Vol.2013-DPS-154 No.13
Vol.2013-CSEC-60 No.13
2013/3/14

ファイルのエントロピー測定による類似度評価の新手法に関する提案

高田慎也† 元田敏浩† 中原慎一†

†NTT セキュアプラットフォーム研究所
180-8585 東京都武蔵野市緑町 3-9-11 NTT 武蔵野研究開発センタ
{takada.shinya, motoda.toshihiro ,nakahara.shinichi }@lab.ntt.co.jp

あらまし 類似するファイルをPCやサーバの中から高速かつ高精度に見つけ出すことに対するニーズは高く、ファジーハッシュ測定法、エントロピー測定法といった様々な研究が行われている。本稿では特に、エントロピーを使った類似度測定法に着目する。従来のエントロピー測定法では、ファイル全体のエントロピー値を比較するため、無関係のファイル対について偶然類似度を高く評価してしまう事例が散見される。これを解決するためにファイルを区分に分け、個々の区分でのエントロピー値を求め、そのスペクトルを比較することで類似度評価を行う新手法を提案し、その有効性を評価実験により示す。

キーワード エントロピー、類似度、デジタルフォレンジック

A new method for similarity evaluation by measuring partitioned entropy

Shinya Takada† Toshihiro Motoda† Shinichi Nakahara†

†NTT Secure Platform Laboratories
3-9-11, Midori-cho, Musashino-shi, Tokyo 180-8585 JAPAN
{takada.shinya, motoda.toshihiro, nakahara.shinichi }@lab.ntt.co.jp

Abstract The needs to detect similar files from PCs and servers are high, and to achieve this many methods are reported like fuzzy hashing, entropy evaluation. In this paper we focus on entropy evaluation. In conventional method entropy of entire file is measured and for this in some case is failed to detect unrelated files similar. To improve this we suggest new method separating files and measuring partitioned entropy and comparing spectrum of partitioned entropy. We also report the experimental results of new method.

Keyword entropy, similarity, digital forensic

1 はじめに

類似するファイルを高速かつ高精度に見つけ出すことに対するニーズは高く、情報セキュリティ分野においては、例えば、多変形性のマルウェアに関する類似コードを発見し、駆逐することに多くの努力が払われている。こうした分野で使用されるファイル類似度の評価方法としては、例えば、ファイルのエントロピー値を比較することで類似度を測定する方法の研究が盛んに行われている[1][2][3][4]。Mccreight らは、測定法をさらに発展させ、ファイルサイズで重みを付けた Weighted Entropy を使って、類似度を評価することを提案している[1]。しかしファイル全体のエントロピーを使ったファイルの類似度評価は、無関係の 2 つのファイルが偶然大きな類似度をとる事象が多々発生しうるという問題があった[1][3]。そこで今回エントロピー値をファイルの

Vol.2013-DPS-154 No.13
Vol.2013-CSEC-60 No.13
2013/3/14

区分ごとに計算し、得られるファイル区分エントロピーを比較することで、より詳細な類似度を判定する方式を新たに考案したので報告する。

本稿の構成は以下の通りである。最初に第 2 章で従来の類似度評価手法の変遷を詳述し、第 3 章では、中でも既存のエントロピーを用いた類似度評価の問題点を実測値により明らかにする。次に第 4 章でエントロピーを用いた類似度が偶然一致した場合でも、ファイルの相違をさらに分別するファイル区分エントロピー比較法を提案する。第 5 章で提案方式の有効性評価のために行った実験の結果を報告する。最後に第 6 章でまとめと今後の課題を述べる。

2 類似度評価手法の変遷

2.1 Diff（txt ファイルの差分を見つける手段）

類似ファイルの発見に関する初期の試みでは、類似するコードの目視による発見や"Diff"による解析が行われていた。Diff は 2 つのファイルの相違点を出力するファイル比較ユーティリティである。Diff はテキストファイルの行ベースで行われた改変を表示するものである。目視による比較という点でこの手法による類似性の評価は高いものであるが、機械的な処理ではなく、労働集約的な取り組みとなり、企業レベルの情報処理にはスケールしないものである。

2.2 ハッシュ（完全一致ファイルを特定する手段）

コンピュータフォレンジック分野でのハッシュの応用は良く知られている。ハッシュアルゴリズムは無制限のサイズのファイルを入力とし、ハッシュ値と呼ばれる固定長の値を出力する。ハッシュ値はそれ自体意味、文脈を持たず、ハッシュ値からファイルの特徴を類推することはできない。ハッシュ値群は、原本ファイルについて完全一致を前提として、手早くファイルを特定するのに用いられている。[2]

