Tuijin Jishu/Journal of Propulsion Technology
ISSN: 1001-4055
Vol. 45 No. 1 (2024)

# Unveiling the Cryptographic Maze: A Comprehensive Exploration of Crypto-Ransom ware Lifecycle and Mitigation Strategies

**Subalakshmi. R[1], Ramya. D[2]**

[1]*Department of Computer Applications, PSG CAS, Coimbatore 641014, Tamil Nadu, India*

[2]*Department of Statistics, PSG CAS, Coimbatore 641014, Tamil Nadu, India*

***Abstract:-*** Decoding the Menace: Unraveling the Crypto-Ransomware Encryption Techniques and introducing the Pre-Encryption Detection Algorithm (PEDA). This paper delves into the pervasive threat of ransomware, a malicious software that obstructs user access to systems and files, coercing victims into paying a ransom. Specifically, the focus is on crypto-ransomware, which employs advanced encryption methods to lock diverse file types, demanding payment for decryption keys. The evolution of ransomware encryption techniques underscores the need for robust detection systems. In response, this study introduces the Pre-Encryption Detection Algorithm (PEDA), designed to identify crypto- ransomware at the pre-encryption stage, before any files are locked. The algorithm's initial phase involves signature comparisons, paving the way for an enhanced defense against the growing threat landscape.

***Keywords:*** *Ransomeware, Security, Detection, Prevention, Pre-Encryption Detection Algorithm, application program interface API*

## 1. Introduction

Encryption is a method of protecting a message's content from unauthorized access by converting it from a readable form to an unreadable form. Network security makes use of it frequently to guarantee that only the intended recipient can view its content. Because the data can be easily stolen, it is crucial to encrypt the data, particularly during its transit via the network. Instead, it has now been established that encryption is a double-edged sword. It was a great method for achieving the goals of data secrecy, data integrity, data authenticity, and non- repudiation for cyber security at one point. Cybercriminals utilized it last to lock their victims' files and demand ransom. This improper handling is present in a type of intrusion attack known as crypto-ransomware.

The term "ransomware" refers to a modern subtype of intrusion attack that aims to demand a ransom from its target. Ransomware mostly comes in three different categories. Scareware is the most common type. The first one scares the victim into paying the ransom even if there is no actual threat to them. Scareware employs imitation of authorities to lead victims into making mistakes with their actions. Make a demand for a ransom to avoid being arrested. The next scareware, which is a little different, calls itself leakware because it threatens to reveal the victim's wrongdoing to their friends and relatives. Locker- ransomware is the following class of ransomware. By showing a login page, this type of ransomware locks the victim's system. The victim must pay the fee in order to obtain the password necessary to unlock the system.Locker- ransomware is regarded as being marginally hazardous due to the fact that restarting the system in Safe Mode frequently reveals the attack. Crypto-ransomware is the third type of ransomware [1]. Since it will encrypt the victim's data, making it impossible to access them without a working decryption key, this sort of ransomware is considered to be exceedingly hazardous [2]. Depending on how strong the encryption mechanism is, decrypting with a brute-

