Journal of King Saud University – Computer and Information Sciences 34 (2022) 627–635

Contents lists available at ScienceDirect

Journal of King Saud University – Computer and Information Sciences

journal homepage: www.sciencedirect.com

# A cryptographic algorithm using wavelet transforms over max-plus algebra

Subiono [a], Joko Cahyono [b], Dieky Adzkiya [a,*], Bijan Davvaz [c]

[a] Department of Mathematics, Institut Teknologi Sepuluh Nopember, Kampus ITS Sukolilo-Surabaya 60111, Indonesia
[b] Study Program of Informatics Engineering, Sekolah Tinggi Teknik Atlas Nusantara, Malang, Indonesia
[c] Department of Mathematics, Yazd University, Yazd, Iran

ARTICLE INFO

*Article history:*
Received 4 December 2019
Revised 14 February 2020
Accepted 20 February 2020
Available online 28 February 2020

*Keywords:*
Cryptography
Max-plus algebra
Wavelet transforms

ABSTRACT

Cryptography has a role to secure an important information. Until now, many varieties of cryptographic algorithms are available in the literature. In this paper, we propose a cryptographic algorithm based on Type IVa max-plus wavelet transforms (MP-Wavelets). Encryption and decryption algorithms are constructed based on the analysis and synthesis process of Type IVa MP-Wavelets, respectively. The encryption key contains the number of channels in all levels. The encryption key is chosen such that multiplication of the number of channels in all levels is greater than or equal to the number of characters in the Plaintext. The decryption key consists of the encryption key and a sequence generated by the binary encoding of detail components. This guarantees that the decryption key is very difficult to obtain using the brute-force method. The cryptographic process involves only maximization and addition operations as main operations. The experiments and analysis show that the algorithm is a good cryptographic algorithm based on the correlation between Plaintext and Ciphertext, encryption quality, the decryption key space, cryptanalysis (Ciphertext-only attack) and security analysis (entropy analysis, key sensitivity, Plaintext sensitivity). This algorithm is also efficient in the running time, because the complexity is linear w.r.t. the number of characters in the Plaintext.


## 1. Introduction

Cryptography has been associated with the problem of designing and analyzing encryption schemes, i.e. schemes that provide secret communication over insecure communication media. The problem of providing secret communication over insecure media is the most traditional and basic problem of cryptography. The setting consists of two parties communicating over insecure media. The encryption scheme is a protocol allowing these parties to communicate secretly with each other (Goldreich, 2001).

Typically, the encryption scheme consists of a pair of algorithms. The first algorithm, called encryption, is applied by the sender, i.e. the party sending a message. The other algorithm, called decryption, is applied by the receiver, i.e. the party receiving the message. Hence, in order to send a message, the sender first applies the encryption algorithm to the message and sends the result, called the Ciphertext, over the insecure media. Upon receiving a Ciphertext, the other party, i.e. the receiver, applies the decryption algorithm to the Ciphertext to retrieve the original message called the Plaintext. In order for this scheme to provide secret communication, the communicating parties must know the encryption and decryption keys (Goldreich, 2001).

Wavelet transforms have been widely used in signal processing. There are two processes in wavelet transforms: analysis and synthesis. A high resolution signal is decomposed by the analysis process to obtain the approximation and detail signals. The approximation signal represents the lower resolution approximation of the main signal. The detail signal ensures that the high resolution signal can be recovered by the synthesis process (Boggess and Narcowich, 2015). There are two types of wavelet transforms: discrete and continuous. The simplest discrete wavelet transform is Haar wavelet transform. In the literature, there are many

* Corresponding author.
*E-mail addresses:* subiono2008@matematika.its.ac.id (Subiono), j0k0_cahy0n0@yahoo.com (J. Cahyono), dieky@matematika.its.ac.id (D. Adzkiya), davvaz@yazd.ac.ir (B. Davvaz).

This work was supported by World Class Professor (WCP) program 2019, scheme B number T/77/D2.3/KK.04.05/2019.

Production and hosting by Elsevier

https://doi.org/10.1016/j.jksuci.2020.02.004
1319-1578/

varieties of cryptographic algorithms (Zhou et al., 2017; Zhou et al., 2018; Zhou et al., 2018; Zhou et al., 2018; Waqas et al., 2019; Khan et al., 2019). Some cryptographic algorithms leverage the wavelet transforms. Goswami et al. proposed a cryptographic algorithm using Daubechies wavelet transform (Goswami et al., 2011). Shankar and Elhoseny (2019) used Haar wavelet transforms for image security in wireless sensor networks. Sivasankari and Krishnaveni (2019) upgraded the image security level using optimal wavelet coefficients based steganography.

Max-plus algebra is an algebraic structure which uses two operations: maximization and addition over the set of real numbers (Baccelli et al., 1992; Heidergott et al., 2006). On the other hand, min-plus algebra uses minimization and addition operations. Classical works on these algebra focus on the computation of eigenvalue and eigenvectors (Fahim et al., 2017; van der Subiono, 2000). Such algebra has been widely used in many applications, such as scheduling (Kubo and Nishinari, 2018; Subiono et al., 2018), robotics (Lopes et al., 2014), cryptography (Durcheva, 2015; Grigoriev and Shpilrain, 2014) and wavelet transforms. With regards to the application of max-plus algebra to wavelet transforms, based on Haar wavelet transform, Nobuhara et al. constructed three types of wavelet transforms using max-plus algebra (MP-Wavelets): Type I, Type II and Type III (Bede and Nobuhara, 2009; Nobuhara et al., 2010). Then, Fahim and Yunus (2017) constructed two types of MP-Wavelets: Type IVa and Type IVb. Recently, Kannoth and Kumar (2018) proposed multi-image enhancement technique using max-plus algebra-based morphological wavelet transforms. The advantage of MP-Wavelets is that there are no floating point calculations, so the problem of round-off errors does not exist. Furthermore, MP-Wavelets are computationally simple and efficient.

In this paper, we develop a cryptographic algorithm using Type IVa MP-Wavelets. Type IVa is an extension of Type III in the sense that Type IVa is computationally more efficient than Type III. In our previous work (Cahyono and Subiono, 2016), we constructed a cryptographic algorithm using Type A MP-Wavelets. Type A and Type IVa MP-Wavelets are different in the analysis and synthesis process. Detail component of analysis operator of Type A is defined based on the preferred center pixel, whereas Type IVa is defined based on the difference of neighbor pixels. The analysis process in Type IVa MP-Wavelets is used for the encryption process, whereas the synthesis process is used for the decryption process. The encryption key contains the number of channels in all levels where the multiplication of the number of channels in all levels has to be greater than or equal to the number of characters in the Plaintext. The decryption key consists of the encryption key and a sequence of finitely many non-negative integers generated from the binary encoding of detail components. The feasibility of the proposed algorithm is analyzed. The analysis is conducted using the correlation value between the Plaintext and Ciphertext, encryption quality, decryption key space, running time, cryptanalysis (Ciphertext-only attack), security analysis (entropy analysis, key sensitivity, Plaintext sensitivity) and complexity analysis. The complexity of encryption and decryption algorithms is linear w.r.t. the length of Plaintext. We show that the number of possible decryption keys is very large, even if the Plaintext is short. This guarantees that the decryption key is very difficult to obtain by using the brute-force method. The cryptographic algorithm has been implemented in Scilab 5.5.2. We have applied the proposed encryption and decryption algorithms on some examples. According to the results of those examples, the encryption quality is close to the maximum quality, the correlation value is close to zero and the running time is fast.

