AI Endorsed Transactions
on Scalable Information Systems
Research Article EAI.EU

Research on video encryption algorithm based on SM4-CFB mode

Hongjie Tang

Liaoning Police College, Dalian, Liaoning, China

Abstract

INTRODUCTION: There are challenges of video data leakage, unauthorized and misconnected devices, and the potential for illegal tampering of video surveillance information in current public security IP video surveillance systems.
OBJECTIVES: To solve the problems, a video encryption scheme based on the Cipher Feedback (CFB) mode of the nationally recognized SM4 algorithm was proposed.
METHODS: A complete video encryption and decryption process framework was designed. To verify the effectiveness of the scheme, an experimental platform was established, and multiple test video sequences with varying characteristics were selected. Comparative experiments and analysis were conducted with the AES-CFB and DES-CFB algorithms from multiple perspectives, including subjective visual characteristics, objective statistical characteristics, encryption efficiency (and anti-interference capability.
RESULTS: The experimental results demonstrate that the proposed SM4-CFB video encryption scheme effectively destroys the statistical characteristics of video data, resulting in a visually scrambled video after encryption. Its encryption efficiency is comparable to that of AES and significantly higher than that of DES, meeting the requirements of real-time video transmission and demonstrating robustness to common interference during transmission.
CONCLUSION: It is a secure, efficient, and practical video encryption solution.

Keywords: SM4 algorithm; CFB mode; Video encryption; Video surveillance

Received on 04 February 2026, accepted on 11 May 2026, published on 27 May 2026

doi: 10.4108/eetsis.11830

*Corresponding author. Email: thongjie@163.com

1. Introduction

With the in-depth implementation of the "Internet +" strategy and the rapid development of 5G technology, network video applications such as video conferencing, online live broadcasting, telemedicine, and smart security have penetrated into all aspects of social life [1]. However, when video data is transmitted in public network channels, it faces many security threats such as eavesdropping, tampering, copying, and copyright infringement. Therefore, secure encryption of video content and protection of user privacy have become a key issue that needs to be addressed urgently [2]. Traditional file encryption algorithms have high computational complexity and are difficult to meet the low latency requirements of real-time transmission of high-definition video [3]. Therefore, it is of great significance to study a video encryption algorithm that can ensure both security and real-time performance.

Current video encryption methods can be generally classified into two basic strategies: full encryption and selective encryption. In the full encryption scheme, Cheng et al. [4] designed a scheme that combines the H.264/AVC video encoding process with the Advanced Encryption Standard (AES) to achieve encryption of video content, in which the encryption key is dynamically generated and updated by a pseudo-random number generator (PRNG). Although this scheme provides relatively reliable protection for video data, its use of traditional high-intensity encryption algorithms also significantly increases the computational overhead, resulting in reduced encryption and decryption

efficiency, making it difficult to meet the low-latency performance requirements in real-time video communication. In terms of selective video encryption, Hamidouche et al. [5] proposed a scheme based on scalable HEVC extension (SHVC). This scheme uses a multi-layer coding structure to adapt to video representations at different spatial levels and introduces a chaotic coding mechanism to enhance security. SHVC helps to reduce the computational load of the encoder or transcoding server, but its coding efficiency decreases as the number of layers increases, and it also increases the decoding complexity to a certain extent. On the other hand, Liang Jian and He Junhui [6] developed an encryption method for H.264/AVC. This method generates a virtual random sequence frame by frame based on the macroblock type, and uses it to randomly permute the residual coding parameters and residual data during the macroblock encoding process, thereby realizing the encryption of the prediction mode in the I frame and the motion vector difference symbol in the P frame and B frame. Although the above method improves the execution efficiency of video encryption to varying degrees, its protection mechanism mainly focuses on the video data content itself and does not extend to the security level of control signaling.

To address the high computational complexity, long processing latency, and inability to meet real-time transmission requirements of full video encryption methods, this study proposes a comprehensive security enhancement solution. This solution leverages a national secret algorithm to achieve two security enhancements: first, a two-way authentication mechanism based on digital certificates is introduced at the control signaling level to ensure the trustworthiness and integrity of signaling transmission; second, selective encryption of the video data itself, based on a national secret algorithm, balances security and processing efficiency.

2 Research on Video Encryption Algorithms

2.1 Video data structure and characteristics

Modern video coding standards use a hybrid coding framework and employ related techniques to remove redundant information[7]. Video data has strong spatial and temporal correlations, which manifests itself in the raw pixel domain as highly similar values of adjacent pixels and uneven distribution of their grayscale histograms[8]. One of the goals of encryption is to break this correlation, so that the encrypted data has statistical properties similar to white noise.

H.264 is a widely used technical standard for video compression. Its main goal is to efficiently compress video data to reduce the network bandwidth occupied during transmission [9]. The encoder converts the original video signal into a bit stream that conforms to a specific syntax structure, namely the H.264 bit stream. The basic building block of the bit stream is the variable length coding (VLC) or fixed length coding (FLC) word. These code words are composed of multiple fields, each carrying different types of information and playing a corresponding role in the decoding process.

