

DOI: 10.37943/HOJD6191

Igor Kiyashko

PhD Student, Information Security Systems Department
igor.kiyashko1990@gmail.com, orcid.org/0000-0002-4237-0021
Satbayev University, Kazakhstan

Bakytzhan Zhumadilla

PhD Student, Department of Computer Science and ICT
zhumadilla_bakytzhan@korkyt.kz, orcid.org/0009-0005-0976-6250
Korkyt Ata Kyzylorda University, Kazakhstan

Nurbek Konyrbaev

PhD, Department of Computer Science
n.konyrbaev@mail.ru, orcid.org/0000-0002-8788-4149
Korkyt Ata Kyzylorda University, Kazakhstan

Gul Jussupova

Candidate of Technical Sciences, Department of Digitalization and Public Services
g.jussupova@sci.gov.kz, orcid.org/0009-0009-7205-3666
Ministry of Science and Higher Education of the Republic of Kazakhstan,
Kazakhstan

LIMITS OF ENTROPY-BASED RANSOMWARE DETECTION: SEPARATING ENCRYPTED FROM COMPRESSED FRAGMENTS

Abstract: Ransomware detectors and forensic tools often treat a file or disk block as encrypted when its Shannon entropy approaches 8 bits per byte. Compressed formats reach almost the same entropy, which causes false alarms, and partial encryption or text encoding of the ciphertext can lower entropy below common thresholds. This study measures how far byte-level statistics separate encrypted from compressed data and how the answer depends on fragment size. A corpus of 595 files in 17 formats, including Deflate, bzip2 and LZMA compressed archives, documents and images, was built from the Govdocs1 collection and encrypted with AES-256 or ChaCha20; evasion variants used intermittent encryption, Base64 encoding and encryption of the first megabyte. Fragments of 512 bytes, 4 kilobytes and 64 kilobytes were described by entropy, the chi-square statistic and four further byte statistics, and threshold rules, logistic regression and random forest were evaluated with cross-validation grouped by file. Entropy and chi-square ranked fragments almost identically, as expected from their known asymptotic equivalence, so chi-square did not separate the classes better; its advantage was a size-independent threshold. A fixed entropy threshold of 7.9 bits per byte missed every encrypted 512-byte fragment and flagged 73% of unencrypted 64-kilobyte fragments. Deflate-based formats became separable only with large fragments, whereas xz and 7z archives remained close to chance for every method. Base64 encoding removed detection completely, and intermittent encryption was rarely detected in large fragments of uncompressed files. Byte statistics can therefore show that data are not plaintext but cannot confirm that compressed data are encrypted; detectors need size-aware thresholds, decoding of text encodings and format validation.

Keywords: ransomware detection; Shannon entropy; chi-square test; compressed data; file fragment classification; intermittent encryption; digital forensics

Introduction

Ransomware encrypts or steals the data of its victims and demands payment for their release, and it remains one of the most damaging forms of malware for organisations of every size [1]. Recent reviews describe a shift towards double extortion, in which data are exfiltrated before they are encrypted, but the encryption of user files is still the stage that causes the most direct loss and the stage that most detectors try to observe [2-3]. Detectors monitor application programming

interface calls, input and output requests or file system activity, often with the aim of early detection [4], and a review of 118 anti-ransomware studies found that their evaluations often have limited internal and external validity [5].

A recurring signal in file system approaches is the randomness of the data written by a process. CryptoDrop combines the Shannon entropy of written files with file type changes and similarity digests [6], ShieldFS uses the entropy of write operations among the features of its per-process models [7] and UNVEIL compares the entropy of read and write buffers at the same file offset [8]. The same idea is applied in storage and cloud settings, where a file whose entropy approaches the maximum of 8 bits per byte is treated as encrypted; on a set of document and system files one such method reported detection without any false positive [9]. In digital forensics, high-entropy fragments are likewise the candidates for encrypted content [10].

Well-compressed data, however, are also close to random. ZIP and 7Z archives, Office Open XML documents, JPEG and PNG images and most audio and video formats have entropy values close to those of ciphertext. Pont et al. [11] derived thresholds for the five statistics used in the anti-ransomware literature (Shannon entropy, chi-square, arithmetic mean, Monte Carlo estimate of π and serial correlation) and tested them on 84,327 image, compressed and encrypted files; false positive rates were often above 50% and sometimes above 90%. Davies et al. described the same difficulty and proposed a classifier based on the entropy curve of file headers [12], compared 53 tests for the identification of ransomware-encrypted files [13] and released NapierOne, a modern mixed file data set that complements the older Govdocs1 corpus [14].

Separating compressed from encrypted data has also been studied directly. HEDGE applies randomness tests to network packets [15], and a recent study reached an F1 score of about 0.95 on 64 KB segments of AES and Camellia ciphertext against ZIP, RAR, BZIP2 and GZIP data using entropy, chi-square and NIST features [16]. De Gaspari et al. [17] found that byte-distribution statistics fail on a large standardised data set and proposed a learned classifier with accuracies of about 82% at 512 B and 92% at 8 KB, and a systematic survey covers the problem [18]. General fragment classifiers such as FiFTy [19], ByteRCNN [20] and byte embeddings [21] still find compressed and encrypted fragments difficult because they lack exploitable patterns [10], and their accuracy falls with fragment size: ByteRCNN reaches 83.9% at 4096 B and 71.1% at 512 B, and a recent hierarchical classifier 76.3% over 75 file types on 512-byte blocks [22].

Ransomware can also weaken this signal deliberately. LockFile encrypted alternating 16-byte blocks, so that the chi-square value of an encrypted text file stayed at 1,789,811 instead of about 334 after full encryption [23], and families such as BlackCat, PLAY, Black Basta and Agenda encrypt only parts of each file, which keeps encrypted files similar to their originals [24]. Storage-level classifiers with online learning have been tested against partial and intermittent encryption, where a Hoeffding tree on write-buffer statistics performed best [25]. A second route changes the representation of the ciphertext. Base64 encoding lowers entropy enough to neutralise threshold detectors [26], format-preserving transformations give ciphertext an entropy similar to that of plaintext [27], entropy sharing masks high-entropy blocks [28], and machine learning countermeasures against encoded ciphertext have been proposed [29]. Behavioural classifiers can be evaded as well, for example by spreading the encryption over many cooperating processes [30].

Three questions remain open in this literature. First, entropy and the chi-square statistic are usually treated as different tests, and chi-square is used by some protection software in place of entropy [23], but the practical consequences of the close mathematical relation between the two are rarely discussed. Second, results are usually reported for one fragment size or pooled over formats, which hides where false positives come from. Third, when fragments are split at random, fragments of one file can appear in both training and test data and inflate the accuracy of learned detectors; a split by file avoids this.

This study therefore measures, on one corpus and under one protocol, how far byte-level statistics separate encrypted from compressed data. RQ1 asks how well Shannon entropy separates ciphertext from each compressed format; RQ2, whether the chi-square statistic, the Monte Carlo

estimate of π , the serial correlation or the mean byte value separate them better; RQ3, how performance depends on the fragment size (512 B, 4 KB and 64 KB); and RQ4, how robust the detectors are to intermittent encryption and to Base64 encoding of the ciphertext. The working hypothesis was that entropy hardly separates ciphertext from strong compression, that chi-square and other uniformity statistics are clearly better from 4 KB upwards, and that all measures degrade on small blocks and under intermittent encryption. The results show that entropy and chi-square rank fragments almost identically and differ mainly in calibration, that separability ranges from almost complete for JPEG from 4 KB upwards to near chance for LZMA archives at every size, that Base64 encoding defeats all byte-level detectors, and that under intermittent encryption the detectors either miss mixed blocks or flag compressed files.

Methods and Materials

Corpus

The corpus was assembled from Govdocs1 [31], a public collection of about one million files crawled from web servers of the United States government that serves as a common benchmark in file type and fragment research [14]. Eight thread archives (000 to 007) with 7,888 files were downloaded. For ten formats present in the collection (TXT, CSV, HTML, XML, PDF, DOC, XLS, PPT, JPEG and GIF), 35 files per format were sampled at random among files between 8 KB and 8 MB. Sizes in this paper use binary multiples (1 KB = 1,024 bytes, 1 MB = 1,048,576 bytes). A binary file was accepted only when it started with the signature expected for its extension, and a text file only when at least 90% of its first 4 KB were printable ASCII characters or white space.

