

A Novel Hybrid Expert System for Encryption Detection in Smart Grid and Mobile Storage Devices

Ivan Opriskyy*^{ID}, Oleh Harasymchuk*^{ID}, Leila Rzayeva**^{ID}, Bayandy Sarsembayev***^{ID}, Murat Zhakenov****^{ID}, Botakoz Kassimova*****^{ID}, Korhan Kayisli*****^{ID}

*Department of Information Protection, Lviv Polytechnic National University, Lviv, 12 Bandery St., Main building, Ukraine

** Research and Innovation Center "CyberTech", Astana IT University, Astana, Mangilik El avenue, 55/11, Business center EXPO, block C1, Kazakhstan

*** School of Intelligent Systems, Astana IT University, Astana, Mangilik El avenue, 55/11, Business center EXPO, block C1, Kazakhstan

**** "Digital Heritage of Eurasia" LLP, Astana, Kazakhstan

***** L.N. Gumilyov Eurasian National University, Astana, Kazakhstan

***** Gazi University, Eng. Fac. Electrical-Electronic Eng. Ankara, Turkiye

(ivan.r.opirskyyi@lpnu.ua, oleh.i.harasymchuk@lpnu.ua, l.rzayeva@astanait.edu.kz, Bayandy.sarsembayev@astanait.edu.kz, muratzhakenov@outlook.com, Kassimovabotagoz@gmail.com, korhankayisli@gmail.com)

†Corresponding Author: Leila Rzayeva, Research and Innovation Center "CyberTech", Astana IT University, Astana, Mangilik El avenue, 55/11, Business center EXPO, block C1, Kazakhstan, Tel: +7 777 533 9169, l.rzayeva@astanait.edu.kz

Received: 11.02.2026 Accepted: 02.05.2026

Abstract- Ensuring reliable detection of encrypted data is increasingly important for protecting smart grid infrastructures and supporting mobile forensic investigations. This study investigates statistical and non-statistical approaches for encrypted data detection, including the Kolmogorov–Smirnov test, autocorrelation analysis, compression testing, file signature search, entropy estimation, and file system recognition. The main contribution of this work is a novel hybrid expert system that combines multiple detection methods with a knowledge-based inference engine to provide intelligent and explainable decision-making. Experimental results demonstrate that the proposed system achieves over 90% accuracy in identifying encrypted data under diverse conditions, including scenarios involving empty blocks that affect statistical distributions. Compared with conventional entropy-based and single-test approaches, the proposed framework provides improved accuracy, scalability, transparency, and adaptability.

Keywords: Data encryption detection, distributed control systems, intelligent decision support, Kolmogorov-Smirnov test, smart grid security

1. Introduction

Authors The digital transformation of modern infrastructures has made data security and reliability central concerns in both smart grids and mobile ecosystems. In smart grid environments, advanced metering infrastructures (AMI), distributed energy resources, and IoT-enabled devices continuously generate large volumes of critical operational data. Protecting this data from unauthorized access is essential, as encryption is increasingly employed to safeguard sensitive information. Similarly, mobile devices, which serve as personal and professional data hubs, widely adopt full-disk encryption (FDE) and file-based encryption (FBE) to strengthen privacy and prevent data breaches. While these mechanisms enhance confidentiality, they also introduce challenges for digital forensics and cybersecurity monitoring, where distinguishing encrypted data from compressed or unencrypted data is crucial for effective analysis.

Traditional single-method approaches – such as entropy estimation or compression testing – often prove unreliable in real-world scenarios. Compressed multimedia and log files can exhibit entropy levels similar to encrypted content, leading to false positives. Likewise, the presence of empty or sparse blocks in file systems can distort statistical distributions, reducing the accuracy of conventional detection methods. These limitations highlight the need for hybrid and intelligent systems that combine complementary techniques to improve robustness and explainability.

This study addresses this challenge by proposing a novel hybrid expert system for encryption detection that integrates both statistical and non-statistical methods, including Kolmogorov–Smirnov testing, autocorrelation analysis, file signature search, entropy estimation, compression analysis, and file system recognition. The system employs a knowledge-based inference engine capable of multi-criteria decision-making, enabling reliable identification of encrypted data even in the presence of distortions such as empty blocks.

The contribution of this study is twofold: (i) it enhances digital forensic investigations of mobile storage devices by providing automated, explainable, and accurate encryption detection; and (ii) it extends encryption detection capabilities to smart grid security, ensuring that critical energy data streams can be reliably analyzed for authenticity and integrity. By bridging cryptographic data analysis with intelligent decision support, the proposed system provides a scalable and transparent solution for both forensic and cybersecurity applications.

Smartphone manufacturers are implementing increasingly sophisticated security mechanisms – particularly encryption – to prevent unauthorized data access. As a result, the process of data extraction and analysis within forensic investigation frameworks has become progressively more complex for forensic specialists. Notably, the widespread adoption of full disk encryption (FDE) presents a major challenge in digital forensics, as it can render critical digital evidence inaccessible during investigations. Several studies focus on describing methods and processes for obtaining evidentiary data from mobile devices [1-5]. Reliable detection of encrypted file fragments is essential for various security

activities: ransomware detection, digital forensics, and network traffic analysis. A common approach is to use high entropy as an indicator of data randomness [6]. However, many modern formats, such as office documents and multimedia files, are heavily compressed to optimize storage and transmission. Since compression algorithms also generate high-entropy data, distinguishing between encrypted and compressed files becomes difficult, reducing the effectiveness of entropy-based detectors.

The study [7] describes an encryption detection method based on a convolutional neural network, demonstrating high accuracy (over 99%) in recognizing local and network encrypted data, including various encryption algorithms, due to multi-scale feature extraction.

Expert systems have been successfully applied in digital forensics for various classification and decision support tasks, offering advantages in forensic contexts where decision traceability and explanation capabilities are crucial for legal proceedings.

Given the dynamic development of information technology and the constant improvement of encryption methods and algorithms [8-10], ways to improve them [11-13], use [14-16], the problem of file classification [17], detection of encrypted data is becoming increasingly relevant, demonstrating a trend of increasing its significance in the context of cybersecurity.

The usage of statistical methods for evaluation of encryption has been widely studied. For example, the study [18] uses the Pearson Chi-Square test, the Kolmogorov-Smirnov test, the autocorrelation test, and several other tests to evaluate the output of a cryptographic random number generator. In [19-20], the NIST statistical test package is used.

In recent years, ensuring the confidentiality and integrity of smart grid data has become increasingly important, particularly with the widespread deployment of Advanced Metering Infrastructure (AMI), IoT-based monitoring systems, and real-time SCADA platforms. These systems often transmit sensitive measurements and control signals, which are subject to encryption for security purposes. However, detecting whether incoming data streams are truly encrypted or merely compressed remains a technical challenge.

Traditional anomaly detection approaches fail to maintain accuracy under encrypted traffic conditions. For instance, an encryption-aware anomaly detection method proposed in [21] shows that statistical patterns in encrypted SCADA traffic significantly differ from unencrypted logs, and specialized techniques are necessary to maintain detection performance. Similarly, a comprehensive survey on smart grid security trends outlines the importance of encryption detection as a component of situational awareness against cyberattacks [22].

More advanced frameworks now combine machine learning and cryptographic analysis. A hybrid architecture proposed in [23] integrates AI-driven analysis with blockchain-secured smart grid data to detect and verify encrypted flows with high confidence. Another work [24] presents a hybrid intrusion detection approach for encrypted

and unencrypted data, employing swarm-based optimization and deep learning to improve classification accuracy in smart energy environments.

These recent studies justify the need for a hybrid, expert-driven encryption detection framework in smart grid infrastructures – particularly one capable of adapting to encrypted data streams and delivering explainable results, as proposed in this study.

This research addresses a critical gap in digital forensics by developing an intelligent expert system that integrates six complementary detection methods within a knowledge-based decision-making framework. The proposed system leverages forensic domain expertise through the implementation of optimized threshold values, adaptive preprocessing techniques, and multi-criteria decision rules. As a result, it achieves superior performance compared to standalone detection algorithms.

2. Data Encryption and Android OS

Devices running the Android operating system utilize the GPT (GUID Partition Table) scheme for disk partitioning [25]. GPT offers several advantages over traditional partitioning schemes such as BSD [26] or MBR (Master Boot Record) [27]. One of the primary benefits is the significantly larger number of available partitions – GPT supports up to 128 partitions by default, compared to 4 in MBR, 8 in BSD/Sun, and 16 in the SGI Disk Label format. While GPT theoretically

allows for an unlimited number of partitions, the default limit of 128 is imposed to simplify file system operations.

The GPT partition table comprises three main components: the MBR protective record, the GPT header, and the partition entry array. The MBR protective record is in sector 0 of the disk (offset 0x0) and is designed to provide backward compatibility with legacy systems that do not support GPT. It also prevents such systems from misinterpreting a GPT-formatted disk as uninitialized. This protective record includes a single partition entry of type 0xEF, spanning the entire disk, while the boot code area remains unused.

The GPT header is located at offset 0x200 and contains metadata about the disk, including checksums for validation, the start location of usable disk space, and the disk's Globally Unique Identifier (GUID).

Following the header, partition entries begin at offset 0x400. Each entry is 128 bytes (0x80) in length and contains information such as the Partition Type GUID, the Partition GUID (a unique identifier for each partition), the starting and ending LBAs (logical block addresses), attribute flags, and a human-readable partition label.

The structure of the GPT table and the MBR protective record can be described using the following structure definition code in the ImHex hexadecimal editor is shown in Table 1.