force method could take a very long time. Zavarsky and Lindskog (2016) analyzes and discusses the life cycle of Windows-based ransomware and describes the development of ransomware for Windows for Malware analysis system. [3]. Carter, Traynor, and Butler (2016) says ransomware is a hassle that can be fixed by erasing the system or taking out the disk and extracting the user's crucial data [4]. Song, Kim, and Lee (2015) suggested approach combines various network classification techniques with ad hoc network resolutions while analyzing DNS, HTTP, and SSL protocols evaluated the background information's impact on the performance [6]. Bekerman et.al (2015) employed methods like FrequenSel and AppExtractor . FEST analyzes a program on a typical PC in about 6.5 seconds, making it incredibly time-efficient for malware detection in Android markets [7]. Zhao et. al (2015) developed behavioral model network which will be equipped with a realistic portrayal of malware spread via the Cyber Army Modeling and Simulation (CyAMS) model. [8]. Brown et. al (2015) promotes Software Restriction Policies (SRP) and the Remote Desktop Protocol (RDP) are employed in the procedure. Firewalls should be set up to deny access to known malicious IP addresses [9]. Sahay and Sharma (2015) proposes HelDroid which is a quick, organized, fully automated technology that can identify both known and unidentified scareware and ransomware strains. The most important strategy is to agree on whether a mobile application attempts to lock the device, encrypt data, and make threats. [11]. Zanero (2015) represents the behavior of specific malware kinds that damage the Internet and other workplace email systems. A sandbox test environment platform using virtual machines was created to conduct research, imitate real-life malware behavior, and discover its signature at the moment of deployment for proper analysis[12]. Anthony Ayodele et. al (2011) discuss about the sniffer-2 needs more time to search through a database with more rules than the sniffer-1 did [13]. Sunny Behal and Krishan Kumar (2011) discusses a machine learning method, a method based on graph features, and the creation of a virus detection model based on features. In this research, a novel method for extracting structural information from PE files' Control Flow Graph is presented [14]. Zongqu Zhao (2011) proposes a novel technique to categorization based on directory structure and content similarities and its goal is not to reclassify Binary malware [15]. Mirosław Skrzewski (2010) examined the three stages of ransomeware in this study are exploitation, delivery, backup spoliation, file encryption, and clean-up. In order to defend against a ransomware assault, one must be sufficiently prepared and have the ability to spot, stop, and confine any suspicious behavior [16]. CHEN Chia-mei and LAI Gu-hsin (2014), discusses about viruses that spreads its characteristics to other software components dynamically. Malware can perform its functions by dynamically spreading them to apps that have received user or system approval [17]. Ross Brewer and LogRhythm (2016) contrasts methods for detecting malware that use static, dynamic, and hybrid analysis [18]. Byungho Min and Vijay Varadharajan (2014) suggest a methodology for dynamic malware analysis that makes use of resource monitoring and real-time analysis. Static and dynamic analysis is two methods that are applicable to all types of malware investigation. The three main processes in this paradigm are run- time analysis, resource monitoring, and behavior definition [19]. Anusha Damodaran et. al (2015) proposes a method for feature extraction from a hashed matrix used in large-scale malware investigation. The automatic signature generation utilizing Bayesian signature selection inside clusters is suggested in this study. To grasp the bit-vector representation of the virus in each cluster, feature hashing is used. The Bayesian selection method performs well in terms of speed and accuracy. The Bayesian selection technique offers a means of guaranteeing the signature's low false negative and false positive rates [20].

## 2. Proposed Pre-Encryption Detection Algorithm (PEDA)

The stages of infected crypto-ransomware encompass entry, command and control, encryption, and extortion. The initial stage involves system infiltration and setup, accompanied by additional tasks such as privilege elevation, stealth mode activation, environment mapping, payload persistence, restriction of system restore, and further stealth measures. Subsequently, the command and control stage ensues, establishing a communication link with the command and control center, overseen by the campaign manager or creator. In more sophisticated instances, communication masking is employed to evade detection. Following the establishment of communication, information is transmitted to the victim's system to obtain the encryption key. Crypto-ransomware, in its search phase, identifies the targeted files, and, recognizing their significance to the victim, prompts the willingness to pay the ransom. Once the search concludes, the command and control center dispatches the encryption key. The ultimate stage of crypto-ransomware involves the display of an extortion

