \(\rightarrow id(C_i')\): Query function. The data retriever inputs the trapdoor \(T_w^m\) of keyword \(w_1,\ldots,w_m\) and data index \(I_R^i\) file, and the server performs matching and query. If \(w_1,\ldots,w_m \in R_i\), it outputs 1, otherwise it outputs 0.

4.2. Scheme of MCR-SGEC

4.2.1 Key generation. After determining the \(n\) attributes that the data retriever needs to query, the data owner runs the Keygen function to obtain a bit key \(k\) and keep it secret.

4.2.2 Index building. In this scheme, the first 10 attributes in the record are marked as a code-word array as the index \(I_R^i\) of record \(R_i\). The data owner calls the Trapdoor function through the BuildIndex function. For the identified \(n\) attributes in the record, compute

\[
T_w^i = f_k(h(w_i))
\]

\[
X_{i,j} = \hat{f}_{w_{i,j}}(h(id(R_i)))
\]

where \(\hat{f}\) is another pseudo-random function,

\[
\hat{f} : \{0,1\}^* \times \{0,1\}^* \rightarrow \{0,1\}^*
\]

Each codeword \(X_{i,j}\) is randomly sorted to get an array \(I_R^i\) with \(n\) attributes. The data owner attaches the index \(I_R^i\) to the encrypted record \(R_i\) and uploads it to the edge server.

Trapdoor generation

The hash value \(h(w_1),\ldots,h(w_m)\) calculated by Trapdoor function after the keyword \(w_1,\ldots,w_m\) is input by the data searcher, where

\[
h : \{0,1\}^* \rightarrow \{0,1\}^*
\]

Then calculate and output trapdoor,

\[
T_w^m = f_k(h(w_1),\ldots,h(w_m))
\]

where \(f\) is a pseudo-random function,

\[
f : \{0,1\}^* \times \{0,1\}^* \rightarrow \{0,1\}^*
\]

User search

The data retriever inputs the multi-keyword \(w_1,\ldots,w_m\) to query and calculates the trapdoor,

\[
T_w^m = f_k(h(w_1),\ldots,h(w_m))
\]

Then \(T_w^m\) is sent to the edge server. After receiving \(T_w^m\), the server calculates the codeword for each cipher-text record \(R_i(1 \leq i \leq 2^d)\),

\[
X_{w_1,\ldots,w_m} = \hat{f}_{w_{i,j}}(h(id(R_i)))
\]
And check whether the index \( I_R \) contains codeword \( X_{w_1,...,w_n} \). If the index contains the codeword \( X_{w_1,...,w_n} \), the edge server will return the \( id(R) \) list to the data retriever, otherwise, the empty list will be returned to the data retriever.

5. MFSE-SGEC scheme

5.1. MFSE-SGEC Framework

- **\( GenKey(s) \to (K) \):** Key generation function, Input security parameter \( s \) and output key \( K \).
- **\( DataEnc(F,K) \to C \):** The encryption algorithm inputs the original power data \( F \), key \( K \) and outputs cipher-text \( C \).
- **\( BuildIndex(R,K) \to I_R \):** The algorithm inputs data record \( R \) and key \( K \) to generate index \( I_R \) of keywords.
- **\( GenTrapdoor(K,w) \to T_w \):** The trapdoor generating function, inputs key \( K \) and keyword \( w_1,...,w_n \) to generate trapdoor \( T_w \) of keywords.
- **\( UserSearch(T_w,I,R) \to id(C') \):** The user search algorithm, inputs trapdoor \( T_w \) of keyword \( w_1,...,w_n \), index \( I_R \) of data record, and outputs cipher-text result list of multi-keyword fuzzy search.
- **\( UserDec(C',K) \to F' \):** The decryption algorithm, inputs cipher-text \( C' \) and key \( K \), and transmits the retrieved original power data \( F' \).
- **\( CloudSeverSearch(w) \to C_w \):** The cloud server inputs the keyword set \( w \) and outputs the cipher-text \( C_w \) of the keyword set.

5.2. Scheme of MFSE-SGEC

### Key generation

\( DO \) runs the **\( GenKey \)** algorithm to obtain a \( k \)-bit key \( K \), which is sent to authorized search users through a secure channel. Where

\[
K = GenKey(s) \tag{10}
\]

### Data encryption

\( DO \) runs **\( DataEnc \)** algorithm, encrypts the original power data \( F \) by AES algorithm, and uploads the encrypted cipher-text to the edge server.

