Fast numerical generation and hybrid encryption of a computer-generated Fresnel holographic video sequence

Peter Tsang[1], K.W.-K. Cheung[1*], and T.-C. Poon[2,3]

[1]*Department of Electronic Engineering, City University of Hong Kong, Tat Chee Avenue, Kowloon, Hong Kong, China*

[2]*Bradley Department of Electrical and Computer Engineering, Virginia Tech, USA*

[3]*Shanghai Institute of Optics and Fine Mechanics, Chinese Academy of Sciences, Shanghai 201800, China*

[*]*Corresponding author: 50695250@student.cityu.edu.hk*

Received July 2, 2010; accepted August 20, 2010; posted online January 21, 2013

Research demonstrates that a Fresnel hologram can be generated and simultaneously encrypted numerically based on a secret symmetric key formed by the maximal length sequence (M-sequence). The method can be directly extended to encrypt a video holographic clip in a frame-by-frame manner. However, given the limited combination of signals in the family of M-sequence, hacking the secret key through trial and error can be time consuming but not difficult. In this letter, we propose a method that is difficult to crack with brute force for encrypting a holographic video sequence. An M-sequence is first randomly assigned to encrypt each frame of the holographic video signal. Subsequently, the index of the selected M-sequence, which is necessary to decrypt the hologram, is encrypted with the RSA algorithm before transmitting to the receiving end. At the receiving end, the decoder is provided with a private key to recover the index for each frame, and the corresponding M-sequence is used to decrypt the encoded hologram.

OCIS codes: 090.0090, 090.1760, 090.1995.

doi: 10.3788/COL201311.020901.

In a video holography system, generating hologram signals in real-time and incorporating effective encryption are equally important. These processes ensure that the contents can only be accessed with full quality by legitimate viewers. A number of studies on these two aspects have been conducted. Techniques for encrypting holographic/optical images[1−7] and for the fast generation of digital holograms[8,9] have been reported. However, these two areas are generally addressed separately, and little has been mentioned on how they can be integrated as a single entity. An attempt on the fast generation and encryption of a hologram has been recently reported[10]. In this approach, a hologram is generated at video frame rate from an interim signal that is encrypted with the maximal length sequence (M-sequence)[11]. The latter is taken as a private symmetric key in the encryption process. At the receiving end, the hologram has to be decoded with the same private key prior to reconstruction. The method can be directly extended to the encryption of video holographic sequences. However, given the limited number of M-sequence, hacking the private key by trial and error is easy. In this letter, we propose a method for encrypting a holographic video sequence that is difficult to crack with brute force.

An overview of the integrated hologram generation and encryption method in Ref. [10] (hereafter referred to as the parent method) is provided. For clarity, the description is based on the essential block of the method shown in Fig. 1. The input is a three-dimensional (3D) object scene represented by a point set $P = [p_0(x_0, y_0, z_0), p_1(x_1, y_1, z_1), \cdots, p_{N-1}(x_{N-1}, y_{N-1}, z_{N-1})]$, and the output is an on-axis hologram $D(x, y)$. The 3D object scene is initially partitioned into a vertical set of evenly spaced horizontal scan planes.

Assuming that the range of depth of the object points is small and centered at $z = z_o$, the on-axis hologram (according to Fresnel diffraction) contributed by each horizontal scan plane can be approximated as

$$D(x,y)_\tau \approx \sum_{i=0}^{N(\tau)} \frac{a_i}{r_i} \exp\left[\mathrm{j}k\frac{(x-x_i)^2}{2z_i}\right] \exp\left[\mathrm{j}k\frac{(y-\tau)^2}{2z_o}\right]$$

$$= \exp\left[\mathrm{j}k\frac{(y-\tau)^2}{2z_o}\right] \sum_{i=0}^{N(\tau)} \frac{a_i}{r_i} \exp\left[\mathrm{j}k\frac{(x-x_i)^2}{2z_i}\right]$$

$$= R(y-\tau)\, O(x,\tau), \quad (1)$$

where a_i represents the intensity of the ith point; r_i is the distance between the object point and a point (x, y) on the diffraction plane; $k = \frac{2\pi}{\lambda}$ is the wavenumber; λ is