The H.264 bitstream includes structures such as the sequence parameter set (SPS), picture parameter set (PPS), and coded slices [10]. The SPS stores global parameters such as the SPS_id, profile level, image width and height, and number of reference frames; while the PPS contains information such as the picture parameter set id, the referenced SPS_id, entropy coding mode, reference frame list, and initial quantization parameter (QP). In audio and video coding applications, the SPS, PPS, and slice header only provide limited security mechanisms because the information they contain is relatively fixed and the format is relatively uniform. In fact, the key information in H.264 coding is mainly concentrated in the syntax structure of the slice layer, so parsing this layer can obtain richer video content information.

SVAC is one of the current mainstream video coding standards. In terms of coding performance, SVAC supports scalable coding, enabling the generation of video streams at various bit rates. In contrast, the H.264 standard achieves more efficient compression while maintaining high image quality, significantly reducing bit rates. In real-time transmission applications (such as video conferencing or streaming), H.264's lower encoding and decoding latency generally gives it an advantage over SVAC, which can introduce higher latency. Furthermore, SVAC exhibits excellent adaptability, dynamically adjusting video quality and bitrate based on real-time network conditions and device performance, enabling smooth, adaptive streaming. While H.264 also has certain adaptive mechanisms, its flexibility is inferior to SVAC. Taking into account coding efficiency, real-time performance, and widespread adoption, this article chooses to base video data processing on the H.264 standard, which is more suitable for real-time transmission scenarios.

2.2 SM4 block cipher algorithm

The SM4 algorithm was designated as a Chinese national standard in 2012 and is a block cipher algorithm[11]. Similar to DES and AES, SM4 is also a block cipher algorithm. Its overall framework includes three parts: encryption, decryption, and key expansion. The algorithm uses a 128-bit block and key in each round of processing, and both the encryption process and key expansion require 32 rounds of iteration.

In terms of encryption mechanism, SM4 is based on multiple rounds of basic function iterations, integrating ciphertext feedback and stream cipher processing methods [12]. Each round of encryption produces a four-word output and a one-word intermediate ciphertext, which is then used in the next round of operations. After all 32 rounds of iteration, a four-word ciphertext is finally generated. The entire encryption process can be compared to a sliding track

with a width of four words. After each round of encryption, the track moves backward by one word until all 32 rounds are completed. The encryption algorithm can be expressed as follows:

$$X(i+4)=F(X_i,X_{(i+1)},X_{(i+2)},X_{(i+3)},rki)$$
$$=X_i\oplus T(X_{(i+1)}\oplus X_{(i+2)}\oplus X_{(i+3)}\oplus rki),i=0,1,\ldots 31)$$

(1)

Where (X_0, X_1, X_2, X_3) are the input plaintexts, and rk_i is the round key.

As a symmetric encryption algorithm, the algorithm mechanism of SM4 determines that its encryption and decryption process is symmetric but reverse, that is, the decryption process is the reverse process of the encryption process [13]. The most important difference between the two is reflected in the calling order of the round keys. The decryption round key is the reverse sequence of the encryption round key. If the input ciphertext is (Y_0, Y_1, Y_2, Y_3), then the input round key is rk_i, i =31, 30, ..., 1, 0, then the algorithm can be expressed as:

$$X_{i+4}=F(X_i,X_{i+1},X_{i+2},X_{i+3},rk_i)$$
$$=X_i\oplus T(X_{i+1}\oplus X_{i+2}\oplus X_{i+3}\oplus rk_i),i=31,30,\ldots 1,0$$

(2)

$$(Y_0,Y_1,Y_2,Y_3)=(X_{35},X_{34},X_{33},X_{32})$$

(3)

The key generation process of the SM4 algorithm relies on a specific constant FK and a fixed parameter CK, which can be used to iteratively generate the specific key required for each round of encryption. The key expansion algorithm is as follows:

$$(K_0,K_1,K_2,K_3)=(SK_0\oplus FK_0,SK_1\oplus FK_1,SK_2\oplus FK_2,SK_3\oplus FK_3)$$

(4)

$$RK_i=K_{i+4}=K_i\oplus T'(K_{i+1}\oplus K_{i+2}\oplus K_{i+3}\oplus CK_i)$$

(5)

The variable i ranges from 0 to 31 and is used to iteratively generate all 32 round keys. The transformation function T' used in the decryption process is essentially the same as the round function structure used in encryption, differing only in the linear transformation L employed. The remaining components are the same. The SM4 algorithm boasts a rigorous and standardized structure, high security, and excellent efficiency in software implementation, providing a crucial cryptographic foundation for safeguarding my country's information security system.

2.3 Cipher Feedback (CFB) mode

The SM4 encryption algorithm supports multiple working modes, including CFB [14]. In CFB mode, the previous ciphertext block is used as input to encrypt the current plaintext block, thereby realizing the function of a stream cipher. This mode can directly encrypt without selecting a cryptographic algorithm and can process variable-length data streams, making it particularly suitable for real-time encryption and streaming scenarios. However, its encryption process is serial, making it difficult to implement parallel processing. After comprehensively comparing the above modes, given the advantages of the CFB mode in real-time and stream data processing, this paper selects the SM4 algorithm to encrypt video data in CFB mode.