### Index building

Firstly, the key words are identified from the original power data \( F \), that is, the \( N \) attributes in the set \( n \) are keywords, which are recorded as record \( R \). A keyword \( w \) in record \( R \) is processed by \( N \)-gram algorithm to obtain multiple features of the keyword \( w \), where

\[
Gramset = I_{w \text{-gram}}(R) \tag{11}
\]

In this scheme, in order to get more accurate search results, set \( N = 2 \), For example, when \( w \) is final, four features of \( \{F_i,in,na,al\} \) are obtained after 2-gram processing. Then the standard hash algorithm (HMAC-SHA-1 or HMAC-MD5) is used to hash the feature sets of keyword.

The hash values of these elements are mapped to vector \( V \) one by one. If the value on bit \( i \) of hash value is 0, bit \( i \) of vector \( V \) is subtracted by 1; if the value of bit \( i \) of hash value is 1, the bit of vector \( V \) is added by 1. Finally, vector \( F \) is mapped to vector \( S \), if the value of bit \( i \) of hash value is less than 0, the value of bit \( i \) of vector \( S \) is 0; otherwise, the value of bit \( i \) of vector \( S \) is 1, that is, vector \( S \) is the fingerprint of this keyword \( w \).

\[
Fingerprint = X_{\text{fingerprint}}(\text{Gramset}_{w\text{-gram}}) \tag{12}
\]
The index $I_s$ can be obtained by performing the above operations for each keyword in record $R$.

Trapdoor generation

In this scheme, the first 10 attributes in record data set 1 are marked as a codeword array, and the first five attributes in data set 2 are marked as a codeword array as index $I_R$ of record $R_i$. After inputting the keyword $w_1, \ldots, w_m$, the data retriever processes it with N-gram and Simhash algorithm to get the fingerprint of the keyword $w_1, \ldots, w_m$, which is the trapdoor $T_{w_i}$ of the keyword.

User search

The user inputs the keyword $w_1, \ldots, w_m$, calculates its trapdoor $T_{w_i}$, and finally sends the trapdoor $I_s$ to the edge server. After receiving trapdoor $T_{w_i}$, edge server calculates Hamming distance $d_w$ between trapdoor $T_{w_i}$ and fingerprint in index $I_s$. If $d_w < 3$, server will return $\text{id}(R)$ list to data retriever, otherwise, it will return empty list to data retriever.

When the user inputs the keyword "final" to "finel" due to carelessness or forgetting, the value of the keyword fingerprint is actually determined by multiple features of the keyword because of the n-gram processing of the keyword. When a user makes a spelling mistake, two of the four features of the keyword are changed (na, al becomes ne, el). Although the hash values of the two features will change greatly compared with before, the fingerprint of keywords is determined by the four features together. Therefore, compared with other related keyword fingerprints, the Hamming distance between the misspelled keyword fingerprint and the correct keyword fingerprint is smaller.

According to this principle, users can find the correct keywords by comparing the Hamming distance between fingerprints even if they misspell. The distance between "final" and "finel" can be shown in the Fig. 3.

6. Security analysis

The primary privacy issues in MCR-SGEC and MFSE-SGEC scheme are confidentiality of power system data, privacy of index and trapdoor, and unlinkability of trapdoor.

6.1. Confidentiality of Power System Data

In the two schemes, the outsourcing power data is encrypted by the traditional symmetric encryption AES algorithm. It has been proved in [17] that the AES encryption algorithm is secure, and the key is sent to the data retriever through a secure channel. Without the key, no entity can recover the encrypted data, so the confidentiality of the encrypted data can be achieved.
6.2. Privacy Protection of Index and Trapdoor
In MCR-SGEC scheme, SHA-256 algorithm is used to hash the codeword, and pseudo-random function is used to write the codeword into the index array randomly, which can ensure that the attacker can’t learn the original keywords from the index. In MFSE-SGEC scheme, the one-way hash function with key ensures the privacy of fingerprint. Fingerprint is only used to match cipher-text and does not reveal privacy. Therefore, the server or attacker can’t know the specific information of keywords from the index. Therefore, both schemes can guarantee the certainty of index and the confidentiality of query.

6.3. Unlinkability of Trapdoors
The edge server can infer and identify keywords by analyzing the trapdoors generated by keywords. In view of this, the codewords of the two schemes are constructed by introducing the line number of the data as the identifier, that is, the same keywords in the original power data have different codewords in the index, thus realizing the unlinkability of trapdoors.

