Healing the Hill Cipher, Improved Approach to Secure Modified Hill against Zero-plaintext Attack

Mohammad Hadi Valizadeh
Computer Science and Engineering Department,
University of Connecticut, Storrs, CT, USA
mohammad.valizadeh@uconn.edu

Abstract - *Hill Cipher is a symmetric cryptosystem that was claimed to suffer from known-plaintext attack for many years. Different methods have been proposed to make this cipher more secure against known attacks. The introduced classic Hill cipher by Tourani and Falahati in 2011 that was devised in two variants and based upon affine transformation, was considered to be more secure against known attacks. Recently, this well modified Hill cipher is claimed to be vulnerable to zero-plaintext attack. In this paper, by using a chaotic map and scrambling methods, a novel cryptosystem based on Tourani and Falahati Hill cipher is presented which overcomes the zero-plaintext attack. The proposed Hill cipher is more reliable and faster.*

Keywords: Hill Cipher; Zero-plaintext Attack; Chaotic Map, Arnold Transformation

1. Introduction

The famous classical symmetric ciphering algorithm, known as Hill cipher which is based on matrix transformation, is said to be vulnerable to cryptanalysis that has rendered it inapplicable in practice. However, it still serves as an important educational material in both cryptology and linear algebra. It employs simply the matrix multiplication and inversion to encrypt and decrypt, causing to conceal letter frequencies of the plaintext. Throughput and high speed of operation, are of many advantages that are offered by the Hill cipher [3], but it suffers from the known-plaintext attack [4]. Many methods have been prescribed to abandon this security defect [5-7]. Most recent of all, by Tourani and Falahati [8], in 2011, proved commonly trusted Lin et al [6] is not so efficient and it can be exposed to the chosen-ciphertext attack because of an evident security flaw in the underlying protocol. Their presented method named as affine Hill cipher and was based on Lin et al.'s scheme, it was emanated with two protocols for data communication between Alice and Bob. We refer to their cipher as Tourani-Falahati Hill Cipher and shorten the name to TFHC. However in 2013, TFHC was proved by Keliher and Delaney [9] that the proposed scheme is vulnerable to Zero-plaintext attack in spite of its many security improvements.

This present manuscript aims to introduce a further secure Hill cipher cryptosystem. It includes every security advantages of TFHC by employing Arnold transformation to heal the only

security drawback claimed by [9]. The encryption core has the same structure deployed by TFHC and is based on the affine Hill cipher. HMAC can be additionally used regarding the generation of corresponding random number for each block in a hash chain. This facilitates more randomization into the linear structure of the affine Hill cipher over ordinary hash functions. Arnold Transform which is a well-known chaotic map is also deployed to stop the possibility of Zero-plaintext attack, proposed by Keliher and Delaney [9].

2. Background and Related Work

2.1. The Hill cipher and TFHC

In Hill cryptosystem, the cipher-text content is extracted from the plaintext through a linear transformation. Each plaintext row vector **X,** is encrypted to a cipher-text row vector $\mathbf{Y}_{1\times n} = \mathbf{X}_{1\times n}\mathbf{K}_{n\times n} (\mathrm{mod}\, m)$. The key matrix **K**, must be shared between the participators of the protocol securely where $k_{xy} \in Z_m$ and Z_m is a ring of integers modulo *m*, in which *m* is a natural number greater than one which can be selected optionally. The cipher text **Y** is decrypted as $\mathbf{X} = \mathbf{Y}\mathbf{K}^{-1} (\mathrm{mod}\ \mathrm{m})$. For the feasibility of decryption, the key matrix **K** must be invertible and it should satisfy $\gcd(\det \mathbf{K} (\mathrm{mod}\, m), m) = 1$ equivalently [4]. Bearing in mind, many of square matrices are not invertible over Z_m. The key-space for the Hill cipher is $GL(n, Z_m)$, the group of $n \times n$ matrices that are invertible over Z_m [8]. The probability of a randomly selected square matrix to be invertible is almost one for any large prime modulus as demonstrated by [10], while it is almost zero for a composite modulus. Moreover, choosing a prime modulus leads to a larger key-space in comparison with a composite modulus [10]. The key-space also increases with the increase of *n*, the rank of the key matrix. Therefore, an increase in the rank of the key matrix and selecting a large enough prime number as the modulus, can in consequence lead to a larger key-space and higher security. However, there is obviously a tradeoff between the whole system's favored security and efficiency that are additional parameters causing an augmentation with the running time and reduces the efficiency.

