Intelligent and Converged Networks ISSN 2708-6240
2023, 4(2): 127−141 DOI: 10.23919/ICN.2023.0014

Operating system network security enhancement scheme based on trusted storage

Longyun Qi, Xiaoliang Lv, Lianwen Sun, Tianle Yao, Jianye Yu, and Lei Wang*

Abstract: Data storage security has become the core of many network security issues. In order to achieve trusted storage and trusted measurement of network community data, this paper proposes a secure storage model based on trust extension for existing trusted storage technologies. In the process of document encryption, the key information is encrypted as well as decentralized stored by optimizing the ciphertext inverted index structure and update policy to ensure the security of index information. In the process of user access control mechanism, SAML and XACML are used in combination with role-based access control in order to achieve flexible and efficient authorization and access control. In the process of result query, ontology technology is introduced to better express the user's query intention and improve the query accuracy. A large number of experiments demonstrate the effectiveness and feasibility of the scheme.

Key words: trusted storage; trust extension; trusted platform model (TPM); access control

1 Introduction

With the rapid development of information technology, network security and user privacy leakage are becoming increasingly serious problems, and the security of data has become a major issue facing society[1−3]. The traditional network protection mechanism can no longer effectively guarantee the security of network data, and trusted computing has become one of the most effective means to achieve network data security. Facing the threat of network data vulnerability, trusted computing can fundamentally change the traditional defense model and basically realize active immunity and trusted management of data in the whole process of storage, transmission and processing. Trusted computing technology relies on the trusted platform model (TPM), a tamper-proof hardware chip containing a cryptographic operator for key generation, encryption and decryption, and a storage component for key storage. The core of data security is cryptography, which is controlled by key management mechanism.

Trusted storage refers to the secure storage realized by using external storage devices such as hard disk on the basis of trusted platform control module, which ensures the physical security of hardware[4]. Trusted storage system can be formed by binding the trusted platform control module with physically secure external storage. Trusted computing sealing refers to the combination of data and platform configuration and other information for sealing, followed by storage. TPM can bind the data to the platform configuration register (PCR) value specified within the security chip, when unsealing the data, it can correctly unseal the data by judging whether the PCR value is the same as the original value. Trusted storage technology is also constantly being improved and optimized with the development of the Internet[5].

With the rapid development of emerging technologies such as 5G wireless technology, data mining, artificial intelligence, fog computing, cloud

• Longyun Qi, Xiaoliang Lv, Lianwen Sun, and Jianye Yu are with the State Grid Electric Power Research Institute, Nanjing 210003, China. E-mail: qilongyun@sgepri.sgcc.com.cn; lvxiaoliang@sgepri.sgcc.com.cn; sunlianwen@sgepri.sgcc.com.cn; yujianye@sgepri.sgcc.com.cn.
• Tianle Yao and Lei Wang are with the State Grid Beijing Electric Power Company, Beijing 100031, China. E-mail: 562430814@qq.com; wl_sgcc@163.com.
* To whom correspondence should be addressed.
Manuscript received: 2023-04-23; revised: 2023-05-23; accepted: 2023-06-14

computing, software-defined networking (SDN), and network function virtualization (NFV), trusted storage faces many new challenges[6, 7]. To address the shortcomings of existing trusted storage technologies, this paper proposes a secure storage model based on trust extension, the whole scheme includes five modules: document pre-processing, inverted index construction, trust extension, retrieval result sorting, and access control. The feasibility and effectiveness of the proposed scheme are verified through experiments.

2 Related work

To effectively ensure that data are not accessed by unauthorized users and the information is in a secure state, in 2008, He and Xu[8] introduced the concept of trusted storage (TS) and advanced the implementation of TS by analyzing several access scenarios in practice. Subsequently, various optimization and solutions have been proposed. Kühn et al.[9] proposed an attribute-based sealing scheme, which no longer relies on the metric value in PCR for sealing as in the traditional scheme, but binds the attribute information of the selected PCR value for operation. The authors in Ref. [10] proposed the idea of storing a small amount of trusted data while ensuring the storage of a large amount of untrusted data. Yan et al.[11] pointed out the gradual concentration of current research efforts on the development and extension of trusted computing (TC) modules, trusted software assurance, and trusted execution environments (TEEs). Trusted storage systems based on TEE are mainly integrated in existing TEEs, and only trusted storage systems are introduced in Open-TEE, which lacks specific design methods and performance analysis. While in Linaro's OP-TEE version 2.1[12], there are two ways of trusted storage, where storage objects are stored in rich execution environment (REE) file system and replay protected memory block (RPMB), and data are stored in blocks for read, write, and store. Besides, to facilitate in-store computing, Kang et al.[13] built a lightweight trusted execution environment, called IceClave, for in-store computing that achieves secure isolation between in-store programs and flash management functions. Performance is greatly improved compared to traditional host-based trusted computing approaches. Han et al.[14] proposes a trusted certificateless authenticated public-key encryption with keyword search scheme (TCA-PEKS) using keyword search in cloud storage, and constructs an open and transparent smart contract to limit the malicious behavior of ECS and ensure trusted retrieval. Wang et al.[15] proposed a blockchain based trusted data storage architecture for national infrastructure, which solves the problem of data interoperability through the Internet of Things (IoT) and federated learning technology.

The development of trusted storage is also inseparable from a reasonable trusted measurement mechanism. Xin et al.[16] proposed to perform trustworthiness measurement by establishing a vector-based uncertainty mathematical model with dynamic adaptability, but it lacks a corresponding risk assessment mechanism and is not applicable to big data scenarios. In view of cloud users' concerns about the trustworthiness of cloud computing services, Ma et al.[17] proposed to use information entropy and Markov chain as a trustworthiness evaluation method to build a trusted cloud service attribute model, taking into account the uncertainty and correlation between trustworthiness factors in the measurement process. Wang and Liu[18] proposed TMMDP, a trusted measurement model based on dynamic policy and privacy protection of infrastructure as a service (IaaS) security domain. The existing trusted authentication mechanisms lack real-time measurement and tracking of sensing nodes. To solve the above problem, Gong et al.[19] proposed a dynamic metric-based authentication mechanism for IoT sensing nodes. Xu et al.[20] proposed a reputation-aware supplier assessment system (SAS) using technologies such as Canopy, K-medoids, and backpropagation neural networks. Some scholars also turned their attention to online collaborative data caching in edge computing, and achieved rich results[21, 22]. To achieve collaborative edge storage based on blockchain, Yuan et al.[23] proposed CSEdge, a new decentralized system where the performance of the server is recorded on the blockchain for future reputation evaluation.

In summary, the implementation of trusted storage

should be based on the TPM trusted storage module, but the efficiency of binding and sealing the platform information for storage still needs to be improved, and the trustworthiness and efficiency of verification need to be improved when performing data storage verification.

3 Secure storage model based on trust extension

Secure storage model based on trust extension includes four roles: data owner, TPM, trusted storage server, and users. The model is shown in Fig. 1.

