Statistical Testing for Disk Encryption Modes of Operations

Mohamed Abo El-Fotouh and Klaus Diepold

Institute for Data Processing (LDV)
Technische Universität München (TUM)
Munich, Germany
Emails:[mohamed, kldi]@tum.de

Abstract. In this paper we present a group of statistical tests that explore the random behavior of encryption modes of operations, when used in disk encryption applications. The results of these tests help us to better understand how these modes work. We tested ten modes of operations with the presented statistical tests, five of the narrow-block type and the other five of the wide-block type. Our analysis shows some weakness in some of these modes.

Keywords: Disk encryption, modes of operations, randomness, AES.

1 Introduction

Data security on lost or stolen PC devices is a growing concern among security experts and corporate executives. The data stored on the PC asset is often significantly more valuable to a corporation than the asset itself, and the loss, theft or unwanted disclosure of that data can be very damaging [1]. Thus, this data should be encrypted to minimize the loss. Disk encryption is usually used to encrypt all the data on the hard disk, where all the hard disk is encrypted with a single/multiple key(s) and encryption/decryption are done on the fly, without user interference.

Disk encryption usually encrypts/decrypts a whole sector at a time. There exist dedicated block ciphers to encrypt the whole sector at a time like Bear, Lion, Beast and Mercy [2, 2–4]. Bear, Lion and Beast are considered to be slow, as they process the data through multiple passes. And Mercy was broken in [5]. The other method is to let a block cipher like the AES [6] (with 16 bytes as a block size) to process the data within a mode of operation. These modes of operation can be divided to two main classes the narrow-block and wide-block modes. The narrow-block modes operate on relatively small portions of data (typically 16 bytes when AES is used), while the wide-block modes encrypt or decrypt a whole sector (typically 512 bytes) at a time [7].

One of the criteria used to evaluate block ciphers is their demonstrated suitability as random number generators. That is, the evaluation of their outputs utilizing statistical tests should not provide any means by which to computationally distinguish them from truly random sources [8]. And that is the case when the modes of operation are used to encrypt data. A study was done in [9] for testing the randomness of the final five candidates of the AES algorithms. This study was done on the block ciphers themselves, the data sets described in this paper were inspired from their work. In this paper, we designed 21 different data sets, that can help us analysis the randomness of the disk encryption modes of operations. We used the NIST statistical tool [10] to analysis these data sets, using the current default parameters.

We are going to study ten modes of operations, five narrow-block modes (CFB, CBC, CTR, LRW and XTS)[11, 11–14] and five wide-block modes (EME, EME*, XCB, ABL4 and AES-CBC + Elephant diffuser "Windows Vistas disk encryption algorithm - we will use only the term ELF in the rest of the paper" [15–18, 1]). For all the mentioned modes, we are going to use the AES as the working block cipher.

We studied 21 different data sets for each mode (if applicable), to help us evaluating the random behavior of each mode of operation. These tests explore the random behavior of the mode of operation, when used to encrypt sectors.

In section 2 we will present out test methodology to test the randomness behavior of the modes of operation dealing with different patterns of plaintext and tweaks. In section 3 we describe the used data sets. In section 4, we will test the narrow-block modes of operations and comment on the results. In section 5 we will test the wide-block modes of operations and comment on the results. In section 6 we perform further analysis on some modes of operations (where some modifications were done to these modes, to examine there behavior when some of their internal functions generate unexpected outputs "low/high density sequences"), summary of our analysis and some recommendations and a comparison among the modes of operations. In section 7 we will present our conclusions.

2 Testing methodology

During our analysis of the randomness of the studied modes of operations, fifteen different statistical tests have been applied to each data set. Some tests have been applied several times with different parameters. Each sequence in each data set is subject to 188 different statistical tests [9] as shown in table 1.

2.1 The Statistical Tests

The used statistical tests are:

Frequency Test [19]: The purpose of this test is to determine whether the number of ones and zeros in a sequence is approximately the same as it would be expected for a truly random sequence.

Table 1. Breakdown of the 188 statistical tests applied during experimentation.