Hill cipher's actual stability depends upon two main parameters. Secrecy of the key matrix **K** and its rank. For an unknown *n* and a diminutive modulus *m,* the attacker could simply examine successive values of *n* till **K** is found. The Hill Cipher's weaknesses to the known-plaintext attack is considered as its most important security imperfection since it can be cracked by taking *n* distinct pairs of plaintext and corresponding cipher-text [4-11].

One extension to the Hill cipher is the Affine Hill which appends it with a nonlinear affine transformation with encryption expression form of $\mathbf{Y} = \mathbf{X}\mathbf{K} + \mathbf{V} (\mathrm{mod}\, m)$ [4].

Moreover, as mentioned, TFHC is a variant of Affine Hill. A brief description of their scheme is presented as follows. Alice decides to send a $n \times n$ message **M** to Bob. She breaks M into *n* tuple row vectors $\mathbf{X}_t$, for $t = 1,2, \ldots, n$. In order to construct a secure communication, Alice

picks a random integer and computes a group of random numbers by means of a preferred one-way hash function or HMAC to prevent excessive random number generation. For each step of encryption, Alice builds an auxiliary row vector $\mathbf{V}_t$, in a specific manner. She also generates a new key matrix $\mathbf{K}_t = v_0\mathbf{K}$. The encrypted version of each $\mathbf{X}_t$ is $\mathbf{Y}_t = \mathbf{X}_t\mathbf{K}_t + \mathbf{V}_t$. Alice repeats this procedure for n times and constructs the whole encrypted message $E(\mathbf{M}) = (\mathbf{Y}_1, \mathbf{Y}_2, \dots, \mathbf{Y}_n)$. She uses a secure protocol introduced in [8] and sends the necessary information to Bob. Consequently, Bob, by the given information that Alice provided him due to the protocol, can decrypt each $\mathbf{X}_t$ using $\mathbf{X}_t = (\mathbf{Y}_t - \mathbf{V}_t){\mathbf{K}_t}^{-1}$to recover the original message **M**. Recently, the authors of [9] announced the vulnerability of this cryptosystem to zero-plaintext attack. They declared that all the entries of the key matrix in each encryption step can be revealed by their chosen-plaintext attack in which if a plaintext $\mathbf{X}_t = 0$, then the corresponding ciphertext is $\mathbf{Y}_t = 0\mathbf{K}_t + \mathbf{V}_t = \mathbf{V}_t$, that is, the ciphertext, is equal to $\mathbf{V}_t = (v_1, v_2, \dots, v_n)$. This eventually leads to $\binom{n-1}{2}$linear equations each involving two entries in **K**. A more detailed study of this attack can be found in [9].

2.2. Arnold Transform

Scrambling methods are used widely in many digital processing methods particularly in digital watermarking applications for changing the distribution of the error bits in an image to improve the robustness of digital watermarking technology [12]. Arnold Transform named after Vladimir Arnold, also known as Arnold's Cat Map (ACM), is a chaotic map in mathematics from the torus into itself [13]. As an example, when ACM applied to a digital image it randomizes the original position of its pixels and the image becomes ambiguous. Two main features of Arnold scrambling algorithm are its simplicity and periodicity. According to the periodicity of Arnold scrambling, the original image can be restored after several cycles. Generally, the cycle of Arnold transformation is not directly proportional to the image size. Original ACM algorithm is based on square digital image mostly with $n \times n$ pixels. The normal Arnold's cat map method uses the following equation for transformation [12]:

$$\begin{bmatrix} a' \\ b' \end{bmatrix} = \begin{bmatrix} 2 & 1 \\ 1 & 1 \end{bmatrix} \begin{bmatrix} a \\ b \end{bmatrix} (mod\ n) \qquad (1)$$

In which $a, b \in \{0,1,2, \dots, n-1\}$ and n is the input image size. A new image will be produced when all the image points are manipulated by the above equation. ACM is a simple but powerful transform and is normally applied over digital images as described in the following:

Consider an $n \times n$ image I, for which c is the period of the transform. By applying Arnold Transform for a random repetition of t times, $t \in [1, c)$, a scrambled image I' which is totally different from I, is obtained. For decryption, this process is repeated $(c - t)$ times to regain the original image. In our proposed method ACM is applied at the last steps of encryption with a random iterations in order to strengthen the cryptosystem against zero-plaintext attack. In order to clarify, we represent ACM(X, c) as applying Arnold Transform to X matrix with c

number of iterations which leads to a new permuted matrix as X'. Fig. 1. Shows ACM's scrambling period for different size of input matrices.