2.3 ファジーハッシュ

時にコンテキスト駆動型区分ハッシュと表記されるように、ファジーハッシュは類似度評価に関する既存のハッシュ処理の応用であることを意味する。ファジーハッシュは ssdeep[5]や基になった spamsum[6]に代表されるように、その処理過程において、ファイルを小さな区分に分けて、個々の区分でのハッシュの一致を測定することで類似度をパーセントで評価する。ハッシュ値の集合は 64 文字以下の特徴量として保持される。ハッシュとは言えセキュリティ分野で用いられる数百ビット程度のハッシュ値を出力するハッシュ関数ではなく、計算量の少ない FNV ハッシュを 6 ビットの出力で用いている。そのため各区分について 1/64 程度の確率でハッシュ衝突が発生するため、それに基づく類似度も一定の誤差を含むことを考慮する必要がある。ファジーハッシュでは、コンテキストを意識して各区分の分割点が一致しなければ類似度の比較ができない。これには入力データの N バイト分によってのみ値が定まるローリングハッシュという特殊なハッシュ関数を用いて、その出力値の整数 M による余剰が M-1 に一致した点を分割点にする手法を用いる。それでも M の値が一致しなければ比較できないため、1 つの特徴量の中に M の値、M による分割ハッシュ 2M による分割ハッシュの両方を保持する工夫をおこなっている。

芹田らは M として 2 のべき乗 2^t を選択し、入力データが最初に分割される t の値 t_max から 1 ずつ係数を減らして入れ子で分割数を増やした複数階層のハッシュ値を保持することで編集によって長さが大きく変化した部分データの比較を可能とした方式を提案している[7][8]。

ファジーハッシュは上述した分割点の最適化のために計算量がやや多い傾向がある。

2.4 エントロピー

エントロピーは閉域系における順序性の程度の指標値である。情報理論としてのエントロピーは、電子データを２５６通りで表現されるバイトの集合とみなす。そして、そのバイト集合に偏りがある場合は、電子データが規則性のある状態（エントロピー値＝0）、反対に偏りが存在しな

Vol.2013-DPS-154 No.13
Vol.2013-CSEC-60 No.13
2013/3/14

い場合はランダムな状態（エントロピー値＝8）と見なす。そして、計算されたデータの"ランダムさ"は、「エントロピー値」という絶対値として表現される。エントロピー値、及び順序性考慮型エントロピー値は以下の式により表わされる。

$$E = -\sum_{i=0}^{255} P_i \log_2(P_i) \qquad [式\ 1]$$

$$M1E = -\sum_{i=0}^{255} P_i \sum_{j=0}^{255} P_i(j) \log_2\left(P_i(j)\right) \qquad [式\ 2]$$

データの状態をエントロピー値で表現することで、ストレージなどに保存された大量の情報を分類したり、特定ファイル間での近似を検証したりするのに役立つ。また、オリジナルに対し、意図的に変更を加えて検知を逃れるタイプのマルウェアが存在するため、オリジナルと複製の近似を客観かつ定量的に表現することは、フォレンジック調査にとって非常に大きな意味がある。対象データの一部を変更した場合、オリジナルデータからの変化の量に応じてエントロピーの値は連続的に変化する。これに対して、フォレンジックでデータの識別に用いられるハッシュでは、データとハッシュ値の変化量に相関関係は存在しないため、ファイルの近似の検証という点においては、エントロピーが有利である。また、ハッシュと比べ、計算コストが低くすむため、大量のデータをすばやく検証するのにもエントロピーは向いている[3]。

エントロピーは N-gram の特殊形とも言える。N-gramとは評価対象を文字単位で分解し、後続の N-1 文字を含めた状態で出現頻度を求める方法である。エントロピーは中でも Uni-gram の出現頻度の偏りを1つの数値として特徴量化したものとみなせる。出現頻度の計数はカウンタのみで済むためハッシュに比べて計算量は小さい。

情報量や重み付けエントロピー（Weighted Entropy）はエントロピーにファイル長やファイル長の対数値を掛け合わせた数値で、特に後者は数値の桁数増大が抑えられる事からデジタルフォレンジックの分野で用いられている。Weighted Entropy は以下の式で表わされる[1]。

$$WEntropy = Entropy \times \ln(Filesize) \qquad [式\ 3]$$

また、前後のデータの差分のエントロピーや一次マルコフ過程のエントロピーを用いる Weighted Entropy も提案されており、計算量や必要とする記憶容量は増加するが一般的なエントロピーに比べて識別精度が若干高まる[4]。

