Computers, Materials & Continua
DOI:10.32604/cmc.2020.013104
Article

Tech Science Press

A Novel Approach to Data Encryption Based on Matrix Computations

Rosilah Hassan[1], Selver Pepic[2], Muzafer Saracevic[3], Khaleel Ahmad[4,*] and Milan Tasic[5]

[1]Centre for Cyber Security, Faculty of Information Science and Technology, Universiti Kebangsaan Malaysia, 43600 UKM, Bangi Selangor, Malaysia
[2]Technical Machine School of Professional Studies, Radoja Krstića 19, Trstenik, 37240, Serbia
[3]University of Novi Pazar, Dimitrija Tucovića bb, Novi Pazar, 36300, Serbia
[4]Maulana Azad National Urdu University, Hyderabad, Telangana, 500032, India
[5]University of Nis, Višegradska 33, Niš, 18106, Serbia
[*]Corresponding Author: Khaleel Ahmad. Email: khaleelahmad@manuu.edu.in

Received: 26 July 2020; Accepted: 22 August 2020

Abstract: In this paper, we provide a new approach to data encryption using generalized inverses. Encryption is based on the implementation of weighted Moore–Penrose inverse $A^{\dagger}_{MN}(nxm)$ over the $nx8$ constant matrix. The square Hermitian positive definite matrix N_{8x8} p is the key. The proposed solution represents a very strong key since the number of different variants of positive definite matrices of order 8 is huge. We have provided NIST (National Institute of Standards and Technology) quality assurance tests for a random generated Hermitian matrix (a total of 10 different tests and additional analysis with approximate entropy and random digression). In the additional testing of the quality of the random matrix generated, we can conclude that the results of our analysis satisfy the defined strict requirements. This proposed MP encryption method can be applied effectively in the encryption and decryption of images in multi-party communications. In the experimental part of this paper, we give a comparison of encryption methods between machine learning methods. Machine learning algorithms could be compared by achieved results of classification concentrating on classes. In a comparative analysis, we give results of classifying of advanced encryption standard (AES) algorithm and proposed encryption method based on Moore–Penrose inverse.

Keywords: Security; data encryption; matrix computations; cloud computing; machine learning

1 Introduction

The level of security of data stored on the cloud is primarily based on the identification of sensitive and confidential databases, and it is necessary to apply additional protection, encryption, and monitoring. It is important to consider whether it is possible to encrypt data at all levels, where they are designed, and how encryption algorithms are tested. Data encryption became of great importance in many fields including healthcare [1], and several encryption methods have been investigated including triple data encryption [2]. The most basic way cloud providers provide data is encryption. Indeed, using clouds in order to provide data demand services is developing to be an attractive answer to services demanding scalability and cost reduction [3]. Applying encryption in a cloud environment can further protect data from theft and unauthorized use. It is important to emphasize that data security is based on an understanding of risk. Users need to be aware of the potential risks when storing their data in the cloud or using cloud applications. Therefore, they must apply different control processes and techniques to manage these risks and reduce them to an acceptable level. The dynamics of information processing in the IoT and cloud environment and the fluidity of information are the reasons. It is important to provide two types of security in the IoT and cloud environments. The first security refers to cloud providers, and the second refers to user security. Compliance of cloud providers with requirements and standards in the field of cloud security is implemented and enforced to meet security guidelines, recommendations, laws, and regulations. Some of the key challenges for IoT and cloud security are: storing data in multiple locations, storing data on media and resources shared by multiple users, availability of data after termination of the contract with the provider, in case of sale or merger companies, compliance with legal regulations, the problem of external supervision, as well as the restoration of data in the event of natural disasters or due to human error. It is especially important to take care of three aspects: data location, data control, and secure data transfer.

The major contribution of this paper regarding the issues of security and efficiency may refer to multiple different encryptions based on the random key (in form of matrix), while data encryption is based on different inversions (in this paper we have presented one of them). We present a novel method of data encryption based on matrix calculations and Weighted Moore–Penrose inverses (MP Encryption). The Moore Penrose inverses have found many applications in various areas of research. This proposed MP encryption method can be applied effectively in the encryption and decryption of images in multi-party communications.

The structure of the present paper is as follows. In the second section are exposed the similar research from the field of problems application matrices in cryptography. Also, we have presented some similar research in the field of data encryption in a cloud environment. The third section consists of the basic properties of weighted Moore–Penrose inverse (MP inverse) and presented ways to compute source matrix which can presented text or image. In the fourth section, are listed the examples for the encryption method based on weighted Moore Penrose inverse and Hermitian positive definite matrix as a cryptographic key in image encryption cases. Also, in this section, we have provided the National Institute of Standards and Technology (NIST) quality assurance tests for random generated Hermitian matrix (a total of 10 different tests and additional analysis with approximate entropy and random digression). The fifth section contains the comparative analysis of encryption methods between machine learning methods. Machine learning algorithms could be compared by achieved results of classification concentrating on classes. Sensitivity and specificity are mostly used performance measuring of complex data during classification. In this research sensitivity and specificity define achieved results of classifying Advanced Encryption Standard (AES) and MP Encryption, respectively. The sixth section lists the conclusions and suggestions for further works.

2 Related Works

Cloud computing, a recently emerged paradigm faces major challenges in achieving the privacy of migrated data, network security, etc. Too many cryptographic technologies are raised to solve these issues based on identity, attributes, and prediction algorithms yet. These techniques are highly prone to attackers. This would raise a need for an effective encryption technique, which would ensure secure data migration [4]. Cloud computing has been investigated in many research works including the factors and the impacts of its implementation in the public sector [5], Authors in [6] present a framework with data encryption, distribution, and decryption in a cloud environment. Ensuring security for data transmission and storage is the biggest concern and challenge of the Internet of Things (IoT) [7]. Cloud services are naturally located in locations that are far from premises in which the client organization is located. As soon as data and services are accessed from a remote location, the unprotected Internet is used as an access medium, which opens a new front for a potential attacker [8]. In [9,10], new techniques are presented to provide security of data in a cloud environment. In [11], a novel encryption scheme for a concrete model (client-server architecture) has been presented. Also, authors in [12] propose a new verifiable model reduces the computational overhead of encoding and decoding. The authors in [13] deal with the topic of personal data protection with an emphasis triggering moves to unlock its insights by relocating it in the cloud. First of all, the authors survey prominent clouded data approaches such as multiparty computation, blockchain, privacy, and encryption. To achieve the goal of confidentiality of data, many encryption algorithms are available in the cloud environment. Authors in [14] proposed the first self-updatable encryption model secure against a relevant form of chosen-ciphertext security. This approach is a new kind of public-key encryption, motivated by cloud computing. In [15], the authors presented the access control model for the security of data by using attribute-based encryption in the cloud computing environment. The paper [16] proposed a security model to protect cloud data from unauthorized access using a hybrid cryptosystem. The proposed approach provides high-level security to data stored in cloud computing and ensures secure data transmission over the network. In [17] authors discussed security issues and challenges in cloud computing and study various security algorithms in this environment. In [18], a method called hyperdata encryption is proposed. The proposed solution is suitable for cloud platforms. For the secure transfer of sensitive data, the data should be encrypted before sending. Achieving the integrity goal ensures that data is not modified, damaged, or corrupted either accidentally or intentionally. The goal of data integrity is just as important as the other two main goals of computer security in a cloud environment. In [19], a multi-client universal computation model for encrypted cloud data. To achieve computer security goals on the cloud available numerous technologies. Data protection technologies, frameworks, and implementation, which achieve greater security of cloud data, are a complex and broad area, which is regularly used in the implementation of everyday activities of cloud computing.

The importance of the application of matrix computations in the encryption procedure is stated in [20–22]. Specifically, in this paper, we provided a new approach to data encryption using generalized inverses. Also, authors in [23] state some applications generalized inverses in public key cryptosystem design. In [24] authors state applications of the Drazin inverse to the Hill cryptographic system. Papers [25,26] presents secure encryption and decryption technique using generalized inverse and decimal expansion of an irrational number. Authors in [27] present a novel approach that leads in a natural manner to the Moore–Penrose's generalized inverse between the subspaces of activation of the matrix under study. Paper [28] deals with the application of generalized inverses of matrices over finite fields and the method of least squares in linear codes. It is proven that if the Moore–Penrose inverse of a generator matrix of a linear code exists, a unique word approaching a received word near the codewords of the code can be found. Authors in [29,30] analyses about Hill cipher and public key cryptosystem using Hill secure algorithm. In [31] authors present secure communication protocols based on a computation of the MP inverse of matrices over fields of specific characteristics.