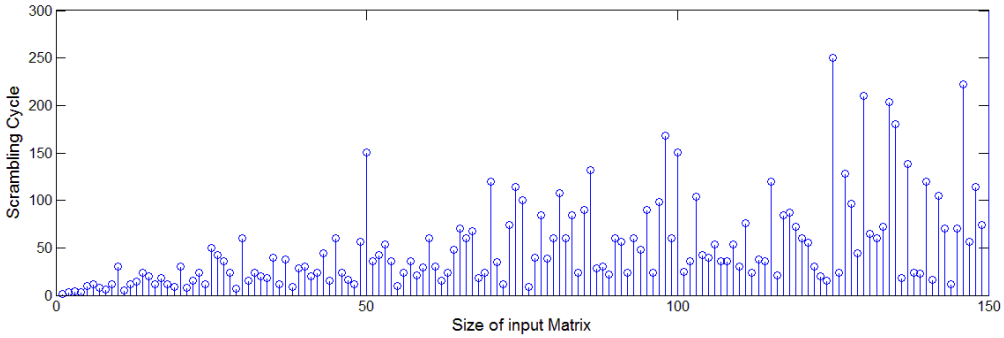


Figure 1. ACM's scrambling period for different sizes of input matrix

3. The Proposed Cryptosystem

The encryption and decryption procedures of the proposed cryptosystem is depicted in Fig. 2. Each $n \times n$ block of input data is encrypted using a unique random number in order to enforce more randomization to the introduced scheme and reinforcing it against the common attacks. In the same line, after generating the first random number, at the start of encryption process, a corresponding group of random numbers are recursively generated employing HMAC or a one-way hash function in a chain manner to avoid multiple random number generations. The primary random number that is generated at the beginning must be securely distributed among the participators. Consequently, a suitable protocol is essential to be performed. The recommended cryptosystem using an uncomplicated one-pass protocol for encryption and decryption of each $n \times n$ blocks of data is described in the following steps:

Encryption	**Decryption**
$v_0 = a_t = H(a_{t-1})$	$v_0 = a_t = H(a_{t-1})$
If $a_t \equiv 0$: $v_0 = 1(mod\ p)$	If $a_t \equiv 0$: $v_0 = 1(mod\ p)$
For n times: $\begin{Bmatrix} j_i = (v_{i-1}\ mod\ n) + 1 \\ \hat{v}_{i-1} = 2^{\lfloor \gamma \rfloor} + \left(v_{i-1}\ mod 2^{\lfloor \gamma \rfloor}\right) \\ \gamma = \lfloor log_2 v_{i-1} \rfloor + 1 \\ v_i = (k_{ij_i} + \hat{v}_{i-1} v_0)(\text{mod p}) \\ \mathrm{V}_t = (v_1, v_2, \dots, v_n) \end{Bmatrix}$ $i = 1, \dots, n$	For n times: $\begin{Bmatrix} j_i = (v_{i-1}\ mod\ n) + 1 \\ \hat{v}_{i-1} = 2^{\lfloor \gamma \rfloor} + \left(v_{i-1}\ mod 2^{\lfloor \gamma \rfloor}\right) \\ \gamma = \lfloor log_2 v_{i-1} \rfloor + 1 \\ v_i = (k_{ij_i} + \hat{v}_{i-1} v_0)(\text{mod p}) \\ \mathrm{V}_t = (v_1, v_2, \dots, v_n) \end{Bmatrix}$ $i = 1, \dots, n$
$\bar{a} = \lfloor \frac{\sum_{t=1}^{n} a_t}{n} \rfloor (mod\ p)$	$\bar{a} = \lfloor \frac{\sum_{t=1}^{n} a_t}{n} \rfloor (mod\ p)$
$\mathrm{V}_{n\times n} = \begin{bmatrix} \mathrm{V}_1 \\ \mathrm{V}_2 \\ \vdots \\ \mathrm{V}_N \end{bmatrix}$	$\mathrm{V}_{n\times n} = \begin{bmatrix} \mathrm{V}_1 \\ \mathrm{V}_2 \\ \vdots \\ \mathrm{V}_N \end{bmatrix}$
$\mathrm{K}' = \mathrm{ACM}(\mathrm{K}, \bar{a})$	$\mathrm{K}' = \mathrm{ACM}(\mathrm{K}, \bar{a})$
$\mathrm{V}' = \mathrm{ACM}(\mathrm{V}, \bar{a})$	$\mathrm{V}' = \mathrm{ACM}(\mathrm{V}, \bar{a})$
$\mathrm{Y}_{n\times n} = \mathrm{X}_{n\times n}\mathrm{K}'_{n\times n} + \mathrm{V}'_{n\times n}$	$\mathrm{X}_{n\times n} = (\mathrm{Y}_{n\times n} - \mathrm{V}'_{n\times n})\mathrm{K}'^{-1}{}_{n\times n}$