3 既存のエントロピー類似度評価方式の問題点

Weighted Entropy を用いたファイルの類似度評価式は、Mccreight らの特許[1]には参考として

$$類似度\ 1 = \log(E_1 - E_2)\log(S_1 - S_2) \qquad [式\ 4]$$

で与えられている。しかしながら、この式は一例であって有意な値をとらない。例えば、E_1-E_2 が負の値を取る場合、対数計算が行えない点や、E_1-E_2 の値が 1 以下の場合、類似度が負の値となってしまう点で実用には向かない。このため、類似度評価式として、例えば、

$$類似度\ 2 = \left(1 - \frac{|E_1S_1 - E_2S_2|}{E_1S_1 + E_2S_2}\right) \times 100 \qquad [式\ 5]$$

を使用することとする[4]。ここで、E_1、E_2 はそれぞれ比較元のファイルのエントロピーと比較先のエントロピー、S_1、S_2はそれぞれ比較元のファイルサイズ、比較先のファイルサイズである。式はエントロピーとサイズの積が 2 つのファイルで一致するとき最大値 100 をとり、どちらかのエントロピーとサイズの積が0になる時、最小値 0 を取る。この評価式を、例えば Web 上から無作為抽出した 15 個のビットマップファイルについて適用した。個々のファイルの各種エントロピー値を表1に示す、また類似度の計算結果を表2に示す。表 1 からは、エントロピーの値を比較してみると、No.2とNo.15 のように非常に近い値を取るペアが存在することが分かる。また、表 2 を観察すると無作為に抽出した関係のないファ

表 2 BMP ファイルのエントロピー測定値

File Name	No.	Entropy	M1Entropy	WEntropy
ae.bmp	1	7.36303	4.758377	87.8226
af.bmp	2	**5.968411**	**3.792961**	**71.18828**
at.bmp	3	7.092342	4.848116	84.59397
ef.bmp	4	6.441338	4.875889	76.82912
imagesCA6MA9NM.bmp	5	7.376312	4.890895	87.98103
imagesCA83PY61.bmp	6	7.505963	5.646639	89.52744
imagesCACBRG4N.bmp	7	7.103144	4.52063	84.72282
imagesCAQI7U2A.bmp	8	7.23413	5.006533	86.28515
imagesCATI07Y0.bmp	9	3.828329	1.515625	45.66242
imagesCAUZ25S0.bmp	10	7.664049	6.288557	91.41301
imagesCAW6KN27.bmp	11	7.213798	5.803957	86.04263
re.bmp	12	5.456418	3.285703	65.08147
tr.bmp	13	2.791953	1.489784	33.30104
V4.bmp	14	5.509465	1.651926	65.7142
we.bmp	15	**5.963461**	**4.139408**	**71.12923**

イルであるにも関わらず、極めて近い類似度(99%)を取る事象が散見される(13 ペア)ことが見て取れる。このように既存のエントロピーを用いた類似度評価は、デジタルデータ全体として1つの数値を特徴量とすることから、識別分解能に限界があることが分かる。

4 提案方式

類似度評価値が一致した場合に、さらに類似性を細かく分析する手法として、ファイル区分エントロピー比較法を提案する。一番単純な例としては、

$$D = \frac{\sum_{i=1}^{n} |E1_i - E2_i|}{n} \qquad [式 6]$$

で表現される。この式では、比較対象の 2 つのファイルを始点から固定長でそれぞれ区分に分割し、各区分でのエントロピー値をそれぞれ

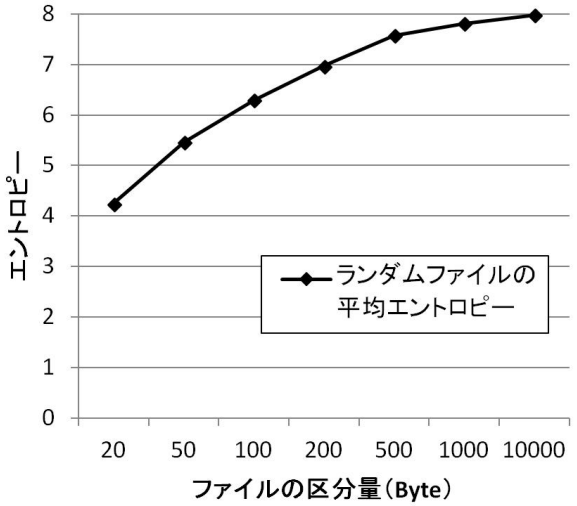

図 1 ランダムファイルの区分量に対するエントロピー値

($E1_i$, $E2_i$)求め、この例では値の差を取り、これをファイルの最後まで繰り返した後、差の平均を計算することで、さらに類似性(D)を評価する。換言すれば個々のファイルの区分エントロピースペクトルを測定し、差の平均を求める。差の平均が 0 の時 2 つのファイルは一致し、差の増大とともに 2 つのファイルの類似度は低くなり、最大値は 8 となる。