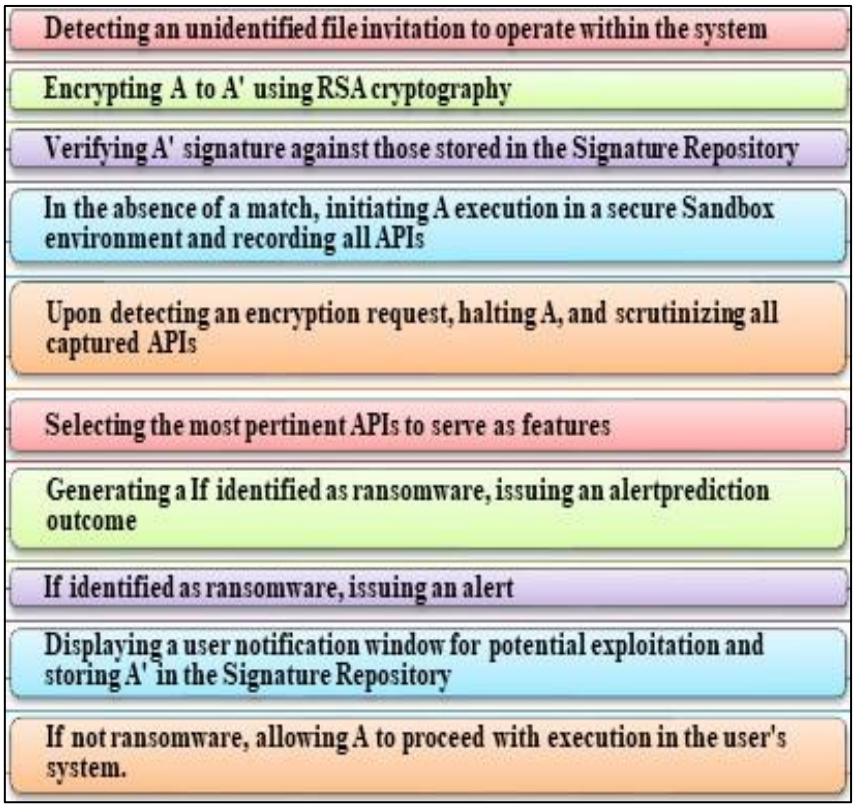
message, demanding a ransom in exchange for the release of the decryption key. This study introduces the Pre-Encryption Detection Algorithm (PEDA), designed to identify crypto-ransomware before the encryption process begins, safeguarding valuable files from ransomware attacks. The algorithm comprises two phases: the learning algorithm in phase I and the signature repository algorithm in phase II – the signature repository algorithm.

**2.1 Phase-I: Learning Algorithm (LA)**

The Cuckoo Sandbox serves as the foundation for implementing the Learning Algorithm (LA). Initially, an unknown file is injected into the sandbox, and at each executable file execution stage, the tool queries the API or System call from the kernel. Upon identifying the encryption API, the execution is promptly terminated.

The Learning Algorithm (LA) plays a crucial role in analyzing the gathered APIs, and it comprises two primary steps: Discretization and Prediction. In the first stage of discretization, numerical values are converted to binary values to expedite the classification process in the prediction step. Each API is treated as a distinct feature, resulting in a dataset with numerous APIs, high dimensionality, and susceptibility to over fitting.

**Figure: 1 Proposed PEDA Model**




The final step involves prediction, employing a homogenous selection based on √F, where F represents the number of features. Randomly chosen features are used to build a decision tree, with node splits determined by the Gini Index

$$(Gini(t) = 1 - \sum Ni=1P(Ci\ |t)2) \quad \text{----------(1)}$$

as expressed in Equation (1). The decision tree's predictions are aggregated to form the final prediction, mitigating overfitting and noise in the data.

$$Gini(t)=1-\sum i=1NP(Ci|t)2$$

In the implementation of the Pre-Encryption Detection Algorithm (PEDA), a labeled dataset trains the prediction model in this phase. The model, designed to identify crypto-ransomware, determines whether the

unknown file warrants execution. If deemed safe, the execution proceeds; otherwise, a warning is issued to the user, and a message is relayed to the Signature Repository.