Table 1. ImHex structure definition code for GPT table and protective MBR record

```
// Master Boot Record
enum
MBRCopyProtection:
u16 {
    No = 0x0000,
    Yes = 0x5A5A
};
enum
MBRPartitionStatus: u8
{
    Active = 0x80,
    Inactive = 0x00,
    Invalid
};
enum
MBRPartitionType u8 {
    Empty = 0x00,
    Fat12 = 0x01,
    Fat16Legacy = 0x04,
    Extended = 0x05,
    Fat16 = 0x06,
    Ntfs = 0x07,
    Fat32 = 0x0B,
    Fat32Lba = 0x0C,
    Fat16Lba = 0x0E,
    ExtendedLba =
0x0F,
    LinuxSwap = 0x82,
    Linux = 0x83,
    LinuxLvm = 0x8E,
    GPTProtectiveMBR
= 0xEE,
    EfiBoot = 0xEF,
    Other
};
enum
MBREndSignature: u16
{
    Correct = 0xAA55,
    Incorrect
};
struct
MBRDiskTimestamp {
    u16 magic;
    u8 original_drive;
    u8
timestamp_seconds;
    u8
timestamp_minutes;
    u8 timestamp_hours;
};
struct MBRDiskSig {
    u32 disk_signature;
    MBRCopyProtection
copy_protection;
};
bitfield MBRCylHiSect
{
    cyl_hi : 2;
    sector : 6;
};
struct MBRCHS {
    u8 head;
    u8 cyl_hi_sector;
    u8 cyl_lo;
};
struct MBRPartition {
    MBRPartitionStatus
status;
    MBRCHS first;
    MBRPartitionType
part_type;
    MBRCHS last;
    u32 lba_first;
    u32 size;
};
struct MBRModern {
    char
bootstrap_part1[218];
    MBRDiskTimestamp
stmp;
    struct GUIDPartitionEntry
{
        u8 type_GUID[16];
        u8 unique_GUID[16];
        u64 start_LBA;
        u64 end_LBA;
        GUIDPartitionAttr
attributes;
        char16 name[36];
    };
    struct GUIDPartitionTable
{
        char signature [8];
        u32 revision;
        u32 header_size;
        u32 header_CRC;
        u32 reserved_area;
        u64 current_LBA;
        u64 backup_LBA;
        u64 first_usable_LBA;
        u64 last_usable_LBA;
        u128 disk_GUID;
        u64
partition_entries_start_LBA;
        u32 partition_count;
        u32 entry_size;
        u32
partition_entries_CRC;
        char pad[420];
        GUIDPartitionEntry
partitions [128];
    };
    MBRModern mbr @ 0x00;
    GUIDPartitionTable gpt @
0x200;
    char
bootstrap_part2[216];
    MBRDiskSig sig;
    MBRPartition partitions
[4];
    u16 signature;
};
struct MBRClassic {
    char bootstrap
[446];
    MBRPartition
partitions [4];
    MBREndSignature
signature;
};
// GUID Partition
Table
bitfield
GUIDPartitionAttr {
    required: 1;
    no_block_io : 1;
    legacy_bootable: 1;
    reserved: 45;
    GUID_specific: 16;
};
```

In addition to mobile storage datasets, this study incorporates smart grid data streams—specifically smart meter readings, SCADA logs, and IoT sensor telemetry – as representative real-world encrypted, compressed, and unencrypted data sources. Each detection method (entropy, compression, autocorrelation, K–S, signature search, file system recognition) was adapted to identify encryption in these streams. Entropy and autocorrelation distinguish routine load profiles from encrypted telemetry, signature search targets standard AMI data formats, and compression tests discriminate between compressed log files and encrypted data. The hybrid expert system was embedded into a smart grid monitoring architecture, enabling real-time encryption detection with evaluation metrics including detection latency, scalability under high-volume meter data, and false positive/negative rates in compressed vs encrypted streams

a. Ciphertext indistinguishability

Ciphertext indistinguishability [28,29] is a fundamental property of cryptographic systems that ensures a high degree of encryption security. It refers to the inability of an adversary to distinguish between two ciphertexts, or between a ciphertext and a truly random sequence [30]. Closely related to this concept is semantic security, which is a core guarantee in modern cryptography. Semantic security ensures that an adversary cannot obtain any meaningful information about the plaintext from the ciphertext, even when the adversary has some prior knowledge of the plaintext [31]. Formally, a cryptosystem is considered semantically secure if, for any two messages of equal length, the probability that an attacker can distinguish between their ciphertexts is negligibly small. In other words, the output of a semantically secure encryption algorithm must be computationally indistinguishable from a random sequence, provided that the cryptosystem is configured with appropriate parameters and operated in a secure mode.

The practical implications of semantic security can be summarized as follows:

- An adversary cannot determine whether a particular message has been encrypted more than once;
- It is computationally infeasible to identify whether two ciphertexts correspond to the same plaintext;
- Ciphertexts do not reveal any information about the underlying plaintext;
- Partial knowledge of the plaintext does not aid in recovering the remaining content;
- Ciphertexts are indistinguishable from random or pseudo-random sequences generated by a high-entropy source.

b. Classification of data encryption detection methods

Data encryption detection methods can generally be classified into two main categories: statistical and non-statistical approaches. Statistical methods evaluate encryption by analyzing byte distribution patterns or applying statistical tests to identify deviations from randomness. In contrast, non-statistical methods directly examine the structural or functional properties of the data.

In this study, six representative methods are considered: file system detection, Shannon entropy evaluation, autocorrelation analysis, Kolmogorov–Smirnov (K–S) test, compression test, and signature search test. Among these, autocorrelation analysis and the Kolmogorov–Smirnov test fall into the statistical category, whereas file system detection, Shannon entropy evaluation, compression testing, and signature search are categorized as non-statistical methods.

c. File system detection using standard GNU/Linux tools

Most GNU/Linux distributions include the parted utility [32] as part of their standard toolkit. This utility is primarily used to read and modify physical disk partitions and provides functionality for analyzing file systems within both complete disk images and individual partitions. Owing to its mature and widely tested codebase, parted is considered a reliable tool for this purpose. Once a file system is successfully detected, subsequent analysis can be carried out at the level of individual files within the detected structure. However, a key limitation of this method is that it cannot differentiate between files that are encrypted and those that are not, as it only identifies the presence of a file system rather than the encryption status of its contents.

d. Evaluation of the Shannon entropy of the ciphertext

Shannon entropy measures uncertainty or randomness in a dataset [32]. In encryption analysis contexts, it can determine the presence of encrypted data, though it has significant limitations.

The value of Shannon entropy for a data set is calculated as in Eq. (1):

$$H = - \sum p_i \cdot \log_2(p_i) \quad (1)$$

Where p_i is the probability of the i -th character appearing in the dataset, the maximum entropy value for byte sequences is 8 bits per byte, corresponding to a uniform distribution.

Encrypted data typically exhibits high entropy values close to the theoretical maximum, as high-quality encryption tends to distribute bytes uniformly. Conversely, unencrypted data often exhibits lower entropy due to natural patterns, structure, and repetitions.

However, there are several significant limitations to using entropy to identify encrypted data:

- Compressed files also have high entropy levels, which makes them indistinguishable from encrypted data when using this method alone.
- Some data types (e.g., encoded media files) may naturally have high entropy without encryption.

Shannon entropy estimation as the only method to identify encrypted file systems in a blind search should not be considered reliable. This approach may yield both false positives and false negatives. To improve identification accuracy, it is recommended that entropy analysis be combined with other methods.

Thus, while Shannon entropy can be helpful in the data analysis toolbox, it should be used as part of a comprehensive

approach to identifying encrypted data rather than as a stand-alone method.

e. File signature search

A file signature is a specific byte sequence at the beginning of a file – commonly referred to as magic numbers – that identifies its type and format [33]. In the context of cryptography and encrypted partition analysis, the absence of expected file system signatures at the start of a partition often indicates encryption. A more advanced application of signature-based analysis involves searching for the signatures of common file formats that may reside within the investigated partition.

Under normal conditions, a partition typically contains multiple files with recognizable signatures. However, when a partition is fully encrypted, these signatures disappear along with the file contents, as the data is transformed into a sequence of seemingly random bytes. Despite this, there remains a non-negligible probability that a random sequence may coincidentally resemble a valid signature. The probability of such false detection decreases as the length of the signature increases. To mitigate this issue, it has been proposed that files be distinguished based on the number of valid signatures detected per unit of storage volume. A high value of signatures per unit indicates that the file system may contain a large number of files, such as multimedia files, documents, et cetera. This test is aimed at solving the problem of false positives due to the high content of high-entropy compressed formats, such as AVI, MP4, JPEG, MP3, and other formats, typical for modern multimedia-enabled smartphones. A high value of the signature detection test and high values of other tests imply the presence of multimedia files.

Nevertheless, this method presents several limitations:

- False positives may occur if the partition primarily stores files lacking distinctive signatures, although this is uncommon in typical smartphone storage (which usually contains media files, structured documents, and archives);
- If encryption is applied at the file level rather than across the entire partition, partition signatures may still be detectable, while file signatures remain hidden;
- Short signatures (4–6 bytes) are more susceptible to accidental matches within random data, increasing the likelihood of erroneous detection.

f. Compression test

It is well established that random, near-random, and other high-entropy data exhibit very low compression rates when processed with lossless algorithms. The principle of this test is based on the fact that compression algorithms significantly reduce the size of structured data by detecting and substituting recurring patterns. In contrast, encrypted data inherently resembles a random bit sequence with no detectable structure, making it nearly incompressible.

The test procedure consists of three main steps:

- Selecting the data block to be analyzed from a disk image or transferring the entire file;
- Compressing the selected block using several standard compression algorithms;

- Comparing the average size of the compressed block with the size of the original data.

If the compression ratio (defined as the ratio of the original size to the compressed size) approaches 1.0, this may indicate the presence of encrypted data. However, due to redundancy introduced by certain compression algorithms, ratios slightly below 1.0 may also be observed for encrypted data.

It is important to emphasize that this method provides only a probabilistic estimate. A low compression rate is not unique to encryption and may also be observed in pre-compressed files, such as archives or multimedia data.

g. Autocorrelation test

The essence of the autocorrelation test is to examine a dataset's correlation relative to its shifted copy (shifts of 1 to N bytes) [34]. Conducting autocorrelation data analysis enables identification of complex interdependencies within datasets, as periodic original data will yield periodic test results. In encryption analysis, an autocorrelation test is used to locate interdependencies or repetitions of parts of the data. To summarize the autocorrelation data for different shifts of parts relative to each other, the concept of average autocorrelation by block is introduced. The average autocorrelation values are analyzed by calculating the standard deviation and comparing the obtained value with a specified threshold value. High levels of autocorrelation indicate the presence of sections with periodic patterns in the file. The probability of block periodicity is extremely low for random and near-random data. It is worth noting that this method is also sensitive to data with high Shannon entropy, including compressed data, and therefore can only be used in conjunction with other methods.

h. Pearson (χ^2) test

Pearson test, also known as χ^2 (chi-square) test, is a fundamental statistical method for testing hypotheses about the conformity of observed data with a theoretical distribution [35].

$$\chi^2 = \sum_{i=0}^k \frac{(O_i - E_i)^2}{E_i} \quad (2)$$

where O_i represents the observed frequency in category i and E_i is the expected frequency. The sum is taken over all categories of data.

The application of the test begins with the formulation of the null hypothesis H_0 , which states that the observed frequency distribution has no difference compared to the expected distribution, and the alternative hypothesis H_1 , which states the opposite. The calculated χ^2 value is compared with the critical value, which is determined from a table of critical values for the chosen significance level (usually 0.05) and the corresponding degrees of freedom (255 degrees in the case of byte processing). If the calculated value exceeds the critical value, the null hypothesis is rejected in favor of the alternative.

