See discussions, stats, and author profiles for this publication at: <https://www.researchgate.net/publication/282703606>

A Secure Lightweight Texture Encryption Scheme

Conference Paper in Lecture Notes in Computer Science · November 2015

DOI: 10.1007/978-3-319-30285-0_28

CITATIONS

4

READS

115

3 authors:

Alireza Jolfaei

Federation University Australia

27 PUBLICATIONS 305 CITATIONS

SEE PROFILE

Xin-Wen Wu

Indiana University of Pennsylvania

91 PUBLICATIONS 486 CITATIONS

SEE PROFILE

Vallipuram Muthukumarasamy

Griffith University

137 PUBLICATIONS 986 CITATIONS

SEE PROFILE

Some of the authors of this publication are also working on these related projects:

Technology and law: robots, blockchain--what is the role of the human in law now? [View project](#)

Flow based Anomaly detection [View project](#)

A Secure Lightweight Texture Encryption Scheme

Alireza Jolfaei^{1(✉)}, Xin-Wen Wu¹, and Vallipuram Muthukkumarasamy¹

School of Information and Communication Technology,
Griffith University, Gold Coast, QLD 4222, Australia
`alireza.jolfaei@griffithuni.edu.au`,
`{x.wu,v.muthu}@griffith.edu.au`

Abstract. Due to the widespread application of augmented and virtual environments, the research into 3D content protection is fundamentally important. To maintain confidentiality, encryption of 3D content, including the 3D objects and texture images, is essential. In this paper, a novel texture encryption scheme is proposed which complements the existing 3D object encryption methods. The proposed method encrypts texture images by bit masking and a permutation procedure using the Salsa20/12 stream cipher. The method is lightweight and satisfies the security requirement. It also prevents the partial disclosure of the encrypted 3D surface geometry by protecting the texture patterns from being partially leaked. The scheme has a better speed-security profile than the full encryption and the selective (4 most significant bit-plane) encryption by 128-bit AES. The encryption schemes are implemented and tested with 500 sample texture images. The experimental results show that the scheme has a better encryption performance compared to the full/selective encryption by 128-bit AES.

Keywords: Texture image · 3D object · Encryption · Salsa Dance · Permutation · Lightweightness · Security

1 Introduction

Virtual reality and augmented reality are about to become explosive growth markets. Over the past decade, a substantial investment and a wide range of exciting prototypes have been made from the tech heavyweights such as Microsoft and Google. It is anticipated that the market for virtual reality and augmented reality will reach \$1.06 billion by 2018 at a Compound Annual Growth Rate (CAGR) of 15.18 % from 2013 to 2018 [1]. The growing applicability of 3D content and its potential revenue suggest the necessity for protecting such assets. The privacy-sensitive content in 3D environments, such as Second Life, are in risk of being recorded or monitored by malicious entities [2]. This allows manufacturing and selling real (counterfeit) objects, which is a great loss for their owner. A solution to this problem is encryption. Since the 1970s, a large number of encryption schemes have been proposed, some of which have been standardized and adopted

worldwide, such as Data Encryption Standard (DES) [3] and Advanced Encryption Standard (AES) [4]. However, the problem of 3D content encryption is beyond the simple application of established and well-known encryption algorithms. This is primarily due to the constraints imposed by the data structure and the application requirements, such as content usability, format compliance, real-time performance, complexity, and the security level. To address these concerns, several attempts have been made to develop robust encryption schemes for 3D content [5–9]. However, all these efforts are mainly focused on the encryption of 3D models rather than texture images.

Texture images are fundamental drawing primitives that add realism to computer graphics by improving surface details. Notionally, a texture image is a 2D image that is wrapped onto the geometry of a 3D model, to give the illusion of a specified pattern to the complex object [10]. Texture images contain intelligible information due to the strong correlation among adjacent elements. As each element is assigned to a particular vertex, texture patterns provide strong cues to the surface orientation, curvature and 3D surface geometry. Therefore, there is a strong correlation between the geodesic distance between pairs of points on the surface and the distance between corresponding pairs of points in the texture image [11]. This relationship provides a lot of information about the 3D geometry. Also, texture leakage may lead to a disclosure of the 3D surface geometry. It is therefore necessary to confuse this relationship by encrypting the texture image.