3 Preliminaries About Weighted Moore–Penrose Inverse

For any matrix $A \in C^{mxn}$ (A be the set of complex numbers, c^{mxn} be the set of mxn complex matrices of rank $C_r^{mxn} = \{C^{mxn} \mid rank(X) = r\}$) and positive definite Hermitian matrices M and N of the orders m and n respectively, consider the four conditions in X, where $*$ denotes conjugate and transpose. The first condition is $AXA = A$, *the* second condition is $XAX = X$, *the* third condition is $(MAX)* = MAX$ and the fourth condition is *(NXA)* = NXA*. Moreover, this system of matrix equations has a unique solution. The matrix X that satisfies all four conditions is called the *weighted MP inverse* and is denoted with $X = A^{\dagger}_{MN}$. The weighted MP inverse $A^{\dagger}_{MN}$ is the generalization of the MP inverse $A^{\dagger}$. If $M = I_m$, $N = I_n$, then $A^{\dagger}_{MN} = A^{\dagger}$, i.e., the weighted Moore–Penrose inverse is reduced to Moore–Penrose inverse [32]. The idea for applying weighted MP inverses in cryptography came from our previous papers [33–38], where we analyzed its basic properties. Milošević in [37] generalized Greville's method to the weighted MP inverse.

Theorem 3.1 (Wang et al. [33]) *Let $A \in C^{m\times n}$ and A_k is a submatrix which consist the first k columns of matrix A. For $k = 2, \ldots, n$ the matrix A_k is represented with*

$$\mathrm{A_k} = [\mathrm{A_{k-1}}|\mathrm{a_k}] \tag{1}$$

and $\mathrm{N_k} \in \mathrm{C^{k\times k}}$ is a submatrix of the matrix N. Then the matrix $\mathrm{N_k}$ is given as

$$\mathrm{N_k} = \begin{bmatrix} \mathrm{N_{k-1}} & \mathrm{l_k} \\ \mathrm{l_k^*} & \mathrm{n_{kk}} \end{bmatrix} \tag{2}$$

Let the matrices $\mathrm{X_{k-1}}$ and $\mathrm{X_k}$ and the vectors $\mathrm{d_k}$ and $\mathrm{c_k}$ are defined as

$$\mathrm{X_{k-1}} = (\mathrm{A_{k-1}})^{\dagger}_{\mathrm{MN_{k-1}}}\mathrm{X_k} = (\mathrm{A_k})^{\dagger}_{\mathrm{MN_k}}, \tag{3}$$

$$\mathrm{d_k} = \mathrm{X_{k-1}a_k} \qquad . \tag{4}$$

$$\mathrm{c_k} = \mathrm{a_k} - \mathrm{A_{k-1}d_k} = (\mathrm{I} - \mathrm{A_{k-1}X_{k-1}})\mathrm{a_k}. \tag{5}$$

Then

$$\mathrm{X_k} = \begin{bmatrix} \mathrm{X_{k-1}} - \left(\mathrm{d_k} + (\mathrm{I} - \mathrm{X_{k-1}A_{k-1}})\mathrm{N_{k-1}^{-1}l_k}\right)\mathrm{b_k^*} \\ \mathrm{b_k^*} \end{bmatrix}, \tag{6}$$

where

$$\mathrm{b_k^*} = \begin{cases} \left(\mathrm{c_k^*Mc_k}\right)^{-1}\mathrm{c_k^*M} & \mathrm{c_k} \neq 0 \\ \quad \delta_\mathrm{k}^{-1}\left(\mathrm{d_k^*N_{k-1}} - \mathrm{l_k^*}\right)\mathrm{X_{k-1}} & \mathrm{c_k} = 0 \end{cases} \tag{7}$$

$$\delta_\mathrm{k} = \mathrm{n_{kk}} + \mathrm{d_k}^*\mathrm{N_{k-1}d_k} - \mathrm{l_k}^*\mathrm{d_k} - \mathrm{d_k}^*\mathrm{l_k} - \mathrm{l_k}^*(\mathrm{l} - \mathrm{X_{k-1}A_{k-1}})\mathrm{N_{k-1}^{-1}l_k} \tag{8}$$

Different variants of calculation of the weighted MP inverse in combination with relational databases are given in the paper [37]. Since the number of combinations of the possible positive definite Hermitian 8 × 8 matrix is extremely large, it is clear that this type of matrix could be used in developing of the encryption algorithm and would be exceptionally strong.

For calculation of Algorithm 1, from [33], is needed an auxiliary Algorithm 2.

The partition method of Wang for calculation of the weighting MP inverse has been extended to a set of rational and polynomial matrices with one variable [38]. In the paper [26] is given the following equation:

$$\mathrm{A^{\dagger}_{M,N}} = \mathrm{N}^{-\frac{1}{2}}\left(\mathrm{M}^{\frac{1}{2}}\mathrm{AN}^{-\frac{1}{2}}\right)^{\dagger}\mathrm{M}^{\frac{1}{2}} \tag{9}$$

In this case, $A^{\dagger}_{M,N}b$ is the M-least squares solution of $Ax = b$ which has minimal N-norm. This notion can be extended where M and N are positive semi-definite matrices, and G is a matrix such that Gb is a minimal N semi-norm, M-least squares solution of $Ax = b$. In this case, G must satisfy the following four conditions [31]:

Algorithm 1: Calculation of MP inverse $\mathbf{A_{MN}}(\mathbf{s})^{\dagger}$ from [33] 10 : if $c_k \neq 0$ then

```
Require: Let $A \in C^{m \times n}$ and $M$ and $N$ are positive definite matrices with dimension $m \times m$ and $n \times n$, respectively.
1 : $A_1 = a_1$
2 :  if $a_1 = 0$ then
3 :      $X_1^{\dagger} = \frac{1}{a_1^* M a_1} a_1^* M$
4 :  else
5 :      $X_1 = 0$
6 :  end if
7 : for $k = 2$ to $n$ do
8 :      $d_k = X_{k-1} a_k$
9 :      $c_k = a_k - \hat{A}_{k-1} d_k$
10 for $C_k \neq 0$ then
11 :          $b_k^* = \left(c_k^* M c_k\right)^{-1} c_k^* M$, go to Step 16
12 :      else
13 :          $\delta_k = n_{kk} + d_k^* N_{k-1} d_k - \left(d_k^* l_k + l_k^* d_k\right) - l_k^* \left(I - X_{k-1}\hat{A}_{k-1}\right) N_{k-1}^{-1} l_k$
14 :          $b_k^* = \delta_k^{-1}\left(d_k^* N_{k-1} - l_k^* X_{k-1}\right)$
15 :      end if
16 :     $X_k = \begin{bmatrix} X_{k-1} - \left(d_k + \left(I - X_{k-1}\hat{A}_{k-1}\right) N_{k-1}^{-1} l_i\right) b_k^* \\ b_k^* \end{bmatrix}$
17 : end for
18 :  return $A_{MN}^{\dagger} = X_n$
```

Algorithm 2: Calculation of $\mathbf{Ni}^{-1}$ for the rational matrices

```
1 : $N_1^{-1} = n_{11}^{-1}$
2 : for $k = 2$ to $n$ do
3 :     $g_{kk} = \left(n_{kk} - l_k^* N_{k-1}^{-1} k\right)^{-1}$
4 :     $f_{ii} = -g_{kk} N_{k-1}^{-1} k$
5 :     $E_{k-1} = N_{k-1}^{-1} + g_{kk}^{-1} f_k f_k^*$
6 :     $N_k^{-1} = \begin{bmatrix} E_{k-1} & f_k \\ f_k^* & g_{kk} \end{bmatrix}$
7 : end for
8 :  return $N^{-1} = N_n^{-1}$
```

$MAGA = MA$, $NGAG = NG$, $(MAG)^* = MAG$, $(NGA)^* = NGA$. When N is positive definite, then there exists a unique solution for G.