The CFB mode does not require padding of plaintext and is particularly suitable for real-time encryption of streaming data (such as video and voice). The encryption process is as follows[15]:

(1) Initialization, store the initial vector (IV) into the shift register.
(2) Encryption: Use the key K to encrypt the contents of the shift register and obtain the encrypted result.
(3) Take the leftmost j bits of the encrypted result (usually j=8, i.e. one byte).
(4) Generate ciphertext and generate j-bit ciphertext through XOR operation.
(5) Shift, feed the ciphertext back to the shift register, and shift the register left by j bits.
(6) Repeat, repeat steps 2-5 until all plaintext is encrypted.

The decryption process is similar to the encryption process. You only need to change the plaintext input in step 4 to ciphertext input.

The CFB mode has the characteristic of error propagation. A single-bit error can affect the decryption of multiple subsequent blocks. However, its self-synchronization capability can ensure that the impact of the error is limited and it automatically recovers after a certain period of time. This feature makes it robust to transmission errors.

2.4 SM4-CFB mode video encryption algorithm

SM4 is an iterative block cipher algorithm that includes three main parts: encryption, decryption, and key expansion. The algorithm uses a 128-bit block length and a 128-bit key [16]. This study uses the SMC-CFB mode to encrypt the video stream. The session key is passed to the requesting end through the signaling channel after two-way authentication based on digital certificates during the signaling registration phase.

SM4-CFB mode is ideal for encryption and decryption of real-time audio and video data, meeting the stringent requirements for real-time performance, security, and low latency. This mode supports block-by-block processing of video data, enabling transmission of each block without waiting for the entire video to be fully encrypted. Each data block is independently encrypted and immediately transmitted, and the receiving end can also decrypt it in real time, effectively ensuring real-time and smooth communication.

Furthermore, if ciphertext errors occur during video data stream encryption, CFB mode offers excellent error containment: the error only affects the decryption of the current data block; subsequent data can still be decrypted normally if the ciphertext is correct. This significantly

improves the reliability of audio and video transmission. Furthermore, because the decryption of each block depends on the output of the previous ciphertext block, CFB mode offers self-synchronization capabilities. Even if data blocks are lost or transmission errors occur, the receiver can still recover decryption without relying on external synchronization mechanisms. The workflow of SM4-CFB mode is shown in Figure 1.

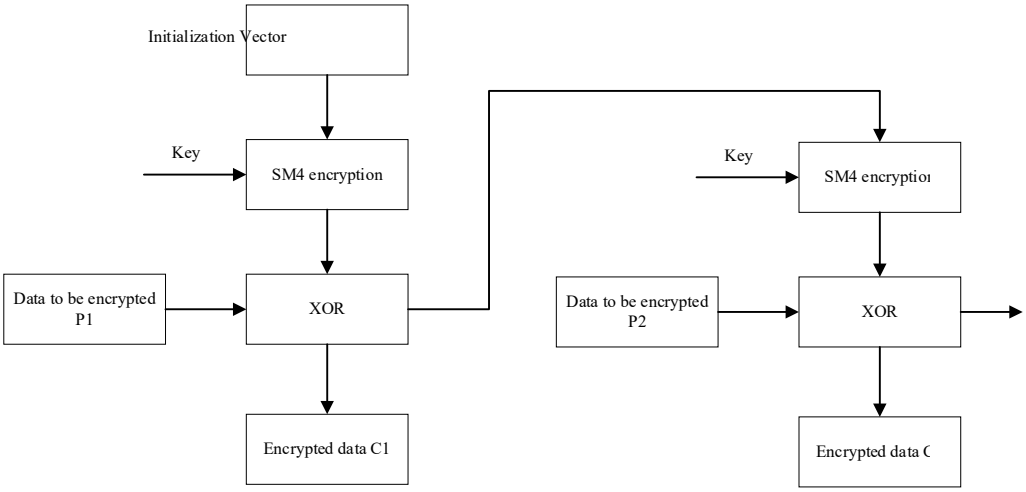


Figure 1. SM4-CFB operating mode

As shown in Figure 1, in the SM4-CFB mode, the method comprises the following specific steps of: (1) initializing a shift register and loading an initial vector IV; (2) encrypting the content of the register through using an SM4 algorithm; (3) taking the leftmost j bit of an encryption result and performing Xor with a plaintext to generate a ciphertext; (4) feeding the ciphertext back to the shift register and shifting j bits to the left; and (5) repeating the steps (2) to (4) until all data is encrypted. In the SM4-CFB mode, ciphertext packets are generated through an exclusive-OR operation. When using the SM4 algorithm to encrypt audio and video data, the data must first be converted to a block size that meets the algorithm's requirements. This article uses the PKCS#5 standard for padding.

Although SM4-CFB mode supports real-time data encryption, full encryption is still time-consuming due to the large volume of video data. Therefore, selective encryption is used to reduce the computational load. This paper chooses to selectively encrypt three types of syntax elements: IPM, MVD, and residual data in SM4-CFB mode.