実装上、分割する区分の最大数は、特徴量、メタデータとして付与可能な最大長が、流用性、保存の観点から 256 バイト程度と思われることから、1 区分のエントロピーを 4 文字で格納することとすると、最大 64 区分程度が限界だと思われる。

また、分割する区分のサイズの大きさにも注意が必要である。あまりに小さい区分、例えば 20 バイトとした場合には、表現空間が限られて

表 1 無作為抽出した BMP ファイルの類似度(式 5 を使用)

File Name	No.	1	2	3	4	5	6	7	8	9	10	11	12	13	14	15
ae.bmp	1	100	89	98	93	**99**	**99**	98	**99**	68	97	98	85	54	85	89
af.bmp	2	89	100	91	96	89	88	91	90	78	87	90	95	63	96	**99**
at.bmp	3	98	91	100	95	98	97	**99**	**99**	70	96	**99**	86	56	87	91
ef.bmp	4	93	96	95	100	93	92	95	94	74	91	94	91	60	92	96
imagesCA6MA9NM.bmp	5	99	89	98	93	100	**99**	98	**99**	68	98	98	85	54	85	89
imagesCA83PY61.bmp	6	99	88	97	92	99	100	97	98	67	98	98	84	54	84	88
imagesCACBRG4N.bmp	7	98	91	99	95	98	97	100	**99**	70	96	**99**	86	56	87	91
imagesCAQI7U2A.bmp	8	99	90	99	94	99	98	99	100	69	97	**99**	85	55	86	90
imagesCATI07Y0.bmp	9	68	78	70	74	68	67	70	69	100	66	69	82	84	81	78
imagesCAUZ25S0.bmp	10	97	87	96	91	98	98	96	97	66	100	96	83	53	83	87
imagesCAW6KN27.bmp	11	98	90	99	94	98	98	99	99	69	96	100	86	55	86	90
re.bmp	12	85	95	86	91	85	84	86	85	82	83	86	100	67	**99**	95
tr.bmp	13	54	63	56	60	54	54	56	55	84	53	55	67	100	67	63
V4.bmp	14	85	96	87	92	85	84	87	86	81	83	86	99	67	100	96
we.bmp	15	89	99	91	96	89	88	91	90	78	87	90	95	63	96	100

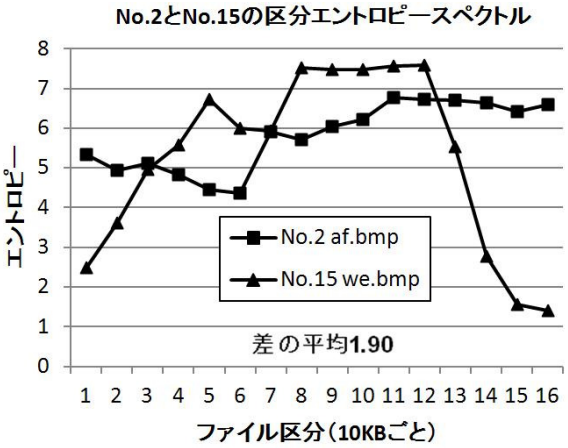

図 2 BMP ファイルの区分エントロピースペクトル

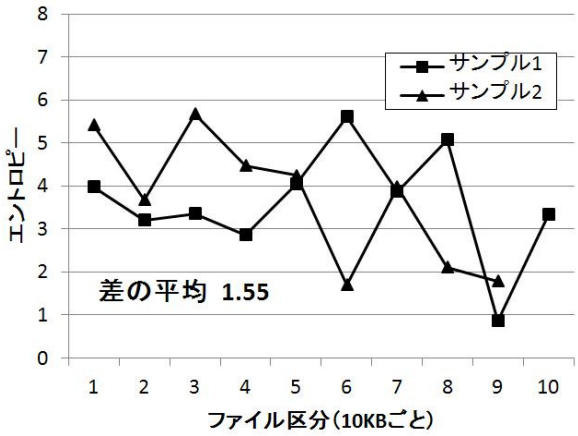

図 4 Word ファイルの区分エントロピースペクトル

しまうため、エントロピーとして意味のある値を取らない。図 1 は中身がランダムなファイル(エントロピー値＝8)のエントロピー値を区分の大きさを変えて測定したものである。10KB ではランダムファイルのエントロピー値＝8 付近を取るものの、区分を小さくしていくに従って、エントロピーの値が小さなものとなってしまう。このため分割する区分の最小値は 10KB 程度にする必要がある。