Texture encryption is a subclass of image encryption in which maintaining the real-time rendering performance, and preventing the partial disclosure of the 3D surface geometry by the texture pattern, are principally important in addition to providing confidentiality for texture images. These requirements may not be an issue in many image or video applications, but they are vital for most 3D applications. To meet these requirements, it is more important to obfuscate the coarse pattern rather than the detail of the texture image. This can reduce the capability of a competent adversary to reconstruct 3D objects exploiting the texture images. Also, it is necessary to keep encryption complexity as low as possible to save resources, such as computation, memory and bandwidth. One potential solution to the problem of texture encryption is in the use of a lightweight encryption scheme with a high level of security, tailored for maintaining the real-time performances. Using this idea, this paper proposes a novel texture encryption scheme that satisfies the need for both lightweightness and security. The proposed cipher uses Salsa20/12 [12] as its core encryption primitive. Salsa20 is one of the finalists of the eSTREAM project [13] and is constructed by a simple and scalable design, which is appropriate for software implementations. Although Salsa20 has not received its deserved attention compared to AES, it has good potential for being used in multimedia applications, in which high-speed encryption is required. It is shown that in comparison with full encryption methods using the 128-bit AES, the proposed texture encryption method provides comparable level of security but with much faster performance. Also, compared to selective encryption methods [14], in which only a subset of

the input bitstream is encrypted using the 128-bit AES, the proposed texture encryption method is much faster and more secure because it protects the entire input bitstream. Furthermore, it is shown that Salsa Dance conceals the shape and boundaries of underlying 3D objects, while the full and selective encryption using AES cannot protect such information.

This is the first paper that proposes a technical solution for the confidentiality problem of texture images. Although many image encryption methods have been proposed in the literature, they are not designed primarily for addressing the technical requirements of 3D applications. In this paper, the proposed encryption scheme is compared with AES, because AES is currently the main industrial encryption standard used in many multimedia applications. AES is a well-studied cipher and no practical attack has been found against it to date. In addition, AES is fast and on a Core 2 architecture, for example, runs around 12 cycles/byte for long streams. This speed is quite fast compared to other multimedia operations such as compression (For instance, Lempel-Zev and ZLIB Compressions). However, it is shown that AES cannot sufficiently address the confidentiality requirements of texture images, and therefore texture images encrypted by AES may leak crucial information about the protected 3D models. It is also shown that in comparison with AES, the proposed encryption scheme not only maintains the confidentiality of texture images but also maintains the security of protected 3D models from surface reconstruction attacks.

The remainder of this paper is organised as follows. Section 2 describes the encryption and decryption procedures. In Sect. 3, the performance of the proposed cipher is evaluated. Section 4 evaluates the security of the cipher from the data level and the semantic level. Finally, Sect. 5 concludes that the proposed texture encryption method is secure, relatively lightweight and prevents the partial disclosure of the protected 3D surface geometry by the texture pattern.

2 Proposed Texture Encryption Scheme

In a true color (24-bit) representation, 94.125 % of the total information is stored in the upper nibbles (4 bit-planes) of the texture image. This suggests employing a strict strategy to encrypt the upper nibbles and a lenient scheme for the encryption of lower nibbles. This approach improves the encryption performance and reduces the memory usage. The proposed scheme encrypts the upper nibble-image using a fast stream cipher, that is, Salsa20/12 [12], and scrambles the bit stream of the lower nibble-image by a zigzag pattern permutation. This mechanism is consistent with the movement performed in the (Latin American) Salsa dance. We therefore call our encryption mechanism ‘Salsa Dance’.