Figure 2. Encryption and Decryption cores for the suggested cryptosystem

1. Alice will choose a random integer, $a_0 \in [1, p-1]$ and computes $a_1, a_2, \ldots, a_n$, where $a_t = H(a_{t-1})$ for $t \geq 1$. $H(.)$ Can be any selected one-way hash function.
2. For $1 \leq t \leq n$, Alice assigns $v_0 = a_t \pmod p$, unless the resulting value of v_0 is 0, in which case $v_0 = 1$.
3. Alice constructs the row vector $\mathbf{V}_t = (v_1, v_2, \ldots, v_n)$ for $1 \leq t \leq n$ as follows: for$1 \leq i \leq n$, she sets $j_i = (v_{i-1} \, mod \, n) + 1$ and $v_i = (k_{ij_i} + \hat{v}_{i-1} v_0) \pmod p$, where $\hat{v}_{i-1} = 2^{\lceil \gamma/2 \rceil} + (v_{i-1} \, mod 2^{\lceil \frac{\gamma}{2} \rceil})$ and $\gamma = \lfloor log_2 v_{i-1} \rfloor + 1$ in which $\lceil . \rceil$ and $\lfloor . \rfloor$ represent the ceiling and floor functions respectively.
4. Alice builds the V matrix $V_{n \times n} = \begin{bmatrix} \mathbf{V}_1 \\ \mathbf{V}_2 \\ \vdots \\ \mathbf{V}_n \end{bmatrix}$, then she encrypts each X plaintext using $\mathbf{Y}_{n \times n} = \mathbf{X}_{n \times n} \mathbf{K'}_{n \times n} + \mathbf{V'}_{n \times n}$ where $\mathbf{K'} = \mathrm{ACM}(K, \bar{a})$ and $\mathbf{V'} = \mathrm{ACM}(V, \bar{a})$ in which $\bar{a} = \lfloor \frac{\sum_{t=1}^{n} a_t}{n} \rfloor (mod \, p)$
5. Alice selects a random integer $b \in [1, n^2]$ and computes $x = \lceil \frac{b}{n} \rceil$ and $y = b - n(x-1)$. She then computes $r = a_0 k_{xy} (mod \, p)$.
6. Alice sends (Y, b, r) to Bob.
7. Bob derives x and y from b as in step 5, and obtains k_{xy} from $\mathbf{K}$. He recovers a_0 by computing $r k_{xy}^{-1} (mod \, p) = a_0 k_{xy} k_{xy}^{-1} (mod \, p) = a_0$ and consequently each a_t which leads to calculating $\bar{a}$. Bob can then compute each $\mathbf{V}_t$ in the same manner as Alice, which leads to computing $\mathbf{K'}$ and $\mathbf{V'}$ and finally recovers the original message using $\mathbf{X}_{n \times n} = (\mathbf{Y}_{n \times n} - \mathbf{V'}_{n \times n}) \mathbf{K'^{-1}}_{n \times n}$.

