An Encryption System for Sensitive Data

Yang Li[1,a,*], Yanlian Zhang[2,b]

[1]DIGITAL CHINA(CHINA)LIMITED, Beijing, China

[2] China Flight test establishment , Xi'an, China

[a,*] digital9898@sina.com

Keywords: Sensitive Data, Data Collection, Encryption Cluster, Key Management

Abstract. A sensitive data encryption system is proposed. The sensitive data will be obtained through the data acquisition platform, the digital certificate will be issued through the CA system, and the sensitive data can be encrypted effectively through the key management system and the encryption machine cluster, which can be provided to the third party application System. The system can effectively improve the security of the sensitive data.

1. Introduction

With the development of network information, the network affects all aspects of people's lives, followed by increasing security requirements, the network activities on the confidentiality requirements are also increasing, especially for sensitive data, to ensure important sensitive data security, to ensure data on the network security and confidentiality.

2. System Architecture

Sensitive data encryption system consists of key management system, sensitive data processing engine, encryption machine cluster. The key management system is responsible for data key generation, compliance review, storage, distribution, recovery, and destruction. Sensitive data processing engine is responsible for interface file temporary storage, sensitive data filtering comparison, encryption and decryption, job scheduling, policy management, system monitoring, access control. Other work encryption machine cluster system provides encryption machine based services, including encryption and decryption. The data collecting platform is a sensitive data source, and the third application system is a data output side. In addition, the CA system issues a digital certificate for the relevant system, providing basic protection of identity for data access control.

The system is divided into three levels: data storage, data processing and application interface. The bottom layer is the data storage layer, which provides the database support for the system, including the key store and the sensitive database. It provides the key database interface and the sensitive database interface respectively for the data processing layer. The data processing layer is the core part of the system, and the data query process. The key and sensitive data are obtained by calling the interface of the data storage layer, and the key used in the data storage is stored in the database through the sensitive database interface. The application layer is the external business layer, and the system is open to the data access interface. Third-party applications access system data.

Copyright © 2017, the Authors. Published by Atlantis Press.
This is an open access article under the CC BY-NC license (http://creativecommons.org/licenses/by-nc/4.0/).

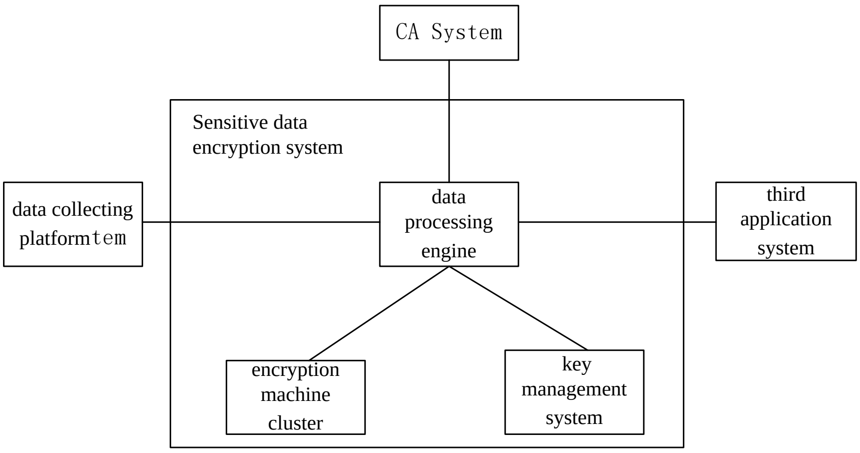
CA System
Sensitive data encryption system
data collecting platformtem
data processing engine
third application system
encryption machine cluster
key management system

Figure 1 System architecture

2.1. System Function

Sensitive data management system consists of data processing engine, encryption machine cluster, and key management system. The part of the data processing engine includes functions such as rights management, configuration management, system monitoring, encryption strategy, log management, data checking, filtering comparison, data caching and data querying etc. and is responsible for managing and monitoring the whole system and password equipment. The encryption part of the cluster includes encryption and decryption, key update, dynamic expansion of the encryption function, responsible for the system call encryption machine interface response. Key management system provides symmetric key, asymmetric key generation, key checking, and key management. It provides key for encryption and decryption in data processing, and manages the key storehouse at the same time. Data processing engine is the core of the sensitive data management system. It is a logical operation platform for handling files and sensitive data and supporting client service interfaces. It is responsible for calling cryptographic equipment and database clusters to complete the actual cryptographic operations and the distribution and storage of operational results. Data management system to deal with the core, the data processing center to provide monitoring and alarming functions, by monitoring the alarm interface, system management can be part of the real-time access to service running.

2.2. Data Process Engine

2.2.1. Management of User

Management of user include add, view, delete and modify functions. Configuration management can add a new server configuration, delete the existing server configuration. To achieve encryption machine IP address, connection password configuration, database data source, user name, password configuration.