Theorem 3.2. In the set of complex matrices let $\mathrm{A}^{\dagger}_{\mathrm{M,N}}$ is weighted MP inverse of matrix $\mathrm{A} \in \mathrm{C}^{\mathrm{m\times n}}$ and $\mathrm{C}^{\mathrm{m\times n}}_{\mathrm{r}} = \{\mathrm{X} \in \mathrm{C}^{\mathrm{m\times n}} :\ \mathrm{rank(X)} = \mathrm{r}\}$, M and N are Hermitian, positive definite matrices with order m and n, respectively. If M is an identity matrix, then the matrix A can be represented as

$$A = N^{\frac{1}{2}}A^{\dagger}_{M,N}N^{\frac{1}{2}} \tag{10}$$

Proof. From Eq. (9) is necessary to find the value of the matrix A that is represented in the expression.

This is the process by which we will restore the encoded image value to the actual one.

$$N^{\frac{1}{2}}A^{\dagger}_{M,N}M^{-\frac{1}{2}} = N^{\frac{1}{2}}N^{-\frac{1}{2}}\left(M^{\frac{1}{2}}AN^{-\frac{1}{2}}\right)^{\dagger}M^{\frac{1}{2}}M^{-\frac{1}{2}}$$

$$N^{\frac{1}{2}}A^{\dagger}_{M,N}M^{-\frac{1}{2}} = \left(M^{\frac{1}{2}}AN^{-\frac{1}{2}}\right)^{\dagger} \tag{11}$$

Let

$$M^{\frac{1}{2}} = \ M_1 \ and \ N^{-\frac{1}{2}} = \ N_1 \tag{12}$$

From $(MAX)^{*} = MAX$ we obtain

$$N_1^{-1}A^{\dagger}_{M,N}M_1^{-1} = M_1AN_1 \tag{13}$$

So,

$$M_1^{-1}N_1^{-1}A^{\dagger}_{M,N}M_1^{-1}N_1^{-1} = M_1^{-1}M_1AN_1N_1^{-1} \tag{14}$$

and

$$M_1^{-1}N_1^{-1}A^{\dagger}_{M,N}M_1^{-1}N_1^{-1} = A \tag{15}$$

This can be rewritten as,

$$A = M^{-\frac{1}{2}}N^{\frac{1}{2}}A^{\dagger}_{M,N}M^{-\frac{1}{2}}N^{\frac{1}{2}} \tag{16}$$

In order to simplify the expression and calculation process, without affecting the protection degree in this way we can assume that the matrix M is given as an identity matrix from order n, where n represents the number of rows of the encrypted matrix A.

Because M is the identity matrix, in this case, the expressions $M^{\frac{1}{2}}$ and $M^{-\frac{1}{2}}$ gives the identity matrix and from Eq. (16) we obtain

$$A = N^{\frac{1}{2}}A^{\dagger}_{M,N}N^{\frac{1}{2}} \tag{17}$$

Eq. (17) allows us to obtain the original matrix, which in our case represents the image, based on the weighted MP inverse and the Hermitian positive definite matrix N which is the key.

4 Proposed Method for Data Encryption

Cloud security can be automated by a combination of a number of services available, with the goal of creating an integrated platform for monitoring, reporting, and responding to events that could compromise the security of cloud data.

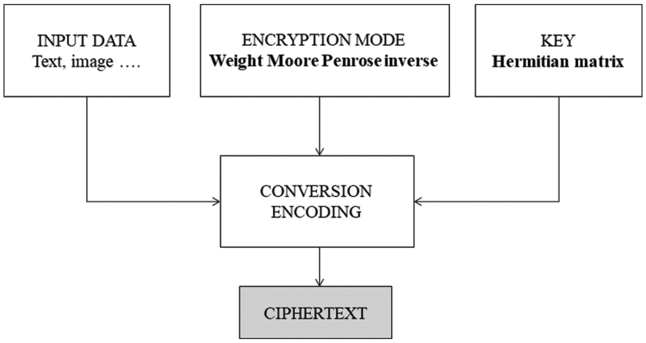


Figure 1: Proposed model framework

The method for data encryption has four phases (see the general model framework on Fig. 1):

1. Loading data (text or image)
2. Generation of Hermitian positive definite matrix (cryptographic key)
3. Converting the input data into a binary string.
4. Application of weighted Moore Penrose inverse in data encryption.

After converting the Base64 string into a binary record, is applied the weighted MP inverse where we use the Hermitian positive definite matrix (key), where obtained a new binary string which is converted to the Base64 string, which in this case represents the cipher of the image.

Example 1. Given is the image in PNG format and the Hermitian positive definite matrix 8 × 8 which presented key.

M_8 × 8 =

{{339, −87, −110, 119, 9, −41, −20, 10},

{−87, 514, −119, 10, 48, −55, −360, 45},

{−110, −119, 395, −225, −30, 81, −16, −129},

{119, 10, −225, 392, 43, −8, 180, 109},

{9, 48, −30, 43, 473, 93, −188, 90},

{−41, −55, 81, −8, 93, 552, −3, −44},

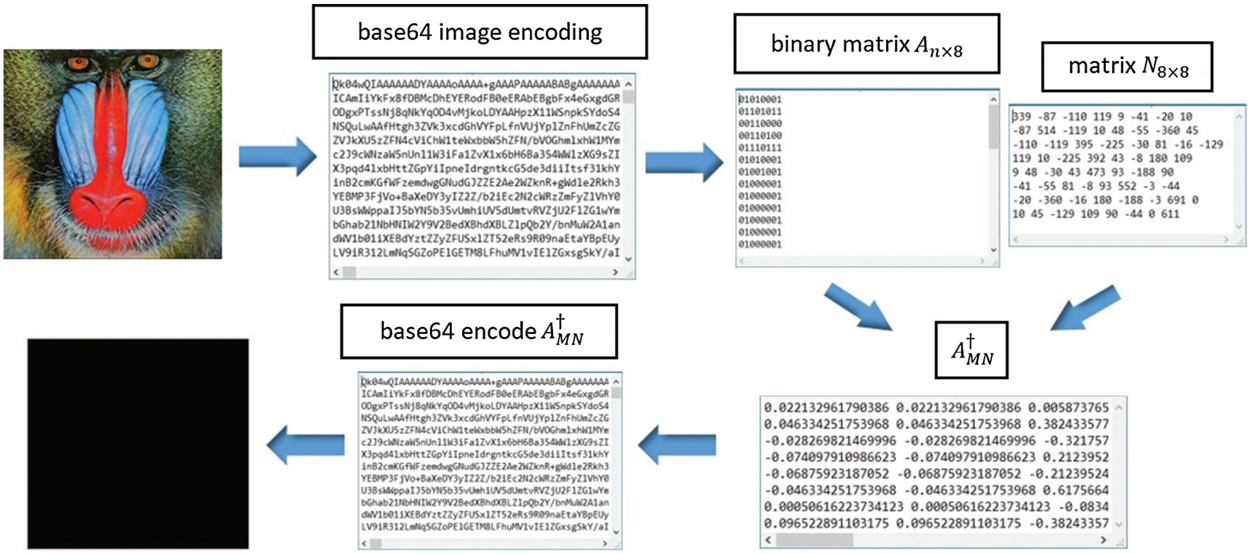
{−20, −360, −16, 180, −188, −3, 691, 0},

{10, 45, −129, 109, 90, −44, 0, 611}}.

The first phase is loading image and application of the Base64 image encoder converting of the received Base64 string into a binary string. Then, the next phase is the generation of the Hermitian positive definite matrix (in this case order 8).

The third phase is the application of the weighted Moore Penrose inverse in image encryption where we use the Hermitian positive definite matrix as key, where we get a binary string that represents the cipher of the image (see Fig. 2).