Govdocs1 contains few modern compressed containers, so seven further formats were produced locally from a disjoint pool of 1,536 other Govdocs1 files. PNG images were obtained by decoding Govdocs1 JPEG photographs and storing them losslessly with the default settings of Pillow. DOCX documents were written with python-docx from the text of TXT, HTML, XML and CSV files. ZIP (Deflate), TAR.GZ, TAR.BZ2 and TAR.XZ archives were created with the Python standard library and 7Z archives (LZMA2) with py7zr, in all cases with default compression settings. Each archive holds a random group of pool files whose total size was drawn log-uniformly between 256 KB and 6 MB, so archives mix text, office documents, PDF files and images. The resulting 595 files (521.6 MB, median size 291.3 KB) are summarised in Table 1. JPEG, GIF, PNG, DOCX, ZIP, TAR.GZ, TAR.BZ2, TAR.XZ and 7Z are called compressed formats below because their payload is entropy coded; PDF and PPT files often embed compressed streams and are reported next to them. The whole workflow is shown in Figure 1.

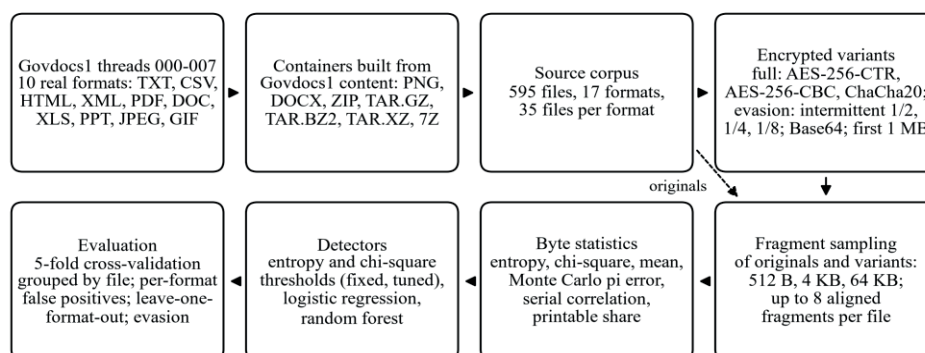


Figure 1. Experimental pipeline, from the Govdocs1 source files to the evaluation of detectors

Table 1. Composition of the corpus (35 files per format) and number of fragments per class for each fragment size

Format	Source	Median size (KB)	Fragments per class		
			512 B	4 KB	64 KB
TXT	Govdocs1	108.9	280	247	86
CSV	Govdocs1	37.7	280	239	90
HTML	Govdocs1	19.0	280	177	10
XML	Govdocs1	21.1	280	181	20
PDF	Govdocs1	206.3	280	269	134
DOC	Govdocs1	212.5	280	272	140
XLS	Govdocs1	155.0	280	264	122
PPT	Govdocs1	694.5	280	280	193
JPEG	Govdocs1	52.3	280	248	60
GIF	Govdocs1	33.7	280	202	37
PNG	Built from Govdocs1	472.3	280	280	190
DOCX	Built from Govdocs1	240.0	280	280	140
ZIP	Built from Govdocs1	1417.7	280	280	264
TAR.GZ	Built from Govdocs1	1531.1	280	280	263
TAR.BZ2	Built from Govdocs1	1580.2	280	280	264
TAR.XZ	Built from Govdocs1	1353.2	280	280	250
7Z	Built from Govdocs1	1718.9	280	280	257

Encryption and evasion scenarios

Every file was encrypted in full with one of three ciphers of the cryptography library: AES-256 in counter mode, AES-256 in cipher block chaining mode with PKCS#7 padding, and ChaCha20. The cipher was assigned in rotation within each format after a seeded shuffle, which gives 12, 12 and 11 files per format. Keys and nonces came from a seeded generator, and decryption of sample files reproduced the originals. Three evasion scenarios were applied to every file with AES-256 in counter mode. In intermittent encryption every second, fourth or eighth 4 KB block is encrypted and the other blocks are left intact [23-24]. In the encoding scenario the fully encrypted file is converted to Base64 text, which maps ciphertext onto 64 printable symbols [26]. In the partial scenario only the first 1 MB is encrypted, which was applied to the 167 files larger than 1 MB.

Fragments and byte statistics

For each variant and each fragment size N of 512, 4,096 and 65,536 bytes, up to eight non-overlapping fragments aligned to multiples of N were drawn at random, as storage monitors and carving tools read aligned sectors, pages and clusters. A file shorter than N contributes no fragment of that size, so 64 KB fragments come from 423 of the 595 files. In total 74,075 fragments were analysed.

Each fragment is described by six statistics that follow the definitions of the ent test program. Let n_i be the number of occurrences of byte value i in a fragment of N bytes. Shannon entropy [32] is computed as in equation (1) and expressed in bits per byte, so that ideal random data approach 8.

$$H = - \sum_{i=0}^{255} p_i \log_2 p_i, \quad p_i = \frac{n_i}{N} \quad (1)$$

The chi-square statistic (2) compares the byte histogram with the uniform distribution; under uniformity it follows a chi-square distribution with 255 degrees of freedom.

$$\chi^2 = \sum_{i=0}^{255} \frac{(n_i - E)^2}{E}, \quad E = \frac{N}{256} \quad (2)$$

The arithmetic mean of the bytes (127.5 for random data) was added, together with the relative error of the Monte Carlo estimate of π , in which consecutive groups of six bytes are read as two 24-bit coordinates and the share of points inside the inscribed quarter circle estimates $\pi/4$. The serial correlation coefficient r in equation (3) measures the dependence between adjacent bytes u_j , with the last byte paired with the first. The last statistic is the share of printable ASCII characters (0x20 to 0x7E, tab, line feed and carriage return), which equals $98/256 \approx 0.383$ for uniform bytes. The Kullback-Leibler divergence to the uniform distribution equals $8 - H$ bits and therefore adds no information beyond equation (1).

$$r = \frac{N \sum_{j=1}^N u_j u_{j+1} - (\sum_{j=1}^N u_j)^2}{N \sum_{j=1}^N u_j^2 - (\sum_{j=1}^N u_j)^2}, \quad u_{N+1} = u_1 \quad (3)$$

The first two statistics are linked by known results, which this study uses to calibrate detection rather than presents as new. With $G = 2 \sum n_i \ln(n_i/E)$, the likelihood-ratio statistic of the uniform hypothesis, which equals $2N$ times the Kullback-Leibler divergence of the byte histogram from the uniform distribution in nats, the identity $8 - H = G/(2N \ln 2)$ holds exactly, and Pearson's χ^2 is the second-order approximation of G for histograms close to uniform, which gives equation (4).

$$8 - H = \frac{G}{2N \ln 2} \approx \frac{\chi^2}{2N \ln 2}, \quad G = 2 \sum_{i=0}^{255} n_i \ln \frac{n_i}{E} \quad (4)$$

Under uniformity both G and χ^2 follow asymptotically a chi-square distribution with 255 degrees of freedom [33], so the expected entropy of an ideal random fragment is approximately $8 - 255/(2N \ln 2)$, that is 7.641, 7.955 and 7.997 bits per byte for 512 B, 4 KB and 64 KB fragments. This is the first-order bias of the plug-in entropy estimator, known as the Miller-Madow correction, which is documented for short samples of bytes drawn from maximum-entropy sources [34]; the approximation loses accuracy when the expected count per byte value, $E = N/256$, is small. A fixed entropy threshold therefore corresponds to a different trade-off between misses and false alarms at each fragment size, whereas the null distribution of χ^2 does not depend on N .

