Second International Congress on Technology, Communication and Knowledge (ICTCK 2015) November, 11-12, 2015 - Mashhad Branch, Islamic Azad University, Mashhad, Iran

Analysis of an Asymmetric Cryptosystem based on Boolean Matrices and Permutations

Mina Nejati
Department of Information Technology,
Islamic Azad University,
Mashhad Branch
Mashhad, Iran
mina.nejatiii@gmail.com

Maryam Kargozar
Department of Information Technology,
Islamic Azad University,
Mashhad Branch
Mashhad, Iran
marykargozar@gmail.com

Omid Mirzaei
Computer Security Lab (COSEC)
Universidad Carlos III de Madrid
Madrid, Spain
omid.mirzaei@uc3m.es

Vahid Chahkandi
Department of Computer Engineering,
Islamic Azad University,
Mashhad Branch,
Mashhad, Iran
chahkandi.vahid@gmail.com

***Abstract*—With widespread use of internet based services and applications, there is a real need for ensuring the reliability and security of data transmitted over this platform. One of the major tools to guarantee the security of information is data encryption. However, the proposed encryption methodology is usually expected to have some key characteristics to be applicable in computer networks. A suitable encryption algorithm should be fast, reliable, and, more importantly, it should impose less meta-data to the original data. Aiming at these characteristics, an asymmetric encryption system has been proposed recently based on Boolean product of matrices. This cryptosystem is analyzed in this paper from another point of view. Here, it is shown that Boolean information is really sensitive, and a minor change in binary data might lead to a significant change in the retrieved values through the decryption procedure. This issue becomes even more critical when the application area is a computer network in which numerous incidents may happen to data before they reach the desired destination; for instance, different kinds of noises are capable of changing the originality of information. Simulations and experiments conducted in this paper show that this fast cryptosystem is not a suitable tool to ensure the security of data in computer networks.**

KEYWORDS: Asymmetric Encryption, Boolean Matrices, Boolean Permutations

I. INTRODUCTION

Nowadays, Internet is being used by many organizations and corporations to offer a wide variety of services to their customers. It means that many information are transferred by this platform daily. Therefore, this huge amount of data transmission should be kept safe and secure. One of the main ways of ensuring the reliability and safety of information is data encryption [1]. Many encryption algorithms have been proposed up to now for numerous application areas. However, a few of them have been developed for a specific purpose such as data encryption in computer networks.

Cryptographic algorithms can be classified into two main groups, including symmetric and asymmetric [2]. In symmetric cryptography, the sender and receiver of the message share a common key for both encryption and decryption procedures [2]. This key should be shared through secret communication; otherwise, a third party can easily decrypt the message. On the other hand, in asymmetric cryptography, also known as public key cryptography, there are two types of keys which involved in the process of encryption and decryption [3]. Here, the receiver of the message sends a public key to the sender through any communication channel, either safe or non-safe. Having this key, the sender would be able to encrypt the message and send it to the receiver. Then, the receiver can decrypt the message in the destination using a private key [3]. Nobody else could be able to decrypt this message without having this private key.

Cryptosystems which are developed to encrypt and decrypt data in computer networks are usually expected to have two important characteristics except being capable of resisting different kinds of attacks. Firstly, they are expected to be fast, i.e. they should have the least possible delay in the process of encryption since many applications need to send or receive data in the least possible time, or, often, in real-time [4]. Secondly, and, more importantly, they are expected to be lightweight [4], i.e. they should impose the least possible amount of meta-data on the original information which are going to be encrypted. This characteristic is really vital, because engineers are always seeking for ways to decrease the size of data transmitted over computer networks. However, it needs to be mentioned that developing a secure and strong cryptosystem leads to an increase in the computations, and, therefore, is in contradiction with the complexity, computational burden, and the speed of encryption and decryption algorithms. Consequently, a trade-off needs to be considered [5].

978-1-4673-9762-9/15/$31.00 ©2015 IEEE

Recently, a fast asymmetric cryptosystem has been introduced in [6] based on Boolean product of matrices. This encryption algorithm is fast due to use of Boolean operations in its calculations. Moreover, it is secure because of the stability of its public key. This stability has been achieved as a result of difficulty which exists in inverting Boolean product of large matrices [6]. It is also a lightweight system that impose a very low amount of additional information to the original data. However, despite this advantages, it has not been tested on an application area such as computer networks in which binary data are very probable to change before they reach their target. Binary information are really sensitive to changes, i.e. even a change to one bit may lead to a significant change in the magnitude of original data.