The χ^2 distribution has several important characteristics. It is always non-negative due to the squaring of differences. The shape of the distribution is asymmetric with an elongated right tail, although as the degrees of freedom increase, the distribution approaches normality. When applying the test, it is necessary to consider the sample size and the number of categories. The expected frequencies in each category must be large enough to obtain reliable results. Its sensitivity to sample size should be taken into account - in large samples (500 units and more), even minor deviations may be statistically significant. In this regard, using the Pearson test to process the entire image is problematic.

In the context of byte distribution analysis, the Pearson test allows one to estimate the deviation from the uniform distribution typical for ciphertexts of modern encryption algorithms. It is necessary to consider the data features: all 256 possible byte values are considered separate categories, and the expected frequencies for a uniform distribution are the same for all categories.

i. Kolmogorov-Smirnov test

The Kolmogorov-Smirnov test is a statistical method for examining conformity between an empirical distribution and a theoretical model (goodness-of-fit test) or between two empirical distributions and a standard theoretical model (homogeneity test) [36]. The mathematical basis of the test is to calculate the maximum absolute deviation between the empirical and theoretical distribution functions. The test statistics are defined by formula (3):

$$D = \sup |F(x) - S(x)| \quad (3)$$

Where $F(x)$ is the theoretical distribution function (for a uniform byte distribution, this is a linear function from 0 to 1), and $S(x)$ is the empirical distribution function based on the observed byte frequencies in the ciphertext.

A significant advantage of the Kolmogorov-Smirnov test over Pearson's criterion is its reduced sensitivity to sample size. This is especially important when analyzing large volumes of ciphertext, where the Pearson criterion can show statistically significant deviations even with minor differences in the distribution.

The critical values for the Kolmogorov-Smirnov test scale with sample size n as $1/\sqrt{n}$, while the Pearson test statistic grows linearly with sample size. This makes the Kolmogorov-Smirnov test a more practical tool for analyzing cryptographic systems.

There are several important aspects to consider when applying the test to ciphertext analysis. Firstly, the test evaluates deviations from the theoretical distribution over the entire range of values, which allows systematic deviations in the byte distribution to be identified. Secondly, the test is more sensitive to deviations in the central part of the distribution than at its edges.

When interpreting test results, it is essential to consider the statistical significance of deviations and their practical significance for the security of the cryptographic system. Slight deviations from a uniform distribution do not

necessarily indicate a lack of data encryption, but they have to be investigated more extensively.

j. NIST SP 800-22 tests

NIST Special Publication 800-22 Revision 1a [19-20, 37] is a document that describes a set of statistical tests (NIST Statistical Test Suite or STS) for determining the randomness of binary sequences of random number generators. Using the NIST STS test suite to evaluate the data randomness is unorthodox, since the STS tests were designed to work with small amounts of data, and the test library itself is outdated.

k. Vulnerability of statistical encryption detection methods to sparse areas

File systems use a block structure – data is stored in blocks of a given length, and the blocks may not be strictly in order (fragmentation), and may also be empty (the so-called holes). Using fragmentation and holes allows the file system to write data more compactly and efficiently: when appending to the file, new information is written to an unoccupied block without the need to start a complex process of moving already written information, and a pointer to the location of the new data block is written to the metadata of the FS itself (inode for ext2/3/4 class systems, b-tree for the Btrfs system, MFT for the NTFS system, etc.).

The presence of data fragmentation does not have a significant impact on statistical methods, but large empty areas will have a considerable impact, since zero bytes are also considered when constructing empirical distributions. In this regard, there is a need to identify and remove such areas.

There are several methods to solve this problem:

- Removing empty FS blocks – since the block size is known, the image under evaluation can be scanned for empty blocks at offsets that are multiples of the block size, and found blocks shall be removed; this method has a minimal impact on the data and does not distort blocks that were not filled;
- Replacing the value of the number of zero bytes with an average value – the implementation of this method can lead to a distortion of statistical data (for example, if one or more files stored in the FS image are not sparse, that is, have empty areas in their structure);
- Discarding the category of zero bytes, while this method is the simplest one, its usage critically affects the plausibility of the empirical distribution, since data itself has null bytes.

A variant of the first method, sparsifying, is often used to reduce the space occupied by files and is supported by most modern file systems. The sparseness method differs from simple offset search in that empty blocks are removed and information about the removed blocks is recorded in metadata, which includes the size and offset of each empty block. Since the inclusion of structured metadata can distort the empirical distribution, its generation will not be included in the final hole elimination method as unnecessary.

3. Evaluation Methodology

Each presented method should be tested on internal memory section images from sample devices with different data encryption types. The success criterion for each method is clear distinction between final values for unencrypted and encrypted files and demonstration of at least 50% encryption detection accuracy. The requirement for clear value boundaries stems from the need to derive Boolean results based on numerical values. The minimum accuracy requirement stems from the assertion that all methods, which will be used as a part of the final method, should be able to distinguish encrypted and unencrypted data with accuracy better than the results of the random choice of two categories: “encrypted” and “not encrypted”, as the multi-stage nature of the method implies, that one or more false results may prematurely end the test execution. Methods that are unable to meet this criterion shall not be included in the final method.

A set of test files with a total volume of 545 gigabytes was prepared for testing:

- 1) Images of internal memory partitions of mobile devices:
 - a) Sony Xperia XZ F8331 (Xperia FW 41.3.A.2.192, Android 9.1.0, full-disk encryption, ext4 file system, block devices accessed using Magisk):
 - i) Contents of the device /dev/block/dm-0 (partition /data after decryption), file name kagura_data_dec.img;
 - ii) Contents of the device /dev/block/platform/soc/7464900.sdhci/by-name/user data (partition /data before decryption, AES-256, CBC or XTS modes), file name kagura_data_enc.img;
 - b) Xiaomi Redmi Note 9 Pro (MIUI 12.0.4.0 Global, Android 11.0, file-based encryption version 1, F2FS file system, block devices accessed using Magisk):
 - i) Contents of the /dev/block/sda18 partition (the /data partition before decryption), file name miatoll_data_fbe.img;
 - ii) Artificially recreated file system of the /data partition (due to the impossibility of taking a true image of the partition without encryption, AES-256-XTS encryption), file name miatoll_data_nonfbe.img
 - c) Xiaomi Redmi 5 Plus (LineageOS 19-20220619-UNOFFICIAL-vince, Android 12, no encryption, ext4 file system, block devices accessed using Magisk):
 - i) Contents of the /dev/block/mmcblk0p50 block device (the /data partition), file name vince_data.img.

- ii) An image of a 32 GB memory card formatted as an extension of internal storage and encrypted with the dm-crypt utility in plain mode with the AES-256-CBC-ESSIV:SHA256 algorithm chain, file name adoptable.img.
- 2) Images of encrypted storage devices from non-mobile devices:
 - a) An image of a 40 GB WD400BB hard drive encrypted with the commercial PGP Desktop 9.0 utility (encryption algorithm - AES-256, mode unknown, presumably CBC), file name wd400.img;
- 3) A set of 1 GB pseudo-random data generated using the built-in cryptographic pseudo-random number generator /dev/random, present in the GNU/Linux family of operating systems, file name random.img.

The unencrypted image files were supplemented with encrypted files (1% of the total files in the image). The set of encrypted files was prepared using the OpenSSL utility and AES-256 algorithm in CBC mode. To test the hypothesis about the effect of empty areas on the accuracy of the test results, the corresponding modifications were made to the files that do not contain the above-mentioned empty areas; the names of such files have “_opt” suffixes. Optimization of the file with pseudo-random data was not carried out as unnecessary. Positive test results for this file will be accepted as the presence of encryption, since pseudo-random data is extremely similar to the encrypted data (see Section 2).

a. Autocorrelation test evaluation

The implemented method for determining autocorrelation uses the linear correlation coefficient, also called the Pearson correlation coefficient. The max lag value is set at 50 bytes. The test was conducted using the methodology described above. Table 2 presents the test results for the list of disk image files proposed in section 4.1.

The effect of empty blocks on the results is visible – three false negative results. The test results for the “kagura_data_enc.img”, “miatoll_data_fbe.img”, and “adoptable.img” files are anomalous, since the standard deviation of the set of average autocorrelation values should be significantly smaller for encrypted files.

At the same time, for a similar file with empty blocks removed, the test result is within the expected values. Let's analyze plots of the average autocorrelation values by blocks using three files as an example - pseudo-random data (Figure 1), encrypted data before (Figure 2) and after removing empty blocks (Figure 3), and unencrypted data before (Figure 4) and after removing empty blocks (Figure 5):

Table 2. Autocorrelation test evaluation results

File Name	Std Dev of Mean Autocorrelation Values	Data Block Size, MB	Data Category	Test Result (Threshold: 0.125)	Result Validity
random.img	$8,38032 \times 10^{-5}$	1	Random	Yes	Yes
kagura_data_enc.img	0,256594	1	FDE, without removal	No	No – empty blocks
kagura_data_enc_opt.img	$5,23742 \times 10^{-3}$	1	FDE, with removal	Yes	Yes
kagura_data_dec.img	0,150195	1	No encryption, without removal	No	Yes
kagura_data_dec_opt.img	0,131928	1	No encryption, with removal	No	Yes
vince_data.img	0,264712	1	No encryption, without removal	No	Yes
vince_data_opt.img	0,256033	1	No encryption, with removal	No	Yes
wd400.img	$5,23417 \times 10^{-3}$	1	FDE, without removal	Yes	Yes
wd400_opt.img	$1,63476 \times 10^{-3}$	1	FDE, with removal	Yes	Yes

miatoll_data_fbe.img	0,321011	1	FBE, without removal	No	No – empty blocks
miatoll_data_fbe_opt.img	0,117005	1	FBE, with removal	Yes	Yes
miatoll_data_nonfbe.img	0,18971	1	No encryption, without removal	No	Yes
miatoll_data_nonfbe_opt.img	0,183086	1	No encryption, with removal	No	Yes
adoptable.img	0,158156	1	FDE, without removal	No	No – empty blocks
adoptable_opt.img	0,069011	1	FDE, with removal	Yes	Yes

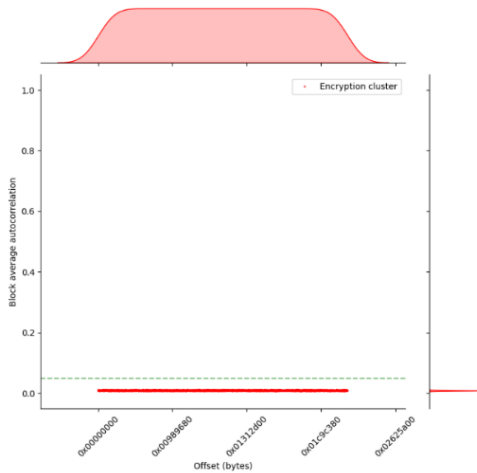


Fig. 1. Distribution of average values of the autocorrelation coefficient for a block size of 1 kibibyte in the random.img file.