The data owner is mainly responsible for pre-processing documents, including extracting keywords for each document, counting the frequency of keywords and other information, then generating document metadata, encrypting document metadata to generate cryptographic metadata, and finally uploading cryptographic metadata to TPM for index building. The ciphertext metadata and TPM related identification information are encapsulated and stored. At the same time, the data owner is also responsible for encrypting the original documents to generate a collection of ciphertext documents and uploading them directly to the trusted server for storage.

TPM is responsible for receiving the cryptographic metadata uploaded by the data owner, then constructing a secure inverted index based on this cryptographic metadata, and then uploading the index to the database for subsequent retrieval work. After receiving the query keywords submitted by users, the keywords are trustfully extended based on the ontology knowledge base to generate a new set of query keywords, encrypt them and submit query requests to the trusted storage server.

The trusted storage server is responsible for storing ciphertext document collections and ciphertext inverted index files. After receiving the query request submitted by TPM, it extracts the data blocks and metadata that need to be verified, and retrieve the ciphertext document. In the retrieval, the relevance of keywords and documents is calculated. The retrieved documents are sorted according to their relevance and the top N documents are finally returned according to user requirements.

A user is a person or an organization that is authorized to access data in the server. They submit query keywords to TPM and receive retrieved documents back from the trusted storage server, download them locally for decryption, and use them for themselves.

Specifically, the model of secure storage based on trust extension consists of four modules: document pre-processing module, inverted index construction module, trust extension module, and retrieval result sorting module.

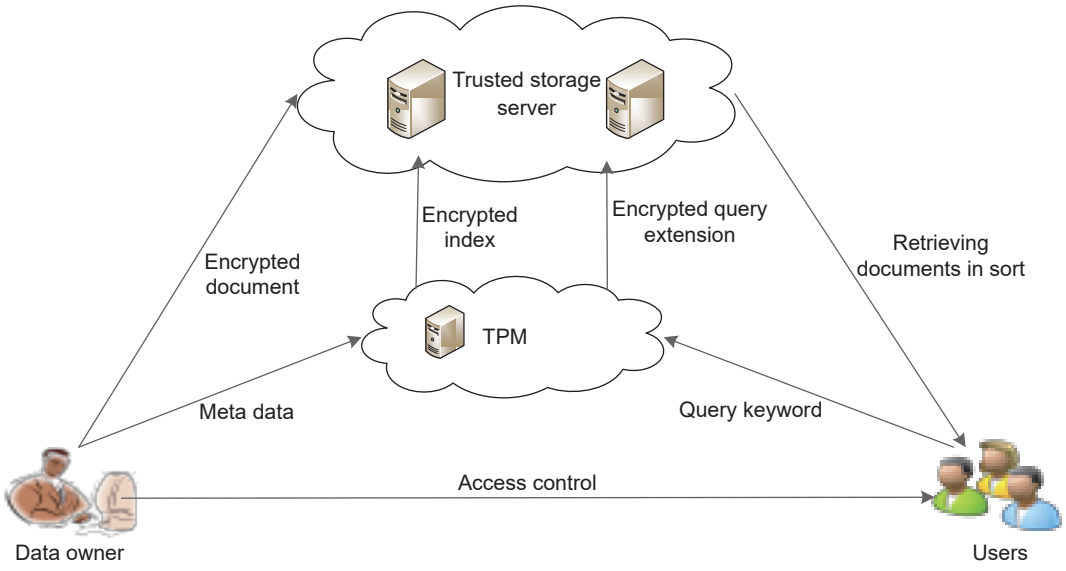


Fig. 1 Model of secure storage based on trust extension.

Among them, the document pre-processing module is responsible for generating document metadata based on the original document, including the process of word separation, word frequency statistics, and data encryption. The inverted index construction module is responsible for establishing a secure query index for the document and generating a ciphertext inverted index based on the document metadata. The trust extension module is responsible for extending the trust semantic level based on the query keywords submitted by users using ontology knowledge, which mainly includes two parts: trust extension and extension result filtering. The retrieval result sorting module is responsible for searching according to the submitted query keywords and sorting according to the relevance to return the top *N* documents that are most relevant to the query. The whole retrieval process is shown in Fig. 2.

To facilitate the description of the scheme below, the symbols to be used in the scheme are given first, as shown in Table 1.

3.1 SAML- and XACML-based access control for role-based access control (RBAC) models

Security assertion markup language (SAML), is a traditional encoding specification used to encode user authentication data in XML, based on which cross-platform accessibility is achieved. XACML is a common access control policy language and access decision language for protecting resources proposed by the International Organization for Standardization OASIS Security Services Association. The policy language is used to define generic access control requirements and ultimately generate a deny or permit decision. The access decision language is used to set whether to allow access to certain server resources[24].

XACML has extensibility and can support parameterized policy descriptions, using XACML, it is good for access control of services. SAML can be used to exchange authentication and authorization credentials in different security domains of the Internet, with the ability to provide single sign-on authentication, using SAML can achieve single sign-on. However, all of these are only specifications, for the specific authorization authentication does not give the actual solution. At this point, authorization access control must be achieved through a specific policy scheme[25, 26]. Role management is introduced to achieve a relatively simple authorization method, using roles to act as intermediaries for exercising permissions. When the system assigns a specific role to a user, the user logs in using the role, using SAML for single sign-on, avoiding users with the same role from making permission judgments every time they log into the service site, simplifying the complexity of authorization management, and achieving effective access control.

In this paper, we use SAML and XACML combined

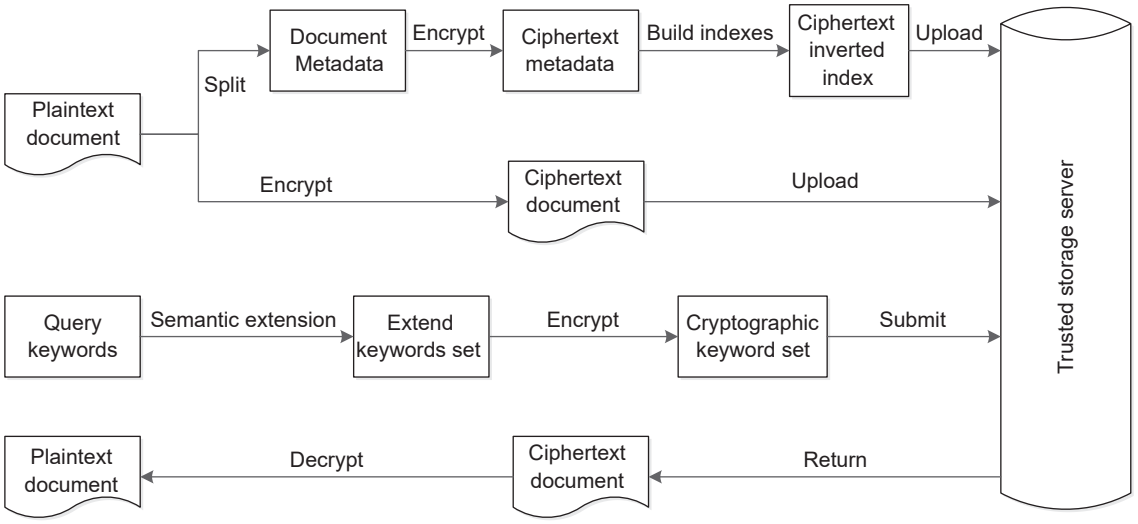


Fig. 2 Ciphertext retrieval process based on trust extension.

