Research on Android Ransomware Protection Technology

Jian Wei Hu¹, Yu Zhang¹,²* and Yan Peng Cui¹
¹School of Cyber Engineering, Xidian Univ., Xi’an 710075, China
²E-mail: 2992740225@qq.com

Abstract. With the popularity of smart devices such as smartphones and pads, the attack of Android ransomware is becoming increasingly serious. Compared with other malicious software, ransomware is widely favored by hackers because of hard restoration and the directness of obtaining benefits, which also brings serious spiritual and property damage to users. To protect our smart devices from ransomware and reduce threats and losses, researchers conduct a lot of research on Android ransomware and propose many practical detection schemes. This paper first summarizes the characteristics of Android ransomware, and then summarizes the existing research work on detecting and safeguarding against ransomware on the Android platform and makes a comprehensive analysis and comparison on them. Finally, it points out that the remaining problems of these solutions, puts forward corresponding suggestions and proposes future research directions.

1. Introduction
Ransomware is a very typical piece of malware that seeks a ransom from the user for profit by locking the user's device or encrypting the data and resources in the device. Because ransomware generally uses strong encryption to encrypt user data, its consequences are almost irreversible, which brings great privacy and security threats to users and enterprises. The earliest ransomware can be traced back to the end of the 1980s [1], but only in recent years did it begin to erupt on a large scale. According to statistics, the number of ransomwares in 2013 increased by more than 500% year-on-year, gradually becoming one of the most compelling malware threats [2]. The number of ransomwares soared to 1.7 million in 2016, known as "the year of ransomware". According to Symantec's data report [3], the number of ransomware variants increased by 46% in 2017, and the outbreak spread across the globe. In 2018, although the ransomware activity has generally decreased, the ransomware goal has begun to shift to enterprises, and the number of enterprise infections has increased by 12%, and the harm caused still cannot be underestimated [4]. Ransomware originally targeted at the Windows platform, but with the rapid development of Android smartphones, its increasingly open application market and relatively relaxed review mechanism gave ransomware an opportunity. The threats are also increasing day by day. Kaspersky labs found 1113 samples of ransomware for Android devices in the first quarter of 2015 and mobile ransomware surged again in 2016, four times that of 2015. Symantec's 2019 Internet Security Threat Report [4] shows that the proportion of ransomware infections on mobile devices in 2018 was one third higher than in 2017, and the incidence of ransomware infections on mobile phones has increased rapidly.

As mobile ransomware is getting more and more rampant, the prevention and detection of ransomware on the Android platform has been paid more and more attention, and many related research results have been proposed by researchers. However, there is currently no systematic work to
review the research on prevention and detection technology of ransomware on Android platform, more relevant are the studies of Monika [5] and Bander [6] and Desai [7]. However, the research on Android ransomware protection technology in these three articles is not very comprehensive. At present, more progress has been made in the prevention and detection technology of Android ransomware, which needs to be summarized again. Therefore, this article searches all the protection technologies for Android ransomware in the IEEE Explorer, ACM, Springer and Elsevier database and analyses the advantages and disadvantages of different technologies and discusses future research directions.

2. Analysis of Android ransomware

Ransomware is a piece of malware whose fundamental purpose is to force users to pay a ransom to regain access to the system. Ransomware can be divided into two categories: lock screen ransomware and encrypted ransomware. Lock screen ransomware, as its name implies, can lock user's mobile device and then display a payment message on the screen. Encrypted ransomware is to encrypt data in the device to achieve the purpose that users cannot access their devices.

2.1. Typical Android ransomware

In 2017, the infamous WannaCry ransomware broke out globally, mainly targeting Windows platforms with a large number of users. Affected by this, WannaLocker ransomware attacked the Android platform in a similar way, disguising itself as a game plugin, tricking users into downloading and running. This pseudo-plugin encodes the data in the external storage and encrypts these files using the AES encryption algorithm. When encryption is complete, a ransom note will appear on the screen.

Another notorious ransomware was DoubleLocker. DoubleLocker spread by impersonating Adobe Flash updates to attacked websites. It was the first ransomware to abuse Android accessibility, which is usually used to steal bank credentials by Android banking Trojans.

Another special ransomware is called LeakerLocker. Unlike the usual ransomware which encrypts files, LeakerLocker threatens the personal data stored in the user's mobile phone. After hacking into the device, LeakerLocker will collect as much private information as possible, and then display a ransom notice stating that all personal data has been copied to the attacker's cloud, and these information will be sent to the ransom unless ransom is paid.

2.2. The features of Android ransomware

2.2.1. Lock screen. Lock screen Ransomware locks smartphone screen by gaining administrative rights. There are two main ways to lock screen. One of them is setting a special full screen window by using WindowManager.LayoutParams to control the flags property. The other is to set the system lock screen password by activating the device manager. As a result, the user cannot enter the system operation interface [8] [9]. This type of ransomware is well-targeted and often targets specific design vulnerabilities.

