A New Hybrid Asymmetric Key-Exchange and Visual Cryptographic Algorithm for Securing Digital Images

Quist-Aphetsi Kester
Lecturer, Faculty of Informatics
Ghana Technology University College Accra, Ghana
Email: kquist-aphetsi@gtuc.edu.gh / kquist@ieee.org

Laurent Nana
Lab-STICC (UMR CNRS 6285)
European University of Brittany
University of Brest
20 avenue Victor Le Gorgeu, BP 817 - CS 93837, 29238 Brest cedex, France
Laurent.Nana@univ-brest.fr

Anca Christine Pascu
Lab-STICC and HCTI
European University of Brittany
University of Brest
CS 93837, 29238 Brest cedex, France
Anca.Pascu@univ-brest.fr

***Abstract*—In today's cyber world where interception of data by third parties is very common, there is a demanding need for encryption of data via secured and unsecured communication networks. Sensitive digital images stored in databases, such as in the cloud, need to be encrypted.**

This paper proposed a hybrid method of encryption of digital images based on asymmetric encryption algorithm and a visual cryptographic algorithm. The encryption key used was based on a public key-exchange algorithm and the algorithm was implemented using MATLAB on an mxn image size.

Keywords: Asymmetric, Cryptography, Encryption, Algorithm, Digital image

I. INTRODUCTION

The use of digital media over the internet and network has gained tremendous growth over the years, ranging from social networks, personal websites, cloud storage systems, etc. and has emphasized the need for protection of these media such as images, audio clips, videos and etc [1].

The rising issues of concern in the current information age are prevention of access to Information Systems by backdoor means, securing of communications such as chat messages on social networks, sms, e-mails etc., and security management of information infrastructure and development of secure Information Systems in distributed networks and data centric environments. These key areas have seen major advancement over the past years [2]. Cryptographic algorithms have provided security for transmitted and stored information. Most of the cryptographic encryption algorithms used for the storage of large volumes of data is symmetric cipher encryption methods and in recent years, a number of standardized symmetric encryption schemes have fallen foul of attacks [3].

Asymmetric encryption is a method where a message encrypted with a recipient's public key cannot be decrypted by anyone except a possessor of the matching private key, presumably, this will be the owner of that key and the person associated with the public key used. This is used for confidentiality. [4] This paper engaged asymmetric or public key encryption algorithm and visual cryptographic encryption method in the encryption of digital images in such a way that only the one authorized to view the image can access and decrypt the content successfully.

The paper has the following structure, section II: Methodology, section III: summary of key-exchange and encryption process, section IV: mathematical algorithm, section V: results and analysis, and section VI: conclusion.

II. METHODOLOGY

In this paper, we produced a hybrid digital encryption algorithm based public key and visual cryptography. The ciphering of the plain image was done using the image encryption algorithm but dependent on the keys engaged.

The public-key was deduced based on a randomly chosen private key of n length with the engagement of a forward hash function algorithm [4]. For the image to be encrypted and decrypted, the two or more parties involved have to chose a random private key and generate a public key from it. They can therefore interchange the public keys in order to communicate. The algorithm that generated the public key was a forward hash function and that made it difficult for an adversary to easily generate the private key back.

The exchanged keys were used to encrypt the image. The algorithm depended on a shared secret key (which was generated from the combination of a public key and a private key). The following security requirements in cryptography were met at the end of the work.

- Authentication: The process of proving one's identity and ensuring that only the authorized person can decrypt the image and have access to it.
- Privacy/confidentiality: Ensuring that no one can read the message except the intended receiver.
- Integrity: Assuring the receiver that the received message has not been altered in any way from the original and any alteration during transmission will render the image data useless.
- Non-repudiation: A mechanism to prove that the sender really sent this message with the engagement

978-1-4799-3067-8/13/$31.00 ©2013 IEEE

of the sender's public key and the recipient's private key can confirm the sender.

