International Journal of Computing and Digital Systems
ISSN (2210-142X)
Int. J. Com. Dig. Sys.14, No.1 (Oct-23)

http://dx.doi.org/10.12785/ijcds/1401102

# Crypto-Ransomware Detection and Prevention Techniques and Tools: A Survey

**Hesham Alshaikh[1], Hesham A. Hefny[2,4] and Nagy Ramadan Darwish[3,4]**

[1]*Sadat Academy for Management Sciences (SAMS), Egypt*
[2]*Department of Computer Science*
[3]*Department of Information Systems and Technology*
[4]*Faculty of Graduate Studies for Statistical Research (FGSSR), Cairo University, Egypt*

*Received 31 Dec. 2022, Revised 31 Jul. 2023, Accepted 29 Aug. 2023, Published 1 Oct. 2023*

**Abstract:** Crypto-ransomware is among the extremely prevalent malware cyberattacks worldwide. it is typically spread using phishing emails and compromised websites, where crypto-ransomware is downloaded stealthily after luring users to click on malicious links. Additionally, attackers may take advantage of available vulnerabilities in software installed on the victim's device or use a zero-day vulnerability in the operating system. Crypto-ransomware employs encryption techniques against users' data and resources, rendering them inaccessible and demanding a cryptocurrency ransom for decryption. Crypto-ransomware attacks have witnessed massive growth within the past few years, resulting in massive financial loss across the globe. Different detection and prevention techniques have been proposed to overcome crypto-ransomware attacks, and many tools have been implemented. In this work the authors present a summary of the most recently used techniques and tools, highlighting different employed strategies that address crypto-ransomware attacks in the different stages of the attack chain.

**Keywords:** Ransomware, Malware, Attack Chain, Cybersecurity, Encryption, Static Analysis, Dynamic Analysis, Cybercriminals

## 1. INTRODUCTION

Malware, especially crypto-ransomware, constantly evolves its techniques to evade detection. In the earlier few years, crypto-ransomware has become the most widely spread malicious threat by targeting public and private sector entities in all business fields, especially the healthcare sector. According to Steve Morgan, chief editor of Cybersecurity Ventures, the estimated cost of global cybercrime damages is $8 trillion in 2023, which is $15.22 million per minute, and it is predicted to reach $10.5 trillion in 2025, which is $20 million per minute. Furthermore, global ransomware damage is predicted to transcend $265 billion in 2031 with a ransomware attack every two seconds [1], [2].

The COVID-19 pandemic has encouraged cyber criminals to execute more cybercrimes because of its global prevalence [3]. Therefore, the crypto-ransomware attacks in the healthcare sector have quadrupled over the past 3 years. As a result, it is forecasted that the healthcare industry will spend $125 billion on cybersecurity from 2021 to 2025. Attacking critical sectors such as healthcare providers and hospitals has a direct impact on citizens' lives. For example, a patient in Dusseldorf, Germany, died after she relocated because of a ransomware attack on a major hospital [4].

The main contribution of this study is that the researchers present a summary of the most recently used techniques and tools to detect and prevent crypto-ransomware attacks. Where, they classified the detection and prevention methods employed in the reviewed studies, according to the different attack chain stages, to reveal the prevalent detection techniques more appropriate for detecting crypto-ransomware attacks in each attack chain stage. Consequently, they emphasize the importance of making more efficient detection and prevention tools, to break the attack chain as soon as possible to prevent crypto-ransomware attacks.

This paper is organized into five sections, the first one being the present introduction which, exposes the devastating impact of crypto-ransomware attacks on the economy and the healthcare sector. Then, section 2 describes the phases of the crypto ransomware attack chain, and the main techniques used to detect crypto ransomware attacks. Section 3 explains the defenses applied at the infection, installation, communication, execution, extortion, and emancipation stages. Section 4 discusses the presented detection and prevention techniques, which are summarized in Table I. Finally, the concluding remarks and future work are presented in section 5.

*E-mail address: hr@pg.cu.edu.eg, hehefny@ieee.org, nagyrd@cu.edu.eg*

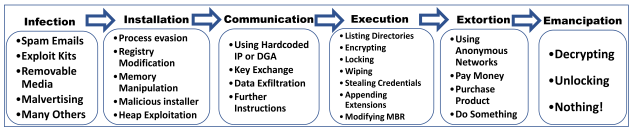


Figure 1. Crypto-Ransomware Attack Chain

## 2. BACKGROUND

The ransomware literature confirms that crypto ransomware attacks go through several phases, to successfully encrypt organizations and individuals' important files and exfiltrate valuable information [5], [6], [7], [8].

So, segmenting the crypto ransomware attack into distinctive phases allows cybersecurity analysts to understand the ransomware attack flow and, assists developers in designing more efficient security systems.

For instance, the infection phase reveals the different types of attack vectors used to deliver crypto ransomware malicious payloads, which increases the awareness of how these attacks are primarily initiated. Thus, it will help information technology teams to take more proactive countermeasures and assist cybersecurity developers in implementing more effective security systems to crack the attack chain and stop the crypto ransomware attack as soon as possible.

Therefore, to avoid, prevent, and mitigate the catastrophic impact of crypto ransomware attacks, it is substantial first to understand the crypto ransomware attack stages, where most crypto ransomware families typically go through six steps, to infect a user device as shown in Figure 1.

### A. Crypto-Ransomware Attack Chain Stages

#### 1) The Infection Stage

In this stage, the initial intrusion into the system happens more probably through the weakest component in any organization "people" via social engineering. So, crypto ransomware may attack targets using many attack vectors including but not limited to the following [9].

- Social engineering.
- Spam emails.
- Phishing emails.
- Compromised websites.
- Malicious advertising.
- Vulnerable ports.
- Back doors.
- Exploit Kits (EK) such as "Angler, Neutrino", "Magnitude", and "SweetOrange".
- Available vulnerabilities in installed software.
- Zero-day vulnerability in an installed application or the OS.
- Malicious scripts embedded normal files.
- Infected removable drive.
- Ignorant workers of ransomware attacks.
- Legitimate credentials from previously leaked information.

#### 2) The Installation Stage

In this stage, the crypto ransomware extracts and installs its malicious components on the victim device and modifies the compromised system registry to maintain persistence to run whenever the infected system starts up. Also in this stage, the crypto ransomware starts to generate a list of running processes in the infected device to identify both processes vulnerable to injection to exploit it, and security processes such as antivirus, antimalware, and any other security software, to terminate in a later step.

