Scalable secret sharing of compressed multimedia

Shreelatha Bhadravati[1], Pradeep K. Atrey[1,2], and Majid Khabbazian[3]

[1] *University of Winnipeg, MB, Canada*
[2] *State University of New York, Albany, NY, USA*
[3] *University of Alberta, AB, Canada*

Abstract

Traditional secret image sharing methods have an all-or-nothing property, which is not suitable for applications which require gradual reconstruction. In this paper, we propose a scalable secret image sharing (SSIS) method that provides gradual reconstruction with smooth scalability. Furthermore, we extend this method to videos and propose a scalable secret video sharing (SSVS) method. These two methods are designed for compressed multimedia (i.e. JPEG images and H.264 videos). In both methods, the size of the shadow images (shares) is reduced to an optimal value. Experimental results and analyses show that the proposed methods are computationally and semantically secure.

Keywords: Secret sharing, Scalable reconstruction, Compressed media

1. Introduction

Addressing the security concerns of multimedia data is a challenging problem as the size of the data is huge, therefore processing in real time is a constraint [23]. Furthermore, the diversified application areas of the digital content make it more challenging to develop security measures as each application can have different end users and specific requirements. For example, in the case of digital videos, in one scenario a low quality video is made available to all the users as a means to promote the content as in pay-TV, HDTV. Adversely, in applications like video conferencing and surveillance, the video must be completely unintelligible to unauthorized users, in order to preserve the privacy of the people and objects involved.

The task of securing digital images and videos has been studied extensively in the past. The conventional method of ensuring security is to encrypt the data using block encryption methods such as Advanced Encryption Standard (AES) [1]. Applying these block encryption methods, and therefore considering image [24] [36] or video [13] [20] [32] data to be byte data, is not a practical solution because of the computational complexity involved in the traditional encryption methods. To overcome this problem, selective encryption methods were proposed in which only certain parts of the image [7] or video [19] are encrypted, as opposed to encrypting the entire image or video data. However, these methods were insufficient in protecting the confidentiality of the data, as the unencrypted portions of the image and video revealed considerable amount of information. Further, the encryption methods require the use of a secret key. Storing the cryptographic key is also problematic as a single person cannot be entrusted with the security of the key. Storage of the key at a single place can lead to single point failure [6] and if the key is corrupted or lost, there is no way to reconstruct the original data. The key is further vulnerable to security attacks and can be compromised if stored at multiple places.

In 1979, Shamir [18] developed the secret sharing scheme (we call it "Shamir's Secret Sharing" (SSS) scheme throughout this paper) to overcome the problem of cryptographic keys. This method is said to be information theoretically secure. In this (k, n), $(2 \leq k \leq n)$ threshold scheme, the secret is divided into n shares and any k shares are required to reconstruct the secret. Any number of shares less than k cannot reconstruct the secret. Furthermore, the minimum size of the shares has to be equal to the size of the secret to be information theoretically secure [10]. Using SSS on digital data can make it information theoretically secure, provided the size of the shares is at least the same as the size of the secret data. However, this causes an increase in the storage space as it is similar to keeping n copies of the secret data. Hence, it would be ideal if the size of the shares was reduced to $1/k$ of the size of the secret data. Unfortunately, by reducing the size of the shares, the information theoretic security property of SSS is lost. Recently many methods have been proposed to reduce the size of the shares in the computationally secure model [3] [26]. The main idea in these methods, though not mentioned, is to use the Reed-Solomon error correction (RSEC) scheme [12], which is similar to SSS. The RSEC scheme can be used for information dispersal but it cannot be used for hiding information because it does not provide semantic security, as was shown in our previous work [5].

In a computationally secure model, a basic method is to encrypt the image using a secret key and then apply SSS on the secret key. In this method, every user gets the same copy of the encrypted image and a share of the secret key. Though this method may be computationally more efficient (depending on k, n and the encryption method), the size of the shares is still the same as the size of the original image. In [11], it was proposed to use the information dispersal methods along with the basic method discussed above in order to reduce the size of the shares to $1/k$ of the size the secret image.

The SSS scheme also has the property that either the secret image is reconstructed completely when k shares are available or it is not reconstructed at all. This all-or-nothing [9] property might not be suitable for certain applications, which require the secret data to be gradually reconstructed based on the number of shares available.

1.1. Paper Goal and Contributions

The goal of this paper is to provide a semantically and computationally secure method for scalable sharing of compressed images and videos. The proposed method can be used in the applications where a gradual reconstruction of secret images and videos is required[1].

The main contributions of this paper are described as follows:

1. We propose a (k, n) scalable secret image sharing (SSIS) method which provides scalability, such that when k shares are available the image reconstructed is of low quality, achieving the highest quality when all n shares become available. This method possesses semantic security and it reduces the size of the share images by $1/k$.
2. We extend this scalable method of image sharing to provide (k, n) scalable secret video sharing (SSVS) for compressed videos, where k shares are required to reconstruct the base layer of the video. As the number of shares available increases, enhancement layers of the video are reconstructed. To the best of our knowledge, this is the first attempt to propose a scalable secret sharing method on compressed videos.

We demonstrate the utility of the proposed method on compressed images (JPEG) and videos (H.264/SVC). We choose JPEG and H.264/SVC because these are the most widely used encoding standards for images and videos.

[1] A motivation video for this work can be found at: http://youtu.be/XGVXF9Rh3kU

1.2. Paper Organization

The rest of this paper is organized as follows. Section 2 contains the background details of image and video standards that are used in this paper. Also the existing secret sharing methods for images and videos is discussed, and the novelty of the proposed methods against them is highlighted. In Section 3, the proposed SSIS and SSVS methods are described. Section 4 provides the implementation details, performance results and security analysis of the SSIS and SSVS methods. Section 5 concludes this paper.

2. Background and Literature Review

In this section, the background details of compressed images and videos and the existing works related to secret image sharing is presented. In Section 2.1, JPEG and H.264/SVC compression standards are elaborated to get an understanding of how scalability is achieved. In Section 2.2, SSS, RSEC and the existing methods on SSIS and some of the encryption methods on compressed videos are discussed.

2.1. Background

2.1.1. JPEG

JPEG is the acronym for Joint Photographic Experts Group [27], which is the most widely used compression standard for images. In JPEG format, the encoding can be done in four modes. Baseline mode is the most widely used encoding method for JPEG. The second mode of encoding in JPEG format is progressive mode, which is gaining importance recently due to the development of devices with varying bandwidth requirements, though it has not been universally accepted. In the progressive mode of encoding, the images are compressed in multiple passes. The complete image is rendered in the first scan but with low quality and the quality of the image increases with respective scans. For the proposed SSIS method, the progressive mode of JPEG encoding is used, which helps to provide scalability for the compressed images.

There are two types of progressive mode JPEG encoding, which make use of the spectral characteristics of the Discrete Cosine Transform (DCT) coefficients. In each scan, the quantized DCT coefficients are partially encoded either by spectral or successive approximation. The progressive mode of JPEG encoding can also be customized based on the requirements of the user. To have a better understanding of these methods, a generalized description of the two types of progressive mode is presented next.

In spectral selection [22], DCT coefficients are divided into multiple bands and each band is encoded in the respective scan. A typical spectral scan contains the *DC* coefficients from each block of the image encoded in the first scan and the subsequent scans contain 6 - 7 *AC* coefficients, taking an equal number from every block.

In successive approximation, the DCT coefficients are encoded by using their binary representation [22]. In this method, the most significant bits are encoded in the initial scan and then the less significant bits are encoded in the later scans. A typical successive approximation method uses 6 scans, which are given as:
Scan 1: First 6 bits of the *DC* coefficients of all the blocks are encoded,
Scan 2: Next 1 bit of all the *DC* coefficients of all the blocks is encoded,
Scan 3: Last 1 bit of all the *DC* coefficients of all the blocks is encoded,
Scan 4: First 6 bits of all *AC* coefficients of the blocks are encoded,
Scan 5: Next 1 bit of all the *AC* coefficients of the blocks is encoded,
Scan 6: Last 1 bit of all the *AC* coefficients is encoded.

In this paper, a customized spectral type of encoding has been used for the proposed SSIS method for images. The discussion on the other modes of JPEG is omitted for brevity.

2.1.2. H.264/SVC

Traditional digital video transmission and storage systems are based on H.222.0/MPEG-2 standards. These systems are used for broadcasting services over satellite, cable and terrestrial channels and H.320 is used for conventional video conferencing services [17]. These channels are typically characterized by a fixed spatial-temporal format of the video signal. Modern digital video transmission and storage systems are typically characterized by wide range of connection qualities and receiving devices. The varying connection quality is due to the users' varying data throughput requirements. The varying receiving devices range from cell phones to high-end PCs. To fulfill these requirements, it is beneficial to transmit or store the videos with a variety of spatial, temporal resolutions or qualities. H.264/SVC (Scalable video coding) [17] is a promising solution which addresses these varying requirements of modern transmission systems. H.264/AVC (Advanced video coding) [31], which is the most widely used video encoding method, was extended to provide this scalability. H.264/SVC [17] has gained dominance because of its ability to provide good quality videos at lower bit rates. An SVC bitstream provides temporal, spatial and quality scalability by dropping a subset of the bitstreams from the high quality video bitstream. This extension benefits all applications of streaming, conferencing, surveillance, broadcast and storage.

As per the scalable extension of the H.264/AVC standard [17], the SVC bitstream consists of a H.264/AVC compatible base layer and one or more enhancement layers. The base layer contains fundamental information of the video stream while the enhancement layers contain additional data for video resolution, frame rate and picture quality. Scalability is achieved by removing parts of the video bit stream in order to adapt to the various needs and preferences of the end users, varying terminal capabilities and network conditions. The resulting substream with parts of the video removed forms another valid bitstream for some target decoder. SVC provides scalability most commonly in terms of spatial (video resolution), temporal (frame rate) and quality. The other rarely used scalability modes are region-of-interest and object-based scalability.

Temporal scalability is achieved by partitioning the access units into the temporal base layer and one or more temporal enhancement layers. Temporal base layers are used as references for motion-compensated prediction of frames for all the other temporal enhancement layers. Temporal enhancement layer pictures are typically coded as P or B pictures using a hierarchical prediction structure in which the pictures belonging to higher temporal resolution layers are predicted from pictures belonging to the same temporal resolution layers or lower resolution layers. Hence, a low bit rate SVC stream is obtained by discarding some pictures from the video bitstream to provide temporal scalability.

In spatial scalability the base layer is a stream of low resolution video. Adding the enhancement layer to the base layer stream, increases the resolution of the video bitstream. To support spatial scalability, SVC follows the conventional approach of multilayer coding. In each spatial layer, motion compensated prediction and intra-prediction mechanisms are employed. In order to improve the coding efficiency, additional inter layer prediction mechanisms are incorporated. Spatial enhancement layers are predicted from the lower level spatial layers and temporally neighbouring pictures.