The abovementioned encryption system is analyzed in this paper from another point of view. Here, several experiments are conducted to test the efficiency of the proposed algorithm in the presence of noises which are a normal incident in computer networks. Doing so, the robustness of this cryptosystem is evaluated through numerous simulations and its weak points are also highlighted.

The remainder of this paper is organized as follows. Section II provides a brief yet clear explanation of the proposed encryption and decryption system in [6]. This cryptosystem is then evaluated and discussed in Section III. Finally, Section IV concludes the paper and presents some suggestions as future works.

II. Description of Encryption and Decryption Procedures

In the proposed cryptosystem in [6], the plaintext P and the ciphertext C have assumed to be integers ranging from 0 to 2^{n-1}; however, in our paper, they have assumed to be gray scale images containing pixels with decimal values in the same range for the sake of simplicity. These integer values which are between 0 and 2^{n-1} can be expressed as a binary data of n bits.

A. Data Encryption

Considering this preliminary assumption, the overall encryption algorithm can be summarized and described in several steps as follows:

1) *Converting the decimal plaintext values to binary values:*

In this step, the initial plain-image ($P_{M\times N}$) pixels are converted from decimal to their equivalent binary values. This process has been demonstrated in Fig. 1 to provide a better comprehension. Considering this figure, P is the original plain-image and P^C is the converted version of the plain-image containing binary values. Such a matrix with entries that are either 0 or 1 is known to be a Boolean matrix [7].

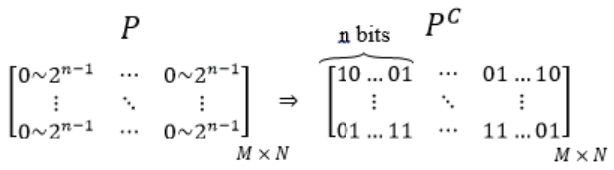

Fig. 1. Conversion of a gray-scale image containing decimal values to the same image containing equivalent binary values

2) *Creating A Boolean Permutation:*

A Boolean permutation is actually a collection of Boolean functions of n variables as below [8]:

$$BP = [f_1(x), f_2(x), \ldots, f_n(x)] \quad (1)$$

A Boolean function of n variables is a mapping from an n-dimentional vector space over the binary field $F_2 = \{0,1\}$ to itself ($F_2 \to F_2$) [8]. Each of the above functions in Eq. 1 is considered as a machine with one bit output and n-bit input and can be written as follows:

$$f_i(x) = c_0 \oplus c_1x_1 \oplus c_2x_2 \oplus \ldots \oplus c_nx_n, \quad i = 1,2,..,n \quad (2)$$

Where $x = (x_1, x_2, \ldots, x_n)$ is the set of all variables. The above equation is actually the sum of all min-terms which can also be expressed in the form of a truth table. A Boolean function is said to be balanced if there are equal number of zeros and ones in its truth table. Moreover, it is called a linear function if $c_0 = 0$.

Considering the above preliminary explanations, a Boolean permutation of size $n \times k$ is constructed where n stands for different min-terms of input variables, and k is the number of Boolean functions. In [6], the Boolean permutation is $BP = [f_1(x), f_2(x), f_3(x)]$ which has been constructed over three variables $x = (x_1, x_2, x_3)$ as demonstrated in Table I. A Boolean permutation can be directly used to design a public key cryptosystem if the number of functions within the permutation is reasonably small [8]. Therefore, three input variables constituting eight min-terms ($n = 8$), and also three Boolean functions ($k = 3$) have been considered in [6] for encryption and decryption procedures.

Table I. The Boolean permutation consisting of three Boolean functions and three input variables

x_1, x_2, x_3	$f_1(x)$	$f_2(x)$	$f_3(x)$
000	0	0	0
001	1	0	1
010	0	1	1
011	1	1	1
100	1	1	0
101	1	1	1
110	1	1	1
111	1	1	1

After the creation of the Boolean permutation, all binary values of the P^C matrix are converted to another stream of binary values using the above look-up table as illustrated in Fig. 2.

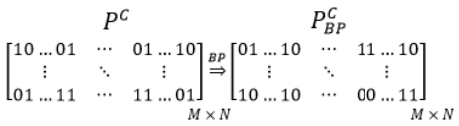

Fig. 2. Conversion of the P^Cmatrix using the Boolean permutation BP

Where P^C_{BP} is a new matrix resulted from converting binary values of P^C using the look up table presented in Table I, called BP.