#### 3) The Communication Stage

In this stage, the crypto ransomware such as some strains of CryptoLocker, CryptoWall, and TorrentLocker, starts to communicate with its Command-and-Control server (C2) to obtain the encryption keys, which will be used to encrypt users' valuable files [5]. Next, the crypto ransomware starts to enumerate all directories on the infected device to list all valuable files and documents according to predefined criteria. But, not all ransomware families propagate through the directories in the same way. Where crypto ransomware strains spread in a victim's device differently, depending on the directory size order, and it may iterate randomly, or alphabetically. However, if all files are placed in the same directory without subdirectories, these differences will not be noticeable [10].

Crypto ransomware does not encrypt files randomly, but it looks alphabetically in most cases [11]. Similarly, directories are listed and processed alphabetically [12]. Moreover, crypto-ransomware typically does not encrypt the entire hard disk content because if it does, the computer will stop functioning, and the cyberattack group responsible for the attack would not get paid since their ransom message will not be furnished to the victim [12], [13]. So, in most cases crypto-ransomware targets specific file types such as files with the following extensions: .docx, .doc, .xlsx, .xls, .pptx, .ppt, .mdb, .accdb, .iso, .zip, .jpg, .bmp, .mp4, .mp3 [12], [14]. These are samples of files containing critical business and personal data that may enforce organizations and individuals to pay the ransom since they are considered among the most valuable assets in the computing era and will affect most users and corporations if lost. Moreover, some crypto ransomware encrypts not only the content of files but also file names, making it more challenging for

the victim to know the extent the assailants have gone and which files have been lost [12], [15].

Also, in this stage, the crypto ransomware inspects the network to execute lateral movements to infect other vulnerable devices in the local network and locate the most valuable assets in the current network, such as backup servers and network files to encrypt them. In addition to escalating their privileges by obtaining admin credentials to import extra malicious components from its C2 server, to conform to the different existing systems in the infected network to take over complete control of the organization's digital environment.

Furthermore, in this stage, the crypto ransomware starts to exfiltrate any valuable files containing personal or financial information. For example, crypto-ransomware such as modern editions of Cerber steals the user's Bitcoin wallets also, Reveton, CryptoLocker, Urausy, SamSam, Cryptowall, Kovte, and Ryuk steal user's critical data. Crypto-ransomware attackers use stolen data as leverage to ensure victims will pay the ransom by threatening to leak it to other cybercriminals on the dark web [16], [17], [18] and if the attackers' demands are not fulfilled, the encrypted data remain inaccessible, and the decryption key will be deleted permanently [19]. So, backup strategies unfortunately can't mitigate the consequences of leaking sensitive information. Therefore, safeguarding user important data from spreading on the dark web by stealer-ransomware is crucial.

*4) The Execution Stage*

In this stage, the crypto ransomware starts to encrypt the critical files and documents in all enumerated directories in the infected system to prohibit users from using them. Moreover, the crypto ransomware changes the encrypted file extension, and file names to disturb the victims as much as possible, to enforce victims to pay the ransom.

*5) The Extortion Stage*

This stage starts with displaying the ransom demand, which includes how to pay the ransom using cryptocurrency with a reminder of the payment deadline. Also, the stolen data will be used to extort the attacked victim by threatening to publish the sensitive exfiltrated information on the dark web to other cybercriminals or to sell it to competitors if the ransom is not paid during the specific period Figure 2.

*6) The Emancipation Stage*

In this stage, if the victims paid the required ransom, then they will be waiting to receive the decryption key, which may happen or not, according to the threat actors' intention. Where the wiper crypto ransomware families delete the encryption keys immediately after encrypting the target files or use random encryption keys. While other families never send the decryption keys, even after receiving the ransom. But some crypto ransomware attackers keep their promises and send the decryption key to victims who paid the ransom. Otherwise, users ought to use data recovery tools and hopefully be lucky and retrieve some of their important files.


Figure 2. Phobos Crypto Ransomware Note

## B. Detection Technique

The crypto ransomware literature confirms that signature-based and behavior-based methods are the major techniques used for detecting crypto ransomware. Where each of them may be used separately, combined, or add to other methods forming hybrid techniques.

*1) The Signature-Based Approach*

This approach uses static-based analysis techniques to analyze the malicious file characteristics, such as file metadata, file size, and source code contents. Where signatures are generated using unique patterns such as a distinctive sequence of bytes, the order of call functions, and the content of the ransom demand message, to detect similar files in the future. The generated signatures are saved in a database. Next, any suspicious files are scanned by an anti-ransomware to compare their signatures against the signatures database, to detect any malicious patterns in the inspected file.

The signature-based techniques are preferable methods to detect crypto ransomware because they are fast methods, with a low false positive ratio, and can detect potentially malicious payloads before executing the suspicious file, where defending actions are triggered if a malicious pattern is found. So, methods such as Machine Learning (ML), Neural Network (NN), anomaly detection, and classification, employs file signatures to detect crypto ransomware attacks.

Nevertheless, signature-based approaches cannot be used to detect new strains of crypto ransomware in real-time, since ransomware rapidly applies multiple mutations to its source code. Also, different ransomware families use many advanced obfuscation mechanisms to evade signature-based security systems. Moreover, [19] predicted the features of future ransomware, its expected impact, and how

it will be difficult to be detected if polymorphic, metamorphic, and other obfuscation techniques are used by crypto ransomware.

Therefore, we assert that the signature-based technique alone is incapable to overcome the obfuscated code in crypto ransomware and cannot discover new strains of crypto ransomware till they are analyzed by analysts to generate the corresponding signatures [20], [21].

*2) The Behavior-Based Approach*

The dynamic-based detection mechanisms are based on testing ransomware samples in an isolated environment, using sandbox techniques that provide a realistic execution environment. While observing how the crypto ransomware behaves using monitoring tools. Through analyzing the Indicators of Compromise (ICs) obtained from file system activity logs, the crypto ransomware malicious behaviors such as encrypting file contents, file name changes, extension changes, and rapid file deletions are revealed. So, dynamic-based techniques are regarded as effective to detect new variants of crypto ransomware in real-time, by monitoring and inspecting anomalies of:

- Running processes behavior.
- Registry keys changes.
- Application Programming Interface (API) calls.
- Crypto functions call.
- System files change.
- Hardware performance of CPU, Random Access Memory (RAM), Hard Disk Drive (HDD), and power consumption for instance.
- Network activities.
- File system event log files.