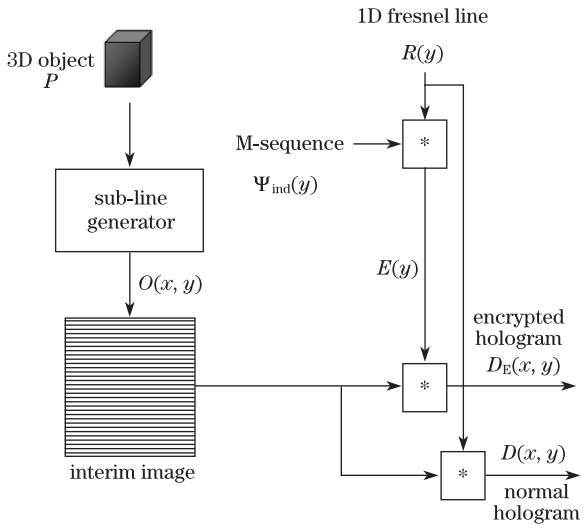

Fig. 1. Integrated hologram generation and encryption in Ref. [10].

the wavelength of the object wave. τ and $N(\tau)$ are the vertical position of the plane and the number of object points on it, respectively. The total object beam is then equal to the superposition of the contribution in each scan plane as

$$D(x,y) = \sum_{\tau}^{D(x,y)_{\tau}} = \sum_{\tau}^{O(x,\tau)R(y-\tau)} . \tag{2}$$

Equation (2) is equivalent to a convolution process that can be represented by

$$D(x,y) = O(x,y) * R(y) . \tag{3}$$

To encrypt the hologram, the function $R(y)$ is first convolved with an M-sequence $\Psi_{\mathrm{ind}}(y)$ prior to convolving with the interim image. The encrypt hologram is given by

$$\begin{aligned} D_E(x,y) &= O(x,y) * R(y) * \Psi_{\mathrm{ind}}(y) \\ &= O(x,y) * E(y) . \end{aligned} \tag{4}$$

A comparison of Eqs. (3) and (4) reveals no extra computation involved in the encryption process. To reconstruct the original 3D contents at the receiving end, the article indicates that the encrypted hologram has to be decoded by correlating it with the same M-sequence adopted in the encrypting process. Thus, mathematically,

$$\begin{aligned} D(x,y) &= D_E(x,y) \circ \Psi_{\mathrm{ind}}(y) \\ &= O(x,y) * R(y) * [\Psi_{\mathrm{ind}}(y) \circ \Psi_{\mathrm{ind}}(y)] \\ &= O(x,y) * R(y) * \delta(y) = D(x,y) . \end{aligned} \tag{5}$$

The decryption process in Eq. (5) can be realized with inverse filtering in the frequency domain with a small amount of computation. This process also implies that the encryption and decryption is lossless.

As abover-mentioned, the parent method can be directly extended to encrypt each frame of a holographic video sequence. However, with sufficient time and computing resources, the private key can be easily hacked by decrypting the encrypted hologram with all the possible M-sequences and selecting the sequence that provides the correct result. Afterwards, the same identified M-sequence can be used to decrypt the rest of the holographic frames. To overcome this problem, we propose the novel encryption method that is depicted in Fig. 2. Similar to the parent method, an interim image $O(x,y)$ comprising of a vertical stack of sub-lines, each corresponding to a row of the object scene, is generated for each frame in the holographic video sequence. Subsequently, a hologram frame is generated by convolving each column of the interim image with a one-dimensional (1D) signal $E(y)$. The latter is derived from the convolution of a 1D Fresnel line $R(y)$ with a randomly selected M-sequence from a family of M-sequence members (e.g., a group of 16 M-sequences with $M = 8$) by a random number generator. This random selection ensures that each hologram frame has a different key that cannot be obtained from another frame. The selected M-sequence is taken as a symmetrical encryption key refreshed with each input frame. Next, the index representing the selected encryption key is padded (a security measure explained later) and encoded with the RSA algorithm[12] based on a random public key.