3) *Creating BM_1, BM_2, and BM_3 martrices*

In the next step of the encryption procedure, three matrices called BM_1, BM_2, and BM_3 are constructed. BM_1 is a Boolean matrix which is actually the transpose of the three Boolean functions presented in Table I and is of size 3×8 $(k \times n)$. Fig. 3 demonstrates this process much more clearly.

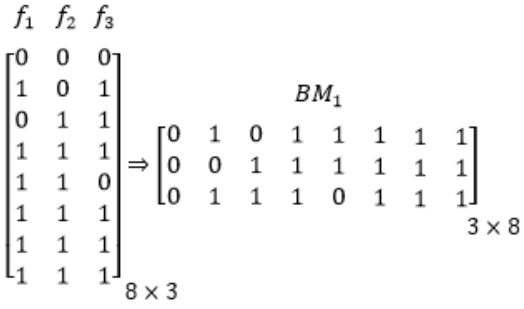

Fig. 3. The process of creating BM_1 matrix

In addition, BM_2 and BM_3 matrices are of sizes 8×3 $(n \times k)$ and 5×8 $((n-k) \times n)$ respectively and are obtained using the following properties [6] Where I_k is the identity matrix of size 3×3 $(k \times k)$.

$$BM_1 \times BM_2 = I_k \tag{3}$$

$$BM_3 \times BM_2 = 0 \tag{4}$$

However, a more convenient way to create BM_1, BM_2, and BM_3 matrices is to choose an arbitrary nonsingular matrix C of size 8×8 $(n \times n)$ such that its first k rows are BM_1. The remaining $n-k$ rows represent BM_3, and BM_2 is constructed by the first left k columns of C^{-1}, that is the inverse of C. The process of extracting these three matrices has been presented in Fig. 4 to provide a better imagination.

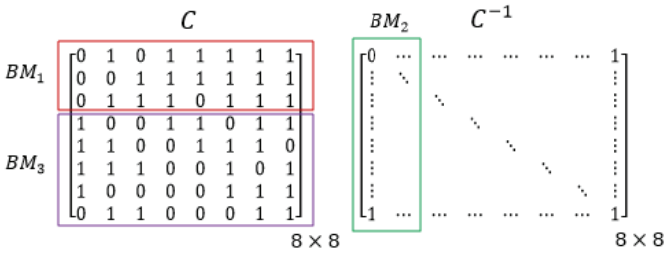

Fig. 4. The process of extracting BM_1,BM_2, and BM_3 matrices from the nonsingular matrices C and C^{-1}

4) *Creating the Public Key for Encryption*

The final step of the encryption procedure leads to the creation of a public key for the purpose of encryption. Here, in this algorithm, four variables (matrices) are used to construct the public key as demonstrated in Eq. 5. Three of these matrices were specified before which are: P^C_{BP}, the result of applying the Boolean permutation on the Boolean plain-image, BM_1, and BM_3. However, the last matrix is another arbitrary collection of Boolean functions, a Boolean permutation, $R = [r_1, r_2, \dots, r_{n-k}]$.

$$P_E = (P^C_{BP} \times BM_1) \oplus (R \times BM_3) \tag{5}$$

The result of the above equation is indeed the ciphered image and can be shown symbolically as below where P is the initial plain-image, and C is the result of applying this encryption procedure with the above public key on the plain-image.

$$C = E_{P_E}(P) \tag{6}$$

B. Data Decryption

As any encrypted data in a source needs to be decrypted in the destination considering a computer network, the process of data decryption presented at [6] is also described as follows:

1) *Creating BP^{-1}*

To retrieve the original data, the inverse of the Boolean permutation, BP, discussed in the encryption phase, is obtained initially as depicted in the Table II. It is clear from this table that all the initial min-terms can be precisely reproduced using the inverse Boolean permutation, BP^{-1}.

Considering Table II, an important feature of this inverse permutation is revealed as it can have similar output values which are produced from different input min-terms. For instance, the Boolean permutation, BP, has "111" output for the min-terms "011", "101", and "110". This circumstance makes difficult the regeneration of input values from output values using the inverse permutation, BP^{-1}. This feature and such a circumstance discovers a trapdoor one-way function which is the basis for public-key or asymmetric cryptography [3].