Statistical Test	No. of P-values	Test ID
Frequency	1	1
Block Frequency	1	2
Cusum	2	3-4
Runs	1	5
Long Runs of Ones	1	6
Rank	1	7
Spectral DFT	1	8
A periodic Templates	144	9-156
Periodic Template	1	157
Universal Statistical	1	158
Approximate Entropy	1	159
Random Excursions	8	160-167
Random Excursions Variant	18	168-185
Serial	2	186-187
Linear Complexity	1	188

Block Frequency Test [19]: The purpose of this test is to determine whether the frequency of m-bit blocks in a sequence appears as often as it would be expected for a truly random sequence.

Cumulative Sums Forward (Reverse) Test [20]: The purpose of this test is to determine whether the maximum of the cumulative sums in a sequence is too large or too small; indicative of too many ones or zeroes in the early (late) stages.

Runs Test [21]: The purpose of this test is to determine, whether the number of runs of ones and zeros of various lengths is as expected for a random sequence. In particular, this test determines whether the oscillation between such substrings is too fast or too slow.

Long Runs of Ones Test [21]: The purpose of this test is to determine whether the distribution of long runs of ones agree with the theoretical probabilities.

Rank Test [22]: The purpose of this test is to determine whether the distribution of the rank of 32x32 bit matrices agree with the theoretical probabilities.

Spectral (Discrete Fourier Transform) Test [23]: The purpose of this test is to determine whether the spectral frequency of the binary sequence agree with what would be expected for a truly random sequence.

Non-periodic Templates Test [24]: The purpose of this test is to determine whether the number of occurrences for a specified non-periodic template agree with the number expected for a truly random sequence.

Overlapping Template Test [25]: The purpose of this test is to determine, whether the number of occurrences for a template of all ones agrees with what is expected for a truly random sequence.

Universal Statistical Test [25]: The purpose of this test is to determine whether a binary sequence does not compress beyond what is expected of a truly random sequence.

Approximate Entropy Test [26]: The purpose of this test is to compare the frequency of overlapping blocks of two consecutive/adjacent lengths (m and m+1) against the expected result for a normally distributed sequence. In short, it determines whether a sequence appears more regular than it is expected from a truly random sequence.

Random Excursion Test [27]: The purpose of this test is to examine the number of cycles within a sequence and determine whether the number of visits to a given state, [-4, -1] and [1, 4], exceeds the expected for a truly random sequence.

Random Excursion Variant Test [27]: The purpose of this test is to determine if the total number of visits to states between [-9, -1] and [1, 9] exceeds the expected for a truly random sequence.

Linear Complexity Test [28]: The purpose of this test is to determine whether or not the sequence is complex enough to be considered truly random.

Serial [29]: The purpose of this test is to determine whether the number of occurrences of the 2m m-bit overlapping patterns is approximately the same as would be expected for a random sequence.

For more details on these tests refer to [8] which contains much theoretical explanation. Table 2 shows The NIST Test Suite parameters that are used in our analysis.

Table 2. The NIST Test Suite parameters.

Parameter	**Value**
Block Frequency block length	128
Long Runs substring length	10,000
Aperiodic Templates template length	9
Periodic Template template length	9
Universal Statistical number of blocks	7
Universal Statistical initialization block length	1280
Approximate Entropy block length	10
Serial block length	16
Linear Complexity block length	500
Number of bit sequences m for the datasets	128
Bit sequence lengths in bits for data sets 9, 10 and 11	2,147,483,648/T (where T is the length of the tweak)
Bit sequence lengths in bits for all other data sets	16,809,984
Significance level α (*)	0.01

2.2 Randomness Test Strategy

Randomness testing was performed using the following strategy:

1. Input parameters such as the sequence length, sample size and significance level (0.01) were fixed for each sample. For each binary sequence and each statistical test, a P-value (probability that the test succeeded) was reported.
2. For each P-value, a success/failure assessment was made based on whether or not it exceeded or fell below the pre-selected significance level.
3. For each statistical test and each sample, two evaluations were made. First, the proportion of binary sequences in a sample that passed the statistical test was calculated. The P-value for this proportion is equal to the probability of observing a value equal to or greater than the calculated proportion. Second, an additional P-value was calculated, based on a χ^2 (chi-square) test (with nine degrees of freedom) applied to the P-values in the entire sample to ensure uniformity.
4. For both measures described in step (3), an assessment was made. A sample was considered to have passed a statistical test if it satisfied both the proportion and uniformity assessments. If either of the two P-values for a test in step (3) fell below 0.0001, this test is considered to have failed the randomness testing.
5. For each data set a "Total" is calculated, which is the number of succeeded tests.