Quality scalability is considered a special case of spatial scalability with identical picture sizes for base and enhancement layers [17]. The quality base layer is coded at a lower visual quality and the quality enhancement layers are predicted from the corresponding base layer and temporally neighbouring pictures. The inter-layer prediction mechanism without upsampling and the inter-layer deblocking for intra-coded reference layer macroblocks are employed to achieve quality scalability.

2.2. Literature Review

2.2.1. SSS

In the SSS method, the secret key is reconstructed using any k shares. Any $(k-1)$ shares cannot gain information about the secret key. The shares are generated using a polynomial function $f(x)$ of degree $(k-1)$ using k coefficients. The first coefficient a_0 is the secret number that needs to be protected and the other coefficients $a_1, a_2, \ldots, a_{k-1}$ are random numbers chosen from the finite field Z_q. The polynomial function $f(x)$ is shown as:

$$f(x) = (a_0 + a_1 x + a_2 x^2 + ... + a_{k-1} x^{k-1}) modq \tag{1}$$

In the above expression, q is a prime number higher than the secret number and the value of the function $f(x)$ is computed modulo q.

Any k shares $((1, f(1)), (2, f(2)), \ldots, (k, f(k)))$ can be used to reconstruct the polynomial (Eq 1) by using Lagrange interpolation as:

$$f(x) = f_j(x) \times \prod_{i=1, i \neq j}^{k} (\frac{x - x_i}{x_j - x_i}) modq \tag{2}$$

where, $f_j(x)$ is the value of the polynomial function generated using Eq 1. The secret value is determined by evaluating the equation $f(x)$ at $x = 0$.

2.2.2. RSEC

In 1960, Reed and Solomon proposed RSEC scheme, which are non-binary error-correction codes used in coding theory. This scheme can detect and correct multiple random symbol errors. Consider a message c which is b digit long, i.e. $c = (c_0, c_1, \ldots, c_{b-1})$, the polynomial $P(x)$ to generate the code word is given as:

$$P(x) = c_0 + c_1 x + c_2 x^2 + ... + c_{b-1} x^{b-1} \tag{3}$$

The Reed-Solomon codes are performed under the finite field. And any b pieces of the codeword are needed to reconstruct the information. SSS is considered a special case of RSEC where only the first coefficient is the message and the remaining coefficients are random values.

2.2.3. SSIS

The Secret Image Sharing (SIS) methods [26] [3] [4] [35] [3] [5] proposed earlier have the property that either the secret image is reconstructed completely with any k shares or it is not reconstructed at all. This all-or-nothing property is not suitable for applications where a secret image has to be gradually revealed based on the number of shares available.

In 2007, Wang et al. [28] proposed the SSIS method to overcome the all-or-nothing property of the previous SIS methods. In this (k, n)-SSIS method the secret image is divided into n shares such that no single share reveals information about the secret image. Any k, $(2 \leq k \leq n)$, shares are used to reconstruct the secret image in a scalable manner such that when all the n shares are available, the secret image is reconstructed completely. In this method, the secret image I is first divided into n disjoint sub images $P_1, P_2, \ldots, P_n$. On each sub image P_j $(1 \leq j \leq n)$, Thein and Lin's [26] $(2, 2)$-SIS method is applied to generate sub-share images $Q_j^{(1)}$, $Q_j^{(2)}$. The sub-share images generated for all the sub images are encoded to generate share images $S_1, S_2, \ldots, S_n$. The sub-share images $Q_j^{(1)}$ and $Q_j^{(2)}$ are distributed such that both the sub-shares are not available with a single share i.e. $Q_j^{(1)}$, $Q_j^{(2)} \notin S_j$. A sharing mechanism is devised such that sub-share images $Q_j^{(1)}$ and $Q_j^{(2)}$ are available together only when k or more shares are available, thus ensuring the gradual reconstruction of the secret image. The disadvantage of [28] is that k is limited to a value of 2 and a sharing mechanism is required to ensure that both shares of the same sub image were not available with $(k - 1)$ shares.

The $(2, n)$-SSIS method of Wang et al. [28] was generalized by Yang et al. [34] to the (k, n)-SSIS method. Yang et al. [34] proposed two approaches. Approach 1 uses $\binom{n}{k}$ disjoint sub images while Approach 2 uses $\binom{n}{k-1}$ disjoint sub images of the secret image I. Each sub image P_j generates n sub-share images by applying Thein and Lin's (n, n)-SIS method in both approaches (namely, Approach 1 and Approach 2). Each sub-share image generated for all the sub images is placed into matrix $B_{n,k}$ in Approach 1 and matrix $B_{n,k-1}$ in Approach 2, where every column vector has a Hamming weight k. The sub-share images of the sub images are placed in the matrix at positions with the value "1" and ϕ at positions with value "0" . The share images are further generated by uniting the elements of the i^{th} row. The size of the share images generated using Approach 1 and Approach 2 is $\frac{|I|}{n}$ and $\frac{|I|}{k}$, respectively.

The (k, n)-SSIS methods proposed by Wang et al. [28] and Yang et al.

[34], though they provided scalability, did not provide smooth scalability i.e. the amount of information available for reconstruction is proportional to the number of shares generated. To overcome this disadvantage, Yang et al. [33] proposed the (k, n)-SSIS with smooth scalability method. This (k, n)-SSIS with smooth scalability method is based on the (t, n)-SIS scheme where $(k \leq t \leq n)$. The secret image is divided into $(n - k + 1)$ sub images. The sub image of the size $\frac{k \times |I|}{n}$ is encrypted by using the (k, n)-SIS scheme and the other sub images by $(k+1, n)$-SIS, $(k+2, n)$-SIS, $\ldots$, (n, n)-SIS schemes. Therefore, when t shares are available, t sub images are reconstructed, and as the number of shares increases to n, the complete image is reconstructed. In [33], when k shares are available, the secret image reconstructed is partial, with only some parts of the image being available. With less than k shares, the secret image cannot be reconstructed. Therefore, the disadvantage is that even when all k shares are available the secret image is not reconstructed thus smooth scalability is not provided. By smooth scalability we mean that the secret image has to be reconstructed completely when all k shares are available, but with a low quality. As the number of available shares increases, the quality of the reconstructed image improves, attaining the best quality when all n shares are available. In Table 1 we provide a comparison of the proposed method with the existing SSIS methods.

Note that in our previous work [5], we showed that the existing SIS schemes [26] [3] [4] are semantically insecure and since the existing SSIS methods [28][34][33] are based on [26], they are also not semantically secure. In [5], we also proposed a new semantically secure SIS method, which we use in the SSIS and SSVS methods proposed in this paper.

2.2.4. SSVS

There have been several works in the past based on the encryption of compressed MPEG videos [19] [21] [25]. Since encryption of the complete video is not a feasible solution, [19] [21] focused on selective encryption. In these methods, only selected portions, i.e. only the I frames of the MPEG video, are encrypted. However, these schemes were insufficient as the unencrypted B and P frames revealed the contents of the original video. Furthermore, these methods were also susceptible to known-plaintext and ciphertext-only attacks. Tang [25] proposed encryption of MPEG video by the permutation method. In this method, the DCT coefficients were scanned in a random order rather than in the usual zigzag order. This method was also proved to be insecure by Qiao and Nahrstedt [15], and like the previous methods it is

Table 1: A comparison of the proposed SSIS method with the existing SSIS methods

Method	Gradual reconstruction	Smooth reconstruction	Implemented for	Semantically secure
Wang et al.'s $(2, n)$-SSIS [28]	Yes	No	Images	No
Yang et al.'s (k, n)-SSIS [34]	Yes	No	Images	No
Yang et al.'s (k, n)-SSIS with smooth scalability [33]	Yes	Yes, but only some parts of the image are available with minimum k shares and complete image is reconstructed with all n shares	Images	No
Proposed SSIS method	Yes	Yes, a low quality image/video is available with minimum k shares and the quality increases with the number of shares, providing the highest quality with all n shares	Images, Videos	Yes

vulnerable to known-plaintext and ciphertext-only attacks. The SSS method was extended to MPEG videos by Raju et al. [16]. In this method, based on the number of non-zero AC coefficients in the DCT block, either $(4, 5)$-SSS, $(8, 9)$-SSS or $(12, 13)$-SSS is applied to the DCT coefficients.

The encryption techniques developed for H.264/SVC can be largely classified into two categories. One is Transparent encryption and the other is for Confidentiality. In Transparent encryption techniques [2] [14] [30], a low quality video is made available to users. In confidentiality [29] applications, the video has to be completely unintelligible. There have been many works done in both areas and a few of them are discussed here. The transparent encryption method proposed by Wei et al. [30] is used when the base layer of the video is left clear and the enhancement layers are encrypted. To encrypt the enhancement layers, they proposed block based encryption of the macroblocks in the video in three modes: Intra-MB, Group-MB and 4Group-MB, with increasing levels of security in each mode. Magli et al. [14] proposed multiple algorithms to encrypt the DCT coefficients, motion vectors and redundant slices. The proposed SSIS method is different from the transparent encryption techniques, as the base layer is not left clear in the proposed SSIS method. When k shares are available only the base layer of the video is reconstructed. The other layers of the video are reconstructed when more shares become available.

Secret sharing of H.264/SVC was used in [8]. In this method, the different layers of the video are encrypted using different symmetric keys. In order to reconstruct the video, a set of three keys is required. This mechanism requires the sharing and periodic generations of keys among a group of participants. The set of different keys generated is communicated by the use of the shares. The difference between this work and the proposed SSVS method is that secret sharing is used only for the communication of the keys among the participants in [8], whereas in the proposed SSVS method secret sharing is used to generate shares of the encrypted videos.

3. Proposed Work

In this section, a detailed description of the work proposed in this paper is presented. First, in Section 3.1, the requirements for the proposed methods are discussed. Next, in Section 3.2, we introduce the SIS method that we proposed in our previous work [5]. This method is used in the proposed SSIS and SSVS methods, which are described in Section 3.3 and 3.4, respectively.

Finally, in Section 3.5, the security analysis for the proposed SSIS and SSVS methods is provided.

3.1. Security Types and Requirements

3.1.1. Types of security

The security of the cryptosystems is based on either the computational infeasibility of breaking it (*computational security*), or the theoretical impossibility of breaking it, even using infinite computing power (*information theoretic* or unconditional security). In computationally secure methods, the algorithms are designed on computational hardness assumptions, making the algorithms hard to break practically e.g. security of RSA relies on computational hardness of prime factorization problem. In contrast, the information theoretic security is based on the assumption that the attacker cannot break the algorithm even with unlimited computing power. In other words, the attacker does not have sufficient information to break the cryptosystem and hence it is considered cryptanalytically unbreakable e.g. SSS. Information theoretic security is considered to provide the highest security to the cryptosystems.