Detectors

Four families of detectors were compared. The entropy rule labels a fragment as encrypted when H reaches a threshold, either fixed at 7.9 bits per byte or tuned on the training folds by maximising Youden's index (true positive rate minus false positive rate). Absolute thresholds of deployed detectors are rarely published [11]; reported values range from 7.58 bits per byte (a normalised 0.948, regarded as very high in ShieldFS [7], as noted by Pont et al. [11]) through 7.95 [13] to 7.99 [11]. The value 7.9 was taken as representative of this range, and the other values are evaluated in Table S3. The chi-square rule labels a fragment as encrypted when χ^2 does not exceed either 310.46, the 0.99 quantile of the chi-square distribution with 255 degrees of freedom (Pont et al. [11] used the 0.95 quantile, 293.25), or a tuned value. Logistic regression with L2 regularisation and random forest with 300 trees (scikit-learn defaults otherwise) use all six statistics, with χ^2 entered as the decimal logarithm of $\chi^2 + 1$ and standardised inputs for logistic regression; both label a fragment as encrypted at a predicted probability of at least 0.5.

Evaluation protocol

Fragments of fully encrypted copies form the positive class and fragments of original files the negative class, with equal numbers of fragments. Performance was estimated by five-fold

cross-validation grouped by source file (StratifiedGroupKFold, random state 42), so that an original file and all its variants fall in the same fold, with seven files of each format per fold; thresholds and models were fitted on the training folds only. Precision, recall, F1 score ($2TP/(2TP + FP + FN)$), false positive rate (FPR) and the area under the receiver operating characteristic curve (ROC-AUC) are reported as mean $\pm$ standard deviation over folds; precision is reported as n/a when a detector flags no fragment, and a fixed and a tuned threshold on one statistic share its ROC-AUC. Per-format FPR was computed from pooled out-of-fold predictions with 95% bootstrap intervals over files (1,000 resamples). ROC-AUC was also computed between all encrypted fragments and the originals of each format, using the distance of each statistic from its value for random data ($8 - H$, χ^2 , $|\text{mean} - 127.5|$, the π error, $|r|$ and $|\text{printable share} - 0.383|$).

Three further analyses were run. In the leave-one-format-out test, all detectors were trained on 16 formats and tested on the originals and encrypted copies of the remaining one. In the evasion test, the models of each fold, trained on originals and full encryption only, scored the evasion variants of the test files, and the share of fragments labelled as encrypted is reported, also per file (a file is flagged when any of its sampled fragments is flagged). Finally, the permutation importance of the random forest inputs was measured on the test folds as the decrease in ROC-AUC over ten permutations, for each statistic alone and for entropy and χ^2 permuted jointly.

Software and data availability

The experiment was implemented in Python 3.11.9 with NumPy 2.4.6, pandas 3.0.6, SciPy 1.17.1, scikit-learn 1.9.1, cryptography 50.0.1, py7zr 1.1.3, python-docx 1.2.0, Pillow 12.3.0 and Matplotlib 3.11.2, and was run on a single workstation with 64 logical processors under Windows Server 2019. All random operations use the seed 42, and the pipeline from download to figures runs with one command (python run_all.py) in about 10 minutes once the Govdocs1 archives are downloaded. Source code, file manifests, fragment features and all result tables are archived at [ZENODO DOI]. The Govdocs1 files are public and are downloaded by the pipeline; generated containers and encrypted variants are recreated from the seeds.

Results

Distributions of entropy and chi-square

Figure 2 and Figure 3 show the distribution of entropy and χ^2 for each format, class and fragment size. Text formats (TXT, CSV, HTML, XML) and XLS files stayed far from ciphertext at every size. The entropy of encrypted fragments depended on the fragment size, with means of 7.590, 7.955 and 7.997 bits per byte for 512 B, 4 KB and 64 KB. At 4 KB and 64 KB these values match the prediction from equation (4); at 512 B the mean lies 0.050 bits per byte below the predicted 7.641. The mean χ^2 of the encrypted fragments was between 254.7 and 255.2 at every size, whereas the mean G statistic, which is proportional to the entropy deficit, was 290.7 at 512 B, 257.6 at 4 KB and 255.3 at 64 KB (Table 2). At 512 B the fixed threshold of 7.9 bits per byte lies above all encrypted fragments (Figure 2). Fragments of 7Z and TAR.XZ archives overlapped with encrypted fragments in both statistics at every size, while TAR.BZ2, TAR.GZ, ZIP, DOCX and PNG overlapped at 512 B and moved away from ciphertext as fragments grew (Figure 3). Base64 encoded ciphertext had a median entropy between 5.910 and 5.999 bits per byte.

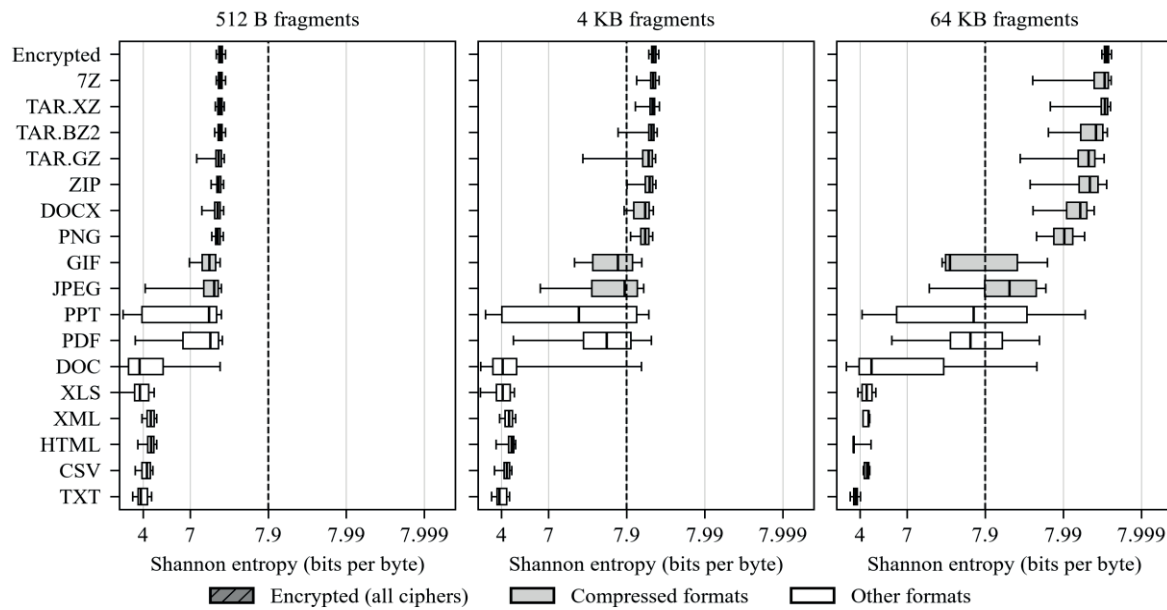


Figure 2. Shannon entropy of original and fully encrypted fragments by format and fragment size. The horizontal axis is logarithmic in 8 minus entropy; boxes show quartiles and whiskers the 5th and 95th percentiles; the dashed line marks 7.9 bits per byte

Over all original and fully encrypted fragments, the Spearman rank correlation between entropy and $-\chi^2$ was 0.973 at 512 B, 0.996 at 4 KB and 0.9995 at 64 KB, and among fragments with entropy above 7 bits per byte it was 0.949, 0.994 and 0.9998 (Table 2). The tuned entropy threshold moved from 7.539 at 512 B to 7.997 at 64 KB, whereas the tuned χ^2 threshold stayed between 291.8 and 310.6, close to the fixed critical value of 310.46.