3 Data sets

We designed 21 different random data sets that we believe could help us better understand the random behavior of the tweakable block ciphers.

3.1 General notes.

1. The term low density tweak (used in this paper) refers to a tweak with at most two ones.
2. The term high density tweak (used in this paper) refers to a tweak with at most two zeros.
3. The low density tweaks generated in the data sets described below follow the following order. The first tweak consists of all zeros. Then T tweaks with a single one and the other bits of the tweak are zeros (the one appears in each of the possible T positions), and the rest of the tweaks are of two ones appearing in different random positions.
4. The high density tweaks generated in the data sets described below follow the following order. The first tweak consists of all ones. Then T tweaks with a single zero and the other bits of the tweak are ones (the zero appears in each of the possible T positions), and the rest of the tweaks are of two zeros appearing in different random positions.
5. The low density plaintext generated in the data sets described below follow the following order. The first plaintext consists of all zeros. Then 512 plaintext with a single one and the other bits of the plaintext are zeros.

6. The high density plaintext generated in the data sets described below follow the following order. The first plaintext consists of all ones. Then 512 plaintext with a single zero and the other bits of the plaintext are ones.
7. As AES with different key sizes is shown to be random (even with low and high density keys) [9], we used simply random keys.
8. We used the AES with 256-bit in our analysis.

3.2 The examined data sets:

1-Random plaintext / random tweak: In order to examine the randomness of the ciphertext (based on the tested mode of operation), 128 sequences were constructed. Each sequence was a result of the concatenation of 4104 ciphertext blocks, resulting of encrypting of 4104 random plaintexts and a random tweak, encrypted with the examined mode of operation in ECB manner.

2-Low density plaintext: In order to examine the sensitivity of the examined mode of operation to low deinsty plaintext, 128 sequences were constructed. Each sequence was a result of the concatenation of 4104 ciphertext blocks, resulting of encrypting of 4104 low deinsty plaintexts and a random tweak, encrypted with the examined mode of operation in ECB manner.

3-High density plaintext: In order to examine the sensitivity of the examined mode of operation to low deinsty plaintext, 128 sequences were constructed. Each sequence was a result of the concatenation of 4104 ciphertext blocks, resulting of encrypting of 4104 high deinsty plaintexts and a random tweak, encrypted with the examined mode of operation in ECB manner.

4-Plaintext avalanche with random tweak: In order to examine the sensitivity of the examined mode of operation to the change in the plaintext, when a random tweak is used, 128 sequences were constructed. Each sequence was a result of the concatenation of 4104 derived blocks. Each derived block is based on the XOR of the "ciphertext formed using a random plaintext and a fixed random tweak", and the "ciphertext formed using the perturbed random 4096-bit plaintext with the ith bit changed, for $0 \leq i \leq 4095$ and the fixed random tweak".

5-Plaintext avalanche with low density tweak: In order to examine the sensitivity of the examined mode of operation to the change in the plaintext, when a low density tweak is used, 128 sequences were constructed. Each sequence was a result of the concatenation of 4104 derived blocks. Each derived block is based on the XOR of the "ciphertext formed using a random plaintext and a low density tweak", and the "ciphertext formed using the perturbed random 4096-bit plaintext with the ith bit changed, for $0 \leq i \leq 4095$ and the fixed low density tweak".

6-Plaintext avalanche with high density tweak: In order to examine the sensitivity of the examined mode of operation to the change in the plaintext, when a high density tweak is used, 128 sequences were constructed. Each sequence was a result of the concatenation of 4104 derived blocks. Each derived block is based on the XOR of the "ciphertext formed using a random plaintext and a high density tweak", and the "ciphertext formed using the perturbed random 4096-bit plaintext with the ith bit changed, for $0 \leq i \leq$ 4095 and the fixed high density tweak".