The other type of security is the notion of *semantic security.* A cryptosystem is said to be semantically secure if any probabilistic polynomial-time algorithm that is given the ciphertext of a certain message, and the message's length, cannot determine any partial information on the message with probability non-negligibly higher than all other polynomial-time algorithms that only have the access to length of the message. For example, say the attacker has two plaintext messages. Based on the flip of a coin, one of the plaintext messages is encrypted and the corresponding ciphertext is given to the attacker. The attacker cannot determine which of the two plaintext messages was chosen to produce the given ciphertext with a probability not greater than 0.5, which is the success rate of random guessing.

3.1.2. Security requirements

The requirements for the proposed methods are as follows,

1. Having less than k shares, an attacker is unable to recover the original image.
2. Having less than k shares, an attacker does not gain an advantage in recognizing the original image from a group of images that includes the original image.

The first requirement is the threshold property that $(k-1)$ shares reveal no information about the secret. The second requirement is an extra security requirement i.e. the semantic security.

3.2. SIS Method

In our previous work [5], a simple and efficient semantically secure (k, n)-SIS method was proposed by mixing a simple stream cipher with the RSEC method. The implementation of the proposed SIS scheme was provided for both uncompressed and compressed images. For the sake of completeness of this paper, the two phases (share generation and secret reconstruction) of the SIS method proposed in [5] are described as follows.

3.2.1. Share generation phase for SIS

For the share generation phase of the proposed SIS method the following steps are used:

1. Perform encryption on the secret image I using a stream cipher. In the uncompressed domain, the pixel values of the image are added and in the compressed domain, the DCT coefficients are added, with pseudorandom numbers generated using a cryptographically secure pseudorandom number generator to obtain the encrypted image E.
2. Using RSEC, the encrypted image E is partitioned into n fragments $E_1, E_2, \ldots, E_n$.
3. Using SSS scheme, n shares of the seed $(K_1, K_2, \ldots, K_n)$ are generated, which are used to generate random numbers for the encryption step.
4. Each share $s_i = (E_i, K_i)$, $1 \leq i \leq n$, is distributed to the i^{th} user.

Note that the security of the above scheme depends on the security of the pseudorandom function used in Step 1 and the use of stream cipher along with RSEC scheme offers semantic security to the secret image.

3.2.2. Secret reconstruction phase for SIS

The secret reconstruction phase of the proposed SIS method is as follows:

1. Collect any k shares from the participants.
2. Using the RSEC, reconstruct E from the shares collected, $E_i, i = 1, 2, \ldots, k$.
3. Using Lagrange interpolation, recover the seed value K out of $K_i, i = 1, 2, \ldots, k$ shares, using Eq 2.

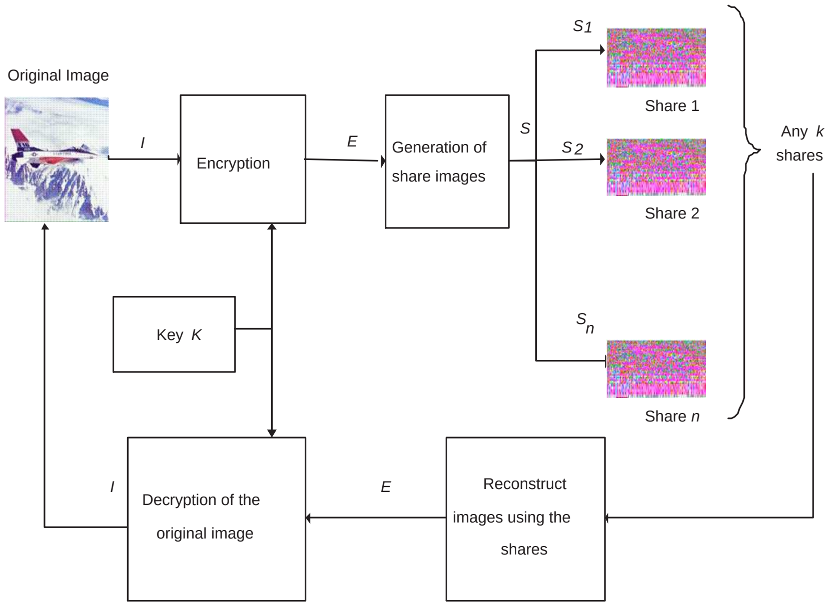

Original Image
I
Encryption
E
Generation of share images
S
S1
Share 1
S2
Share 2
Sn
Share n
Any k shares
Key K
I
Decryption of the original image
E
Reconstruct images using the shares

Figure 1: Diagrammatic representation of the proposed SSIS method.

4. Decrypt E to recover the secret image I by first generating random numbers using the seed constructed in the previous step, and then subtracting the random numbers from the reconstructed encrypted image E.

3.3. SSIS Method

Figure 1 gives the diagrammatic representation of the proposed SSIS method. The two phases (share generation and secret reconstruction) of this method are described as follows. Note that, in rest of this paper, (k, n)-SIS refers to our previous work [5] (described in Section 3.2). As mentioned earlier, we use it in the proposed SSIS and SSVS methods.

3.3.1. Share generation phase for SSIS

For the generation of image shares, the following steps are used:

1. Perform encryption of the secret image I to generate the encrypted image E. In the compressed domain, DCT coefficients are added with cryptographically secure pseudo random numbers, generated using the seed K.

2. The encrypted image E, is fragmented into $(n-k+1)$ parts E_1, E_2, $\dots$, $E_{(n-k+1)}$ such that,
   E_1 contains the DC, AC_1, AC_2 coefficients of the encrypted image E
   E_2 contains AC_3, $\dots$, AC_u coefficients of the encrypted image E
   E_3 contains AC_{u+1}, $\dots$, AC_v coefficients of the encrypted image E
   $\vdots$
   E_{n-k+1} contains $\dots$, AC_{62}, AC_{63} coefficients of the encrypted image E.
   where u, v are intermediate indices of the DCT coefficients. The DCT coefficients can be fragmented into different scans based on the users requirement.
3. Now, for each partition, apply (k,n)-SIS, where $2 \leq k \leq n$. Increment the value of k for every partition.
   Apply (k,n)-SIS to E_1 to generate shares E_{11}, E_{12}, $\dots$, E_{1n}
   Apply $(k+1,n)$-SIS to E_2 to generate shares E_{21}, E_{22}, $\dots$, E_{2n}
   $\vdots$
   Apply (n,n)-SIS to $E_{(n-k+1)}$ to generate shares $E_{(n-k+1)1}$, $E_{(n-k+1)2}$, $\dots$, $E_{(n-k+1)n}$
   For share generation, each partition E_i $(2 \leq i \leq n)$ is divided into k segments and the polynomial is constructed using the coefficient values from each of the k sections, i.e. the values for a_0, a_1, $\dots$, a_{k-1} in Eq 1 are the DC and AC coefficient values from each of the k partitions.
4. Encode the shares generated from each partition, which are distributed to the users.
   $S_1 = E_{11} \cup E_{21} \cup \cdots \cup E_{(n-k+1)1}$
   $S_2 = E_{12} \cup E_{22} \cup \cdots \cup E_{(n-k+1)2}$
   $\vdots$
   $S_n = E_{1n} \cup E_{2n} \cup \cdots \cup E_{(n-k+1)n}$.
5. Using SSS scheme, generate n shares of the seed K which is used to generate the pseudo random numbers i.e. $K_1, K_2, \dots, K_n$.
6. Each share $s_i = (S_i, K_i)$, $1 \leq i \leq n$ is distributed to the user.

3.3.2. Secret reconstruction phase for SSIS

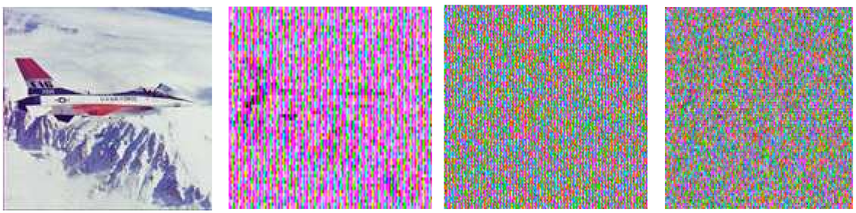
The following steps are used to reconstruct the secret image in a scalable manner in the proposed SSIS method:

1. Collect share images s_i, where $k \leq i \leq n$.
2. Reconstruct the encrypted image E from the shares collected $E_1, E_2, \dots, E_i$, where $k \leq i \leq n$.

3. Reconstruct the seed K by Lagrange interpolation given by Eq 2 using the shares $K_1, K_2, \ldots, K_i$, where $k \leq i \leq n$.
4. Decrypt the encrypted image E by subtracting the DCT coefficient values from the pseudo random numbers generated using the seed K, to obtain I.

3.3.3. An illustrative example for the SSIS method

In this section, an example of the proposed SSIS method is provided. (k, n)-SSIS method is applied, where $k = 3, n = 5$ on the secret image Jet. A stream cipher is used to encrypt the secret image by adding pseudo random numbers to the DCT coefficients of the secret image. Then the encrypted image is divided, as shown in Figure 2, into 3 sub images such that the first encrypted sub image E_1, as shown in Figure 2(b), contains the DC, AC_1 and AC_2 coefficients of the encrypted image E, the second sub image E_2, as shown in Figure 2(c), contains the AC coefficients 3 to 10 of the encrypted image E and the third sub image E_3, as shown in Figure 2(d), contains the remaining AC coefficients, 11 to 63 of the encrypted image E.