At the receiver, the encrypted index is decoded with a private key, unpadded, and used to select the corresponding M-sequence to recover the encrypted hologram as shown in Fig. 3.

The RSA algorithm[12] has been reported in a number of studies, and only an overview is presented herein. A RSA algorithm is a kind of public key cryptography where the encryption is based on a pair of keys each comprising of an ordered pair of integers. The first integer is a public key (e,n) that is publicly available, and the second integer is a private key (d,n) that is only known by the user. Anyone who is given the public key can encrypt a message, but only the user who holds the private key can decrypt and recover the content. Table 1 shows the steps for generating the pairs of keys.

The secret message $'ind'$, which is an integer representing the index of the randomly selected M-sequence for encrypting the hologram, is encrypted into a ciphertext as

$$C \equiv ind^{e} \bmod n. \tag{6}$$

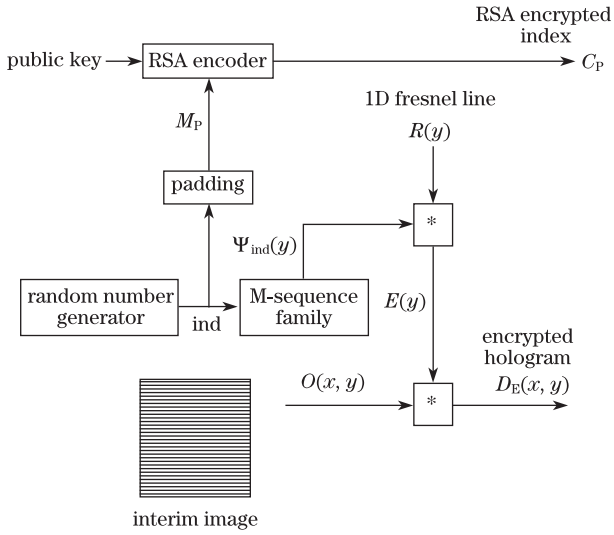

Fig. 2. Proposed encryption scheme.

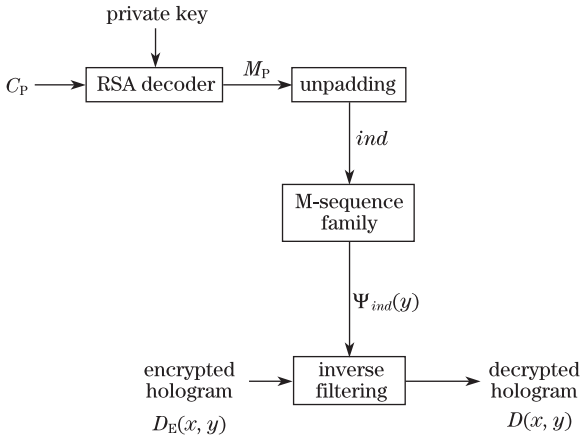

Fig. 3. Decryption of the hologram encrypted with the proposed scheme.

Table 1. Generation of the Public and Private Keys in the RSA Algorithm

Step	Operation
1	Generate two different large prime numbers p and q of similar size.
2	Determine $n = pq$ and $m = \Phi(p)\,\Phi(q)$, where $\Phi(p)$ is Euler's totient of a defined as $\Phi(p) = (p-1)$.
3	Select an integer such that $e < m$ and $\gcd(e, m) = 1$, where $\gcd(a, b)$ denotes the greatest common divisor of two numbers a and b.
4	Determine d such that $de \equiv 1 \bmod m$, where mod denotes the modulus operation.
5	Take the pair of numbers e and n to be the public key, as well as d and n to be the private key.