To elaborate the steps of the encryption algorithm, denote by $\mathbf{P}$ the plain-image, $\mathbf{N}$ the nibble-image, and $\mathbf{C}$ the cipher-image. In a 24-bit true color representation, each plain-image, nibble-image or cipher-image is represented by three $M \times N$ matrices, namely, $\mathbf{R}$, $\mathbf{G}$, and $\mathbf{B}$ color layers. In any color layer of RGB, for any x ($1 \leq x \leq M$), and y ($1 \leq y \leq N$), let $p(x, y)$, $n(x, y)$

and $c(x, y)$, be the entry value at the position (x, y) of the plain-image, nibble-image and cipher-image, respectively. $p(x, y)$ and $c(x, y) \in \{0, 1, \dots, 255\}$, and $n(x, y) \in \{0, 1, \dots, 15\}$.

The encryption procedure is described as follows, for one color layer of a 24-bit texture image. The encryption procedure is the same for the other color layers. Firstly, the plain-image is divided into two nibble-images, which are $\mathbf{N}_1$ and $\mathbf{N}_2$, by splitting every entry into upper and lower nibbles. For any x ($1 \leq x \leq M$) and y ($1 \leq y \leq N$), $n_1(x, y)$ and $n_2(x, y)$ are defined as follows:

$$n_1(x, y) = p(x, y) \bmod 2^4, \quad (1)$$

$$n_2(x, y) = (p(x, y) - n_1(x, y)) \cdot 2^{-4}. \quad (2)$$

In the upper nibble-image encryption, the binary stream of the upper-nibble images with size $M \times 4N$ is masked with a binary stream of the same size generated by the Salsa20/12 stream cipher. This procedure not only protects the coarse shape (major information) of the texture images from being leaked, but also prevents the disclosure of the 3D surface geometry. In the lower nibble-image encryption, the lower nibble-image is first extended to a bit-plane image with size $M \times 4N$, which is constructed by expanding every column of the lower nibble-image into 4 bit-plane columns. The bit-plane image then undergoes a bit-level zigzag pattern permutation process $\text{Perm}(\cdot)$. Displacement of the bit locations not only annihilates the high correlation among the nibbles but also increases the security level of the encrypted texture images. This process is as follows:

Assume that the entries of the bit-plane image are scanned in a raster order and they are enumerated by positive integers. Let $\mathbf{R}$ denote the matrix of the entry (bit) locations, that is,

$$\mathbf{R} = \begin{bmatrix} 0 & M & \cdots & 4MN - M \\ 1 & M + 1 & \cdots & 4MN - M + 1 \\ \vdots & \vdots & \vdots & \vdots \\ M - 1 & 2M - 1 & \cdots & 4MN - 1 \end{bmatrix}. \quad (3)$$

An additional binary sequence of length $\lceil \log_2(4MN) \rceil$ with value s is iterated by Salsa20/12. Then, $\text{mod}(s, MN)$ is used to select an entry in the bit-plane image, which determines the starting point for the zigzag-pattern permutation of the entries. To clarify further, Fig. 1 shows a zigzag path for the scanning of entries in a bit-plane image with size 3×4 . In Fig. 1a, if $\text{mod}(s, 12) = 7$, then the entry scanning commences from the 7-th entry, and stops at the 9-th entry which is previous to the initial one, that is 7. During the scanning process, bits encountered in the path are arranged sequentially, column by column in the same matrix. On completion of the permutation process, not only is every bit dislocated (diffusion), but also nibble values are modified (confusion) within the bit-plane image. For $\text{mod}(s, 12) = 7$, the permutation result of the test bit-plane image is depicted in Fig. 1b. Following the permutation process, the encrypted lower nibble-image with size $M \times N$ is reconstructed by combining

Fig. 1. (a) A zigzag path to scramble bits of a bit-plane image, (b) Permutation result of the bit-plane image for $\text{mod } (s, 12) = 7$.

every 4 consecutive columns of the scrambled bit-plane image. Finally, the cipher-image is constructed by the radix 2^4 combination of the encrypted upper and lower nibble-images.