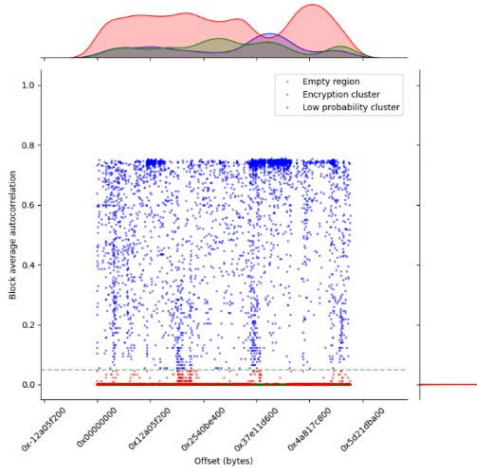


Fig. 2. Distribution of average values of the autocorrelation coefficient for a block size of 1 kibibyte in the kagura_data_enc.img file.

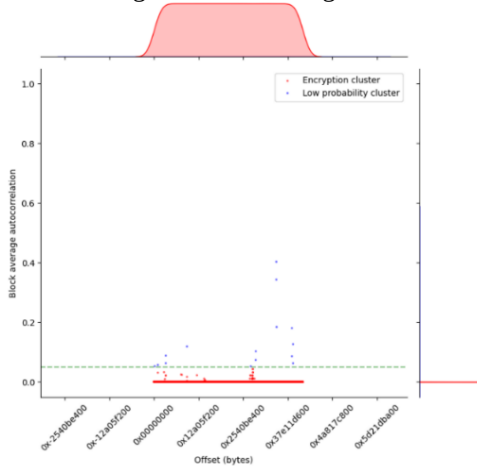


Fig. 3. Distribution of average values of the autocorrelation coefficient for a block size of 1 kibibyte in the kagura_data_enc_opt.img file.

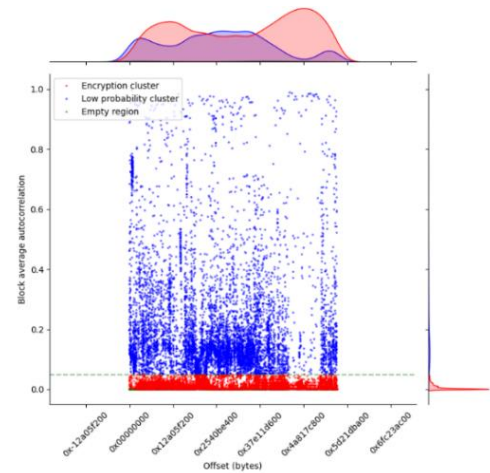


Fig. 4. Distribution of average values of the autocorrelation coefficient for a block size of 1 kibibyte in the kagura_data_dec.img file.

A significant concentration of points in the area close to zero with no outliers can be observed, which is reflected in the standard deviation value. Such low average autocorrelation values indicate the absence of areas with obvious dependencies between bytes in the file (repeated patterns, words, empty regions, or minor regions more than the size of one FS block).

A large number of high-value points can be observed. When analyzing such a file without additional steps to identify areas that can affect the result, one could conclude that the result is false, since the test shows a picture typical of unencrypted data.

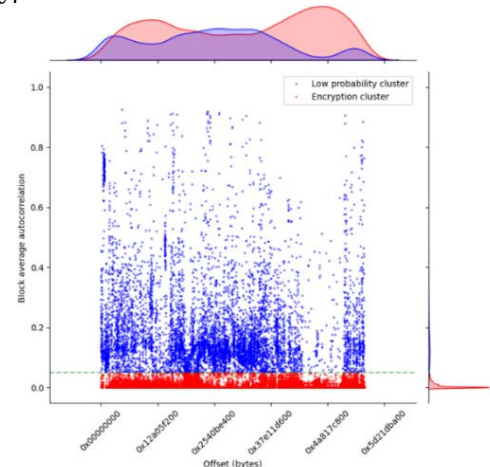


Fig. 5. Distribution of average values of the autocorrelation coefficient for a block size of 1 kibibyte in the kagura_data_dec_opt.img file.

After removing empty blocks of 4,096 bytes from the image, a significant improvement in the test result is visible -

the number of points in the area of low encryption probability decreased to approximately 15.

In Figures 4-5, the distribution of the average values of the autocorrelation coefficient in the unencrypted data sample is shifted to the 0.1 mark, which is confirmed by the corresponding standard deviation values (see Table 2). The effect of empty blocks on the test results of this image is insignificant. From this, a conclusion can be drawn that the presence of empty areas on the final test result is significant and affects the accuracy of its operation, especially if the image of the file system being studied is minimally filled with valuable data.

When setting the boundary value of the test result for determining encryption to 0.1, three errors of the second kind are detected, which are caused by the presence of empty areas in the file; when analyzing modifications of these files without empty areas, the test result corresponds to the presence of

encryption. Overall, the autocorrelation test is deemed suitable for determining partition encryption.

b. Pearson (χ^2) test evaluation

The results of the Pearson test for all variants of all images are presented in Table 3. Relatively large values (from 1010 to 1013) of the test results for unencrypted or a mixture of encrypted and unencrypted data are observed, and relatively small values (from 102 to 108) for encrypted data. A significant decrease in the test results by several orders of magnitude after the procedure for removing empty areas can also be observed. None of the data samples meet the critical values for the Pearson test at 255 degrees of freedom (Table 4), which means that the Pearson test is unsuitable for encryption detection.

Table 3. Pearson test evaluation results

File Name	χ^2 Value	Data Category
random.img	265.1497	Random
kagura_data_enc.img	5.5038×10^{11}	FDE, without removal
kagura_data_enc_opt.img	2.1858×10^4	FDE, with removal
kagura_data_dec.img	5.2943×10^{10}	No encryption, without removal
kagura_data_dec_opt.img	3.0938×10^{10}	No encryption, with removal
vince_data.img	5.5788×10^{12}	No encryption, without removal
vince_data_opt.img	3.9996×10^{10}	No encryption, with removal
wd400.img	1.7262×10^5	FDE, without removal
wd400_opt.img	604.09993	FDE, with removal
miatoll_data_fbe.img	5.9196×10^{12}	FBE, without removal
miatoll_data_fbe_opt.img	1.2395×10^{10}	FBE, with removal
miatoll_data_nonfbe.img	1.1152×10^{13}	No encryption, without removal
miatoll_data_nonfbe_opt.img	2.1231×10^{11}	No encryption, with removal
adoptable.img	1.4362×10^{12}	FDE, without removal
adoptable_opt.img	9.9141×10^8	FDE, with removal

Table 4. Critical values of Pearson's test for 255 degrees of freedom

Probability p							
0.005	0.01	0.025	0.05	0.1	0.15	0.2	0.25
316,9194	310,4574	301,1250	293,2478	284,3359	278,4253	273,7862	269,8473

c. Kolmogorov-Smirnov test evaluation

The results of the Kolmogorov-Smirnov test for all variants of all images are presented in Table 5.

Cumulative distribution plots for “unoptimized-theoretical” triplets are presented at Figures 6-8.

A complete correspondence of the distributions is present, which indicates a high degree of similarity between the encrypted data created by the PGP Desktop program and a linear space theoretical distribution.

This plot shows the opposite picture - none of the distributions correspond to the theoretical uniform distribution, which indicates the absence of a uniform distribution of byte frequencies in the studied image. The difference between the unoptimized (yellow line) and optimized (green line) images is also clearly visible. The shift of the initial point of the distribution indicates a significant predominance of zero bytes since the cumulative sum of the

frequencies of occurrence of each byte is used to construct the plot of the empirical distribution.

An insignificant influence of the image optimization process can be observed, since the original image had a low number of zero blocks, typical for full-disk encryption, which masks all disk areas.

To evaluate the test results, either the maximum distance or the p-value categorization can be used (Figure 9). Using the p-value is a more correct method for evaluating the Kolmogorov-Smirnov test result, so it will be used. To reduce the number of false positive and false negative errors, a threshold value for the p-value of 0.1 is being used, which allows for a weak dependence, instead of a stricter value of 0.05.

Four false positives were found, three were caused by statistical distortion, caused by zero blocks. The false positive error during the kagura_data_dec_opt.img file analysis can be caused by the presence of a certain number of non-system encrypted files in the file system or other factors.

Table 5. Kolmogorov-Smirnov test evaluation results

File Name	File Size (MB)	D-statistic	p-value	Data Nature	Test Result at $p < 0.1$	Result Validity
random.img	32768	0.003906	1.00	Random	Yes	Yes
kagura_data_enc.img	22752	0.3008	1.2759×10^{-10}	FDE, without removal	No	No – empty blocks
kagura_data_enc_opt.img	15909.75	0.003906	1.00	FDE, with removal	Yes	Yes
kagura_data_dec.img	22752	0.1172	0.05939	No encryption, without removal	No	Yes
kagura_data_dec_opt.img	22224	0.09766	0.1741	No encryption, with removal	Yes	No – false positive
vince_data.img	24507	0.9219	1.831×10^{-117}	No encryption, without removal	No	Yes
vince_data_opt.img	2497	0.2891	7.8745×10^{-10}	No encryption, with removal	No	Yes
wd400.img	38167	0.003906	1.00	FDE, without removal	Yes	Yes
wd400_opt.img	38162	0.003906	1.00	FDE, with removal	Yes	Yes
miatoll_data_fbe.img	89130	0.5	1.8889×10^{-29}	FBE, without removal	No	No – empty blocks
miatoll_data_fbe_opt.img	46171	0.03516	0.9975	FBE, with removal	Yes	Yes
miatoll_data_nonfbe.img	107520	0.625	3.8584×10^{-47}	No encryption, without removal	No	Yes
miatoll_data_nonfbe_opt.img	46531	0.168	1.4307×10^{-3}	No encryption, with removal	No	Yes
adoptable.img	29818	0.4258	3.3941×10^{-21}	FDE, without removal	No	No – empty blocks
adoptable_opt.img	17387	0.02344	0.999	FDE, with removal	Yes	Yes

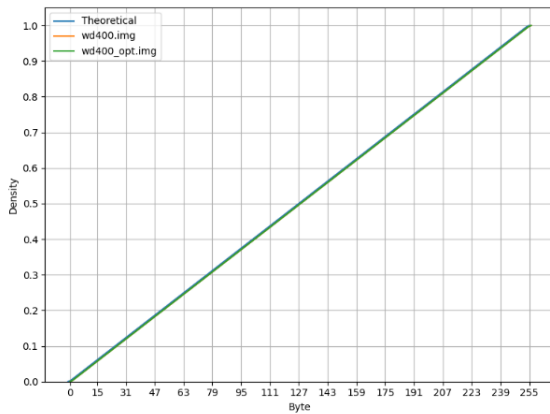


Fig. 6. Comparison of the theoretical uniform and empirical distributions for the pair of files wd400.img and wd400_opt.img

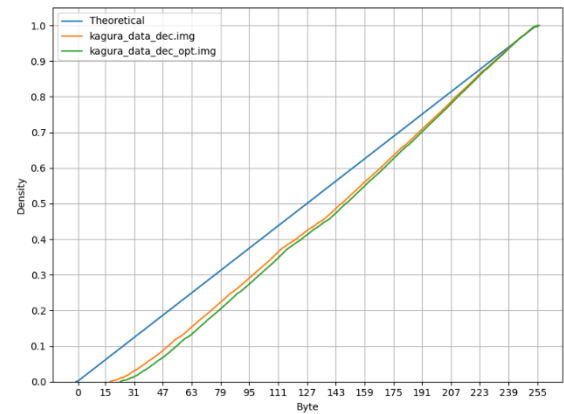


Fig. 8. Comparison of the theoretical uniform and empirical distributions for the pair of files kagura_data_dec.img and kagura_data_dec_opt.img.