The ciphertext can be used to recover the message as

$$ind \equiv C^d \bmod n. \tag{7}$$

However, this is not safe from the security point of view because the private key can be obtained from sufficient pairs of message and its corresponding ciphertext. To prevent this loophole, the symmetric key is first encoded with the type 1 PKCS1-V_1.5[13] padding (insertion of data which is unrelated to the message) prior to encryption with RSA. Given an index ind, the padded message $M_{\rm p}$ becomes

$$M_{\rm P} = Z \parallel T \parallel PS \parallel Z \parallel ind, \tag{8}$$

where Z and T denote an octet of value zero and two, respectively; PS is a sequence of one or more octets each with a randomly selected value; $\parallel$ is the concatenation operator. When $M_{\rm P}$ is decrypted at the receiving end, the rest of the message is discarded and only the index is extracted. After applying padding, Eqs. (5) and (6) are revised as

$$C_P \equiv (M_P)^e \bmod n, \tag{9}$$

and

$$M_P \equiv (C_P)^d \bmod n. \tag{10}$$

Considering the short length of the message $M_{\rm P}$, the index of the M-sequence can be encrypted and decrypted with very few arithmetic operations.

We evaluate our method with three frames of an animated sequence shown in Figs. 4(a)–(c). Each frame contains a first object "1" and a second object "2," both enclosed in a circle and located at 0.245 m (z_1) and 0.255 m (z_2) from the hologram, respectively. The wavelength λ of the light beam is 680 nm. A 2 048×2 048 hologram with a pixel size of 7 μm is generated based on Eq. (3). The parameter $z_{\rm o}$ in Eq. (1) is set to 0.25 m, which is halfway between z_1 and z_2. The three holograms are optically reconstructed with a 1 920 (horizontal) by 1 080 (vertical) liquid crystal on silicon (LCOS) display having a pixel size of 7 μm. To enable optical reconstruction on the LCOS that is only capable of displaying a real image, a reference planar wave at an illumination angle of 1.2° is added to the complex hologram. The real part of the result is extracted as an off-axis hologram to be displayed on the LCOS. Figures 5(a)–(c) show the optical reconstructed images of the unencrypted holograms. The small size of the LCOS causes a moderately prominent discrepancy in depth of the pair of objects "A" and "B". We adjust the focal length of the camera so that both objects are clearly displayed. The images are found to be successfully reconstructed, although the quality is not as good as the original images because of the imperfection of the LCOS display.

Subsequently, we apply Eq. (4) to generate the encrypted holograms for the three selected frames. For clarity and preservation of generality, we assume that the M-sequences for each frame are randomly selected from a small family of M-sequences generated with $M = 8$. A total of 16 different sequences are present in the family and they are indexed from 0 to 15. We select sequence numbers 3, 8, and 15 to encrypt the three hologram frames in our evaluation. Subsequently, the index of the M-sequence that is employed to encrypt each hologram frame is encrypted with the RSA algorithm and transmitted to the receiving end. In this process, the index of each sequence is firstly padded with the PKCS1-v_1.5 scheme[13] and encrypted with the RSA algorithm. For clarity and preservation of generality, we establish a test case with the set of parameters $\{p, q, n, e, d\}$ adopted in generating the public and private keys, as listed in Table 2.

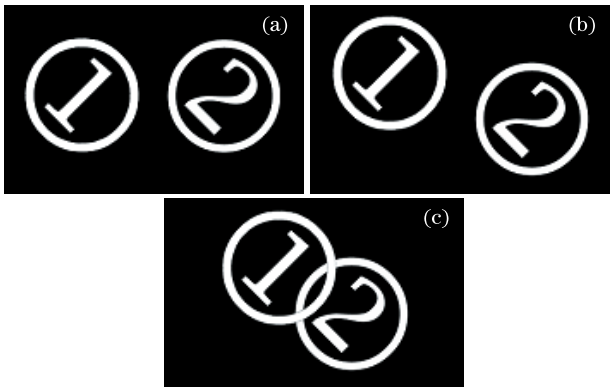

Fig. 4. The (a) first, (b) second, and (c) third frames of the video sequence.