Static-based analysis and dynamic-based analysis are major techniques used for detecting malware. The static-based approach is a fast method for detecting potentially malicious payloads before executing portable executable (PE) files, but it cannot detect new strains of malware in real-time. Different ransomware families use many obfuscation techniques to bypass signature-based security systems. Therefore, new variants of ransomware can be detected through dynamic-based approaches, where samples are examined in an isolated environment using a sandbox technique to monitor and inspect running processes, registry alterations, and network activities, which is regarded as an effective defense against crypto-ransomware attacks [13], [22].

## 3. RELATED WORK

Due to the catastrophic impact of crypto ransomware attacks, whether on ordinary users, multinational enterprises, or governments. It is highly recommended to discover crypto ransomware attacks in their early stages as quickly as possible. The crypto ransomware attacks are executed in sequential stages, which is known as the attack chain that had illustrated in the previous chapter in Figure 1. In each stage, the crypto ransomware executes some activities on the victim's device. Thus, we will review defensive approaches tackled by the recent research in each stage.

*1) Defenses at The Infection Stage*

Static-based techniques are mainly used in this early stage, where file signatures are used to identify ransomware attacks. Hence, once the malicious payload is delivered through any aforementioned infection vector in the section "1) The Infection Stage", the loaded files are analyzed to detect malicious codes using static-based analysis approaches.

Some researchers, such as [23], [24] converted static data extracted from the binary files into images and used machine learning (ML) algorithms to detect ransomware, with an accuracy of 93.33%, and 98.77%, respectively, whereas [25] converted the entire file into an image and used the local binary pattern (LBP) algorithm, achieving detection accuracy of 87.9%. Additionally, [26] converted each binary file into a hexacode formula and utilize it to generate an image using an image transfiguration algorithm and using a convolutional neural network algorithm (CNN), they achieved a detection accuracy of 63%. While [27] applied the ML algorithm on the hexacode, achieving an accuracy of 88.39%. But researchers such as [28], [29] combined extracted static features with ML algorithms and used random forest (RF) and deep learning (DL) classifiers, and achieved an accuracy of 97.74% and 99.30%, respectively.

On the other hand, [30] employed hybrid techniques applying static-based analysis and dynamic-based analysis on PE files to extract behavioral properties, such as dynamic link library (DLL), and static properties, such as file size and file entropy. RF achieved an accuracy of 98.34% and 99.25%, respectively, in classifying static and dynamic features. Also, [31] used the same technique but for static features, N-gram was used to detect important sequences, whereas wrapper-based mutual information was used to reduce 4000 dynamic features to 300 important features. RF achieved an accuracy of 98% and 92% using static and dynamic features, respectively.

Furthermore, [32] proposed a three-level security architecture to prevent ransomware infection using a browser extension, virtual machine, and anti-ransomware. When a file is downloaded from the internet, the browser extension sends it to a cloud server built over a virtual machine that has an anti-ransomware that scans files, and the user is given the option to save that file locally on the user device only if it appears safe otherwise, it will be deleted.

*2) Defenses at The Installation Stage*

In this stage, dynamic-based analysis techniques are essentially used, to detect abnormal activities. Reference

[33] used a finite-state machine, which is a decision-making mathematical model that analyzes events collected about user files, retention state of applications, lateral movement of files, and abnormalities in system resource usage patterns. This method achieved ransomware detection accuracy of 99.5% and a 0% false-positive rate (FPR).

Reference [34] proposed DeepRa, a DL-based early detector, and classifier for ransomware. They used the term frequency-inverse document frequency (TF-IDF) as a term-weighting method to extract semantic information from a time-series host log. A recurrent neural network (RNN) was used to detect abnormal activities with an accuracy of 99.87%. Crypto-ransomware was classified using attention-based bidirectional long/short-term memory (BiLSTM), achieving an accuracy of 96.5%. The underlying model is frequently updated with new observations using a backpropagation algorithm to evade over time model quality degradation.

Whereas [35] proposed RansomSpector, which resides in the hypervisor layer to make it hard for ransomware to bypass by privilege escalation. Thus, ransomware can be detected without requiring any OS modifications, where a malicious process in the guest OS is automatically eliminated by clearing the memory page that stores the process code. This approach has an average overhead of 2.49% and ¡5% for network and device performance, respectively, and only 2.67% of user files are lost.

*3) Defenses at The Communication Stage*

Through this stage, crypto ransomware moves laterally to infect other nods in the same network and other interconnected networks. Therefore, many studies conducted to detect and prevent crypto ransomware propagation such as [36], who studied the emerging cyber threat to crucial infrastructure and emphasized the role of the network segmentation approach, in prioritizing the security of production network devices and limiting ransomware propagation.

Also, crypto-ransomware such as Simplocker, TorrentLocker, CryptoLocker, and families with similar behavior, attempt to communicate with their C2 servers to obtain encryption keys to encrypt user files. Thus, hindering this communication will prevent this crypto-ransomware from initiating the encryption process. So, some studies are based on monitoring the network traffic to detect malicious network flows by using neural networks, DL, and ML models to prevent ransomware attacks.

Reference [37] proposed a real-time network analysis model using DL methods to predict irregular ransomware traffic. Using a CNN combined with RNN, they achieved a result of 96.5% in tracking down the ransomware communications. Additionally, [38] proposed an adaptive security architecture using open source to monitor a network to detect ransomware attacks, hence implementing a coordinated logging mechanism that correlates the captured security logs to increase the network resilience to confront malware attacks. Therefore, using the proposed framework in addition to the Windows AppLocker can prevent ransomware attacks on the Windows platform.

Similarly, [39] proposed DeepMal, which is based on an RNN combined with a CNN to capture the underlying malicious traffic statistics and learn spatiotemporal features from raw flow traffic, achieving a malware classification accuracy of 98.6%. Furthermore, [40] proposed an ML-based technique to capture ransomware in encrypted network traffic, by analyzing information about the network connection, certificates, and encryption. They used the Bro intrusion detection system (IDS) to generate the network traffic log. They applied the RF algorithm, which achieved a 99.9% detection rate of ransomware traffic.