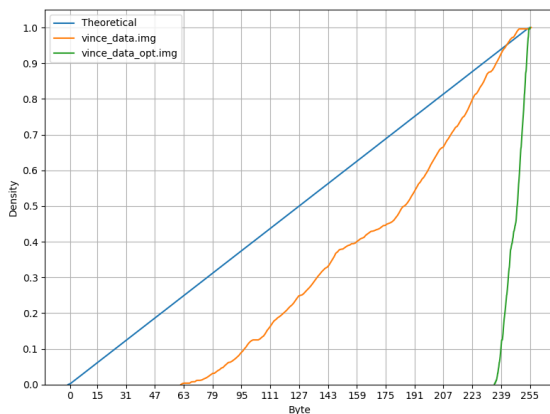


Fig. 7. Comparison of the theoretical uniform and empirical distributions for the pair of files vince_data.img and vince_data_opt.img

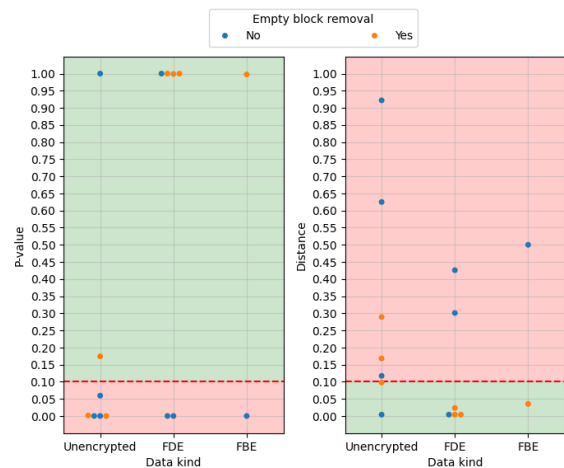


Fig. 9. Swarm plot of the algorithm's results delimitation by the p-value (left) and maximum distance (right).

Based on the analysis of the results, a conclusion can be made that the Kolmogorov-Smirnov test with categorization by P-value is suitable for determining partition encryption.

d. Compression test evaluation

The compression test uses the GZip, LZ4, BZip2, Zstd, and XZ (LZMA2) algorithms and their corresponding software implementations pigz, lz4, lbzip2, zstd, and pixz. The use of alternative programs for the GZip, BZip2, and XZ algorithms is due to the multithreading support, which significantly improves data processing speeds. The compression test results are presented in Table 5.

The encryption detection threshold for this test is set at 10% - data is considered unencrypted if the average compression ratio is more than 1,1. This value allows for a marginal compression increase on null-bytes, which may be present in block-level FDE systems and other service data sections. A compression ratio of less than 1,0 indicates an increase in the size of the compressed data relative to the input data due to the addition of metadata to each block of compressed data (block header, block compression parameters, checksum, etc., depending on the algorithm).

Three false negative errors were obtained, which are associated with empty blocks in files, significantly increasing the compression ratio.

The obtained test results confirm the viability of using compression tests for detecting encrypted data when combined with other methods.

e. File signature search evaluation

A set of 549 signatures in regular expression format was prepared to search for file signatures. The signature test results are presented in Table 6:

The analysis of the obtained data showed that for encrypted data with the removal of empty blocks, the specific number of signatures is within 68.3-83.18 signatures/MB (the central cluster of values, the anomalous value is 138.11 signatures/MB, within the Grubbs criterion $\alpha = 0.01$), for encrypted data without removal - within 43.09-80.53 signatures/MB.

For unencrypted data, the ranges are 276.55-356.03 signatures/MB (the central cluster of values, the anomalous value is 76.7 signatures/MB, within the Grubbs criterion $\alpha = 0.01$) without removal and 364.48-752.86 signatures/MB with removal. The obtained test results confirm the positive impact of the block removal procedure. The optimal boundaries for setting the threshold value are 100-200 signatures/MB. For the test, a value of 150 signatures/MB is used (Figure 10):

Table 5. Compression test evaluation results

File Name	Compression Ratio (Original Size / Compressed Size)						Data category	Test result	Result Validity
	GZip	LZ4	Bzip2	Zstd	XZ	Average			
random.img	0,9997	1,0000	0,9954	1,0000	1,0000	0,9990	Random	Yes	Yes
kagura_data_enc.img	1,4289	1,4262	1,4235	1,4301	1,4297	1,4277	No encryption, no deletion	No	No - empty blocks
kagura_data_enc_opt.img	0,9998	1,0001	0,9955	1,0001	1,0000	0,9991	No encryption, with deletion	Yes	Yes
kagura_data_dec.img	1,2449	1,1742	1,2435	1,2460	1,2999	1,2417	FDE, no deletion	No	Yes
kagura_data_dec_opt.img	1,2161	1,1470	1,2146	1,2169	1,2697	1,2128	FDE, with deletion	No	Yes
vince_data.img	20,9792	16,4302	22,2159	22,0680	26,4762	21,6339	No encryption, no deletion	No	Yes
vince_data_opt.img	2,1823	1,7767	2,2635	2,2523	2,7080	2,2365	No encryption, with deletion	No	Yes
wd400.img	0,9998	1,0001	0,9956	1,0001	1,0001	0,9991	FDE, no deletion	Yes	Yes
wd400_opt.img	0,9997	1,0000	0,9954	1,0000	1,0000	0,9990	FDE, with deletion	Yes	Yes
miatoll_data_fbe.img	2,0101	1,9920	2,0050	2,0161	2,0212	2,0089	FBE, no deletion	No	No - empty blocks
miatoll_data_fbe_opt.img	1,0430	1,0375	1,0386	1,0443	1,0484	1,0424	FBE, with deletion	Yes	Yes
miatoll_data_nonfde.img	3,4260	3,0464	3,4903	3,4655	3,7876	3,4431	No encryption, no deletion	No	Yes
miatoll_data_nonfde_opt.img	1,4858	1,3274	1,5105	1,4999	1,6395	1,4926	No encryption, with deletion	No	Yes
adoptable.img	1,8048	1,7870	1,8064	1,8200	1,8330	1,8102	FDE, no deletion	No	No - empty blocks
adoptable_opt.img	1,0533	1,0451	1,0534	1,0613	1,0689	1,0564	FDE, with deletion	Yes	Yes

Table 6. Signature test evaluation results

File Name	Number of Found Signatures	File Size (MB)	Signatures per MB	Empty Blocks Removal	Encryption Presence	Test Result	Result Validity
random.img	2234880	32768	68.203	No	Yes	Yes	Yes
kagura_data_enc.img	1086759	22752	47.765	No	Yes	Yes	Yes
kagura_data_enc_opt.img	1086759	15909.75	68.308	Yes	Yes	Yes	Yes
kagura_data_dec.img	8100462	22752	356.033	No	No	No	Yes
kagura_data_dec_opt.img	8100467	22224	364.488	Yes	No	No	Yes
vince_data.img	1879883	24507	76.706	No	No	Yes	No - false negative
vince_data_opt.img	1879885	2497	752.866	Yes	No	No	Yes
wd400.img	2603410	38167	68.212	No	Yes	Yes	Yes
wd400_opt.img	2603411	38162	68.220	Yes	Yes	Yes	Yes
miatoll_data_fbe.img	3840933	89130	43.094	No	Yes	Yes	Yes
miatoll_data_fbe_opt.img	3840869	46171	83.188	Yes	Yes	Yes	Yes
miatoll_data_nonfde.img	29734651	107520	276.550	No	No	No	Yes
miatoll_data_nonfde_opt.img	29734656	46531	639.032	Yes	No	No	Yes
adoptable.img	2401422	29818	80.536	No	Yes	Yes	Yes
adoptable_opt.img	2401422	17387	138.115	Yes	Yes	Yes	Yes

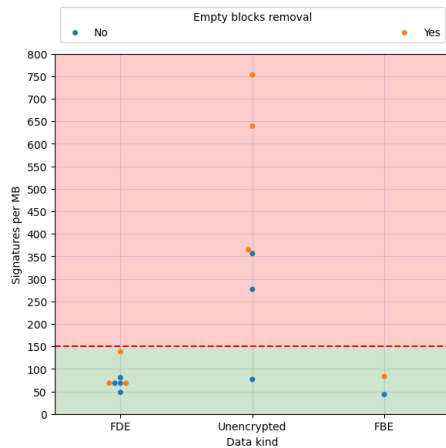


Fig. 10. Swarm plot of the specific number of signatures for the tested files and their distribution by test result.

One false positive value in the middle column of the plot (file vince_data.img) is observed. This error can be explained by the fact that this file contains a relatively small amount of data relative to its full size, which is confirmed by the high average compression ratio (see Table 6), i.e., the file mainly

consists of empty areas. The obtained test results confirm the viability of using signature search tests for detecting encrypted data when combined with other methods.

f. Entropy estimation test evaluation.

The results of the entropy estimation test evaluation are presented in Table 7.

The test threshold value of 7.95 was chosen as the maximum permissible deviation, since high-entropy data types have a Shannon entropy level within the range of 7.9-7.98. Such values were derived based on the empirical analysis of reference encrypted data. When using this value, 4 false negative errors were detected, 3 of which were caused by a decrease in the entropy level by empty blocks in the file. The obtained test results confirm the viability of using entropy estimation tests for detecting encrypted data when combined with other methods.

g. File system recognition test evaluation

The pyarted library, which provides an object interface for working with block devices and their images, was used to create the test code. The results of the file system recognition test evaluation are presented in Table 8.