Table 2. Entropy, chi-square and G statistic of encrypted fragments, tuned thresholds (mean $\pm$ standard deviation over folds) and rank correlation between entropy and chi-square

Quantity	512 B	4 KB	64 KB
Mean entropy of ciphertext (bits per byte)	7.5905	7.9546	7.9972
5th and 95th percentiles of ciphertext entropy	7.5347; 7.6459	7.9479; 7.9610	7.9968; 7.9976
Miller-Madow approximation $8 - 255/(2N \ln 2)$	7.6407	7.9551	7.9972
Tuned entropy threshold	7.5386 ± 0.0036	7.9474 ± 0.0008	7.9966 ± 0.0000
Mean χ^2 of ciphertext	255.0	254.7	255.2
Mean G of ciphertext, $G = 2N \ln 2 (8 - H)$	290.7	257.6	255.3
Tuned χ^2 threshold	291.8 ± 0.4	297.5 ± 3.3	310.6 ± 4.1
Spearman $\rho(H, -\chi^2)$, all fragments	0.9731	0.9962	0.9995
Spearman $\rho(H, -\chi^2)$, fragments with $H > 7$	0.9491	0.9944	0.9998
Median entropy of Base64 ciphertext (bits per byte)	5.910	5.989	5.999

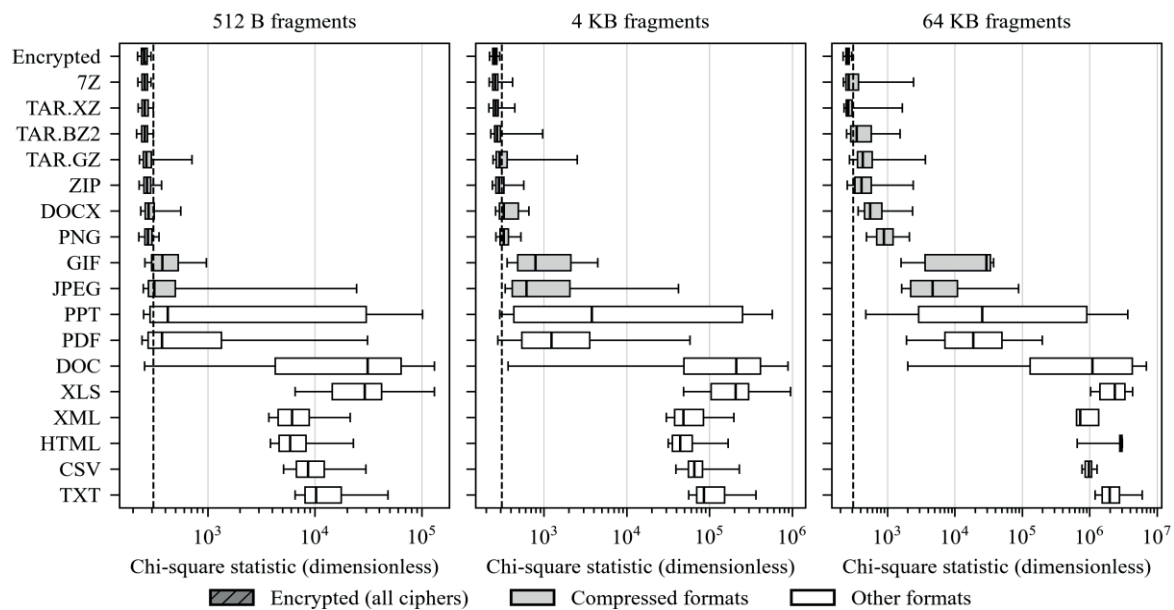


Figure 3. Chi-square statistic of the byte histogram by format and fragment size (logarithmic axis). Boxes show quartiles and whiskers the 5th and 95th percentiles; the dashed line marks the critical value 310.46 (255 degrees of freedom, 1% level)

Detection performance and fragment size

Table 3 and Figure 4 summarise the cross-validated performance. At 512 B the fixed 7.9 rule flagged no fragment, and the other detectors reached F1 scores of 0.801 to 0.808 with FPR of 0.353 to 0.461. At 4 KB the fixed 7.9 rule detected every encrypted fragment but also flagged 51.7% of the originals; the tuned entropy and χ^2 rules lowered the FPR to 0.276 and 0.280, and the random forest to 0.234 with the best F1 score, 0.874. At 64 KB the FPR of the fixed 7.9 rule rose to 0.729, the tuned thresholds and the fixed χ^2 rule reached 0.228 to 0.229, and the random forest 0.194, with an F1 score of 0.902 and a ROC-AUC of 0.910. The fixed χ^2 rule maintained a recall of 0.991 at every size. No other published fixed entropy value worked at all three sizes (Table S3): at 512 B the highest recall was 0.580 (7.58 bits per byte), whereas at 64 KB every value flagged at least 54.5% of the originals and 78.0% of the compressed ones. The ROC-AUC of entropy and χ^2 differed by at most 0.005, and the two models added at most 0.014 to that of entropy. With only the nine compressed formats as negatives, the ROC-AUC of entropy was 0.689, 0.801 and 0.860 at 512 B, 4 KB and 64 KB, and that of the random forest 0.702, 0.810 and 0.871 (Figure 4b).

Table 3. Detection of fully encrypted fragments against all original fragments (mean $\pm$ standard deviation over five folds grouped by file). n/a: the rule flagged no fragment, so precision is undefined; $F1 = 2TP/(2TP + FP + FN)$ is then 0

Size	Detector	Precision	Recall	F1	FPR	ROC-AUC
512 B	Entropy, fixed 7.9	n/a	0.000 $\pm$ 0.000	0.000 $\pm$ 0.000	0.000 $\pm$ 0.000	0.818 $\pm$ 0.012
	Entropy, tuned	0.699 $\pm$ 0.011	0.938 $\pm$ 0.010	0.801 $\pm$ 0.008	0.404 $\pm$ 0.020	0.818 $\pm$ 0.012
	Chi-square, fixed	0.683 $\pm$ 0.009	0.991 $\pm$ 0.002	0.808 $\pm$ 0.006	0.461 $\pm$ 0.019	0.823 $\pm$ 0.010
	Chi-square, tuned	0.704 $\pm$ 0.010	0.947 $\pm$ 0.011	0.808 $\pm$ 0.009	0.399 $\pm$ 0.018	0.823 $\pm$ 0.010
	Logistic regression	0.715 $\pm$ 0.009	0.924 $\pm$ 0.014	0.806 $\pm$ 0.009	0.369 $\pm$ 0.016	0.831 $\pm$ 0.013
	Random forest	0.721 $\pm$ 0.010	0.909 $\pm$ 0.019	0.804 $\pm$ 0.012	0.353 $\pm$ 0.015	0.825 $\pm$ 0.013
4 KB	Entropy, fixed 7.9	0.659 $\pm$ 0.008	1.000 $\pm$ 0.000	0.795 $\pm$ 0.006	0.517 $\pm$ 0.020	0.886 $\pm$ 0.003
	Entropy, tuned	0.777 $\pm$ 0.009	0.958 $\pm$ 0.019	0.858 $\pm$ 0.004	0.276 $\pm$ 0.018	0.886 $\pm$ 0.003
	Chi-square, fixed	0.761 $\pm$ 0.007	0.991 $\pm$ 0.003	0.861 $\pm$ 0.005	0.311 $\pm$ 0.013	0.887 $\pm$ 0.003
	Chi-square, tuned	0.775 $\pm$ 0.009	0.965 $\pm$ 0.010	0.860 $\pm$ 0.005	0.280 $\pm$ 0.015	0.887 $\pm$ 0.003
	Logistic regression	0.786 $\pm$ 0.006	0.978 $\pm$ 0.007	0.872 $\pm$ 0.002	0.266 $\pm$ 0.011	0.896 $\pm$ 0.003
	Random forest	0.804 $\pm$ 0.007	0.959 $\pm$ 0.009	0.874 $\pm$ 0.005	0.234 $\pm$ 0.011	0.891 $\pm$ 0.004

Size	Detector	Precision	Recall	F1	FPR	ROC-AUC
64 KB	Entropy, fixed 7.9	0.579 ± 0.011	1.000 ± 0.000	0.733 ± 0.009	0.729 ± 0.033	0.902 ± 0.008
	Entropy, tuned	0.813 ± 0.015	0.990 ± 0.004	0.893 ± 0.010	0.229 ± 0.022	0.902 ± 0.008
	Chi-square, fixed	0.813 ± 0.013	0.991 ± 0.005	0.893 ± 0.009	0.228 ± 0.019	0.902 ± 0.008
	Chi-square, tuned	0.813 ± 0.015	0.990 ± 0.006	0.893 ± 0.010	0.228 ± 0.023	0.902 ± 0.008
	Logistic regression	0.811 ± 0.016	0.999 ± 0.002	0.895 ± 0.010	0.233 ± 0.024	0.902 ± 0.010
	Random forest	0.835 ± 0.007	0.981 ± 0.004	0.902 ± 0.004	0.194 ± 0.010	0.910 ± 0.008