2.2.2. File encryption. Encryption ransomware usually uses a strong encryption algorithm to encrypt files. So once a file is encrypted, it is almost impossible to recover it without a key. The encryption key used for encryption is usually hard-coded in a binary file or obtained by interacting with C&C server.

2.2.3. Communication with C&C server. After successful installation, most Android ransomware will send reports to the C&C server. Modern ransomware mainly obtains keys through the C&C server. Some ransomwares will establish a permanent communication channel with the C&C server, which is used to constantly monitor and execute commands sent by the ransomware attacker. In addition to displaying warning messages for ransom, these commands include sending emails, accessing messages, turning off mobile data, tracking users' GPS information and so on.
2.2.4. The permission. Compared with other malwares, ransomware has no obvious malicious behaviors. In contrast, ransomware is very similar to benign software in terms of opening pop-up windows, file encryption and so on. ACCESS_NETWORK_STATE, READ_PHONE_STATE and INTERNET are the permissions generally required by ransomware and malware, because these permissions are necessary for network communication with the server. Considering the behavior specific to ransomware, it is more inclined to request permissions related to the device, such as WAKE_LOCK, DISABLE_KEYGUARD, SYSTEM_ALERT_WINDOW, KILLBACKGROUND_PROCESS, GET_TASK, etc. The ransomware will display a warning ransom message after the smartphone restarts and hijacking is the main task of locking the phone.

2.2.5. The ransomware warning message. Ransomware needs to display a warning message to the user, which is used to display the payment method, the amount of ransom, the payment address, the payment period and so on. Most ransomware requires payment of ransom through cryptocurrencies such as Bitcoin to achieve the effect of hiding the identity of the attacker. In addition, some ransomwares display warning messages that contain intimidating content, such as Locker ransomware, which claims that it must pay a fine because they found the pornographic pictures on the user's device, otherwise it will report a warning message to the FBI.

3. Android ransomware protection technology

There are many research results on ransomware prevention and detection on the Android platform. According to the differences of detection methods, this article classifies them into the following categories: program analysis methods, formal methods, machine learning and based on user intent.

3.1. Based on program analysis

Program analysis methods are generally divided into static analysis, dynamic analysis and hybrid analysis. Static analysis is the analysis of program source code using an automated static analysis tool (such as IDA Pro) when the program is not running. When analyzing an Android executable file, you need to reverse the executable program and find malicious code and resources by analyzing the files such as suspicious strings, suspicious images, and suspicious opcode sequences. Dynamic analysis is to find a pattern or behaviors similar to a malicious program by monitoring the operation of the program while the program is running. Hybrid analysis is a combination of static analysis and dynamic analysis with higher detection accuracy.

3.1.1. Based on static analysis. Wang [10] et al. proposed a real-time detection method, RansomGuard, based on a static analysis method. This method extracts typical characteristics of ransomware from a large number of samples, such as permissions and threat text, and then uses modular rule induction learning algorithms to obtain classification rules. The author uses 2721 ransomwares and 5396 benign samples as the data set, with an accuracy rate of up to 95.3% and a false alarm rate of only 0.6%. However, because that RansomGuard is based on static analysis, the use of code encryption, reflection and other obfuscation techniques will make it impossible to extract certain features, resulting in reduced accuracy.

Maiorca et al. [11] extract the API package to detect the Android ransomware by static method, and build a model named R-PackDroid based on this. This model does not rely on prior knowledge of ransomware encryption, but only depends on the API package and its calling frequency in the application and its detection accuracy is higher than HelDroid, but its anti-aliasing ability is poor and the false alarm rate is high.

Kanwal et al. [12] extend the static analysis method. In addition to using static analysis methods to detect ransomware threat information from text and images, it also provides hardware interaction. In the case of users suffering from lock screen ransomware, users can get the right of the screen control by using the broadcast receiver to press the power button multiple times. However, the author has only used a specific application for testing, so its versatility and performance are yet to be tested.
3.1.2. Based on dynamic analysis. Song [13] et al. propose a dynamic analysis method, which aims to monitor the process and the specific directory to detect abnormal behaviors through the use of CPU, memory and I/O. Since this method can be implemented in the Android source code, the detection speed can be very fast but cannot detect lock-type ransomware. Besides, because this method only detects abnormal behavior in CPU and I/O usage, simulated attacks can circumvent this detection.

Hong et al. propose an Android APP named Sdguard for ransomware detection [14]. The application is based on the fine-grained permission control of the Linux DAC mechanism and can detect encrypted and locked ransomware. When an external application operates on a file, Sdguard first checks its permissions according to the access control list, dumps all I/O operations to a log file, and then detects malicious behaviors by analyzing the log file. At the same time, Sdguard detects the Android system's active stack, if an application is found to be on the top stack for a long time, it is regarded as a locked ransomware and kills the process, but unfortunately the technology has not been tested in practice.