(a) Original image I

(b) Encrypted image E_1 with DC, AC_1, AC_2

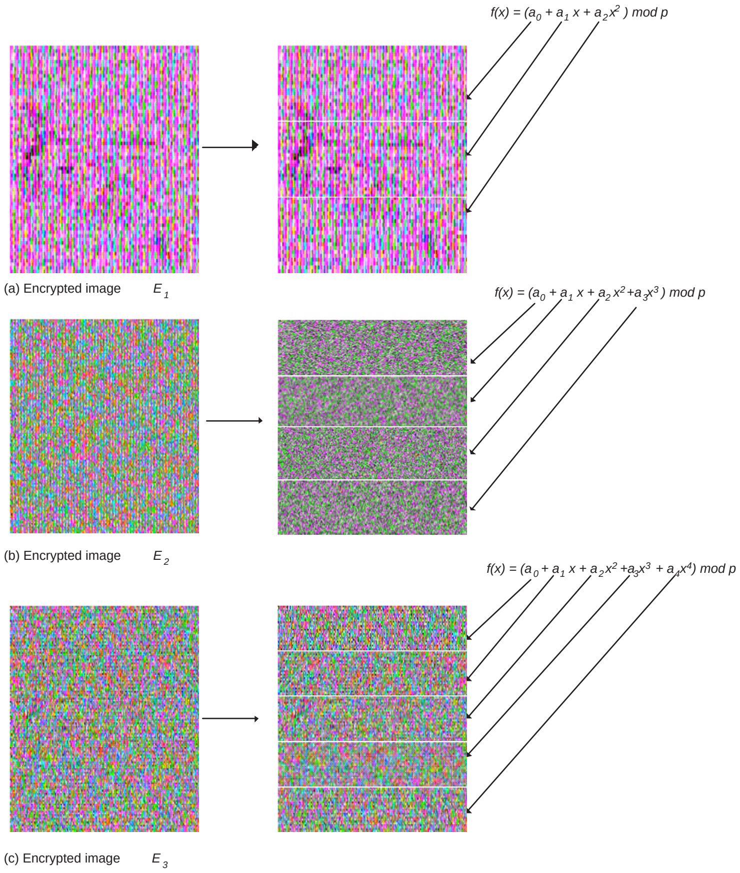
(c) Encrypted image E_2 with AC_3 to AC_{10}

(d) Encrypted image E_3 with AC_{11} to AC_{63}

Figure 2: Partitions of the secret image Jet

Next, (k, n)-SSIS is applied on the first sub image E_1 with $k = 3, n = 5$, on the second sub image E_2 with $k = 4, n = 5$ and on the third sub image E_3 with $k = 5, n = 5$. The polynomial for secret sharing is constructed by dividing each sub image into k sections and using the coefficient values from each of the k sections as shown in Figure 3. The share images generated are distributed to the n users along with the share of the seed value.

When $k(= 3)$ shares are available, the image reconstructed is of low quality. As the number of shares available increases, the quality of the reconstructed image improves. The image is reconstructed fully when all n shares are available.


f(x) = (a0 + a1 x + a2x2) mod p
f(x) = (a0 + a1 x + a2x2+a3x3) mod p
f(x) = (a0 + a1 x + a2x2+a3x3 + a4x4) mod p

(a) Encrypted image E_1

(b) Encrypted image E_2

(c) Encrypted image E_3

Figure 3: Use of coefficient values for the polynomials in SSIS method

3.4. SSVS Method

In this section, the share generation and secret reconstruction phases for the videos are discussed.

3.4.1. Share generation phase for SSVS

For the share generation of the video, the steps given below are used:

1. Determine the value of n, l, where n is the total number of shares to be created and l is the total number of layers in the video V, which includes base and enhancement layers.
2. Decode the secret video V to generate frames V_1, V_2, $\ldots V_f$, where f is the total number of frames in V.

3. Encrypt the secret video V using pseudo random numbers, generated using the seed K, to obtain the encrypted video E, with frame $E_1, E_2, \ldots, E_f$.
4. Calculate the value of k, which is given by $k = \lfloor \frac{n+1}{l} \rfloor$.
5. If the frame E_j, where $(1 \leq j \leq f)$ belongs to the base layer or enhancement layer 1, create $(k-1)$ partitions of the frame. If it belongs to enhancement layer 2 or enhancement layer 3, create k partitions. Frames are divided such that each partition gets the row values at a displacement of k.
   E_{11}, E_{12}, $\ldots$, $E_{1(k-1)}$ for E_1, which is the base layer frame
   E_{21}, E_{22}, $\ldots$, $E_{2(k-1)}$ for E_2, which is enhancement layer 1 frame
   E_{31}, E_{32}, $\ldots$, E_{3k} for E_3, which is enhancement layer 2 frame
   $\vdots$
   E_{f1}, E_{f2}, $\ldots$, $E_{f(k)}$ for E_f, which can be enhancement layer 2 or enhancement layer 3 frame.
6. Apply SIS to generate the shares of each of these frames. If the frame belongs to the base layer i.e for frame E_1
   E_{11} apply $(2, n)$-SIS to generate shares E_{11}^1, E_{11}^2, $\ldots$, E_{11}^n
   E_{12} apply $(3, n)$-SIS to generate shares E_{12}^1, E_{12}^2, $\ldots$, E_{12}^n
   $E_{1(k-1)}$ apply (k, n)-SIS to generate shares $E_{1(k-1)}^1$, $E_{1(k-1)}^2$, $\ldots$, $E_{1(k-1)}^n$.
7. Apply SIS for the all the enhancement layers i.e.
   For enhancement layer 1, apply $(k+1, n)$-SIS, $(k+2, n)$-SIS, $\ldots$, $(2k-1, n)$-SIS for all the partitions of the frame
   For enhancement layer 2, apply $(2k, n)$-SIS, $(2k+1, n)$-SIS, $\ldots$, $(3k-1, n)$-SIS for all the partitions of the frame
   $\vdots$
   For enhancement layer l, apply (lk, n)-SIS, $(lk+1, n)$-SIS, $\ldots$, $(lk+k-1, n)$-SIS for all the partitions of the frame.
   To generate the shares, luma components of the video are used and the chroma components are left clear. Shares are generated using Eq 1, where a_0, a_1, $\ldots$, a_{k-1} are the luma values from each of the k sections of the frame.
8. Encode the shares generated from all the layers to generate share videos S_1, S_2, $\ldots$, S_n, where
   $S_1 = E_{11}^1 \cup E_{12}^1 \cup \cdots \cup E_{21}^1 \cup \cdots \cup E_{f1}^1 \cup \cdots \cup E_{f(k-1)}^1$.
   $S_2 = E_{11}^2 \cup E_{12}^2 \cup \cdots \cup E_{21}^2 \cup \cdots \cup E_{f1}^2 \cup \cdots \cup E_{f(k-1)}^2$.

$\vdots$

$S_n = E^n_{11} \cup E^n_{12} \cup \cdots \cup E^n_{21} \cup \cdots \cup E^n_{f1} \cup \cdots \cup E^n_{f(k-1)}$.

9. Generate shares of the seed K using (k, n)-SSS scheme i.e $K_1, K_2, \ldots, K_n$.
10. Distribute the share $s_i = (S_i, K_i)$ to the users, where $1 \leq i \leq n$.

3.4.2. Secret reconstruction phase for SSVS

In the secret reconstruction phase, as more shares become available, the base and enhancement layers are reconstructed. Any k shares are needed to reconstruct the base layer, which gives us the video with the frames of the base layer. As the number of shares increases, the enhancement layers are reconstructed, which give more number of frames of the video.

For scalable reconstruction of the secret video, the following steps are used:

1. Collect the share videos from the users, say s_i, where $1 \leq i \leq n$.
2. Using the shares collected, generate the seed value K and generate the random numbers.
3. Decode the share videos to generate the frames,
   SF_{11}, SF_{12}, $\ldots$, SF_{1f} for share video S_1, where f is the total number of frames in the video
   SF_{21}, SF_{22}, $\ldots$, SF_{2f} for share video S_2 and so on.
4. Using the share frames, apply the Lagrange interpolation method as given by Eq 2 on the share frames to generate the encrypted video frames $E_1, E_2, \ldots, E_n$. If any 2 shares are available, solve for the values x_0 and x_1. If any 3 shares are available, solve for the values of x_0, x_1, x_2 and so on. The luma values of the share frames are used for $f_j(x)$ as per Eq 2.
5. Encode the frames to generate the encrypted secret video E. When k $(k \leq n)$ shares are available the base layer which was partitioned into $E_{1(k-1)}$ parts, can be reconstructed completely. With $(2k - 1)$ shares, the enhancement layer 1 is reconstructed and with all n shares available, all the partitions of the enhancement layers can be reconstructed. Hence, all the layers of the secret video are reconstructed.
6. Decrypt the encrypted video E using the random numbers to generate the secret video V.

3.4.3. An illustrative example for the SSVS method

In this section, an illustrative example for the proposed SSVS method is provided. For example, consider the value of $l = 4$, with the base layer and 3 enhancement layers, and the number of users $n = 11$, so the value of k is 3. Consider the sample video foreman.yuv which has 300 frames with an intra period of 8 frames as shown in the Figure 4. In this figure, the first 4 frames of the video are shown. Frame 0 belongs to the temporal layer 0, which is the base layer frame. Frame 4 belongs to temporal layer 1, which is the enhancement layer 1 frame. Frame 2 belongs to temporal layer 2, which is the enhancement layer 2 frame and frame 1 belongs to temporal layer 3, which is the enhancement layer 3 frame. Next, the base layer frame is divided into $(k - 1)$ partitions, i.e 2 parts, then $(2, n)$-SIS and $(3, n)$-SIS is applied on the two partitions of the base layer. Next the enhancement layer 1 is divided into 2 parts and $(4, n)$-SIS, $(5, n)$-SIS is applied to the two parts. Enhancement layer 2 is divided into 3 parts and $(6, n)$-SIS, $(7, n)$-SIS, $(8, n)$-SIS is applied on each of the parts. For enhancement layer 3, $(9, n)$-SIS, $(10, n)$-SIS, $(11, n)$-SIS is applied on the three parts, respectively. Thus during reconstruction, if k shares are available (in this case, 3 shares), then the base layer of the video is reconstructed. With 5 shares, the base layer and enhancement layer 1 are reconstructed and with 8 shares, the base layer plus two enhancement layers are reconstructed. With 11 shares the complete video is reconstructed.

3.5. Security Analysis

3.5.1. Computational security

The proposed methods are said to be computationally secure, if with $(k - 1)$ or lesser shares it is infeasible for an attacker to obtain any information about the secret. For images, consider the size of the image to be $w \times h$. For an uncompressed image, the pixel value of the image can have 251 values. For performing encryption, which is adding pseudo random numbers generated in the finite field, consider that the random numbers generated are in the range $[0 - 251]$, the number of operations to be performed is $251^{(2\times w\times h)}$. For compressed images, encryption on all the nonzero coefficients (nzc) of the DCT block is performed. An image with pixel values in the range $[0 - 255]$ will produce DCT coefficients in the range $[-1024 \text{ to} + 1023]$. So for compressed images, the number of operations to be performed is $(2048^{(nzc\times \frac{w\times h}{8\times 8})} + 251^{(nzc\times \frac{w\times h}{8\times 8})})$. For videos, a luma component

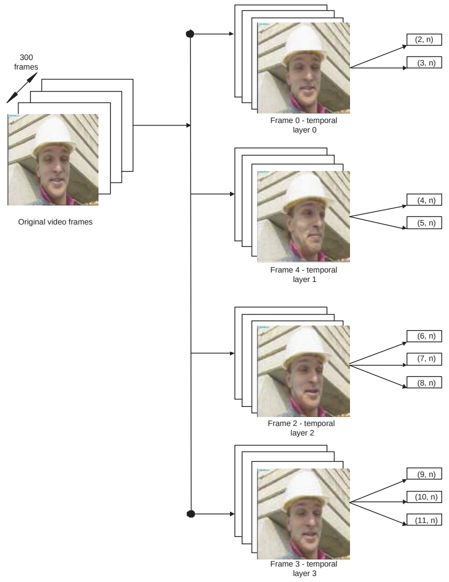

300 frames
Original video frames
(2, n)
(3, n)
Frame 0 - temporal layer 0
(4, n)
(5, n)
Frame 4 - temporal layer 1
(6, n)
(7, n)
(8, n)
Frame 2 - temporal layer 2
(9, n)
(10, n)
(11, n)
Frame 3 - temporal layer 3

Figure 4: Use of coefficients value for the polynomials in SSVS method

can have 220 values. So for a video $w \times h$, with f being the total number of frames in the video, an attacker has to perform $(220^{(w\times h\times f)} + 251^{(w\times h\times f)})$ operations. Therefore, the proposed methods are computationally secure.