*4) Defenses at The Execution Stage*

This stage is crucial for capturing new strains of crypto ransomware, where crypto ransomware executes all its malicious behavior. Therefore, dynamic-based analysis techniques are effectively used to detect anomalous behavior. In this stage, malicious activities are detected by monitoring hidden patterns in the I/O request packet (IRP) as shown in the study [41], which is based on executing the ransomware in an isolated environment to generate an IRP log, where actionable insights are debriefed from the generated IRP logs using an ML model. This model detects ransomware with an accuracy of 99.7%.

[42] proposed an early detection tool termed "RWGuard" based on detecting anomalous behavior through monitoring canary files, abnormal file changes, running processes, and encryption calls. The model achieved 98.7% detection accuracy. Additionally, [43] developed a host-based model that depends on monitoring API calls, CPU performance, running processes, registry keys, and environment sniffing-related events. This model identifies malicious processes with an accuracy of 84%. Also, [44] proposed a prediction model based on a NN. The NN classifier was trained using data about disk space, CPU and memory usage, file read, write, create, and delete. This model detects ransomware with 99.98% accuracy. Moreover, [45] proposed a self-defense technique using ML. They used the "Tiny Tracer" app to collect system behavior indicators, such as used DLLs, called APIs, and usage of HDD and CPU to detect ransomware attacks.

Also, [46] used a mixed detection model that employs the Markov chain model to capture the API call patterns and a statistical ML model to detect misclassified ransomware. This model achieved 79.28% overall accuracy, 4.83% FPR, and a 1.47 false-negative rate. In addition, [47] proposed an ML detection model based on a Support Vector Machine (SVM) and Artificial Neural Networks (ANNs) to detect ransomware behavioral ICs. Where selected features such as API calls and registry keys are ranked using TF-IDF. This model achieved an accuracy of 98.7% with ¡ 3% FPR.

Similarly, [48] used the same features in addition to

monitoring the file magic number and file entropy in their prediction model to detect new strains of ransomware and achieved an accuracy of 97%. Furthermore, [49] proposed an ML-based model, where the "intel PIN tool" is used to extract file-related API, which is used to generate feature vectors using Class Frequency-Non-Class Frequency (CF-NCF). This model achieved an accuracy of 98.65% for ransomware detection.

Whereas, [50] proposed an anomaly-based mitigation framework named "RATAFIA". Where they utilized the "perf tool" to collect Hardware Performance Counters (HPCs) from some events to identify malicious encryption processes. Then, they managed to reduce the FPR using a Long/Short-Term Memory (LSTM) autoencoder. This framework backs up active files to be recovered if encrypted. This is similar to their previous work [51] where they proposed a tool called RAPPER to monitor the HPCs. The proposed tool is a two-step detection framework, that uses an ANN to learn the normal behavior of the system by analyzing the statistics obtained from the HPCs.

It is worth noticing, that some benign applications may be considered as crypto ransomware using this framework since they encounter irregular reads according to the HPCs indicators. Although, some crypto ransomware families have a low footprint which leads to encrypting several files before the crypto ransomware is detected. Furthermore, this framework is implemented for Linux only and relies on user opinion to determine if a disk encryption process is malicious or not, which is a defective strategy.

*5) Mitigations at The Extortion Stage*

Although it is late-stage, it can provide an opportunity for recovering from crypto ransomware attacks. Where, [52] argued that by identifying the employed key generation method, a defense can be used to recover the encryption key. So, different techniques are utilized to recover from ransomware attacks without paying the ransom. [53] employed a recovery technique that managed to recover the encryption keys from the "Magniber v2" ransomware with up to 2128 attack complexity by exploiting a vulnerability in the used Pseudo-Random Number Generator (PRNG).

Another technique is based on preventing malicious processes from accessing the PRNG. This technique was employed by [54] who proposed blocking any unauthorized calls to the Windows secure PRNG, to prevent cryptographic ransomware malicious encryption. But, neither this approach nor their previous work, [55] is immune to malicious process injection, where they proposed allowing only white-listed application calls. So crypto API calls may be occurred by the infected legitimate process.

A different technique was proposed by [56] where they suggested a security approach based on replacing the OS' PRNG with a customized random number generator. Therefore, encrypted files can easily be decrypted by reproducing the keys.

Another technique is based on retrieving the encryption keys from the memory of infected devices as given by [57] where proposed an in-memory attack tool named "Pickpocket" that can access the encryption keys during the encryption process. Additionally, [58] employed digital forensics tools named "Findaes, Interrogate, and RansomAES" to retrieve used AES symmetric encryption keys from the infected system's captured memory.

Other techniques are based on recovering deleted user files as in [59], [60], or recovering the ransomware PE files to analyze them as in [61]. The study in [59] proposed a self-recovery service that runs at the kernel level. They employ a rule-based detection logic, monitoring the activity of users and important System files. Therefore, they managed to mitigate the ransomware damage on the Internet of Things edge servers by recovering data from the backup node after establishing a secure network connection. Furthermore, [60] proposed a ransomware detection and data recovery technique dubbed "SSD-insider++," which is embedded in the SSD controller as a form of firmware. Where a lazy detection algorithm evaluates entropy value changes, and if a difference is detected, the "SSD-insider++" informs the user, offering to recover the original files. They leveraged the delayed deletion feature of the SSD, and a recovery algorithm was used to recover the original files. The SSD-insider++ has 100% detection accuracy.

However, [61] recovered ransomware PE self–deleted files using forensics tools to be reverse-engineered and grasp more information on how to prevent and mitigate this strain of crypto-ransomware. To find evidence, they checked the Windows Registry. Though some crypto ransomware families such as "Ordinypt" and "Petya", encrypt files with randomly generated keys, which will be tedious to decrypt all encrypted files. Also, recovering the encryption keys may be ineffective against crypto ransomware families such as RansomEXX and RansomPoc, which encrypt just a small part of each file rapidly. Consequently, the time window to recover the encryption keys will be very short, which may prohibit key retrieval and leads to permanent data loss [62] and [63].

*6) The Emancipation Stage*

In this stage, the user's available options are very limited to recovering deleted files using recovery tools or restoring them from an intact offline backup. The worst case is submitting to the adversary's demands and paying a ransom hoping to obtain the decryption tool. This may apply to cybercriminals groups that fulfill their obligations, such as the "TeslaCrypt", "TorrentLocker", and "CTB-Locker" creators. However, untrusted attackers receive the ransom and never send the decryption tools such as "NotPetya" and "WannaCry" creators.