Table 1 List of symbols.

Symbol	Description of meaning
D	The set of plaintext documents, which can be expressed as $D = \{D_1, D_2, ..., D_n\}$
K	The set of keywords, which can be expressed as $K = \{K_1, K_2, ..., K_m\}$
W	The set of keyword weights in the document $W = \{W_1, W_2, ..., W_m\}$
E	Symmetric encryption algorithm
T_K	Keyword K's query door trap
S_K	Keyword K's trusted extended keyword collection

with role-based access control to construct an RBAC model based on SAML and XACML to achieve flexible and efficient authorization and access control. Its access control process is shown in Fig. 3.

• **Process 1**. The client selects the service and sends an access request to the PEP in SAML language. If the client has been authenticated and assigned a role, the SAML cofactor will be carried in the transmitted message.

• **Process 2**. PEP parses the received SAML message, checks whether it contains SAML adjuncts, and if it does not, it redirects to the authentication center/role assignment service and asks the user to authenticate first. Otherwise, PEP requests the authentication assertion and role assertion from the authentication center based on the auxiliary parts, the authentication center returns the SAML response result, PEP verifies the legality of the assertion, and if the verification is passed, it enters the fourth step, otherwise it returns an error prompt to the client.

• **Process 3**. The authentication center/role assignment service uses the user name and password submitted by the user for authentication, and after successful authentication, assigns the user to the corresponding role according to the role server, then generates authentication assertions and role assertions and generates the corresponding adjuncts, returns the adjuncts to the client for saving by the client, and redirects the user request to the PEP, which contains the adjuncts in the redirection URL. The PEP receives the redirect response and gets the corresponding assertion from the authentication center according to the auxiliary.

• **Process 4**. PEP generates an extended SAML authorization decision request based on the request information sent by the client and the role assertion obtained from the authentication center, which is then sent to the PDP.

• **Processes 5 and 6**. After the PDP receives the request, it queries the PIP for the relevant attribute values, including the subject, the accessed resource, the action, the environment, etc.

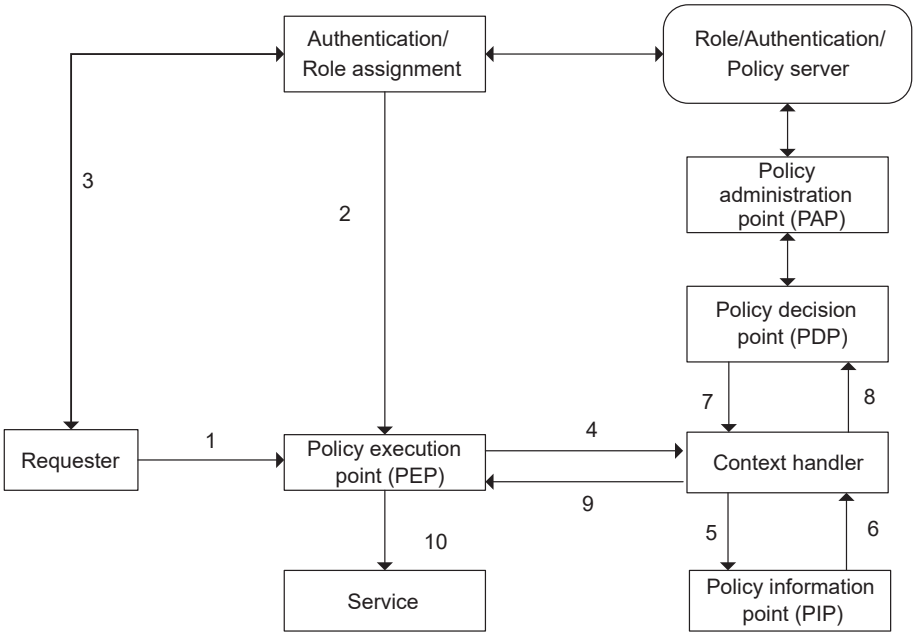


Fig. 3 Access control process of RBAC model based on SAML and XACML.

• **Processes 7 and 8**. The PDP generates an extended SAML policy request with the user request, role, and attribute values obtained in the previous two steps and sends it to the PAP. The PAP retrieves the access control policy matching the target from the policy server and returns it to the PDP based on the constraints of the target contained in the request, which is an XACML-based RBAC policy including role policy set, access permission policy set, etc.

• **Process 9**. The PDP evaluates the returned access control policy, determines whether the current role has the appropriate access rights, and then makes an authorization decision and sends it to the PEP.

• **Process 10**. The PEP decides whether to allow users to access the corresponding Web services based on authorization decisions.

3.2 Document pre-processing

3.2.1 Word segmentation

Word segmentation is the process of dividing a sentence into a number of words, and there are mainly Chinese and English word segmentation depending on the language. For Chinese word segmentation, the same Chinese word in different order has different meanings, and the same Chinese word in different contexts also has different lexical properties and meanings, which causes the complexity of Chinese language, so how to make accurate word segmentation for Chinese is a difficult but meaningful topic. English word segmentation is relatively simple, because English words are usually separated by spaces, and computers can easily distinguish different words.

3.2.2 Stop words filtering

Stop words are words that have no plausible semantic impact on the original statement in the process of word segmentation, including punctuation, conjunctions, auxiliaries, onomatopoeia, tone words, etc. For example, the words “a”, “so”, “on”, “the”, “this”, etc., in English. These words have no real meaning to the credible semantics of the sentence, and cannot be keywords in the process of building query indexes, so they are filtered out in the document pre-processing stage. After splitting, we get a single word, and we first need to filter out the stop words that have no specific meaning in the splitting result using the stop words table.

3.2.3 Word frequency counting

Word frequency counting is then performed, and words that appear less frequently in the document need to be eliminated in the process. If a word appears very few times in a document, it means that the word is not very relevant to the document, so we can filter out these words that have little credible semantic response to the document and further narrow down the set of keywords.

3.2.4 Generate document metadata

After segmenting the original document, stopping word filtering, word frequency counting, and low-frequency word filtering, the remaining words in the document are used to construct document metadata, which includes the keywords extracted from the document and the corresponding word frequency information, location information, and so on. They are organized as a unit of document, and the structure diagram is shown in Fig. 4.

In Fig. 4, *D* denotes the document number, *K* denotes the keyword, *TF* denotes the word frequency information of the keyword, and *Positions* denotes the position information of the keyword in the document.

Define abbreviations and acronyms the first time they are used in the text, even after they have been defined in the abstract.