In the reverse case it is needed base64 encode $A^{\dagger}_{M,N}$ and Hermitian positive definite matrix. So, is needed loading of two input parameters: valid cryptographic key and the weighted MP inverse. If the key, or weighted MP inverse matrix $A^{\dagger}_{M,N}$ are not correct the image cannot be done (see Fig. 3).


base64 image encoding
binary matrix $A_{n\times8}$
matrix $N_{8\times8}$
$A^{\dagger}_{MN}$
base64 encode $A^{\dagger}_{MN}$

Qk04wQIAAAAAADYAAAAoAAAA+gAAAPAAAAABABgAAAAAAA
ICAmIiYkFx8fDBMcDhEYERodFB0eERAbEBgbFx4eGxgdGR
ODgxPTssNj8qNkYqOD4vMjkoLDYAAHpzX11WSnpkSYdoS4
NSQuLwAAfHtgh3ZVk3xcdGhVYFpLfnVUjYplZnFhUmZcZG
ZVJkXU5zZFN4cViChW1teWxbbW5hZFN/bVOGhmlxhW1MYm
c2J9cWNzaW5nUnl1W3iFa1ZvX1x6bH6Ba354WWlzXG9sZI
X3pqd4lxbHttZGpYiIpneIdrgntkcG5de3diiItsf31khY
inB2cmKGfWFzemdwgGNudGJZZE2Ae2WZknR+gWdle2Rkh3
YEBMP3FjVo+BaXeDY3yIZ2Z/b2iEc2N2cWRzZmFyZ1VhY0
U3BsWWppaIJ5bYN5b35vUmhiUV5dUmtvRVZjU2FlZG1wYm
bGhab21NbHNIW2Y9V2BedXBhdXBLZlpQb2Y/bnMuW2A1an
dWV1b01iXEBdYztZZyZFUSx1ZT52eRs9R09naEtaYBpEUy
LV9iR312LmNqSGZoPE1GETM8LFhuMV1vIE1ZGxsgSkY/aI

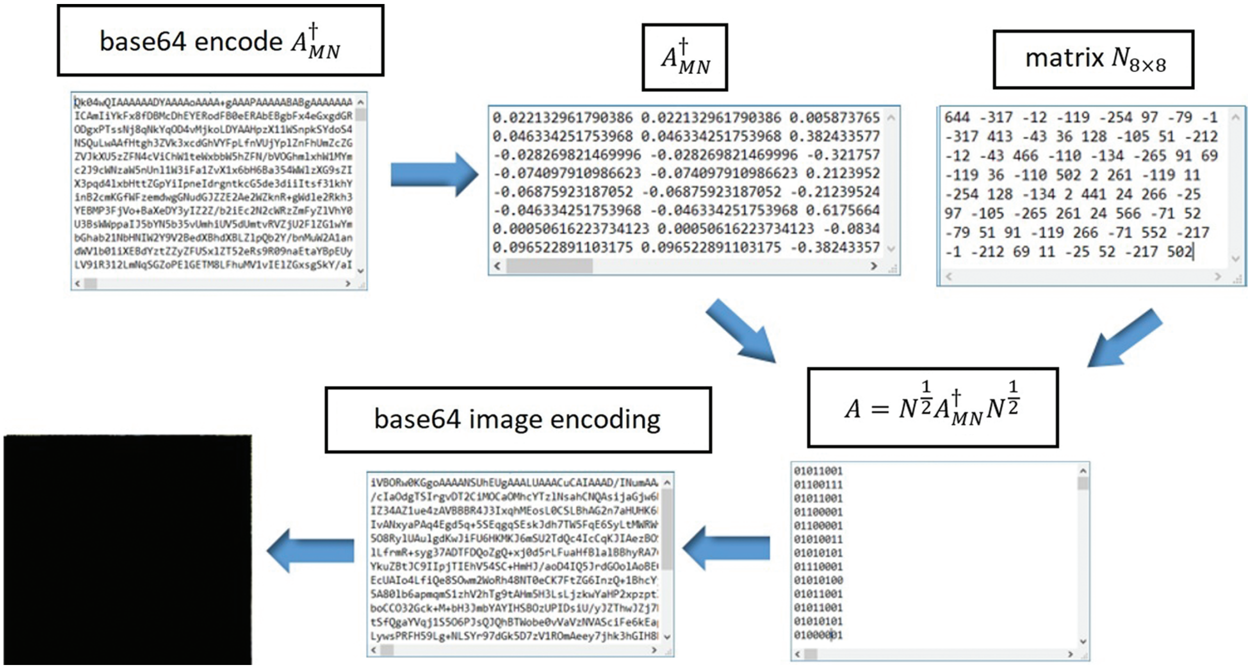
```
01010001
01101011
00110000
00110100
01110111
01010001
01001001
01000001
01000001
01000001
01000001
01000001
01000001
```

```
339 -87 -110 119 9 -41 -20 10
-87 514 -119 10 48 -55 -360 45
-110 -119 395 -225 -30 81 -16 -129
119 10 -225 392 43 -8 180 109
9 48 -30 43 473 93 -188 90
-41 -55 81 -8 93 552 -3 -44
-20 -360 -16 180 -188 -3 691 0
10 45 -129 109 90 -44 0 611
```

```
0.022132961790386 0.022132961790386 0.005873765
0.046334251753968 0.046334251753968 0.382433577
-0.028269821469996 -0.028269821469996 -0.321757
-0.074097910986623 -0.074097910986623 0.2123952
-0.06875923187052 -0.06875923187052 -0.21239524
-0.046334251753968 -0.046334251753968 0.6175664
0.00050616223734123 0.00050616223734123 -0.0834
0.096522891103175 0.096522891103175 -0.38243357
```

Qk04wQIAAAAAADYAAAAoAAAA+gAAAPAAAAABABgAAAAAAA
ICAmIiYkFx8fDBMcDhEYERodFB0eERAbEBgbFx4eGxgdGR
ODgxPTssNj8qNkYqOD4vMjkoLDYAAHpzX11WSnpkSYdoS4
NSQuLwAAfHtgh3ZVk3xcdGhVYFpLfnVUjYplZnFhUmZcZG
ZVJkXU5zZFN4cViChW1teWxbbW5hZFN/bVOGhmlxhW1MYm
c2J9cWNzaW5nUnl1W3iFa1ZvX1x6bH6Ba354WWlzXG9sZI
X3pqd4lxbHttZGpYiIpneIdrgntkcG5de3diiItsf31khY
inB2cmKGfWFzemdwgGNudGJZZE2Ae2WZknR+gWdle2Rkh3
YEBMP3FjVo+BaXeDY3yIZ2Z/b2iEc2N2cWRzZmFyZ1VhY0
U3BsWWppaIJ5bYN5b35vUmhiUV5dUmtvRVZjU2FlZG1wYm
bGhab21NbHNIW2Y9V2BedXBhdXBLZlpQb2Y/bnMuW2A1an
dWV1b01iXEBdYztZZyZFUSx1ZT52eRs9R09naEtaYBpEUy
LV9iR312LmNqSGZoPE1GETM8LFhuMV1vIE1ZGxsgSkY/aI

Figure 2: Image encoding procedure using weighted MP inverse


base64 encode $A^{\dagger}_{MN}$
$A^{\dagger}_{MN}$
matrix $N_{8\times8}$
$A = N^{\frac{1}{2}}A^{\dagger}_{MN}N^{\frac{1}{2}}$
base64 image encoding