H.264 video is encoded in slices. Key semantic elements within each slice are encrypted according to the formula $C_i = \text{Ekey}(C_i\text{-}1) \oplus P_i$. P_i is the current plaintext block, encrypted with the same key and of the same size as C_i, allowing the previous ciphertext block to be directly used to encrypt the current block. Key semantic elements are extracted from P_i to form a plaintext data block. If its length $L(P_i)$ is less than the standard block length L (128 bits), PKCS#5 padding is performed by adding zeros to the end of the data block until the length L is reached. This padding mechanism not only enhances algorithm security but also ensures that the encrypted bitstream remains compliant with the H.264 standard despite significant changes to the data content, effectively safeguarding video security.

After the audio and video data are collected and encoded, they are encrypted in the media access control module by calling the GmSSL open source cryptographic library[17]. Specifically, the SM4-CFB mode in the library is compiled and used to perform selective encryption operations on the video data.

3 Experimental results and data analysis

3.1 Experimental environment

To verify the effectiveness of this video encryption algorithm, a corresponding test environment was constructed for validation and analysis. The experimental environment consisted of an Intel Core i7-12700H @ 2.70GHz CPU, 16GB of RAM, and Windows 11. The algorithm was implemented and tested using Python 3.9 combined with the OpenCV library. Four standard test video sequences with varying resolutions and motion characteristics were selected for testing. Detailed information is shown in Table 1.

Table 1. Test video sequence information table

Sequence name	Resolution	Frame rate	Scene characteristics
Akiyo	352×288	30	Slight head movement, still background

Foreman	352×288	30	Medium-speed motion, complex background
Soccer	640×360	30	High-speed motion, complex scenes
Traffic	1920×1080	25	Large static area with rich details

For the images in this study, the average indicators used are mainly MAE, RMSE, PSNR and SSIM[18]. They can be expressed as:

$$MAE = \frac{|X - \hat{X}|}{M \times N} \tag{6}$$

$$RMSE = \sqrt{\frac{(X - \hat{X})^2}{M \times N}} \tag{7}$$

$$PSNR = 10 \times \log(\frac{L^2}{RMSE^2}) \tag{8}$$

$$SSIM = \frac{(2u_X u_{\overline{X}} + C_1)(2\sigma_X \sigma_{\overline{X}} + C_2)}{(u_X^2 + u_{\overline{X}}^2 + C_1)(\sigma_X^2 + \sigma_{\overline{X}}^2 + C_2)} \tag{9}$$

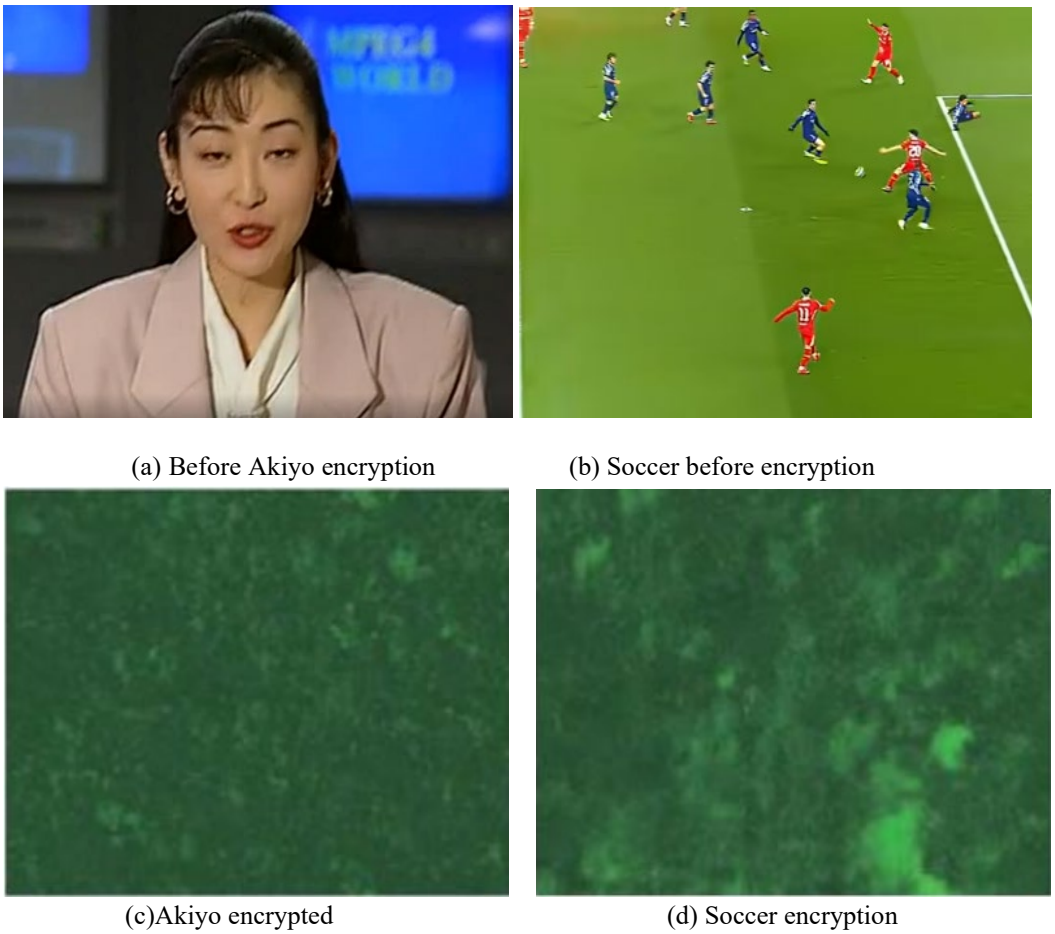
In the formula, μ is the image grayscale, C_1 and C_2 are constants, X is the original image, and $\overline{X}$ is the denoised image.

