Abstract: - Security of multimedia files is a topic of crucial importance and attracts more and more attention worldwide. Encryption schemes proposed in literature usually consider multimedia files as sequences of bits, without taking into consideration semantic information (video objects). Different video objects may need different levels of protection, some video objects may be the only regions of an image that need protection, or should be accessed separately according to specific privileges. Towards this direction, in this paper an automatic video objects encryption system is proposed. Initially stereoscopic pairs are analyzed and video objects are automatically extracted using a normalized Motion Geometric Spaces (MGSs) approach. Next each video object is decomposed into two levels and the pixels of the lowest resolution level are encrypted iteratively. Finally the encrypted regions are propagated to the highest resolution level. The system not only presents robustness against known cryptanalytic attacks and enables layered access to multimedia files but it also leads to enhanced overall security since region topology is also involved during encryption/decryption.

Key-Words: — chaotic cipher, motion geometric space, multiresolution cryptographic scheme, region topology, stereoscopic video objects.

1. INTRODUCTION

Security of images and video sequences becomes more and more important in a large range of applications such as transmission of medical data, military, pay-TV, confidential video conferencing and video surveillance. To control access multimedia encryption schemes were proposed, where decryption of the data requires a key [1], [2], [3].

Generally it is useful to include a scrambling scheme into secure multimedia systems [4]. In particular, image-scrambling schemes transform an image into another unintelligible image, based on keys only known to the senders and receivers. Early video scrambling schemes were simple due to analog electronics devices; however, the arrival of VLSI technology enabled the storage and digital processing of images and video. Preliminary digital scrambling techniques include line reversal, line dispersal, line segment swapping and line permutation [5]. However these techniques lack robustness against correlation attacks. A more advanced way for pixel permutation was proposed in [6], where the scan order was changed according to a space-filling curve. Another possibility is to scramble content in a transformed domain, as presented in [7], where scrambling of MPEG video sequence was performed in the DCT domain.

On the other hand, lately, some chaos-based encryption systems were proposed. In particular in [8], a chaotic key-based algorithm is proposed, functioning as value substitution cipher. An encryption algorithm that uses the iterations of the chaotic tent map is proposed in [9]. In [10], each plain character is encrypted as the number of iterations performed by the logistic equation. According to another algorithm, each plaintext byte is encrypted using a different chaotic map [11].

However, most of the aforementioned works do not consider video objects, just bits and pixels. Consequently,
issues such as object-based secure distribution of multimedia content, layered access to regions of interest, and efficient transmission/rate control over low bandwidth networks should be considered.

Towards this direction in this paper an automatic stereoscopic video objects multiresolution cryptographic scheme is proposed. The proposed system consists of a stereoscopic video objects extraction module, a multiresolution decomposition/propagation module, and an iterative cipher mechanism. Initially video objects are extracted based on normalized motion geometric spaces. Afterwards each video object is decomposed into two levels. This multiresolution module results in better information diffusion during scrambling and faster effective encryption. Afterwards each video object is decomposed into two levels. This multiresolution module results in better information diffusion during scrambling and faster effective encryption. The system is encrypted using a key for each object. Finally the results are propagated to the highest resolution level. The system is suitable for object-based access to multimedia content, while it provides enhanced security due to topology issues.

2. MGsS BASED VIDEO OBJECTS EXTRACTION

Motion Geometric Spaces (MGsSs) were proposed in [12], aiming at accelerating the process of video object extraction. The scheme is based on the concept of moving onto specified routes and stopping according to color changes. In this paper we extend the previous method by proposing a neighboring MGsS direction normalization algorithm and using a weighted stop-function to obtain a smoother video object contour. More specifically, let us assume that after analyzing a stereoscopic pair of frames, a depth segments map is produced [13]. According to the proposed scheme, firstly a stereoscopic pair of frames, a depth segments map is produced. After that a neighboring MGsS direction normalization algorithm and using a weighted stop-function to obtain a smoother video object contour.

Let us symbolize as MGS(x,y) the Motion Geometric Space of the ith initial point P_i=(x,y). Let us also assume that P_{i-1}=(x_{i-1}, y_{i-1}), P_i=(x_i, y_i) and P_{i+1}=(x_{i+1}, y_{i+1}) are three consecutive initial feature points and u_{i-1,i}, u_{i,i+1} is the vector connecting P_{i-1} (start point) and P_{i+1} (end point). Then the MGsS(P_i) for P_i is the set which contains all points p that satisfy the following equation:

$$MGS(x_0,y_0)=\{p=(x,y): y-y_0=\lambda_i(x-x_0)\},$$

with $\lambda_i=\frac{1}{A_{p_{i-1},P_{i+1}}}$

where $\lambda_{u_{i-1,i},u_{i+1}}$ is the direction coefficient of vector $u_{i-1,i},u_{i+1}$.