表 2 で、類似度が 99%だったペア、No.2 af.bmp と No.15 we.bmp について計算したエントロピースペクトルと差の平均の値を図 2 に示す。図のように両者のスペクトルは異なった形となることが分かり、ファイル全体のエントロピー値を比較した場合に高い類似度を取ってしまうファイルペアについても、機械的にさらなるエントロピースペクトルの評価を行うことで、この場合両者が異なるものと判断でき、課題を解決できることが分かった。表 2 で類似度 99%となった他の 12 ペアについても、同様の分析を行った結果、区分エントロピーのスペクトルの形に相違が見られ、区分エントロピー値の差の平均は 0.23～1.90 となることが分かった。

5 評価実験

同様のファイル区分エントロピー比較法の適用実験を Excel(xls)、Word(doc)、PowerPoint (ppt)、txt、jpeg のファイルについて行った。

AP ごとに Web 上から無作為に収集した 15～17 個のファイルに対して、式 5 で類似度が 99%のものを分析すると図 3、4、5 の通りとなり有意な差が見られた。

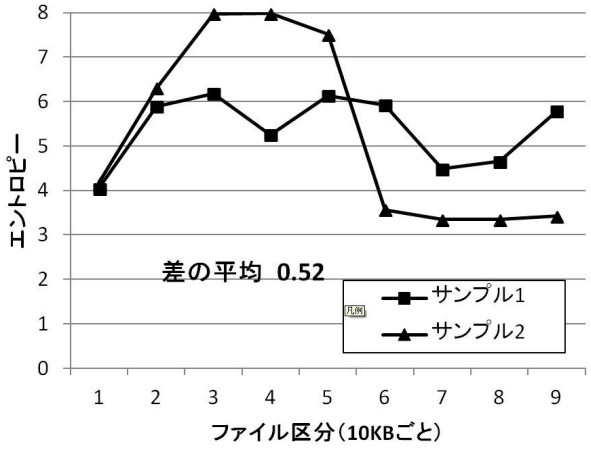

図 3 Excel ファイルの区分エントロピースペクトル

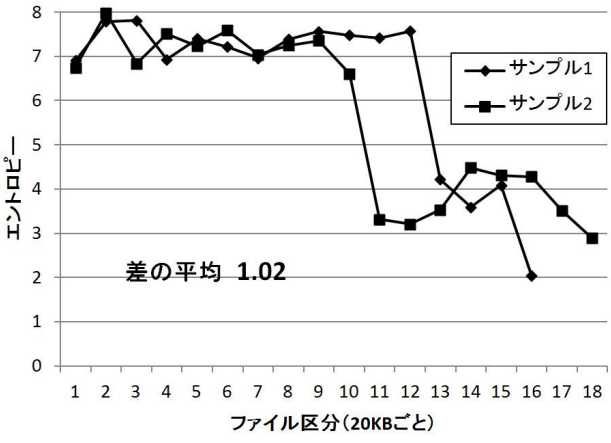

図 5 PPT ファイルの区分エントロピースペクトル

Vol.2013-DPS-154 No.13
Vol.2013-CSEC-60 No.13
2013/3/14

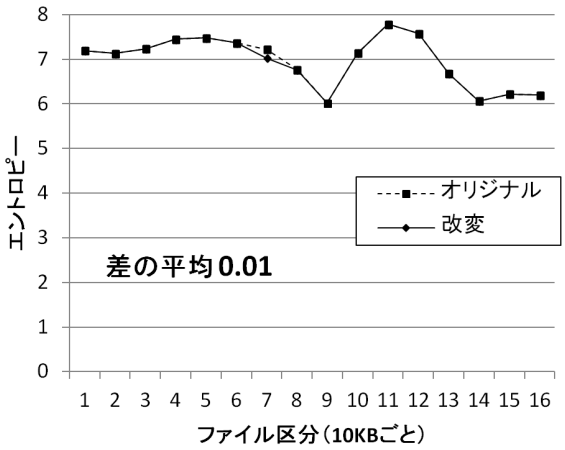

図 6 BMP 改変区分エントロピースペクトル

具体的には図 3 の Excel ファイルでは 15 ファイル中 5 個のペアについて類似度が 99%になり、スペクトルの差の平均は 0.74～2.25 であった。