In summary, the whole encryption process $E(\cdot)$ is as follows:

$$P = 2^4 \cdot N_2 + N_1, \quad (4)$$

$$C = E(P) = 2^4 \cdot E_2(N_2) + E_1(N_1), \quad (5)$$

where

$$E_2(N_2) = \text{Salsa20/12}(N_2), \quad (6)$$

$$E_1(N_1) = \text{Perm}(N_1). \quad (7)$$

P , N_1 , N_2 , and C denote the plain-image, lower nibble-image, upper nibble-image and the cipher-image, respectively. In decryption, the cipher-image is divided into lower and upper nibble-images. The upper nibble-image is decrypted by the same key stream used in encryption, and the lower nibble-image is decrypted by the inverse permutation procedure. Note that in 24-bit texture images there is a strong correlation among different color layers of the image. Therefore, encryption of different color layers using the same key may reveal the underlying pattern. To address this issue, Salsa20/12 has a 64-bit nonce which changes after each color layer encryption. This ensures that whenever the same message is encrypted twice, the ciphertext is always different. If the same nonce and key are used to encrypt two different plaintexts, then the keystream can be cancelled out by masking the two different ciphertexts together.

3 Performance Analysis

To evaluate the performance of the proposed cipher, we implemented: (i) the full encryption by 128-bit AES, (ii) selective encryption of the 4 most significant bit-planes using 128-bit AES, and (iii) Salsa Dance, on a machine with Intel Core 2 2.4 GHz processor and 4 GB of installed memory. In this paper, the ECB mode of AES algorithm has been chosen to encrypt the texture images. AES supports several modes of operation, among which ECB allows parallelised

Fig. 2. Encryption results of a sample texture image: (a) Original image, (b) encrypted image using full AES, (c) encrypted image using selective AES, (d) encrypted image using Salsa Dance.

Table 1. Comparison of the relative CPU time

Encryption schemes	Relative CPU time
Selective AES	2.47
Full AES	4.95
Proposed (Salsa Dance)	1.00

encryption/decryption and achieves better performance with trivial sequential message scheduling [15]. It is also suitable for applications that require random read/write to encrypted data. We tested the encryption performance using 500 sample texture images from CGTextures [16]. Figure 2 shows one test texture image with its corresponding encryption results. It is observed that Salsa Dance dissipates the correlation among the entries of the texture image while the full and selective encryption using AES cannot annihilate the coarse pattern of the texture image.

In the proposed encryption method, 24-bit texture images with size MN (that is, 24 MN bits in total) are encrypted by a pseudorandom binary sequence with size $12MN + 3\lceil \log_2(4MN) \rceil$. In other words, the proposed cipher encrypts the input data by generating a pseudorandom sequence with the size of almost 50% of the data. This means that compared to conventional full encryption methods, the proposed method reduces the computational cost to approximately half. This reduction in the computational cost can therefore save computational power, storage space, processing time, and transmission bandwidth; and, therefore, it would allow more processes to be executed in parallel.

Compared to the 10 rounds 128-bit AES, Salsa20/12 is considerably faster. On a Core 2 architecture, for example, Salsa20/12 runs at 2.54 cycles/byte for long streams, while the fastest speed reported for 128-AES is 12.59 cycles/byte [17]. This implies that Salsa20/12 is almost 5 times faster than the 10 rounds 128-bit AES. Therefore, Salsa20 provides a much better speed-security profile than AES. To evaluate the encryption speed of the proposed cipher, numerous encryption timing tests were performed. In addition, to have an accurate benchmark result, each timing test was executed 10 times and the average time was recorded. The results of timing tests demonstrated that the 4 bit-planes

selective encryption methods have speed overheads of 247 % on average compared to Salsa Dance. Also, the full AES schemes have 495 % speed overheads compared to Salsa Dance. Table 1 compares the execution time of the encryption methods. Hence, the experimental results indicate that Salsa Dance has a better encryption performance than the full and selective encryption using AES.

4 Security Analysis

From the data level point of view, the security of encryption, including the upper nibble method and the lower nibble method, relies on the security of the encryption primitive, that is, Salsa20/12. To the best of the authors' knowledge, the best cryptanalysis breaks 8 out of 20 rounds of Salsa20 to recover the 256-bit secret key in 2^{251} operations, using 2^{31} keystream pairs [18]. Also, it is conjectured that Salsa20 and AES reach security with about the same number of rounds [12]. This means that the upper nibble encryption method offers a high confidentiality level. In addition, the lower nibble encryption method, that is, the permutation procedure, is secure as well, because the pseudorandom key stream controlling the permutation is generated by Salsa20/12. The generated key stream is different even for the same color layer encrypted at different sessions. This makes the permutation scheme robust to known/chosen plaintext attacks. Hence, the only attack model applicable to the permutation method is the ciphertext-only attack [19], in which the attacker can only access the lower nibble-image of the cipher-image and attempts to recover the lower nibble-image of the original image by trying all possible permutations ($4MN$ possible arrangements in each color layer). This attack becomes cumbersome and even impractical by increasing the input size MN (This increases the data complexity of the attack).

