Research on Data Security Protection Methods for Enterprise Private Clouds: Postprint

Authors: Chenzhuang, Qi Feng

Date: 2019-01-03T00:00:00+00:00

Abstract

To address the challenges of data secure storage and integrity verification in existing enterprise private clouds, a novel data linear scrambling hybrid encryption protection method is proposed. First, fine-grained data partitioning and linear segmentation are performed prior to data encryption; second, each segmented sub-data block undergoes data scrambling processing; finally, hybrid encryption and integrity verification are conducted on the scrambled data blocks using domestic cryptographic algorithms. The proposed algorithm is compared with SM4 and SM2 encryption algorithms, and experiments are conducted to evaluate its correctness, supported file types, encryption/decryption efficiency, and security. Experimental results demonstrate that, compared with the other two encryption algorithms, the proposed algorithm achieves substantial security improvements while maintaining encryption/decryption efficiency.

Full Text

Preamble

Vol. 37 No. 3
Application Research of Computers
ChinaXiv Partner Journal

Research on Data Security Protection Methods for Enterprise Private Clouds

Chen Zhuang, Qi Feng
(College of Computer Science & Engineering, Chongqing University of Technology, Chongqing 400054, China)

Abstract: To address the challenges of secure data storage and integrity verification in existing enterprise private clouds, this paper proposes a novel data linear scrambling hybrid encryption protection method. The approach involves

three key stages: first, performing fine-grained data partitioning and linear division before encryption; second, applying data scrambling processing to each partitioned sub-block; and finally, employing domestic cryptographic algorithms for hybrid encryption and integrity verification of the scrambled blocks. The proposed algorithm is compared against SM4 and SM2 encryption algorithms through experimental evaluation of correctness, supported file types, encryption/decryption efficiency, and security. Results demonstrate that the proposed algorithm achieves substantial security improvements while maintaining efficient encryption and decryption performance compared to the other two algorithms.

Keywords: private cloud; domestic cryptographic algorithm; data linear scrambling hybrid encryption; data integrity

0 Introduction

In recent years, cloud computing technology has rapidly developed, gradually replacing traditional IT infrastructure and heralding what is considered the third IT wave. With broad application prospects in IoT, mobile computing, and big data industries, cloud computing offers advantages such as on-demand scalability, storage consolidation, and cost-effectiveness, making it increasingly popular among enterprise users. While there is currently no unified global standard for cloud computing classification, industry consensus categorizes services into Infrastructure-as-a-Service (IaaS), Platform-as-a-Service (PaaS), and Software-as-a-Service (SaaS), with operational models comprising public cloud, private cloud, and hybrid cloud.

However, the rapid development of cloud computing has been accompanied by increasingly prominent data security issues and continuous security incidents. For example, in September 2016, Yahoo announced during negotiations with Verizon that at least 500 million usernames, passwords, and phone numbers had been leaked. In June 2017, a database hosted on AWS S3 by marketing firm Deep Root Analytics was exposed, compromising voting data of nearly 200 million Americans. In November 2017, reports emerged that student data from Qudian may have been leaked. Consequently, an increasing number of enterprise users are opting for private cloud deployments to address data security concerns, primarily because enterprises retain control over their data and can deploy infrastructure within their own data center firewalls. While firewalls can effectively block attacks from the physical to application layers, they cannot prevent illegal intrusions at the business logic or data layers, such as SQL injection, data scraping, or privilege abuse, and thus cannot fundamentally prevent data leakage.

To ensure data security in enterprise private cloud environments, this paper proposes a novel data security protection method that does not rely entirely on keys: the Data Linear Scrambling Hybrid Encryption (DLSHE) algorithm.

1 Related Work

Private clouds offer higher security compared to public and hybrid clouds, as users do not lose control over their data. However, they still face the risk of core enterprise data leakage, merely narrowing the scope of potential information exposure. Current research in academia and industry primarily focuses on two aspects: secure cloud data storage and security auditing.

In encryption storage technology, most cloud providers currently offer encryption services based on U.S. cryptographic standards. Modern cryptography can be broadly classified into symmetric and asymmetric encryption algorithms based on key types. Symmetric encryption uses identical encryption and decryption keys, offering high efficiency, with common algorithms including AES, DES, and 3DES [2~4]. Asymmetric encryption employs different keys for encryption and decryption, providing higher security, with typical algorithms such as RSA and ECC [5,6]. While significant research exists in homomorphic encryption and attribute-based encryption, these technologies have not achieved large-scale commercial deployment due to various limitations [7].