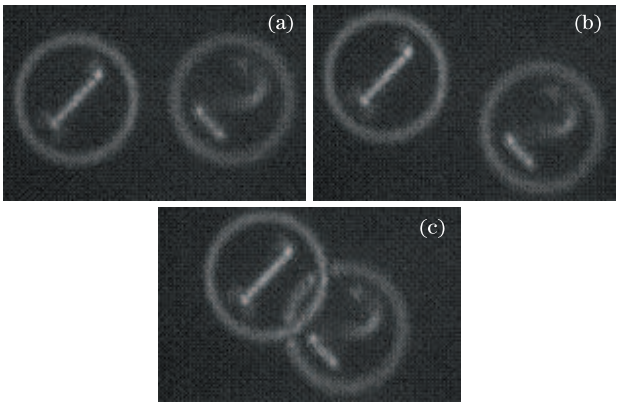

Fig. 5. Optical reconstructed images of the (a) first, (b) second, and (c) third hologram frames (without encryption).

Table 2. Parameters $\{p, q, n, e, d\}$ Expressed in the Radix-64 Format

First Prime Number p	0wzdF7nX2YjJvcK3ptWVUCWs+HBJtU271vbOyVZUyB0=
Second Prime Number q	4ar+QQ+7I97XuC54MLoYYkmZ7nt6igTNb/72iU3mobs=
Modulus $n = pq$	ugtGeRKx1d5lO7iwCrCUtTY/rIoXXIYG3p1ohDDOlBc+K NhpVJvXtOvDgp3aGEMVG5sJ3 AdhiShhobI+5NZqLw==
E	AQAB
D	AvJWxWgkZOzG0bFVKHYvhMc9LvBo+lAtH62CkEVtoC 7PJHj9oBYdzweUPu9tHtiu3 WffvcpyakcO1xZDj6pk0Q==

For simplicity, we only assign a single octet of random value for the padding string *PS*. However, the padding string can be easily extended to multiple octets. As an example, Table 3 shows the value of the index $ind = 03$ after padding and the encrypted results based on the public key (e, n).

When the encrypted ciphertexts are decrypted with the correct private key (d, n), the original padded indices are perfectly recovered. Each index is extracted by retaining only the last byte in the padded message. Figures 6(a)–(c) show the optical reconstructed images of the three encrypted holograms. The contents in the original images are heavily distorted. Suppose the private key is available to the receiver and the correct index is retrieved from the RSA decoder. Each encrypted holograms is decrypted with the correct M-sequence based on Eq. (5). Figures 7(a)–(c) show the results, which are almost identical to the images reconstructed with the unencrypted holograms.

We then study the case when the encrypted holograms are incorrectly decrypted with the M-sequence "4," which is different from the ones used in the encryption. This case simulates the scenario when the receiver does not have the private key and extracted a wrong M-sequence

Table 3. Padding and Encryption of the Indices Expressed in Hexadecimal Format

ind	*PS*	Padded Index	Encryption of the Padded Index C_P
03	30	0002300003	006658B5C945CD0F9DAD8528628 590ECD3403BDB7D1A37AB6E7AF D725C8E240E33B82BE9652F9EE EA66603E28D064B0B81F635F35 07AB06A6F5925489DD9E920

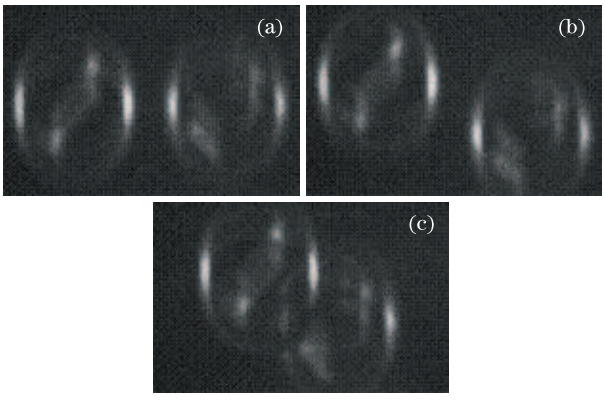

Fig. 6. Optical reconstructed images of the (a) first, (b) second, and (c) third encrypted hologram frames.