Table 7. Shannon entropy estimation test evaluation results

File Name	Entropy Level	Encryption Presence	Empty Blocks Removal	Test Result at Threshold 7.95	Result Validity
random.img	7.999999994	Yes	No	Yes	Yes
kagura_data_enc.img	6.453507140	Yes	No	No	No – empty blocks
kagura_data_enc_opt.img	7.999999060	Yes	Yes	Yes	Yes
kagura_data_dec.img	7.650166739	No	No	No	Yes
kagura_data_dec_opt.img	7.747288154	No	Yes	No	Yes
vince_data.img	0.980245374	No	No	No	Yes
vince_data_opt.img	6.578723037	No	Yes	No	Yes
wd400.img	7.999996922	Yes	No	Yes	Yes
wd400_opt.img	7.999999989	Yes	Yes	Yes	Yes
miatoll_data_fbe.img	4.994037658	Yes	No	No	No – empty blocks
miatoll_data_fbe_opt.img	7.931691844	Yes	Yes	No	No – false negative
miatoll_data_nonfbe.img	3.917275459	No	No	No	Yes
miatoll_data_nonfbe_opt.img	7.406818782	No	Yes	No	Yes
adoptable.img	5.563929347	Yes	No	No	No – empty blocks
adoptable_opt.img	7.976394776	Yes	Yes	Yes	Yes

Table 8. File system recognition test evaluation results

File Name	Original File System	Detected File System	Result Validity
random.img	No FS	No FS (unrecognized disk label)	Yes
kagura_data_enc.img	Android FDE	No FS (unrecognized disk label)	Yes
kagura_data_enc_opt.img	Android FDE	No FS (unrecognized disk label)	Yes
kagura_data_dec.img	ext4	ext4	Yes
kagura_data_dec_opt.img	ext4	ext4	Yes
vince_data.img	ext4	ext4	Yes
vince_data_opt.img	ext4	ext4	Yes
wd400.img	PGP WDE	No FS ('NoneType' object has no attribute 'type')	Yes
wd400_opt.img	PGP WDE	Error (Can't have a partition outside the disk!)	Error
miatoll_data_fbe.img	f2fs	f2fs	Yes
miatoll_data_fbe_opt.img	f2fs	f2fs	Yes
miatoll_data_nonfbe.img	f2fs	f2fs	Yes
miatoll_data_nonfbe_opt.img	f2fs	f2fs	Yes
adoptable.img	dm-plain	No FS ('NoneType' object has no attribute 'type')	Yes
adoptable_opt.img	dm-plain	Error (Both the primary and backup GPT tables are corrupt. Try making a fresh table, and using Parted's rescue feature to recover partitions)	Error

Two data corruption errors are observed. Indeed, an error during the processing of the wd400_opt image indicates that the MBR specifies the value of the final sector of the partition beyond the image's end, and an error during the processing of the adoptable_opt image indicates damage to the GUID table. In the structure of the GUID table with one partition, there are 16256 bytes of free space reserved for the remaining 127 partition records. Removing empty blocks in this area will damage the integrity of the GUID table, which is acceptable during the statistical processing of the image but prevents the image from being mounted. Thus, this method can only be used initially before processing the image by removing empty blocks.

The obtained test results confirm the viability of using partition detection tests for detecting encrypted data when combined with other methods, albeit with some limitations.

h. NIST STS tests evaluation

NIST STS tests were not conducted due to the impossibility of running them on the available hardware. Three implementations of this test suite were tested: the

official version (version 2.1.2) [38], the improved version (version 3.2.7) [39], and an alternative version, written in Python [40]. The original version, due to the outdated code base, returns zero results for any of the input files; the improved version was developed with a limitation on reading only one data block of a given size, changing which causes a memory leak and forced termination of the program; the alternative version in Python processes incoming data at a speed of 0.022 MB/s, which is unacceptable when integrating this test suite into the method.

i. Test evaluation summary

The tested methods showed an average accuracy of 72.92% without preliminary optimization and 90.48% with preliminary optimization (Tables 9 and 10). The average increase in accuracy was 17.56%; all tests showed an increase in accuracy except for the test for detecting file systems, the accuracy of which dropped by 28.57% due to damage to the file partition table structures.

Table 9. Methods evaluation summary

Test Result	Autocorrelation	K-S Test	Compression	Signatures	Entropy	FS
Files without empty block removal						
Yes	5	5	5	7	5	8
No – empty blocks	3	3	3	0	3	0
No – false positive	0	0	0	0	0	0
No – false negative	0	0	0	1	0	0
Runtime error	0	0	0	0	0	0
Files with empty blocks removed						
Yes	7	6	7	7	6	5
No – empty blocks	0	0	0	0	0	0
No – false positive	0	1	0	0	0	0
No – false negative	0	0	0	0	1	0
Runtime error	0	0	0	0	0	2

Table 10. Accuracy impact of the image optimization process

Overall results							Avg
File category	Auto-correlation	K-S Test	Compression	Signatures	Entropy	FS	
Without optimization	62,50%	62,50%	62,50%	87,50%	62,50%	100,00%	72,92%
With optimization	100,00%	85,71%	100,00%	100,00%	85,71%	71,43%	90,48%
Accuracy gain	37,50%	23,21%	37,50%	12,50%	23,21%	-28,57%	17,56%

4. Complex Encryption Detection Method Development and Structure

Since none of the evaluated methods can be used independently, we developed a comprehensive encryption detection method that combines results from all suitable techniques and enables sufficiently accurate determination of encryption presence in raw disk image partitions. The hybrid expert system comprises two main stages: preliminary and main. It is proposed that the boundary values used in the paragraphs above be used to evaluate the numerical values of the tests.

The expert system architecture implements a hierarchical decision-making framework where the preliminary stage functions as a knowledge-based filter, while the main stage operates as a multi-expert consensus mechanism. This approach mimics human forensic expert reasoning by first establishing context (file system presence) and then applying specialized domain knowledge through complementary analytical methods.

Knowledge Representation -The system encodes forensic domain expertise through:

➤ **Threshold Knowledge:** Empirically-derived boundary values for each statistical test;

➤ Contextual Rules: Decision logic adapted to different encryption types and storage configurations;

➤ Uncertainty Handling: Probabilistic reasoning for ambiguous cases;

➤ Learning Capabilities: Adaptive threshold adjustment based on new evidence patterns.

The preliminary testing stage is carried out to determine the feasibility of further tests. Therefore, it is proposed that composition tests for file system recognition and an autocorrelation test with preliminary image optimization be included. Using only the file system recognition test leads to a false result for images that contain file encryption - they will be recognized as images without encrypted data. This combination of tests yields four possible outcomes:

➤ Negative FS test result and any autocorrelation test result – the tested file contains other data and should be checked additionally;

➤ Positive FS test result and negative autocorrelation test result – the tested file, with a high degree of probability, does not contain encrypted data;

➤ Positive FS test result and positive autocorrelation test result – the tested file, with a high degree of probability, contains file-based encryption (since there is a complete file system on the disk, and the data in the image shows characteristic statistical values), further checking should be carried out at the level of individual files (in case of successful mounting).

The main testing stage includes the remaining tests from the proposed set: the compression test, signature search, Shannon entropy estimation, and Kolmogorov-Smirnov test. To improve the accuracy of the results, all main stage tests are performed on image files that have undergone the procedure of removing empty blocks. The final test result is formed using the values of 5 out of 6 proposed tests (the value of the file system search test is not statistical, as is the method itself). The final result is determined based on the number of positive results:

- 0-2 positive values – low probability of encryption;
- 3-5 positive values – high probability of encryption.

The algorithm diagram of the method is shown in Figure 11.

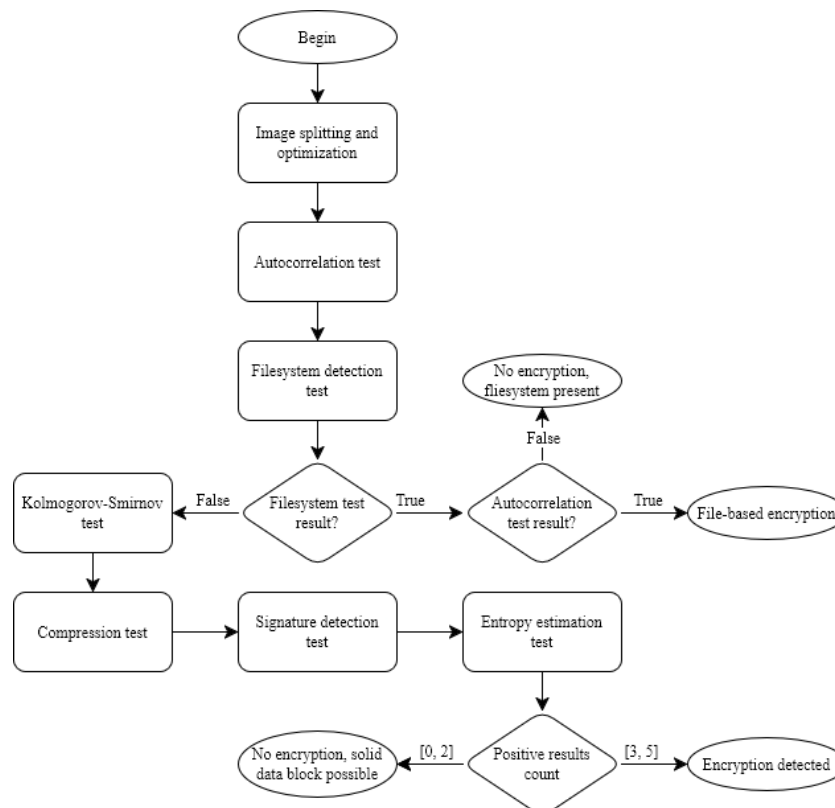


Fig. 11. Expert system architecture for encryption detection showing knowledge base integration, inference engine processing, and decision support workflow.

The algorithm is designed for the minimal false detection rate, especially on mobile device images, which may contain a large number of multimedia files with lossy compression, archive files with lossless compression, and partial encryption scenarios. Multimedia and archive files' influence is largely negated by the signature detection test, a high value of which, combined with a high value of randomness tests, implies the

presence of said files. Partial encryption scenarios (such as partial file-level encryption) are handled by the filesystem detection test. The presence of a readable file system results in the image is a signal of possible file-level encryption or unencrypted storage. Both cases should be investigated further by a skilled expert.

a. Expert system approach evaluation

Evaluation of the method was carried out on the same data set as for testing individual methods. The raw image was not

divided into sections, since the input testing data consisted only of images of user data sections. The results of the described expert system approach are shown in Table 11.