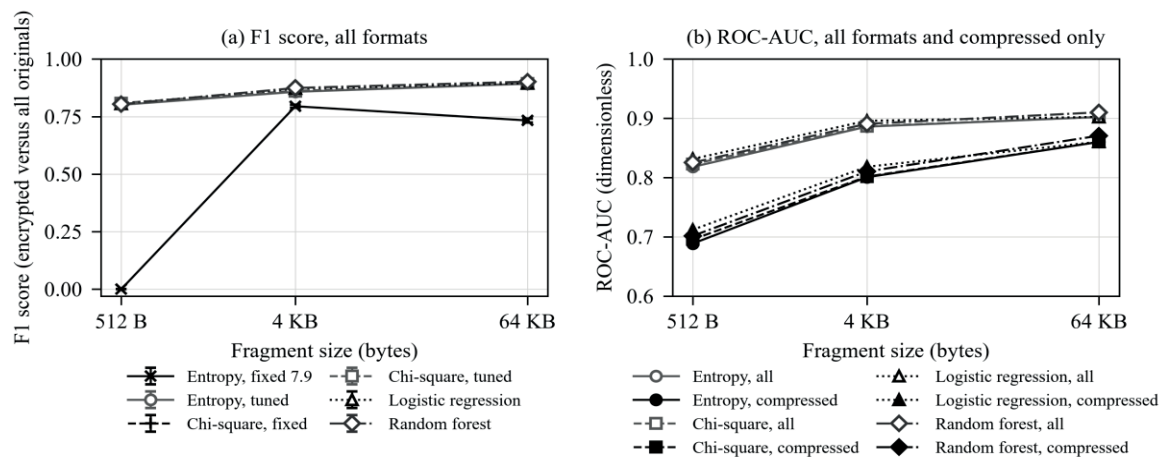


Figure 4. Effect of fragment size: (a) F1 score of each detector against all original fragments (mean and standard deviation over folds); (b) ROC-AUC of entropy, chi-square and the two models against all formats (open markers) and against compressed formats only (filled markers)

Separability by format

Table 4 lists the ROC-AUC of entropy, χ^2 and the random forest for every format with compressed content. JPEG, GIF, DOC and PDF fragments were separated from ciphertext with a ROC-AUC of at least 0.963 from 4 KB upwards. The Deflate-based formats (ZIP, TAR.GZ, DOCX and PNG) reached an entropy ROC-AUC of 0.652 to 0.736 at 512 B, 0.789 to 0.942 at 4 KB and 0.940 to 1.000 at 64 KB, and TAR.BZ2 reached 0.529, 0.681 and 0.890. The LZMA-based TAR.XZ and 7Z archives stayed between 0.522 and 0.629 for entropy and did not exceed 0.648 for the random forest at any size. The mean byte value, the π error, the serial correlation and the printable share never exceeded the ROC-AUC of entropy for the same format and size by more than 0.012.

Table 4. ROC-AUC between encrypted fragments and original fragments of each format for entropy (H), chi-square (χ^2) and random forest (RF)

Format	512 B			4 KB			64 KB		
	H	χ^2	RF	H	χ^2	RF	H	χ^2	RF
7Z	0.522	0.535	0.522	0.543	0.546	0.564	0.627	0.627	0.647
TAR.XZ	0.560	0.553	0.562	0.597	0.595	0.594	0.629	0.629	0.648
TAR.BZ2	0.529	0.535	0.562	0.681	0.689	0.686	0.890	0.891	0.896
TAR.GZ	0.652	0.654	0.691	0.809	0.809	0.837	0.970	0.970	0.980
ZIP	0.671	0.677	0.680	0.789	0.791	0.820	0.940	0.940	0.956
DOCX	0.736	0.755	0.721	0.923	0.923	0.904	1.000	1.000	1.000
PNG	0.728	0.738	0.800	0.942	0.940	0.966	1.000	1.000	1.000
GIF	0.927	0.930	0.922	0.999	0.999	0.997	1.000	1.000	1.000
JPEG	0.872	0.883	0.853	1.000	1.000	0.998	1.000	1.000	1.000
PPT	0.906	0.907	0.909	0.982	0.983	0.983	0.974	0.974	0.982
PDF	0.850	0.866	0.853	0.971	0.972	0.963	1.000	1.000	1.000
DOC	0.952	0.951	0.955	0.994	0.995	0.992	1.000	1.000	1.000
All compressed	0.689	0.696	0.702	0.801	0.802	0.810	0.860	0.860	0.871

Format	512 B			4 KB			64 KB		
	H	χ^2	RF	H	χ^2	RF	H	χ^2	RF
All formats	0.818	0.823	0.825	0.886	0.887	0.891	0.902	0.902	0.910

Figure 5 shows where the false positives occur. The fixed 7.9 rule flagged 89% to 99% of the 4 KB fragments of every Deflate, bzip2 and LZMA format and 47% of JPEG fragments, and at 64 KB between 97% and 100% of the fragments of these formats and 73% of JPEG fragments. The tuned rules and the models held the FPR for JPEG, GIF, PDF and DOC at or below 0.082 at 4 KB and at zero at 64 KB, where PNG and DOCX also reached zero (0.007 for DOCX with logistic regression). At 4 KB the random forest reduced the FPR of PNG from 0.279 (tuned entropy) to 0.104, of ZIP from 0.575 to 0.421 and of TAR.GZ from 0.511 to 0.407. For TAR.XZ and 7Z every detector except the fixed entropy rule had an FPR between 0.69 and 0.98 at all sizes, and no detector produced a false positive on TXT, CSV, HTML, XML or XLS.

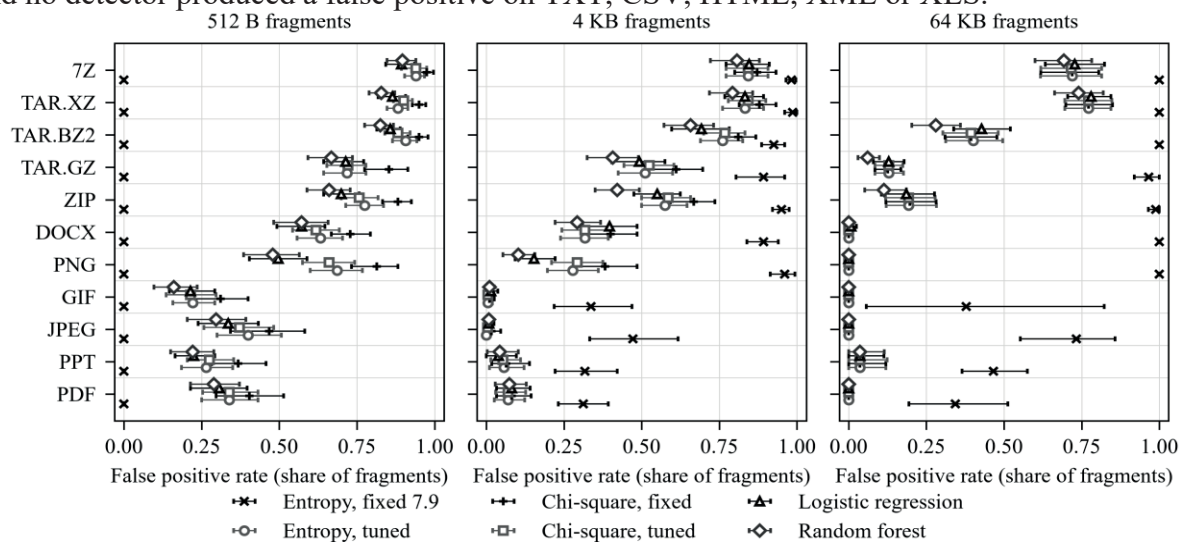


Figure 5. False positive rate by format for each detector and fragment size (out-of-fold predictions, bars show 95% bootstrap intervals over files). TXT, CSV, HTML, XML and XLS had no false positives under any detector; values for DOC are given in Table S1

Leaving a format out of training changed its FPR by at most 0.076 (random forest, TAR.BZ2, 64 KB) and on average by at most 0.012, and the recall on encrypted copies of the held-out format stayed at or above 0.85 for every detector, format and size.