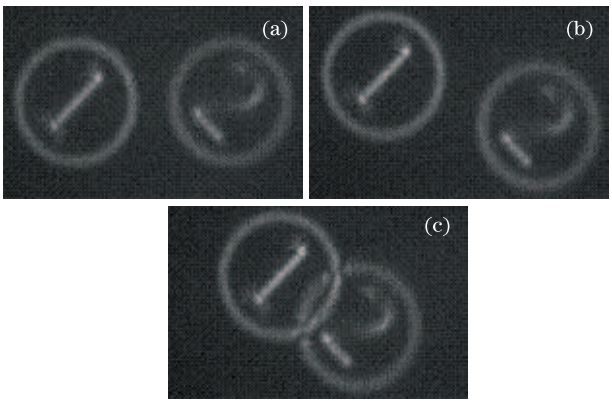

Fig. 7. Optical reconstructed images of the (a) first, (b) second, and (c) third encrypted and correctly decrypted hologram frames.

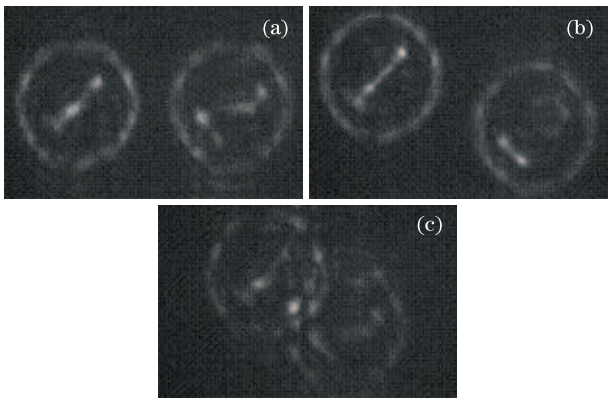

Fig. 8. Optical reconstructed images of the (a) first, (b) second, and (c) third encrypted and incorrectly decrypted hologram frames.

index. Figures 8(a)–(c) show the optical reconstructed images of the decrypted holograms. The reconstructed images are heavily distorted compared with the original images.

In conclusion, we propose a method for the fast generation and encryption of a holographic video sequence. Each frame of the encrypted hologram can only be correctly recovered with a randomly assigned M-sequence. The identity of the latter is represented by an index that is encrypted with the RSA algorithm and transmitted together with each encrypted hologram frame to the receiving end. Experimental results demonstrate that when the sequence of indices is retrieved with the correct private key, the hologram sequence is decrypted without any loss of information. Otherwise, heavy distortion is imposed on the reconstructed images. Therefore, our

method has two major advantages over existing encryption techniques (such as Refs. [1]–[7]). First, our encryption scheme is embedded as an integrated component of a fast, sub-line based hologram generation process and does not impose any additional computation load on the latter. Second, we incorporate the RSA scheme for further protecting the symmetric key (index of the M-sequence) from attacks through illegitimate means.

References

1. Y.-H. Seo, H. J. Choi, and D. W. Kim, Opt. Commun. **282,** 367 (2009).
2. K. B. Doh, K. Dobson, T.-C. Poon, and P. S. Chung, Appl. Opt. **48,** 134 (2009).
3. J. Li, T. Zheng, Q.-Z. Liu, and R. Li, Opt. Commun. **285,** 1704 (2012).
4. Q. Guo, Z. Liu, and S. Liucora, Opt. Commun. **284,** 3918 (2011).
5. H. Hwang, H. Chang, and W. Lie, Opt. Express **17,** 13700 (2009).
6. G. Situ and J. Zhang, Opt. Lett. **29,** 1584 (2004).
7. S. Kishk and B. Javidi, Appl. Opt. **41,** 5462 (2002).
8. P. Tsang, W. Cheung, T. Poon, and C. Zhou, Opt. Express **19,** 15205 (2011).
9. T. Shimobaba, H. Nakayama, N. Masuda, and T. Ito, Opt. Express **19,** 19504 (2010).
10. P. W. M. Tsang, T.-C. Poon, and K. W. K. Cheung, Appl. Opt. **50,** B46 (2011).
11. S. W. Golomb, *Shift Register Sequences* (Aegean Park Press, California, 1981).
12. R. L. Rivest, A. Shamir, and L. Adleman, Communication of the ACM **21,** 120 (1978).
13. "RFC 3447: Public-Key Cryptography Standards (PKCS)", http://rfc-ref.org/RFC-TEXTS/3447/chapter7.html#d4e442871.