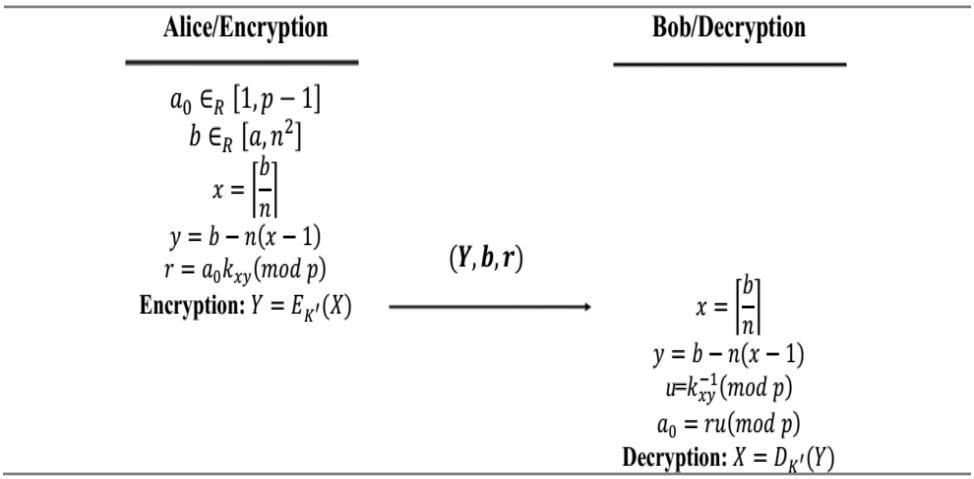


Figure 3. The structure of one-pass protocol to deploy in the suggested system

Fig. 3 depicts a one-pass protocol for the proposed cryptosystem and is based on TFHC. Even though this protocol does not include any authentication phase, it is secure and does not expose any clandestine information. It is applicable when both partakers are not online [8].

3.1. Properties of the proposed scheme

The suggested cryptosystem which is introduced above is a variant and an improvement of TFHC [8] with the main difference of securing the zero plaintext attack. It is able to confuse the attacker using an Arnold Cat Map that permutes the **V** and **K** matrices. In spite of security improvements in the proposed algorithm presented by [8], it was proved by Keliher et al. that it can be easily broken by a chosen-plaintext attack in which if a plaintext $\mathbf{X}_t = 0$, then the corresponding ciphertext is $\mathbf{Y}_t = 0\mathbf{K}_t + \mathbf{V}_t = \mathbf{V}_t$, i.e., the ciphertext $\mathbf{V}_t = (v_1, v_2, \dots, v_n)$. Using each v_i and the information in step 3 from their proposed scheme, each $\hat{v}_i$ can be computed. This eventually leads to $\binom{n-1}{2}$ linear equations each involving two entries in **K**. In TFHL, the attacker can benefit the covered information in the **V** vector by means of a zero-plaintext attack. In this work, instead of using a vector we produced a random square matrix, then we permuted this matrix using the ACM with a random number of iteration in order to make attacker more confused. In this way, by using the proposed scheme in this paper the vulnerability to zero-plaintext attack is totally defeated because, by placing the input plaintext to 0, the attacker can reach a $\mathbf{V}'$matrix for each $n \times n$ input blocks that doesn't provide him with any useful information about the **V** matrix. In fact $\mathbf{V}'$ is a permuted version of **V** matrix, after applying ACM to **V** with a random number of iteration. The iteration value, $\bar{a}$ is also vague for the attacker by means of using the safe proposed protocol. In the same manner, the security of the system has been improved significantly due to generating a different key matrix in each encryption step using a chaotic map with a random number of iterations in order to make the attacker more confused through the final step of encryption.

ACM's scrambling cycle for some selected sizes of input matrix is presented in table 1. It is important to notice that the scrambling cycle can be the same for different input sizes (as it is independent of the input size), however, the larger the scrambling cycle, the better from a security point of view.

Table 1. ACM's Scrambling cycle for different input matrix size

Size of input Matrix(N)	Scrambling Cycle	Size of input Matrix(N)	Scrambling Cycle
50	**150**	100	**150**
70	**120**	125	**250**
74	**114**	130	**210**
86	**132**	150	**300**
98	**168**	256	**192**

3.2. Computational Costs

An important criterion for evaluating a cryptosystem is its computational costs that is evaluated in this section. However, since the proposed scheme is mainly based on TFHC, and a comprehensive study of TFHC's computational costs is presented by the same authors in [8], hence, we focus on the extra costs of applying ACM to TFHC cryptosystem. By neglecting required computations of the protocol and considering only the computational costs of the ciphering core, we have:

$$T_{Enc} \cong (n^2 + n)T_{Mul} + (n^2)T_{Add} + T_H + T_{ACM} \quad (2)$$

$$T_{Dec} \cong (n^2 + n)T_{Mul} + (n^2)T_{Add} + T_H + T_{ACM} + T_{Inv} \quad (3)$$

where encryption and decryption of each block of data are represented by T_{Enc} and $T_{Dec}as$ running times respectively. T_H is the running time for the Hash calculations that is determined by the kind of embedded hash function [16]. T_{ACM} is the required Arnold Scrambling time. T_{Mul}, T_{Inv} and T_{Add} are the time necessary for the modular multiplication, inversion and addition calculations respectively.