III. SUMMARY OF KEY-EXCHANGE, ENCRYPTION AND DECRYPTION PROCESSES

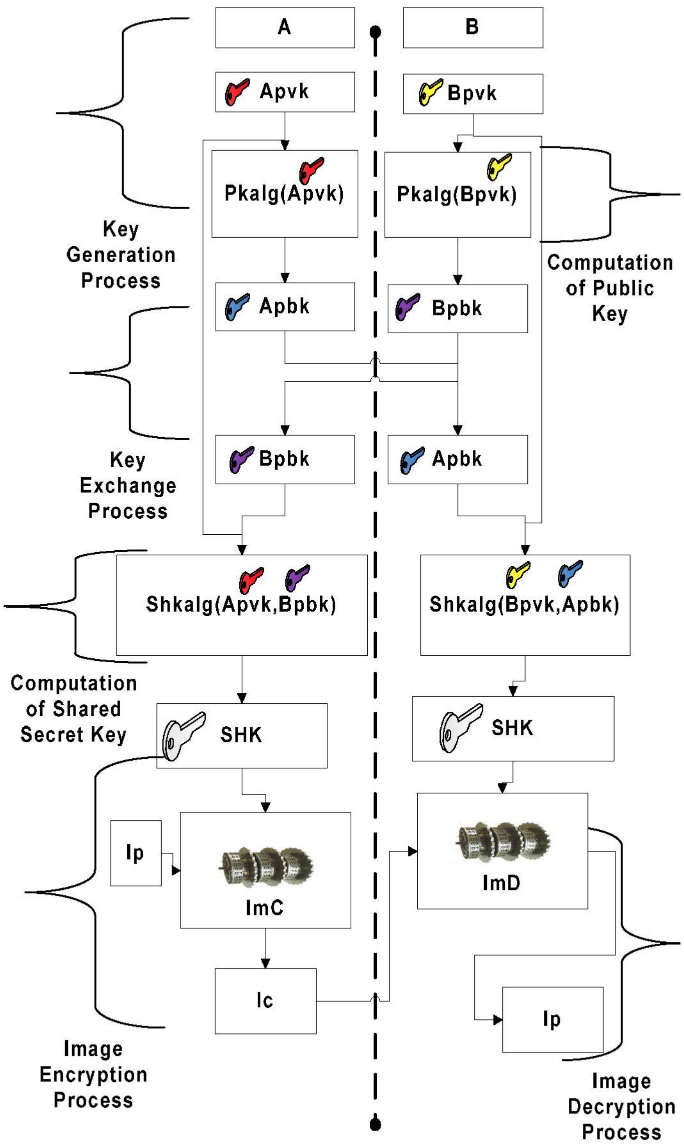

Figure 1: The key-exchange, encryption and the decryption process

From figure 1, we have the definitions of the statements in each box as follows:

A = the first party being engaged in a key exchange process with *B*.

B= the second party being engaged in a key exchange process with *A*.

Apvk = the randomly chosen private key by party *A*.
Bpvk = the randomly chosen private key by party *B*.
Pkalg(Apvk)= the function *Pkalg()* that operates on *Apvk* to produce *Apkb*.

Pkalg(Bpvk)= the function *Pkalg()* that operates on *Bpvk* to produce *Bpkb*.

Apkb=the public key of party *A*.

Bpkb=the public key of party *B*.

Shkalg(Apvk, Bpkb) = the function *Shkalg()* that operates on *Apvk* and *Bpkb* to produce *SHK*.

Shkalg(Bpvk, Apkb) = the function *Shkalg()* that operates on *Bpvk* and *Apkb* to produce *SHK*.

SHK = the share secret key for both party *A* and *B*.

Ip=the plain Image

Ic=the cipered image

Imc = the algorithm for encryption of the plain image.

ImD = the algorithm for the image decryption.

In the encryption and the decryption processes, the images used had no pixel expansion. The ciphering of the images for this research was dependent solely on the private and public keys engaged.

IV. THE MATHEMATHECAL EXPLANATION

A. The Key-Exchange Algorithm

For a common communication to be established between two parties *A* and *B* in public key cryptography, there have to be a common shared secret key [5], *SHK*.

Let *Apvk* and *Bpvk* be the randomly chosen private key of party *A* and *B* respectively, Private keys are random number less than n, where n is a domain parameter.

$$Apvk = X{:}x \epsilon X \ \ and\ x{:}x \epsilon I \wedge 0<x<+\infty$$

$$Bpvk = X{:}x \epsilon X \ \ and\ x{:}x \epsilon I \wedge 0<x<+\infty$$

Let $Apkb = f(Apvk)$
$= Pkalg(Bpvk)$ be the public key of party *A*

Let $Bpkb = f(Bpvk)$
$= Pkalg(Bpvk)$ be the public key of party *B*

A and *B* exchanged their public keys
A computes SHK_A

$SHK_A = f(Apvk, Bpkb)$
$\quad = Shkalg(Apvk, Bpkb)$

B computes SHK_B

$SHK_B = f(Bpvk, Apkb)$
$\quad = Shkalg(Bpvk, Apkb)$

The shared secret key, *SHK,* for A, $=SHK_A$ *and B,* $= SHK_B$, and, if $SHK_A = SHK_B$, then there can be communication between A and B successfully.