図 4 の Word ファイルも同様に 15 ファイル中 1 個のペアについて類似度が 99%になり、スペクトルの差の平均は 1.55 であった。

図 5 の PowerPoint ファイルでは 17 ファイル中 2 個のペアについて類似度が 99%になり、スペクトルの差の平均は 0.63 と 1.02 であった。

次にオリジナルのファイルを改変した場合、区分エントロピースペクトルがどの程度異なったものになるか調べた。

図 6 はビットマップファイルの一部を変更（25×15 ピクセルを塗りつぶし）した場合の区分エントロピースペクトルの変化の測定結果である。

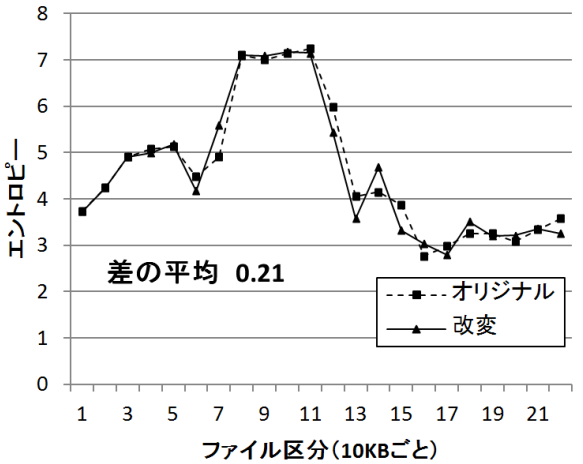

図 7 Excel 改変区分エントロピースペクトル

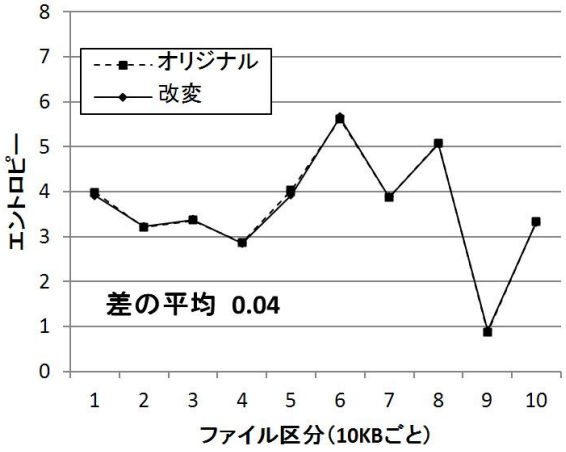

図 8 Word 改変区分エントロピースペクトル

スペクトルはほぼ一致し、スペクトルの差の平均は 0.01 であった。

同様に図 7 は Excel ファイルの一部を変更（20 セルを削除）した場合の区分エントロピースペクトルの変化の測定結果である。スペクトルはほぼ一致し、スペクトルの差の平均は 0.21 であった。

図 8 は Word ファイルの一部を変更（表中の 20 文字を削除、本文 50 文字を 5 文字程度で置換を 3 箇所）した場合の区分エントロピースペクトルの変化の測定結果である。こちらもスペクトルはほぼ一致し、スペクトルの差の平均は 0.04 であった。