Qk04wQIAAAAAADYAAAAoAAAA+gAAAPAAAAABABgAAAAAAA
ICAmIiYkFx8fDBMcDhEYERodFB0eERAbEBgbFx4eGxgdGR
ODgxPTssNj8qNkYqOD4vMjkoLDYAAHpzX11WSnpkSYdoS4
NSQuLwAAfHtgh3ZVk3xcdGhVYFpLfnVUjYplZnFhUmZcZG
ZVJkXU5zZFN4cViChW1teWxbbW5hZFN/bVOGhmlxhW1MYm
c2J9cWNzaW5nUnl1W3iFa1ZvX1x6bH6Ba354WWlzXG9sZI
X3pqd4lxbHttZGpYiIpneIdrgntkcG5de3diiItsf31khY
inB2cmKGfWFzemdwgGNudGJZZE2Ae2WZknR+gWdle2Rkh3
YEBMP3FjVo+BaXeDY3yIZ2Z/b2iEc2N2cWRzZmFyZ1VhY0
U3BsWWppaIJ5bYN5b35vUmhiUV5dUmtvRVZjU2FlZG1wYm
bGhab21NbHNIW2Y9V2BedXBhdXBLZlpQb2Y/bnMuW2A1an
dWV1b01iXEBdYztZZyZFUSx1ZT52eRs9R09naEtaYBpEUy
LV9iR312LmNqSGZoPE1GETM8LFhuMV1vIE1ZGxsgSkY/aI

```
0.022132961790386 0.022132961790386 0.005873765
0.046334251753968 0.046334251753968 0.382433577
-0.028269821469996 -0.028269821469996 -0.321757
-0.074097910986623 -0.074097910986623 0.2123952
-0.06875923187052 -0.06875923187052 -0.21239524
-0.046334251753968 -0.046334251753968 0.6175664
0.00050616223734123 0.00050616223734123 -0.0834
0.096522891103175 0.096522891103175 -0.38243357
```

```
644 -317 -12 -119 -254 97 -79 -1
-317 413 -43 36 128 -105 51 -212
-12 -43 466 -110 -134 -265 91 69
-119 36 -110 502 2 261 -119 11
-254 128 -134 2 441 24 266 -25
97 -105 -265 261 24 566 -71 52
-79 51 91 -119 266 -71 552 -217
-1 -212 69 11 -25 52 -217 502
```

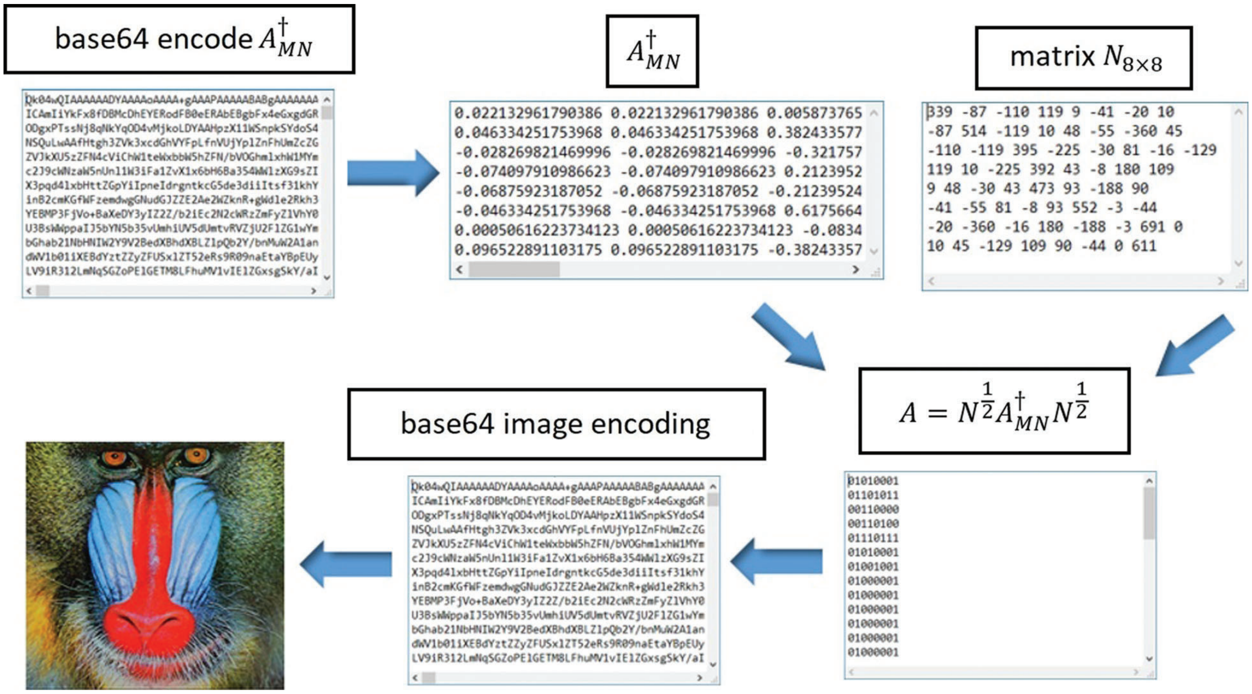
```
01011001
01100111
01011001
01100001
01100001
01010011
01010101
01110001
01010100
01011001
01011001
01010101
01000001
```

iVBORw0KGgoAAAANSUhEUgAAALUAAACuCAIAAAD/INumAA
/cIaOdgTSIrgvDT2CiMOCaOMhcYTzlNsahCNQAsijaGjw6
IZ34AZ1ue4zAVBBBR4J3IxqhMEosL0CSLBhAG2n7aHUHK6
IvANxyaPAq4Egd5q+5SEqgqSEskJdh7TW5FqE6SyLtMWRW
5O8Ry1UAulgdKwJiFU6HKMKJ6mSU2TdQc4IcCqKJIAezBO
1LfrmR+syg37ADTFDQoZgQ+xj0d5rLFuaHfBlalBBhyRA7
YkuZBtJC9IIpjTIEhV54SC+HmHJ/aoD4IQ5JrdGOolAoBE
EcUAIo4LfiQe8SOwm2WoRh48NT0eCK7FtZG6InzQ+1BhcY
5A80lb6apmqmS1zhV2hTg9tAHmSH3LsLjzkwYaHP2xpzpt
boCCO32Gck+M+bH3JmbYAYIHSBOzUPIDs1U/yJZThwJZj7
tSfQgaYVqj1S5O6PJsQJQhBTWobe0vVaVzNVASciFe6kEa
LywsPRFH59Lg+NLSYr97dGk5D7zV1ROmAeey7jhk3hGIH8

Figure 3: An example of using an invalid weighted MP inverse matrix or key

In other cases, if we use all correct parameters, then image decoding is successful (see Fig. 4).

In our paper [35], the weighted MP inverse and LM inverse relationship are analyzed and it shows that these are techniques that give the same result. Consequently, our proposed method for data encryption can also be used by the LM inverse. On the other hand, the key must be of type Hermitian matrix (positive definite) according to the theorem for the weight MP inverse.