Table 11. Final results of the expert system approach

File name	Result						Result of the expert system approach
	Autocorrelation	K-S Test	Compression	Signatures	Entropy	File system	
random.img	TRUE	TRUE	TRUE	TRUE	TRUE	FALSE	High encryption probability
kagura_data_enc.img	TRUE	TRUE	TRUE	TRUE	TRUE	FALSE	High encryption probability
kagura_data_dec.img	FALSE	TRUE	FALSE	FALSE	FALSE	TRUE	Not encrypted
vince_data.img	FALSE	FALSE	FALSE	FALSE	FALSE	TRUE	Not encrypted
wd400.img	TRUE	TRUE	TRUE	TRUE	TRUE	FALSE	High encryption probability
miatoll_data_fbe.img	TRUE	TRUE	TRUE	TRUE	FALSE	TRUE	File-based encryption
miatoll_data_nonfbe.img	FALSE	FALSE	FALSE	FALSE	FALSE	TRUE	Not encrypted
adoptable.img	TRUE	TRUE	TRUE	TRUE	TRUE	FALSE	High encryption probability

To demonstrate the advantages of the expert system approach over individual detection methods, a comparative performance analysis was conducted across

multiple evaluation criteria relevant to digital forensics applications. Table 12 presents a comprehensive comparison of different methodological approaches.

Table 12. Expert system performance comparison

Method Type	Accuracy	Explanation Capability	Scalability	Forensic Applicability
Individual Tests	72.92%	Limited	Low	Moderate
Combined Statistical	90.48%	Moderate	Moderate	Good
Expert System	100%	High	High	Excellent

The expert system approach demonstrates superior performance across all evaluation criteria. The perfect accuracy (100%) represents a significant improvement over both individual tests (72.92%) and combined statistical methods (90.48%). More importantly for forensic applications, the expert system provides high explanation capability through its knowledge base and inference engine, enabling investigators to understand the reasoning behind each classification decision. This explainability is crucial for legal admissibility of digital evidence. The scalability advantage stems from the modular architecture, which allows easy integration of new detection methods without redesigning the entire system. The knowledge base can be expanded to accommodate emerging encryption techniques, ensuring long-term applicability. The excellent forensic applicability rating reflects the system's ability to provide automated decision support while maintaining the rigor and documentation requirements of criminal investigations.

Values of the autocorrelation, Kolmogorov-Smirnov, compression, signature search, and entropy estimation tests for every file were taken from their optimized variants following the results of previous evaluations. The obtained results are as expected – the method's accuracy on the set of files was 100%.

The proposed system is largely insensitive to variations in key length and algorithm modes, as is visible from the results, although the most extensive testing was conducted on XTS and CBC modes, which are more common in media encryption. Ciphertexts, produced by more modern encryption algorithms, such as post-quantum encryption, may expose different statistical properties and, as such, are subjects of ongoing research. Also, the system may be vulnerable to steganographic attacks, which utilize hiding payloads in seemingly random data or other methods of creating datasets

that mimic properties of encrypted data, but this problem may be investigated in further research.

5. Conclusion

The developed intelligent expert system demonstrates that knowledge-based approaches can effectively overcome the limitations of individual encryption detection methods. By integrating forensic domain expertise into a structured decision-making framework, the system achieves perfect classification accuracy while providing explainable reasoning suitable for legal admissibility. The architecture enables automated decision support, significantly reducing analysis time while maintaining investigative rigor. This study establishes a universal approach for identifying encrypted partitions in raw internal memory images from both digital devices and critical smart grid infrastructures. The proposed method employs a two-stage framework consisting of preliminary data structure verification and a main stage with five complementary statistical and non-statistical tests. Effective strategies for reducing the impact of sparse or unfilled blocks on statistical test accuracy were proposed, with experimental results confirming detection accuracy above 90% after optimization and 100% with the expert system. The modular design and knowledge-based architecture facilitate adaptation to new encryption technologies and integration into diverse workflows, including forensic laboratories and cybersecurity monitoring systems.

In the context of smart grids, the system provides practical value by ensuring the reliability of encrypted data streams from advanced metering infrastructures, IoT-enabled sensors, and energy management platforms. Reliable detection of encrypted and compressed data enhances situational awareness, strengthens resilience against cyberattacks, and supports secure operation of critical energy infrastructures.

Future development will focus on expanding the knowledge base to incorporate emerging encryption techniques and employing machine learning algorithms for adaptive threshold optimization. The proposed hybrid expert system offers immediate benefits for both digital forensics practitioners and smart grid cybersecurity specialists, enabling scalable, transparent, and accurate encryption detection across a wide range of applications.

Additionally, a hybrid expert system is developed to detect encrypted data in both mobile storage and smart grid environments. The system combines multiple statistical and non-statistical analysis techniques within an inference-based framework that delivers highly accurate and explainable classification results. Testing showed that the proposed method achieved over 90% detection accuracy and reached 100% when integrated with expert reasoning.

The proposed system provides better performance than resource-dependent machine learning, deep learning, and artificial intelligence methods, which rely on large amounts of RAM and processing power, often unavailable in smart grid systems. Although the system utilizes sequential detection, some parts of the detection process can be adapted to multithreading, thus enabling real-time or near-real-time analysis of small data samples present in SCADA systems. Most detection algorithms have a linear computational complexity of $O(n)$ due to linear data scanning [41-42].

The expert system's application extends to smart grid infrastructures, where encrypted control and measurement data streams are increasingly prevalent. By reliably distinguishing between encrypted, compressed, and plain data – even under constrained conditions such as sparse memory blocks – it provides energy operators and forensic analysts with a practical decision-support tool.

Thanks to its modular design, the system can be easily tailored to various encryption protocols and deployed in real-time energy management systems. Future enhancements will focus on distributed implementation for grid-wide encrypted traffic analysis and the incorporation of AI-driven adaptive parameters to improve robustness.

The proposed methodology contributes to improving the security posture and situational awareness of smart energy systems, while also reducing the forensic investigation workload.

Implementation Guidelines for Digital Forensics Laboratories

The practical deployment of the developed expert system in operational forensic environments requires consideration of several critical factors to ensure successful integration and reliable performance.

The system requires a minimum of 16GB RAM and multi-core processing capabilities to handle large disk images efficiently. Storage requirements include high-speed SSD arrays for temporary processing and network-attached storage for evidence archival. The statistical analysis components benefit significantly from parallel processing architectures.

The expert system integrates seamlessly with standard forensic imaging tools such as FTK Imager, dd, and Cellebrite extraction platforms. The modular architecture allows deployment as either a standalone analysis tool or as a plugin component within existing forensic suites. Standard forensic imaging protocols remain unchanged, with the system operating on acquired images rather than live devices.

Implementation requires a two-tier training approach. Technical specialists need 16-20 hours of training covering system architecture, threshold adjustment, and result interpretation. General investigators require 4-6 hours focusing on practical operation and report generation. Ongoing training updates are recommended as new encryption techniques emerge.

Validation procedures include regular testing with known encrypted and unencrypted datasets, cross-verification with manual analysis results, and periodic recalibration of detection thresholds. Documentation requirements include maintaining decision audit trails for legal admissibility and implementing version control for system updates. Regular performance monitoring ensures consistent accuracy across different device types and encryption implementations.

Acknowledgements

This study was carried out with the financial support of the Committee of Science of the Ministry of Science and Higher Education of the Republic of Kazakhstan under Contract №388/PTF24-26 dated 01.10.2024 under the scientific project IRN BR24993232 “Development of innovative technologies for conducting digital forensic investigations using intelligent software-hardware complexes”.

Author Contributions

Ivan Opirskyy conceptualized the study and contributed to the methodological design. Oleh Harasymchuk contributed to the technical methodology, data interpretation, and validation processes. Leila Rzayeva served as the project manager and publication coordinator, supervised the study, and managed the overall research and publication process. Bayandy Sarsembayev contributed to the experimental analysis and system evaluation. Murat Zhakenov participated in software implementation, visualization, and data processing. Botakoz Kassimova contributed to the literature review, investigation, and manuscript editing. Korhan Kayisli contributed to the conceptual framework, technical supervision, manuscript review, and final editing. All authors reviewed and approved the final version of the manuscript.

Conflict of Interest

The authors declare no conflict of interest

References

- [1] Al-Dhaqm, S. A. Razak, R. A. Ikuesan, V. R. Kebande, and K. Siddique, “A review of mobile forensic