2.2.2. System Monitor

Check the current operation of each data server, the server CPU memory and the current total CPU memory usage and other information. Check the data files that the server is currently processing. View the current service status of the encryption device, including the IP address, support algorithm, etc. For more information about the encryption machine, can find an alarm when the encryption device is abnormal.

2.2.3. Encryption Policy

Check a list of algorithms supported by the encryption machine, including details of each algorithm. The encryption algorithm is built into the configuration file and can not be modified by the user. Configure the encryption algorithm for each level of encryption. The encryption algorithm only accepts encryption algorithms supported by the encryption engine. The encryption profile can be configured, modified, and deleted. The goal of the encryption template is that different systems use different encryption levels to selectively encrypt sensitive fields.

2.2.4. Data Check

In order to ensure the received data is consistent with the expected data, excluding the network transmission errors caused by the correctness of the data impact, the server and client APIs need to check the validity of the data, including packet inspection and integrity checks. The packet inspection is responsible for determining the received packet conforms to the protocol. When the system receives the request packet to the interface call or the client API receives the packet of the system response information, it performs security check to filter the packets that do not meet the requirements of the protocol. The integrity check is responsible for ensuring that the received data is not lost in the network transmission. Every interaction between the system and the application should be HASH operation, the HASH value attached to the request or response information, to ensure the integrity of communication data.

2.2.5. Screening Comparison

Data file authentication: data authentication of sensitive data server is divided into verification data file legitimacy, verifying data file integrity, file decompression and other steps. Data processing: sensitive data server in the decompression of the correct data file, need to verify the data file and the actual business date, the contents of the current file hash operation, the hash results with a number of days before the file hash value comparison.

After the file is validated, the sensitive fields in the data file are extracted and the sensitive fields are encrypted and stored. The main steps are as follows: parse the data file, analyse the file according to the encryption template, encrypt the sensitive data Submitted to the pending queue by rule combination. The duplicated data is then removed by data cache alignment. According to the rules set for the sensitive data HASH, get the data unique identification information. Through the HASH value and file parsing the data in the cache to remove duplicate records in the file, extracted to valid data. Finally, the cache does not exist in the records, the encryption is stored in the database after the success of the cipher text, together with the value of the data stored in the HASH value with the file to resolve the cache.

2.2.6. Data Cache

When system start up, it load all the necessary data from the database, that is, the current province of the corresponding server data, according to the rules of data pre-processing into the cache. Cache provides the validity of data control, when the system service is shut down or abnormal, the cache data isolation or physical elimination, in order to protect the security of data. When parsing a file, all valid data after de-weighting is inserted into both the cache and the database.

2.2.7. Data Query

Select the query criteria and enter the query ID, and the query results is shown on the page. Display request time, system information table, query field, data type, data date, request status, response time and SFTP server IP and query result file path. Detailed records of the user's bulk query are returned.

2.2.8. Encryption and Decryption

According to the actual usage of the key in the encryption template, setting the key type and algorithm to be loaded in the cache; and setting the size of the cache according to the replacement cycle in the key replacement policy to make it conform to the replacement frequency of the key. The key identifier that needs to be preloaded with a part of the backup key is stored in the system, cache at the time of system startup, and the corresponding key is stored in the cache of the encryption machine at the same time. When the key replacement condition in the key policy is reached, the encryption key identifier in the cache is changed, and the interface of the encryption machine is called at the same time, and the key corresponding to the identifier is deleted from the encryption machine cache.

When encrypting, need to use the key cache to obtain the current provincial data encryption key identifier, the sensitive data identification and key identification sent to the password device at the same time, the data encryption, and return to the cipher text.

When decrypting, the key identifier of the data is obtained according to the data identification. The sensitive data encryption system sends the key identifier to the key management system, and the key management system sends the key to the encryption machine. The sensitive data encryption system then sends cipher text and data identification to the encryption machine. After the encryption machine decrypts, returns the plaintext

2.2.9. Key Update

Before using the system, need to set the key usage period and the maximum number of key usage times in the system to ensure that the key can be updated without being broken. The key management system receives the key update request sent by the data engine and sends a request to the encryption machine, and the new key is updated by the encryption machine cluster.

2.2.10. Dynamic Expansion

The encryption machine is responsible for encrypting / decrypting the upload data in the system. As the request data volume is large, a single encryption machine cannot meet the encryption and decryption performance requirements. Therefore, the system needs to use the encryption machine cluster to provide encryption and decryption services. Taking into account the future growth of the system data volume, encryption machine cluster need to support the dynamic expansion of the encryption machine, to ensure that the amount of data in the existing cluster encryption and decryption performance bottlenecks can be increased when the encryption machine is heavy.

Symmetric key includes key generation, key use, key distribution several processes, the system present the minimum number of key threshold and key life cycle, after the key is sent to the encryption machine. When the system key is less than the minimum number of key configured, the system performs an automatic task generation key. The system support symmetric key of SM1/4, 3DES, AES and other commonly used symmetric algorithms. The algorithm can be configured to set the appropriate algorithm.