base64 encode $A^{\dagger}_{MN}$
$A^{\dagger}_{MN}$
matrix $N_{8\times 8}$
$A = N^{\frac{1}{2}} A^{\dagger}_{MN} N^{\frac{1}{2}}$
base64 image encoding

```
Qk04wQIAAAAAADYAAAAoAAAA+gAAAPAAAAABABgAAAAAAA
ICAmIiYkFx8fDBMcDhEYERodFB0eERAbEBgbFx4eGxgdGR
ODgxPTssNj8qNkYqOD4vMjkoLDYAAHpzX11WSnpkSYdoS4
NSQuLwAAfHtgh3ZVk3xcdGhVYFpLfnVUjYplZnFhUmZcZG
ZVJkXU5zZFN4cViChW1teWxbbW5hZFN/bVOGhmlxhW1MYm
c2J9cWNzaW5nUnl1W3iFa1ZvX1x6bH6Ba354WW1zXG9sZI
X3pqd41xbHttZGpYiIpneIdrgntkcG5de3diiItsf31khY
inB2cmKGfWFzemdwgGNudGJZZE2Ae2WZknR+gWdle2Rkh3
YEBMP3FjVo+BaXeDY3yIZ2Z/b2iEc2N2cWRzZmFyZ1VhY0
U3BsWWppaIJ5bYN5b35vUmhiUV5dUmtvRVZjU2F1ZG1wYm
bGhab21NbHNIW2Y9V2BedXBhdXBLZ1pQb2Y/bnMuW2A1an
dWV1b01iXEBdYztZZyZFUSx1ZT52eRs9R09naEtaYBpEUy
LV9iR312LmNqSGZoPE1GETM8LFhuMV1vIE1ZGxsgSkY/aI
```

```
0.022132961790386 0.022132961790386 0.005873765
0.046334251753968 0.046334251753968 0.382433577
-0.028269821469996 -0.028269821469996 -0.321757
-0.074097910986623 -0.074097910986623 0.2123952
-0.06875923187052 -0.06875923187052 -0.21239524
-0.046334251753968 -0.046334251753968 0.6175664
0.00050616223734123 0.00050616223734123 -0.0834
0.096522891103175 0.096522891103175 -0.38243357
```

```
339 -87 -110 119 9 -41 -20 10
-87 514 -119 10 48 -55 -360 45
-110 -119 395 -225 -30 81 -16 -129
119 10 -225 392 43 -8 180 109
9 48 -30 43 473 93 -188 90
-41 -55 81 -8 93 552 -3 -44
-20 -360 -16 180 -188 -3 691 0
10 45 -129 109 90 -44 0 611
```

```
01010001
01101011
00110000
00110100
01110111
01010001
01001001
01000001
01000001
01000001
01000001
01000001
01000001
```

```
Qk04wQIAAAAAADYAAAAoAAAA+gAAAPAAAAABABgAAAAAAA
ICAmIiYkFx8fDBMcDhEYERodFB0eERAbEBgbFx4eGxgdGR
ODgxPTssNj8qNkYqOD4vMjkoLDYAAHpzX11WSnpkSYdoS4
NSQuLwAAfHtgh3ZVk3xcdGhVYFpLfnVUjYplZnFhUmZcZG
ZVJkXU5zZFN4cViChW1teWxbbW5hZFN/bVOGhmlxhW1MYm
c2J9cWNzaW5nUnl1W3iFa1ZvX1x6bH6Ba354WW1zXG9sZI
X3pqd41xbHttZGpYiIpneIdrgntkcG5de3diiItsf31khY
inB2cmKGfWFzemdwgGNudGJZZE2Ae2WZknR+gWdle2Rkh3
YEBMP3FjVo+BaXeDY3yIZ2Z/b2iEc2N2cWRzZmFyZ1VhY0
U3BsWWppaIJ5bYN5b35vUmhiUV5dUmtvRVZjU2F1ZG1wYm
bGhab21NbHNIW2Y9V2BedXBhdXBLZ1pQb2Y/bnMuW2A1an
dWV1b01iXEBdYztZZyZFUSx1ZT52eRs9R09naEtaYBpEUy
LV9iR312LmNqSGZoPE1GETM8LFhuMV1vIE1ZGxsgSkY/aI
```

Figure 4: An example of correct using of MP inverse

In order for this encryption method to provide a high secrecy we used prescribed statistic NIST tests. NIST tests is applied only binary sequences. Therefore, in our testing, we first need to convert the cryptological key to binary (matrix from Example 1). The NIST quality assurance test results for randomly generated matrix (cryptographic key) are given in Tab. 1.

Table 1: NIST quality assurance tests for random generated Hermitian matrix

NIST—Quality assurance test	8 × 8 random matrix (2504 bits key)
Frequency test	P = 0.8752, success
Block frequency test	P = 0.7851, success
Runs test	P = 0.0128, success
Longest runs of one's test	P = 0.0121, success
FFT—Fourier transform	P = 0.4877, success
Non-periodic templates	P = 0.1893, success
Linear complexity	P = 0.2896, success
Serial test *(P1/P2)*	P1 = 0.0111, success; P2 = 0.0127, success
Cumulative sums *(Forward/reverse)*	Forward = 0.7999, success; Reverse = 0.7985, success

After the test, we can conclude that all tests met the condition $P \geq 0.01$ (a condition for a binary sequence to be considered random, set by NIST). The fundamental terms of information theory, such as entropy, relative entropy, and mutual information are defined as probability distribution functions. These functions well describe the behavior of random variables of long sequences. We conducted additional testing for approximate entropy of a randomly generated Hermitian matrix (see Tab. 2).

Table 2: Approximate entropy for random generated Hermitian matrix

Input parameters	*ApEn*	χ^2	*P*
N1 = 512 (first matrix)	0.05281	77.002	0.9841
N2 = 512 (second matrix)	0.11077	138.2502	0.9217

Examination of random digression test is a series of eight tests (and conclusions), i.e., one test and a conclusion for each of the states: −4, −3, −2, −1 and +1, +2, +3, +4. For seven states, is $P \geq 0.01$, which leads to the conclusion that the binary sequence for the Hermitian matrix is random (see Tab. 3).

In the additional testing of the quality of the random matrix generated, we can conclude that the results of our advanced analysis (such as *approximate entropy and random digression)* satisfy the *NIST* requirements.

Table 3: Examination of random digression test

Input parameters	State (x)	Output P	Conclusion
ε = 1000000 bits binary extension n = 1000^2 J = 8 × 8 = 64 × 8 = 512 bits	−4	0.1789	*Random*
	−3	0.1745	*Random*
	−2	0.1425	*Random*
	−1	0.0110	*Random*
	+1	0.2563	*Random*
	+2	0.1078	*Random*
	+3	0.3327	*Random*
	+4	0.1996	*Random*

5 Performance Evaluation and Experimental Results

Machine learning algorithms could be compared by achieved results of classification concentrating on classes. Finding classification performance is a challenging part if we use inadequate data. By comparing the means of misclassified instances, we can make a comparison between machine learning methods. Several machine learning methods are used in order to distinguish two types of ciphertexts: (1) encrypted by the AES algorithm, and (2) encrypted by the proposed encryption method based on weighted MP inverse. The basic questions of the analysis are:

1. Is it possible to identify the type of encryption method by machine learning models learned only from information in encrypted text?
2. Are there significant differences between the AES and the proposed MP encryption method?

Hence, the most commonly used measure which is not focused on different classes quantity of right labels is accuracy:

$$accuracy = \frac{tp + tn}{tp + fp + fn + tn}$$

On the other hand, two measures that distinctly approximate a classifier's presentation on diverse classes are $sensitivity = \frac{tp}{tp + fn}$ and $specificity = \frac{tn}{fp + tn'}$
where are correctly classified: *tp*—true positive; *tn*—true negative and misclassified: *fp*—false positive; *fn*—a false negative.

Specificity and sensitivity are mostly used performance measuring of complex data during classification. In a comparative analysis, we give results of classifying of AES algorithm and encryption method based on Moore–Penrose inverse, respectively. In this study, we used two datasets which are obtained by extraction and decoding of a message in combination with different machine learning techniques. In order to produce an efficient machine learning algorithm, which will be able to satisfy all requirements, we tested both datasets on different types of machine learning methods.

5.1 Experiment Without Feature Selection

In this experiment, we used both datasets in combination with different ensemble machine learning methods. The result of this experiment is given in the following tables. We apply different machine learning techniques on both datasets without any feature extraction. Results obtained in that way are presented in Tab. 4.

Table 4: Classification result without feature selection

Feature selection method	*Data set 1*			*Data set 2*		
	AES	*MP encrypt*	*AVG*	*AES*	*MP encrypt*	*AVG*
ADTree	62	54	58	54	62	58
AttributeSelectedClassifier	18	96	57	96	18	57
Random tree	52	60	56	50	40	45
Decision table	14	96	55	84	14	49
MultiBoostAB (DecisionStump)	48	62	55	62	48	55
ANN	36	52	44	42	36	39
SVM	36	42	39	42	38	40

As you can see from the table, the best result with an average accuracy of 58% is achieved using the *ADTree* classifier. Besides that, it is obvious that tree classifiers are giving much better accuracy than other types of classifiers. Ensemble classifiers as *AttributeSelectedClassifier* and *MultiBoost* are also achieving significant accuracy. *Artificial Neural Network* and *Support Vector Machine* are giving the worst accuracy, which means that this kind of database is not suitable for this kind of classifier. Another interesting result from this experiment with dataset 1 is that the accuracy for the MP Encryption label is much greater for most classifiers. When we apply *AttributeSelectedClassifier* and Decision table it is 96% which is great accuracy compared to average accuracy for those classifiers. It means that some classifiers are producing much better results when they classify one class compared to another. In terms of this, *AdTree* and Random Tree classifiers are achieving the best-balanced accuracy for both classes.