3.2 Comparison of experimental results

3.2.1 Security and effectiveness of encryption algorithms

This section experimentally evaluates the security and efficiency of the proposed video encryption algorithm using an H.264 encoder. The test used two video sequences with different motion characteristics: a CIF-formatted "Akiyo" video and a QCIF-formatted "Soccer" video. These videos contain dynamic objects such as a moving person and a flying soccer ball, with representative backgrounds and motion patterns.

The encryption process occurs in real time during video playback. This operation shifts key syntax elements in the H.264 bitstream, significantly blurring and distorting the video image's color, structure, and details, effectively ensuring the security of the video content. Figure 2 shows a comparison of the visual effects before and after encryption, with (a) and (b) showing the original images and (c) and (d) showing the encrypted images.



(a) Before Akiyo encryption (b) Soccer before encryption

(c)Akiyo encrypted (d) Soccer encryption

Figure 2. Video encryption effect

The comparison results in Figure 2 show that the encrypted video undergoes significant changes in color, texture, and detail, resulting in a highly distorted and blurred image that is difficult to visually identify or understand. This result demonstrates that encryption effectively protects the video content at the visual level.

3.2.2 Peak signal-to-noise ratio and structural similarity tests

In addition to subjective visual evaluation, this section quantitatively assesses the quality of encrypted videos. PSNR is a widely used objective evaluation method. It is generally considered that when the PSNR value falls below 20dB, the image content is unrecognizable. SSIM is more closely aligned with human visual characteristics and has a value between 0 and 1, with values closer to 1 indicating higher quality.

Table 2 lists the PSNR and SSIM values of the first 60 frames of the two test video sequences after encryption, which serve as an objective evaluation basis for the encryption effect.

Table 2. Comparison of different test sequences before and after encryption

Video sequence	Type	PSNR(dB)		SSIM
		Original sequence	Encryption Sequence	
Akiyo	CIF	37.83	11.28	0.25
Soccer	QCIF	36.73	8.92	0.15

As shown in Table 2, the PSNR values of the unencrypted original video sequences are all above 35 dB, indicating clear visual discernibility. However, the PSNR values of the encrypted videos all drop significantly to below 11.5 dB, well below the 20 dB threshold for human perception, indicating that encryption has severely degraded the video quality, making it unwatchable. Furthermore, the SSIM values of the encrypted videos approach 0, further demonstrating the significant difference in video quality between before and after encryption.

3.2.3 Encryption efficiency analysis

This paper conducts comparative experiments to evaluate the efficiency and performance of the National Security Agency's SM4 algorithm for video encryption. The experiments use the traditional symmetric encryption algorithm AES to encrypt MP4 files of varying sizes as a baseline for comparison. Our method selectively encrypts only key syntax elements within the H.264 bitstream, while the AES algorithm treats the entire H.264 bitstream as a binary file and encrypts it byte by byte. Table 3 shows the efficiency comparison between the two methods.

Table 3. Encryption time and bit rate analysis

Video sequence	File size before encryption	SM4 encryption takes time	SM4 decryption time	AES encryption takes time	AES decryption time	Encrypted file size
Akiyo	38 KB	467 ms	35 ms	511 ms	55 ms	36.8KB
Soccer	615 KB	673 ms	169 ms	809 ms	179 ms	624.7 KB

From Table 3, we observe that the size of the video file before and after encryption does not change significantly, which indicates that the encryption operation does not have a significant impact on the video quality, compression rate and bit rate. The experimental results in Table 3 show that the SM4 algorithm takes less time to encrypt the same H.264 file than the AES algorithm. During decryption, SM4 takes significantly less time than AES. From an algorithm design perspective, SM4 features longer keys and more encryption rounds, resulting in higher security against various attacks. Overall, SM4 offers advantages in both efficiency and security. The encrypted video does not occupy too much extra space during storage and transmission, nor does it increase the computational complexity and time overhead. Therefore, it has little impact on system performance.

Table 3 shows that the size of the video files before and after encryption does not change significantly, indicating that the encryption operation has no significant impact on video quality, compression ratio, or bit rate. Therefore, the encrypted video does not occupy excessive additional space during storage and transmission, nor does it increase computational complexity or time overhead, and has a minimal impact on system performance.