A comparison between the required number of operations for encrypting/decrypting each block of data are revealed in Table 2 for the proposed scheme and other schemes. Regardless of the security advantages of the proposed scheme over the previously existing ones, it has a high computational efficiency due to *one-time* encryption/decryption process instead of *n time* encryption/decryption for an $n \times n$ input block of data.

Table 2. Number of operations required for Encryption and Decryption of each $1 \times n$ block of data by different methods

Scheme	Operation	T_{Mul}	T_{Add}	T_H	T_{ACM}	T_{Inv}
Original Hill	Enc/Dec	n^2	$n^2 - n$	-	-	-
Affine Hill	Enc/Dec	n^2	n^2	-	-	-
TFHC	Enc	$n^2 + 2n$	$n^2 + n + 1$	*	-	-
	Dec	$n^2 + 2n$	$n^2 + n + 1$	*	-	*
Proposed Scheme	Enc	$n^2 + n$	n^2	*	*	-
	Dec	$n^2 + n$	n^2	*	*	*

The total processing time for enciphering/deciphering the whole block of plaintext/ciphertext can be assessed by multiplying the running time of each $1 \times n$ block of data with the whole number of data blocks. Considering a plaintext to have a length of *L* letters that is not a multiple of *n*, padding can then be deployed to obtain a multiple of *n* to reach the number of data blocks as $\lceil L/n \rceil$. However, if data length is fixed, increasing *n* will decrease the number of data blocks and vice versa [8], so the running time for encrypting/decrypting the whole plaintext/ciphertext can be followed by relations 4 and 5:

$$T_{Total\_Enc} \cong \lceil \frac{L}{n} \rceil \left((n^2 + n) T_{Mul} + (n^2) T_{Add} + T_H + T_{ACM} \right) \quad (4)$$

$$T_{Total\_Dec} \cong \lceil \frac{L}{n} \rceil \left((n^2 + n) T_{Mul} + (n^2) T_{Add} + T_H + T_{ACM} + T_{Inv} \right) \quad (5)$$

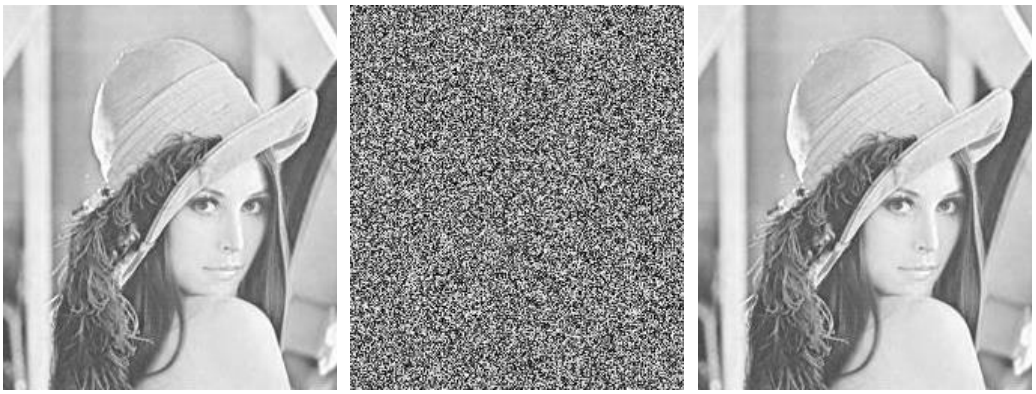
For evaluating the computational costs of ACM, ACM is applied to 32×32 and 64×64 different input matrix and are utilized for Macintosh and Windows icon sizes respectively. The computational cost of ACM is also calculated for 200×200 and 250×250 input matrices which are standards for Internet Ads. The simulations executed on a Lenovo idea pad s410p laptop, running on a 64-bit operating system with an Intel Core i5 4200U processor, bearing a 4GB DDR3 RAM.

Table 3 presents the average runtime of applying ACM to some selected input matrices. The simulations performed 200 times for each matrix dimension considering random iteration values of Arnold Transform.