However, from the semantic level point of view, encrypted texture images may contain redundant information which may be employed to not only retrieve the original texture images but also to reconstruct 3D objects. To evaluate the security of encryption to redundancy based attacks, several measurements were performed, including a correlation analysis, a key sensitivity analysis, and an edge detection analysis. Each of these measurements is described in detail in the following subsections.

4.1 Correlation Analysis

In the texture images, each pixel is highly correlated with its adjacent pixels. Therefore, the adversary may study the correlation among the pixels to determine a meaningful pattern inside the encrypted texture image. An ideal encryption algorithm should completely dissipate such relationship and produce cipher-images with no correlation in the adjacent pixels. A correlation of a pixel with its neighbouring pixel is then given by a 2-tuple (x_i, y_i) where y_i is the adjacent pixel of x_i . The following equation is used to study the correlation between two adjacent pixels in horizontal, vertical and diagonal orientations.

Fig. 3. Correlation analysis and distribution of two adjacent pixels in the plain-image and cipher-image.

$$corr_{(x,y)} = \frac{1}{n-1} \sum_{i=0}^n \left(\frac{x_i - \bar{x}_i}{\sigma_x} \right) \left(\frac{y_i - \bar{y}_i}{\sigma_y} \right), \quad (8)$$

where, x and y are intensity values of two neighbouring pixels in the image, n represents the total number of 2-tuples (x_i, y_i) , and σ_x and σ_y represent the local standard deviation, respectively. To test the impact of encryption by Salsa Dance on the correlation among the adjacent pixels, we performed several correlation tests. Figure 3 shows the correlation distribution of two adjacent pixels in the plain-image shown in Fig. 2a and its corresponding cipher-image. It is observed that neighbouring pixels in the plain-image are highly correlated, while the neighbouring pixels in the encrypted image are almost uncorrelated.

Table 2 shows the results for correlation coefficients of the ciphers under study. The numerical results indicate that the correlation coefficients of plain-images are far apart from cipher-images. Also, results show that the selective/full AES and Salsa Dance efficiently dissipate the correlation among pixels within each color layer. Furthermore, we computed the 2D correlation coefficients between every two color layers of the encrypted images. Table 3 shows the correlation coefficients between different color layers of the cipher-images produced by the AES encryption and Salsa dance. It is observed that Salsa Dance can reduce the strong correlation between the color layers much better than the encryption using 128-bit AES. Hence, the results of the correlation analysis indicate that compared to the full and selective encryption using 128-bit AES, Salsa Dance has a better encryption performance and is more robust to redundancy based attacks.

4.2 Key Sensitivity Analysis

It is possible for an adversary to induce modifications in the secret key via tampering or fault injection [20]. This helps the adversary to observe the redundancy under different encryption keys and deduce a relationship between the used keys. To resist such kinds of analyses, a texture image encryption scheme should be sensitive to changes to the secret key. In other words, a change in a single bit of the secret key should produce a completely different cipher-image. The more the visual data is sensitive toward the secret key, the higher would be the amount of data randomness. To test the key sensitivity of the proposed algorithm, a number of texture images were encrypted using the selective/full AES and Salsa Dance with an original secret key ($K = 0$, $IV = 0$) and a slightly modified secret key ($K' = 1$, $IV = 0$). Numerical results show that the proposed technique is highly sensitive toward the small alterations of the secret key, that is, a different cipher-image is produced when the secret key is slightly changed. For comparison purposes, we used the PSNR measure. The higher the PSNR, the closer the images are. Table 4 provides the PSNR values of the encrypted images for the test image shown in Fig. 2a. It is observed that given the test image shown in Fig. 2a, encryption by slightly different secret keys creates different cipher-images by the selective/full AES and Salsa Dance. However, Table 4 shows that compared to the encryption by selective/full AES, Salsa Dance produces more dissimilar cipher-images with only 1-bit of change in the secret key. This indicates the high sensitivity of the proposed method to changes of the key, which makes the analysis of Salsa Dance even harder for the adversary in respect to finding any relationship between the used keys.