7-Low density tweak: In order to examine the sensitivity of the examined mode of operation to low deinsty tweak, 128 sequences were constructed. Each sequence was a result of the concatenation of 4104 ciphertext blocks, resulting of encrypting of a random plaintext with 4104 low deinsty tweaks, encrypted with the examined mode of operation in ECB manner.

8-High density tweak: In order to examine the sensitivity of the examined mode of operation to high deinsty tweak, 128 sequences were constructed. Each sequence was a result of the concatenation of 4104 ciphertext blocks, resulting of encrypting of a random plaintext with 4104 high deinsty tweaks, encrypted with the examined mode of operation in ECB manner.

9-Tweak avalanche with random plaintext: In order to examine the sensitivity of the examined mode of operation to the change in the tweak, when a random plaintext is used, T sequences were constructed. Each sequence was a result of the concatenation of (524,288/T) derived blocks. Each derived block is based on the XOR of the "ciphertext formed using a random plaintext and a fixed random tweak", and the "ciphertext formed using the perturbed random T-bit tweak with the ith bit changed, for $0 \leq i \leq T$ and the fixed random plaintext".

10-Tweak avalanche with low density plaintext: In order to examine the sensitivity of the examined mode of operation to the change in the tweak, when a low density plaintext is used, T sequences were constructed. Each sequence was a result of the concatenation of (524,288/T) derived blocks. Each derived block is based on the XOR of the "ciphertext formed using a low density plaintext and a random tweak", and the "ciphertext formed using the perturbed random T-bit tweak with the ith bit changed, for $0 \leq i \leq T$ and the low density plaintext".

11-Tweak avalanche with high density plaintext: In order to examine the sensitivity of the examined mode of operation to the change in the tweak, when a high density plaintext is used, T sequences were constructed. Each sequence was a result of the concatenation of (524,288/T) derived blocks. Each derived block is based on the XOR of the "ciphertext formed using a high density plaintext and a random tweak", and the "ciphertext formed using the perturbed random T-bit tweak with the ith bit changed, for $0 \leq i \leq T$ and the high density plaintext".

12-Random plaintext correlation: In order to examine the correlation between of plaintext/ciphertext pairs (based on the tested mode of operation), 128 sequences were constructed. Each sequence was a result of the concatenation of 4104 derived blocks (where a derived block is the result of applying the XOR operator on the plaintext block and its corresponding ciphertext block computed in the ECB mode using random plaintexts and a random tweak, encrypted with the examined mode of operation).

13-Low density plaintext correlation: In order to examine the correlation between of plaintext/ciphertext pairs (in case of low density plaintext), 128 sequences were constructed. Each sequence was a result of the concatenation of 4104 derived blocks (where a derived block is the result of applying the XOR operator on the plaintext block and its corresponding ciphertext block computed in the ECB mode using low deinsty plaintexts and a random tweak, encrypted with the examined mode of operation).

14-High density plaintext correlation: In order to examine the correlation between of plaintext/ciphertext pairs (in case of high density plaintext), 128 sequences were constructed. Each sequence was a result of the concatenation of 4104 derived blocks (where a derived block is the result of applying the XOR operator on the plaintext block and its corresponding ciphertext block computed in the ECB mode using high deinsty plaintexts and a random tweak, encrypted with the examined mode of operation).

15-Low density tweak correlation: In order to examine the correlation between of plaintext/ciphertext pairs (in case of low density tweak), 128 sequences were constructed. Each sequence was a result of the concatenation of 4104 derived blocks (where a derived block is the result of applying the XOR operator on the plaintext block and its corresponding ciphertext block computed in the ECB mode using low deinsty tweaks and a random plaintext, encrypted with the examined mode of operation).

16-High density tweak correlation: In order to examine the correlation between of plaintext/ciphertext pairs (in case of high density tweak), 128 sequences were constructed. Each sequence was a result of the concatenation of 4104 derived blocks (where a derived block is the result of applying the XOR operator on the plaintext block and its corresponding ciphertext block computed in the ECB mode using high deinsty tweaks and a random plaintext, encrypted with the examined mode of operation).