Table 3. Average runtime of applying ACM to some selected input matrices

Input matrix dimension	32×32	64×64	200×200	250×250
Avg. ACM Runtime(sec)	**0.0096**	**0.0775**	**2.0665**	**3.3228**

The simulation results show that the $n \times n$ matrix manipulation presented here saves considerable computation costs. This makes the computational costs due to added ACM insignificant. A comprehensive study of ACM's computational cost can be found on [17]. Fig. 4 shows an encrypted Lena Image using the proposed method.



(a) (b) (c)

Figure 4. (a) The original 256*256 Lena image used in simulation, (b) The encrypted image using the proposed method, (c) The decrypted image using the proposed method

4. Conclusions

In this paper, a further secure Hill cipher cryptosystem, which is actually a variant of the Tourani and Falahati Hill cipher, is presented. The proposed cryptosystem employs Arnold transformation to heal the only security drawback claimed by [9] and it can be implemented by the use of a simple cryptographic protocol without compounding any excess computational costs. In all published methods on the subject, in order to encrypting an $n \times n$ block of input data, each block of $1 \times n$ input data, is encrypted and the encryption process must be repeated for *n* times, but as it is shown in this manuscript, the encryption process will be performed for a $n \times n$ block of data and it contains fewer number of operations which leads to lower computational costs. In addition to lower computational costs, all the known attacks to Hill Cipher including zero-plaintext attack can be thwarted by means of the proposed cryptosystem and its underlying protocol introduced in this paper. In the same line, security of key matrix is provided by using a different key for each encryption process.

References

[1] Hill LS. Cryptography in an Algebraic Alphabet. *American Mathematical Monthly* 1929; *36:* 306-312.

[2] Hill LS. Concerning Certain Linear Transformation Apparatus of Cryptography. *American Mathematical Monthly* 1931; *38:* 135-154.
[3] Ismail IA, Amin M, Diab H. How to repair the Hill cipher. *Journal of Zhejiang University-Science A* 2006, *7:* 2022-2030.
[4] Stinson DR. *Cryptography Theory and Practice.* Chapman & Hall/CRC, 2006.
[5] Saeednia S. How to Make the Hill Cipher Secure. *Crypto logia Journal* 2000; *24:* 353-360.
[6] Lin CH, Lee CY, Lee CY. Comments on Saeednia improved scheme for the Hill cipher. *Journal of the Chinese institute of engineers* 2004; *27:* 743-746.
[7] Li C, Zhang D, Chen G. Cryptanalysis of an image encryption scheme based on the Hill cipher. *Journal of Zhejiang University - Science A* 2008; *9:* 1118-1123.
[8] Tourani, Mohsen, and Falahati Abolfazl. "A secure cryptosystem based on affine transformation." *Security and Communication Networks* 4.2 (2011): 207-215.
[9] Keliher, Liam, and Anthony Z. Delaney. "Cryptanalysis of the Toorani-Falahati Hill Ciphers." *Computers and Communications (ISCC), 2013 IEEE Symposium on*. IEEE, 2013.
[10] Overbey J, Traves W, Wojdylo J. On the Keyspace of the Hill Cipher. *Cryptologia Journal* 2005; *29:* 59-72.
[11] Yeh YS, Wu TC, Chang CC, Yang WC. A New Cryptosystem Using Matrix Transformation. *25th IEEE International Carnahan Conference on Security Technology* 1991: 131-138.
[12] Li, Min, Ting Liang, and Yu-jie He. "Arnold Transform Based Image Scrambling Method." (2013).
[13] Ashtiyani, Meghdad, Parmida Moradi Birgani, and Hesam M. Hosseini. "Chaos-based medical image encryption using symmetric cryptography." *Information and Communication Technologies: From Theory to Applications, 2008. ICTTA 2008. 3rd International Conference on*. IEEE, 2008.
[14] Knuth DE. *The Art of Computer Programming.* Addison-Wesley: Massachusetts, 1981; 2: 1-33.
[15] Rosen KH. *Elementary Number Theory and Its Applications.* Addison-Wesley, Massachusetts, Second edition, 1988.
[16] Elkeelany, O., et al. "Performance analysis of IPSec protocol: encryption and authentication." *Communications, 2002. ICC 2002. IEEE International Conference on*. Vol. 2. IEEE, 2002.
[17] Soleymani A., Nordin M., and Sundararajan E., A Chaotic Cryptosystem for Images Based on Henon and Arnold Cat Map. The scientific World Journal, 2014.