4.3 Edge Detection Analysis

From the semantic point of view, the coarse pattern of the visual data (that is, the shape information) carries more information than the details. Disclosure of the shape information not only may help a competent adversary in retrieving the texture image but also may facilitate the reconstruction of underlying

Table 3. 2D correlation coefficients between the RGB color layers of the cipher-images

Encryption schemes	Selective AES	Full AES	Proposed (Salsa Dance)
Between R and G	0.3639	0.3237	0.0016
Between R and B	0.3484	0.3041	0.0050
Between G and B	0.3834	0.3041	0.0018

Table 4. Comparison of the PSNR values

Encryption schemes	Selective AES	Full AES	Proposed (Salsa Dance)
Between the original and encrypted image with original key	6.0403dB	6.0345dB	6.0839dB
Between the original and encrypted image with 1-bit different key	6.2709dB	6.1846dB	6.0821dB
Between the encrypted images using the original and modified keys	8.5122dB	8.3964dB	7.7680dB

3D objects. Therefore, the adversary would attempt to identify and locate the boundaries of the protected object within the encrypted texture images. The object boundaries, as well as sharp variations in surface structure, are typically manifested by sharp changes in pixel intensities. However, the randomness of encrypted images makes the edge detection hard. To this end, the adversary may use nonlinear operations, such as median filtering, to reduce the noise while maintaining the edges. He/she may then employ gradient and Laplacian operators for the edge-detection. This information is essential for the correct reconstruction of 3D surfaces [21]. To evaluate the resistance of the proposed texture encryption scheme to this kind of analysis, we examined the cipher-images using different edge detection methods [22], including the Canny method. Figure 4 shows the results of the edge detection analysis on the cipher-images of Fig. 2 using the Canny method. It is observed that Salsa Dance discloses no information about the shape and boundaries of the underlying 3D object, while the full and selective encryption using AES cannot resist the edge detection analysis.

Fig. 4. Results of edge detection by median and Canny filtering on the encrypted image of (a) full AES, (b) selective AES, and (c) Salsa Dance.

5 Conclusion

To overcome the limitations of the current techniques in addressing the confidentiality requirement of texture images, this paper proposes a technical solution that meets the constraints imposed by the structure of texture images, such as large data volume, and the application requirements, such as real-time performance, complexity, and the security level. The proposed cipher encrypts texture images by bit masking and a permutation procedure using the Salsa20/12 stream cipher. Compared to the full/selective encryption using 128-bit AES, the proposed cipher is relatively lightweight and provides a better encryption performance. Salsa Dance also considerably dissipates the correlation among the entries of the texture image. This annihilates the coarse pattern of the plain-image and prohibits the data leakage from texture images. The key sensitivity analysis showed that even a single bit change in the secret key will result in an entirely different cipher-image. Thus, the original texture image cannot be recovered even though there is a slight difference between the encryption and decryption keys. Furthermore, Salsa Dance conceals the shape and boundaries of the underlying 3D object, while the full and selective encryption using AES is not secure from the edge detection analysis. Therefore, texture encryption by Salsa Dance not only maintains the confidentiality of texture images but also maintains the security of protected 3D models from surface reconstruction attacks using the data provided by the texture images. Due to space limitation, some preliminary results related to the security of Salsa Dance have been presented in this paper. Detailed results backed by theory and cryptanalysis will be presented in the extended version of this paper.