7. Performance evaluation
The index type and search complexity performance comparison of the literature described in this paper is shown in Table 3 [15,16]. The purpose of this scheme is to improve the efficiency and security of cipher-text retrieval, so that attackers can’t get any information about plaintext records from the index. Because the codeword is randomly inserted, the user search complexity is $O(2^d \cdot n)$, and the number of attributes set $n$ by the data searcher is considered to be a very small constant in practical application, so the search scheme is effective.

| SSE Scheme | Index  | Update | Search Complexity |
|------------|--------|--------|-------------------|
| SWP[2]     | no     | easy   | $O(2^d \cdot N)$  |
| Z-IDX[3]   | direct | easy   | $O(2^d)$          |
| SSE-1[4]   | inverted | hard  | $O(1)$            |
| EMRS[5]    | direct | hard   | $O(2^d)$          |
| MCR-SGEC   | direct | easy   | $O(2^d \cdot n)$  |
| MFSE-SGEC  | direct | easy   | $O(2^d \cdot n)$  |

7.1. Experimental Data
The test environment used in this paper is based on Linux platform, the specific hardware configuration is Intel Xeon (R) CPU e5-1650v4 processor, with 62.8gb memory and 1Gbps campus network environment. The experimental data set 1 is shown in Fig. 4. It uses 47 months' power consumption data collected by a family in sceaux (7km of Paris, France). The data set contains nine pure digital properties such as time, date, voltage, global active power, global reactive power and global strength [18]. Data set 2 uses aim data provided by the energy information administration (EIA), which includes more than 20 attributes such as the monthly electricity consumption of the US electric power company in 2016, and its data set type is a mixed type of string and number [19].

Fig. 4. Visualization of household energy consumption data
7.2. Result Analysis

In order to verify that the MCR-SGEC scheme has higher search efficiency, the MCR-SGEC scheme is compared with SEM-SG scheme in literature [4] on different power data sets, mainly from the index construction time, trapdoor generation time, user search time. The implementation results are shown in Fig. 5 and Fig. 6. It can be seen from Fig. 5 that the index building time of SEM-SG scheme in Fig. 5(a) is slightly longer than that of MCR-SGEC scheme. The generation time of keyword trapdoor in SEM-SG scheme in Fig. 5(b) is linear, while the method of MCR-SGEC scheme is almost parallel. The search time of SEM-SG scheme in Fig. 5(c) is significantly longer than that of MCR-SGEC scheme. Therefore, the MCR-SGEC scheme has greatly improved the index establishment, trapdoor generation and search efficiency. Similarly, it can be seen from Fig. 6 that the trapdoor generation time and user search time of MCR-SGEC scheme are significantly reduced compared with those in SEM-SG scheme. To sum up, MCR-SGEC scheme realizes multi-keyword cipher-text retrieval on different power data sets.

In order to verify that MFCR-SGEC scheme has better user experience effect, comparative experiments are conducted on different power data sets between MFCR-SGEC scheme and SEM-SG scheme, mainly from index construction time, trapdoor generation time and user search time. The implementation results are shown in Fig. 7 and Fig. 8. It can be seen from Fig. 7 that the MFCR-SGEC scheme in Fig. 7(a) takes longer to build the index than the SEM-SG scheme. The reason is that the MFCR-SGEC scheme calculates each keyword in the attribute by n-gram and Simhah algorithm, and its calculation amount exceeds the index construction time of SEM-SG scheme. Compared with the keyword trapdoor generation time of SEM-SG scheme in Fig. 7(b) with that of MFCR-SGEC scheme, the trapdoor generation time of SEM-SG scheme is linear with the number of keywords input by users, while the trapdoor generation time of MFCR-SGEC scheme is much less than that of SEM-SG scheme. The search time of SEM-SG scheme in Fig. 7(c) is basically the same as that of MFCR-SGEC scheme. The reason is that SEM-SG scheme needs to calculate trapdoor and match with index to check whether trapdoor is included in index, while MFCR-SGEC scheme needs to calculate Hamming distance between trapdoor and array index. When Hamming distance is less than 3, cipher-text will be output. Compared with SEM-SG scheme, MFCR-SGEC scheme needs to calculates Hamming distance. Each value in the group, so the time required depends on the keywords entered by the user. The more keywords are input, the more times the Hamming distance

![Comparison of index building time](image1)
![Comparison of trapdoor generation time](image2)
![Comparison of user search time](image3)

Fig. 5. Index construction, trapdoor generation and query efficiency comparison of MCR-SGEC on dataset 1
Fig. 6. Index construction, trapdoor generation and query efficiency comparison of MCR-SGEC on dataset 2

Fig. 7. Index construction, trapdoor generation and query efficiency comparison of MFSE-SGEC on dataset 1

Fig. 8. Index construction, trapdoor generation and query efficiency comparison of MFSE-SGEC on dataset 2
needs to be calculated, the longer the time is required. As shown in Fig. 8, the MFCR-SGEC scheme achieves the same effect as Fig. 7. To sum up, MFCR-SGEC scheme realizes multi-keyword fuzzy search on different power data sets.