Security auditing technology primarily addresses data possession and integrity issues. In public and hybrid cloud scenarios, research focuses on untrusted or semi-trusted storage systems. Erway et al. [8] proposed a provable dynamic data possession mechanism supporting all dynamic operations, but with low execution efficiency. Wang et al. [9] introduced a cloud storage integrity auditing mechanism with substantial computational costs. Xu et al. [10] proposed an outsourced data authentication model that detects modifications by comparing server-returned evidence with pre-computed correctness proofs, but this scheme does not support public auditing. Private clouds can resolve untrusted storage system issues but still require robust data integrity verification mechanisms.

This work adopts domestically certified cryptographic algorithms recognized by the State Cryptography Administration, which have been widely deployed in banking, telecommunications, healthcare, and other industries in China, demonstrating practical applicability. In terms of algorithm security, internationally standardized hash algorithms such as MD5, SHA-1, and HAVAL have been compromised by Chinese cryptographer Wang Xiaoyun [11,12]. Following the public release of Chinese cryptographic algorithms, numerous domestic and international experts have conducted comprehensive security evaluations. Current public assessments indicate that Chinese cryptographic algorithms offer overall superior security compared to their U.S. counterparts [13-15].

2.1 Algorithm Design

The data scrambling concept originates from signal scrambling techniques in military communications to prevent unauthorized eavesdropping. In cryptography, data scrambling effectively conceals plaintext statistical characteristics and enhances diffusion resistance. The DLSHE algorithm is a data linear scrambling hybrid encryption method based on domestic cryptographic algorithms SM2,

SM3, and SM4 [16~18]. The algorithm focuses on five key aspects: encryption granularity, linear division lines, data scrambling control, integrity verification, and dual-key management.

Encryption granularity refers to the initial minimal partitioning of plaintext data. Linear division lines determine the proportion for data segmentation, where data before the division line is encrypted using SM4, and data after the line uses SM2. The combination of encryption granularity and linear division lines produces sub-plaintext blocks. Data scrambling involves reading each sub-block row-by-row and appending the last three digits of the current time function' s millisecond value to each row, generating truly randomized scrambled plaintext. Integrity verification comprises message digest algorithms and digital signature algorithms to prevent illegal data tampering and user repudiation. Dual-key management involves encrypting the work key with a protection key, storing the work key in ciphertext form in the cloud while users only need to store the protection key locally. The DLSHE algorithm 流程 is illustrated in [Figure 1: see original paper].

The detailed DLSHE encryption process is as follows: a) Read the plaintext data to be encrypted. b) Partition the plaintext data according to the segmentation strategy, creating sub-plaintext blocks. The segmentation strategy is determined by encryption granularity and linear division lines, which users can dynamically configure based on requirements. c) Read each sub-data block for scrambling processing, appending time function values to generate scrambled sub-data. d) Encrypt the scrambled sub-data using alternating SM4 and SM2 algorithms to produce sub-ciphertext blocks. e) To prevent tampering and repudiation, generate a message digest of the plaintext using SM3, then create a digital signature with the SM2 private key. The signature data is scrambled and encrypted with the SM2 public key. f) Merge all sub-ciphertext blocks to form the final ciphertext. g) Encrypt the SM4 private key (work key) with the SM2 public key to create a key ciphertext stored in the cloud, while users only store the SM2 private key (protection key) locally.

The detailed DLSHE decryption process is as follows: a) Decrypt the key ciphertext using the SM2 private key to obtain the SM4 private key. b) Split the ciphertext data into sub-ciphertext blocks. c) Decrypt sub-ciphertext blocks according to the decryption strategy, alternating between SM4 and SM2 algorithms to produce scrambled sub-data. d) Descramble each sub-data block to recover sub-plaintext data. e) Merge all sub-plaintext blocks to reconstruct the original data. f) Decrypt the signature ciphertext using SM2 to produce a scrambled digital signature, then descramble to obtain the signature and verify it against the original message digest. g) Generate a new message digest from the decrypted plaintext and compare it with the original. If they match, the data is intact; otherwise, tampering or signature issues are detected.