図 9 は PowerPoint ファイルの一部を変更（表を 1 列削除、文字列を 20 字 2 箇所削除）した場合の区分エントロピースペクトルの変化の

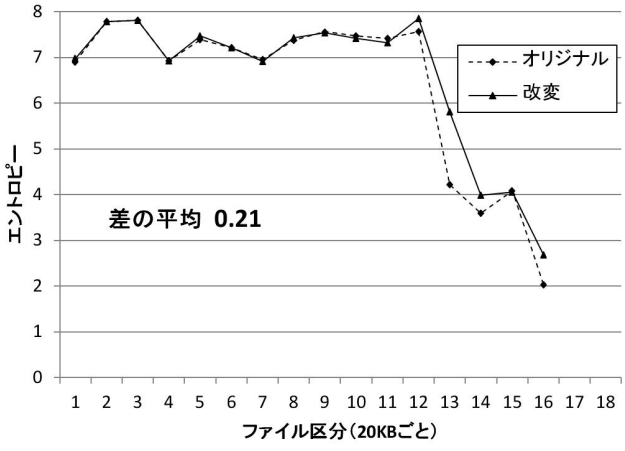

図 9 PPT 改変区分エントロピースペクトル

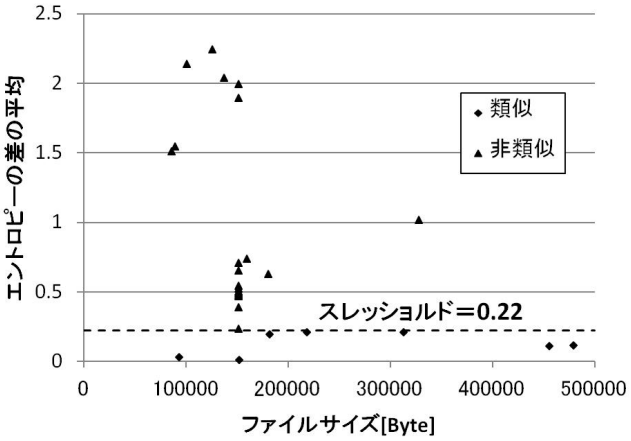

図 10 類似判定のスレッショルド

測定結果である。こちらもスペクトルはほぼ一致し、スペクトルの差の平均は 0.21 であった。

以上のような試験を繰り返して行った結果としてファイルサイズとエントロピーの差の平均をプロットした結果を図 10 に示す。図から類似ファイルのスペクトル差分平均値の最大値と非類似ファイルのスペクトル差分平均値の最小値から分解点を求めた結果、スペクトル差分の平均値 0.22 が分解点（スレッショルド）となることが分かった。

区分エントロピー測定法によって、上述したような有用な測定結果が得られた半面、エントロピーを測定するということに起因した適用限界も 2 例ほど抽出された。

1 例目はテキストファイルの区分エントロピー

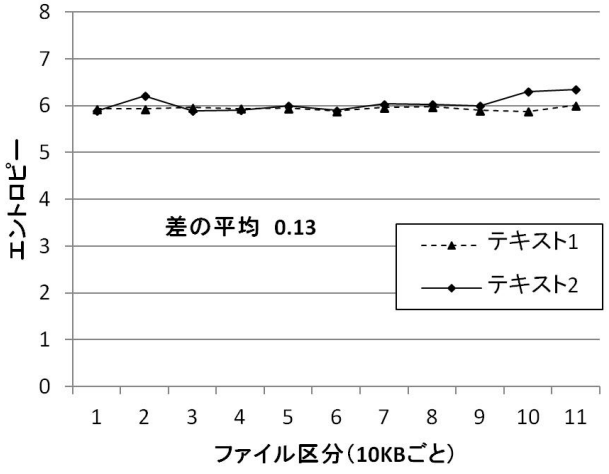

図 11 TXT ファイルのエントロピースペクトル

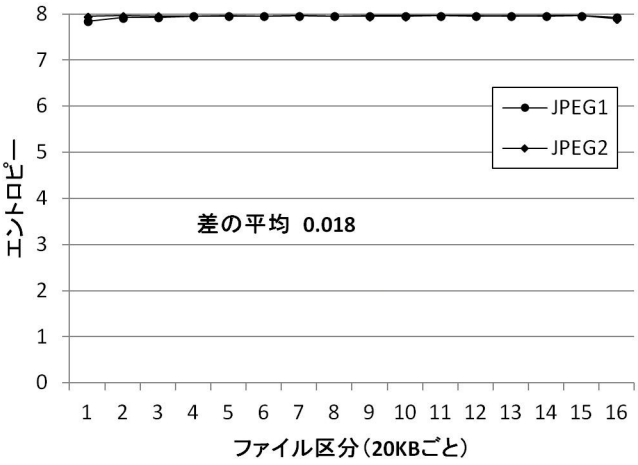

図 12 JPEG ファイルのエントロピースペクトル

測定である。図 11 に示した通り、テキストファイルは情報量としての偏りが少ないため、全く異なったテキストファイルでも同じようなエントロピースペクトルを取ってしまい分別出来なかった。ただし、これは順序考慮型のエントロピー値を比較することで改善される可能性があると思われる。

2 例目は JPEG ファイルや ZIP ファイルである。これらのファイルでは情報量を詰め込む（＝エントロピー値を高くする）ことでファイルを圧縮している。このため、図 12 のように異なったファイルでもエントロピー値 8 付近をとるため、類似ファイル、非類似の区別ができなかった。

またファイルの分割方法にも改良の余地がある。今回の評価では、ファイルの先端から固定長でファイルを切り出しているため、最後に小さな(10KB 以下の)データが残る可能性がある。前述したとおりこのようなデータについては適切なエントロピー計算が行えないため、例えばファイルを均等割りにしたりするといった対応等を施す必要がある。また、こうした場合には比較する 2 つのファイルでファイルの分割点が一致しないことが生じる、こうした場合の比較法についても、例えば直線で近似して外挿するといった方法等を検討していく必要がある。図 13 は区分データ量の違う（10KB と 20KB）同一ファイルに関するエントロピースペクトルを測定したものである。図を観察することで 10KB 以上の区分でスペクトル化したものについては区分