8. Conclusion
This paper reviews and compares several existing SSE schemes for smart grid, analyzes their advantages and disadvantages, and proposes a multi-keyword cipher-text retrieval framework for smart grid edge computing, a direct index structure based on hash algorithm and pseudo-random function, and relative user search algorithm. At the same time, a multi-keyword fuzzy cipher-text retrieval framework for smart grid edge computing, index structure based on N-gram and Simhash algorithm and relative user search algorithm are proposed, and multi-keyword fuzzy search is realized according to Hamming distance. The above two algorithms are based on power data to establish a cipher-text retrieval experimental platform, which verifies the effectiveness of the scheme. The experimental results show that the MCR-SGEC scheme supports multi-keyword cipher-text retrieval, and has the advantages of higher security, query efficiency and convenient update. MFSE-SGEC scheme supports multi-keyword fuzzy cipher-text retrieval, improves user comfort and is easy to update. In the future work, we will further expand the scheme to consider the problem that the forwarding information data may be leaked, and further improve the security of the scheme.

Acknowledgement
Foundation item: National Key R&D Program of China (2018YFB0904900, 2018YFB0904905); National Natural Science Foundation of China (61802039, 61772098).

References
[1] M. Wen, J. Li, Z. Yan, “Searchable encryption mechanism of data in smart grid,” Journal of Shanghai Electric Power University. vol. 29, pp. 513-517, June 2013.
[2] C. Zhang, X. Y. Fan, X. T. Liu, et al, “Risk management of smart grid data assets,” Big data. vol. 005, pp. 64-78, February 2019.
[3] H. L. Dai, G. Yang, Z. E Min, “Multi-keyword parallel ciphertext retrieval scheme in distributed environment,” Journal of Computer Applications. vol. 39, pp. 2948-2954, October 2019.
[4] J. N. Li, X. Y. Niu, J. Y. Sun, et al, “A practical searchable symmetric encryption scheme for smart grid data,” ICC 2019 - 2019 IEEE International Conference on Communications (ICC). Shanghai, China, pp. 1-6, 2019.
[5] W. H. Sun, B. Wang, N. Cao, “Privacy-preserving Multi-keyword Text Search in the Cloud Supporting Similarity-based Ranking, IEEE Parallel and Distributed Technology Systems and Applications, vol. 25, pp. 3025-3035, November 2013.
[6] J. D. Yu, P. Lu, Y. M. Zhu, “Toward Secure Multikeyward Top-k Retrieval over Encrypted Cloud Data,” IEEE Transactions on Dependable and Secure Computing. vol. 10, pp. 239-250, April 2013.
[7] N. Cao, C. Wang, M. Li, “Privacy-Preserving Multi-Keyword Ranked Search over Encrypted Cloud Data,” IEEE Transactions on Parallel and Distributed Systems. 2014, pp. 222 – 233.
[8] J. Sun, L. L. Ren, S. P. Wang, et al, “Multi-keyword searchable and data verifiable attribute-based encryption scheme for cloud storage,”IEEE Access. vol. 2169, pp. 66655-66667, 2019.
[9] W. B. Peng, “Principle and application of MD5 algorithm,” Information network security. pp. 44-46, May 2004.
[10] C. Yang, Z. Ding, “Good news from crypto’2004: Chinese scholars cracking MD5,” Information network security. pp. 12-13, October 2006.
[11] Y. L. Yang, D. N. Liu, P. Zhao, “Linux kernel data encryption based on AES and SHA-256,” Journal of Sichuan Institute of Technology (self SCIENCE EDITION). vol. 25, pp. 44-48, June 2012.
[12] X. H. Yang, Z. B. Dai, “Implementation of SHA-256 algorithm based on FPGA,” Microcomputer information. vol. 22, pp. 146-148, November 2006.

[13] R. M. He, J. Ma, “security analysis of SHA-256 algorithm,” Electronic design engineering. vol. 22, pp. 31-33, March 2014.

[14] Z. L. Liu, X. Dong, D. F. Li, “Hardware implementation of SHA-2 (256384512) series algorithm,” Microelectronics and computer. vol. 29, pp. 51-54, December 2012.

[15] J. W. Li, C. F. Jia, Z. L. Liu, “A survey of searchable encryption technology,” Acta software Sinica. vol. 26, pp. 109-128, January 2015.

[16] J. L. Zhang, Y. C. Zhao, B. Chen, “A survey of data security and privacy protection in edge computing,” Journal of communications. vol. 39, pp. 1-19, March 2018.

[17] F. Niels, S. Richard, W. Doug, “A simple algebraic representation of Rijndael,” Lecture Notes in Computer Science. vol. 2259, pp. 103-111, 2001.

[18] https://archive.ics.uci.edu/ml/datasets/individual+household+electric+power+consumption.

[19] https://www.eia.gov/electricity/data/eia861m/index.html#ammete