17-Random plaintext correlation': In order to examine the correlation between of plaintext/ciphertext pairs (based on the tested mode of operation), 128 sequences were constructed. Each sequence was a result of the concatenation of 4104 derived blocks (where a derived block is the result of applying the XOR operator on the ciphertext using the parent mode of operation of the examined mode of operation (table 3 list the parent of each mode of operation "if applicable") and that using the examined mode of operation computed in the ECB mode using random plaintexts and a random tweak).

18-Low density plaintext correlation': In order to examine the correlation between of plaintext/ciphertext pairs (in case of low density plaintext), 128 sequences were constructed. Each sequence was a result of the concatenation of 4104 derived blocks (where a derived block is the result of applying the XOR operator on the ciphertext using the parent mode of operation of the examined mode of operation and that using the examined mode of operation computed in the ECB mode using low deinsty plaintexts and a random tweak).

19-High density plaintext correlation': In order to examine the correlation between of plaintext/ciphertext pairs (in case of high density plaintext), 128 sequences were constructed. Each sequence was a result of the concatenation of 4104 derived blocks (where a derived block is the result of applying the

XOR operator on the ciphertext using the parent mode of operation of the examined mode of operation and that using the examined mode of operation computed in the ECB mode using high deinsty plaintexts and a random tweak).

20-Low density tweak correlation': In order to examine the correlation between of plaintext/ciphertext pairs (in case of low density tweak), 128 sequences were constructed. Each sequence was a result of the concatenation of 4104 derived blocks (where a derived block is the result of applying the XOR operator on the ciphertext using the parent mode of operation of the examined mode of operation and that using the examined mode of operation computed in the ECB mode using low deinsty tweaks and a random plaintext).

21-High density tweak correlation': In order to examine the correlation between of plaintext/ciphertext pairs (in case of high density tweak), 128 sequences were constructed. Each sequence was a result of the concatenation of 4104 derived blocks (where a derived block is the result of applying the XOR operator on the ciphertext using the parent mode of operation of the examined mode of operation and that using the examined mode of operation computed in the ECB mode using high deinsty tweaks and a random plaintext).

Table 3. Parents modes.

Mode	Parent
CBC	None
CFB	None
CTR	None
LRW	ECB
XTS	ECB
EME	ECB
EME*	ECB
ELF	CBC
XCB	CTR
ABL4	CTR

3.3 Notes

1. In table 3, the primitive modes CBC, CFB and CTR have no parents, while all the other modes are considered derivative modes and have parents.
2. Not all the data sets can be applied to each mode, table 4 presents the applicable data set for each mode, where $\surd$ means that this data set is applicable and X means that it is not applicable.
3. All the data sets are applicable to all the wide modes of operations.

4. The narrow-block modes of operations are sensitive to data sets that are highly correlated, when encrypted with the same tweak, that is why all the narrow modes failed to pass data sets number 2, 3, 4, 5, 6, 13, 14, 18 and 19. And we did not apply these data sets to the narrow-block modes, as a small change in the plaintext will lead to a small change in the ciphertext.

Table 4. Applicable data sets for each mode.

	1	2	3	4	5	6	7	8	9	10	11	12	13	14	15	16	17	18	19	20	21
CBC	√	X	X	X	X	X	√	√	√	√	√	√	X	X	√	√	X	X	X	X	X
CFB	√	X	X	X	X	X	√	√	√	√	√	√	X	X	√	√	X	X	X	X	X
CTR	√	X	X	X	X	X	√	√	√	√	√	√	X	X	√	√	X	X	X	X	X
LRW	√	X	X	X	X	X	√	√	√	√	√	√	X	X	√	√	√	X	X	√	√
XTS	√	X	X	X	X	X	√	√	√	√	√	√	X	X	√	√	√	X	X	√	√
EME	√	√	√	√	√	√	√	√	√	√	√	√	√	√	√	√	√	√	√	√	√
EME*	√	√	√	√	√	√	√	√	√	√	√	√	√	√	√	√	√	√	√	√	√
ELF	√	√	√	√	√	√	√	√	√	√	√	√	√	√	√	√	√	√	√	√	√
XCB	√	√	√	√	√	√	√	√	√	√	√	√	√	√	√	√	√	√	√	√	√
ABL4	√	√	√	√	√	√	√	√	√	√	√	√	√	√	√	√	√	√	√	√	√