2.2.11. Asymmetric Key

Asymmetric key has standby, in use, write-off, expired four states. The newly generated key state is standby; when the key is applied by the system certificate or the system user association, the state becomes active; when the application system certificate is cancelled or the system user logs off, the asymmetric key associated with it is cancelled; When the application system certificate expires or the system user expires, the associated asymmetric key expires at the same time.

When the system key is less than the minimum number configured in the key policy, the system executes the automatic task generation key, and the system needs a small amount of asymmetric key to configure a moderate minimum number of keys. The key archiving is performed in the same way as the automatic task in the archiving cycle configured in the key policy.

2.2.12. Key Compliance Check

Key data recorded in system including key length, type, and corresponding standard data. According to the requirements of the national standard, the system regularly checks the key quality and compliance by checking the data encryption and decryption process by calling the key, compares the standard data, and checks if the key does not meet the standard requirements.

2.2.13. Keystore Management

The keystore contains the backup library, the in-use library, and the archive library. In order to view the key usage and ensure the availability of the keystore, the keystore needs to provide the corresponding functions, including key query, key statistics, keystore backup and keystore restore.

2.3. Data Processing and Query

2.3.1. Data Processing

Validate and unzip the file package to obtain sensitive data files:

- According to the encryption strategy to analyse the contents of the file, read out the data representation and sensitive fields, and calculate the hash value of sensitive fields;
- Determine whether there is the cache of sensitive data values.
- If does not exist, call encryption and decryption module to encrypt sensitive data into the database, and the identity and hash value into the cache.
- Exists and the hash value is equal: discard the data.
- Exists and the hash value is not equal: call encryption and decryption module to encrypt sensitive data stored in the database, the original sensitive data marked as expired state. And replace the old hash value in the data cache with the new hash value.

2.3.2. Data Query

Performing an access authentication process;

- The application system call data query interface, enter the sensitive data identification;
- Use encryption and decryption to find identifier and sent to the key management system
- The key management system detects the key and sends it to the encryption machine;
- Encryption and decryption module detect the data ciphertext in the sensitive database , sent to the encryption machine;
- The encryption machine decrypts the sensitive data and returns it.

2.4. Interface Requirements

2.4.1. Internal Interface

The CA system is deployed on the public network, supports real-time calling interfaces, and provides certificate services through certificate import. As the sensitive data deployed in the network, so the call interface of CA system is not open real-time, use certificate into USB key way to provide services.

The external application system communicates with the system through the IF2 interface to request the submission and data information.

2.4.2. External Interface

External interface include IF3, IF4, IF5, IF6 interface.

- IF3 interface: Data acquisition platform through the IF3 interface will be collected to provide the original data to the system.
- IF4 interface: The system communicates with the encryption machine cluster through the IF4 interface, provides the data to be encrypted or the ciphertext data to the encryption machine cluster, calls the IF4 interface to complete the data encryption and data decryption process.
- IF5 interface: The system interacts with the key management system through the IF5 interface, provides the key identifier for the unencrypted data to be encrypted, provides the key for the encrypted data to be used for decryption, and encrypts the data with the key in the key management system Corresponding relationship into the database.
- IF6 interface: The encryption machine obtains the key number by querying the database, obtains the key number, and obtains the key required for the encryption/decryption through the IF6 interface from the key management system to perform the encryption/decryption.

3. Conclusions

A sensitive data encryption system is proposed, includes various functions, data processing and query flow and interface requirements. Through the practical application, it proves that the system can encrypt the sensitive data effectively and improve the security of the sensitive data.

References

[1] Roan Simõcs da Silva, (2016)On the use of proxy re-encryption to control access to sensitive data on information centric networking, 2016 International Conference on Information Networking (ICOIN), 7 - 12

[2] P. Fanfara; E. Danková; M. Dufala, (2012) Usage of asymmetric encryption algorithms to enhance the security of sensitive data in secure communication, 2012 IEEE 10th International Symposium on Applied Machine Intelligence and Informatics (SAMI), 213 - 217

[3] Ricardo Rodriguez Garcia; Julie Thorpe, (2014) Crypto-assistant: Towards facilitating developer's encryption of sensitive data, 2014 Twelfth Annual International Conference on Privacy, Security and Trust, 342 - 346

[4] Qi Dejiang; Tang Henan; Yan Hui,(2011) Research on sensitive data encryption based on B/S network, 2011 International Conference on Mechatronic Science, Electric Engineering and Computer (MEC), 2209 – 2212

[5] Necla Bandirmali; Ismail Erturk; Celal Ceken, (2009) Securing Data Transfer in Delay-sensitive and Energy-aware WSNs Using the Scalable Encryption Algorithm, 2009 4th International Symposium on Wireless Pervasive Computing, 1 - 6