### 2.2 Phase-II: Signature Repository

The pivotal role of the Signature Repository (SR) in identifying crypto-ransomware is highlighted in this phase. The file's signature is securely stored in an SQL database, facilitating the early identification of unknown files as potential crypto-ransomware. Utilizing the Rivest-Shamir- Adleman (RSA) method, the file is encrypted sans a file extension, yielding a unique signature. The primary objective of the Signature Repository is to proactively recognize known crypto-ransomware instances. PEDA intervenes when a file is on the verge of execution, encrypting it with the RSA algorithm to generate a file signature. SQL queries are then employed to compare this signature with known crypto- ransomware signatures in the Signature Repository.Upon identification of a match, a pop-up notification alerts the user, preventing the execution of the flagged file. Conversely, if no match is found, Phase I of the file execution proceeds. The absence of crypto-ransomware detection in Phase I grants permission for the user system to execute the file. However, in the event of a crypto-ransomware determination, the generated signature is retained in the Signature Repository, and a pop-up alert warns the user.

## 3. Results and Discussion

The comparative analysis involves Random Forest (RF), Naive Thomas Bayes (NB), and an ensemble of RF and NB from PEDA-Phase-I, termed PEDA. The performance is evaluated using four analytical measures, wherein PEDA-Phase-I outperforms others with an impressive FTO of 0.9930, a minimal FPR of 0.1156, and a check error as low as 0.0295. The high FTO underscores the rule's effectiveness in distinguishing between favorable and unfavorable outcomes.

A low FPR implies a reduced likelihood of issuing false warnings, while a minimal check error signifies accurate predictions for both true positive and true negative outcomes. The ensemble exhibits excellent prediction accuracy for favorable outcomes, boasting the best detection rate (0.9872). However, the ensemble rule, despite these merits, lags in performance compared to RF and PEDA-Phase-I.

In the pursuit of enhancing RF performance, the second experiment compares the performance of PEDA-Phase-I with RF using two feature selection methods. Correlation-based Feature Selection (CFS) proves beneficial for RF, achieving an optimal zero detection rate of 0.9647. Notably, Principal Component Analysis (PCA) fares worse than RF alone, indicating its unsuitability as a feature choice for RF.

The third trial contrasts the EldeRan rule with PEDA-Phase-I. The findings suggest a close performance match, with EldeRan exhibiting a slightly better FTO of 0.9949 and a detection rate of 0.9634. EldeRan boasts a lower check error of 0.0238, while PEDA-Phase-I demonstrates a lower false positive rate of 0.0156, implying a lesser likelihood of issuing warnings. PEDA-Phase-I's superior performance over RF and NB is attributed to these results.

Additionally, the experiment configuration analysis reveals essential differences between EldeRan and PEDA-Phase-I. EldeRan utilizes the comprehensive Resilient System Security (RISS) dataset, encompassing all five classes with a total of 30,967 options. In contrast, PEDA-Phase-I operates with a smaller selection, focusing solely on the API class with 232 choices. This demonstrates PEDA-Phase-I's ability to achieve comparable results with a more limited set of choices, emphasizing the significance of API in ransomware detection.

While EldeRan may present a sophisticated model for detecting ransomware, its reliance on cryptography may result in knowledge loss, evident in the inclusion of the born files class in the RISS dataset. PEDA-Phase-I and EldeRan exhibits close performance, with PEDA-Phase-I showcasing efficiency with a smaller set of choices and API's pivotal role in ransomware detection.

## 4. Conclusion

In comparison to other learning algorithms such as RF and NB, the Learning Algorithm (LA) demonstrates superior performance across various metrics, including AUC, test error, false positive rate (FPR),

and detection rate. LA's consistent outperformance holds true even when pitted against the Ensemble of RF and NB, as well as RF with feature choice. This substantiates that PEDA Phase-I or LA effectively fulfills its role as a reliable prediction model specifically designed for detecting crypto-ransomware targeting API data.

The composition of LA involves an ensemble of multiple call Trees and the co-integration of information discretization. Discretization proves instrumental in reducing information redundancy, while ensemble techniques mitigate overfitting concerns associated with high-dimensional data. Assessing AUC, test error, FPR, and detection rate metrics collectively underscores LA's superior overall performance compared to RF and NB.