4 Narrow-block modes analysis

The results of applying the NIST statistical tool on the data sets of narrow-block modes of operations are summarized in table 5. Table 6 shows the tweak length used by each mode of operation. In the following sub-sections, we are going to interpret the results in table 5. Notes:

Table 5. Narrow-block test results.

	1	7	8	9	10	11	12	15	16	17	20	21
CBC	187	185	185	187	188	187	187	185	185	X	X	X
CFB	187	184	186	188	185	188	38	186	185	X	X	X
CTR	188	185	185	188	188	188	0	133	140	X	X	X
LRW	188	12	25	0	0	0	187	18	21	187	20	11
XTS	187	187	188	188	185	188	187	188	188	187	187	186

- The data sets were generated three times (using three different random number generators) and only the best results are noted here.
- In table 5, the numbers presented range between 0 (did not pass a single test) to 188 (did pass all the tests).

Table 6. Narrow-block tweak length.

Mode	Tweak length in bits
CBC	128
CFB	128
CTR	128
LRW	192
XTS	192

4.1 CBC mode

The tweak in the CBC mode is used as the initial vector (IV). This mode has a good random profile for all the applicable data sets.

4.2 CFB mode

The tweak in the CFB mode is used as the initial vector (IV). This mode has a good random profile for all the applicable data sets, except the data set number 12 (Random plaintext correlation), where the output of this data set does not seem to be random.

4.3 CTR mode

The tweak in the CTR mode is used as the initial counter. This mode has a good random profile for all the applicable data sets, except the data set number 12, 15 and 16, where the output of these data sets do not seem to be random.

4.4 LRW mode

The tweak in the LRW mode is divided as following: the first 128-bits are assigned to F, and the last 64-but are assigned to sector number (for more details refer to [13]). This mode has a good random profile for only the data sets number 1, 12 and 17, otherwise the data sets do not seem to be random.

4.5 XTS mode

The tweak in the XTS mode is divided as following: the first 128-bits are assigned to Key2 and the last 64-bits are assigned to the Data Unit Sequence Number (for more details refer to [14]). This mode has a good random profile for all the applicable data sets.

5 Wide-block modes analysis

The results of applying the NIST statistical tool on the data sets of wide-block modes of operations are summarized in table 7.

Table 8 shows the tweak length used by each mode of operation. In the following sub-sections, we are going to interpret the results in table 7.

Table 7. Wide-block test results.

	1	2	3	4	5	6	7	8	9	10	11	12	13	14	15	16	17	18	19	20	21
EME*	186	188	187	177	179	179	188	187	188	187	186	188	188	188	186	187	184	188	186	186	187
XCB	187	188	186	177	177	179	185	185	187	187	188	187	188	186	185	186	187	188	187	186	185
EME	186	186	188	179	178	179	183	185	188	187	188	187	187	188	186	186	187	186	188	186	186
ABL4	188	188	188	179	178	176	186	183	188	188	188	188	188	188	185	185	188	186	187	184	186
ELF1	187	188	188	179	176	178	184	186	187	187	188	186	188	188	185	186	186	188	188	184	184
ELF2	186	188	187	179	179	179	185	185	188	188	188	187	188	188	185	183	186	187	188	185	186
ELF3	188	187	187	178	178	178	185	184	188	185	187	184	188	188	185	186	187	188	188	186	185

Table 8. Wide-block tweak length.

Mode	Tweak length in bits
EME*	256
EME	128
XCB	128
ABL4	128
ELF	128

5.1 EME mode

The tweak in the EME mode is used as T defined in [15]. This mode has a good random profile for all the data sets.

5.2 EME* mode

The tweak in the EME* mode is divided into two parts, the first part is L (128-bit) and the second part is R(128-bit) and T is left empty, for more information refer to [16]. This mode has a good random profile for all the data sets.