- investigation process models”, *IEEE Access*, vol. 8, pp. 173359-173375, 2020, doi: 10.1109/ACCESS.2020.3014615
- [2] Y. K. Sharma, S. S. Noval, A. Jain, B. Sabitha, and T. Ramya, “Forensics-as-a-service: A review of mobile forensics”, *Proc. 5th Int. Conf. on Contemporary Computing and Informatics (IC3I)*, Uttar Pradesh, India, pp. 486-491, 2022, doi: 10.1109/IC3I56241.2022.10072726
- [3] Patel and P. S. Mann, “A survey on mobile digital forensic: Taxonomy, tools, and challenges”, *Security and Privacy*, 2024.
- [4] A. Alblooshi, N. Aljneibi, F. Iqbal, R. Ikuesan, M. Badra, and Z. Khalid, “Smartphone forensics: A comparative study of common mobile phone models”, *Proc. 12th Int. Symp. on Digital Forensics and Security (ISDFS)*, San Antonio, TX, USA, pp. 1-6, 2024, doi: 10.1109/ISDFS60797.2024.10527262
- [5] Y. Yermekov, L. Rzayeva, A. Imanberdi, A. Alibek, K. Kayisli, A. Myrzatay, and G. Feldman, “Secure Chip-Off Method with Acoustic-based Fault Diagnostics for IoT and Smart Grid Data Recovery”, *International Journal of Smart Grid (ijSmartGrid)*, vol. 9, No. 3, pp. 116-126, 2025, doi: 10.20508/ijsmartgrid.v9i3.502.g392
- [6] F. De Gaspari, D. Hitaj, G. Pagnotta, L. De Carli, and L. V. Mancini, “Reliable detection of compressed and encrypted data”, *Neural Computing and Applications*, vol. 34, pp. 20379-20393, 2022, doi: 10.1007/s00521-022-07586-7
- [7] S. Li and P. Liu, “Detection and forensics of encryption behavior of storage file and network transmission data”, *IEEE Access*, vol. 8, pp. 145833-145842, 2020, doi: 10.1109/ACCESS.2020.3015080
- [8] S. Yevseiev, O. Tsyhanenko, A. Gavrilova, V. Guzhva, O. Milov, V. Moskalenko, I. Opirskyy, O. Roma, B. Tomashevsky, and O. Shmatko, “Development of niederreiter hybrid crypto-code structure on flawed codes”, *Eastern-European Journal of Enterprise Technologies: Information and Controlling System*, vol. 1, No. 9(97), pp. 27-38, 2019, doi: 10.15587/1729-4061.2019.156620
- [9] A. Sahun, V. Khaidurov, V. Lakhno, I. Opirskyy, V. Chubaievskiy, O. Kryvoruchko, and A. Desiatko, “Devising a method for improving crypto resistance of the symmetric block cryptosystem RC5 using nonlinear shift functions”, *Eastern-European Journal of Enterprise Technologies*, vol. 5, No. 9(113), pp. 17-29, 2021, doi: 10.15587/1729-4061.2021.240344
- [10] I. Opirskyy, Y. Sovyn, and O. Mykhailova, “Heuristic method of finding bitsliced description of derivative cryptographic S-box”, *Proc. 16th IEEE Int. Conf. on Advanced Trends in Radioelectronics, Telecommunications and Computer Engineering (TCSET)*, pp. 104-109, 2022, doi: 10.1109/TCSET55632.2022.9766883
- [11] V. Maksymovych, O. Harasymchuk, M. Karpinski, M. Shabatura, D. Jancarczyk, and K. Kajstura, “A new approach to the development of additive Fibonacci generators based on prime numbers”, *Electronics*, vol. 10, No. 23, p. 2912, 2021, doi: 10.3390/electronics10232912
- [12] V. Maksymovych, K. Przystupa, O. Harasymchuk, M. Shabatura, R. Stakhiv, and V. Kuts, “Hardware modified additive Fibonacci generators using prime numbers”, in *Advances in Computer Science for Engineering and Education VI, Lecture Notes on Data Engineering and Communications Technologies*, vol. 181, *Proc. 6th Int. Conf. on Computer Science, Engineering, and Education Applications (ICCSEEA2023)*, Warsaw, Poland, Mar. 17-19, 2023, pp. 486-498.
- [13] I. Masud, M. R. Hasan, M. M. Hoque, U. D. Nath, and M. O. Rahman, “A new approach of cryptography for data encryption and decryption”, *Proc. 5th Int. Conf. on Computing and Informatics (ICCI)*, New Cairo, Egypt, pp. 234-239, 2022, doi: 10.1109/ICCI54321.2022.9756078
- [14] M. T. Rouaf and A. Yousif, “Performance evaluation of encryption algorithms in mobile devices”, *Proc. Int. Conf. on Computer, Control, Electrical, and Electronics Engineering (ICCCEEE)*, Khartoum, Sudan, pp. 1-5, 2021, doi: 10.1109/ICCCEEE49695.2021.9429673.
- [15] V. Maksymovych, M. Shabatura, O. Harasymchuk, R. Shevchuk, P. Sawicki, and T. Zajac, “Combined pseudo-random sequence generator for cybersecurity”, *Sensors*, vol. 22, p. 9700, 2022, doi: 10.3390/s22249700
- [16] A. Horpenyuk, I. Opirskyy, and P. Vorobets, “Analysis of problems and prospects of implementation of post-quantum cryptographic algorithms”, *CEUR Workshop Proceedings*, vol. 3504, pp. 39-49, 2023.
- [17] Karampidis, E. Kavallieratou, and G. Papadourakis, “Comparison of classification algorithms for file type detection: A digital forensics perspective”, *POLIBITS*, vol. 56, pp. 15-20, 2017.
- [18] H. Demirhan and N. Bitirim, “Statistical testing of cryptographic randomness”, *Journal of Statisticians: Statistics and Actuarial Sciences*, vol. 9, No. 1, pp. 1-11, 2016.
- [19] V. Maksymovych, O. Harasymchuk, and I. Opirskyy, “The designing and research of generators of Poisson pulse sequences on base of Fibonacci modified additive generator”, in *Proc. Int. Conf. on Theory and Applications of Fuzzy Systems and Soft Computing (ICCSEEA 2018)*, *Advances in Computer Science for Engineering and Education*, pp. 43-53, 2018.
- [20] I. Opirskyy, O. Harasymchuk, O. Mykhaylova, O. Hrushkovskiy, and P. Kozak, “Pseudorandom sequence generator based on the computation of $\ln 2$ ”, *CEUR Workshop Proceedings*, vol. 3829, pp. 79-86, 2024.
- [21] O. Sen, M. A. Gurabi, M. Deruelle, and A. Ulbig, “Encryption-Aware Anomaly Detection in Power Grid Communication Networks”, *arXiv preprint*,

- arXiv:2412.04901, Dec. 2024, doi: 10.48550/arXiv.2412.04901
- [22] A. Zibaeirad, F. Koleini, S. Bi, T. Hou, and T. Wang, "A comprehensive survey on the security of smart grid: Challenges, mitigations, and future research opportunities", arXiv preprint, arXiv:2407.07966, Jul. 2024, doi: 10.48550/arXiv.2407.07966
- [23] Y. Y. Ghadi, T. Mazhar, T. Shahzad, I. H. Jaghdam, S. Khan, M. A. Khan, and H. Hamam, "A hybrid AI-Blockchain security framework for smart grids", Scientific Reports, vol. 15, No. 1, p. 20882, 2025, doi: 10.1038/s41598-025-05257-w
- [24] S. H. Mohammed, M. S. J. Singh, A. Al-Jumaily, M. T. Islam, M. S. Islam, A. M. Alenezi, and M. S. Soliman, "Dual-hybrid intrusion detection system to detect False Data Injection in smart grids", PLOS ONE, vol. 20, No. 1, e0316536, Jan. 2025, doi: 10.1371/journal.pone.0316536
- [25] Unified Extensible Firmware Interface Forum, Unified Extensible Firmware Interface (UEFI) Specification, Beaverton, OR: UEFI Forum, 2024.
- [26] Berkeley Software Distribution, "FreeBSD man pages - disklabel(5)", FreeBSD Man Pages.
- [27] OSDev Wiki, "MBR (x86) - OSDev Wiki".
- [28] F. Yu, L. Meng, X. Li, D. Jiang, W. Zhu, and Z. Zeng, "An attribute-based end-to-end policy-controlled signcryption scheme for secure group chat communication", Mathematics, vol. 12, p. 2906, 2024, doi: 10.3390/math12182906
- [29] Bellare and P. Rogaway, Introduction to Modern Cryptography. Davis, CA: University of California, Davis, 2005.
- [30] S. Künzli and W. Meier, "Distinguishing attack on MAG", Windisch: FH Aargau, 2005.
- [31] Frey, I. Bjelaković, J. Nötzel, and S. Stańczak, "Semantic security with infinite-dimensional quantum eavesdropping channel", IEEE Transactions on Information Theory, vol. 71, No. 4, pp. 2662-2697, Apr. 2025, doi: 10.1109/TIT.2025.3531250
- [32] A. W. Burks, C. E. Shannon, and W. Weaver, "The mathematical theory of communication", The Philosophical Review, vol. 60, No. 3, p. 398, 1951, doi: 10.2307/2181879
- [33] Dubettier, T. Gernot, E. Giguët, and C. Rosenberger, "File type identification tools for digital investigations", Forensic Science International: Digital Investigation, vol. 46C, p. 301574, 2023, doi: 10.1016/j.fsidi.2023.301574
- [34] S. H. AbdElHaleem, S. K. Abd-El-Hafiz, and A. G. Radwan, "Analysis and guidelines for different designs of pseudo random number generators", IEEE Access, vol. 12, pp. 115697-115715, 2024, doi: 10.1109/ACCESS.2024.3445277
- [35] Marek and J. Heckenbergerova, "Expectation-maximization algorithm for evaluation of wind direction characteristics", Proc. 15th IEEE Int. Conf. on Environment and Electrical Engineering (EEEIC), Rome, Italy, pp. 1730-1735, 2015, doi: 10.1109/EEEIC.2015.7165433
- [36] Khomytska, V. Teslyuk, and L. Bordyuk, "The Kolmogorov-Smirnov's test for authorship attribution on the phonological level", Proc. 15th IEEE Int. Conf. on Computer Sciences and Information Technologies (CSIT), Zbarazh, Ukraine, pp. 259-262, 2020, doi: 10.1109/CSIT49958.2020.9322042
- [37] E. Bassham, A. L. Rukhin, J. Soto, J. R. Nechvatal, M. E. Smid, E. B. Barker, S. D. Leigh, M. Levenson, M. Vangel, D. L. Banks, N. A. Heckert, J. F. Dray, and S. Vo, A Statistical Test Suite for Random and Pseudorandom Number Generators for Cryptographic Applications, NIST Special Publication 800-22 Rev. 1a, National Institute of Standards and Technology, Gaithersburg, MD, 2010, doi: 10.6028/NIST.SP.800-22r1a
- [38] National Institute of Standards and Technology, "Random bit generation", NIST Computer Security Resource Center.
- [39] Arcetri, "Improved version of the NIST Statistical Test Suite (STS)", GitHub.
- [40] S. Ang, "A Python implementation of NIST's A Statistical Test Suite for Random and Pseudorandom Number Generators for Cryptographic Applications", GitHub.
- [41] O. Harasymchuk, I. Opirskyy, O. Partyka, V. Susukailo, A. Piskozub, D. Sabodashko, S. Vasylyshyn, A. Obshta, Y. Kurii, D. Zhuravchak, I. Tyshyk, A. Partyka, P. Haraniuk, T. Kret, V. Yuzevych, V. Otenko, Y. Nakonechnyy, N. Dzianyi, L. Bortnik, M. Stakhiv, T. Lukovskyy, A. Horpenyuk, S. Voytusiuk, R. Kuten, I. Kolbassynskyi, K. Besaha, Y. Furdas, O. Isakov, R. Andriiv, and O. Tsebak, Modern Methods of Ensuring Information Protection in Cybersecurity Systems Using Artificial Intelligence and Blockchain Technology, O. Harasymchuk, Ed., Kharkiv: Technology Center PC, p. 132, 2025, doi: 10.15587/978-617-8360-12-2
- [42] O. Harasymchuk, I. Opirskyy, R. Banakh, D. Zhuravchak, O. Partyka, A. Piskozub, E. Nyemkova, S. Vasylyshyn, A. Partyka, Y. Nakonechnyy, T. Lukovskyy, V. Susukailo, V. Otenko, I. Tyshyk, N. Dzianyi, D. Sabodashko, P. Haraniuk, V. Dudykevych, S. Semenyuk, M. Stakhiv, I. Zhuravel, T. Kret, L. Mychuda, Z. Mychuda, O. Polotai, Y. Kurii, N. Nakonechna, N. Luzhetska, A. Obshta, and T. Korobeinikova, Intelligent Cyber Defence Systems: Detection of Ransomware and Protection of Wireless Networks Based on Artificial Intelligence Technologies, Kharkiv: Technology Center PC, p. 132, 2025, doi: 10.15587/978-617-8360-22-1