3.2.5 Generate ciphertext metadata

In order to ensure the security of document metadata, it

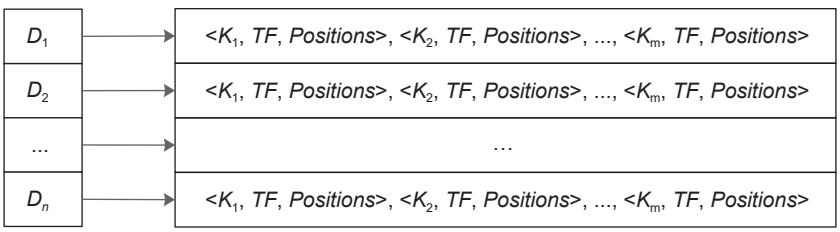


Fig. 4 Schematic diagram of document metadata structure.

is necessary to encrypt it. Considering the relatively large scale of data, the keywords in the document metadata are encrypted by symmetric encryption algorithm with faster encryption and decryption speed.

Similarly, to ensure the security of the document set in the trusted storage server, the original document set is encrypted using symmetric encryption algorithm to generate a ciphertext document set, and then uploaded to the trusted storage server. The trusted storage server is responsible for storing the ciphertext document set and retrieving the document set according to the user's query threshold and returning the relevant documents that satisfy the retrieval criteria to the user.

3.3 Inverted index construction

The inverted index building module is responsible for building a secure inverted index for a document after receiving the ciphertext metadata submitted by the user[27]. In this paper, we propose an improved secure inverted index in the field of ciphertext retrieval.

An inverted index is an index structure describing the correspondence between a keyword collection and a document collection, which maps a retrieved keyword to an inverted table and indicates the documents containing this keyword as well as the word frequency information and location information of the keyword. The index is constructed as follows: the ciphertext metadata is scanned and the keywords are extracted sequentially to form the set of ciphertext keywords $E(K) = (E(K_1), E(K_2), \ldots, E(K_m))$, and then for each keyword $E(K_i)$, the document D_i containing the keyword and the corresponding word frequency information and location information are extracted and inserted as a tuple into the inverted documents.

The encryption of keywords in the index file can prevent the attacker's trusted semantic analysis attack to a certain extent, but the logical record pointer in the index file still has security risks, and the attacker can obtain the correspondence between ciphertext keywords and ciphertext documents through the logical record pointer by tracing the retrieval process, so as to carry out the attack. For this reason, it is necessary to shield the connection between ciphertext keywords and ciphertext documents, to encrypt the logical record pointers as well, and to facilitate the construction of the index, the same encryption algorithm is used for keyword encryption. The improved index structure is shown in Fig. 5.

In Fig. 5, $E(P_i)$ denotes the pointer to the encrypted logical record, and the encryption algorithm is the same as the keyword encryption algorithm.

Then, from the perspective of inverted file security, each record in an inverted file is a collection of ciphertext documents corresponding to a keyword, and these documents all contain the keyword, so if an attacker learns the information of one of the documents, they can use this connection between the documents to infer the information of other documents. In the traditional inverted file, the document information is stored continuously, which brings convenience in retrieval, but is also vulnerable to the cascading attack described above. Based on the above analysis, a chained-table storage structure can be used instead of continuous storage, and the improved ciphertext index structure is shown in Fig. 6.

In the improved inverted index structure, the ciphertext documents are stored in a chained table, and the chain table pointer refers to the chain table in the index file, which also uses symmetric encryption, so that even if an attacker obtains a ciphertext document, they cannot further decrypt more related documents.

For the keyword frequency, location, and other

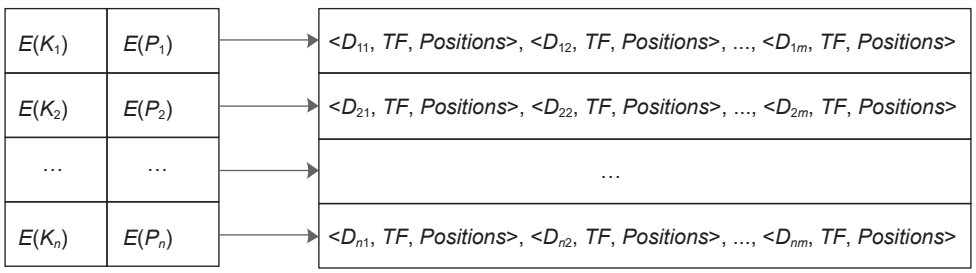


Fig. 5 Schematic diagram of the inverted index structure with encrypted keywords and encrypted pointers.

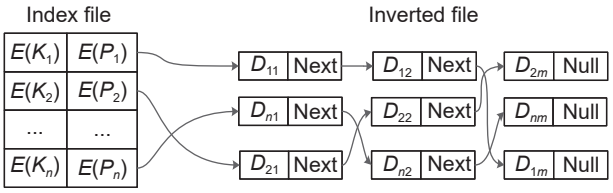


Fig. 6 Schematic diagram of the inverted index structure for linked-table storage.

information in the document, they are used to calculate the relevance of the keywords and the document when retrieving, and are the key data for sorting the document when returning the retrieval results. Generally speaking, the ciphertext data cannot perform the relevant arithmetic operations, so the information that needs to be calculated, such as the keyword frequency and location, is not encrypted in this paper. The security of the whole ciphertext retrieval is ensured by the encryption algorithm and the improved inverted index.

The update of the inverted index in ciphertext retrieval involves two main aspects: one is the addition of indexes due to the addition of documents, and the other is the deletion of indexes due to the deletion of documents.

For index addition, suppose there is a document D to be added to the set of cipher-text documents, firstly, the document will be processed by word segmentation, and for a certain keyword K_i, the symmetric encryption algorithm will be used to encrypt to get $E(K_i)$, then the document D will be encrypted by the same algorithm to get the ciphertext document $E(D)$, if the keyword $E(K_i)$ exists in the index file, the ciphertext document information will be added to the corresponding index file; otherwise, a new ciphertext keyword is added to the index file, and then the ciphertext document information is added to the corresponding index file. In the specific implementation, we can create new indexes for the added documents first, and then merge the indexes when the newly added indexes reach a certain size, so as to avoid the extra overhead caused by frequent merging of indexes.

For the index deletion, suppose there is a document D to be deleted from the set of ciphertext documents, first obtain all the keywords of the document through the word segmentation, and for a certain ciphertext keyword $E(K_i)$, search in the index file to get the document chain table in the inverted file corresponding to the document, and then delete the node belonging to the information of document D in the chain table, if the document chain table is empty after the deletion, then also delete keyword $E(K_i)$ in the index file. This shows that the deletion operation needs to reconstruct the index, and reconstruct the entire index brings a large overhead. Here we use the idea of chunking, as shown in Fig. 7, where the inverted file is stored in blocks. Assuming that the file is divided into M blocks, then the overhead of the operation to delete a keyword will be reduced to the original $1/M$.

3.4 Trust extension

The traditional exact keyword matching technique focuses on the search completion rate, often ignoring some credible semantically related documents, making it difficult to guarantee the accuracy of this technique.