Evasion

Figure 6 reports the share of fragments flagged in each evasion scenario for non-compressed and compressed source files. No detector flagged any Base64 fragment. With 512 B and 4 KB fragments every aligned fragment of an intermittently encrypted file is either fully encrypted or fully intact; the encrypted ones were flagged at rates of 0.889 to 1.000 (except by the fixed 7.9 rule at 512 B) and the intact ones at the rate of the original format. At 64 KB every fragment mixes encrypted and intact blocks, and for non-compressed source files the trained detectors and the fixed χ^2 rule flagged only 0.9% to 2.4% of them, while the fixed 7.9 rule flagged 22.8% to 32.8%, compared with 20.4% of the intact originals. For compressed source files the trained detectors flagged 31.0% to 53.7% of such fragments, against 28.0% to 33.6% of the originals. When only the first 1 MB was encrypted, fragments inside the encrypted region were flagged at rates of 0.91 to 1.00 (except by the fixed 7.9 rule at 512 B).

At file level, a rule that flags a file when any of its sampled 4 KB fragments is flagged found 85.4% to 93.6% of the non-compressed files with one block in eight encrypted, but it also flagged 68.9% to 91.1% of the unencrypted compressed files. With 64 KB fragments the same rule found 0.6% of the non-compressed files with one block in eight encrypted for all trained detectors.

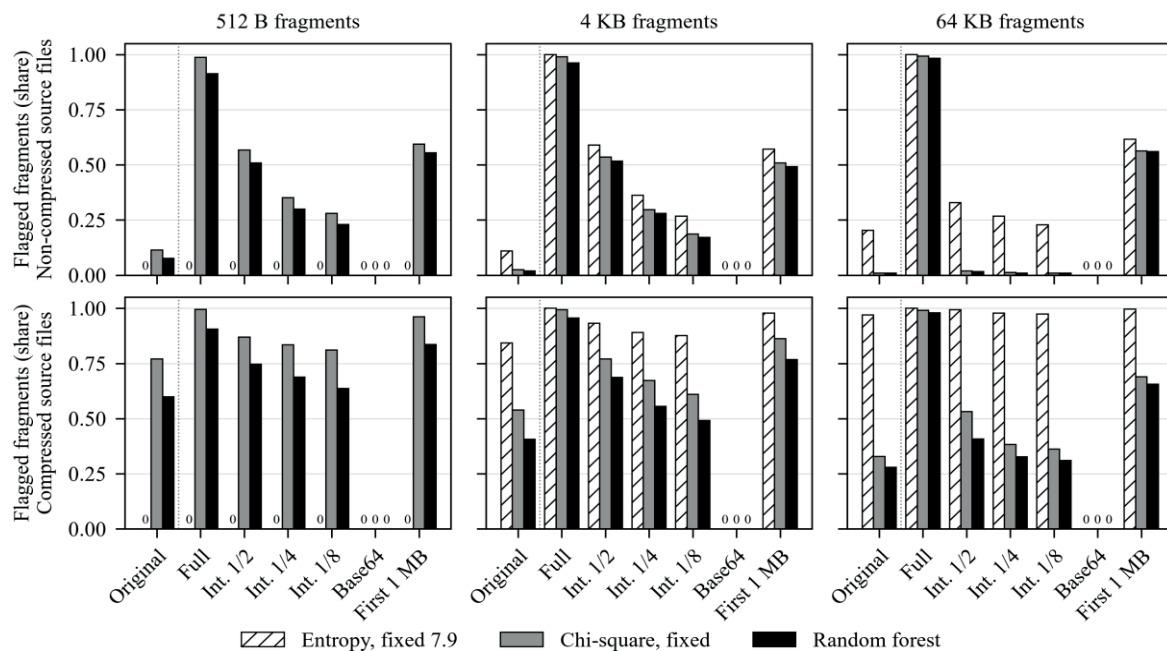


Figure 6. Share of fragments labelled as encrypted for originals, full encryption and the evasion scenarios (Int. 1/2, 1/4, 1/8: intermittent encryption of every second, fourth or eighth 4 KB block), for non-compressed (top) and compressed (bottom) source files

Importance of the statistics

Figure 7 shows the permutation importance of the random forest inputs. Permuting entropy alone reduced the ROC-AUC by 0.008 to 0.021 and permuting χ^2 alone by 0.028 to 0.080, while permuting both together reduced it by 0.180, 0.230 and 0.250 for 512 B, 4 KB and 64 KB fragments. Serial correlation (0.011 to 0.025) and the mean byte value (0.015 to 0.026) contributed small amounts, the printable share at most 0.008 and the π error at most 0.002.

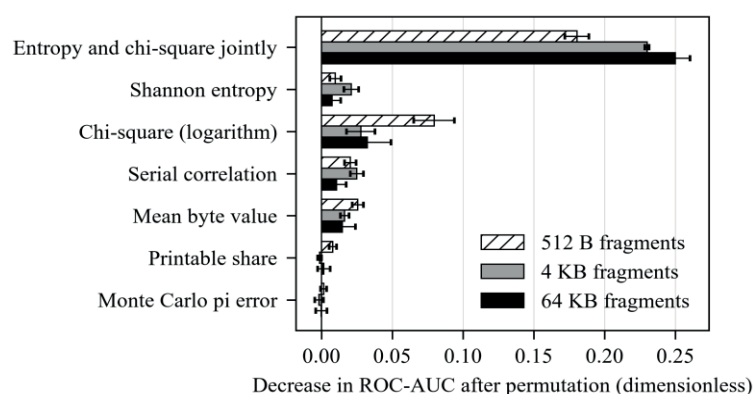


Figure 7. Permutation importance of the random forest inputs (mean and standard deviation over five folds)

Discussion

For RQ1, the separability of ciphertext by entropy depends on the compression family rather than on compression as such. Text and legacy office formats were separated without error at every fragment size. JPEG and GIF fragments were separated from 4 KB upwards, consistent with structure that survives their entropy coding, such as the byte stuffing after 0xFF in JPEG

scans. Deflate-based containers (ZIP, TAR.GZ, DOCX, PNG) and bzip2 archives were separated well only with large fragments, and LZMA-based archives (TAR.XZ, 7Z) were not separated at all: their ROC-AUC stayed below 0.65 for every statistic and model, and the random forest still flagged 69% to 74% of their 64 KB fragments. This agrees with the high false positive rates on compressed files reported by Pont et al. [11] and with the finding of De Gaspari et al. [17] that byte-distribution statistics fail to separate encrypted from compressed fragments. A test set dominated by JPEG or legacy office files therefore yields a much higher apparent accuracy than one dominated by modern archives, which reconciles perfect results on document collections [9] with high false positive rates on compressed files.

For RQ2, the chi-square statistics and the other randomness statistics did not rank fragments better than entropy. This follows from equation (4): the entropy deficit is an exact multiple of the G statistic, and G and χ^2 are asymptotically equivalent [33], so near uniformity both carry the same information. The measured rank correlation was 0.973 to 0.9995, the ROC-AUC values differed by at most 0.005, and permuting either statistic alone hardly affected the random forest, while permuting both removed most of its skill. The Monte Carlo π error, the serial correlation, the mean byte value and the printable share were weaker than entropy for almost every format. The hypothesis that χ^2 is clearly better from 4 KB upwards is therefore not supported for separability. It holds for calibration: the null distribution of χ^2 does not depend on the fragment size, so one critical value maintained a recall of 0.991 at every size, whereas a fixed entropy threshold is too strict for small fragments and too loose for large ones. The tuned entropy thresholds (7.539, 7.947 and 7.997 bits per byte) followed the first-order approximation at 4 KB and 64 KB. At 512 B the mean entropy of ciphertext lay 0.050 bits per byte below it, because the second-order expansion that equates G with χ^2 underestimates the deficit when the expected count per byte value is only two; the mean G of 290.7 against a mean χ^2 of 255.0 shows the size of the neglected terms, which agrees with the known behaviour of entropy estimators on short byte samples [34], so thresholds for small blocks should be calibrated on data rather than from equation (4). Gains reported for combinations of entropy and chi-square with NIST-style tests [16] are therefore most likely to come from tests that use information beyond the byte histogram, such as runs or bit-level patterns; a direct comparison with such tests was not part of this study.