Figure 3 compares the amount of data processed by the traditional AES algorithm, which encrypts all data, and the

SM4 algorithm, which only encrypts the key syntax elements of H.264.

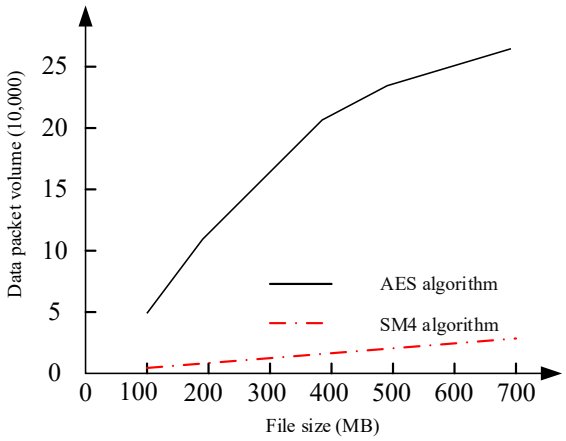


Figure 3. SM4 and AES encrypted data packets

As shown in Figure 3, as the data size of the H.264 file increases, the amount of data required to be processed by the SM4 algorithm used in this paper significantly decreases compared to the AES algorithm. This demonstrates that this encryption method not only reduces the actual encrypted data size but also improves overall encryption efficiency. Experimental results demonstrate that compared to traditional video encryption methods, SM4-based selective encryption achieves higher processing efficiency while ensuring video security. Therefore, this method has great application potential in the field of video security protection.

The complexity of this article's video data encryption stems primarily from the extraction and encryption of key H.264 syntax elements. H.264's slice-level syntax structure makes it easy to locate elements within the bitstream, such as slices, macroblocks, and their headers. Consequently, the time required to extract key syntax elements is minimal. Furthermore, this article uses the SM4-CFB mode for selective encryption of key syntax elements. This mode inherently supports encrypted transmission of real-time media data, making the entire encryption process less time-consuming. Figure 4 illustrates the ratio of the amount of key syntax data to the total data volume of the corresponding slice during encryption.

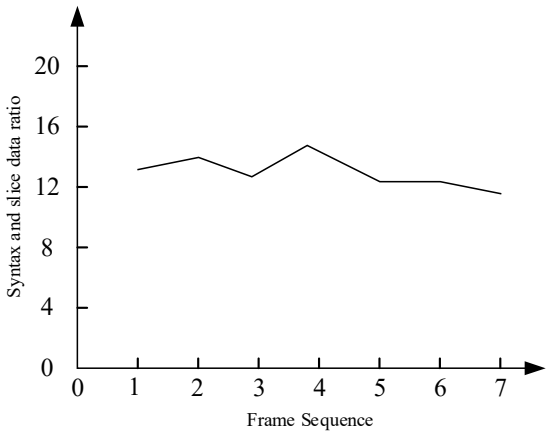


Figure 4. Ratio of encrypted key semantic element data to slice data

In video encryption, efficiency is often measured by the amount of data processed. The larger the amount of data to be encrypted, the lower the overall encryption efficiency. As shown in Figure 4, the encrypted key syntax element data only accounts for approximately one-quarter of the total data in the corresponding slice. Therefore, the encryption method employed in this paper significantly reduces the actual amount of data to be processed, effectively improving video encryption efficiency.

3.2.4 Comparison of different encryption algorithms

To evaluate the performance, the SM4-CFB algorithm implemented in this paper is compared with the AES-CFB (128-bit) [19] and DES-CFB algorithms [20] which also use CFB mode.

Adjacent pixels in different directions are randomly selected and their correlation coefficients are calculated. The comparison results are shown in Table 4

Table 4. Comparison of adjacent pixel correlation coefficients

Video sequence	Direction	Original image	After SM4-CFB encryption	After AES-CFB encryption	After DES-CFB encryption
Akiyo	level	0.9689	0.0012	0.0015	-0.0023
	vertical	0.9821	-0.0008	0.0004	0.0011
	diagonal	0.9577	0.0005	-0.0002	0.0008
Traffic	level	0.9915	-0.0011	0.0009	-0.0015
	vertical	0.9872	0.0007	-0.0006	0.0013
	diagonal	0.9798	0.0003	0.0012	-0.0009

The data in Table 4 show that after encryption using the three algorithms, the correlation coefficients are all very close to 0, indicating that there is almost no linear correlation between adjacent pixels and the encryption algorithm effectively destroys the spatial statistical characteristics of the image.

The information entropy of the video before and after encryption is calculated, and the results are shown in Table 5

Table 5. Image information entropy comparison table

Video sequence	Original image	After SM4-CFB encryption	After AES-CFB encryption	After DES-CFB encryption
Akiyo	7.21	7.9993	7.9994	7.9992
Foreman	7.58	7.9992	7.9991	7.9993
Soccer	7.45	7.9994	7.9993	7.9991
Traffic	7.32	7.9993	7.9992	7.9994

Table 5 shows that the information entropy of all test sequences after encryption is very close to the theoretical maximum value of 8 and significantly higher than the value before encryption. This indicates that the encrypted data is highly random and contains significant information uncertainty. SM4-CFB performs comparably to AES-CFB and DES-CFB on this metric.