To address the above problems, this paper proposes an ontology-based trust extension method. Through the constructed ontology knowledge base, trustworthy semantic reasoning is performed on the query keywords input by users to obtain the extended keyword set, and at the same time, in order to avoid noise to the original query due to the extended keyword set being too large, the trustworthy semantic similarity calculation is performed on the extended keywords, and the threshold value of the similarity value is set so that only words with similarity value greater than the threshold value are selected to join the final extended keyword set. Then the new extended keyword set is

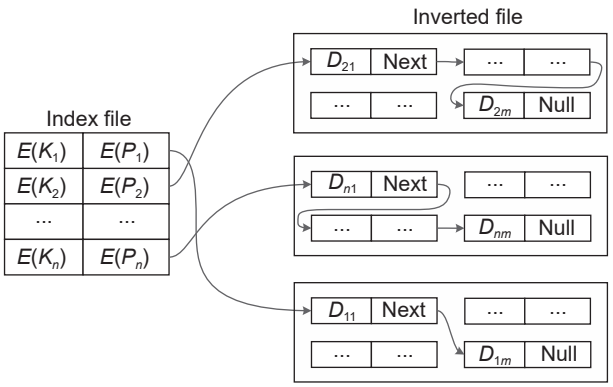


Fig. 7 Schematic diagram of the inverted index structure for chunked storage.

used for retrieval to fully express the user's query intent, thus improving the search accuracy.

The ciphertext retrieval module based on trust extension is shown in Fig. 8, which consists of two parts: the trust extension module and the retrieval result sorting module. The trust extension module receives query keywords from users and then uses ontology techniques to extend the keywords on trust semantics, and finally outputs a new set of query keywords centered on the original query keywords and submits them to the retrieval module. The retrieval result ranking module performs retrieval based on the submitted query keywords using the constructed inverted index, and ranks the documents according to the relevance of the query keywords and documents, and returns the top N documents with higher ranking to the user.

3.4.1 Trust extension algorithm

Ontologies are formal specification descriptions of shared conceptual models. Using ontologies, we can obtain specific descriptions of the domain in question, obtain relevant knowledge of the domain, obtain commonly accepted words in the domain, and be able to give interrelationships between these words. With the help of the generic ontology WordNet knowledge base, we can perform query inference on the user's query keywords and obtain words that are plausibly semantically related to the query keywords, together as a new set of query keywords to better understand the user's query intent.

The ontology-based trust extension scheme proposed in this paper is as follows: assuming that this scheme is a single-keyword search, when a user submits a query keyword k, a trusted semantic tree with the original query as the root node is constructed using the complete trusted semantic structure hierarchy of the WordNet knowledge base, and words related to the trusted semantics of the original query keyword are expanded as the child nodes of the trusted semantic tree. To control the scale of the extended keywords, the credible semantic similarity between each node on the trusted semantic tree and the root node is calculated, and the words with similarity greater than a set threshold are selected as the extended set and used as the new set of query keywords together with the original query keywords. The algorithm for query keyword expansion is described as follows.

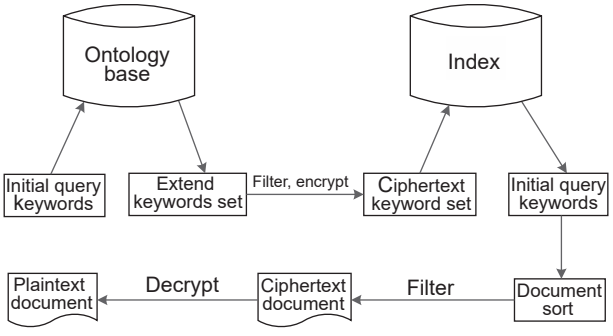


Fig. 8 Schematic diagram of the ciphertext retrieval module based on trust extension.

(1) Initialize the trusted semantic tree with the original query keyword k as the root node;

(2) For the n sense terms of keyword k, the following operations are performed in sequence;

• Take the synonym on the semantic item as the child node of the root node of the trusted semantic tree.

• If there is a superlative on that sense, add it to the subtree with that sense as the root node.

• If there is an inferior term on that sense, add it to the subtree with that sense as the root node.

(3) For each node m on the trusted semantic tree, calculate the similarity between m and the original query keyword $Sim(k, m)$, and for a set threshold σ, add m to the extended keyword set $Knew$ if there is $Sim(k, m) > \sigma$.

(4) Compute $k' = k \cup Knew$ as the new set of query keywords.

(5) Encryption algorithm is used to encrypt the extended keyword set and submit it to the trusted storage server.

3.4.2 Extended result filtering

Trusted semantic similarity is used to describe the degree of similarity between two words, and it is a value between [0, 1].

In the above trust extension algorithm, we use the ontology conceptual model to credibly semantically extend the query keywords submitted by users to form a preliminary set of extended keywords. However, the relevance of these extended keywords is not exactly the same as the original query keywords, their trustworthy

semantic similarity with the original query keywords varies, and if all of them are added to the extended keyword set, noise will be generated to cause bias in the query results. With the help of plausible semantic similarity, these extended keywords are ranked from high to low and a threshold is set so that only words with plausible semantic similarity greater than this threshold can be added to the final set of extended keywords, and the accuracy of the extended results is ensured by filtering the extended keywords. The setting of the threshold value, which is an empirical value derived from the experimental analysis, will be discussed in the experimental section.

The calculation of plausible semantic similarity between two words using ontologies is based on the prerequisite that the two words are related in terms of plausible semantics and that they have at least one connected path in the structural hierarchical network graph of the ontology. The following factors are usually considered when computing trusted semantic similarity between words using ontology structures.

(1) Trusted concept overlap degree. It refers to the number of two words that contain the same superordinate word in the ontology concept, i.e., the number of common nodes in the hierarchical network of ontology structure. The plausible semantic overlap represents the degree of identity between two words, and the more plausible semantic overlap, the greater the similarity of the words.

(2) Trusted semantic distance. It refers to the length of the shortest pathway among the pathways connecting these two nodes in the ontology structure graph. The credible semantic distance is a basic factor to measure the credible semantic similarity, and the credible semantic similarity and the credible semantic distance are negatively correlated: the smaller the credible semantic distance between two words, the greater the similarity between them; conversely, the smaller the similarity between them. The two extreme cases are: if the distance of two words is 0, then their similarity is 1; if the distance of two words is infinity, then their similarity is 0.

(3) Hierarchical depth. For two words at the same distance, the similarity of words is related to the depth they are located. In general, the greater the difference between the depth values of two words, the greater the similarity; the smaller the difference between the depth values, the smaller the similarity.

(4) Moderation factors. In different systems, different moderation factors are set in order to balance the weight of depth and breadth in which the words are placed. Similarity is an empirical value that needs to be set in a specific context.

In this paper, we adopt the method of using common parent node to calculate the trusted semantic similarity, which takes into account not only the depth of the smallest parent node of two words, but also the length of their connection paths to each other. For two words with the same parent node, their trusted semantic similarity decreases as the distance of the connected path increases; for two words with the same connected path length, their trusted semantic similarity increases as the depth of the smallest parent node increases. The specific formula is as follows:

$$sim(c_1, c_2) = \frac{2\times depth(lso(c_1, c_2))}{len(c_1, lso(c_1, c_2))+len(c_2, lso(c_1, c_2))+2\times depth(lso(c_1, c_2))},$$

where $lso\ (\cdot,\cdot)$ denotes the smallest upper-level concept of the two words, *depth* denotes the depth the words are in, and $len\ (\cdot,\cdot)$ denotes the path length of the two words.