5.3 XCB mode

The tweak in the XCB mode is used as Z defined in [17]. This mode has a good random profile for all the data sets.

5.4 ELF mode

The tweak in the ELF mode consists of three 128-bits. In this paper, we studied each one of them separately (results appear in table 7 under ELF1, ELF2 and ELF3), due to the relationship among these tweaks (Tweak1 and tweak2 are the result of the encryption of the same text with different keys. Tweak3 is the result of encrypting the text used to produce tweak2 after modifying one byte using the same key): Tweak1 and tweak 2 are xored with the plaintext and tweak3 is used as the initial vector (IV) for the AES-CBC layer, for more derails refer to [1]. This mode has a good random profile for all the data sets.

5.5 ABL4 mode

The tweak in the ABL4 mode is used as Z defined in [18]. This mode has a good random profile for all the data sets.

6 Further analysis and recommendations

6.1 Further analysis

The XTS and EME* modes modify the tweak before using it by either encryption or hashing. This method can turn a non-random tweaks (with low or high density) to random tweaks. But what if the input tweak after encryption or hashing turned out to be a weak tweak (with low or high density), this is possible as the output of encryption or hashing can be low or high density text. In this sub-section we study the behavior of these modes of operations if the tweak after modification is high or low density text. In order to achieve that, we had to modify each mode a little bit. The results are summarized in table 9. The performed modifications are:

- For XTS: Our modification was not to encrypt the tweak using Key2 refer to [14]. This modification hurts XTS random profile and the outputs of all the data sets (except for data sets 1, 12 and 17) appear not to be random (the same case as LRW).
- For EME*: Our modification was not to hash the R tweak refer to [16]. This modification to EME* does not hurt its random profile.

Table 9. Modified modes test results.

	1	2	3	4	5	6	7	8	9	10	11	12	13	14	15	16	17	18	19	20	21
XTS	188	X	X	X	X	X	22	26	0	0	26	187	X	X	21	11	188	X	X	16	11
EME*	188	187	187	178	179	176	187	187	187	187	-812	188	187	187	188	187	188	186	188	187	188

6.2 Summary and recommendations

The studied disk encryption modes of operations take three parameters :

1. Encryption key: the used cipher AES is considered to have a good random profile [9], even if the input key is low/high density.
2. Plaintext: we do not have any control over the plaintext; we should be able to encrypt anything (low/high density plaintexts appear in practice).
3. Tweak:
   - The tweak is usually a result of an encryption or a hash operation and getting a low or high density tweaks can not be avoided (or a collision can occur ”the reuse of a tweak”).

- LRW has a problem, when the tweak is not random (high density or low density).
- LRW has a problem with the tweak avalanche.
- If the tweak used in XTS after encryption is a high density or low density text, then XTS will face the same problems as LRW.

For the narrow-block modes CBC, CFB and CTR are exposed to bit-flipping attack [30], so we do not recommend using them. LRW and XTS possess a low random profile (when the used "tweak/the used tweak after encryption" is a high density or low density text), we do not recommend using them or any other DESX-Like [31] structures.

For the wide-block modes, all the tested modes possess a high random profile and all of them are superior to those of narrow-block modes.

7 Conclusions

We studied the random behavior of ten disk encryption modes of operation, to explore their weakness. Our study was based on the random behavior of those modes. We perform statistical analysis for 21 data sets for each mode (if applicable). Our study shows that LRW mode possesses a poor random profile when it use high/low density tweaks (the same weakness applies to XTS also), and as the other narrow-block modes (CBC, CTR and CFB) are already exposed to some modification attacks we do not recommend using them either. Our recommendation is to use one of the studied wide-block modes over any of the studied narrow-block modes.