The encryption efficiency of the algorithms was analyzed, and the encryption throughput (Mbps) and average encryption time per frame (ms) of each algorithm for video sequences with different resolutions were tested. The results are shown in Tables 6 and 7.

Table 6. Comparison of encryption throughput of different algorithms

Algorithm	Key length	Theoretical throughput
DES-CFB	56-bit	~90 Mbps
SM4-CFB	128-bit	~750 Mbps
AES-CFB	128-bit	~600 Mbps

Table 6 compares the theoretical throughput of the algorithms. The data shows that, in a software environment without hardware acceleration, SM4-CFB achieves the highest throughput, increasing by approximately 8 times compared to DES-CFB and boasting a 25% advantage over AES-CFB. Overall, SM4-CFB offers superior encryption efficiency while ensuring security.

The average encryption time (ms) per frame of each algorithm for video sequences with different resolutions is tested. The results are shown in Table 7.

Table 7. Comparison of encryption time of different algorithms

Video resolution	Approximate amount of data	DES-CFB time consuming	SM4-CFB time consuming	AES-CFB time consuming
QCIF (176×144)	~0.5 MB	12 ms	10 ms	15 ms
CIF (352×288)	~2 MB	48 ms	40 ms	60 ms
HD (720p)	~8 MB	190 ms	160 ms	240 ms
Full HD (1080p)	~20 MB	480 ms	400 ms	600 ms
4K (2160p)	~80 MB	1900 ms	1600 ms	2400 ms

Table 7 shows that, in a pure software environment, SM4-CFB is generally more efficient than AES-CFB, and both are significantly better than DES-CFB. DES-CFB is generally not recommended for video encryption because its security no longer meets modern requirements and its efficiency is lacking.

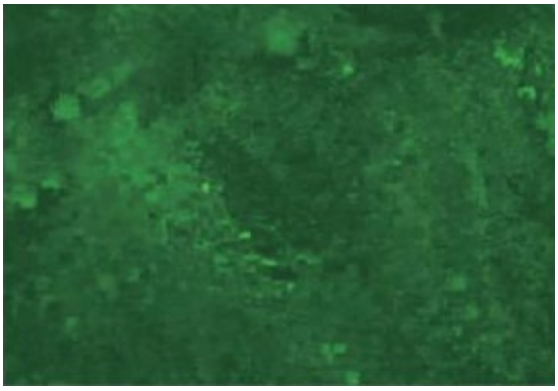
Experimental results show that SM4-CFB and AES-CFB offer comparable throughput, higher encryption efficiency, and shorter encryption times. When processing 1080p video, both achieve frame rates significantly higher than the native video frame rate, demonstrating their ability to fully meet real-time encryption requirements. However, DES-CFB, due to its inherently low number of iterations and short packet length, is computationally much less efficient than the previous two algorithms, making it inadequate for real-time encryption of high-definition video. Overall, the SM4-CFB encryption algorithm proposed in this study offers significant advantages.

3.2.5 Actual application effect

H.264 is a widely used standard format for video coding. To improve video encryption efficiency, this paper proposes a method for encrypting key syntax elements in H.264

bitstreams. This encryption technology is applicable to a variety of practical scenarios, including video surveillance and video conferencing. To verify the algorithm's effectiveness, this study selected an H.264 bitstream from a surveillance video for encryption. The encryption results are shown in Figure 5.

(a) Original video image



(b) Encrypted image

(c) Video decryption image

Figure 5. Video application effect

Figure 5(a) shows the original surveillance image, and Figure 5(b) shows the encrypted image. The encrypted image is severely distorted and blurred, making it unreadable to the human eye, effectively protecting the video content. The decrypted image, shown in Figure 5(c), is completely consistent with the original image upon comparison, demonstrating the good reversibility of the encryption algorithm. The results show that this method not only ensures the security of video data, but also does not affect the quality of the decrypted video, thus maintaining the integrity and security of the data.

4. Conclusion

This paper designs and implements a video encryption scheme based on the CFB mode of the national secret SM4 algorithm. The national secret SM4 algorithm is combined with the CFB stream cipher mode and systematically applied to the field of video encryption. A complete implementation scheme and performance evaluation are provided. The video encryption performance of SM4-CFB is comprehensively tested from multiple dimensions and angles, and compared horizontally with international general algorithms, demonstrating the excellent performance and usability of the national secret algorithm in this application scenario. Through theoretical analysis and experimental verification, it can be found that SM4-CFB encryption can effectively destroy the statistical characteristics of video data, causing its pixel correlation to approach zero and its information entropy to approach its maximum value, resulting in complete visual chaos and resistance to statistical analysis and ciphertext-only attacks. The algorithm is also highly efficient. The encryption efficiency of the SM4 algorithm is comparable to that of the international standard AES and far higher than that of DES, meeting the encryption requirements of real-time transmission of high-definition video.