Let $ki \in k = [k0, k1, k2, k3 \ldots kn]$

Where $0<n<+\infty$ which determines the key length

$Pkalg(k) = (ki + \beta) \bmod u$

Where *u* is a randomly chosen and accepted key by both parties and implemented by the algorithm to produce the public.

And β *is a constant:* $\beta = x\text{: } x \in I \wedge 0<x<+\infty$

$Shkalg(n, p) = Shkalg(Bpvk, Apkb)$
$\quad = (n + p) \bmod u$

The above operations were used for the key exchange process.

B. The Encryption Algorithm with the engangement of the Shared secret Key.

Step 1. Start

Step 2. Input the plain image

Step 3. Compute the shared key

Step 4. Import data from image and create an image graphics object by interpreting each element in a matrix.

Step 5. Get the size of r as [c, p]
Where c, p represents the width and heath values of the image

Step 6. Repeat steps 7 to 17 using the shared secret key value

Step 7. Engage the key for each shift of share of the plain image

Step 8. Remove the red component as a share, ‘r’

Step 9. Remove the green component as a share, ‘g’.

Step 10. Remove the red component as a share, ‘b’.

Step 11. Let r =Transpose of r

Step 12. Let g =Transpose of g

Step 13. Let b =Transpose of b

Step 14. Reshape r into (r, c, p)

Step 15. Reshape g into (g, c, p)

Step 16. Reshape b into (b, c, p)

Step 17. Concatenate the arrays r, g, b into the same dimension of ‘r’ or ‘g’ or ‘b’ of the original image to obtain the ciphered image.

Step 18. Convert the data into an image format to get the ciphered image.

The inverse of the encryption algorithm will decrypt the encrypted image back into the plain image.

V. THE RESULTS AND ANALYSIS

The keys were analyzed to guess the possibility and the ways that a chosen key length of a private key out of a length of characters can be analyzed.
Engaging combination and permutation,
Let n= the total number of distinct characters.
Let r= the total number of characters chosen as a key from n.
From a US QWERTY keyboard, we have n=107 characters and the key length used was r=20 and r=40.
Where n= { a, b, c, d, e, f, g, h, i, j, k, l, m, n, o, p, q, r, s, t, u, v, w, x, y, z,:,K,),&,%,#,@,!,|,\...........} and for r={a,j,d,2,3,4,5,0,-,l ,b, c,<,>,?,”,;,H,D,t}.
Applying combinations,

$$C(n,r) = \frac{P(n,r)}{r!} = \frac{n!}{r!(n-r)!}$$

Where the order of arrangement is not important and repetition is not allowed,

C (n, r) = C (107, 20) = 2.39182547e+21.

The random private key chosen was based on a condition where the repetition of characters were allowed and the order of arrangement was important for (n=107, r=20).
C (n, r) = C (94, 8)
= 3.86968446e+40.
Hence the probability of getting the right result that is the private key for n=107 and r=20 is 1/ 3.86968446e+40 = 2.5841900297989671230196376269914e-41.

Let Ψ= permutation where the repetition was allowed and the order of arrangement was important for (n=107, r=1≤1x≤n).
Let η= the probability for which a private key was obtained from Ψ for (n=107 r=1≤1x≤n).