However some initial MGsSs may have wrong direction. For this reason a direction normalization process is implemented that utilizes the direction of the neighboring MGsSs. Firstly a unit vector is defined on each MGsS that is parallel to the MGsS. Then, an iterative direction-smoothing algorithm is applied, which can be split into 3 steps:

**Step 1:** For the vector of each MGsS, the vectors of the two neighboring MGsSs are considered and synthesis is performed. More specifically let $v_P$ be the unit vector of MGsS(P), $v_{P_{i-1}}$ the vector of MGsS(P_{i-1}) and $v_{P_{i+1}}$ the vector of MGsS(P_{i+1}). Then two new vectors are produced through synthesis:

$$v_{p_{i-1}}=\frac{v_P+A(i,i-1)\bullet v_{p_{i-1}}}{|v_P+A(i,i-1)\bullet v_{p_{i-1}}|}$$

$$v_{p_{i+1}}=\frac{v_P+A(i,i+1)\bullet v_{p_{i+1}}}{|v_P+A(i,i+1)\bullet v_{p_{i+1}}|}$$

where $A(k,l)$ is a weighting function related to the distance between two neighboring points $P_k=(x_k,y_k)$ and $P_l=(x_l,y_l)$:

$$A(k,l)=\frac{D_{kl}}{D_{kl,\max}} = \frac{\sqrt{(x_k-x_l)^2+(y_k-y_l)^2}}{D_{kl,\max}}$$

Afterwards $v_{p_{i-1}}$ and $v_{p_{i+1}}$ are synthesized:

$$v_f=v_{p_{i-1}}+v_{p_{i+1}}$$

**Step 2:** The angle $\phi$ between $v_f$ and $v_P$ is estimated:

$$\phi=\cos^{-1}\left(\frac{x_fy_f+x_y+y_fy_{f'}}{\sqrt{x_f^2+y_f^2}\sqrt{x_y^2+y_{f'}^2}}\right)$$

where $(x_f,y_f)$ and $(x_y,y_{f'})$ are the coordinates of the heads of vectors $v_f$ and $v_P$, assuming that tails have coordinates (0,0).

**Step 3:** If $\phi < \phi_{th}$ no direction changes occur, else direction changes by $\phi/2$. Next MGsSs that changed direction are kept together with their neighboring MGsSs and the process is repeated.

The direction smoothness algorithm terminates if no direction changes occur during an iteration.

Afterwards each point moves onto its normalized MGsS and stops according to a weighted stop-function, defined as:

$$D=I_D+bM(\theta)$$

where $I_D=I(x',y')$, $I(x,y)$, with $(x',y')$ being the ith initially selected feature point and $(x,y)\in MGS(x_i^{init},y_i^{init})$. Additionally $M(\theta)$ is a smoothness function depending on the angle $\theta$ between vectors $u_{P_{i-1},P_{i}}$ and $u_{P_{i+1},P_{i}}$. The vectors connect points $P_{i-1}, P_i$ and $P_{i+1}$ respectively and these three points are possible stop-positions belonging to three neighboring MGsSs. The role of $M(\theta)$ is to provide a smoother final video object contour, while $b$ is a weighting factor which determines the significance of $M(\theta)$. Then, starting from, say the ith initial feature point, $(x_i^{init},y_i^{init})$, we move onto its respective $MGS(x_i^{init},y_i^{init})$ and the first point at which function $D$ exceeds a threshold $T_D$ is selected as boundary point. At last the video object is extracted by connecting all final points.
3. THE PROPOSED CRYPTOGRAPHIC SCHEME

The proposed cryptographic scheme consists of the encryption and decryption modules, which are described next.

The Encryption Module

In the proposed system and for robustness reasons, for each video object a different 256-bits key is used, which is split into 32 subkeys of 8 bits each (session keys). Furthermore the logistic map function is incorporated:

\[ x_{n+1} = r_i x_n (1 - x_n) \]  \hspace{1cm} (7)

where \( x \) takes values in the interval \([0, 1]\), and \( r_i \) is a parameter that controls the bifurcations of the chaotic map, taking values in the interval \([3.9, 4.0]\).

An overview of the proposed encryption system is depicted in Figure 1. After extracting the video objects and regulating the system parameters the encryption module is activated. Initially each video object is decomposed into two levels and the pixels of the lowest resolution level are scanned from top-left to bottom-right providing plaintext pixels \( p_i \). Afterwards the output of the chaotic map is added to the plaintext pixels producing ciphertext \( c_i \). In every step two successive session keys \( k_i \) and \( k_{i+1} \) are used to regulate the initial conditions of the chaotic map. Session key \( k_i \) passes through box M1, which maps the values of input interval to the domain of the logistic map, and sets the initial value \( x_0 \). On the other hand session key \( k_{i+1} \) participates in the computation of the number of iterations. Furthermore extra robustness is added to the system by a feedback mechanism, which leads the cipher to acyclic behavior. According to this feedback mechanism encryption of each plain pixel depends on the key, the value of the previous cipher pixel and the output of the logistic map.

In particular the feedback mechanism includes two operations: by the first operation the output of the encryption module \( c_i \) is added to session key \( k_{i+1} \) and the result is rounded so that to determine the number of iterations of the logistic map during the next execution. The second operation regulates parameter \( r_i \) (Eq. (7)). In particular the output of the encryption module \( c_i \) passes through box M3, which maps the interval \([0, 255]\) to the interval \([0, 0.1]\). The result is added to constant \( r = 3.9 \) and is fed to the logistic map, so that \( r_i \) takes values in the interval \([3.9, 4.0]\). Additionally box M2 normalizes the output of the logistic map to the interval \([0, 255]\).