TABLE II. THE INVERSE OF THE BOOLEAN PERMUTATION, BP, DEMONSTRATED BY BP^{-1}

$f_1(x), f_2(x), f_3(x)$	x_1	x_2	x_3

000	0	0	0
101	0	0	1
011	0	1	0
111	0	1	1
110	1	0	0
111	1	0	1
111	1	1	0
111	1	1	1

A function f from a set X to a set Y is called a one-way function if $f(x)$ is "easy" to compute for all $x \in X$ but for "essentially all" elements $y \in Im(f)$ it is "computationally infeasible" to find any $x \in X$ such that $f(x) = y$ [3]. A trapdoor one-way function is also a one-way function with the additional property that given some extra information (called trapdoor information) it becomes feasible to find for any given $y \in Im(f)$, an $x \in X$ such that $f(x) = y$. This function has been illustrated in Fig. 5 to provide a clear imagination.

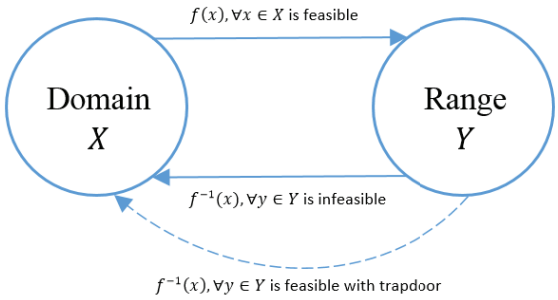

Fig. 5. A trapdoor one-way function

2) *Creating the private key for decryption*

As mentioned before, in asymmetric cryptosystems, the private key is constructed and held in the destination to decrypt the messages once they are received. Therefore, in the proposed system, the private key is created using two main matrices. These are BM_2 which was created before, and Z which is the inverse of the Boolean permutation R constructed in the encryption process. Next, the original data is retrieved using the inverse of the Boolean permutation, BP, as demonstrated in the following equations.

$$P_D = BP^{-1}(Z \times BM_2) \tag{7}$$

$$P = D_{P_D}(C) \tag{8}$$

Where P is the initial plain-image, P_D is the constructed private key, and C is the ciphered-image.

III. Analysis of the Proposed Encryption System

To test the robustness of the described cryptosystem in an environment similar to a computer network, the presented system was implemented initially in MATLAB2009 software on a system with 2.53 GHz CPU, and 3GB RAM. Then, the simulations were all been applied to two well-known samples of USC database, Lena (or Lenna) and Cameraman, with their specifications provided in Fig. 6. It is also worthy to say that all the results presented in this section were obtained through 1000 of executions.

a

b

c

No.	Discription	Size	Color
a	Lena (or Lenna)	300*300	Gray
b	Cameraman	256*256	Gray

Fig. 6. (a) Lena (or Lenna) image sample, (b) Cameraman image sample (c) Specifications of Lena and Cameraman image samples

As described in details previously, the encryption and decryption procedures in [6] are performed using public and private keys constructed by look-up tables which are indeed Boolean permutations. Moreover, it was explained that public and private keys are both created at the destination in an asymmetric cryptosystem, and, then, the public key is passed to the source for the encryption procedure. However, the private key remains at the destination for the decryption process in future. The main issue is that it is very probable for the public key to be affected by different kinds of noises if it is passed on an unsafe channel over a computer network. This may cause the decryption system not to be able to retrieve the original image with the same quality using an inverse Boolean permutation of the original Boolean permutation which was used to create the public key for the encryption procedure. Considering this crucial issue, and in order to test the robustness of the presented algorithm, two separate experiments have been conducted.

In the first experiment, different probabilities of noise varying from 0 to 0.5 have been applied to image samples. Then, the similarity of the original image and the decrypted one at destination has been measured by defining a parameter called "Similarity Percentage" which is obtained through dividing their pixel values. The results of the first experiment have been shown in Fig. 7.

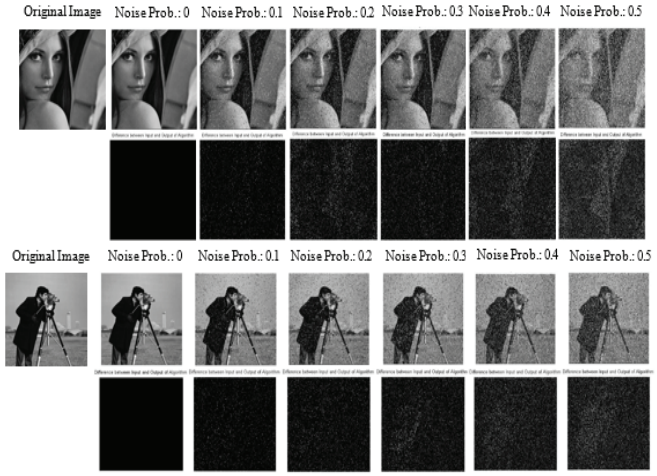

Fig. 7. The difference between the original image and the retrieved image at destination in two image samples. In each sample, the first row demonstrates the retrieved image, while the second row illustrates the difference between the retrieved image and the original one in pixel values