## 4. DISCUSSION

In this paper, the authors emphasize that crypto-ransomware is a dangerously powerful tool in the hands of cybercriminals, wherefore, ransomware is among the very

TABLE I. Summary of Current Research

| | Ref. | Employed Techniques | Evaluated Parameters | Tools / Datasets | Accuracy |
|---|---|---|---|---|---|
| Def. at Infection | [23] | Static-based analysis, CNN | PE file header, code to image | Freeware, Snap files, Portable apps, Virus Share | 93.33% |
| | [24] | Static-based analysis, CNN | Packer information, hash value, PE metadata, dynamic link library | TransFlow /VX Heaven, classification dataset | 98.77% |
| | [25] | Host-based monitoring approach, CNN, LBP algorithm | Detect irregularity in the image texture, number of iterations | Cuckoo, OpenCV, Keras / VirusTotal, Contagio, Open Malware | 87.90% |
| | [26] | CNN, horizontal feature simplification | Android app. source code, hexadecimal code, RGB images | | 63% |
| | [27] | ML, information gain-based feature selection, RF | Binary files, hexacode | "Objdump" Linux command | 88.39% |
| | [28] | ML, static-based analysis, gain ratio mechanism, RF | Binary file raw byte | Weka, VM / VirusTotal, portable apps platform | 97.74% |
| | [29] | DL, static analysis, RF | Binary file, opcode sequences | IDAPro, Scikit-learn | 99.30% |
| | [30] | Static, dynamic analysis,Naïve Bayes, RF | PE file, DLL, file size, time-stamp, entry point address, file entropy | PE explorer, PEid, WinDbg, Sys-internal tool, VM / GitHub, VX Heaven | 98.34% static 99.25% dynamic |
| | [31] | Static, dynamic analysis, transfer learning, deep CNN conventional ML, RF | PE file information, file activities | Cuckoo sandbox, PEFile Python library | 98% static 92% dynamic |
| | [32] | Cloud-based monitoring approach | A browser extension, VM, anti-ransomware | VM | |
| Def. at Installation | [33] | Host-based monitoring, behavior analysis, decision-making module | Pattern of system resources, user files, retention state of apps, lateral movement | Oracle Virtual box, Visual Studio 2013 / VirusTotal, TheZoo | 99.50% |
| | [34] | TF-IDF, RNN, BiLSTM, backpropagation algorithm | Time-series host log | Windows logging services (WLS) | 99.87% detection 96.5% classifi. |
| | [35] | Host-based monitoring approach | Guest OS files activity, network activities | VM, KVM, AVClass / VirusTotal, Virus Share | 86.51% |
| Def. at Communicat. | [37] | CNN, RNN | Network traffic anomalies | Kebana tool, Wire Shark / Kaggle dataset | 96.50% |
| | [38] | Adaptive security monitoring the network | Network traffic anomalies | Snort, OSSEC intrusion detection, rsyslog, VM, Graylog, OPNsense | - |
| | [39] | ML-based arch., RNNs, CNN | Raw flow network traffic | USTCTFC2016 (CTU), Ixia BreakingPoint | 98.60% |
| | [40] | ML-based, RF | Network traffic log | Bro intrusion detection system (IDS) | 99.90% |
| Def. at Execution | [41] | ANN structure, isolated environment | I/O request packet (IRP) log | Sniffer tool, AVClass tool / VirusTotal | 99.70% |
| | [42] | Host-based monitor | Decoy, abnormal file, process monitoring, I/O request packets (IRPs), cryptographic activities | Pmon, Kryptel / VirusTotal, Open Malware, VXVault, Zelster, Malc0de | 98.70% |
| | [43] | Host-based monitor | API calls, CPU, registry, processes performance | Oracle Virtual box | 84% |
| | [44] | Host-based monitor, NN prediction model | Disk, CPU, RAM, files create, delete r/w | SysMon, RanSim tool | 99.98% |
| | [45] | ML | API calls, DLL, HDD, CPU indicators | Tiny tracer, VM | - |
| | [46] | Markov chain, ML model, RF | API calls | Cuckoo sandbox | 97.28% |
| | [47] | SVM, ANN, TF-IDF | API calls, registry keys | Cuckoo sandbox | 98.70% |
| | [48] | Monitoring, RF behavioral prediction | API calls, registry keys, magic number, entropy | Scikit-learn, Cuckoo sandbox / VirusTotal | 97% |
| | [49] | Host-based monitoring, CF-NCF | API calls | Intel PIN, Vmware, Scikit-learn / VirusTotal | 98.65% |
| | [50] | ANN, FFT | (HPCs) | Perf tool | - |
| | [51] | Anomaly detection-framework, LSTM | Hardware performance counters (HPCs) | Perf tool, Keras Python library | - |
| | [52] | Forensics analysis approach | API calls, recover the encryption key | Forensics analysis tools, Protection ID, PEframe, de4dot, Oracle VM | - |
| | [53] | Padding verification, statistical randomnes | Pseudo-random number generator (PRNG) | Virtual environment | - |
| | [54] | PRNG, white-listed applications | API calls | AVClass tool, Cuckoo sandbox / VirusTotal | 97.10% |
| | [55] | PRNG, white-listed applications | API calls | KVM, Cuckoo /Malc0de VirusTotal, Virusign | 94% |
| | [56] | Host-based monitoring approach, customized RNG | API calls | | - |
| | [57] | Forensics analysis approach | Volatile memory, side-channel vulnerability, recovering used key | Pickpocket | 92% |
| | [58] | Forensics analysis | Volatile memory data | Findaes, Interrogate, RansomAES | - |
| Def. at Extortion | [59] | Host-based monitoring, rule-based detection | Monitoring user, system files activity, storage | | - |
| | [60] | Lazy detection algorithm | I / O patterns | SSD-insider++, FSCK tool | 100% |
| | [61] | Forensics analysis, Reverse-engineering | Windows registry, file operations | Virtual environment | - |

active research areas due to its devastating influence financially, economically, and on lives as well. Also, drawing attention to the destructive impact of the stealer-ransomware on the organization's reputation due to the leaking of sensitive information about customers and businesses on the dark web. Hence, the authors cast new light on the importance of encrypting files containing critical information, to protect data at all three states, at rest, in transit, and in use, to avert the devastating effect of stealer-ransomware.