3.5 Sorting search results

After receiving a query request, the trusted storage server searches the entire index file based on the received extended keywords, calculates the comprehensive relevance of the query keyword set to each document, sorts the documents according to the size of the comprehensive relevance value, and returns the top N documents of most interest to the user.

Vector space model (VSM) is a spatial vector-based retrieval model, which represents both query information and documents to be retrieved as spatial vectors, measures the relevance of query information and documents by the similarity between vectors, and returns results in the order of relevance, and has become a widely used model for information retrieval. In this paper, this model is used to model the set of query keywords and the documents to be retrieved, transforming the processing of keywords and

documents into vector operations in vector space, and measuring the credible semantic relevance based on the similarity of vectors. The relationship between the relevance of keywords and documents and the angle between the vectors is as follows: the smaller the angle between two vectors, the greater the relevance of the keywords to the documents; conversely, the larger the angle, the smaller the relevance.

If a document contains n keywords, then the document can be regarded as an n-dimensional vector $D = \{K_1, K_2, \ldots, K_n\}$, and similarly the weights of all the keywords in the document are regarded as a vector $DV = \{W_{1,d}, W_{2,d}, \ldots, W_{n,d}\}$, and $W_{i,d} = 0$ indicates that the keyword K_i is not in the document. The rest are the weights of the keywords K_i in the document. Similarly, the query statement is treated as a simple document and is also represented as a vector $Q = \{K'_1, K'_2, \ldots, K'_n\}$, and the weights corresponding to the keywords are expressed as $QV = \{W_{1,q}, W_{2,q}, \ldots, W_{n,q}\}$, and the values at the corresponding positions in the vector are the plausible semantic relevance of the corresponding extended terms to the original query keywords. For document d, its correlation scoring score with the set of extended keywords q is calculated as follows:

$$score(q,d) = \frac{V_q V_d}{|V_q||V_d|} = \frac{\sum_{i=1}^{n} W_{i,q} W_{i,d}}{\sqrt{\sum_{i=1}^{n} W_{i,q}^2} \sqrt{\sum_{i=1}^{n} W_{i,d}^2}},$$

where V_d is the spatial vector of document d, V_q is the spatial vector of the extended keyword set q, and $W_{i,d}$ and $W_{i,q}$ are the weights of keyword K_i in document d and extended keyword set q, respectively.

The algorithm for calculating keyword weight values is currently widely used term frequency-inverse document frequency (TF-IDF) algorithm. TF-IDF is a statistically based method for evaluating the importance of words to a particular document. The importance of a word to a particular document is proportional to its frequency of occurrence in the document and inversely proportional to its frequency of occurrence in other documents in the document set. *TF* represents the frequency of a keyword in a document, while *IDF* represents the frequency of a keyword in other documents, also called inverse document frequency. Its calculation formula is as follows:

$$W_i = TF \times IDF = TF \times \log_2\left(\frac{N}{DF_i} + 0.01\right),$$

where W_i is the weight of keyword K_i, TF is the frequency of keyword K_i in document D, IDF is the inverse document frequency of the keyword, N is the total number of documents, and DF_i is the number of documents containing keyword K_i.

4 Experiments and analysis

4.1 Experimental environment

The experimental environment and development techniques used for system testing are shown in Table 2. In the experiment, the full-text search engine tool Lucene is used to realize the search function, and the version is 2.4.0. Lucene is not a fully functional full-text search engine, but a full-text search engine architecture that provides a complete query engine and search engine. Developers can make secondary development and customization based on this architecture according to different requirements. The index construction and query retrieval in this experiment are all implemented based on this architecture.

WordNet, the latest Windows-based version 2.1, is used as the general ontology library in the experiment. In the experiment, WordNet is used to extend the query keywords submitted by users credibly, and a new set of query keywords is obtained.

The test document set of the experiment is Request For Comments (RFC), which is a collection of documents about Internet communication protocols,

Table 2 Experimental environment and development techniques.

Function	Tool
Hardware environment	Intel Core i5-450M CPU, 4 GB RAM
Operating system	Windows 7
Java environment	JDK1.8, IntelliJ IDEA 14.0.3
Application server	Apache Tomcat 7.0.63
Search engine	Lucene 2.4.0
Ontology library	WordNet 2.1
Test document set	Request For Comments (RFC)

and almost all Internet standards are included here. At present, the RFC document set contains more than 7700 documents, among which 7000 are selected as the test document set. Different test document subsets are set according to different experimental requirements, and their format is TXT text format.

4.2 Evaluation metrics

Recall rate (*Recall*). It refers to the rate of the number of returned relevant documents to the total number of relevant documents in the document set to be retrieved in a retrieval.

Precision rate (*Precision*). It is the rate of the number of related documents returned to the total number of documents actually returned by the retrieval.

Top N precision ($P@N$). It represents the ratio of the number of related documents in the top N results to N, and is commonly used to measure the precision of the top N result documents that users pay attention to.

$$Recall = \frac{\text{return relative doceumntes}}{\text{total relative documents}} \times 100\%,$$

$$Precision = \frac{\text{return relative doceumntes}}{\text{the actual document returned}} \times 100\%,$$

$$P@N = \frac{\text{return relative doceumntes in top } N}{N} \times 100\%.$$

4.3 Experimental results and analysis

Ciphertext retrieval includes document preprocessing, index building, and keyword retrieval. This experiment will focus on index building performance and index building performance. Through the selected document set, the performance of ciphertext retrieval and plaintext retrieval is compared, and the performance of the proposed ciphertext retrieval based on trusted extension and single keyword ciphertext retrieval is compared. The experiments will be carried out and analyzed from several aspects, such as expansion rate of ciphertext index, construction time of ciphertext index, expansion scale of keywords, response time of ciphertext retrieval, and precision of ciphertext retrieval.

4.3.1 Ciphertext index expansion rate

The plaintext index and ciphertext index of document set are constructed respectively, and compare the change in index size. According to the comparison in Fig. 9, it can be found that both plaintext index and ciphertext index expand with the expansion of document scale, and the scale of index increases in a positive proportion. For a test document set of the same size, the size of the ciphertext index is larger than that of the plaintext index. Because the storage space of the ciphertext is larger than that of the plaintext after the keywords in the document set are encrypted, and the ciphertext size varies with different encryption algorithms, and the expansion rate varies.

4.3.2 Ciphertext index construction time

The plaintext index and ciphertext index of document set are constructed respectively, and compare the change in index build time. According to the comparison in Fig. 10, it can be found that the construction time of ciphertext index and plaintext index increases with the expansion of the scale of document set, which increases in a positive proportion. The construction time of ciphertext index and plaintext index is not much different. Because for plaintext keywords and ciphertext keywords, the index construction process is almost the same. The time difference is mainly to encrypt the keywords, the

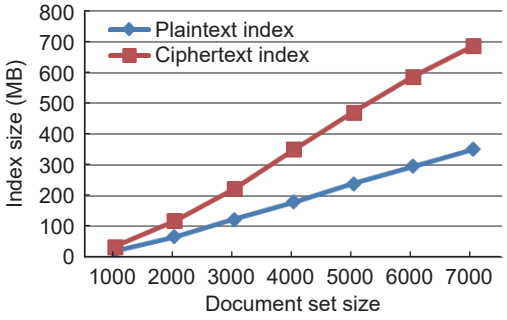


Fig. 9 Scale comparison of ciphertext index and plaintext index construction.