3.5.2. Semantic security

To show that the proposed methods are semantically secure, assume that the attacker has access to $(k-1)$ shares, and wishes to know whether a given image J is the secret image. Below, it is explained informally why the attacker is computationally unable to know the secret despite possessing $(k-1)$ shares. Assume that the attacker possesses the encrypted image or video E, as this does not put the attacker in a weaker position. Note that, having E, one can generate all the shares. Therefore, an attacker who possesses $(k-1)$ shares is not more powerful than one who possesses E. If the generated random numbers were truly random, the attacker could not gain any information by possessing E. In essence, for any image J we have $Pr(J = I|E) = Pr(J = I)$, where I is the secret image or video. Since it is computationally difficult to distinguish between a sequence of numbers generated by the random number generator and a sequence of true random numbers, the attacker is computationally unable to gain any information having only E.

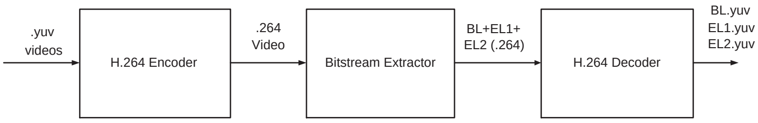


Figure 5: Structure of JSVM codec

4. Results and Analysis

4.1. Implementation Details, Data Set and Performance Parameters

The experiments on images and videos were performed and recorded on an i7 processor, 2.6Ghz Intel machine with 6GB RAM. The experiments performed on both images and videos were developed using C programming language, Microsoft Visual Studio IDE and OpenCV software. Libjpeg library has been used to perform the experiments on JPEG images and JSVM library has been used for the videos.

JSVM is the most popularly used H.264/SVC codec for videos. The block diagram of the JSVM encoder is provided in Figure 5. In the JSVM encoder, YUV format videos are input and converted to a scalable video format (.264 video). From the generated .264 video file, videos of both AVC and SVC format can be decoded, where SVC video has multiple enhancement layers of varying frame rates. The .264 video serves as the input to the bitstream extractor. The bitstream extractor of the JSVM generates the base and enhancement layers of the video in the .264 format. These .264 videos are given to the decoder to generate videos in the *.yuv* format. Individual layers of the video can also be extracted based on the spatial or temporal characteristics, or based on the bit rates of the video. By using the decoder, video can be decoded into frames to perform necessary operations and generate share videos in the *yuv* format.

The data set used for the proposed SSIS method is given in Table 2. The data set used for the proposed SSVS method is given in Table 3. In Table 4, the total number of frames in each video and the number of frames in each layer, i.e. the base layer and all the enhancement layers of the video, are listed.

Performance analyses of the proposed SSIS and SSVS methods are provided by comparing the computation times of the proposed methods with those of the previous methods in both uncompressed and compressed domains. The comparison is also performed analytically by listing the number

Table 2: Dataset used for the evaluation of SSIS

Image name	Resolution in pixels	Size in KB
Lena	225×225	9
Lena gray	225×225	9
Baboon	500×480	137
Baboon gray	225×225	11
Jet	512×512	72
Barbara	720×576	137
Canyon	1200×765	1,412
Nature	2560×1024	808
Navalship	3008×2000	757
Birds	4000×3000	8,757

Table 3: Dataset used for the evaluation of SSVS

Video name	Resolution in pixels	Frame rate in fps	Size in KB
Bus	176×144, 352×288	15, 30	5,569, 22,275
Foreman	176×144, 352×288	15, 30	5,569, 44,550
Football	176×144, 352×288	15, 30	9,653, 38,610

Table 4: Frame details of the videos used in SSVS

Video name	Total frames count	Base layer frame count	Enhancement layer 1 frame count	Enhancement layer 2 frame count	Enhancement layer 3 frame count
Bus	150	19	18	38	75
Football	260	33	32	65	130
Foreman	300	38	37	75	150

of mathematical operations performed in each of the proposed and previous methods.

Peak Signal-to-Noise Ratio (PSNR) values are used to depict the quality of the reconstructed images and videos. Higher values of PSNR indicates that there is less error and the quality of the reconstruction is higher. PSNR is calculated as shown in Eq 4.

$$PSNR = 10 \times \log_{10}(\frac{MAX_I^2}{MSE}) \tag{4}$$

where MAX_I^2 is the maximum pixel value of the image I, and MSE (Mean Square Error) is calculated as,

$$MSE = \frac{1}{w \times h} \sum_{i=0}^{w-1} \sum_{j=0}^{h-1} [I(i,j) - I'(i,j)]^2 \tag{5}$$

where w is the width of the image, h is the height of the image, $I(i,j)$ is the pixel value of the original image I and $I'(i,j)$ is the pixel value of the reconstructed image I'.

The other parameter used to measure the quality of the images and videos is Relative Quality Index (RQI). RQI is used to measure the change in the quality of the reconstructed images w.r.t the image reconstructed with k shares. RQI values for the reconstructed images increase with the number of shares. RQI is calculated as:

$$RQI = \frac{\delta_t - \delta_k}{\delta_n - \delta_k} \tag{6}$$

where δ_t is the PSNR value of the image reconstructed with t shares ($k \leq t \leq n$), δ_k is the PSNR of the image reconstructed with k shares and δ_n is the PSNR of the image reconstructed with n shares. The value of RQI varies between 0 and 1, i.e. when $\delta_t = \delta_k$, $RQI = 0$ and when $\delta_t = \delta_n$, $RQI = 1$. The RQI for videos are the PSNR values of the videos reconstructed with k, t and n shares respectively.

4.2. Performance Analyses

4.2.1. Results for SSIS

To evaluate the proposed SSIS method, $(3,5)$-SSIS is applied on the images of the dataset given in Table 2. In Table 5 and Table 6 the share creation

Table 5: Share creation time of the images using SSIS in ms

Image name	Yang et al.'s Approach 2 [33]	Yang et al.'s smooth scalability method [34]	Proposed SSIS method
Lena	54	56	34
Baboon	181	252	223
Jet	153	228	138
Barbara	194	364	247
Canyon	1,723	1,546	1,270
Nature	1,560	1,487	1,290
Navalship	3,045	2,870	2,156
Birds	7,500	2,865	2,194

and secret reconstruction times for the proposed SSIS method are provided. Also, $(3, 5)$-SSIS is performed on Yang et al.'s [33] Approach 2 and on Yang et al.'s [34] smooth scalability method. The execution times of these previous SSIS methods are compared to that of the proposed SSIS method. Table 5 gives the execution time for creating 5 shares and Table 6 gives the secret reconstruction time using all 5 shares generated. It can be seen from the table that the proposed SSIS method has less processing time when compared to the previous methods. In Table 7, we give the reconstruction times of the proposed method with varying values of the shares.

Figure 6 presents the five share images generated from the secret image Jet. The share images generated are noisy and do not reveal any information of the secret image. Hence, the proposed SSIS method is perceptually secure and the size of the share images is $1/3$ $(1/k)$ of the size of the secret image. In Figure 7, the reconstructed images with varying values of k are shown. Since $(3, 5)$-SSIS has been used, 3 shares are required to reconstruct the image with the lowest quality, as this image contains only the DC, AC_1 and AC_2 coefficients, as shown in Figure 7(b). The quality achieved is much better with 4 shares, as all the coefficients up to AC_{11} are available. It can be seen that Figure 7(c) lacks sharpness because of the remaining AC coefficients. When all 5 shares are available, the image is reconstructed with the best quality, as shown in Figure 7(d).

Figure 8 shows the reconstructed images using Yang et al.'s [33] method, Approach 2. Figure 9 shows the reconstructed images from Yang et al.'s [34]

Table 6: Secret reconstruction time of the images using SSIS in ms

Image name	Yang et al.'s Approach 2 [33]	Yang et al.'s smooth scalability method [34]	Proposed SSIS method
Lena	30	25	20
Baboon	63	59	54
Jet	76	67	52
Barbara	123	105	90
Canyon	1,843	1,679	683
Nature	1,623	1,505	524
Navalship	3,423	3,212	709
Birds	8,249	3,423	2,845

Table 7: Secret reconstruction time with increasing number of shares using SSIS in ms

Image name	3 shares	4 shares	5 shares
Lena	13	17	20
Baboon	48	51	54
Jet	43	47	52
Barbara	83	86	90
Canyon	597	625	683
Nature	412	469	524
Navalship	680	691	709
Birds	1,600	2,004	2,845

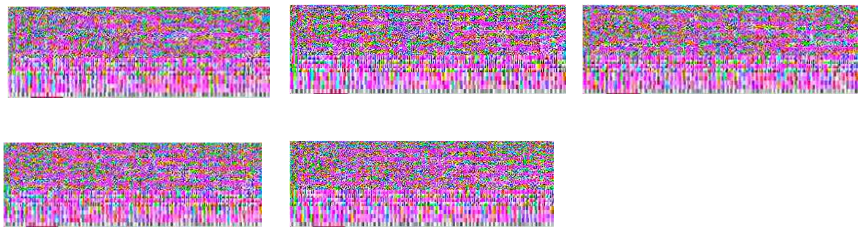
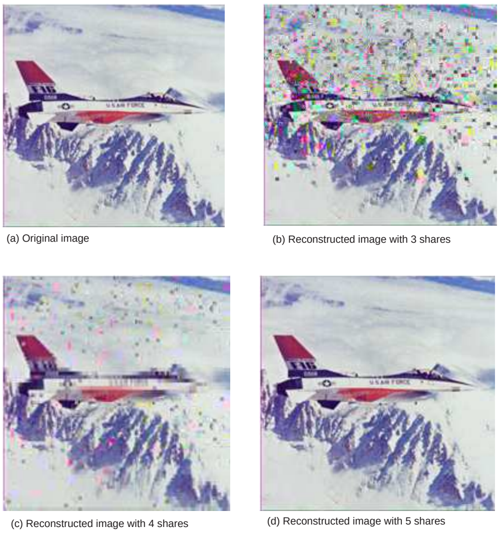