From the presented studies, the researchers confirm that each attack chain stage has prevalent detection techniques that are more appropriate for detecting crypto-ransomware attacks. Even though static-based approaches achieved high detection accuracy, they can't detect new/unknown ransomware in real-time. While dynamic-based approaches are considered more efficient in detecting new and unknown crypto-ransomware in real time.

This study reviews some of the most recent studies that address the crypto-ransomware issue. Where the presented detection and prevention techniques, are summarized in Table I. We hope to be beneficial for researchers in this area to determine the most effective and easiest-to-implement techniques they would like to employ to eliminate ransomware as efficiently as possible. Hence, we encourage other researchers in the crypto-ransomware attack field to regard the different attack chain stages, when planning new methods to detect and prevent crypto-ransomware attacks. To create more efficient detection and prevention tools, that can break the attack chain and prevent crypto-ransomware attacks.

## 5. CONCLUSION AND FUTURE WORK

In this study, the researchers conclude that each phase of the ransomware attack chain has dominant methods to detect and prevent crypto-ransomware. Where static-based approaches are normally employed in the early and late stages, while dynamic-based approaches are typically used in the middle stages. Therefore, the crypto-ransomware attack chain must be considered when implementing new crypto-ransomware solutions. Further, combining more than one approach may incur more computing overheads, but may increase detection accuracy. However, striking this balance would be an interesting future direction of research.

Authors believe that crypto-ransomware which steals user important files before initiating the encryption process is considered a widely critical issue and has a terrifying impact on the digital era. Therefore, and due to this horrifying influence of crypto-ransomware, in future work, authors will aim to find new solutions for protecting user critical files from being leaked to the dark web due to infection by stealer-ransomware. Accordingly, the authors recommend to:

- Keep precious files in an encrypted form which is very much the key component in future attempts to overcome the stealer-ransomware impact.

- Segmenting the network properly isolates important assets and protects them from encryption.
- Regularly create an up-to-date backup and keep them offline to avoid encryption.

## References


[1] S. Morgan, "Boardroom Cybersecurity 2022 Report — cybersecurityventures.com," https://cybersecurityventures.com/boardroom-cybersecurity-report/, 2022, [Accessed 25-Apr-2023].

[2] ——, "Top 10 Cybersecurity Predictions and Statistics For 2023 — cybersecurityventures.com," https://cybersecurityventures.com/top-5-cybersecurity-facts-figures-predictions-and-statistics-for-2021-to-2025/, 2022, [Accessed 25-Apr-2023].

[3] ——, "FBI Cyber Division Section Chief Warns Of Ransomware — cybersecurityventures.com," https://cybersecurityventures.com/fbi-cyber-division-section-chief-warns-of-ransomware/, 2020, [Accessed 25-Apr-2023].

[4] ——, "Ransomware Runs Rampant On Hospitals — cybersecurityventures.com," https://cybersecurityventures.com/ransomware-runs-rampant-on-hospitals/, 2020, [Accessed 25-Apr-2023].

[5] M. Keshavarzi and H. R. Ghaffary, "I2ce3: A dedicated and separated attack chain for ransomware offenses as the most infamous cyber extortion," *Computer Science Review*, vol. 36, p. 100233, 2020.

[6] M. S. Khan, S. Siddiqui, and K. Ferens, "A cognitive and concurrent cyber kill chain model," *Computer and Network Security Essentials*, pp. 585–602, 2018.

[7] D. Kiwia, A. Dehghantanha, K.-K. R. Choo, and J. Slaughter, "A cyber kill chain based taxonomy of banking trojans for evolutionary computational intelligence," *Journal of computational science*, vol. 27, pp. 394–409, 2018.

[8] T. Dargahi, A. Dehghantanha, P. N. Bahrami, M. Conti, G. Bianchi, and L. Benedetto, "A cyber-kill-chain based taxonomy of crypto-ransomware features," *Journal of Computer Virology and Hacking Techniques*, vol. 15, pp. 277–305, 2019.

[9] M. J. Haber, *Privileged attack vectors: building effective cyber-defense strategies to protect organizations*. Apress, 2020.

[10] E. Berrueta, D. Morato, E. Magaña, and M. Izal, "Open repository for the evaluation of ransomware detection tools," *IEEE Access*, vol. 8, pp. 65 658–65 669, 2020.

[11] N. Scaife, H. Carter, P. Traynor, and K. R. Butler, "Cryptolock (and drop it): stopping ransomware attacks on user data," in *2016 IEEE 36th international conference on distributed computing systems (ICDCS)*. IEEE, 2016, pp. 303–312.

[12] A. Liska and T. Gallo, *Ransomware: Defending against digital extortion*. "O'Reilly Media, Inc.", 2016.

[13] D. Sgandurra, L. Muñoz-González, R. Mohsen, and E. C. Lupu, "Automated dynamic analysis of ransomware: Benefits, limitations and use for detection," *arXiv preprint arXiv:1609.03020*, 2016.

[14] M. K. Berntsen, "Biological analogies in malware: Proposing a phylogenetics-inspired system for malware research," Master's thesis, University of Oslo, 2020.

[15] C. Boyton, N.-A. Le-Khac, K.-K. R. Choo, and A. Jurcut, "Forensic investigation of ransomware activities—part 2," *Cyber and Digital Forensic Investigations: A Law Enforcement Practitioner's Perspective*, pp. 79–108, 2020.

[16] N. Hassan, *Ransomware Revealed*. Springer, 2019.

[17] D. O'Brien, "Symantec 2021 Cyber Security Predictions – Looking Toward the Future — symantec-enterprise-blogs.security.com," https://symantec-enterprise-blogs.security.com/blogs/feature-stories/symantec-2021-cyber-security-predictions-looking-toward-future, 2020, [Accessed 26-Apr-2023].

[18] L. Tung, "Ransomware: Cybercriminals are adding a new twist to their demands — zdnet.com," https://www.zdnet.com/article/ransomware-cybercriminals-are-adding-a-new-twist-to-their-demands/, 2019, [Accessed 26-Apr-2023].