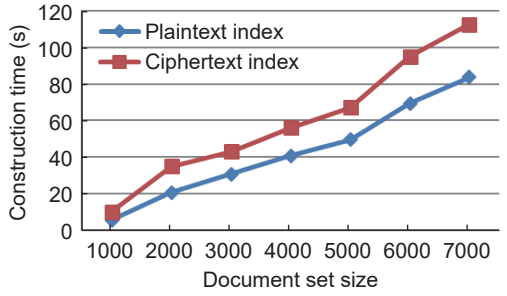


Fig. 10 Comparison of construction time between ciphertext index and plaintext index.

length of the encryption time depends on the speed of the encryption algorithm itself.

4.3.3 Influence of expanded scale on precision

The number of keywords in the extended keyword set is determined by setting different thresholds σ to find the optimal scale range. As can be seen from Fig. 11, when the number of extended keywords is between 16 and 24, the precision is relatively high. Because if the number of keywords is too small, the query intention cannot be fully expressed. If the number of keywords is too large, noise will be generated, which deviates the trusted semantic of the query from the original intention. Therefore, we can control the number of keyword extensions within this range by setting the value of σ to achieve a high precision.

4.3.4 Precision of ciphertext retrieval

Single keyword retrieval and trusted extension retrieval are performed respectively on the test document set. Calculate the time to return the first 10, 15, 20, 25, and 30 documents. According to Fig. 12, it can be found that the precision of the trusted extension scheme is significantly improved compared with that of the single keyword, which provides users with more accurate retrieval results. Through the above experimental analysis, it can be concluded that the scheme of improving precision by trusted extension is feasible and effective in index performance and retrieval performance.

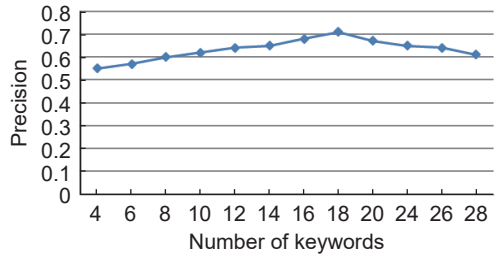


Fig. 11 Comparison of precision between ciphertext index and plaintext index.

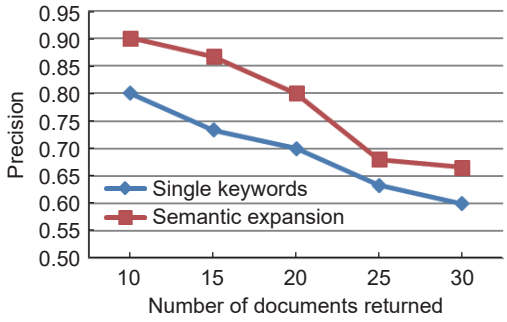


Fig. 12 Comparison of precision between single keyword and trusted extension.

5 Conclusion

In this paper, we propose a secure storage model based on trust extension, which includes four roles: data owner, TPM, trusted storage server, and user. And the whole scheme includes five modules: document pre-processing, inverted index construction, trust extension, retrieval result sorting, and access control. In the process of document encryption, the key information is encrypted as well as decentralized storage by optimizing the ciphertext inverted index structure and update policy to ensure the security of index information. In the process of user access control mechanism, SAML and XACML are used in combination with role-based access control in order to achieve flexible and efficient authorization and access control. In the process of result query, ontology technology is introduced to better express the user's query intention and improve the query accuracy. Experiments show the effectiveness and feasibility of the scheme and model.

There are still many issues that need further research regarding the research content of this article, mainly including:

(1) The ontology-based trusted extension technology proposed in this article is based on a premise that the keywords to be extended are words and concepts within the ontology. However, in practical retrieval, the situation is complex and variable, and the query keywords submitted by users may not necessarily belong to existing concepts within the ontology. Therefore, how to map non-native words to the concept of the ontology has become a worthwhile research topic.

(2) The ontology used in the article is a universal ontology library. The advantage of a universal ontology is that it can describe concepts within a universal scope and their relationships. The disadvantage is that its description of concepts is not precise enough. In order to provide more accurate

knowledge of certain specific fields, it is necessary to build a domain ontology, but building a domain ontology also requires the participation of domain experts. To utilize ontology technology for trusted extension applications, the research and construction of domain ontology is an indispensable link.

Acknowledgment

This research was supported by the Science and Technology Project of State Grid Corporation of China (No. 5700-202258188A-1-1-ZN).

References

[1] X. Zhou, X. Yang, J. Ma, and K. I. K. Wang, Energy-efficient smart routing based on link correlation mining for wireless edge computing in IoT, *IEEE Internet Things J.*, vol. 9, no. 16, pp. 14988–14997, 2022.

[2] L. Qi, W. Lin, X. Zhang, W. Dou, X. Xu, and J. Chen, A correlation graph based approach for personalized and compatible web APIs recommendation in mobile APP development, *IEEE Trans. Knowl. Data Eng.*, vol. 35, no. 6, pp. 5444–5457, 2023.

[3] X. Zhou, W. Liang, K. Yan, W. Li, K. I. K. Wang, J. Ma, and Q. Jin, Edge-enabled two-stage scheduling based on deep reinforcement learning for Internet of everything, *IEEE Internet Things J.*, vol. 10, no. 4, pp. 3295–3304, 2022.

[4] W. Liang, Y. Hu, X. Zhou, Y. Pan, and K. I. K. Wang, Variational few-shot learning for microservice-oriented intrusion detection in distributed industrial IoT, *IEEE Trans. Ind. Inform.*, vol. 18, no. 8, pp. 5087–5095, 2021.

[5] Q. He, S. Tan, F. Chen, X. Xu, L. Qi, X. Hei, A. Zomaya, H. Jin, and Y. Yang, EDIndex: Enabling fast data queries in edge storage systems, in *Proc. 46th Int. ACM SIGIR Conf. Research and Development in Information Retrieval*, Taipei, China, 2023.

[6] X. Zhou, X. Xu, W. Liang, Z. Zeng, and Z. Yan, Deep-learning-enhanced multitarget detection for end–edge–cloud surveillance in smart IoT, *IEEE Internet Things J.*, vol. 8, no. 16, pp. 12588–12596, 2021.

[7] S. Wu, S. Shen, X. Xu, Y. Chen, X. Zhou, D. Liu, X. Xue, and L. Qi, Popularity-aware and diverse web APIs recommendation based on correlation graph, *IEEE Trans. Comput. Soc. Syst.*, vol. 10, no. 2, pp. 771–782, 2023.