This research still has room for improvement, and future work could be carried out in the following areas. Research on hardware implementations of the SM4-CFB cipher based on FPGAs or ASICs is needed to achieve higher throughput and lower power consumption. Research on deeper integration of encryption with steps like entropy coding in next-generation coding standards such as H.266/VVC is needed to further improve efficiency while ensuring security. Research on designing more robust joint encryption-coding schemes in highly error-prone wireless network environments, combining techniques such as forward error correction (FEC).

Acknowledgments

The study is supported by the “Research on Video Stitching Tamper Detection Based on Adjacent Image Features”, which belongs to the Joint Program of Science and Technology Plan of Liaoning Province (General Program of Natural Science Foundation of China). The project started from 2025-11-01, with the number of 2025-MSLH-337.

References

[1] Hao Yang, Zhou Hua, Wang Daiqiang. Design of H.264 video encryption scheme based on national secret SM2[J]. Electronic Measurement Technology, 2025, 48(12): 1-8.

[2] Yuan Zhi, Qiu Tian, Lin Zhanwu, Zhang Xin, Haiyan, Zhou Fei. Design of SM4 video encryption system based on FPGA[J]. Information Technology and Informatization, 2025, (01): 24-27.

[3] Tu Li,Liu Zhen,Wang Yan,Yang Gelan.A new video image encryption algorithm based on improved composite mapping[J].ITM Web of Conferences,2025,7701043-01051.

[4] Cheng,S.,et al. A selective video encryption scheme based on coding characteristics. Symmetry, 2020.12(3): p. 332-339.

[5] Hamidouche,W.,et al. Selective video encryption using chaotic system in the SHVC extension[C]. in 2015 IEEE International Conference on Acoustics, Speech and Signal Processing (ICASSP). 2015, 1660-1669.

[6] Liang Jian, He Junhui. H.264/AVC video encryption method based on adaptive replacement of macroblock coding information[J]. Computer Science, 2022, 49(01): 314-320.

[7] Lin Hao, Huang Yiping, Liang Zichen. Design and implementation of encrypted video playback system and its security analysis[J]. Electronic Design Engineering, 2024, 32(12): 150-156+161.

[8] Shen Zhelin, Xiong Xianming. Video file encryption and decryption system based on national secret algorithm SM2[J]. Industrial Control Computer, 2024, 37(04): 80-81+84.

[9] Zhang Yan, Wang Jinling. Research on user access identity identification algorithm for laboratory surveillance video based on attribute-based encryption[J]. Journal of Jiamusi University (Natural Science Edition), 2024, 42(07): 40-43.

[10] Chen Guanxu, Wu Siyao, Zhang Zifeng, Chen Yuming, Han Jingpeng. Research and application of privacy enhancement algorithm for video streams[J]. Science and Technology Vision, 2023, 13(35): 67-69.

[11] Xu Shengwei, Deng Ye, Liu Changhe, Tan Li. A selective encryption scheme for audio and video based on national secret algorithm[J]. Information Network Security, 2023, 23(11): 48-57.

[12] Parekh Aryan,Antani Mayav,Suvarna Kartik,Mangrulkar Ramchandra,Narvekar Meera. Multilayer symmetric and asymmetric technique for audiovisual cryptography[J].Multimedia Tools and Applications, 2023,83(11):31465-31503.

[13] Jin Wang,Jiandong Liu,Haoqiang Xu.H.264/AVC video encryption algorithm based on integer dynamic cross-coupling tent mapping model[J].Multimedia Tools and Applications, 2023,83(5): 13369-13393.

[14] Su Pengfei. Design of cloud-based video surveillance system for smart campus based on encryption algorithm[J]. Electronic Products World, 2023, 30(02):21-24.

[15] Suo Gao,Jiafeng Liu,Herbert Ho Ching Iu,Uğur Erkan,Shuang Zhou,Rui Wu,Xianglong Tang. Development of a video encryption algorithm for critical areas using 2D extended Schaffer function map and neural networks[J].Applied Mathematical Modelling, 2024, 134520-537.

[16] Tu Li, Wang Yan. A new video image encryption algorithm based on Lorenz-Duffing chaotic system[J]. Journal of Hunan City University (Natural Science Edition), 2022, 31(05): 67-72.

[17] Jin Hua, Zhu Jingyu, Wang Changda. Review of Video Privacy Protection Technology [J]. Computer Science, 2022, 49 (01): 321-328.

[18] Wang Chaoqing, Jia Liping. A digital video encryption method based on domestic cryptographic algorithm[J]. Television Technology, 2021, 45(07): 111-113.

[19] Gou Zhixiong, Xu Changxing, Xing Gengli, Fan Dianliang. An authentication and encryption method for Internet-based video surveillance system[J]. Police Technology, 2021, (04): 83-86.

[20] Tao Shuo, Wang Xiaofang, Chen Bin. Design of composite encryption algorithm for secure transmission of video surveillance system[J]. Journal of Anqing Normal University (Natural Science Edition), 2021, 27(02): 37-43.