[19] N. K. Popli and A. Girdhar, "Behavioural analysis of recent ransomwares and prediction of future attacks by polymorphic and metamorphic ransomware," in *Computational Intelligence: Theories, Applications and Future Directions-Volume II: ICCI-2017*. Springer, 2019, pp. 65–80.

[20] P. S. Goyal, A. Kakkar, G. Vinod, and G. Joseph, "Crypto-ransomware detection using behavioural analysis," in *Reliability, Safety and Hazard Assessment for Risk-Based Technologies: Proceedings of ICRESH 2019*. Springer, 2020, pp. 239–251.

[21] S. Goyal, P. Bedi, S. Kumar, J. Kumar, and N. R. Karahroudi, "Application of deep learning in honeypot network for cloud intrusion detection," in *Proceedings of International Conference on Computational Intelligence and Data Engineering: ICCIDE 2021*. Springer, 2022, pp. 251–266.

[22] L. B. Bhagwat and B. M. Patil, "Detection of ransomware attack: A review," in *Proceeding of International Conference on Computational Science and Applications: ICCSA 2019*. Springer, 2020, pp. 15–22.

[23] F. Manavi and A. Hamzeh, "A new method for ransomware detection based on pe header using convolutional neural networks," in *2020 17th International ISC Conference on Information Security and Cryptology (ISCISC)*. IEEE, 2020, pp. 82–87.

[24] S. Kim, S. Yeom, H. Oh, D. Shin, and D. Shin, "Automatic malicious code classification system through static analysis using machine learning," *Symmetry*, vol. 13, no. 1, p. 35, 2020.

[25] S. Sharma and S. Singh, "Texture-based automated classification of ransomware," *Journal of The Institution of Engineers (India): Series B*, vol. 102, pp. 131–142, 2021.

[26] M. Kakavand, L. Arulsamy, A. Mustapha, and M. Dabbagh, "A novel crypto-ransomware family classification based on horizontal feature simplification," *Advances in Computer, Communication and Computational Sciences: Proceedings of IC4S 2019*, pp. 3–14, 2021.

[27] B. V. Reddy, G. J. Krishna, V. Ravi, and D. Dasgupta, "Machine learning and feature selection based ransomware detection using hexacodes," in *Evolution in Computational Intelligence: Frontiers in Intelligent Computing: Theory and Applications (FICTA 2020), Volume 1*. Springer, 2021, pp. 583–597.

[28] B. M. Khammas, "Ransomware detection using random forest technique," *ICT Express*, vol. 6, no. 4, pp. 325–331, 2020.

[29] H. Zhang, X. Xiao, F. Mercaldo, S. Ni, F. Martinelli, and A. K. Sangaiah, "Classification of ransomware families with machine learning based onn-gram of opcodes," *Future Generation Computer Systems*, vol. 90, pp. 211–221, 2019.

[30] D. Vidyarthi, C. Kumar, S. Rakshit, and S. Chansarkar, "Static malware analysis to identify ransomware properties," *International Journal of Computer Science Issues (IJCSI)*, vol. 16, no. 3, pp. 10–17, 2019.

[31] A. Ashraf, A. Aziz, U. Zahoora, M. Rajarajan, and A. Khan, "Ransomware analysis using feature engineering and deep neural networks," *arXiv preprint arXiv:1910.00286*, 2019.

[32] A. Ren, C. Liang, I. Hyug, S. Broh, and N. Jhanjhi, "A three-level ransomware detection and prevention mechanism," *EAI Endorsed Transactions on Energy Web*, vol. 7, no. 26, 2020.

[33] G. Ramesh and A. Menen, "Automated dynamic approach for detecting ransomware using finite-state machine," *Decision Support Systems*, vol. 138, p. 113400, 2020.

[34] K. C. Roy and Q. Chen, "Deepran: Attention-based bilstm and crf for ransomware early detection and classification," *Information Systems Frontiers*, vol. 23, pp. 299–315, 2021.

[35] F. Tang, B. Ma, J. Li, F. Zhang, J. Su, and J. Ma, "Ransomspector: An introspection-based approach to detect crypto ransomware," *Computers & Security*, vol. 97, p. 101997, 2020.

[36] A. Zimba, Z. Wang, and H. Chen, "Multi-stage crypto ransomware attacks: A new emerging cyber threat to critical infrastructure and industrial control systems," *Ict Express*, vol. 4, no. 1, pp. 14–18, 2018.

[37] D. Arivudainambi, V. K. KA, P. Visu *et al.*, "Ransomware traffic classification using deep learning models: ransomware traffic classification," *International Journal of Web Portals (IJWP)*, vol. 12, no. 1, pp. 1–11, 2020.

[38] P. B. Caliaberah, S. Armoogum, and X. Li, "An adaptive security architecture for detecting ransomware attack using open source software," in *Advances in Information and Communication: Proceedings of the 2020 Future of Information and Communication Conference (FICC), Volume 1*. Springer, 2020, pp. 618–633.

[39] G. Marín, P. Caasas, and G. Capdehourat, "Deepmal-deep learning models for malware traffic detection and classification," in *Data Science–Analytics and Applications: Proceedings of the 3rd International Data Science Conference–iDSC2020*. Springer, 2021, pp. 105–112.

[40] J. Modi, I. Traore, A. Ghaleb, K. Ganame, and S. Ahmed, "Detecting ransomware in encrypted web traffic," in *Foundations and Practice of Security: 12th International Symposium, FPS 2019, Toulouse, France, November 5–7, 2019, Revised Selected Papers 12*. Springer, 2020, pp. 345–353.

[41] M. A. Ayub, A. Continella, and A. Siraj, "An i/o request packet (irp) driven effective ransomware detection scheme using artificial neural network," in *2020 IEEE 21st International Conference on Information Reuse and Integration for Data Science (IRI)*. IEEE, 2020, pp. 319–324.

[42] S. Mehnaz, "Fine-grained anomaly detection for in depth data protection," Ph.D. dissertation, Purdue University Graduate School, 2020.

[43] A. AlSabeh, H. Safa, E. Bou-Harb, and J. Crichigno, "Exploiting ransomware paranoia for execution prevention," in *ICC 2020-2020 IEEE International Conference on Communications (ICC)*. IEEE, 2020, pp. 1–6.

[44] E. Ketzaki, P. Toupas, K. M. Giannoutakis, A. Drosou, and D. Tzovaras, "A behaviour based ransomware detection using neural network models," in *2020 10th International Conference on Advanced Computer Information Technologies (ACIT)*. IEEE, 2020, pp. 747–750.