As soon as all pixels of the lowest resolution level are encrypted, the results are propagated to the next resolution level, where each cipher pixel of the previous resolution level is placed at its position at the current resolution level, and the encryption process is repeated. Here it should be mentioned that only the plain pixels of the current resolution level are encrypted. The already encrypted pixels of the previous resolution level participate in the encryption process but do not change values. They serve as initial seeds for better
scrambling of the rest of the pixels. The procedure is terminated as soon as the highest resolution level is processed, where cipher video objects are produced. Combination of the encrypted video objects leads to the final encrypted content, where no clues relevant to the contours of the video objects or the structure of the visual information are provided.

**The Decryption Module**

A diagram of the proposed decryption module is given in Figure 2. The decryption module receives at its input an image with encrypted video objects, the contours of the video objects, the secret keys used during encryption and the initial cipher value $c_0$ (used at the first feedback).

**Figure 3:** Depth segmentation. (a) Left channel, (b) Right channel, (c) Depth segments map.

**Figure 4:** Video objects extraction. (a) Initial and normalized MGSs, (b) Connection of the final points, (c) Foreground video object and (d) Background video object.

Afterwards for each video object (detected by the corresponding contour) multiresolution decomposition is performed and the pixels of the lowest resolution level are
scanned from top left to bottom right (as during encryption). Again the chaotic module produces the specific values used during encryption, but now chaotic map values are subtracted from the cipher pixels to produce the plaintext pixels. After finishing with the lowest resolution level, the results are propagated to the next resolution level. The procedure is terminated after the highest resolution level is considered, leading to decryption of the plaintext video objects.

4. EXPERIMENTAL RESULTS

For evaluation purposes the proposed video objects multiresolution cryptographic scheme is thoroughly examined in terms of security and efficiency.

In particular the proposed system is applied to the stereoscopic pair of frames depicted in Figures 3(a) and 3(b), where each frame is of size 668x808 pixels. Initially the stereoscopic pair is analyzed and a depth segments map is estimated (Figure 3(c)). Next initial and normalized MGSs are estimated as depicted in Figure 4(a). In particular for presentation purposes MGSs are depicted for 6 points only. Initial MGSs are shown with a dashed line while a solid line is used for normalized MGSs. Afterwards each point moves onto its MGS and stops according to the stop function of Equation (6). Final points are connected by a spline to detect the foreground video object as it can be observed in Figure 4(b), while the extracted foreground and background video objects are depicted in Figures 4(c) and 4(d) respectively.

Afterwards for each video object multiresolution decomposition is performed (Figure 5 - decomposition in 2 levels) and the cryptographic algorithm is activated for the lowest resolution level. At last the results are propagated to the highest resolution level and the final encrypted content is produced (Figure 6). As it can be observed, the final content (encrypted foreground and background objects) is completely scrambled without containing clues relevant to the number and position of video objects.

The encrypted content can be transmitted to a recipient. In this case the smallest $n-1$ contours (for $n$ video objects) are transmitted together with the encrypted image and the encryption keys. In the case under consideration the total extra elements (coordinates of the pixels comprising the contour) are 3664, as the contour of the foreground video object is comprised of 1832 pixels. These extra elements lead to a bit rate increase of about 0.3 %.

Now the security of the proposed scheme is further examined. Towards this direction let us assume that an unauthorized user knows the contours of the video objects and tries to decrypt the content by brute force attack. If the exact keys are detected then the content can be decrypted (Figure 7(a)). However in case the user tries keys that differ by just one bit, the content will not be decrypted as it can be seen in Figure 7(b). Furthermore assuming that unauthorized users know the exact keys but do not know the contours of the video objects they cannot decrypt the visual content as shown in Figure 7(c). Finally in case that only the contour and the key of the foreground video object are known, only the foreground video object can be decrypted, as shown in Figure 7(d). Similar results for the background video object are presented in Figure 7(e). According to the previous results, the proposed scheme can be used for layered access.
to visual content, where users can view only the content they are authorized to view.

Figure 7: Security of the proposed system. (a) Decryption result when both keys and the contour of the foreground video objects are known, (b) Decryption result when decryption key is different by one bit and contour is known, (c) Decryption result when keys are correct but contours are unknown, and (d) Decryption result when only the contour and the key of the foreground video object are known. (e) Decryption result when only the contour and the key of the background video object are known.

5. CONCLUSION

The proposed system of this paper confronts the problem of video objects encryption in case of stereoscopic sequences, where depth information can be reliably estimated. Initially video objects are extracted by a segmentation method based on normalized MGSs. The extracted video objects undergo multiresolution decomposition and encryption. The multiresolution module is very important since it leads to better scrambling of the plaintext content. The final image consists of the encrypted video objects and does not provide clues relative to the number and positions of video objects. The proposed scheme enables layered access to content, since a different key can be used for each video object. Experimental results illustrate the enhanced security (based on topology) of the proposed scheme.

REFERENCES: