American J. of Engineering and Applied Sciences 4 (4): 461-469, 2011
ISSN 1941-7020

New Mersenne Number Transform Diffusion Power Analysis

Al-Gailani, M.F. and S. Boussakta
School of Electrical, Electronic and Computer Engineering,
Newcastle University, Newcastle Upon Tyne, NE1 7RU, England, UK

Abstract: Problem statement: Due to significant developments in the processing power and parallel processing technologies, the existing encryption algorithms are increasingly susceptible to attacks, such as side-channel attacks, for example. Designing new encryption algorithms that work efficiently on different platforms and security levels to protect the transmitted data from any possible attacks is one of the most important issues in today's information and network security. The aim is to find more secure, reliable and flexible systems that can run as a ratified standard, with reasonable computational complexity for a sufficient service time. To expand the longevity of the algorithm, it is important to be designed to work efficiently on a variety of block sizes and key lengths according to the security demand. A sensible solution is the suggested use of a parameter transform. **Approach:** The present study evaluates the appropriateness of the New Mersenne Number Transform for security applications by analyzing and estimating its avalanche and diffusion power. **Results:** The results confirm that the transform in general reflects good avalanche characteristics that are for most cases over 50% and can be up to 100%. The lower bound can be further improved by increasing the modulus and/or the transform length. **Conclusion:** This New Mersenne Number Transform is highly flexible and adaptable for this application. It can be involved in the design of a secure cryptosystem for the following reasons; changing a single input element makes drastic changes in the output elements and vice versa (sensitivity), provides variable block size and key length (parameterization). Has long transform length (power of two), is error free and its inverse is the same with a scale factor of (1/N) which simplifies implementation of both encryption and decryption. Finally, it is appropriate for real time implementations such as fast algorithms, which can be applied to it, to speed up processing.

Key words: Diffusion power, Number Theoretic Transform (NTT), fast algorithms, parameter transform, Maximum Distance Separable (MDS), Discrete Fourier Transform (DFT)

INTRODUCTION

Shannon (1949) introduced two main principles for designing secure cryptographic systems; confusion and diffusion. Substitution is one of the processes of confusion, in which the elements of the plaintext are mapped into other elements in order to complicate the relationship between the plaintext and the corresponding cipher text and its strength depends on the strength of the non-linear properties of the applied substitution box (S-box). Diffusion is the process that rearranges the plaintext into the cipher text. Accordingly, the measure of how influential the diffusion process is; can be measured by how the plaintext is redistributed across the cipher text. Additionally a small change in the influencer of the diffusion process (such as the key or the plaintext itself) should have a significant impact on the resulting cipher text. This effect is called the avalanche effect and a system is considered to have good avalanche characteristics if roughly half of the output data is affected for a single input change. Hence this impact is important to verify that the system is resilient to statistical attacks (Feistel, 1973; Heys and Tavares, 1995). However, after differential (Biham and Shamir, 1991) and linear (Matsui, 1994) cryptanalysis have been involved, designing the diffusion part of algorithms by relaying only on elements transposition or permutation has no longer become secure and algorithms become subject to attacks. Hence more sophisticated techniques have been involved to improve and strengthen the diffusion part, such as the use of transforms. For instance, in the Twofish algorithm (Schneier, 1999), a fixed transform, (4×4) Maximum Distance Separable (MDS) matrix over Galois field GF (2^8) is utilized. Where at each round an input vector of

Corresponding Author: Al-Gailani, M.F., School of Electrical, Electronic and Computer Engineering, Newcastle University, Newcastle upon Tyne, NE1 7RU, England, UK

four bytes in length is multiplied by the MDS over GF (2^8). A MDS matrix in hexadecimal form is given in (1) (Schneier, 1999) Eq. 1:

$$MDS = \begin{pmatrix} 01 & EF & 5B & 5B \\ 5B & EF & EF & 01 \\ EF & 5B & 01 & EF \\ EF & 01 & EF & 5B \end{pmatrix} \quad (1)$$

In the current state of the art, the Advanced Encryption Standard (AES) algorithm (Daemen and Rijmen, 2002), which was announced by National Institute of Standards and Technology (NIST) as U.S. Federal Information Processing Standards Publications 197 (FIPS PUB 197) on November 26, 2001 FIPS197 2001, a transform called mix columns is used for diffusion purposes, where the columns of the state are considered as a polynomial over GF(2^8) and a mix columns operation is undertaken by multiplying the columns modulo (x^4+1) with a fixed polynomial c(x). For inverse mix columns, the fixed polynomial d(x) is alternatively used. The c(x) and d(x) in hexadecimal values are given in (2) and (3) respectively (Daemen and Rijmen, 2002) Eq. 2 and 3:

$$c(x) = \text{‘03’ } x3 + \text{‘01’ } x2 + \text{‘01’ } x + \text{‘02’} \quad (2)$$

$$d(x) = \text{‘0B’ } x3 + \text{‘0D’ } x2 + \text{‘09’ } x + \text{‘0E’} \quad (3)$$

These transforms are powerful in diffusing data. However, their lengths are fixed for these dedicated algorithms. The disadvantage of this is that there is a need for an alternative algorithm, should the key length or block size becomes insufficient to suit the security requirements, due to future increases in processor power and parallel processing technologies, as was the case with the previous standard Data Encryption Standard (DES) algorithm FIPS_PUB_46-2 1977. Accordingly, a practical solution is the use of a parameter-based transform such that the key length and/or the block size can be changed by changing the transform size to adhere to the required level of security, i.e., a revision free algorithm, ensuring practical usage for the proposed lifespan.

In this study, a parameter-based New Mersenne Number Transform (NMNT) has been considered for security applications by evaluating its diffusion power and avalanche characteristics.

Consider that diffusion power of the algorithm in the design is very important, as the number of rounds for any iterated block cipher cryptosystem is inversely proportional to that value. Accordingly, building round functions with a higher diffusion rate will likely result in an efficient algorithm with a lesser number of rounds, which improves system performance, regarding speed and complexity.

MATERIALS AND METHODS

New Mersenne Number Transform (NMNT): NMNT is one of the Number Theoretic Transform (NTT) family. NTTs use modular arithmetic operations on a field or ring of integers, without the errors inherent to normal floating-point operations, such as those found in the Discrete Fourier Transform (DFT) for example. NTTs have wide applications in different areas including; digital signal processing (Agarwal, 1980), digital filtering (Agarwal and Burrus, 1974; Boussakta and Holt, 1994), image processing (Boussakta and Holt, 1999), decoding (Reed *et al*., 1978) and cryptography (Yang *et al*., 2010; Yang and Boussakta, 2008).

In the field of cryptography, NTTs are mainly used to improve the diffusion of the algorithm, in addition to other relevant applications in digital image information hiding (Yanqun and Qianping, 2009). The NMNT has been previously involved in the design of a cryptosystem by utilizing a cascade of such a transform with different transform lengths to ensure high diffusion rate throughout the processing (Yang *et al*., 2010). The NMNT is defined modulo of the Mersenne numbers (Mp). A detailed description of the transform can be found in Boussakta and Holt (1992; 1995). The transform can be used in one or multi dimensional (Boussakta and Holt, 1993; Boussakta *et al*., 2001). Both the forward and inverse transforms have a similar appearance, with a scale factor of (1/N) being the only difference. The forward 1-D NMNT X(k) of an integer sequence x(n) with transform length N and its inverse is defined as Eq. 4-12 fellow:

$$X(k) = \left\langle \sum_{n=0}^{N-1} x(n)\beta(nk) \right\rangle_{Mp} \quad k = 0,1,2,....N-1 \quad (4)$$

$$x(n) = \left\langle \frac{1}{N} \sum_{k=0}^{N-1} X(k)\beta(nk) \right\rangle_{Mp} \quad n = 0,1,2,....N-1 \quad (5)$$

Where:

$$\beta(nk) = \beta_1(nk) + \beta_2(nk) \quad (6)$$

$$\beta_1(nk) = \left\langle Re(a_1 + ja_2)_{nk} \right\rangle_{Mp} \quad (7)$$

$$\beta_2(nk) = \left\langle Im(a_1 + ja_2)_{nk} \right\rangle_{Mp} \quad (8)$$

$$Mp = 2^p - 1 \quad (9)$$

$$a_1 = \pm\left\langle 2^q \right\rangle_{Mp} \quad (10)$$

$$a_2 = \pm\left\langle -3^q \right\rangle_{Mp} \quad (11)$$

$$q = 2^{p-2} \quad (12)$$

The above kernels $\beta_1(nk)$ and $\beta_2(nk)$ are calculated for a maximum transform length 2^{P+1} . For transform lengths less than that, their values can be calculated using the following Eq. 13 and 14:

$$\beta_1(nk) = \left\langle \mathrm{Re}((a_1 + ja_2)^d)^{nk} \right\rangle_{Mp} \quad (13)$$

$$\beta_2(nk) = \left\langle \mathrm{Im}((a_1 + ja_2)^d)^{nk} \right\rangle_{Mp} \quad (14)$$

where, Re() and Im() stand for real and imaginary parts of the enclosed term respectively, denotes modulo Mp and d is an integer power of two.

Analysis: It is worth starting by giving a simple example that reflects the sensitivity of the transform for any change in the input or output elements. The example, which is adapted from (Al-Gailani and Boussakta, 2010), illustrates the effect of modifying a single output (transformed) element to the input elements. The text and ASCII representation for both the input elements to the transform and the corresponding output elements are illustrated in Fig. 1. The recovered plaintext results after modifying one of the transformed elements (shadowed) is shown being completely different, confirming the high sensitivity of the transform regarding any changes in the input or output elements. In other words, the transform possesses good avalanche characteristics.

The calculations are achieved by applying (4) and (5) respectively. Where N = 8, Mp = 127 (maximum input value is 121), $\alpha_1 = \alpha_2 = 119$ and $\beta(n) = 1\ 111\ 1\ 0\ 126\ 16\ 126\ 0$.

Plaintext:	A	n	a	l	y	s	i	s
P :	65	110	97	108	121	115	105	115
C :	74	16	113	64	67	110	109	94
Ciphertext:	J		q	@	C	n	m	^
C' :	74	16	113	64	63	110	109	94
P' :	1	47	33	45	57	52	41	52
Plaintext':		/	!	-	9	4	)	4

Fig. 1: 1D NMNT output modification (Al-Gailani and Boussakta, 2010)

Two different techniques are used to scrutinize and verify the diffusion power of the transform. The first technique involves the calculation of the branch number (Daemen and Rijmen, 2002) of the transform; a tool that is used to give an indication to the diffusion power of a linear transformation.

The Branch Number (BN) is calculated based on (15) Eq. 15:

$$Bn(F) = \min_{a\neq 0} \{W(a)+W(F(a))\} \quad (15)$$

where, W(a) is the bundle weight (number of non-zero elements, also called number of active elements) and F is the linear transformation.

The Bn of a transform is upper bounded by Eq. 16 (Daemen and Rijmen, 2002):

$$Bn(F) \leq N+1 \quad (16)$$

The second technique, which has been exploited previously on evaluating the diffusion power of the Fermat Number Transform (FNT) (Al-Gailani *et al*., 2011), is based on probabilities by calculating the diffusion power as a range of probabilities for different cases. These cases are determined according to the kernel matrix analysis listed in (Al-Gailani and Boussakta, 2010). The differences between these cases depend on the number of modified elements and their locations. The type of element modification depends on the modified values, these are; same value, different values with a total sum equal to the modulus for each modified pair/elements and different values with a total sum not equal to the modulus for each modified pair/elements. The range of probabilities for each case is calculated by counting the differences between the elements of the modified and unmodified versions, where diffusion power percentages represents the process results over-all by 100%.

The results are verified by recalculating the above cases by modifying the elements in three different tests. The first test is performed by transforming the input elements and producing the initially diffused elements. Next, the input is modified and transformed and the output is compared to the transformed output of the unmodified input. The second test is performed by modifying the transformed output elements and recalculating the input elements by applying the inverse transform and comparing the original input to the inversely transformed input. The final test involves modifying the mathematical equation according to the relevant cases (in the following explanation), examples are illustrated below; for modifying single element (17), single paired (18) and unpaired elements (19), all-odd

elements (20), all-even elements (21) and all elements respectively (22) Eq. 17-23:

$$(X)k = \left\langle \sum_{n=0}^{N-1} x(n)\beta(nk) + a\beta(ik) \right\rangle_{Mp} \quad k = 0,1,2,...N-1 \tag{17}$$

$$(X)k = \left\langle \sum_{n=0}^{N-1} x(n)\beta(nk) + a_1\beta(ik) + a_2\beta((i+N/2^x)k) \right\rangle_{Mp} \tag{18}$$

$$(X)k = \left\langle \sum_{n=0}^{N-1} x(n)\beta(nk) + a_1\beta(i_1k) + a_2\beta(1_2k) \right\rangle_{Mp} \tag{19}$$

$$(X)k = \left\langle \sum_{n=0}^{N-1} x(n)\beta(nk) + \left| \sum_{n=0}^{N/2^{-1}} a_{2n+1}\beta(2nk) \right\rangle_{Mp} \right. \tag{20}$$

$$X(k) = \left\langle \sum_{n=0}^{N-1} x(n)\beta(nk) + \sum_{n=0}^{N/2^{-1}} a_{2n+1}\beta((2n+1)k) \right\rangle_{Mp} \tag{21}$$

$$X(k) = \left\langle \sum_{n=0}^{N-1} x(n)\beta(nk) + \sum_{n=0}^{N-1} a_n\beta(nk) \right\rangle_{Mp} \tag{22}$$

where, i is the location of the modified element ($0 \le i \le N-1$) and a is the modification value that is added to the initial value.

Considering all these cases is very important so that apart from determining the diffusion power, the cases that provide maximum or minimum diffusion percentages can be exploited or avoided in the design.

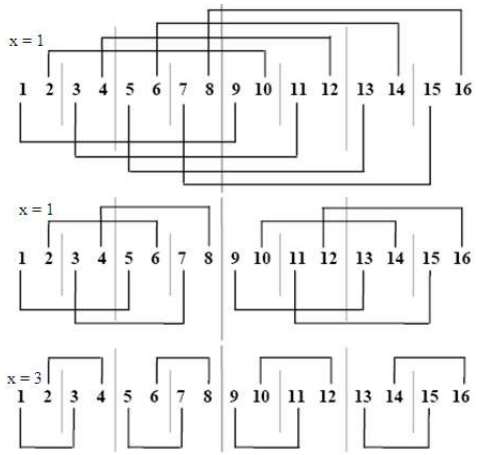

Fig. 2: Pairs distribution (Al-Gailani and Boussakta, 2010)

The elements are modified at the following locations:

- Initially, all of the single elements at even and odd locations are modified
- Next, all of the even/odd numbers of paired elements are modified at their corresponding even/odd/mix locations. This is shown in Fig. 2 using the formula (i, $i+N/2^x$) where ($1 \le x \le \log_2 N-1$)
- Following the modification of even/odd paired elements in the even/odd/mix locations, the remaining unpaired elements are modified which are situated in even/odd/mix locations
- A combination that requires the modification of both the paired elements is performed at even/odd/mix locations (i, $i+N/2^x$) using predetermined values and the remaining unpaired elements are replaced with random values
- The elements that reside in all-even positions, followed by the elements that reside in all-odd locations are modified
- Finally, all of the elements are modified for the last time, completing this particular process within the implementation

RESULTS

The calculations of the branch number for transform lengths (N): 4 and 8 are shown in Table 1. To illustrate that, consider the case for input bundle weight equal 1, under column N = 4, the output weight is 4, in total giving 5, which represents (N+1), signifying that the transform has maximum diffusion power. The same or larger output can be gain for input weights equal 3 or 4. However for input weight equal 2, the output weight is minimum 2, in total 4, this mean that for this case the transform has a lower value than (N+1), providing less diffusion than the maximum. This is especially the case for modifying an even number of active elements and up to (N/2) from the total elements. The details of all cases including those cases that provide low diffusion are explained in detail in the second method represented below.

Table 1: Minimum active bundles for transform lengths (N): 4 and 8

Bundle weight	NMNT (N = 4)	NMNT (N = 8)
1	5	7
2	4	4
3	6	7
4	5	6
5	-	8
6	-	7
7	-	9
8	-	9

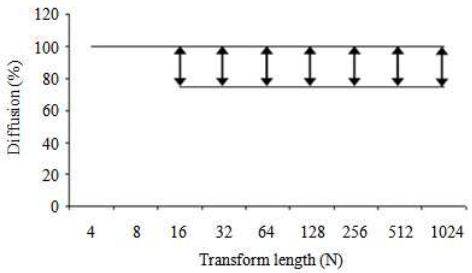

Fig. 3: Single element modifications at odd locations

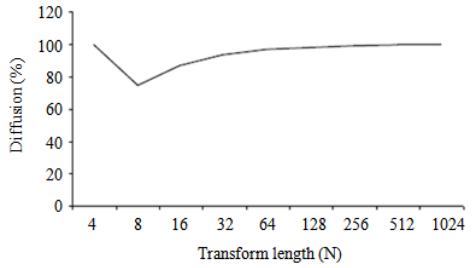

Fig. 4: Single element modifications at even locations

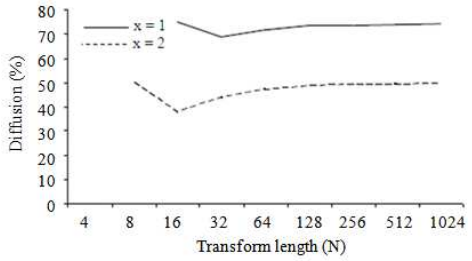

Fig. 5: Lower bounds for modifying odd number of paired elements at even locations with same value