[8] J. He and M. Xu, Research on storage security based on trusted computing platform, in *Proc. 2008 Int. Symp. on Electronic Commerce and Security*, Guangzhou, China, 2008, pp. 448–452.

[9] U. Kühn, K. Kursawe, S. Lucks, A. R. Sadeghi, and C. Stüble, Secure data management in trusted computing, in *Proc. 7th Int. Workshop Cryptographic Hardware and Embedded Systems* (*CHES 2005*), Edinburgh, UK, 2005, pp. 324–338.

[10] U. Maheshwari, R. Vingralek, and W. O. Sibert, Trusted storage systems and methods, https://patents.google.com/patent/US8904188B2, 2013.

[11] Z. Yan, V. Govindaraju, Q. Zheng, and Y. Wang, IEEE access special section editorial: Trusted computing, *IEEE Access*, vol. 8, pp. 25722–25726, 2020.

[12] J. Forissier, LAS16-504: Secure storage updates in OP-TEE, https://www.slideshare.net/linaroorg/las16504-secure-storage-updates-in-optee, 2022.

[13] L. Kang, Y. Xue, W. Jia, X. Wang, J. Kim, C. Youn, M. J. Kang, H. J. Lim, B. Jacob, and J. Huang, IceClave: A trusted execution environment for in-storage computing, in *Proc. 54th Annu. IEEE/ACM Int. Symp. Microarchitecture* (*MICRO-54*), virtual, 2021, pp. 199–211.

[14] M. Han, P. Xu, L. Xu, and C. Xu, TCA-PEKS: Trusted certificateless authentication public-key encryption with keyword search scheme in cloud storage, *Peer Peer Netw. Appl.*, vol. 16, no. 1, pp. 156–169, 2023.

[15] Y. Wang, R. Fan, X. Liang, P. Li, and X. Hei, Trusted data storage architecture for national infrastructure, *Sensors*, vol. 22, no. 6, p. 2318, 2022.

[16] S. Y. Xin, Y. Zhao, J.-H. Liao, and T. Wang, Dynamic trusted measurement model of operating system kernel, *J. Comput. Appl.*, vol. 32, no. 4, pp. 953–956, 2012.

[17] Z. Ma, R. Jiang, M. Yang, T. Li, and Q. Zhang, Research on the measurement and evaluation of trusted cloud service, *Soft Comput.*, vol. 22, no. 4, pp. 1247–1262, 2018.

[18] L. Wang and F. Liu, A trusted measurement model based on dynamic policy and privacy protection in IaaS security domain, *EURASIP J. Inf. Secur.*, vol. 2018, no. 1, pp. 1–8, 2018.

[19] B. Gong, Y. Wang, X. Liu, F. Qi, and Z. Sun, A trusted attestation mechanism for the sensing nodes of Internet of Things based on dynamic trusted measurement, *China Commun.*, vol. 15, no. 2, pp. 100–121, 2018.

[20] X. Xu, J. Gu, H. Yan, W. Liu, L. Qi, and X. Zhou, Reputation-aware supplier assessment for blockchain-enabled supply chain in industry 4.0, *IEEE Trans. Ind. Inform.*, vol. 19, no. 4, pp. 5485–5494, 2023.

[21] X. Xia, F. Chen, Q. He, J. Grundy, M. Abdelrazek, and H. Jin, Online collaborative data caching in edge computing, *IEEE Trans. Parallel Distrib. Syst.*, vol. 32, no. 2, pp. 281–294, 2021.

[22] Q. He, Z. Dong, F. Chen, S. Deng, W. Liang, and Y. Yang, Pyramid: Enabling hierarchical neural networks with edge computing, in *Proc. ACM Web Conf. 2022*, virtual, 2022, pp. 1860–1870.

[23] L. Yuan, Q. He, F. Chen, J. Zhang, L. Qi, X. Xu, Y. Xiang, and Y. Yang, CSEdge: Enabling collaborative edge storage for multi-access edge computing based on

blockchain, *IEEE Trans. Parallel Distrib. Syst.*, vol. 33, no. 8, pp. 1873–1887, 2022.

[24] L. Qi, Y. Yang, X. Zhou, W. Rafique, and J. Ma, Fast anomaly identification based on multiaspect data streams for intelligent intrusion detection toward secure industry 4.0, *IEEE Trans. Ind. Inform.*, vol. 18, no. 9, pp. 6503–6511, 2022.

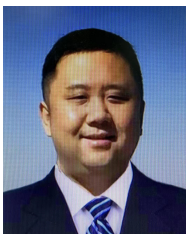
[25] Z. Li, X. Xu, T. Hang, H. Xiang, Y. Cui, L. Qi, and X. Zhou, A knowledge-driven anomaly detection framework for social production system, *IEEE Trans. Comput. Soc. Syst.*, doi: 10.1109/TCSS.2022.3217790.

[26] H. Dai, J. Yu, M. Li, W. Wang, A. X. Liu, J. Ma, L. Qi, and G. Chen, Bloom filter with noisy coding framework for multi-set membership testing, *IEEE Trans. Knowl. Data Eng.*, vol. 35, no. 7, pp. 6710–6724, 2023.

[27] X. Zhou, W. Liang, K. I. K. Wang, and L. T. Yang, Deep correlation mining based on hierarchical hybrid networks for heterogeneous big data recommendations, *IEEE Trans. Comput. Soc. Syst.*, vol. 8, no. 1, pp. 171–178, 2021.

Longyun Qi received the BS degree from Nanjing University of Aeronautics and Astronautics, China, in 2002. He is currently with NARI Group Corporation, State Grid Electric Power Research Institute, where he is mainly responsible for the underlying security research of the operating system to support the development of the entire department. His main research interests include power system information security and trusted computing.

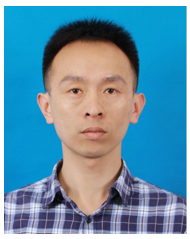
Jianye Yu received the MS degree from Nanjing University of Science and Technology, China, in 2002. He is currently with NARI Group Corporation, State Grid Electric Power Research Institute, where he is mainly responsible for research on power system information security. His main research interest is power system information security.



Lei Wang received the PhD degree from Tsinghua University, China, in 2007. He is the director of Network Security Division, Digitalization Department, State Grid Beijing Electric Power Company. Since 2011, he has been engaged in power information and communication work. His current research interests include network security management in the power industry, in paticular, network security and information system automation operation and maintenance

Tianle Yao received the BS degree in network engineering from Fuzhou University, China, in 2015. He worked in the network security position of State Grid Beijing Electric Power Company, responsible for the overall network security protection of State Grid Beijing Electric Power Company. His main research direction is information security.

Xiaoliang Lv received the BS degree from Ocean University of China, China, in 2007. His research interests include operating system security and trusted computing. Now, he works in NARI Group Corporation, State Grid Electric Power Research Institute, China.



Lianwen Sun received the MS degree from Nanjing Forestry University, China, in 2015. He is a senior software engineer at NARI Group Corporation, State Grid Electric Power Research Institute. His main research interests include operating system security and trusted computing.