The primary objective of the PEDA model is to identify ransomware before it initiates the encryption process, a crucial step in preventing the necessity to pay a ransom. However, this is contingent on the availability of a new dataset with API data from the pre-encryption stage, marking a potential avenue for future research. Furthermore, PEDA-Phase-II aims to preserve every ransomware signature discovered, contributing to the establishment of a Signature Repository. This repository plays a pivotal role in halting ransomware at an early stage, precluding any damage from actual malware. Although the process of signature detection is rigorous, it guarantees accurate and swift ransomware detection.

**Refrences**

[1] Tailor, J.P., Patel, A.D. (2017). A Comprehensive Survey: Ransomware Attacks Prevention, Monitoring and Damage Control. Int. J. Res. Sci. Innov., 4, 2321– 2705.

[2] Askarifar, S., Rahman, N.A.A.; Osman, H, (2018). A review of latest wannacry ransomware: Actions and preventions. J. Eng. Sci. Technol., 13, 24–33.

[3] P. Zavarsky and D. Lindskog, (2016). Experimental Analysis of Ransomware on Windows and Android Platforms : Evolution and Characterization. vol. 94, pp. 465–472.

[4] .H. Carter, P. Traynor, and K. R. B. Butler, (2016). CryptoLock ( and Drop It ): Stopping Ransomware Attacks on User Data.

[5] J. Scott and D. Spaniel, (2016). ICIT Ransomware Report.

[6] S. Song, B. Kim, and S. Lee. (2016). The Effective Ransomware Prevention Technique Using Process Monitoring on Android Platform. vol. 2016.

[7] D. Bekerman, B. Shapira, L. Rokach, A. Bar, and B. Sheva (2015). Unknown Malware Detection Using Network Traffic Classification. pp. 134–142.

[8] K. Zhao, D. Zhang, X. Su, W. Li, and E. Engineering. (2015). Fest : A Feature Extraction and Selection Tool for Android Malware Detection. pp. 714–720,

[9] S. Brown, B. Henz, H. Brown, M. Edwards, M. Russell, and J. Mercurio. (2015). Validation of Network Simulation Model with Emulation using Example Malware. pp. 1264– 1269.

[10] P.Y. Networks. (2016). Protecting Your Networks from Ransomware. U.S Government interagency technical guidance document aimed to inform chief information officers and chief information security officers at critical infrastructure entities.

[11] S. K. Sahay and A. Sharma. (2016). Grouping the executable to detect malwares with high accuracy. Procedia - Procedia Comput. Sci., vol. 78, pp. 667–674.

[12] S. Zanero and F. M. B. (2015). H EL D ROID : Dissecting and Detecting Mobile Ransomware," pp. 382–404.

[13] Anthony Ayodele, James Henrydoss, Walter Schrier, and T.E. Boult (2011). Study of Malware Threats Faced by the Typical Email User, Springer.

[14] Sunny Behal, Krishan Kumar. (2011). An Experimental Analysis For Malware Detection Using

Extrusions", International Conference on Computer & Communication Technology (ICCCT), IEEE.

[15] Zongqu Zhao. (2011). A Virus Detection Scheme Based on Features of Control Flow Graph. IEEE.

[16] Mirosław Skrzewski (2010). Monitoring Malware Activity on the LAN Network, Springer.

[17] CHEN Chia-mei, LAI Gu-hsin. (2014). Research on Classification of Malware Source Code. Springer.

[18] Ross Brewer, LogRhythm. (2016). Ransomware attacks: detection, prevention and cure.

[19] Byungho Min and Vijay Varadharajan (2014). Feature- Distributed Malware Attack:Risk and Defence, Springer.

[20] Anusha Damodaran, Fabio Di Troia, Corrado Aaron Visaggio, Thomas H. Austin, Mark Stamp (2015). A comparison of static, dynamic, and hybrid analysis for malware detection. Springer.