The effect of fragment size (RQ3) follows from equation (2): for random bytes the expected χ^2 is 255 with a standard deviation of $\sqrt{510} \approx 22.6$ at any size (22.4 was measured on the ciphertext fragments), whereas a fixed systematic deviation of the byte histogram from uniformity produces an excess that grows in proportion to N . The median χ^2 of TAR.GZ fragments grew from 268 at 512 B to 292 at 4 KB and 429 at 64 KB, while for TAR.XZ it stayed at 258, 262 and 264. Formats whose coder leaves a small but systematic bias therefore become separable once fragments are large enough, and formats whose coder output is practically uniform at the byte level, such as the range coder of LZMA, do not. At 512 B all byte statistics are dominated by sampling noise, which confirms the degradation on small blocks predicted by the hypothesis and agrees with the loss of accuracy on small fragments reported for learned classifiers [17, 20].

For RQ4, Base64 encoding of ciphertext defeated every detector, because the encoded text uses 64 symbols and its entropy cannot exceed 6 bits per byte, as described by Lee and Lee [26]; decoding before measurement undoes this evasion [27], which is why format-preserving transformations are the harder case. Intermittent encryption interacts with the block size of the detector: small aligned fragments contain pure ciphertext and are detected, but on compressed files they cannot be told apart from untouched compressed blocks, while large fragments average encrypted and intact blocks and, for non-compressed files, were flagged in only 0.9% to 2.4% of cases. This matches the stated purpose of intermittent modes [24] and the difficulty of detecting partial encryption from storage-level statistics reported by Mahboubi et al. [25]. A file-level rule over 4 KB fragments recovered most intermittently encrypted files but also flagged most unencrypted compressed files.

Several recommendations for endpoint detection and forensic triage follow. First, thresholds must be tied to the block size: with the 7.9 rule the entropy of ciphertext in 512 B sectors never reaches the threshold, while the entropy of nearly every archive in 64 KB windows exceeds it; the χ^2 critical value, or an entropy threshold calibrated for the block size, behaves consistently. Second, a byte-level test can establish that data are not plaintext but not that an archive, image or office document is encrypted; that needs other evidence, such as a check that the container structure is valid and decompresses, a comparison with an earlier version of the file as in similarity-based detectors [6], or the behaviour of the writing process [7, 3]. Third, detectors should recognise text encodings such as Base64 and hexadecimal and test the decoded bytes. Fourth, evaluations should split data by file, report false positive rates per format and include LZMA-based archives.

The study has limitations. The seven container formats were generated from Govdocs1 content with default settings, and Govdocs1 predates 2010, so modern formats such as MP4 video, HEIC images, PDF files with object streams and DOCX files with embedded media are missing; a newer data set such as NapierOne [14] would allow the analysis to be repeated on current formats. Only symmetric ciphers applied by a script were used, not real ransomware, so artefacts such as appended key blobs or renamed files were not modelled. All runs used one machine and one seed, and the 64 KB statistics for HTML, XML and GIF rest on few files. The classes were balanced, although encrypted blocks are rare in normal operation, and precision falls quickly with their share: with the measured recall and FPR, a share of 10% encrypted fragments gives a precision of 0.132 to 0.359 across detectors and sizes, and a share of 1% gives 0.014 to 0.048, roughly 20 to 70 false alarms per correctly flagged fragment (Table S2); these values also depend on the format mix of the unencrypted data. Finally, only byte statistics and two classical models were studied; deep fragment classifiers may extract more structure from Deflate and bzip2 streams [22], although the LZMA results suggest a fundamental limit for any method that sees only the bytes of one fragment. Future work should test decoding-aware detectors, evaluate detectors on live ransomware in a sandbox and combine byte statistics with structural validation of containers.

Conclusion

Entropy thresholds remain a common first indicator of ransomware activity, and this study delimits the conditions under which that indicator is informative. On fragments of 595 files in 17 formats, byte-level statistics separated ciphertext from text and legacy office data without error, from JPEG, GIF and PDF data from 4 KB upwards, and from Deflate and bzip2 streams only with 64 KB fragments; for LZMA-based archives no statistic and no model performed much better than chance. The chi-square statistic ranks fragments in practically the same order as entropy, so replacing one with the other changes only the calibration, not what can be detected: its critical value is valid at every block size, while a fixed entropy threshold such as 7.9 bits per byte either misses ciphertext in small blocks or flags most compressed data in large ones. Base64 encoding removed the signal completely, and under intermittent encryption the detectors either missed mixed blocks or flagged compressed files. Byte statistics can show that data are not plaintext, but not that data are encrypted. Detectors for endpoints, storage systems and forensic triage should use size-aware thresholds, decode common text encodings before testing and confirm high-entropy findings with format validation, version history or process behaviour, and their evaluations should report false positive rates per format under file-level data splits.

References

- [1] Oz, H., Aris, A., Levi, A., & Uluagac, A. S. (2022). A survey on ransomware: Evolution, taxonomy, and defense solutions. *ACM Computing Surveys*, 54(11s), Article 238. <https://doi.org/10.1145/3514229>
- [2] McIntosh, T., Susnjak, T., Liu, T., Xu, D., Watters, P., Liu, D., Hao, Y., Ng, A., & Halgamuge, M. (2024). Ransomware reloaded: Re-examining its trend, research and mitigation in the era of data exfiltration. *ACM Computing Surveys*, 57(1), Article 18. <https://doi.org/10.1145/3691340>