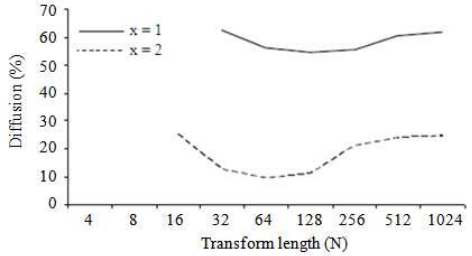

Fig. 6: Lower bounds for modifying even number of paired elements at even locations with same value

It has been shown from the results on (Al-Gailani and Boussakta, 2010) that there has been sufficient analysis performed using the NMNT with different moduli and transform lengths. The results can be classified into two groups. First group summarized the cases that provide good diffusion power that is at minimum 50% and the second group lists the cases that exhibit low diffusion power that is at maximum 50%. All calculations are based on element level i.e., (P-bits).

Cases that provide good diffusion power:

Modifying a single element: Modifying a single element at odd locations (Fig. 3), gives diffusion between 75-100%, depending on the number of elements with zero value at that row in the kernel matrix corresponding to the location of the modified element. Modifying a single element at even locations (Fig. 4), gives diffusion between 75-100%, increasing with larger transform length.

Modifying paired elements: Modifying any number of paired elements (i, $i+N/2^x$) at any location with any value and $\times>1$, gives minimum diffusion 50%. The lower bound improving with larger x and modulus and improving for larger transform length for elements modifying at even locations. Figure 5 explains the case for modifying an odd number of paired elements at any location with the same value, while Fig. 6 clarifies the case for modifying an even number of paired elements with the same value.

Modifying unpaired elements: Modifying even numbers of unpaired elements at any location with any value gives diffusion between 50-100%, increasing to 68-100% for modifying odd numbers of unpaired elements. In both instances, the lower bounds improved in most cases with larger modulus and/or transform length. Figure 7 explains the case for modifying an odd number of unpaired elements at even locations with the same value, while Fig. 8 shows the case for modifying an even number of unpaired elements at even locations with different values, with a total sum equal to Mp.

Modifying all elements: Modifying any number of paired elements with any value at any location and all other elements modified randomly (Fig. 9), gives in general diffusions over 75% increasing with larger moduli and transform lengths.

Cases that produce low diffusion power:

Modifying paired elements: Modifying a number of paired elements ($i,i+N/2^x$) up to N/2-1 pairs for x = 1 and leaving all other elements unchanged, at any location with same value for each pair (Fig. 5 and 6), or different values, with a total sum equal to Mp for each pair, gives in best cases a diffusion of 50%. The lower

bounds improve close to the upper bounds (50%) when modifying pairs at even locations with a larger modulus and/or transform length. The probability of this case arising is $N\times Mp^{-N}\times(Mp-1)$ for a single pair ($N\geq 8$) and become $N\times 2^{N/2-2}\times Mp^{-N}\times(Mp-1)^{N/2-1}$ for N/2-1 pairs. The diffusion power improves for $x > 1$.

Modifying all input elements: Modifying all input elements with the same value (Fig. 10), which is equivalent to adding a DC value, the diffusion percentage becomes $(N^{-1})\times 100\%$. The probability for this case taking place is $Mp^{-N}\times(Mp-1)$.

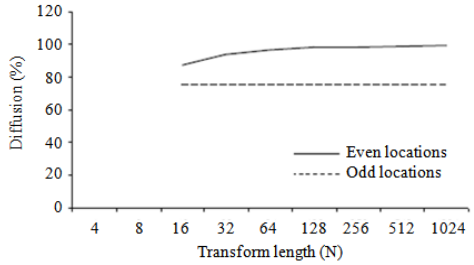

Fig. 7: Lower bounds for modifying odd number of unpaired elements with same value

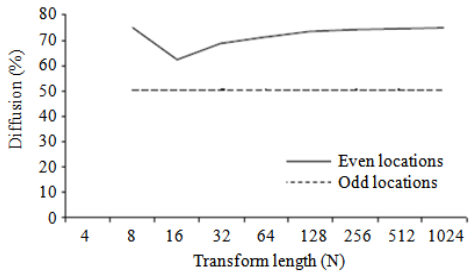

Fig. 8: Lower bounds for modifying even number of unpaired elements with different values their sum equal Mp