2.2.1 Key Security Analysis

A fundamental principle of modern cryptography is that algorithm security depends entirely on key confidentiality. DLSHE employs a dual-key management mechanism to enhance secure key storage. The mechanism works as follows: DLSHE keys consist of work keys and protection keys. Work keys comprise the SM4 private key and SM2 public key for data encryption/decryption, with only the SM4 private key portion encrypted and stored as ciphertext in the cloud. Protection keys consist of the SM2 private key stored locally on the client. Decrypting ciphertext data requires the protection key to decrypt the work key, meaning both keys are necessary for data access.

While the dual-key mechanism ensures work key security, users must still securely store protection keys. To address this, DLSHE incorporates multi-factor security parameters, creating a security mechanism that does not rely entirely on key protection. Details are provided in Section 2.2.3.

2.2.2 Anti-diffusion Analysis

In cryptography, diffusion refers to the property where changing a single plaintext bit affects multiple ciphertext bits, thereby concealing plaintext statistical characteristics. This study selected identical plaintext and keys for experimental comparison across three algorithms, changing only one plaintext bit per test and measuring ciphertext bit changes across six experimental groups. Anti-diffusion test results are shown in [Figure 2: see original paper].

Experimental Platform: A private cloud storage environment was built using Hadoop consisting of six computers with CPU i3-2120 at 3.3 GHz, 4 GB RAM, and 1 TB hard drives, running CentOS 6.5. A LAMP-based cloud storage encryption system was developed to simulate enterprise private cloud data operations.

As shown in [Figure 2: see original paper], when changing a single plaintext bit, the average number of changed ciphertext bits was approximately 68 for SM4, 73 for SM2, and 87 for DLSHE. Since DLSHE adds scrambling data equivalent to multiple plaintext bit changes, it exhibits the most ciphertext bit variation. Experimental analysis demonstrates that DLSHE provides the strongest diffusion capability, most effectively concealing plaintext statistical characteristics.

2.2.3 Anti-attack Analysis

DLSHE' s attack resistance does not rely solely on cryptographic algorithms and keys but also depends on data scrambling methods, encryption granularity, and linear division lines. Table 1 summarizes security parameter dependencies, where A1 represents algorithm security, A2 key security, A3 data scrambling method, A4 encryption granularity, A5 linear division lines, Y indicates dependency, and N indicates non-dependency, with comparison to SM4 and SM2.

Even if attackers obtain partial plaintext and ciphertext, the scrambling process destroys the direct correspondence between them. Since encryption granularity and division lines are user-configurable, attackers cannot easily decrypt the information. Even with stolen keys, attackers must know the algorithm, encryption granularity, division line positions, and scrambling method to recover original plaintext. Thus, DLSHE security depends on multiple factors rather than single-key protection, significantly enhancing attack resistance and providing robust security guarantees for private cloud environments.

3.1 Cloud Storage Encryption System

The cloud storage encryption system meets enterprise user requirements for basic file operations, including upload, download, modification, deletion, and sharing. Fine-grained access control allows users to view and download public files, while system administrators with appropriate keys gain full operation privileges. Enterprise users logging into their personal cloud space have full control over their files and can use the DLSHE algorithm module to encrypt core data. This section focuses on the DLSHE encryption/decryption module, which provides key generation, message digest creation, digital signing, and DLSHE encryption/decryption functions, as detailed in [Figure 3: see original paper].

3.2 DLSHE Algorithm Implementation

a) Correctness Testing: Two lines of plaintext were selected as test data, with encryption granularity using this data as the minimum unit and a linear division ratio of 9:1. SM4 operated in CBC mode with initialization vector EE0CAD6B0059D77EE3D4F0F16CA087E35. Results are shown in Table 2 .