References

1. Markets and Markets: Augmented reality & virtual reality market by technology types, sensors, components, applications & by geography, global forecast and analysis to 2013–2018 (2014)
2. Chen, Y., Kim, T.-K., Cipolla, R.: Inferring 3D shapes and deformations from single views. In: Daniilidis, K., Maragos, P., Paragios, N. (eds.) *Computer Vision – ECCV 2010*. LNCS, vol. 6313, pp. 300–313. Springer, Heidelberg (2010)
3. United States National Institute of Standards and Technology (NIST): Announcing the Data Encryption Standard (DES). Federal Information Processing Standards, Publication **46–3** (1999)
4. United States National Institute of Standards and Technology (NIST): Announcing the Advanced Encryption Standard (AES). Federal Information Processing Standards, Publication **197** (2001)
5. Koller, D., Turitzin, M., Levoy, M., Tarini, M., Croccia, G., Cignoni, P.: Protected interactive 3D graphics via remote rendering. *ACM Trans. Graph.* **23**(3), 695–703 (2004)
6. Phelps, N.: Method for exchanging a 3D view between a first and a second user. US patent 2008/0022408 A1 (2008)
7. Éluard, M., Maetz, Y., Dorr, G.: Geometry-preserving encryption for 3D meshes. In: *Actes de COMpression et REpresentation des Signaux Audiovisuels*, pp. 7–12 (2013)

8. Éluard, M., Maetz, Y., Lelievre, S.: Method and device for 3D object protection by transformation of its points, US Patent 8869292 (2014)
9. Jolfaei, A., Wu, X.-W., Muthukkumarasamy, V.: A 3D object encryption scheme which maintains dimensional and spatial stability. *IEEE Trans. Info. Forens. Secur.* **10**(2), 409–422 (2015)
10. Garcia, E., Dugelay, J.-L.: Texture-based watermarking of 3D video objects. *IEEE Trans. Cir. Syst. Video Tech.* **13**(8), 853–866 (2003)
11. Zigelman, G., Kimmel, R., Kiryati, N.: Texture mapping using surface flattening via multidimensional scaling. *IEEE Trans. Vis. Comput. Graph.* **8**, 198–207 (2002)
12. Bernstein, D.J.: Salsa20 security. (2005). <http://cr.yp.to/snuffle/security.pdf>
13. Babbage, S., Canniere, C.D., Canteaut, A., Cid, C., Gilbert, H., Johansson, T., Parker, M., Preneel, B., Rijmen, V., Robshaw, M.: The eSTREAM portfolio, eSTREAM, ECRYPT Stream Cipher project (2008)
14. Podesser, M., Schmidt, H.P., Uhl, A.: Selective bitplane encryption for secure transmission of image data in mobile environments. In: The 5th Nordic Signal Processing Symposium, pp. 1–6 (2002)
15. Bogdanov, A., Lauridsen, M.M., Tischhauser, E.: Comb to pipeline: Fast software encryption revisited. In: Leander, G. (ed.) FSE 2015. LNCS, vol. 9054, pp. 150–171. Springer, Heidelberg (2015)
16. <http://www.cgtextures.com/>, September 2015
17. Bernstein, D.J.: Which phase-3 eSTREAM ciphers provide the best software speeds (2008). <http://cr.yp.to/streamciphers/phase3speed-20080331.pdf>
18. Aumasson, J.-P., Fischer, S., Khazaei, S., Meier, W., Rechberger, C.: New features of latin dances: analysis of Salsa, Chacha, and Rumba. In: Nyberg, K. (ed.) FSE 2008. LNCS, vol. 5086, pp. 470–488. Springer, Heidelberg (2008)
19. Stinson, D.: *Cryptography: Theory and Practice*. CRC Press, Boca Raton (2006)
20. Bellare, M., Cash, D.: Pseudorandom functions and permutations provably secure against related-key attacks. In: Rabin, T. (ed.) CRYPTO 2010. LNCS, vol. 6223, pp. 666–684. Springer, Heidelberg (2010)
21. Saxena, A., Sun, M., Ng, A.Y.: Make3D: learning 3D scene structure from a single still image. *IEEE Trans. Pattern Anal. Mach. Intell.* **31**(5), 824–840 (2009)
22. Parker, J.R.: *Algorithms for Image Processing and Computer Vision*, 2nd edn. Wiley, New York (2010)