- [3] Begovic, K., Al-Ali, A., & Malluhi, Q. (2023). Cryptographic ransomware encryption detection: Survey. *Computers & Security*, 132, Article 103349. <https://doi.org/10.1016/j.cose.2023.103349>
- [4] Cen, M., Jiang, F., Qin, X., Jiang, Q., & Doss, R. (2024). Ransomware early detection: A survey. *Computer Networks*, 239, Article 110138. <https://doi.org/10.1016/j.comnet.2023.110138>
- [5] McIntosh, T., Kayes, A. S. M., Chen, Y.-P. P., Ng, A., & Watters, P. (2021). Ransomware mitigation in the modern era: A comprehensive review, research challenges, and future directions. *ACM Computing Surveys*, 54(9), Article 197. <https://doi.org/10.1145/3479393>
- [6] Scaife, N., Carter, H., Traynor, P., & Butler, K. R. B. (2016). CryptoLock (and drop it): Stopping ransomware attacks on user data. In *2016 IEEE 36th International Conference on Distributed Computing Systems (ICDCS)* (pp. 303-312). IEEE. <https://doi.org/10.1109/ICDCS.2016.46>
- [7] Continella, A., Guagnelli, A., Zingaro, G., De Pasquale, G., Barengi, A., Zanero, S., & Maggi, F. (2016). ShieldFS: A self-healing, ransomware-aware filesystem. In *Proceedings of the 32nd Annual Conference on Computer Security Applications (ACSAC '16)* (pp. 336-347). Association for Computing Machinery. <https://doi.org/10.1145/2991079.2991110>
- [8] Kharraz, A., Arshad, S., Mulliner, C., Robertson, W., & Kirda, E. (2016). UNVEIL: A large-scale, automated approach to detecting ransomware. In *Proceedings of the 25th USENIX Security Symposium* (pp. 757-772). USENIX Association. <https://www.usenix.org/conference/usenixsecurity16/technical-sessions/presentation/kharaz>
- [9] Lee, K., Lee, J., Lee, S.-Y., & Yim, K. (2023). Effective ransomware detection using entropy estimation of files for cloud services. *Sensors*, 23(6), Article 3023. <https://doi.org/10.3390/s23063023>
- [10] Skračić, K., Petrović, J., & Pale, P. (2023). Classification of low- and high-entropy file fragments using randomness measures and discrete Fourier transform coefficients. *Vietnam Journal of Computer Science*, 10(4), 433-462. <https://doi.org/10.1142/S2196888823500070>
- [11] Pont, J., Arief, B., & Hernandez-Castro, J. (2020). Why current statistical approaches to ransomware detection fail. In W. Susilo, R. H. Deng, F. Guo, Y. Li, & R. Intan (Eds.), *Information security: 23rd International Conference, ISC 2020, proceedings* (Lecture Notes in Computer Science, Vol. 12472, pp. 199-216). Springer. https://doi.org/10.1007/978-3-030-62974-8_12
- [12] Davies, S. R., Macfarlane, R., & Buchanan, W. J. (2021). Differential area analysis for ransomware attack detection within mixed file datasets. *Computers & Security*, 108, Article 102377. <https://doi.org/10.1016/j.cose.2021.102377>
- [13] Davies, S. R., Macfarlane, R., & Buchanan, W. J. (2022). Comparison of entropy calculation methods for ransomware encrypted file identification. *Entropy*, 24(10), Article 1503. <https://doi.org/10.3390/e24101503>
- [14] Davies, S. R., Macfarlane, R., & Buchanan, W. J. (2022). NapierOne: A modern mixed file data set alternative to Govdocs1. *Forensic Science International: Digital Investigation*, 40, Article 301330. <https://doi.org/10.1016/j.fsidi.2021.301330>
- [15] Casino, F., Choo, K.-K. R., & Patsakis, C. (2019). HEDGE: Efficient traffic classification of encrypted and compressed packets. *IEEE Transactions on Information Forensics and Security*, 14(11), 2916-2926. <https://doi.org/10.1109/TIFS.2019.2911156>
- [16] Casino, F., Hurley-Smith, D., Hernandez-Castro, J., & Patsakis, C. (2025). Not on my watch: Ransomware detection through classification of high-entropy file segments. *Journal of Cybersecurity*, 11(1), Article tyaf009. <https://doi.org/10.1093/cybsec/tyaf009>
- [17] De Gaspari, F., Hitaj, D., Pagnotta, G., De Carli, L., & Mancini, L. V. (2022).

Reliable detection of compressed and encrypted data. *Neural Computing and Applications*, 34(22), 20379-20393. <https://doi.org/10.1007/s00521-022-07586-7>

[18] Kurumathur, S. K., Hooker, A., Huang, W., Pataci, H., Vishwamitra, N., & Choo, K.-K. R. (2026). The classification of encrypted and compressed data containers: A systematic survey. *Information and Software Technology*, 198, Article 108236. <https://doi.org/10.1016/j.infsof.2026.108236>

[19] Mittal, G., Korus, P., & Memon, N. (2021). FiFTy: Large-scale file fragment type identification using convolutional neural networks. *IEEE Transactions on Information Forensics and Security*, 16, 28-41. <https://doi.org/10.1109/TIFS.2020.3004266>

[20] Skračić, K., Petrović, J., & Pale, P. (2023). ByteRCNN: Enhancing file fragment type identification with recurrent and convolutional neural networks. *IEEE Access*, 11, 138176-138187. <https://doi.org/10.1109/ACCESS.2023.3340441>

[21] Haque, M. E., & Tozal, M. E. (2022). Byte embeddings for file fragment classification. *Future Generation Computer Systems*, 127, 448-461. <https://doi.org/10.1016/j.future.2021.09.019>

[22] Zou, B., & Liu, H. (2026). Hierarchical deep learning for file fragment classification. *Electronics*, 15(7), Article 1507. <https://doi.org/10.3390/electronics15071507>

[23] Loman, M. (2021, August 27). LockFile ransomware's box of tricks: Intermittent encryption and evasion. *Sophos*. <https://www.sophos.com/en-us/blog/lockfile-ransomwares-box-of-tricks-intermittent-encryption-and-evasion>

[24] Milenkoski, A., & Walter, J. (2022, September 8). Crimeware trends: Ransomware developers turn to intermittent encryption to evade detection. *Sentinel Labs*. <https://www.sentinelone.com/labs/crimeware-trends-ransomware-developers-turn-to-intermittent-encryption-to-evade-detection/>

[25] Mahboubi, A., Aboutorab, H., Camtepe, S., Bui, H. T., Luong, K., Ansari, K., Wang, S., & Barry, B. (2025). Ransomware encryption detection: Adaptive file system analysis against evasive encryption tactics. In W. Susilo & J. Pieprzyk (Eds.), *Information security and privacy: 30th Australasian Conference, ACISP 2025, proceedings, part III* (Lecture Notes in Computer Science, Vol. 15660, pp. 399-414). Springer. https://doi.org/10.1007/978-981-96-9101-2_21

[26] Lee, J., & Lee, K. (2022). A method for neutralizing entropy measurement-based ransomware detection technologies using encoding algorithms. *Entropy*, 24(2), Article 239. <https://doi.org/10.3390/e24020239>

[27] Lee, J., Lee, S.-Y., Yim, K., & Lee, K. (2023). Neutralization method of ransomware detection technology using format preserving encryption. *Sensors*, 23(10), Article 4728. <https://doi.org/10.3390/s23104728>

[28] Bang, J., Kim, J. N., & Lee, S. (2024). Entropy sharing in ransomware: Bypassing entropy-based detection of cryptographic operations. *Sensors*, 24(5), Article 1446. <https://doi.org/10.3390/s24051446>

[29] Lee, J., Yun, J., & Lee, K. (2024). A study on countermeasures against neutralizing technology: Encoding algorithm-based ransomware detection methods using machine learning. *Electronics*, 13(6), Article 1030. <https://doi.org/10.3390/electronics13061030>

[30] De Gaspari, F., Hitaj, D., Pagnotta, G., De Carli, L., & Mancini, L. V. (2022). Evading behavioral classifiers: A comprehensive analysis on evading ransomware detection techniques. *Neural Computing and Applications*, 34(14), 12077-12096. <https://doi.org/10.1007/s00521-022-07096-6>

[31] Digital Corpora. (n.d.). *Govdocs1* [Data set]. Retrieved September 19, 2026, from <https://digitalcorpora.org/corpora/file-corpora/files/>

[32] Shannon, C. E. (1948). A mathematical theory of communication. *Bell System Technical Journal*, 27(3), 379-423. <https://doi.org/10.1002/j.1538-7305.1948.tb01338.x>

[33] Agresti, A. (2018). *An introduction to categorical data analysis* (3rd ed.). Wiley.

[34] Contreras Rodríguez, L., Madarro-Capó, E. J., Legón-Pérez, C. M., Rojas, O., & Sosa-Gómez, G. (2021). Selecting an effective entropy estimator for short sequences of bits and bytes with Metric Entropy. *Entropy*, 23(5), Article 561. <https://doi.org/10.3390/e23050561>

List of Supplementary Materials

Table S1 (file Supplementary_Materials.docx): false positive rate of every detector for each format with compressed content and each fragment size, with 95% bootstrap intervals over files.

Data S1: fragment features, out-of-fold and evasion predictions and all result tables (directory results of the archive at [ZENODO DOI]).

Code S1: scripts that download Govdocs1, build the corpus, encrypt it, compute the features, evaluate the detectors and draw the figures (directory src of the archive at [ZENODO DOI]).

Table S2 (file Supplementary_Materials.docx): precision implied by the cross-validated recall and false positive rate at shares of 50%, 10%, 1% and 0.1% encrypted fragments.

Table S3 (file Supplementary_Materials.docx): recall and false positive rate of the fixed entropy thresholds reported in the literature (7.58, 7.8, 7.9, 7.95 and 7.99 bits per byte).

No references are cited only in the Supplementary Materials.