The results demonstrate that plaintext processed by DLSHE produces corresponding ciphertext, which after splitting, descrambling, and decryption yields original plaintext, message digest, and digital signature identical to the originals, proving DLSHE meets encryption/decryption requirements for enterprise private cloud deployment.

b) File Type Testing: Multiple file types including txt, doc, png, and zip were tested for encryption/decryption (detailed in Table 3), all achieving expected results. Since DLSHE uses file I/O stream-based processing, it supports arbitrary file formats, extending its application scope. To demonstrate encryption effects, a 0.23 MB txt file was encrypted using the DLSHE module shown in [Figure 3: see original paper]. The process involves: (a) key generation creating SM4 private key, SM2 public key, and SM2 private key files; (b) message digest generation from selected plaintext; (c) signing the digest; (d) scrambling the signature; (e) encrypting the scrambled signature with SM2; and (f) performing DLSHE encryption using SM4 private and SM2 public keys to merge all ciphertext. The encrypted txt file is shown in [Figure 4: see original paper].

c) Efficiency Testing: A 1.36 GB compressed file containing txt, pdf, doc, avi,

and other formats was tested. Encryption granularity was set to 2056 bits with a 9:1 division ratio. Hadoop's default block size is 64 MB; to test efficiency, blocks of 2 MB, 4 MB, 8 MB, 16 MB, 32 MB, and 64 MB were used. DLSHE, SM4, and SM2 algorithms were compared across six encryption and six decryption experiments, with average computation times shown in [Figure 5: see original paper] and [Figure 6: see original paper].

Analysis reveals that Hadoop block size significantly impacts efficiency, with optimal stable performance at 16 MB, 32 MB, and 64 MB. Smaller blocks (e.g., 2 MB) drastically increase processing time due to frequent algorithm invocation and distributed sorting overhead. Enterprises should carefully select appropriate block sizes to avoid resource waste. SM2 exhibits the highest time consumption, SM4 the lowest, and DLSHE slightly higher than SM4, though users can adjust division lines to reduce time. With only six ordinary computers in the test cluster, larger-scale deployments would narrow these time differences further.

d) Security Testing: DLSHE security comprises two components: the security of SM4 and SM2 algorithms, and the secrecy of encryption strategies.

Algorithm Security Comparison: SM4' s best-known resistance results include 23 rounds against differential cryptanalysis [19], 23 rounds against linear cryptanalysis [20], 23 rounds against multidimensional linear attacks [20], 18 rounds against rectangle attacks [21], and 14 rounds against integral attacks [22], with no full-round attacks known. SM2 is based on generalized ElGamal but requires Hash function enhancements to achieve IND-CCA2 security against chosen ciphertext attacks [23]. For digital signatures, it employs EUF-CMA standards against chosen message attacks and uses signer ID, public key, and message hashing to defend against key substitution attacks [24]. While SM2 offers superior security to SM4, its computational complexity is higher.

DLSHE combines both algorithms' attack resistance while exceeding each individually. Its dual-key management enhances key storage security, data scrambling provides superior diffusion (see Section 2.2.2), and undisclosed encryption strategies (granularity size, division line position, scrambling method) further strengthen security. In the cloud storage system, encryption granularity and division lines are user-configurable, while scrambling uses current time functions, introducing randomness that prevents attackers from identifying patterns. Security levels were tested across six experimental groups, rated on a 0-1 scale. While SM4 and SM2 security remained constant, DLSHE security varied: groups 2, 4, and 6 with larger granularity and simple division ratios showed reduced security, while groups 1, 3, and 5 with smaller granularity and complex ratios showed increased security, approaching the maximum level of 1 under ideal conditions, as shown in [Figure 7: see original paper].

4 Conclusion

Enterprise private cloud data security encompasses secure storage and integrity verification. This paper proposes the DLSHE algorithm using data linear scram-

bling hybrid encryption to enhance security through multi-factor dependencies, dual-key management for improved key storage, and message digest/digital signature mechanisms for tamper prevention and non-repudiation. Experimental validation demonstrates that DLSHE meets enterprise private cloud security requirements while providing substantial security improvements over traditional encryption algorithms.

References

[1] Feng Dengguo, Zhang Min, Zhuang Yan, et al. Study on cloud computing security [J]. Journal of Software, 2011, 22(1): 71-83.

[2] Yan Lele, Li Hui. Dynamic key AES encryption algorithm based on compound chaotic sequence [J]. Computer Science, 2017, 44(6): 133-138,160.

[3] Zhang Yijiang. Study on network information based on 3DES-ECC encryption algorithm[J]. Science and Technology Bulletin, 2014, 30(4): 229-231,235.