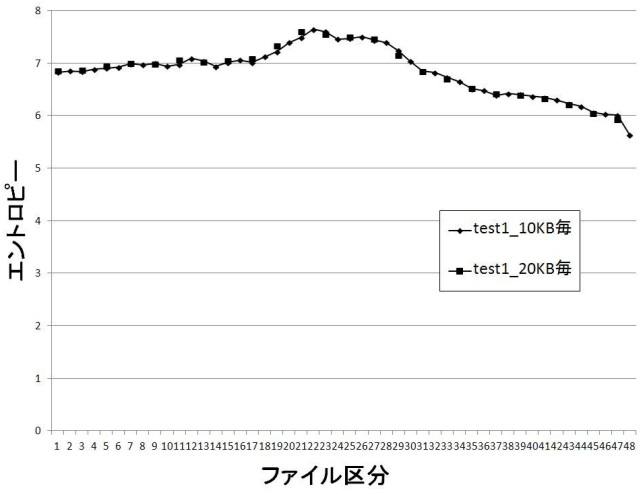

図 13 区分データ量の異なるエントロピースペクトル

データ量が異なっていても、スペクトル形がほぼ一致することがわかる。従って、区分点が一致しなくてもよく、直線外挿でも十分有意な評価をできることが期待される。こうした手法はエントロピースペクトルという連続値を評価しているから可能になるものであり、例えば先に挙げたファジーハッシュでは実行することができない。こうした点もエントロピー値を使った比較法のメリットということができる。

以上の実験によりエントロピースペクトル測定法は、類似度の識別能力が高くなるというメリットがあることが分かった、一方で従来のファイル全体のエントロピー値の計算に比べて演算量が増えるといったデメリットや、出力が 1 個の数値ではないので一次元に並べづらいといったデメリットがある。しかしながら計算量については想定最大区分数 64 倍の log 演算となるものの十分実用時間での計算が可能なものであった、また特徴量の表示に関しては、対象ファイル単体での表示が主な用途ではなく、ファイル同士の類似度評価が目的であることから、スペクトルの差の平均を表示することで 1 個の数値で表現することができ、表示上の問題はないと思われる。

6 まとめと今後の課題

本論文では、既存のエントロピーを用いた類似度判定方式の問題点を考察し、その解決策を考案した。

既存の類似度判定方式では、無関係のファイルについて、高い類似度判定をしてしまうことがかなりの頻度で起こるという問題点がある。この問題点を解決するために、本論文では比較するファイルを区分に分け、区分ごとのエントロピーを計算することでスペクトルを求め、その波形の相違を評価するという類似度判定方式を提案した。また、実測値から区分エントロピースペクトルの差分が 0.22 のスレッショルド以内の場合"類似"と判定できることを明らかにした。これにより発見したマルウェアの亜種の特定等への応用が期待できる。

今後は、評価対象の 2 つの区分エントロピースペクトルの評価方法について、exe、dll ファイルでの有効性を評価したい、また評価方法として各種統計、検定手法の導入を検討したい。またより多くのサンプルで類似度判定を行うことで、スレッショルドの適正値を求めていく、場合によっては、類似判定に SVM 等の導入も視野に入れる必要があるかもしれない。さらには、サイズが大きく異なり、包含関係にあるファイル間の類似度評価の方式を確立したい。

参考文献

[1] Mccreight et al. “System and method for entropy-based near-match analysis.” 国際特許 WO2010/107659 A1
[2]Davis et al.Guidance Software “Utilizing Entropy to Identify Undetected Malware”
[3] 松本ら“エントロピーとフォレンジック” http://www.netagent-blog.jp/archives/51451285.html:2010
[4]高田ほか”類似度を用いたファイル追跡に関する一手法の提案”CSS2012
[5] Jesse Kornblum,”Identifying almost identical files using context triggered piecewise hashing,” Digital investigation 3S(2006) pp.91-97.
[6] Tridgell Andrew. Spamsum README: http://samba.org/ftp/unpacked/junkcode/spamsum/README; 2002.
[7] 芹田ほか, “ファイル伸縮に耐性のある類似ハッシュ算出方式の考察,” IEICE Technical Report ISEC2010-54 LOIS2010-33 pp.31-36
[8] 藤井ほか, “デジタルシーケンス特徴量算出方法及びデジタルシーケンス特徴量算出装置”, 日本国公開特許広報 特開 2012-18549