As it is clear, the decrypted image is affected by noise and it has not the same quality of the original one. In other words, as the probability of noise increases from left to right, the difference between the quality of the decrypted image and the original image increases as well. A dark image in the second row of this figure shows a very close similarity between these two images, while a light image shows a very far similarity between them. To provide a precise and clear imagination of the impact of noise on this encryption-decryption algorithm, the results have also been presented numerically in Table III and Table IV.

TABLE III. SIMILARITY PERCENTAGE OF THE RETRIEVED LENA IMAGE TO THE ORIGINAL IMAGE ON THE PRESENCE OF DIFFERENT NOISE PROBABILITIES

Lena	Noise Prob.	Similarity Percentage
	0	100%
	0.1	93.32%
	0.2	82.15%
	0.3	81.42%
	0.4	75.84%
	0.5	55.86%

TABLE TABLE IV. SIMILARITY PERCENTAGE OF THE RETRIEVED CAMERAMAN IMAGE TO THE ORIGINAL IMAGE ON THE PRESENCE OF DIFFERENT NOISE PROBABILITIES

Cameraman	Noise Prob.	Similarity Percentage
	0	100%
	0.1	93.48%
	0.2	88.94%
	0.3	83.94%
	0.4	76.83%
	0.5	71.91%

In the second experiment, the robustness of the algorithm has been investigated by defining another parameter called "Failure Percentage". It is actually the number of trials in which the original image has not been retrieved through the decryption procedure being affected by different noise probabilities. The results of the two image samples have been gathered in Table V and Table VI.

TABLE V. FAILURE PERCENTAGE OF THE PROPOSED CRYPTOSYSTEM FOR LENA SAMPLE UNDER DIFFERENT NOISE FREQUENCIES

Lena	Noise Prob.	Failure Percentage
	0.1	91.87%
	0.2	93.14%
	0.3	94.67%
	0.4	96.04%
	0.5	97.39%

TABLE VI. FAILURE PERCENTAGE OF THE PROPOSED CRYPTOSYSTEM FOR CAMERAMAN SAMPLE UNDER DIFFERENT NOISE FREQUENCIES

Cameraman	Noise Prob.	Failure Percentage
	0.1	89.64%
	0.2	91.74%
	0.3	92.51%
	0.4	94.81%
	0.5	95.92%

As it is obvious, the failure percentage of the proposed algorithm is significantly high when it is effected by a small amount of noise, and it becomes even higher as the frequency of noise increases. The main reason is that a Boolean permutation is used to create the public key for the encryption procedure and its inverse matrix is used for decryption. Therefore, when the public key is affected by noise, the decryption of the image becomes infeasible due to the production of new Boolean permutations.

IV. CONCLUSIONS AND FUTURE WORKS

Data security is becoming significantly important in internet-based services and platforms due to the high amount of uncertainty and vulnerability exist in such environments [9]. One of the most recent emerging global internet-based information architectures is Internet-of-Things (IoT) [10]. A semantical definition of IoT is a "world-wide" network of interconnected objects uniquely addressable, based on standard communication protocols [11]. The diversity of scenarios where internetworked entities have to exchange information with each other without human interaction is increasing and is planned to extend to almost all environments, from individual customers' everyday life to industrial processes. Accordingly, more and more objects become able to communicate [12].

In such an environment, there is an undeniable need for information security which can be achieved through data encryption. For this purpose, some researches have been conducted to propose cryptosystems developed explicitly for IoT [13][14][15]; However, designing suitable cryptosystems with specific characteristics mentioned before still needs a lot of attention. Among different developed cryptosystems, the encryption-decryption system presented in [6] has the potentials to be applied to such a context due to its important advantages such as its high speed and its low computational burden. Moreover, it is strong as has been shown by conducting numerous attack scenarios [6].