Figure 6: Share images of the secret image Jet


(a) Original image

(b) Reconstructed image with 3 shares

(c) Reconstructed image with 4 shares

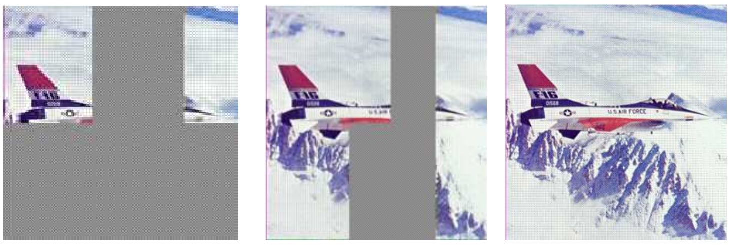
(d) Reconstructed image with 5 shares

Figure 7: Reconstructed images using varying values of k

smooth scalability method with $(3, 5)$-SSIS. It can be seen that the proposed SSIS method achieves gradual reconstruction with smooth scalability when compared to Yang et al.'s Approach 2 [33] and Yang et al.'s [34] smooth scalability method. The other advantage that the proposed method has over [33] and [34] is that as the number of participants increases, the number of sub images to be generated in these methods increases proportionally. If $(3, 10)$-SSIS is performed, the minimum number of sub images generated is 45, using the methods from [33] and [34], so the distribution of the information is not uniform, i.e. the important part of the secret image that has to be concealed is not received uniformly by all the participants. Conversely, in the proposed SSIS method, every participant gets the same amount of important information.

As stated, PSNR is used as a measurement to depict the increase in the quality of the images with the increase in the value of k. Table 8 gives the PSNR values of the reconstructed images with 3, 4 and 5 shares. It can be seen that with the increase in the number of shares available, the PSNR values increase and reach the maximum when all n shares are available.

In Figure 10(a) and Figure 11(a), the graphs generated for RQI values with varying t, $k \leq t \leq n$, $k = 3$ and $n = 10$ for the images Lena and Jet are provided. Figure 10(b) and Figure 11(b) show the corresponding

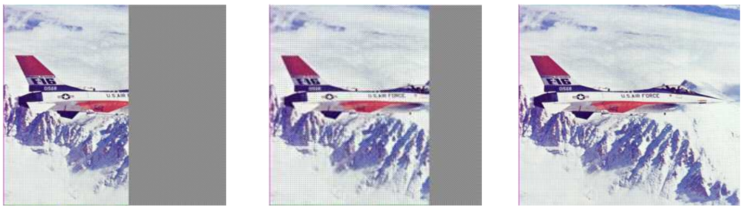


(a) Reconstructed image with 3 shares

(b) Reconstructed image with 4 shares

(c) Reconstructed image with 5 shares

Figure 8: Reconstructed images of Yang et al.'s Approach 2 [33]



(a) Reconstructed image with 3 shares

(b) Reconstructed image with 4 shares

(c) Reconstructed image with 5 shares

Figure 9: Reconstructed images of Yang et al.'s smooth scalability method [34]

Table 8: PSNR values for the images reconstructed using SSIS

Image name	3 shares	4 shares	5 shares
Lena	27.35	33.48	99.98
Baboon	30.38	35.70	99.97
Jet	29.78	34.82	99.99
Barbara	27.45	34.23	99.97
Canyon	26.88	33.71	99.98
Nature	28.76	34.99	99.99
Navalship	29.10	34.87	99.98
Birds	29.37	33.23	99.97

reconstructed images[2]. Here, a larger value of t is used in order to show the linear increase in the quality of the images with the increase in the number of shares. Also, the PSNR values of the images based on the value of t has been provided with the reconstructed images. It can be seen that RQI varies linearly between the values of 0 and 1. Hence, having a higher value of t means that the distribution of the coefficients is more uniform and hence the variation in the quality of the images is linear.

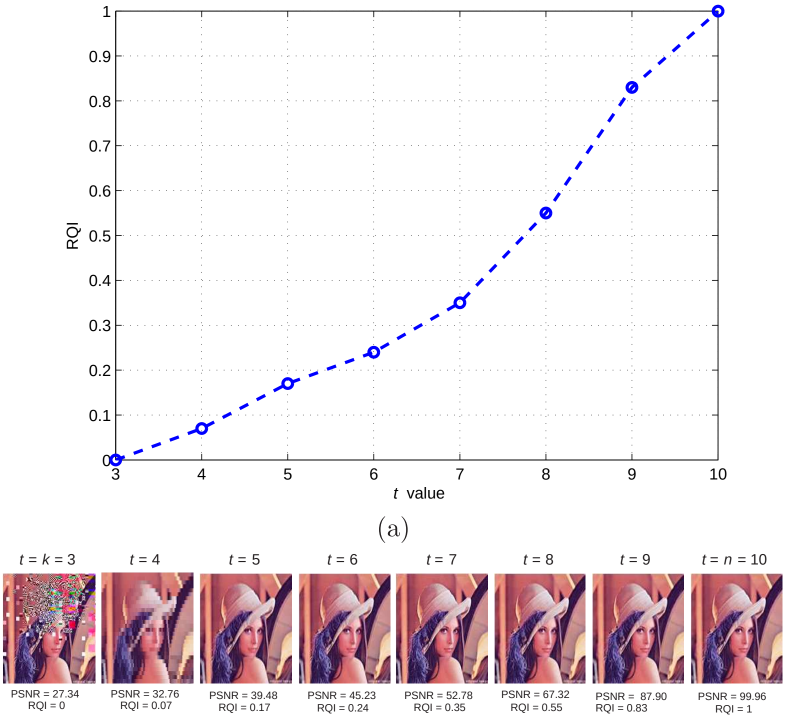
PSNR is used to depict the increase in the quality of the images. A comparison of PSNR measurement with previous SSIS methods is not performed, as firstly the complete image is not reconstructed until all n shares are available, secondly the parts of the image that has been reconstructed has the highest quality. Hence a PSNR measurement has not been provided.

In Figure 12, a graph of the reconstruction times for the image Jet is provided. It can be seen from the graph that as the value of n increases, the time taken to reconstruct the original image increases. This can be the limitation of the proposed SSIS method, when the value of n is high. The processing times can be much higher with images of larger size, which can further increase the reconstruction times.

Considering the number of finite field operations for an image with $w \times h$ pixels, Yang et al.'s [33] method will have to perform $(k-1) \times \frac{w \times h}{\binom{n}{k}}$ addition operations and $k \times \frac{w \times h}{\binom{n}{k}}$ multiplication operations for one share of one sub image, since the image is partitioned into $\binom{n}{k}$ sub images. Yang et al.'s [34] method on smooth scalability performs $(k^2+k+1) \times \frac{w \times h}{n}$ addition and $(k^2+5k+1) \times \frac{w \times h}{n}$ multiplication operations. The proposed SSIS method has to perform $((2k^2+k+1) \times \frac{w \times h}{8 \times 8}$ addition operations and $(k^2+5k+1) \times \frac{w \times h}{8 \times 8}$ multiplication operations, similar to [34]. Though the number of addition operations performed in the proposed SSIS method is higher when compared to the previous SSIS methods, the processing times are less as we perform secret sharing only on the non-zero coefficients of the DCT block.

From the above experiments, it can be deduced that the proposed SSIS method is computationally faster when compared to the previous methods. It is also perceptually secure and provides reconstruction with smooth scalability, as indicated by the PSNR and RQI values. Further, the size of the shares is also reduced to $1/k$ of the size of the secret image.

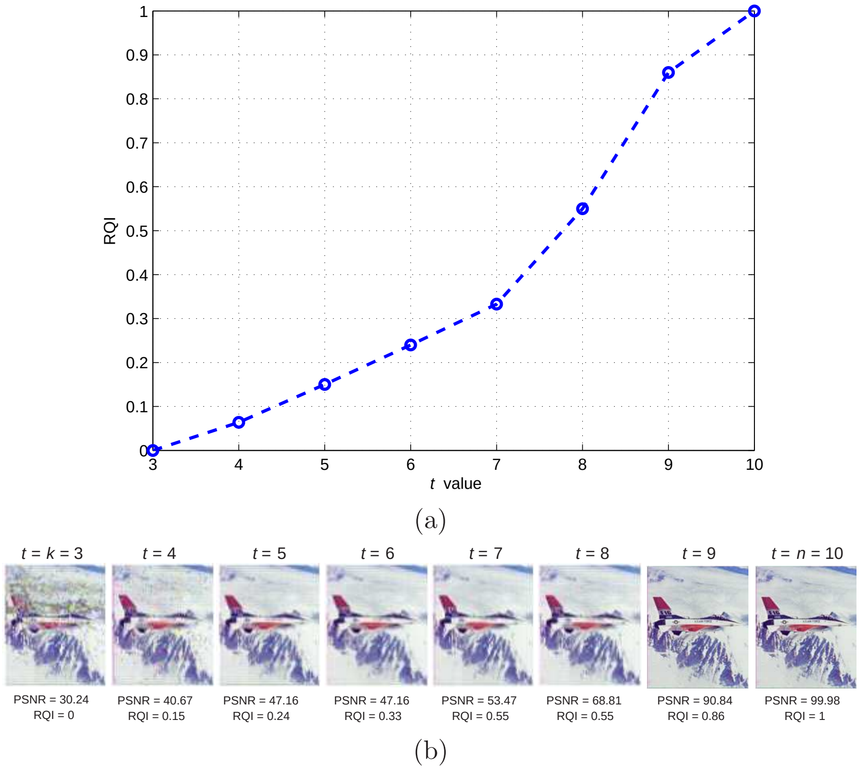
[2] A demo video of the SSIS results can be found at: http://youtu.be/MXlWO1pYdSI



(a)

(b)

Figure 10: RQI of the reconstructed image Lena using SSIS for varying t, $k \leq t \leq n$, $k = 3, n = 10$



(a)

(b)

Figure 11: RQI of the reconstructed image Jet using SSIS for varying t, $k \leq t \leq n$, $k = 3, n = 10$