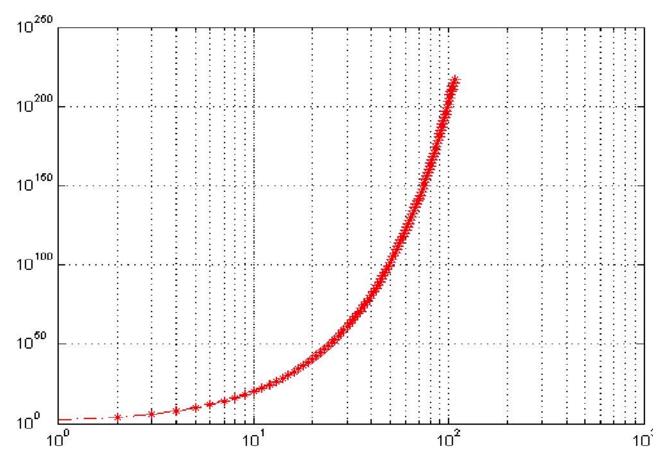

Figure 2: A log-log graph of Ψ versus r

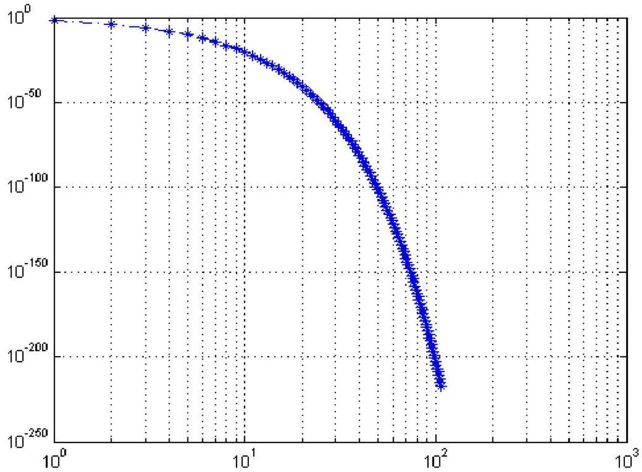

Figure 3: A log-log graph of η versus r

From figure 2 which is a log-log graph of Ψ versus r, it can be observed that as the key length increases the permutation also increases sharply and this increases the number ways one can compute chosen keys based on the characters chosen. The probability of guessing a key at random is represented by figure 3 which is a log-log graph of η versus r. As the keys become longer the probability also approaches more to zero.

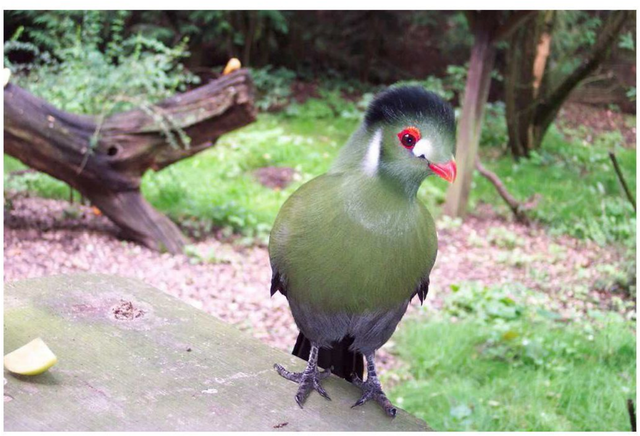

Figure 4: A plain image of a bird

The image [6] in figure 4 is a 1024x683 pixel image used in the implementation process of the algorithm in MATLAB. The image was ciphered using a 640 bits character length and a 1280 bits character length shared key as seen in figure 6 and 8 respectively. The respective graph of the ciphered image for image 4 for both the 640 bits character key and 1280 bits character key are represented by figure 7 and figure 9 respectfully. The plain image's graph was plotted as figure 5.

All the graphs were plotted using the first 10000 pixel values of the image.

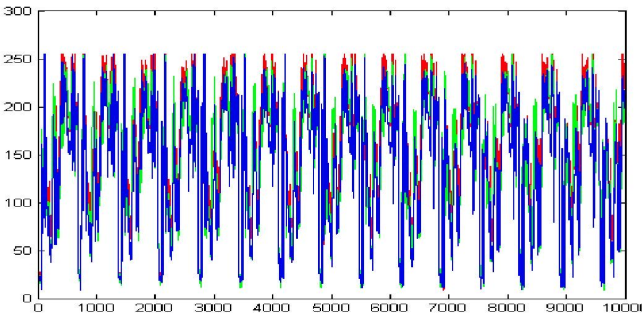

Figure 5: An RGB graph of the plain image of figure 4

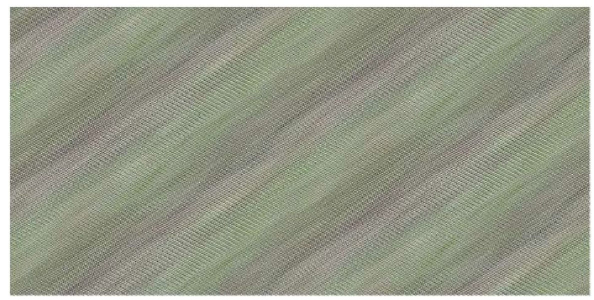

Figure 6: An RGB graph of the ciphered image of figure 4 based on 640 bits secret key