[45] A. Arabo, R. Dijoux, T. Poulain, and G. Chevalier, "Detecting ransomware using process behavior analysis," *Procedia Computer Science*, vol. 168, pp. 289–296, 2020.

[46] J. Hwang, J. Kim, S. Lee, and K. Kim, "Two-stage ransomware detection using dynamic analysis and machine learning techniques," *Wireless Personal Communications*, vol. 112, pp. 2597–2609, 2020.

[47] Y. A. Ahmed, B. Kocer, and B. A. S. Al-rimy, "Automated analysis approach for the detection of high survivable ransomware," *KSII Transactions on Internet and Information Systems (TIIS)*, vol. 14, no. 5, pp. 2236–2257, 2020.

[48] B. Jethva, I. Traoré, A. Ghaleb, K. Ganame, and S. Ahmed, "Multilayer ransomware detection using grouped registry key operations, file entropy and file signature monitoring," *Journal of Computer Security*, vol. 28, no. 3, pp. 337–373, 2020.

[49] S. I. Bae, G. B. Lee, and E. G. Im, "Ransomware detection using machine learning algorithms," *Concurrency and Computation: Practice and Experience*, vol. 32, no. 18, p. e5422, 2020.

[50] M. Alam, S. Bhattacharya, S. Dutta, S. Sinha, D. Mukhopadhyay, and A. Chattopadhyay, "Ratafia: ransomware analysis using time and frequency informed autoencoders," in *2019 IEEE International Symposium on Hardware Oriented Security and Trust (HOST)*. IEEE, 2019, pp. 218–227.

[51] M. Alam, S. Bhattacharya, D. Mukhopadhyay, and A. Chattopadhyay, "Rapper: Ransomware prevention via performance counters," *arXiv preprint arXiv:1802.03909*, 2018.

[52] F. Cicala and E. Bertino, "Analysis of encryption key generation in modern crypto ransomware," *IEEE Transactions on Dependable and Secure Computing*, vol. 19, no. 2, pp. 1239–1253, 2020.

[53] S. Lee, M. Park, and J. Kim, "Magniber v2 ransomware decryption: Exploiting the vulnerability of a self-developed pseudo random number generator," *electronics*, vol. 10, no. 1, p. 16, 2020.

[54] Z. A. Genç, G. Lenzini, and P. Y. Ryan, "Nocry: No more secure encryption keys for cryptographic ransomware," in *Emerging Technologies for Authorization and Authentication: Second International Workshop, ETAA 2019, Luxembourg City, Luxembourg, September 27, 2019, Proceedings 2*. Springer, 2020, pp. 69–85.

[55] ——, "No random, no ransom: a key to stop cryptographic ransomware," in *Detection of Intrusions and Malware, and Vulnerability Assessment: 15th International Conference, DIMVA 2018, Saclay, France, June 28–29, 2018, Proceedings 15*. Springer, 2018, pp. 234–255.

[56] R. Rastogi, G. Agarwal, and R. Shukla, "Interactive security of ransomware with heuristic random bit generator," in *ICCCE 2020: Proceedings of the 3rd International Conference on Communications and Cyber Physical Engineering*. Springer, 2021, pp. 965–973.

[57] P. Bajpai, *Extracting Ransomware's Keys by Utilizing Memory Forensics*. Michigan State University, 2020.

[58] S. R. Davies, R. Macfarlane, and W. J. Buchanan, "Evaluation of live forensic techniques in ransomware attack mitigation," *Forensic Science International: Digital Investigation*, vol. 33, p. 300979, 2020.

[59] I.-S. Lei, S.-K. Tang, I.-K. Chao, and R. Tse, "Self-recovery service securing edge server in iot network against ransomware attack." in *IoTBDS*, 2020, pp. 399–404.

[60] S. Baek, Y. Jung, D. Mohaisen, S. Lee, and D. Nyang, "Ssd-assisted ransomware detection and data recovery techniques," *IEEE Transactions on Computers*, vol. 70, no. 10, pp. 1762–1776, 2020.

[61] M. Dorsett, "Forensic analysis of ransomware families," Ph.D. dissertation, University of South Alabama, 2020.

[62] R. Sefa, "Design and implementation of a virtual file system for hostbased moving target defence in iot devices," 2022.

[63] S. Lee, N.-s. Jho, D. Chung, Y. Kang, and M. Kim, "Rcryptect: Real-time detection of cryptographic function in the user-space filesystem," *Computers & Security*, vol. 112, p. 102512, 2022.

**Hesham Alshaikh** work as an application developer at Sadat Academy for Management Sciences (SAMS), Egypt. He received the postgraduate diploma in 2015, and the pre-masters in information systems in 2016 from the Faculty of Graduate Studies for Statistical Research (FGSSR), Cairo University, Egypt. He currently pursuing a Masters's degree in information systems at the Faculty of Graduate Studies for Statistical Research (FGSSR), Cairo University, Egypt.

**Hesham A. Hefny** Hesham A. Hefny received the B.Sc., M.Sc., and Ph.D. all in Electronics and Communication Engineering from Cairo University in 1987, 1991, and 1998 respectively. He is currently a professor and Head of the Department of Computer Science at the Faculty of Graduate Studies of Statistical Research (FGSSR), Cairo University. Prof. Hefny has authored more than 200 papers in international conferences, journals, and book chapters. His major research interests include: computational intelligence (Neural networks – Fuzzy systems – Genetic algorithms – Swarm intelligence), Data mining, and Uncertain Decision Making. He is a member in the following professional societies: IEEE Computer, IEEE Computational Intelligence, IEEE System, and Man and Cybernetics.

**Nagy Ramadan Darwish** received his PhD. in Information Systems from the Faculty of Computers and Information, Cairo University, Egypt. He is currently a Professor and Head of the Department of Information Systems and Technology, Faculty of Graduate Studies for Statistical Research (FGSSR), Cairo University. He is a reviewer in many national and international conferences and journals, such as IJCSIS, IJACSA, IJARAI, IJST, KJS, and IJCIT. He is an editorial board member of Circulation in Computer Science (CCS Archive). He published about 90 papers in International Journals, conferences, and book chapters. He is a Consultant in Software Project Management, Software Quality, Business Information Systems, and Quality of Education.