However, an important issue still needs to be considered seriously regarding this cryptosystem. The problem is that the encryption and decryption procedures are mainly performed using look-up tables which are indeed Boolean permutations. The main concern that has not been addressed in [6] is related to the retrieval of original data when part of the public key which is actually constructed by Boolean permutations is affected by noises. As public and private keys are both created initially at destination with the use of Boolean permutation and inverse Boolean permutation respectively, and, since public key which contains the Boolean permutation is passed to the source for data encryption might be affected by noise, it is probable for the decryption system not to be able to retrieve the original data.

Here, in this paper, some experiments have been conducted to investigate the effect of noises on the proposed cryptosystem in [6]. The simulations are mainly related to decryption procedure. Since a cryptosystem would not be valid without having an easy and trustable decryption procedure, the analysis performed in this paper can reveal the validity or robustness of the suggested cryptosystem. According to the conducted simulations, the developed cryptosystem is prone to mal-functions in computer networks since numerous degrees or probabilities of noise may cause this system not to be able to retrieve the original data using the previously built look-up tables.

To make a conclusion, the robustness of this cryptosystem is not adequate for application areas such as computer networks which data security plays a crucially important role. Therefore, future works include but not limited to the following areas:

1) Development of a binary information protection mechanism to protect the Boolean permutations or look-up tables, and, as a result, increase the robustness of the proposed encryption algorithm.

2) Development of a new lightweight encryption algorithm based on Boolean permutations and Boolean matrices specifically desinged for Internet-of-Things by considering all the positive points of the analyzed cryptosystem in this paper and eliminating all its negative points.

REFERENCES

[1] K. I. Lakhtaria, "Protecting computer network with encryption technique: A study," Commun. Comput. Inf. Sci., vol. 151 CCIS, no. PART 2, pp. 381–390, 2011.

[2] S. Chandra, S. Paira, S. Safikul Alam, and G. Sanyal, "A comparative survey of symmetric and asymmetric key cryptography," *Int. Conf. Electron. Commun. Comput. Eng.*, pp. 83–93, 2014.

[3] a. Menezes, P. van Oorschot, and S. Vanstone, "Overview of Cryptography," *Handb. Appl. Cryptogr.*, pp. 1–48, 1996.

[4] E. Setyaningsih, C. Iswahyudi, and N. Widyastuti, "Image Encryption on Mobile Phone using Super Encryption Algorithm," vol. 10, no. 4, pp. 835–843, 2012.

[5] A. Jolfaei and A. Mirghadri, "Image Encryption Using Chaos and Block Cipher," *Comput. Inf. Sci.*, vol. 4, no. 1, pp. 172–185, 2011.

[6] Y. Alaverdyan, "Fast asymmetric cryptosystem based on Boolean product of matrices," pp. 392–395, 2009.

[7] J. a. Anderson, J. Anderson, and J. Bell, "Discrete Mathematics with Combinatorics," p. 799, 2000.

[8] V. Varadharajan and C.-K. Wu, "Public key cryptosystems based on boolean permutations and their applications," *Int. J. Comput. Math.*, vol. 74, no. 2, pp. 167–184, 2000.

[9] J. Lee and W. Lin, "A Lightweight Authentication Protocol for Internet of Things," pp. 1–2, 2014.

[10] R. H. Weber, "Internet of Things – New security and privacy challenges," *Comput. Law Secur. Rev.*, vol. 26, no. 1, pp. 23–30, 2010.

[11] "INFSO D.4 Networked Enterprise & RFID INFSO G.2 Micro & Nanosystems, in: Co-operation with the Working Group RFID of the ETP EPOSS, Internet of Things in 2020, Roadmap for the Future, Version 1.1, 27 May 2008."

[12] Y. Ben Saied, A. Olivereau, D. Zeghlache, and M. Laurent, "Lightweight collaborative key establishment scheme for the Internet of Things," *Comput. Networks*, vol. 64, pp. 273–295, 2014.

[13] X. Yao, Z. Chen, and Y. Tian, "A lightweight attribute-based encryption scheme for the Internet of Things," 2014.

[14] K. Nur, P. St, D. Darlis, and S. Si, "AN IMPLEMENTATION OF DATA ENCRYPTION FOR INTERNET OF THINGS USING BLOWFISH ALGORITHM ON FPGA 2 . Related Work," pp. 75–79, 2014.

[15] L. Touati, Y. Challal, and A. Bouabdallah, "C-CP-ABE: Cooperative Ciphertext Policy Attribute-Based Encryption for the Internet of Things," *2014 Int. Conf. Adv. Netw. Distrib. Syst. Appl.*, pp. 64–69, 2014.