[4] Zhou Wenting, Zhu Jiaojiao. An improvement method to implement the DES encryption algorithm [J]. Computer Security, 2012, 18(9): 47-50.

[5] Xiao Zhenjiu, Hu Chi, Jiang Zhengtao, et al. Optimization of AES and RSA algorithm and its mixed encryption system [J]. Application Research of Computers, 2014, 31(4): 1189-1194,1198.

[6] Wang Kui, Li Lixin, Yu Wentao, et al. Design and optimization of TLS protocol based on ECC[J]. Application Research of Computers. 2014,31(11): 3486-3489.

[7] Feng Chaosheng, Qin Zhiguang, Yuan Ding. Techniques of secure storage for cloud data [J]. Chinese Journal of Computers, 2015, 38(1): 150-163.

[8] Erway C, Kupcu A, Papamanthou C, et al. Dynamic provable data possession [C]//Proc of ACM Conference on Computer and Communications Security. NewYork: ACM Press, 2009: 213-222.

[9] Wang Cong, Wang Qian, Ren Kui, et al. Towards secure and dependable storage services in cloud computing [J]. IEEE Trans on Service Computing, 2012, 5(2): 220-232.

[10] Xu Jian, Zhou Fucai, Chen Xu, et al. Data outsourcing authentication model based on authenticated data structures for cloud computing [J]. Journal of Communications, 2011, 32(7): 153-160.

[11] Bai D, Yu H, Wang G, et al. Improved boomerang attacks on round-reduced SM3 and keyed permutation of BLAKE256 [J]. IET Information Security, 2015, 9(3): 167-178.

[12] Wang X, Feng D, Lai X, et al. Collisions for hash functions MD4, MD5, HAVAL-128 and RIPEMD [EB/OL].(2004-07-17) [2018-08-10]. https://eprint.iacr.org/2004/199.pdf.

[13] Zhao Shijun, Xi Li, Zhang Qianying, et al. Security analysis of SM2 key exchange protocol in TPM-2.0 [J]. Security & Communication Networks, 2015, 8(3): 383-395.

[14] Christina B, Naya-Plasencia M, Suder V. Scrutinizing and improving impossible different attacks: Applications to CLEFIA, Camellia, LBlock and Simon [C]//Proc of International Conference on the Theory and Application of Cryptology and Information Security. Berlin: Springer, 2014: 179-199.

[15] Todo Y. Integral cryptanalysis on full MISTY1 [C]//Advances in Cryptology. Berlin: Springer, 2015: 413-432.

[16] Wang Zhaohui, Zhang Zhenfeng. Overview on public key cryptographic algorithm SM2 based on elliptic curves [J]. Journal of Information Security Research, 2016, 2(11): 972-982.

[17] Wang Xiaoyun, Yu Hongbo. SM3 cryptographic hash algorithm [J]. Journal of Information Security Research, 2016, 2(11): 983-994.

[18] Lyu Shuwang, Su Bozhan, Wang Peng, et al. Overview on SM4 algorithm [J]. Journal of Information Security Research, 2016, 2(11): 995-1007.

[19] Su Bozhan, Wu Wenling, Feng Dengguo, et al. Security of the SM4 block cipher against differential cryptanalysis [J]. Journal of Computer Science and Technology, 2001, 26(1): 130-138.

[20] Liu Mingjie, Chen Jiazhe. Improved linear attacks on the chinese block cipher standard [J]. Journal of Computer Science and Technology, 2014, 29(6): 1123-1133.

[21] Xue Ping. Rectangle attack of reduced SMS4 block cipher [D]. Jinan: Shandong University, 2012.

[22] Zhong Mingfu, Hu Yupu, Chen Jie. Square attack on the 14-round block cipher SMS4. Journal of XiDian University: Natural Science, 2008, 35(1): 105-109.

[23] Cramer R, Shoup V. A practical public key cryptosystem provably secure against adaptive chosen ciphertext attack[C]//Proc of International Cryptology Conference on Advances in Cryptology. Berlin: Springer, 1998: 13-25.

[24] Zhang Zhenfeng, Yang Kang, Zhang Jiang, et al. Security of the SM2 signature scheme against generalized key substitution attacks [C]//Proc of International Conference on Research in Security Standardisation. Berlin: Springer, 2015: 140-153.

Note: Figure translations are in progress. See original paper for figures.