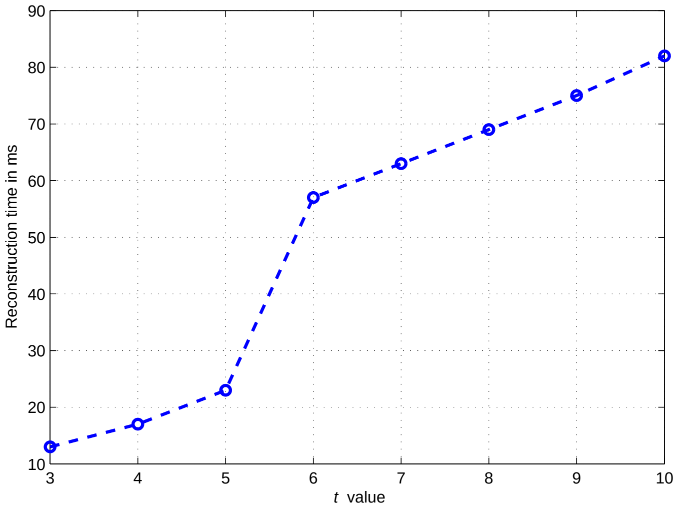


Figure 12: Variation of execution time with t values, $k \leq t \leq n$, $k = 3, n = 10$

Table 9: Single share creation and reconstruction times for complete video using SSVS in s

Video name	Share generation time	Secret reconstruction time
Bus	45	115
Football	58	128
Foreman	67	134

Table 10: Comparison of share creation times per frame in ms

Frame number	Yang et al.'s Approach 2 [33]	Yang et al.'s smooth scalability method [34]	Proposed SSVS method
frame 0	96	97	108
frame 1	87	92	110
frame 2	90	94	107
frame 4	92	96	111

4.2.2. Results for SSVS

For the purpose of the experiments, consider the example discussed in Section 3.4.3 (Chapter 3). The $(3, 11)$-SSVS is applied on the videos from the dataset given in Table 3. In Table 9, the time taken by the proposed SSVS method for single share creation and secret video reconstruction for the complete videos, using 3 shares is presented. The proposed SSVS method is compared to the previous SSIS methods, as there has previously been no work based on SSVS. For this, frames from the video foreman.yuv are extracted and Yang et al's [33] Approach 2, Yang et al.'s [34] smooth scalability and the proposed SSVS method are applied on the frames. Table 10 gives the comparison of share creation times using the three methods. It can be seen that the proposed method has slightly higher processing times when compared to Yang et al.'s [33] Approach 2 and Yang et al.'s [34] smooth scalability method.

The proposed SSVS method is also compared with Wei et al.'s [30] scalable encryption method. In Wei et al.'s [30] method, the enhancement layers of the H.264/SVC video are encrypted. In Table 11, the computation time taken by Wei et al.'s method for creating the encrypted video and the time taken to create a share by the proposed method are provided. The proposed

Table 11: Comparison of computation time in s

Video name	Wei et al. [30]'s method	Proposed SSVS method
Bus	23	45
Football	99	128
Foreman	107	134

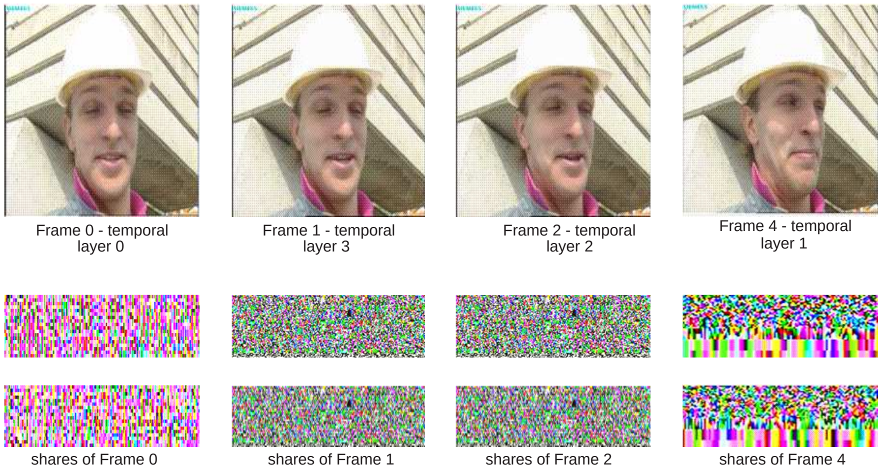

Frame 0 - temporal layer 0
Frame 1 - temporal layer 3
Frame 2 - temporal layer 2
Frame 4 - temporal layer 1
shares of Frame 0
shares of Frame 1
shares of Frame 2
shares of Frame 4

Figure 13: Frame and their shares

method has slightly higher processing time when compared to Wei et al.'s method, but unlike the proposed SSVS method, in Wei et al.'s method the base layer of the video is left in clear.

In Figure 13, the first four frames of the video foreman.yuv and the corresponding shares generated are shown. On frame 0, which is the base layer frame, $(2, n)$-SIS and $(3, n)$-SIS are applied. On frame 4, which belongs to enhancement layer 1, $(4, n)$-SIS and $(5, n)$-SIS are applied. For frame 2, which belongs to enhancement layer 2, $(6, n)$-SIS, $(7, n)$-SIS and $(8, n)$-SIS are applied. On frame 1, which is enhancement layer 3 frame, $(9, n)$-SIS, $(10, n)$-SIS and $(11, n)$-SIS is applied. The corresponding shares are presented in Figure 13. For H.264 video, encryption is performed on the luma components and not on the transform coefficients, so as to preserve the compression efficiency of the H.264 videos.

Figure 14 presents the sequence of frames of the foreman.yuv video. In

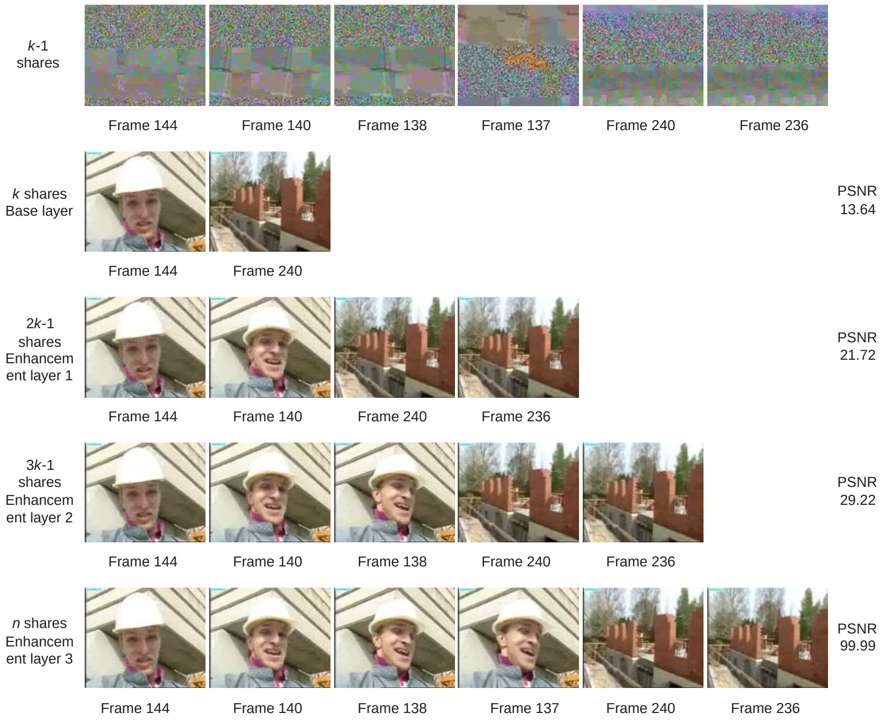


Figure 14: Reconstructed frames of the video with varying t

the first row, frames belonging to $(k-1)$ shares are shown. With k ($k=3$ shares), only the base layer frames are reconstructed which are shown in the second row. Next, with $2k-1$ shares ($k=5$), the frames of enhancement layer 1 are reconstructed, as shown in the third row. When $3k-1$ ($k=8$) shares are available, frames belonging to enhancement layer 2 are reconstructed. Finally, when all n shares are available, all the frames of the video are reconstructed [3].

Table 12 presents the PSNR values of the videos generated with k shares, $(2k-1)$ shares, $(3k-1)$ shares and n shares. Figure 15(a) gives the graph of RQI generated for $(3,11)$-SSVS on the video Bus.yuv and Figure 15(b) shows the corresponding video frames with varying value of t. It can be seen that RQI values of the videos increase linearly with the increase in the number of shares available, having the highest value when n shares are available. Figure

[3] A demo video of the SSVS results can be found at: http://youtu.be/w8uh6FzVIno

Table 12: PSNR values of the videos

Video name	k shares	$(2k-1)$ shares	$(3k-1)$ shares	n shares
Bus	12.36	25.62	32.56	99.98
Football	15.29	23.37	31.19	99.97
Foreman	13.64	21.72	29.22	99.99

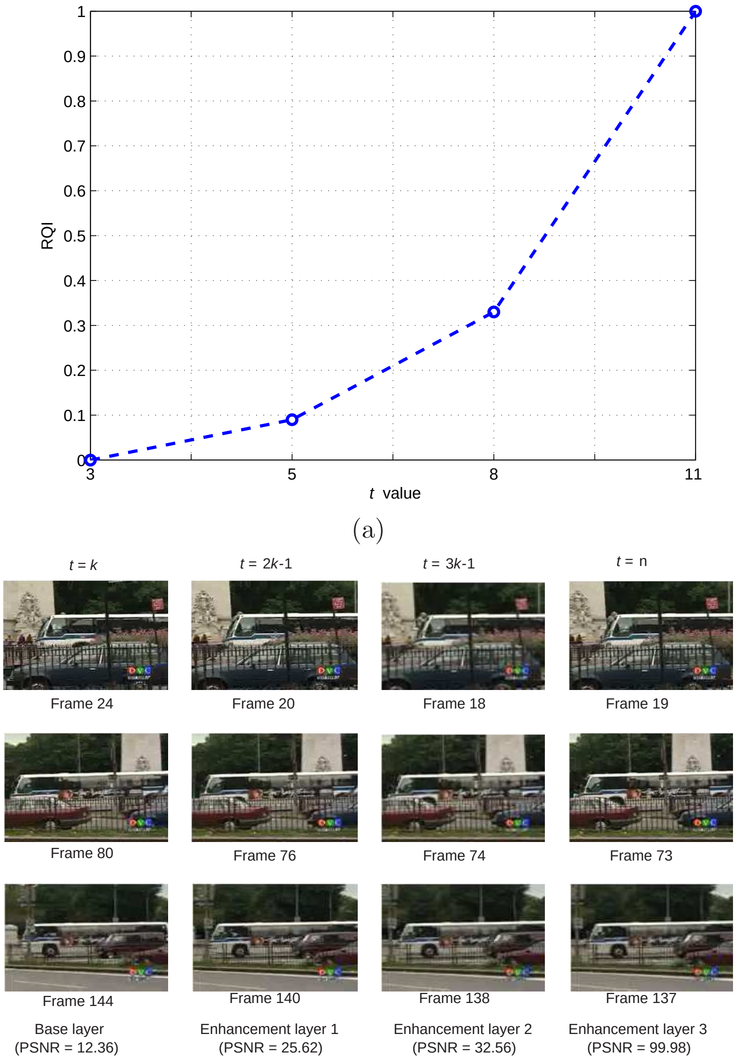
Table 13: Number of operations

n value	Addition operations	Multiplication operations
5	21	19
7	45	55
10	96	167

16 gives the time variation with t, $k \leq t \leq n$, shares for video foreman. The time taken to reconstruct is in seconds. It can be seen that if the size of the input videos is larger, the time taken to reconstruct is also higher, which increases linearly with the value of $t = n$. Having to generate more shares, results in higher computation times.