The paper is structured as follows. Section 2 describes the models. Then the contributions of this paper are discussed in Section 3. The main contribution is the procedure of cryptographic algorithm using Type IVa MP-Wavelets. In Section 4, we analyze the proposed algorithm. Finally, Section 5 concludes the paper.

## 2. Models and preliminaries

In this section, we introduce max-plus algebra and Type IVa max-plus wavelet transforms (MP-Wavelets). We use these notions in the subsequent sections to construct a cryptographic algorithm.

### 2.1. Max-plus algebra

Max-plus algebra (Baccelli et al., 1992; Heidergott et al., 2006) is a class of discrete algebraic systems, also known as an effective tool for modeling and analyzing several types of discrete-event systems. We denote $\mathbb{R}$ as the set of real numbers. The max-plus algebra is defined as $\mathbb{R}_{\max} = (\mathbb{R}_\varepsilon, \oplus, \otimes)$ where $\mathbb{R}_\varepsilon \stackrel{\text{def}}{=} \mathbb{R} \cup \{\varepsilon\}$ and $\varepsilon \stackrel{\text{def}}{=} -\infty$. For every $x, y \in \mathbb{R}_\varepsilon$, the binary operators $\oplus$ and $\otimes$ are defined as follows:

$$x \oplus y \stackrel{\text{def}}{=} \max\{x, y\} \qquad \text{and} \qquad x \otimes y \stackrel{\text{def}}{=} x + y. \tag{1}$$

Thus in max-plus algebra, the addition and multiplication operations are replaced by maximization and the usual addition operation. The symbol $\varepsilon$ is the neutral element with respect to maximization $\oplus$. Similarly, the symbol $e = 0$ denotes the neutral element with respect to addition $\otimes$. In the context of max-plus algebra, $a^{\otimes b} = b \times a$, where $\times$ is the conventional multiplication operator (Heidergott et al., 2006).

### 2.2. Type IVa max-plus wavelet transforms

MP-Wavelets consist of two processes: analysis and synthesis, similar to ordinary wavelet transforms. We define $V_k : \mathbb{Z} \to \mathbb{Z}$ as the signal space at level $k$, where $k \geqslant 0$ and $\mathbb{Z}$ is the set of integers. Notice that $V_0$ represents the set of original signals and $V_k$ represents the set of approximation signals at level $k$ for $k \geqslant 1$. The detail signal at level $k$ consists of $p_k - 1$ channels, where $k \geqslant 1$. We define $W_{k,i} : \mathbb{Z} \to \mathbb{Z}$ as the set of detail signals at level $k$ channel $i$, where $1 \leqslant i \leqslant p_k - 1$ and $k \geqslant 1$. The analysis operator consists of approximation and detail components. The approximation component is defined as $\psi_k^{\uparrow} : V_k \to V_{k+1}$, where $k \geqslant 0$. The detail component for channel $i$ is defined as $\omega_{k,i}^{\uparrow} : V_k \to W_{k+1}$, where $1 \leqslant i \leqslant p_{k+1}$ and $k \geqslant 0$. The scheme of analysis process in MP-Wavelets is given in Fig. 1 (top). The synthesis operator is defined as $\Psi_k^{\downarrow} : V_{k+1} \times W_{k+1,1} \times \ldots \times W_{k+1,p_{k+1}-1} \to V_k$, where $k \geqslant 0$. The synthesis process in MP-Wavelets is shown in Fig. 1 (bottom).

In the remainder of this subsection, we discuss Type IVa MP-Wavelets. The analysis operator is defined by

$$\psi_k^{\uparrow}(x_k)[n] = x_{k+1}[n] = x_k[p_{k+1}n] \oplus x_k[p_{k+1}n + 1], \tag{2}$$

$$\omega_{k,i}^{\uparrow}(x_k)[n] = y_{k+1,i}[n] = x_k[p_{k+1}n + i] - x_k[p_{k+1}n + i - 1], \quad \text{for } i = 1, \ldots, p_{k+1} - 1, \tag{3}$$

where $n \in \mathbb{Z}$ and $k \geqslant 0$.

The synthesis operator in Type IVa MP-Wavelets is

$$\Psi_k^{\downarrow}(z_{k+1}, y_{k+1})[p_{k+1}n] = z_k[p_{k+1}n] = z_{k+1}[n] - (y_{k+1,1}[n] \oplus 0) \tag{4}$$

$$\Psi_k^{\downarrow}(z_{k+1}, y_{k+1})[p_{k+1}n + i] = z_k[p_{k+1}n + i] = y_{k+1,i}[n] \otimes z_k[p_{k+1}n + i - 1], \quad \text{for } i = 1, \ldots, p_{k+1} - 1, \tag{5}$$

where $y_{k+1} = (y_{k+1,1}, \ldots, y_{k+1,p_{k+1}-1}), n \in \mathbb{Z}$ and $k \geqslant 0$.

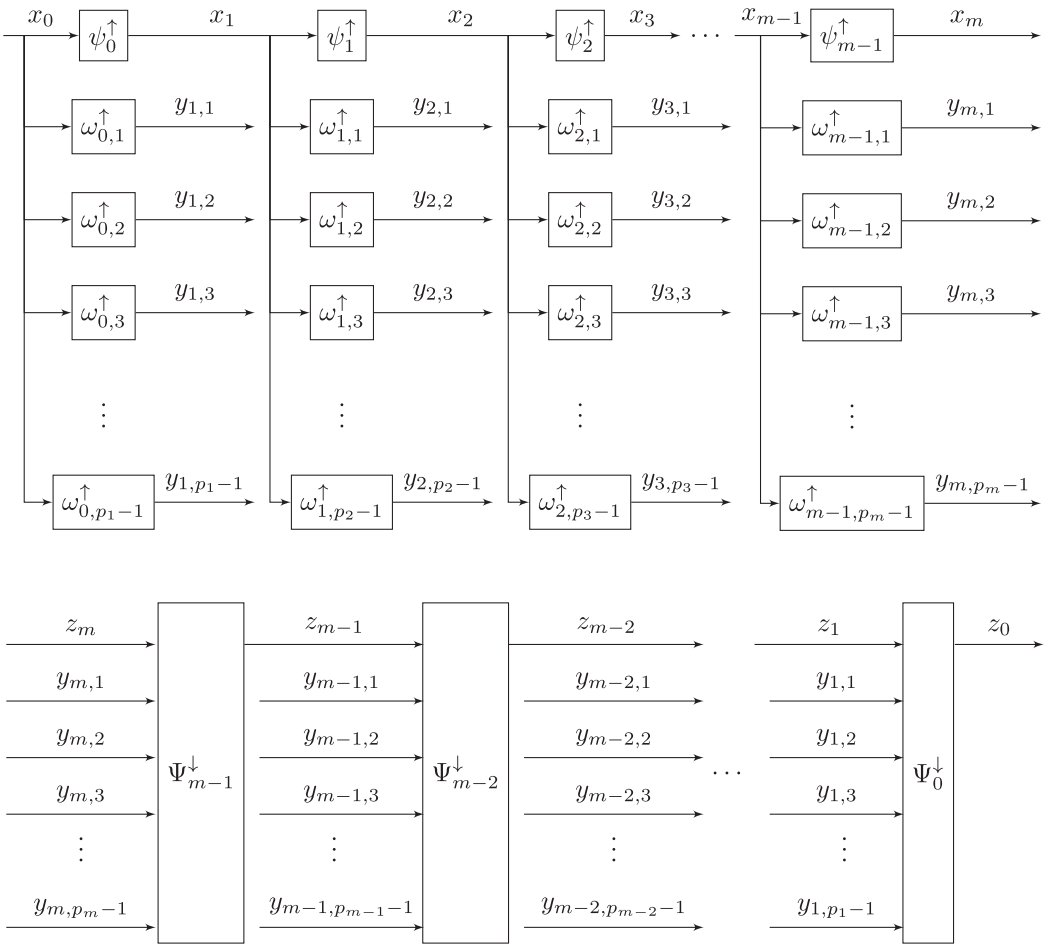

**Fig. 1.** The top and bottom plots display the analysis and synthesis process of MP-Wavelets, respectively.

## 3. Construction of cryptographic algorithm

In this section, we discuss a cryptographic algorithm based on Type IVa MP-Wavelets. The algorithm consists of the encryption technique, decryption key generation and decryption technique. Encryption and decryption algorithms are constructed based on the analysis and synthesis processes of Type IVa MP-Wavelets, respectively. In the end of this section, we explain the detailed steps in the algorithms via a simple example.

### 3.1. Encryption technique

Encryption is a process to transform a Plaintext into a Ciphertext, which is based on the analysis process in Type IVa MP-Wavelets. The encryption algorithm is as follows:

1. The Plaintext is converted into an ASCII code. The ASCII code of Plaintext is called PlainASCII $P$. The number of characters in the Plaintext is denoted by $N$. Thus, Plaintext $P$ is a sequence of finitely many non-negative integers denoted by $P[0], P[1], \ldots, P[N-1]$ where $P[0]$ is the ASCII code of the first character, $P[1]$ is the ASCII code of the second character, ..., $P[N-1]$ is the ASCII code of the last character. The original signal $x_0$ in Type IVa MP-Wavelets is defined as the PlainASCII $P$, i.e. $x_0[n] = P[n]$ for $n = 0, \ldots, N-1$.
2. We decide the encryption key. The encryption key is a sequence of finitely many positive integers $p_1, p_2, \ldots, p_m$ such that $\prod_{k=1}^{m} p_k \geqslant N$. The number $p_k$ represents the number of channels at level $k$, for $k = 1, \ldots, m$ (see Fig. 1 (top)). For simplicity, we denote $N' = \prod_{k=1}^{m} p_k$. It will be clear later that $N'$ represents the length of Ciphertext.
3. If $N' > N$, we extend the original signal $x_0$ such that the length becomes $N'$ by adding some space characters in the end, i.e. $x_0[n]$ is defined as the ASCII code of the space character for $n = N, \ldots, N'-1$. This is used to guarantee the fulfillment of pyramid condition or perfect reconstruction property.
4. The approximation and detail signals at level $1, \ldots, m$ are computed using the analysis process of Type IVa MP-Wavelets described in (2) and (3).
5. We construct the binary encoding of detail signals $y_{m,1}, y_{m,2}, \ldots, y_{m,p_m-1}, \ldots, y_{2,1}, y_{2,2}, \ldots, y_{2,p_2-1}, y_{1,1}, y_{1,2}, \ldots, y_{1,p_1-1}$ by using the following formula: negative value is encoded by 1 and positive value is encoded by 0. The binary encoding $bc$ will be used in the construction of decryption key.
6. The ASCII code of the Ciphertext is denoted by CipherASCII $C$. Thus CipherASCII $C$ is a sequence of finitely many non-negative integers. CipherASCII $C$ is defined as the approximation signal at level $m$ and the absolute value of detail signals added by 32, i.e. $x_m, |y_{m,1}| + 32, |y_{m,2}| + 32, \ldots, |y_{m,p_m-1}| + 32, \ldots, |y_{2,1}| + 32, |y_{2,2}| + 32, \ldots, |y_{2,p_2-1}| + 32, |y_{1,1}| + 32, |y_{1,2}| + 32, \ldots, |y_{1,p_1-1}| + 32$. We add 32 to the detail signals in order to guarantee that all ASCII codes are printable. Notice that all ASCII codes greater than or equal to 32 are printable.
7. The Ciphertext is obtained by converting the CipherASCII $C$ into text.

### 3.2. Generation of the decryption key

The decryption key consists of two parts. The first part is the encryption key. The second part is a sequence of finitely many non-negative integers generated from the binary encoding of detail signals *bc*. The sequence of finitely many non-negative integers is generated from the binary encoding *bc* (see step 5 of the encryption algorithm) in the following way. Starting from the left-most bit (most-significant bit), replace each group of 8 bits by the equivalent decimal digit (in the right-most group, pad the left-most bits with zero if necessary).

### 3.3. Decryption technique

Decryption is a process to transform a Ciphertext into a Plaintext, that is based on the synthesis process in Type IVa MP-Wavelets. The decryption procedure is as follows:

1. The Ciphertext is converted into an ASCII code. The obtained ASCII code is called CipherASCII $C$. Thus, CipherASCII $C$ is a sequence of finitely many non-negative integers.
2. The number of levels $m$ and the number of channels in all levels $p_1, p_2, \ldots, p_m$ are identified from the first part of the decryption key. We denote $N'$ as the multiplication of the number of channels in all levels $\prod_{k=1}^{m} p_k$, which is equal to the length of Ciphertext.
3. We reconstruct the binary encoding of detail components *bc* from the second part of the decryption key. Each number in the second part of the decryption key is converted into 8 bits binary number. Then we concatenate the binary number of each decimal number in the second part into a single binary number. If the length of the binary number is greater than $N'-1$, then we remove the leading zeros corresponding to the right-most group of 8 bits until the length becomes $N'-1$. The result is denoted by *bc*.
4. The approximation component $z_m[0]$ is defined as the first element of CipherASCII $C$, i.e. $z_m[0] = C[0]$. The rest of CipherASCII and the binary encoding are used to reconstruct the detail components in all levels by using the following formula:

$$\text{Detail Components } [n] = (C[n+1] - 32) \times (-1)^{bc[n]}, \qquad (6)$$

where $n = 0, \ldots, N'-2$. Then we identify the detail component in each level and each channel based on the following order $y_{m,1}, y_{m,2}, \ldots, y_{m,p_m-1}, \ldots, y_{2,1}, y_{2,2}, \ldots, y_{2,p_2-1}, y_{1,1}, y_{1,2}, \ldots, y_{1,p_1-1}$.
5. We execute the synthesis process of Type IVa MP-Wavelets based on (4) and (5) to reconstruct the approximation signals $z_{m-1}, z_{m-2}, \ldots, z_0$. From the synthesis process, the PlainASCII $P$ is defined as the original signal $z_0$, i.e. $P[n] = z_0[n]$ for $n = 0, \ldots, N'-1$.
6. The Plaintext can be retrieved by transforming the PlainASCII $P$ to their corresponding ASCII characters.

### 3.4. An illustrative example

In this section, we describe the encryption procedure, generation of the decryption key and the decryption procedure through a simple example. In this example, the sentence `Max-Plus Wavelet Cryptography.` is defined as the Plaintext. The encryption process of this Plaintext is as follows:

1. The Plaintext `Max-Plus Wavelet Cryptography.` is converted into an ASCII code. The obtained ASCII code is called PlainASCII $P$, i.e. $P[0] = 77, P[1] = 97, \ldots, P[29] = 46$. The number of characters in PlainASCII is equal to $N = 30$. The original signal $x_0$ is defined as $x_0[n] = P[n]$ for $n = 0, \ldots, 29$.
2. We choose $p_1 = 2, p_2 = 3, p_3 = 5$ as the encryption key. In this case, $m = 3$. We obtain $N' = 2 \times 3 \times 5 = 30$.
3. Since $N' = N$, the original signal remains the same.
4. The analysis process of Type IVa MP-Wavelets is executed according to (2) and (3). We describe the procedure to obtain the approximation and detail signals at level 1, 2, 3:
   (a) The number of channels in level 1 is $p_1 = 2$. We compute the approximation signal $x_1$ and detail signal $y_{1,1}$, as follows:
   - The approximation signal $x_1$ is computed using (2) for $k = 0$ and $n = 0, \ldots, 14$. For $k = 0$ and $n = 0$, we have $x_1[0] = x_0[0] \oplus x_0[1] = 77 \oplus 97 = 97$. By using the same formula, we obtain $x_1[1] = 120, x_1[2] = 108$, ..., $x_1[14] = 121$.
   - The detail signal $y_{1,1}$ is computed using (3) for $k = 0, i = 1$ and $n = 0, \ldots, 14$. For $k = 0, i = 1$ and $n = 0$, we have $y_{1,1}[0] = x_0[1] - x_0[0] = 97 - 77 = 20$. By using the previous formula, we obtain $y_{1,1}[1] = -75, y_{1,1}[2] = 28$, ..., $y_{1,1}[14] = -75$.

   (b) The number of channels in level 2 is $p_2 = 3$. In the following, we compute the approximation signal $x_2$ and detail signals $y_{2,1}, y_{2,2}$.
   - The approximation signal $x_2$ is computed using (2) for $k = 1$ and $n = 0, \ldots, 4$. For $k = 1$ and $n = 0$, we have $x_2[0] = x_1[0] \oplus x_1[1] = 97 \oplus 120 = 120$. By leveraging the same formula, we obtain $x_2[1] = 117, x_2[2] = 116, x_2[3] = 121, x_2[4] = 114$.
   - The detail signal $y_{2,1}$ is computed using (3) for $k = 1, i = 1$ and $n = 0, \ldots, 4$. For $k = 1, i = 1$ and $n = 0$, we have $y_{2,1}[0] = x_1[1] - x_1[0] = 120 - 97 = 23$. By using the preceding formula, we obtain $y_{2,1}[1] = -30, y_{2,1}[2] = 8, y_{2,1}[3] = -5, y_{2,1}[4] = -2$.
   - The detail signal $y_{2,2}$ is also computed using (3) for $k = 1, i = 2$ and $n = 0, \ldots, 4$. For $k = 1, i = 2$ and $n = 0$, we have $y_{2,2}[0] = x_1[2] - x_1[1] = 108 - 120 = -12$. By using the previous formula, we obtain $y_{2,2}[1] = 31, y_{2,2}[2] = -49, y_{2,2}[3] = -5, y_{2,2}[4] = 9$.

   (c) The number of channels in level 3 is $p_3 = 5$. In what follows, we compute the approximation signal $x_3$ and detail signals $y_{3,1}, y_{3,2}, y_{3,3}, y_{3,4}$:
   - The approximation signal $x_3$ is obtained using (2) for $k = 2$ and $n = 0$, i.e. $x_3[0] = x_2[0] \oplus x_2[1] = 120 \oplus 117 = 120$.
   - The detail signals $y_{3,1}, y_{3,2}, y_{3,3}, y_{3,4}$ are computed using (3) for $k = 2, n = 0$ and $i = 1, \ldots, 4$. For $k = 2, n = 0$ and $i = 1$, we have $y_{3,1}[0] = x_2[1] - x_2[0] = 117 - 120 = -3$. By using the same formula, we obtain $y_{3,2}[0] = -1, y_{3,3}[0] = 5, y_{3,4}[0] = -7$.

5. The detail components are merged in the following order $y_{3,1}[0], y_{3,2}[0], y_{3,3}[0], y_{3,4}[0], y_{2,1}[0], y_{2,2}[0], y_{2,1}[1], y_{2,2}[1], y_{2,1}[2], y_{2,2}[2], y_{2,1}[3], y_{2,2}[3], y_{2,1}[4], y_{2,2}[4], y_{1,1}[0], y_{1,1}[1], y_{1,1}[2], y_{1,1}[3], y_{1,1}[4], y_{1,1}[5], y_{1,1}[6], y_{1,1}[7], y_{1,1}[8], y_{1,1}[9], y_{1,1}[10], y_{1,1}[11], y_{1,1}[12], y_{1,1}[13], y_{1,1}[14]$. The binary encoding *bc* of detail components is $bc[0] = 1, bc[1] = 1$, ..., $bc[28] = 1$.
6. The CipherASCII $C$ is defined as $C[0] = x_3[0] = 120, C[1] = |y_{3,1}[0]| + 32 = 35, C[2] = |y_{3,2}[0]| + 32 = 33$, ..., $C[29] = |y_{1,1}[14]| + 32 = 107$.
7. The Ciphertext is obtained by converting the CipherASCII $C$ to the ASCII characters, i.e. `x#!%'7,> ?(Q%%")4k<"W5'/C'$(l(k`.

Next we discuss the generation of decryption key, which consists of encryption key and a sequence generated by the binary encoding of detail components. The encryption key is 2, 3, 5. The binary encoding of detail components is given by 11010110011110010100000001111. Starting from the left, we divide the binary encoding into 4 groups: 11010110, 01111001, 01000000, 01111. Since the last group contains 5 digits, we add three leading zeros to the last group, that is 11010110, 01111001, 01000000, 00001111. The sequence is obtained by converting each group into a decimal number, which produces 214, 121, 64, 15. The decryption key is 2, 3, 5, 214, 121, 64, 15.

Now we describe the decryption process using Ciphertext obtained in the previous step `x#!%'7,> ?(Q%%")4k<"W5'/C'$(l(k` and decryption key 2, 3, 5, 214, 121, 64, 15. The decryption process of this Ciphertext is as follows:

1. The Ciphertext is converted into an ASCII code. The obtained ASCII code is called CipherASCII $C$, i.e. $C[0] = 120, C[1] = 35, \ldots, C[29] = 107$.
2. The number of levels is $m = 3$. The number of channels in level 1, 2 and 3 is $p_1 = 2, p_2 = 3$ and $p_3 = 5$, respectively. We obtain $N' = 2 \times 3 \times 5 = 30$.
3. Each number in the second part of decryption key 214, 121, 64, 15 is converted to 8 bits binary number. We obtain 11010110, 01111001, 01000000, 00001111. Since the total number of digits in the four binary number is 32, i.e. greater than 29, then we reduce three leading zeros corresponding to the last binary number. The binary numbers become 11010110, 01111001, 01000000, 01111. Then we merge the four binary numbers, i.e. 11010110011110010100000001111. The binary number is denoted as $bc$, i.e. $bc[0] = 1, bc[1] = 1, \ldots, bc[28] = 1$.
4. The signal $z_3[0]$ is defined as the first element in CipherASCII, i.e. $z_3[0] = C[0] = 120$. By using (6), we obtain detail components $y_{3,1}[0] = -3, y_{3,2}[0] = -1, \ldots, y_{1,1}[14] = -75$. Notice that, for each channel, the detail component in level 3, 2 and 1 is a sequence of 1, 5 and 15 integers.
5. The synthesis process of Type IVa MP-Wavelets described in (4) and (5) is executed. We describe the procedure to obtain $z_2, z_1, z_0$.
   (a) We determine the signal $z_2$ by using (4) and (5). Initially, we substitute $k = 2$ and $n = 0$ to (4), i.e. $z_2[0] = z_3[0] - (y_{3,1}[0] \oplus 0) = 120 - (-3 \oplus 0) = 120$. Then we substitute $k = 2, n = 0$ and $i = 1, \ldots, 4$ to (5). For $k = 2, n = 0$ and $i = 1$, we have $z_2[1] = y_{3,1}[0] \otimes z_2[0] = -3 \otimes 120 = 117$. By using the same formula, we obtain $z_2[2] = 116, z_2[3] = 121, z_2[4] = 114$.
   (b) We determine the signal $z_1$ by substituting $k = 1$ and $n = 0, \ldots, 4$ to (4) and for (5), additionally we use $i = 1, 2$. If we substitute $k = 1$ and $n = 0$ to (4), we obtain $z_1[0] = z_2[0] - (y_{2,1}[0] \oplus 0) = 120 - (23 \oplus 0) = 97$. When we substitute $k = 1, n = 0$ and $i = 1$ to (5), we obtain $z_1[1] = y_{2,1}[0] \otimes z_1[0] = 23 \otimes 97 = 120$. If we substitute $k = 1, n = 0$ and $i = 2$ to (5), we obtain $z_1[2] = y_{2,2}[0] \otimes z_1[1] = -12 \otimes 120 = 108$. If we continue the procedure in this way, we have $z_1[3] = 117, z_1[4] = 87, \ldots, z_1[14] = 121$.
   (c) We compute the signal $z_0$ by substituting $k = 0$ and $n = 0, \ldots, 14$ to (4) and we also substitute $i = 1$ to (5). If we substitute $k = 0$ and $n = 0$ to (4), we have $z_0[0] = z_1[0] - (y_{1,1}[0] \oplus 0) = 97 - (20 \oplus 0) = 77$. When we substitute $k = 0, n = 0$ and $i = 1$ to (5), we obtain $z_0[1] = y_{1,1}[0] \otimes z_0[0] = 20 \otimes 77 = 97$. If we continue the procedure, we have $z_0[2] = 120, z_0[3] = 45, \ldots, z_0[29] = 46$. The PlainASCII $P$ is defined as the obtained signal $z_0$, i.e. $P[n] = z_0[n]$ for $n = 0, \ldots, 29$.
6. The last step is transforming the ASCII code in PlainASCII $P$ to text. We obtain `Max-Plus Wavelet Cryptography.` as the Plaintext.

## 4. Empirical study, cryptanalysis and complexity

In this section, we analyze the cryptographic algorithm analytically and empirically. The empirical study consists of the correlation between Plaintext and Ciphertext, the encryption quality and the running time. In the empirical study of the cryptographic algorithm, we use 9 case studies, where the number of characters is between 30 and 20723. The analytical study consists of key space analysis, cryptanalysis, security analysis and time complexity analysis. For the cryptanalysis, we discuss the ciphertext-only attack. With regards to the security analysis, we discuss entropy analysis, key sensitivity and Plaintext sensitivity.

### 4.1. The correlation between plaintext and ciphertext

This correlation analysis is conducted to determine the level of linear relationship between Plaintext and Ciphertext. The correlation coefficient is computed using the following formula (Arul and Venkatesulu, 2012):

$$r = \frac{N'\sum_{n=0}^{N'-1}(P[n]C[n]) - \sum_{n=0}^{N'-1}P[n]\sum_{n=0}^{N'-1}C[n]}{\sqrt{\left(N'\sum_{n=0}^{N'-1}(P[n]^2) - \left(\sum_{n=0}^{N'-1}P[n]\right)^2\right)\left(N'\sum_{n=0}^{N'-1}(C[n]^2) - \left(\sum_{n=0}^{N'-1}C[n]\right)^2\right)}} \tag{7}$$

where $P[n]$ and $C[n]$ are ASCII codes of $(n+1)$-th character in Plaintext and Ciphertext respectively, $N'$ is the length of Ciphertext, and $r$ is the correlation coefficient between Plaintext and Ciphertext. If the correlation coefficient is close to 1 or $-1$, Ciphertext and Plaintext have strong linear relationship. If the correlation coefficient is close to 0, Ciphertext and Plaintext have a weak linear relationship (Walpole, 1982).

The correlation coefficient between the Plaintext and the resulting Ciphertext are shown in Table 1. We compare the correlation coefficient of Type IVa and Type A MP-Wavelets for each Plaintext file.

In the case studies, for Type IVa MP-Wavelets, the correlation coefficients for all Plaintext files are between $-0.5$ and 0.06. If we use Type A MP-Wavelets, the correlation coefficients for all Plaintext files are between $-0.4$ and 0.04. Thus, the linear relationship between Plaintext and Ciphertext is weak for both Type IVa and Type A MP-Wavelets.

### 4.2. The encryption quality

The encryption quality analysis is conducted by comparing the number of occurrences of each letter in the Plaintext and Ciphertext. The encryption quality represents the average number of changes of each letter, that can be expressed mathematically as (Arul and Venkatesulu, 2012):

$$EQ = \frac{\sum_{L=32}^{126} |\, H_L(C) - H_L(P) \,|}{95} \tag{8}$$

**Table 1**
The correlation coefficient between Plaintext and Ciphertext for the cryptographic algorithm based on Type IVa and Type A MP-Wavelets.

| Plaintext files | Number of characters | Encryption key | The Correlation coefficient | |
|---|---|---|---|---|
| | | | Type IVa | Type A |
| plain1.txt | 30 | 2 3 5 | −0.4836257 | −0.3576465 |
| plain2.txt | 300 | 3 2 5 2 5 | 0.0533925 | −0.0329446 |
| plain3.txt | 999 | 2 5 2 5 2 5 | 0.0422082 | 0.0390364 |
| plain4.txt | 3000 | 3 2 5 2 5 2 5 | −0.0049224 | −0.0160149 |
| plain5.txt | 5995 | 2 3 2 5 2 5 2 5 | 0.0123786 | 0.0105287 |
| plain6.txt | 9997 | 5 2 5 2 5 2 5 2 | −0.0216214 | −0.0220514 |
| plain7.txt | 12543 | 4 4 4 4 7 7 | 0.0011877 | 0.0013212 |
| plain8.txt | 16895 | 2 2 5 5 13 13 | −0.0009188 | −0.0001045 |
| plain9.txt | 20723 | 4 4 4 4 9 9 | 0.0070133 | 0.0092732 |

where $H_L(C) = |\{n|C[n] = L\}|$and $H_L(P) = |\{n|P[n] = L\}|$are the numbers of occurrences of the letter with ASCII code $L$in Ciphertext and Plaintext, respectively.

A cryptographic algorithm is better when the encryption quality is higher. The maximum encryption quality is obtained if all letters in the Plaintext are different with the letters in the Ciphertext. In this case, the maximum encryption quality is $2N'/95$, where $N'$ is length of the Ciphertext. The percentage of encryption quality is defined as the ratio between the encryption quality and the maximum encryption quality. In our case studies, as we can see in Table 2, the encryption quality increases when length of the Plaintext increases. The encryption quality is 0.5 when the length of Plaintext is 30, whereas the encryption quality is 325.5 when the length of Plaintext is 20723. The average of percentage of encryption quality among all Plaintext files is 76.71%. If we use the Type A MP-Wavelets, the average of percentage of encryption quality among all Plaintext files is 73.69%. Thus, the encryption quality of Type IVa MP-Wavelets is slightly better than the Type A MP-Wavelets.

### 4.3. The computational times

The cryptographic algorithm based on Type IVa MP-Wavelets has been implemented in Scilab 5.5.2. In order to test the scalability of the cryptographic algorithm, we determine the computational time of encryption and decryption processes.

The experiments have been run on an Intel® Core™ i7-7500U 2.90 GHz laptop with 12 GB of memory. From Table 3, the encryption time is 0.016 and 2.327 seconds when the length of Plaintext is 30 and 20723, respectively. The decryption time is 0.016 and 2.986 seconds when the length of Plaintext is 30 and 20723, respectively. The rate of computational time can be calculated by using the standard regression formula (Walpole, 1982). The encryption time increases by 0.108 seconds if the number of characters in the Plaintext increases by 1000. The decryption time increases by 0.146 seconds if the number of characters increases by 1000. For Type A MP-Wavelets, the encryption time increases by 0.105 seconds and the decryption time increases by 0.141 seconds if the number of characters increases by 1000. Thus, Type IVa MP-Wavelets is slightly slower than Type A MP-Wavelets.

**Table 2**
The encryption quality, maximum encryption quality and the percentage of encryption quality for Type IVa and Type A MP-Wavelets.

| Plaintext files | The encryption quality | | Maximum EQ | Percentage of EQ (%) | |
|---|---|---|---|---|---|
| | Type IVa | Type A | | Type IVa | Type A |
| plain1.txt | 0.5684211 | 0.5684211 | 0.631578947 | 90.0000075 | 90.0000075 |
| plain2.txt | 4.8631579 | 4.8210526 | 6.315789474 | 77.00000008 | 76.33333283 |
| plain3.txt | 16.926316 | 16.673684 | 21.05263158 | 80.400001 | 79.199999 |
| plain4.txt | 49.052632 | 46.694737 | 63.15789474 | 77.66666733 | 73.93333358 |
| plain5.txt | 97.031579 | 95.831579 | 126.3157895 | 76.81666671 | 75.86666671 |
| plain6.txt | 155.96842 | 143.14737 | 210.5263158 | 74.0849995 | 67.99500075 |
| plain7.txt | 196.71579 | 182.56842 | 264.0842105 | 74.48979612 | 69.13265266 |
| plain8.txt | 279.93684 | 277.25263 | 355.7894737 | 78.68047278 | 77.92603506 |
| plain9.txt | 325.51579 | 301.81053 | 436.5473684 | 74.56597234 | 69.13580331 |

**Table 3**
The encryption and decryption time for Type IVa and Type A MP-Wavelets.

| Plaintext files | Encryption time (seconds) | | Decryption time (seconds) | |
|---|---|---|---|---|
| | Type IVa | Type A | Type IVa | Type A |
| plain1.txt | 0.016 | 0.001 | 0.016 | 0.001 |
| plain2.txt | 0.015 | 0.00001 | 0.000 | 0.000 |
| plain3.txt | 0.015 | 0.015 | 0.015 | 0.015 |
| plain4.txt | 0.078 | 0.077 | 0.093 | 0.093 |
| plain5.txt | 0.265 | 0.249 | 0.359 | 0.359 |
| plain6.txt | 0.593 | 0.568 | 0.765 | 0.719 |
| plain7.txt | 0.875 | 0.859 | 1.14 | 1.109 |
| plain8.txt | 1.765 | 1.75 | 2.608 | 2.578 |
| plain9.txt | 2.327 | 2.234 | 2.986 | 2.844 |

### 4.4. Key space analysis

Key space analysis is used to determine the number of possibilities for decryption keys that might be used. In the cryptographic algorithm based on Type IVa MP-Wavelets, the decryption key consists of two parts. The first part contains the number of the channels in all levels. The second part contains a sequence of finitely many non-negative integers generated from the binary encoding of detail components. The key space can be calculated as follows:

- The first part of the decryption key is the number of the channels in all levels, i.e. a sequence of $m$ non-negative integers $p_1,\ldots,p_m$ such that $p_1 \times \ldots \times p_m = N'$, where $N'$ is the length of Ciphertext. We denote $P$ as the number of sequences that satisfy the previous condition.
- The second part of the decryption key is the binary encoding of detail components. The length of detail components is $N'-1$, where $N'$ is the length of Ciphertext. Since each character in the detail components is either 0 or 1, the number of binary numbers of length $N'-1$ is $2^{N'-1}$.

From the above calculation, we conclude that the key space of the cryptographic algorithm based on Type IVa MP-Wavelets is $P \times 2^{N'-1}$. Thus, the cryptographic algorithm has a large key space. So, it is difficult to attack this cryptographic algorithm by the brute-force method.

### 4.5. Cryptanalysis: ciphertext-only attack

In cryptanalysis, it is usually assumed that the attacker knows the cryptographic algoritm. Ciphertext-only attack or known ciphertext attack is an attack model where the attacker has access to a set of Ciphertexts. The attack is completely succesful if the corresponding Plaintexts can be deduced, or even better, the key.

In Type IVa MP-Wavelets, the attacker needs to determine the number of levels and the number of channels in each level. Notice that the multiplication of the number of channels equals the length of Ciphertext. When the attacker uses brute-force method, the attacker needs to try all sequences of non-negative integers such that the multiplication of terms equals the length of Ciphertext.

### 4.6. Entropy analysis

Information entropy is a basic criterion which is used to measure the randomness of data. The entropy $H$ of a message source $M$ can be computed as follows (Shannon, 1948):

$$H(M) = -\sum_{m\in M} p(m)\log_2 p(m), \tag{9}$$

where $p(m)$ represents the probability of symbol $m \in M$.

If the message source $M$ emits $2^n$ symbols with equal probability, then the entropy $H(M) = n$, which corresponds to a true-random source and represents the ideal value of entropy for message source $M$.

We have applied the entropy analysis to the proposed Type IVa MP-Wavelets. The results are shown in Table 4. The table shows that entropy of Ciphertext is higher than entropy of Plaintext. This means Ciphertext has a more uniform distribution than Plaintext.

**Table 4**
Entropy analysis for Type IVa MP-Wavelets.

| File | Entropy of Plaintext | Entropy of Ciphertext |
|---|---|---|
| plain1.txt | 4.2817277 | 4.2980685 |
| plain2.txt | 4.2770654 | 4.8694332 |
| plain3.txt | 4.3052071 | 4.9168251 |
| plain4.txt | 4.2624464 | 5.1034462 |
| plain5.txt | 4.4327231 | 5.0314334 |
| plain6.txt | 4.2658297 | 5.1429457 |
| plain7.txt | 4.2586416 | 5.2139441 |
| plain8.txt | 4.3244679 | 4.957424 |
| plain9.txt | 4.2884509 | 5.1547461 |

### 4.7. Key sensitivity

Key sensitivity is the percentage of change in the modified Ciphertext w.r.t. the original Ciphertext (Mishra and Mankar, 2012). The modified Ciphertext is obtained from the encryption of Plaintext by using the modified key. A cryptographic algorithm has a better key sensitivity when the percentage of change is higher.

In order to analyze the key sensitivity of the encryption process in Type IVa MP-Wavelets, the modified key is obtained by changing the first and last elements of the original key. The percentage of change in the encryption process is shown in Table 5. Observe that the percentage of change is higher when the first element of the key is modified.

Next, we analyze the key sensitivity in the decryption process. Similar with before, the modified key is obtained by changing the first and last element of the original key. The percentage of change in the decryption process is shown in Table 6. Observe that the percentage of change is high in both cases.

**Table 5**
Key sensitivity in the encryption process.

| File | Original Key | Flipped Key | Percentage of Change |
|---|---|---|---|
| plain1.txt | 2 3 5 | 2 5 3 | 36.666667% |
| | | 3 2 5 | 90% |
| plain2.txt | 3 2 5 2 5 | 3 2 5 5 2 | 2.6666667% |
| | | 2 3 5 2 5 | 90% |
| plain3.txt | 2 5 2 5 2 5 | 2 5 2 5 5 2 | 0.9% |
| | | 5 2 2 5 2 5 | 95.1% |
| plain4.txt | 3 2 5 2 5 2 5 | 3 2 5 2 5 5 2 | 0.2666667% |
| | | 2 3 5 2 5 2 5 | 89.3% |
| plain5.txt | 2 3 2 5 2 5 2 5 | 2 3 2 5 2 5 5 2 | 0.0833333% |
| | | 3 2 2 5 2 5 2 5 | 88.816667% |
| plain6.txt | 5 2 5 2 5 2 5 2 | 5 2 5 2 5 2 2 5 | 0.05% |
| | | 2 5 5 2 5 2 5 2 | 94.36% |
| plain7.txt | 4 4 4 4 7 7 | 4 4 4 7 7 4 | 1.3552296% |
| | | 7 4 4 4 4 7 | 95.998087% |
| plain8.txt | 2 2 5 5 13 13 | 2 2 5 13 13 5 | 4.3254438% |
| | | 5 2 2 5 13 13 | 94.544379% |
| plain9.txt | 4 4 4 4 9 9 | 4 4 4 9 9 4 | 1.326196% |
| | | 9 4 4 4 4 9 | 96.180556% |

**Table 6**
Key sensitivity in the decryption process.

| File | Original Key | Flipped Key | Percentage of Change |
|---|---|---|---|
| cipher1.txt | 2 3 5 | 2 5 3 | 80% |
| | | 3 2 5 | 100% |
| cipher2.txt | 3 2 5 2 5 | 3 2 5 5 2 | 70% |
| | | 2 3 5 2 5 | 94% |
| cipher3.txt | 2 5 2 5 2 5 | 2 5 2 5 5 2 | 90% |
| | | 5 2 2 5 2 5 | 96.3% |
| cipher4.txt | 3 2 5 2 5 2 5 | 3 2 5 2 5 5 2 | 70% |
| | | 2 3 5 2 5 2 5 | 94.066667% |
| cipher5.txt | 2 3 2 5 2 5 2 5 | 2 3 2 5 2 5 5 2 | 90% |
| | | 3 2 2 5 2 5 2 5 | 94.75% |
| cipher6.txt | 5 2 5 2 5 2 5 2 | 5 2 5 2 5 2 2 5 | 60.02% |
| | | 2 5 5 2 5 2 5 2 | 96.51% |
| cipher7.txt | 4 4 4 4 7 7 | 4 4 4 7 7 4 | 94.897959% |
| | | 7 4 4 4 4 7 | 98.294005% |
| cipher8.txt | 2 2 5 5 13 13 | 2 2 5 13 13 5 | 93.727811% |
| | | 5 2 2 5 13 13 | 97.544379% |
| cipher9.txt | 4 4 4 4 9 9 | 4 4 4 9 9 4 | 94.444444% |
| | | 9 4 4 4 4 9 | 98.423032% |

### 4.8. Plaintext sensitivity

Plaintext sensitivity is the percentage of change in the modified Ciphertext w.r.t. the original Ciphertext (Mishra and Mankar, 2012). The modified Ciphertext is obtained by changing one character in the original Plaintext. A cryptographic algorithm has a better Plaintext sensitivity when the percentage of change is higher.

In order to determine the Plaintext sensitivity of the proposed cryptographic algorithm, we use the following scenario. We measure the Plaintext sensitivity by changing the first character in the original Plaintext by character ~ (tilde). We choose ~ (tilde) because the character has the highest ASCII value so that the change will have a significant impact in the encryption process. The results are depicted in Table 7. From Table 7, we conclude that the Plaintext sensitivity of the proposed cryptographic algorithm is not very good because the percentage of change is less than 50% in all cases.

### 4.9. The complexity analysis

The complexity of the cryptographic algorithm based on Type IVa MP-Wavelets can be analyzed by counting the number of computations in the algorithm (Rossen, 2012). In analysis and synthesis process, we count the number of addition, subtraction and comparison processes. Symbol $S$ represents addition or subtraction process, whereas $C$ denotes the comparison process.

#### 4.9.1. The analysis process

From (2) and (3), we know that the analysis process consists of 1 comparison and $p_k - 1$ subtractions in level $k$. If there are $p_1, p_2, \ldots p_m$ channels in all levels, then the number of computations can be calculated by:

$$W_1 = \frac{N'}{p_1}(C + (p_1 - 1)S), \ldots, W_m = \frac{N'}{p_1 p_2 \ldots p_m}(C + (p_m - 1)S).$$

The total number of computations is equal to:

$$\begin{aligned} W &= W_1 + W_2 + \ldots + W_m \\ &= \tfrac{N'}{p_1}(C + (p_1 - 1)S) + \ldots + \tfrac{N'}{p_1 p_2 \ldots p_m}(C + (p_m - 1)S). \end{aligned}$$

Since $p_k - 1 < p_k$ and $p_k > 1$ for $k = 1, 2, \ldots m$, it follows that

$$\begin{aligned} W &< \tfrac{N'}{p_1}(p_1 C + p_1 S) + \tfrac{N'}{p_1 p_2}(p_2 C + p_2 S) + \ldots + \tfrac{N'}{p_1 p_2 \ldots p_m}(p_m C + p_m S) \\ &= \tfrac{N'}{p_1} p_1 (C + S) + \tfrac{N'}{p_1 p_2} p_2 (C + S) + \ldots + \tfrac{N'}{p_1 p_2 \ldots p_m} p_m (C + S) \\ &= N'(C + S) + \tfrac{N'}{p_1}(C + S) + \ldots + \tfrac{N'}{p_1 p_2 \ldots p_{k-1}}(C + S) \\ &< N'(C + S) + N'(C + S) + \ldots + N'(C + S) \\ &< mN'(C + S). \end{aligned}$$

The number of subtraction ($S$) and comparison ($C$) processes is $W < mN'$. Thus, we conclude that the complexity of analysis process is $W = O(N')$.

**Table 7**
Percentage of change in the Plaintext sensitivity experiments.

| Plaintext files | Number of characters | Percentage of change |
|---|---|---|
| plain1.txt | 30 | 13.333333% |
| plain2.txt | 300 | 2% |
| plain3.txt | 999 | 0.7% |
| plain4.txt | 3000 | 0.2666667% |
| plain5.txt | 5995 | 0.15% |
| plain6.txt | 9997 | 0.09% |
| plain7.txt | 12543 | 0.0558036% |
| plain8.txt | 16895 | 0.0414201% |
| plain9.txt | 20723 | 0.0337577% |

#### 4.9.2. The synthesis process

From (4) and (5), we know that the synthesis process consists of 1 comparison and $p_k$ addition or subtraction process in level $k$. If there are $p_1, p_2, \ldots, p_m$ channels in all levels, the number of computations can be calculated by:

$$W_1 = \frac{N'}{p_1}(C + p_1 S), \ldots, W_m = \frac{N'}{p_1 p_2 \ldots p_m}(C + p_m S).$$

The total number of computations is equal to:

$$\begin{aligned} W &= W_1 + W_2 + \ldots + W_m \\ &= \tfrac{N'}{p_1}(C + p_1 S) + \tfrac{N'}{p_1 p_2}(C + p_2 S) + \ldots + \tfrac{N'}{p_1 p_2 \ldots p_m}(C + p_m S). \end{aligned}$$

Since $p_k > 1$ for $k = 1, 2, \ldots, m$, it follows that

$$\begin{aligned} W &< \tfrac{N'}{p_1}(p_1 C + p_1 S) + \tfrac{N'}{p_1 p_2}(p_2 C + p_2 S) + \ldots + \tfrac{N'}{p_1 p_2 \ldots p_m}(p_m C + p_m S) \\ &= \tfrac{N'}{p_1} p_1 (C + S) + \tfrac{N'}{p_1 p_2} p_2 (C + S) + \ldots + \tfrac{N'}{p_1 p_2 \ldots p_m} p_m (C + S) \\ &= N'(C + S) + \tfrac{N'}{p_1}(C + S) + \ldots + \tfrac{N'}{p_1 p_2 \ldots p_{m-1}}(C + S) \\ &< N'(C + S) + N'(C + S) + \ldots + N'(C + S) \\ &< mN'(C + S). \end{aligned}$$

We find that number of addition or subtraction ($S$) and comparison ($C$) processes is $W < mN'$. Thus, the complexity of synthesis process is $W = O(N')$. From the above complexity analysis, we conclude that the complexity of this cryptographic algorithm is $W = O(N')$ or linear complexity (Rossen, 2012).

## 5. Conclusions

We have constructed a cryptographic algorithm based on Type IVa max-plus wavelet transforms (MP-Wavelets). Encryption and decryption algorithms are constructed based on the analysis and synthesis process of Type IVa MP-wavelets, respectively. The encryption key consists of the number of channels in all levels. The decryption key consists of two parts. The first part is the encryption key. The second part is a sequence of finitely many non-negative integers generated from the binary encoding of detail components. This ensures that the decryption key is very difficult to obtain by using the brute force method.

From the analytical and empirical study of the cryptographic algorithm based on Type IVa MP-Wavelets, the proposed cryptographic algorithm has a good performance and fast computation time.

## Declaration of Competing Interest

The authors declare that they have no known competing financial interests or personal relationships that could have appeared to influence the work reported in this paper.

## References

Arul, J., Venkatesulu, M., 2012. Encryption quality and performance analysis of gksbc algorithm. J. Inf. Eng. Appl. 2 (10), 26–34.
Baccelli, F., Cohen, G., Olsder, G., Quadrat, J.-P., 1992. Synchronization and Linearity: An Algebra for Discrete Event Systems. John Wiley & Sons Ltd.
Bede, B., Nobuhara, H., 2009. A novel max-plus algebra based wavelet transform and its applications in image processing. In: IEEE International Conference on Systems, Man and Cybernetics, pp. 2585–2588.
Boggess, A., Narcowich, F., 2015. A First Course in Wavelets with Fourier Analysis. John Wiley & Sons.
Cahyono, J., Subiono, 2016. A cryptographic algorithm based on max-plus-wavelet transforms. In: Proceeding of The 6th Annual Basic Science International Conference, pp. 304–308.
Durcheva, M., 2015. Some applications of idempotent semirings in public key cryptography. ACM Comm. Computer Algebra 49 (1), 19.
Fahim, K., Yunus, M., 2017. Max-plus algebra-based wavelet transforms and their FPGA implementation for image coding. Int. J. Tomography Simul. 30 (1), 118–126.

Fahim, K., Subiono, van der Woude, J., 2017. On a generalization of power algorithms over max-plus algebra. Discrete Event Dynamic Syst. 27 (1), 181–203.
Goldreich, O., 2001. Foundations of Cryptography Basic Tools. Cambridge University Press.
Goswami, D., Rahman, N., Biswas, J., Koul, A., Tamang, R., Bhattacharjee, D.A., 2011. A discrete wavelet transform based cryptographic algorithm. Int. J. Comput. Sci. Network Secur. 11 (4).
Grigoriev, D., Shpilrain, V., 2014. Tropical cryptography. Commun. Algebra 42 (6), 2624–2632.
Heidergott, B., Olsder, G., van der Woude, J., 2006. Max Plus at Work-Modeling and Analysis of Synchronized Systems: A Course on Max-Plus Algebra and Its Applications. Princeton University Press.
Kannoth, S., Kumar, H., 2018. Multi-image enhancement technique using max-plus algebra-based morphological wavelet transform. In: International Symposium on Signal Processing and Intelligent Recognition Systems, pp. 421–432.
Khan, M., Masood, F., Alghafis, A., 2019. Secure image encryption scheme based on fractals key with fibonacci series and discrete dynamical system. Neural Comput. Appl., 1–21
Kubo, S., Nishinari, K., 2018. Applications of max-plus algebra to flow shop scheduling problems. Discrete Appl. Math. 247, 278–293.
Lopes, G.A.D., Kersbergen, B., van den Boom, T.J.J., De Schutter, B., Babuka, R., 2014. Modeling and control of legged locomotion via switching max-plus models. IEEE Trans. Rob. 30 (3), 652–665.
Mishra, M., Mankar, V., 2012. Hybrid message-embedded cipher using logistic map. Int. J. Secur., Privacy Trust Manage. 1 (3–4), 81–91.
Nobuhara, H., Trieu, D., Maruyama, T., Bede, B., 2010. Max-plus algebra-based wavelet transforms and their fpga implementation for image coding. Inf. Sci. 180 (17), 3232–3247.
Rossen, K., 2012. Discrete Mathematics and Its Applications. The McGraw-Hill Companies, New York.
Shankar, K., Elhoseny, M., 2019. An optimal haar wavelet with light weight cryptography based secret data hiding on digital images in wireless sensor networks. In: Secure Image Transmission in Wireless Sensor Network (WSN) Applications. Springer, pp. 65–81.
Shannon, C., 1948. A mathematical theory of communication. Bell Syst. Technical J. 27 (3), 379–423.
Sivasankari, A., Krishnaveni, S., 2019. Optimal wavelet coefficients based steganography for image security with secret sharing cryptography model. In: Cybersecur. Secure Inf. Syst.. Springer, pp. 67–85.
Subiono, Fahim, K., Adzkiya, D., 2018. Generalized public transportation scheduling using max-plus algebra. Kybernetika 54 (2), 243–267.
Subiono, van der Woude, J., 2000. Power algorithms for (max,+)-and bipartite (min, max,+)-systems. Discrete Event Dyn. Syst. 10 (4), 369–389.
Walpole, R., 1982. Introduction to Statistic. Macmillan, New York.
Waqas, U., Khan, M., Batool, S., 2019. A new watermarking scheme based on daubechies wavelet and chaotic map for quick response code images. Multimedia Tools Appl., 1–24
Zhou, N., Hu, Y., Gong, L., Li, G., 2017. Quantum image encryption scheme with iterative generalized arnold transforms and quantum image cycle shift operations. Quantum Inf. Process. 16 (6), 164.
Zhou, N., Chen, W., Yan, X., Wang, Y., 2018. Bit-level quantum color image encryption scheme with quantum cross-exchange operation and hyper-chaotic system. Quantum Inf. Process. 17 (6), 137.
Zhou, N., Jiang, H., Gong, L., Xie, X., 2018. Double-image compression and encryption algorithm based on co-sparse representation and random pixel exchanging. Opt. Lasers Eng. 110, 72–79.
Zhou, N., Yan, X., Liang, H., Tao, X., Li, G., 2018. Multi-image encryption scheme based on quantum 3d arnold transform and scaled zhongtang chaotic system. Quantum Inf. Process. 17 (12), 338.