5.2 Experiment With Feature Selection

After we load data and check data distribution, we see that there is a lot of features which does not have any value for any row. So, we decided to apply some feature selection methods before we introduce classification methods. For the feature selection method, we applied *AttributeSelection*, which results in a much smaller number of features and at the same time slightly increase accuracy. In Tab. 5, we present a result which we achieved using the feature selection method before classification.

Table 5: Classification result with feature selection applied

Feature selection method	*Data set 1*			*Data set 2*		
	AES	*MP encrypt*	*AVG*	*AES*	*MP encrypt*	*AVG*
ADTree	22	96	59	96	22	59
AttributeSelectedClassifier	18	98	58	98	18	58
Random tree	22	96	59	96	22	59
Decision table	14	98	56	98	14	56
MultiBoostAB (DecisionStump)	22	84	53	84	22	53
ANN	28	84	56	84	28	56
SVM	44	62	53	90	22	56

In Fig. 5 we present a comparison of average accuracy between classifiers when we don't apply feature selection on a dataset and when feature selection is applied.

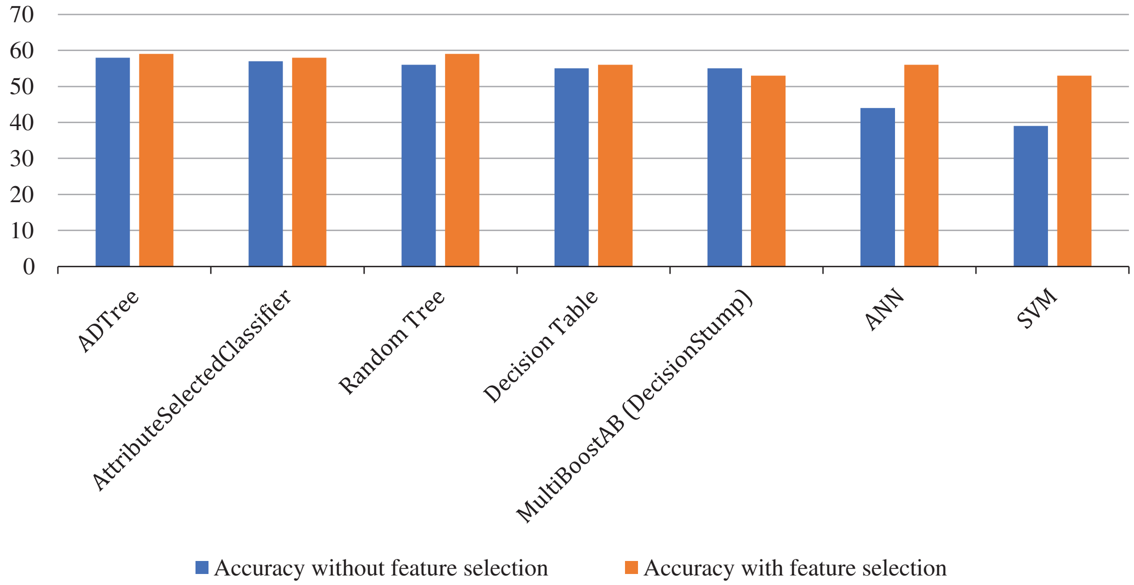


Figure 5: Comparison of accuracy between classification without and with feature selection

As you can see from the presented results, accuracy is increased for almost all machine learning methods which are applied. Some of the machine learning technologies achieved slightly better accuracy, for 1%, but for some of the methods, we achieved accuracy which is higher 14% than the previous one when we used all features. We have a similar situation in terms of accuracy distribution between classes, so again for MP Encrypt class, we have much greater accuracy in comparison with accuracy achieved for AES class.

When we use feature selected database, there are no classifiers which produce balanced accuracy for both classes. As it is obvious, all classifiers except *MultiBoost* are achieved better accuracy results when feature selection is applied. For some of the classifiers, we got significantly higher accuracy. When feature selection is applied, all proposed methods are achieving accuracy greater than 50% which is a great result compared with previous research on this topic. Besides that, time which is needed to build method and to test it is much smaller when we applied feature selection compared with the initial dataset.

6 Conclusion and Further Work

With the development of cloud and computer technologies, tools and software are being developed that violate the security of cloud computing resources. The layered cloud storage architecture is used primarily because different types and kinds of data may have different requirements in terms of storage. It is important to point out that there are often requirements related to encryption and data security. Mathematical systems found a wide application in encryption. The calculation of the weighted MP inverse represents one of those matrix system applications in cryptography. The basic precondition for developing of cryptologic systems with the public key is the efficient generation of a parameter which generates the key.

In this paper is presented a new idea in the form of applications matrix computations and generalized inverses in cryptography. We have provided a new way of encryption of text or images, where the whole process is based on the use of weighted MP inverse over the Hermitian positive definite matrix order 8 which presented key. The number of different combinations of the Hermitian positive definite matrices order 8 is huge so this solution represents a strong and secure key. Also, it was done performed the tests for the Hermitian positive definite matrices-keys generation through several aspects. In the experimental part of this paper, we give a comparison of encryption methods between machine learning methods. Machine learning algorithms could be compared by achieved results of classification concentrating on classes. In a comparative analysis, we give results of classifying of AES algorithm and encryption method based on Moore–Penrose inverse, respectively. Security problems are one of the most important issues related to cloud technologies. Data security and physical access to the location where the equipment was located needed to be constantly improved, as security threats to data and systems are becoming more serious day by day. Progress continues and more people are turning to these technologies because security will be improved without any doubt. Cloud computing is changing the business logic in the world. Due to the transition of the company to cloud computing, the client-server life on it will improve. Larger companies will need more time to move to cloud storage. Security issues are a big problem for them, as well as control over sensitive data.

The future of clouds will be slower in large companies as well as in large urban areas. The directions of our further development of the proposed method could refer to a connection with database management systems and matrix computations using PHP and MySQL technologies [39]. In order to ensure adequate levels of cloud data protection, appropriate mechanisms for cloud data warehouse security need to be established. In this case, cloud security automation enables the storage of data that can later be used for forensic analysis. Also, our future work could go in the direction of secure computation of the Moore–Penrose pseudo-inverse and its application to secure linear algebra, modeled by Cramer et al. [40]. New applications for smartphones will also appear. People and companies will access network software applications through a remote server. It's safe, the business will be based on applications that will be mostly accessed via cloud-enabled network devices.

Funding Statement: The authors would like to acknowledge the support of Network Communication Technology (NCT) Research Groups, FTSM, UKM in providing facilities for this research. This paper is supported under the Dana Impak Perdana UKM DIP-2018-040 and Fundamental Research Grant Scheme FRGS/1/2018/TK04/UKM/02/7.

Conflict of Interest: The authors declare that they have no conflicts of interest to report regarding the present study.

References

[1] M. H. Muhammed, M. M. Hashim, M. S. Taha, A. H. M. Aman, A. H. A. Hashim *et al.,* "Securing medical data transmission systems based on integrating algorithm of encryption and steganography," in *7th IEEE Int. Conf. on Mechatronics Engineering—ICOM'19*, Putrajaya, Malaysia, pp. 1–6, 2019.

[2] M. Marufuzzaman, K. Noorfazila, H. H. Fazida and B. I. R. Mamun, "Triple data encryption standard encryption engine: A hardware approach," in *Proc. of the 4th Int. Conf. on Computer Science & Computational Mathematics*, Langkawi, Malaysia, pp. 53–58, 2015.

[3] A. Zainalabideen, H. M. A. Azana and H. Rosilah, "Cloud query processing analysis: Encryption and decryption," *3C Tecnologia*, vol. 11, no. 3, pp. 64–75, 2019.

[4] S. Hasimi, C. R. Razli, I. Mohammad, F. Ahmad and B. Rogis, "Cloud computing implementation in the public sector: Factors and impacts," *Asia-Pacific Journal of Information Technology and Multimedia*, vol. 7, no. 2, pp. 27–42, 2018.

[5] A. S. Ahmed, R. Hassan and N. E. Othman, "Improving security for IPv6 neighbor discovery," in *Int. Conf. on Electrical Engineering and Informatics*, Denpasar, Indonesia, pp. 271–274, 2015.