In Table 13, the number of addition and multiplication operations performed based on the value of n is shown. It can be seen that using higher values of n can result in more addition and multiplication operations. The number of operations grows linearly with the values of n. The computational complexity as a function of k is also linear as one coefficient term is added to the polynomial equation every time the value of k is incremented. Say, if the value of k is doubled, the multiplication and addition operations gets doubled.

From the above experiments, it can be deduced that the proposed SSVS method has comparable execution times to previous SSIS methods, hence it is comparatively efficient. This method also provides gradual reconstruction and smooth scalability, as indicated by the PSNR and RQI values. Further, the size of the share videos is also reduced. Hence, the proposed SSVS method is robust.



(a)

(b)

Figure 15: RQI values for the reconstructed video Bus.yuv for varying t, $k \leq t \leq n$, $k = 3, n = 11$

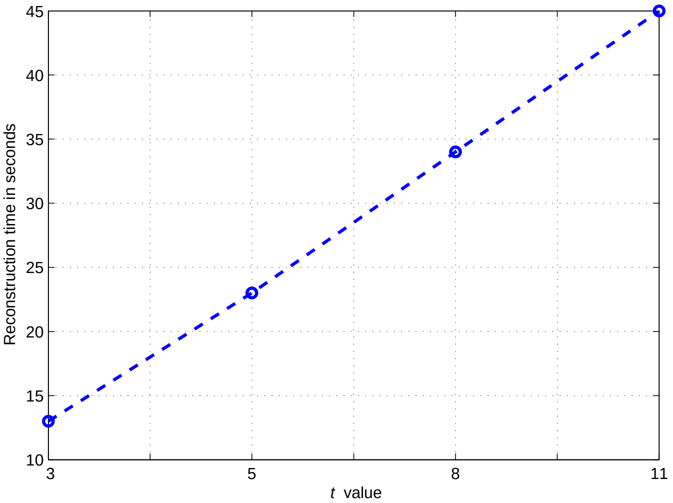


Figure 16: Variation of reconstruction times for foreman.yuv video (with 300 frames) for varying t, $k \leq t \leq n$, $k = 3, n = 11$

5. Conclusion

The proposed SSIS and SSVS methods reduce the size of shares to an optimal value: $\frac{1}{k}$ of the size of the original data. The computation costs of the proposed methods when compared with existing methods by quoting the number of finite field operations and execution time is found reasonable. The security analysis show that the proposed SSIS and SSVS methods are computationally as well as semantically secure. In the future, it will be interesting to explore how the proposed SSVS method can be applied to H.264 videos with multiview video coding (H.264/MVC).

[1] Advanced Encryption Standard, FIPS 197, National Institute of Standards and Technology, 2001.

[2] G. B. Algin and E. T. Tunali. Scalable video encryption of H.264/SVC codec. *Elsevier Journal of Visual Communication and Image Representation*, 22(4):353 – 364, 2011.

[3] S. Alharthi and P. K. Atrey. An improved scheme for secret image sharing. In *Proceedings of IEEE International Conference on Multimedia and Expo, Workshop on Content Protection and Forensics*, pages 1661–1666, Singapore, 2010.

[4] L. Bai, S. Biswas, A. Ortiz, and D. Dalessandro. An image secret sharing method. In *Proceedings of the Ninth International Conference on Information Fusion*, pages 1–6, Florence, Italy, 2006.

[5] S. Bhadravati, M. Khabbazian, and P. K. Atrey. On the semantic security of secret image sharing methods. In *Proceedings of the Seventh International Conference on Semantic Computing*, pages 302–305, Irvine, CA, USA, 2013.

[6] G. R. Blakley. Safeguarding cryptographic keys. In *Proceedings of the AFIPS National Computer Conference*, volume 48, pages 313–317, Arlington, VA, USA, 1979.

[7] H. Cheng and L. Xiaobo. Partial encryption of compressed images and videos. *IEEE Transactions on Signal Processing*, 48:2439–2451, 2000.

[8] A. M. Eskicioglu, S. Dexter, and E. J. Delp. Protection of multicast scalable video by secret sharing: Simulation results. In *Proceedings of the Royal Society of London*, pages 197–204, 2003.

[9] M. Fatemi, T. Eghlido, and M. Aref. A multi-stage secret sharing scheme using all-or-nothing transform approach. In Sihan Qing, ChrisJ. Mitchell, and Guilin Wang, editors, *Information and Communications Security*, volume 5927 of *Lecture Notes in Computer Science*, pages 449–458. Springer, 2009.

[10] M. P. Jhanwar and R. Safavi-Naini. Unconditionally-secure robust secret sharing with minimum share size. In *Proceedings of the 17th International Conference on Financial Cryptography and Data Security*, volume 7859 of *Lecture Notes in Computer Science*, pages 96–110, Okinawa, Japan, 2013.

[11] H. Krawczyk. Secret sharing made short. In *Proceedings of the 13th International Cryptology Conference on Advances in Cryptology*, pages 136–146, London, UK, 1994.

[12] J. Lacan, V. Roca, J. Peltotalo, and S. Peltotalo. Reed-solomon forward error correction (FEC). In *RFC 5510 (Proposed Standard), Internet Engineering Task Force*, 2009.

[13] F. Liu and H. Koenig. A survey of video encryption algorithms. *Elsevier Journal of Computers and Security*, 29:3–15, 2010.

[14] E. Magli, M. Grangetto, and G. Olmo. Transparent encryption techniques for H.264/AVC and H.264/SVC compressed video. *Elsevier Journal of Signal Processing*, 91(5):1103 – 1114, 2011.

[15] L. Qiao and K. Nahrstedt. A new algorithm for mpeg video encryption. In *Proceedings of the First International Conference on Imaging Science, Systems, and Technology*, pages 21–29, Las Vegas, NV, USA, 1997.

[16] C. N. Raju, G. Umadevi, K. Srinathan, and C. V. Jawahar. A novel video encryption technique based on secret sharing. In *Proceedings of 15th IEEE International Conference on Image Processing*, pages 3136–3139, San Diego, CA, USA, 2008.

[17] H. Schwarz, D. Marpe, and T.Wiegand. Overview of the scalable video coding extension of the H.264/AVC standard. *IEEE Transactions on Circuits and Systems for Video Technology*, 17(9):1103–1120, 2007.

[18] A. Shamir. How to share a secret. *Communications of the ACM*, 22:612–613, 1979.

[19] C. Shi and B. Bhargava. A fast MPEG video encryption algorithm. In *Proceedings of the Sixth ACM International Conference on Multimedia*, pages 81–88, Bristol, United Kingdom, 1998.

[20] C. Shi, S. Wang, and B. Bhargava. MPEG video encryption in real-time using secret key cryptography. In *Proceedings of Parallel and Distributed Processing Techniques and Applications*, pages 2822–2828, Las Vegas, NV, USA, 1999.

[21] G. A. Spanos and T. B. Maples. Performance study of a selective encryption scheme for the security of networked, real-time video. In *Proceedings of the Fourth International Conference on Computer Communications and Networks*, pages 2–10, Las Vegas, NV, USA, 1995.

[22] T. Stutz and A. Uhl. Image confidentiality using progressive JPEG. In *Proceedings of the Fifth International Conference on Information, Communications and Signal Processing*, pages 1107–1111, Bangkok, Thailand, 2005.

[23] Z. Su, G. Zhang, and J. Jiang. Multimedia security: A survey of chaos-based encryption technology. In I. Karydis, editor, *Multimedia - A Multidisciplinary Approach to Complex Issue*, pages 53–58. InTech, 2012.

[24] B. Subramanyan, V. M. Chhabria, and T. G. S. Babu. Image encryption based on AES key expansion. In *Proceedings of the Second International Conference on Emerging Applications of Information Technology*, pages 217–220, Kolkata, India, 2011.

[25] L. Tang. Methods for encrypting and decrypting MPEG video data efficiently. In *Proceedings of the Fourth ACM International Conference on Multimedia*, pages 219–229, Boston, MA, USA, 1996.

[26] C. Thien and J. Lin. Secret image sharing. *Elsevier Journal of Computers and Graphics*, 26:765 – 770, 2002.

[27] G. K. Wallace. The JPEG still picture compression standard. *IEEE Transactions on Consumer Electronics*, 38(4):16–34, 1992.

[28] R. Wang and S. Shyu. Scalable secret image sharing. *Elsevier Journal of Signal Processing: Image Communication*, 22(4):363 – 373, 2007.

[29] Z. Wei, X. Ding, R. Deng, and Y. Wu. No tradeoff between confidentiality and performance: An analysis on H.264/SVC partial encryption. In *Communications and Multimedia Security*, volume 7394 of *Lecture Notes in Computer Science*, pages 72–86. Springer, 2012.

[30] Z. Wei, X. Ding, R. Deng, and Y. Wu. Efficient block-based transparent encryption for H.264/SVC bitstreams. *Springer Journal of Multimedia Systems*, 20:165–178, 2014.

[31] T. Wiegand, G. J. Sullivan, G. Bjontegaard, and A. Luthra. Overview of the H.264/AVC video coding standard. *IEEE Transactions on Circuits and Systems for Video Technology*, 13(7):560–576, 2003.

[32] C. Wu and C.-C. J. Kuo. Design of integrated multimedia compression and encryption systems. *IEEE Transactions on Multimedia*, 7:828–839, 2005.

[33] C. Yang and Y. Chu. A general (k, n) scalable secret image sharing scheme with the smooth scalability. *Elsevier Journal of Systems and Software*, 84(10):1726 – 1733, 2011.

[34] C. Yang and S. Huang. Constructions and properties of k out of n scalable secret image sharing. *Elsevier Journal of Optics Communications*, 283(9):1750 – 1762, 2010.

[35] C. Yang and C. Wu. A threshold secret image sharing with essential shadow images. In Jeng-Shyang Pan, Ching-Nung Yang, and Chia-Chen Lin, editors, *Advances in Intelligent Systems and Applications - Volume 2*, volume 21 of *Smart Innovation, Systems and Technologies*, pages 159–166. Springer, 2013.

[36] M. Zeghid, M. Machhout, L. Khriji, A. Baganne, and R. Tourki. A modified AES based algorithm for image encryption. *International Journal of Computer Science and Engineering*, 1:206–211, 2007.