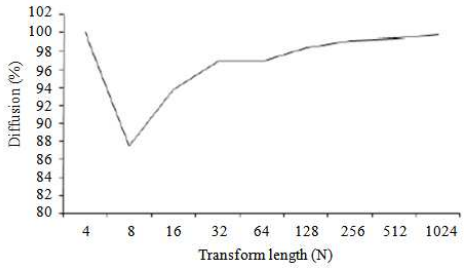

Fig. 9: Lower bounds for modifying any number of paired elements with any value and location and the remaining elements modified randomly

Modifying all even/odd input elements: Modifying all even input elements with the same value or all odd input elements with the same value (Fig. 10) or different values, with a total sum equal to modulus for each pair, same values for all modified pairs at odd locations, the diffusion percentage is $2\times N^{-1}\times 100\%$. The probability for this case raising is:

$$2\times Mp^{-N}\times(Mp-1)+Mp^{-N}\times(Mp-1)^{-N/2+2}\times(Mp-2)^{N/4}$$

The probabilities of the last two cases occurring can be reduced by increasing the modulus and/or the transform length.

Table 2 expands on (Al-Gailani and Boussakta, 2010) and explains some of these results in an example for P = 7, Mp = 127 and N = 16. In the beginning, initial data is required that represents the unmodified version and all comparison is done with it. This data is displayed in the first two rows. In the first two examples (rows 3-6), a single element is modified at odd positions (shadowed), where their diffusion percentage outputs are different. The first example gives 100% diffusion, providing that all of the output elements are completely different (shadowed), while the second example gives 75% diffusion. The reason behind this is related to the number of zero elements in that row within the kernel matrix corresponding to the position of the modified element. The next example (row 7-8), explains the case for modifying a single element at an even position. In every such case, all of the output elements are modified except two. This is because within the kernel matrix there are two zero elements in each even row. Examples on (rows 9-12), explain the case for modifying a single pair (i,i+N/2)with the same value. The results show that the diffusion vary between 37.5 and 50%. Finally, (rows 13-14) explains the case for modifying three elements with the same value, the output is completely different, giving 100% diffusion.

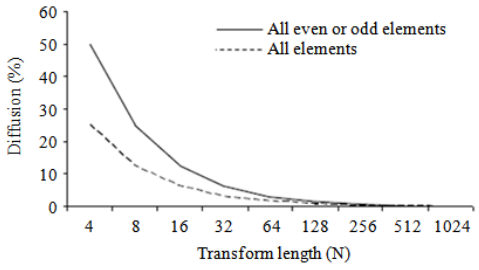

Fig. 10: All elements and all even/odd elements modification with same value

Table 2: 1D NMNT modification comparisons for P = 7, Mp = 127, N = 6

Initial values	I/P	81	43	121	17	44	119	111	69	75	3	26	38	51	29	107	33
	O/P	78	72	98	122	50	69	122	66	11	61	9	30	103	48	15	88
Modified input	I/P	81	43	121	17	64	119	111	69	75	3	26	38	51	29	107	33
	O/P	98	92	78	102	70	89	102	46	31	81	116	10	123	68	122	68
	I/P	81	43	129	17	44	119	111	69	75	3	26	38	51	29	107	33
	O/P	86	71	106	122	42	70	114	66	19	60	17	30	95	49	7	88
	I/P	81	53	121	17	44	119	111	69	75	3	26	38	51	29	107	33
	O/P	88	3	65	53	60	99	122	36	1	3	42	99	93	18	15	118
	I/P	81	43	126	17	44	119	111	69	75	3	31	38	51	29	107	33
	O/P	88	72	108	122	40	69	112	66	21	61	19	30	93	48	5	88
	I/P	81	43	121	19	44	119	111	69	75	3	26	40	51	29	107	33
	O/P	82	72	98	122	46	69	58	66	7	61	9	30	107	48	79	88
	I/P	81	48	126	17	49	119	111	69	75	3	26	38	51	29	107	33
	O/P	93	26	18	19	55	42	112	46	16	84	89	123	98	118	5	98

DISCUSSION

The diffusion power of the NMNT has been considered in this study using two different techniques in order to evaluate the appropriateness of the transform for security applications.

The branch number of the transform which is discussed on the first technique indicates that the transform can provide maximum diffusion power for most cases, exception mostly for even input weight and up to the transform length (N)/2. However, the analysis from the second technique explains deeply this case which obviously arises with very low probability when modifying only pairs of elements (just for x = 1) with the same value or different values with their sum equal to the modulus.

The results of the second technique are classified into two groups; the diffusion power for the first group which represents the cases that provide good diffusions, in general over 50% and the percentages of the lower bounds are further improved with higher modulus and/or transform length. One of the factors that improve the diffusion percentages with higher transform lengths is when the percentages relating to the number of zero elements, explained in (23) is inversely proportional to the transform length as illustrated in Fig. 11:

$$Zp = \left(\frac{\log_2 N - 2}{N} \right) \times 100\% \quad (23)$$

The second group represents the cases that provide low diffusion power, less than 50%. These cases can be avoided by ensuring that the number of modified elements is odd, or alternatively the probability for those cases arising can be reduced by increasing the modulus and/or the transform length. In general, increasing the modulus and/or transform length is beneficial as it either improves the diffusion power, or reduces the probability for those cases arising in the second group.

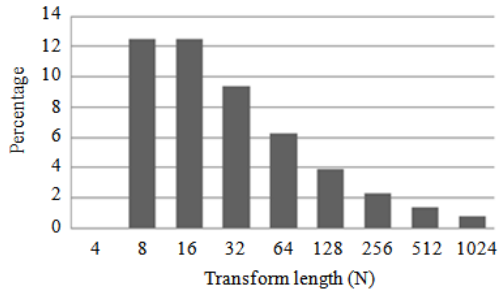

Fig. 11: Percentages of the number of zero elements relative to the total (Al-Gailani and Boussakta, 2010)

Of relevant importance is ensuring that the diffusion power improves with bigger block sizes or key lengths, which may be achieved by increasing the modulus and/or the transform length. This will facilitate the design by providing the possibility of changing the block size or key length to the required level of security without the need to alter the algorithm and at the same time fix the number of rounds for different sizes, which supports the compatibility of the algorithm on different platforms.

CONCLUSION

In conclusion, although the results demonstrate that the transform in certain cases provides lower diffusion than the maximum due to the matrix symmetry that can be avoided in the design, it can be concluded that the transform has many features qualifying it to be used in the design of a secure cryptosystem. Advantages include parameterization; providing flexibility to change the key length and/or block size to meet the required level of security and sensitivity; the diffusion power has been proven that in general it is good. Having a long transform length, these operations are performed without the errors that normally arise through using floating-point operations. Finally, fast

algorithms such as radix-2 (Nibouche *et al*., 2009), radix-4 (Boussakta *et al*., 2003) and split radix (Alshibami *et al*., 2000) can be adapted to it, to speed up processing.

According to the above, the transform is recommended to be employed in the design of a secure cryptosystem as a main diffusion layer for both the traditional cryptosystem like the AES or for applications such as audio or image encryption that require special treatments due to their size. Such applications are usually based on the chaos function, for instance the one found in (Ling *et al*., 2007), which proposes a practical and flexible cryptosystem that can be easily adapted to the international multimedia standards, such as JPEG 2000 and MPEG4.

ACKNOWLEDGMENT

The researchers would like to acknowledge the financial support of the EPSRC under Grant number GR/598160/02.

REFERENCES

Agarwal, R. and C. Burrus, 1974. Fast Convolution using fermat number transforms with applications to digital filtering. IEEE Trans. Acoustics, Speech Signal Proc., 22: 87-97. DOI: 10.1109/TASSP.1974.1162555

Agarwal, R., 1980. Number theory in digital signal processing. IEEE Trans. Acoustics, Speech Signal Proc., 28: 265-266. DOI: 10.1109/TASSP.1980.1163371

Al-Gailani, M.F. and S. Boussakta, 2010. Evaluation of one-dimensional NMNT for security applications. Proceedings of the 7th International Symposium on Communication Systems, Networks and Digital Signal Processing, Jul. 21-23, IEEE Xplore Press, Newcastle upon Tyne, pp: 715-720.

Al-Gailani, M.F., S. Boussakta and J. Neasham, 2011. Fermat number transform diffusion's analysis. Proceedings of the IEEE GCC Conference and Exhibition, Feb. 19-22, IEEE Xplore Press, Dubai, pp: 237-240. DOI: 10.1109/IEEEGCC.2011.5752501

Alshibami, O., S. Boussakta, M. Aziz and D. Xu, 2000. Split-radix algorithm for the new Mersenne number transform. Proceedings of the 7th IEEE International Conference on Electronics, Circuits and Systems, Dec. 17-20, IEEE Xplore Press, Jounieh, Lebanon, pp: 583-586. DOI: 10.1109/ICECS.2000.911607

Biham, E. and A. Shamir, 1991. Differential cryptanalysis of DES-like cryptosystems. J. Cryptol., 4: 3-72. DOI: 10.1007/BF00630563

Boussakta, S. and A.G.J. Holt, 1992. New number theoretic transform. Elect. Lett., 28: 1683-1684. DOI: 10.1049/el:19921070

Boussakta, S. and A.G.J. Holt, 1993. New two dimensional transform. Elect. Lett., 29: 949-950. DOI: 10.1049/el:19930632

Boussakta, S. and A.G.J. Holt, 1994. Filtering employing a new transform. Proceedings of the Oceans Eng. Today's Technol. Tomorrow's Preservation, Sep. 13-16, IEEE Xplore Press, Brest, France, pp: I/547-I/553. DOI: 10.1109/OCEANS.1994.363875

Boussakta, S. and A.G.J. Holt, 1995. New transform using the Mersenne numbers. IEE Proc. Vis. Image Signal Proc., 142: 381-388. DOI: 10.1049/ip-vis:19952323

Boussakta, S. and A.G.J. Holt, 1999. Number theoretic transforms and their applications in image processing. Adv. Imag. Elect. Phys., 111: 1-90. DOI: 10.1016/S1076-5670(08)70216-7

Boussakta, S., O. Alshibami, M. Aziz and A.G.J. Holt, 2001. 3-D Vector radix algorithm for the 3-D new Mersenne number transform. IEE Proc. Vis. Image Signal Process., 148: 115-125. DOI: 10.1049/ip-vis:20010312

Boussakta, S., O. Alshibami and A. Bouridane, 2003. Radix-4 decimation-in-frequency algorithm for the new Mersenne number transform. Proceedings of the 10th IEEE International Conference on Electronics, Circuits and Systems, Dec. 14-17, IEEE Xplore Press, UK., pp: 1133-1136. DOI: 10.1109/ICECS.2003.1301711

Daemen, J. and V. Rijmen, 2002. The Design of Rijndael: AES--the Advanced Encryption Standard. 1st Edn., Springer, Berlin, New York, ISBN: 3540425802, pp: 238.

Feistel, H., 1973. Cryptography and computer privacy. Sci. Am., 228: 15-23.

Heys, H.M. and S.E. Tavares, 1995. Avalanche characteristics of substitution-permutation encryption networks. IEEE Trans. Comput., 44: 1131-1139. DOI: 10.1109/12.464391

Ling, B.W.K., C.Y.F. Ho and P.K.S. Tam, 2007. Chaotic filter bank for computer cryptography. Chaos Solitons Fractals, 34: 817-824. DOI: 10.1016/j.chaos.2006.03.105

Matsui, M., 1994. Linear cryptanalysis method for DES cipher. Lecture Notes Comput. Sci., 765: 386-397. DOI: 10.1007/3-540-48285-7_33

Nibouche, O., S. Boussakta and M. Darnell, 2009. Pipeline architectures for radix-2 new mersenne number transform. IEEE Trans. Circ. Syst., 56: 1668-1680. DOI: 10.1109/TCSI.2008.2008266

Reed, I., T.K. Truong and L. Welch, 1978. The fast decoding of Reed-Solomon codes using Fermat transforms (Corresp.). IEEE Trans. Inform. Theory, 24: 497-499. DOI: 10.1109/TIT.1978.1055902

Schneier, B., 1999. The Twofish Encryption Algorithm: A 128-Bit Block Cipher. 1st Edn., John Wiley, New York, ISBN: 0471353817, pp: 186.

Shannon, C.E., 1949. Communication theory of secrecy systems. Bell. Syst. Tech. J.

Yang, X.B. and S. Boussakta, 2008. A new development of symmetric key cryptosystem. Proceedings of the IEEE International Conference on Communications, May 19-23, IEEE Xplore Press, Beijing, pp: 1546-1550. DOI: 10.1109/ICC.2008.299

Yang, X.B., Boussakta, S. Al-Gailani and M.N. Ruzelita, 2010. A new development of cryptosystem using new mersenne number transform. Proceedings of the 7th International Symposium on Communication Systems, Networks and Digital Signal Processing, Jul. 21-23, IEEE Xplore Press, United Kingdom, pp: 701-705.

Yanqun, Z. and W. Qianping, 2009. A new scrambling method based on arnold and fermat number transformation. Proceedings of the International Conference on Environmental Science and Information Application Technology, Jul. 4-5, IEEE Xplore Press, Wuhan, pp: 624-628. DOI: 10.1109/ESIAT.2009.252