[6] M. G. Aruna and K. G. Mohan, "Secured cloud data migration technique by competent probabilistic public key encryption," *China Communications*, vol. 17, no. 5, pp. 168–190, 2020.

[7] G. Viswanath and P. V. Krishna, "Hybrid encryption framework for securing big data storage in multi-cloud environment," *Evolutionary Intelligence*, vol. 13, no. 4, pp. 1–8, 2020.

[8] S. Kumaresan and V. Shanmugam, "Time-variant attribute-based multitype encryption algorithm for improved cloud data security using user profile," *Journal of Supercomputing*, vol. 76, no. 1, pp. 6094–6112, 2020.

[9] H. Deng, Z. Qin, Q. Wu, Z. Guan, R. Deng *et al.*, "Identity-based encryption transformation for flexible sharing of encrypted data in public cloud," *IEEE Transactions on Information Forensics and Security*, vol. 15, no. 1, pp. 3168–3180, 2020.

[10] G. S. Kumar and A. S. Krishna, "Data security for cloud datasets with bloom filters on ciphertext policy attribute based encryption," *International Journal of Information Security and Privacy*, vol. 13, no. 4, pp. 2–27, 2019.

[11] S. Tahir, S. Ruj, Y. Rahulamathavan, M. Rajarajan and C. Glackin, "A new secure and lightweight searchable encryption scheme over encrypted cloud data," *IEEE Transactions on Emerging Topics in Computing*, vol. 7, no. 4, pp. 530–544, 2019.

[12] P. K. Premkamal, S. K. Pasupuleti and P. J. A. Alphonse, "A new verifiable outsourced ciphertext-policy attribute based encryption for big data privacy and access control in cloud," *Journal of Ambient Intelligence and Humanized Computing*, vol. 10, no. 7, pp. 2693–2707, 2019.

[13] L. Munn, T. Hristova and L. Magee, "Clouded data: Privacy and the promise of encryption," *Big Data & Society*, vol. 6, no. 1, pp. 1–14, 2019.

[14] K. Lee, D. H. Lee, J. H. Park and M. Yung, "CCA security for self-updatable encryption: Protecting cloud data when clients read/write ciphertexts," *Computer Journal*, vol. 62, no. 4, pp. 545–562, 2019.

[15] S. Namasudra, "An improved attribute-based encryption technique towards the data security in cloud computing," *Concurrency and Computation—Practice & Experience*, vol. 31, no. 3, pp. 1–15, 2019.

[16] O. Zibouh, A. Dalli and H. Drissi, "A hybrid model encryption for enhancing data security in cloud computing," in *33rd IBIMA Conf.*, Granada, Spain, pp. 3840–3849, 2019.

[17] K. Rithvik, S. Kaur, S. Sejwal, P. Narwal and P. Jain, "Cloud computing data security using encryption algorithms," *IIOAB Journal*, vol. 10, no. 2, pp. 75–82, 2019.

[18] B. S. Al-Attab, H. S. Fadewar and M. E. Hodeish, "Lightweight effective encryption algorithm for securing data in cloud computing," *Advances in Intelligent Systems and Computing*, vol. 810, no. 1, pp. 105–121, 2019.

[19] W. Liu, Y. Xu, W. Liu, H. Wang and Z. Lei, "Quantum searchable encryption for cloud data based on full-blind quantum computation," *IEEE Access*, vol. 7, no. 1, pp. 186284–186295, 2019.

[20] M. Alsaedi, "Novel scheme for image encryption and decryption based on a Hermite-Gaussian matrix," *Advances in Intelligent Systems and Computing*, vol. 943, pp. 222–236, 2020.

[21] D. P. Jha, R. Kohli and A. Gupta, "Proposed encryption algorithm for data security using matrix properties," in *Int. Conf. on Innovation and Challenges in Cyber Security (ICICCS-INBUSH)*, Noida, India, pp. 86–90, 2016.

[22] M. Es-Sabry, N. El Akkad, M. Merras, A. Saaidi and K. Satori, "A new image encryption algorithm using random numbers generation of two matrices and bit-shift operators," *Soft Computing*, vol. 24, no. 5, pp. 3829–3848, 2020.

[23] W. Chuan-Kun and E. Dawson, "Generalized inverses in public key cryptosystem design," *IEEE Proceedings: Computers and Digital Techniques*, vol. 14, no. 5, pp. 321–326, 1998.

[24] R. E. Hartwig and J. Levine, "Applications of the Drazin inverse to the Hill cryptographic system," *Cryptologia*, vol. 5, no. 4, pp. 213–228, 1981.

[25] R. M. Kumar and S. S. Pradeep Kumar, "A secure encryption/decryption technique using transcendental number," *International Journal of Computer Trends and Technology*, vol. 29, no. 3, pp. 14–18, 2015.

[26] M. K. Viswanath and M. Ranjith Kumar, "A secure cryptosystem using the decimal expansion of an irrational number," *Applied Mathematical Sciences*, vol. 9, no. 106, pp. 5293–5303, 2015.

[27] B. G. Thapa, P. Lam-Estrada and J. López-Bonilla, "On the Moore–Penrose generalized inverse matrix," *World Scientific News*, vol. 9, no. 1, pp. 100–110, 2018.

[28] M. Güllüsaç, "Generalized inverses of matrices and applications to coding theory, Ph.D. Dissertation," Dokuz Eylül University, Turkey, 2016.

[29] B. Acharya, G. S. Rath and P. S. Kumar, "Novel modified Hill cipher algorithm," in *Proc. of ICETAETS*, Gujarat, India, pp. 1–7, 2008.

[30] M. K. Viswanath and M. Ranjith Kumar, "A public key cryptosystem using Hiil's cipher," *Journal of Discrete Mathematical Sciences and Cryptography*, vol. 18, no. 2, pp. 129–138, 2015.

[31] V. Katsikis and D. Pappas, "The restricted weighted generalized inverse of a matrix," *Electronic Journal of Linear Algebra*, vol. 22, no. 1, pp. 1156–1167, 2011.

[32] A. Ben-Israel and T. N. Greville, *Generalized Inverses: Theory and Applications*, 2nd ed. New York, USA: Springer-Verlag, pp. 71–84, 2003.

[33] G. R. Wang and Y. L. Chen, "A recursive algorithm for computing the weighted Moore–Penrose inverse A^{+}_{MN}," *Journal of Computational Mathematics*, vol. 4, no. 1, pp. 74–85, 1984.

[34] M. B. Tasić and P. S. Stanimirović, "Symbolic and recursive computation of different types of generalized inverses," *Applied Mathematics and Computation*, vol. 199, no. 1, pp. 349–367, 2008.

[35] M. B. Tasić, P. S. Stanimirović and S. H. Pepić, "About the generalized LM inverses and the Weighted Moore Penrose inverse," *Applied Mathematics and Computation*, vol. 216, no. 1, pp. 114–124, 2010.

[36] S. H. Pepić, "Weighted Moore–Penrose inverse: PHP *vs.* MATHEMATICA," *Facta Universitatis, Series: Mathematics and Informatics*, vol. 25, no. 1, pp. 35–45, 2020.

[37] D. Milošević, S. H. Pepić, M. Saračević and M. Tasić, "Weighted Moore–Penrose generalized matrix inverse: MySQL *vs.* Cassandra database storage system," *Sadhana: Academy Proceedings in Engineering Sciences*, vol. 41, no. 8, pp. 837–846, 2016.

[38] M. B. Tasić, P. S. Stanimirović and M. D. Petković, "Symbolic computation of weighted Moore–Penrose inverse using partitioning method," *Applied Mathematics and Computation*, vol. 189, no. 3, pp. 1317–1331, 2007.

[39] M. B. Tasić, P. S. Stanimirović and S. H. Pepić, "Computation of generalized inverses using PHP/MySQL environment," *International Journal of Computer Mathematics*, vol. 88, no. 11, pp. 2429–2446, 2011.

[40] R. Cramer, E. Kiltz and C. Padró, "A note on secure computation of the Moore–Penrose pseudoinverse and its application to secure linear algebra," in *Annual Int. Cryptology Conf. CRYPTO*, Santa Barbara, CA, USA, pp. 613–630, 2007.