References

1. N. Ferguson. AES-CBC + Elephant diffuser : A Disk Encryption Algorithm for Windows Vista. http://download.microsoft.com/download/0/2/3/0238acaf-d3bf-4a6d-b3d6-0a0be4bbb36e/BitLockerCipher200608.pdf, 2006.
2. R. Anderson and E. Biham. Two practical and provable secure block ciphers: Bear and lion. In *Dieter Gollmann, editor, Fast Software Encryption: Third International Workshop (FSE'96)*, 1996.
3. S. Lucks. Beast: A fast block cipher for arbitrary blocksizes. In *Patrick Horster, editor, Communications and Multimedia Security II, Proceedings of the IFIP TC6/TC11 International Conference on Communications and Multimedia Security*, 1996.
4. P. Crowley. Mercy: a fast large block cipher for disk sector encryption. In *Bruce Schneier, editor, Fast Software Encryption: 7th International Workshop, FSE 2000*, 2001.
5. S. Fluhrer. Cryptanalysis of the mercy block cipher. In *Mitsuru Matsui, editor, Fast Software Encryption, 8th International Workshop, FSE 2001*, 2002.
6. J. Nechvatal, E. Barker, L. Bassham, W. Burr, M. Dworkin, J. Foti, and E. Roback. Report on the Development of the Advanced Encryption Standard (AES). Technical report, 2000.

7. IEEE P1619 homepage on Wikipedia. $http : //en.wikipedia.org/wiki/IEEE_P1619$.
8. J. Soto. Randomness testing of the advanced encryption standard candidate algorithms, 1999.
9. J. Soto and L. Bassham. Randomness testing of the advanced encryption standard finalist candidates, 2000.
10. NIST statistical Suite. available at http://csrc.nist.gov/rng/rng2.html.
11. A. Menezes, P. Van Oorschot., and S. Vanstone. *Handbook of Applied Cryptography.* CRC Press, 1996.
12. D. McGrew. Counter mode security: Analysis and recommendations.
13. M. Liskov, R. Rivest, and D. Wagner. Tweakable block ciphers. In *CRYPTO '02 (LNCS, volume 2442)*, 2002.
14. P. Rogaway. Efficient instantiations of tweakable blockciphers and refinements to modes OCB and PMAC.
15. S. Halevi and P. Rogaway. A parallelizable enciphering mode. http://eprint.iacr.org/2003/147.
16. S. Halevi. EME*: extending EME to handle arbitrary-length messages with associated data, 2004.
17. D. McGrew and S. Fluhrer. The extended codebook (xcb) mode of operation. Cryptology ePrint Archive, Report 2004/278, 2004. http://eprint.iacr.org/.
18. D. McGrew and J. Viega. Arbitrary block length mode, 2004.
19. N. MacLaren. Cryptographic pseudorandom numbers in simulation. In *Fast Software Encryption, Cambridge Security Workshop*, pages 185–190, London, UK, 1994. Springer-Verlag.
20. P. Revesz. Random walk in random and non-random environments. World Scientific, 1990.
21. A. Godbole and S. Papastavridis. Runs and patterns in probability: Selected papers. Dordrecht: Kluwer Academic, 1994.
22. G. Marsaglia. and H. Tsay. Matrices and the structure of random number sequences. Linear Algebra and its Applications, 1985.
23. R. Bracewell. The fourier transform and its applications. New York:McGraw-Hill, 1986.
24. A.Barbour, L. Holst., and S. Janson. Poisson approximation. Oxford: Clarendon Press, 1992.
25. N. Johnson, S. Kotz, and A. Kemp. Discrete distributions. John Wiley,2nd ed. New York, 1996.
26. A. Rukhin. Approximate entropy for testing randomness. Journal of Applied Probability, Vol. 37, 2000.
27. M. Baron and A. Rukhin. Distribution of the number of visits for a random walk. Communications in Statistics: Stochastic Models. Vol. 15, 1999.
28. H. Gustafson, E. Dawson, L. Nielsen, and W. Caelli. A computer package for measuring the strength of encryption algorithms. Computers and Security, 1994.
29. A. Rukhin. A statistical test suite for the validation of cryptographic random number generators. NIST Computer Security Division/Statistical Engineering Division Internal Document, 1999.
30. C. Fruhwirth. New methods in hard disk encryption. http://clemens.endorphin.org/nmihde/nmihde-A4-ds.pdf, 2005.
31. P. Rogaway. The security of desx, 1996.