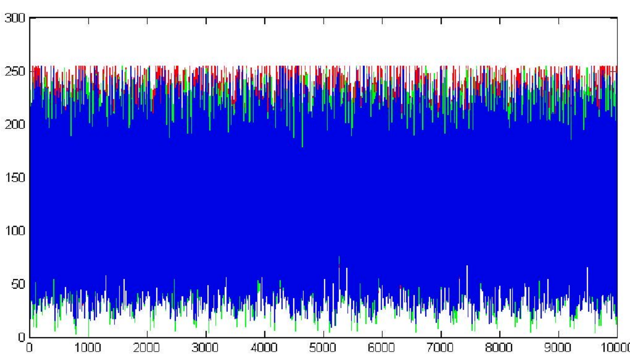

Figure 7: An RGB graph of the ciphered image of figure 6

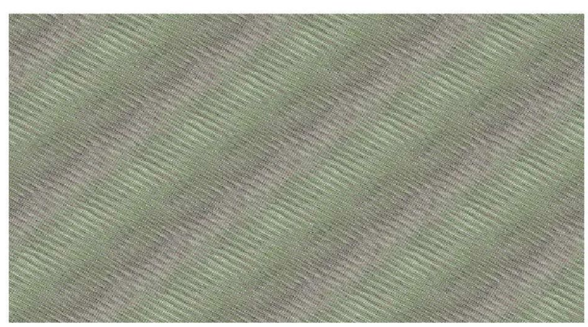

Figure 8: An RGB graph of the ciphered image of figure 4 based on 1280 bits secret key

978-1-4799-3067-8/13/$31.00 ©2013 IEEE

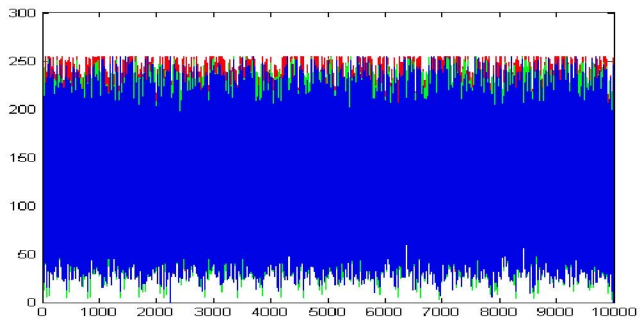

Figure 9: An RGB graph of the ciphered image of figure 8

VI. CONCLUSION

A hybrid cryptographic algorithm has been proposed in this paper and the implementation was done using MATLAB. From the work, it was observed that the effectiveness and the robustness of the ciphering process is dependent on the length of the shared secret key and the computer resources available such as processing power and speed. And due to the public-key engagement in the ciphering of the image, there has been effective implementation of security requirements into visual cryptography such as confidentiality, integrity, authentication and non-repudiation.

ACKNOWLEDGEMENTS

This work is part of a PhD research work supported by Lab-STICC (UMR CNRS 6285) at UBO France, AWBC Canada and the DCSIT, UCC.

REFERENCES

[1] Dehkordi, A. B., Esfahani, S. N., & Avanaki, A. N. (2011, May). Robust LSB watermarking optimized for local structural similarity. In Electrical Engineering (ICEE), 2011 19th Iranian Conference on (pp. 1-6). IEEE.

[2] Mikko T. Siponen , Harri Oinas-Kukkonen, A review of information security issues and respective research contributions, ACM SIGMIS Database, v.38 n.1, February 2007 [doi>10.1145/1216218.1216224]

[3] Boldyreva, A., Degabriele, J. P., Paterson, K. G., & Stam, M. (2012). Security of symmetric encryption in the presence of ciphertext fragmentation. In Advances in Cryptology–EUROCRYPT 2012 (pp. 682-699). Springer Berlin Heidelberg.

[4] Kester, Q.-A.; Danquah, P., "A novel cryptographic key technique," Adaptive Science & Technology (ICAST), 2012 IEEE 4th International Conference on, vol., no., pp.70,73, 25-27 Oct. 2012 doi: 10.1109/ICASTech.2012.6381068.

[5] Diffie, W.; Hellman, M.E., "New directions in cryptography," Information Theory, IEEE Transactions on , vol.22, no.6, pp.644,654, Nov 1976 doi: 10.1109/TIT.1976.1055638.

[6] Squidoo. Ideas for Squidoo Lenses. Free Images. Retrieved from : http://www.squidoo.com/free_images