3.1.3. Based on hybrid analysis. The first method for Android ransomware detection, HelDroid was proposed in 2015[15]. The tool uses a hybrid analysis method consisting of a threat text detector based on the NLP supervised classifier, an encryption detector that tracks the encrypted stream of the file using static stain analysis, and a heuristic lock detector based on lightweight simulation technology. The disadvantage of the tool is that it relies heavily on text detectors. In addition, because this method relies heavily on language dictionaries, different language threat phrases need to be trained, and other non-English speakers must switch to other languages. From a performance perspective, the tool identified 375 ransomwares in 443 samples, failed to detect 11 ransomwares due to the use of unsupported languages, and misreported 9 out of 12,842 samples, the false alarm rate is about 0.07%.

In 2017, Zheng et al. further expanded the system and named it GreatEatlon [16]. It improves the encryption detector and text detector and adds a module for the detection of device management API abuse. It can detect threat texts ransomware through images. In addition, the system also uses a supervised learning classification algorithm for lightweight pre-filtering to distinguish benign and suspicious software. However, the system still does not overcome the weakness of over-reliance on text detectors. And if ransomware stores the threat text in multiple images, the image detector will fail.

Yang [17] et al. pointed out that static analysis can specifically analyze the Android system's access permissions, API call sequences, executable programs disguised as resource files, and verifying signatures. Dynamic analysis can be performed from sensitive paths, malicious domain names, malicious charges, and permissions bypass. However, the method is still at the proposal level and has not been implemented, so its effectiveness cannot be proven.

In 2017, Gharib et al. [18] proposed a framework named DNA-Droid for real-time detection of Android ransomware. The static analysis module analyzes from four angles: text classification, image classification, API calls, and permission permissions. The dynamic analysis module detects API call sequence by running in the sandbox, and compares it with ransomware API call sequence with a cycle of five minutes. DNA-Droid has 98.1% accuracy and a false alarm rate of less than 1.5%. However, an obvious disadvantage of the method is that severely confusing ransomware can bypass static analysis detection and thus avoid dynamic analysis.

Due to the high accuracy of the hybrid static dynamic detection method, Ferrante et al. [19] also use this method for ransomware detection in 2018. The static analysis of their method is based on the frequency of opcodes, while the dynamic analysis is based on the CPU, memory, and network usage and system call information. The experiment shows that the method can detect ransomware with 100% accuracy.

Abdulrahman et al. [20] proposed a detection scheme based on structural similarity—RanDroid. This solution uses static analysis technology to extract string text and images from the parsed APK file, and uses dynamic analysis technology to capture application activity. If a threat text or the screen locked is detected, the text is extracted and the image is captured. Then the extracted text and image are compared with the text and image of the ransomware stored in the database. If the threshold is
exceeded, it is considered as ransomware and an alarm is processed. The author experimented with a data set consisting of 100 ransomware and 200 benign software, and the results reached 91% accuracy. But because the system cannot recognize many languages other than English, the false negative rate is high.

3.2. Based on formal method
In computer science, especially software engineering and hardware engineering, formal methods are a collection of symbols and techniques based on logical calculus, formal language, automata theory, program semantics, type systems and other theoretical foundations to describe and analyze computer systems [21]. It is a special mathematical-based technique used for the specification, development, and verification of software and hardware systems. This method uses a formal specification language to define characteristics of the system. The language is very special and uses a unique syntax that includes object, relation and rule components that can verify the correctness of each feature [22].

It describes a method for detecting Android ransomware based on a formal method in [23]. This method can detect ransomware and identify malicious code parts in the ransomware. This method mainly identifies ransomware by Java bytecode and is a completely static method. Detecting Android ransomware by bytecode instead of directly on the source code can be independent of the source code programming language and has many advantages such as no need for decompilation, easier analysis for the underlying code and anti-obfuscation. The authors evaluated 2,477 sample sets consisting of ransomware and benign software. The experiment shows that the method can achieve a 99% recall rate. In addition, the method does not need to be decompiled because it is targeted at Java bytecode instead of source code. However, the main weakness of the method is that it requires manual operation to construct logic rules. Cimitile et al. [24] expanded on this basis of it later, and proposed a model-based mobile ransomware detection tool—Talos. But it has the same disadvantage.

3.3. Based on machine learning
Machine learning refers to the scientific study of computer systems that performs algorithms for specific tasks and statistical models relying on patterns and reasoning [25]. The learning process needs to be given some training data. It can find some pattern in these training data so that it can make better decisions or predictions given any examples in the future. With the advent of the Internet of Things era, the subject of artificial intelligence is developing rapidly, and machine learning also plays an important role in the field of ransomware detection.

Karimi et al. [26] proposed a method of converting an executable instruction sequence into a grayscale image, and then use LDA to select the best opcode from the image as a feature to distinguish ransomware from benign software. LDA is a method used for statistics, pattern recognition, and machine learning to find linear combinations of features in order to separate two or more types of objects. The author tested 280 sample sets and the method can achieve 97.3% accuracy.

Scalas [27] [28] et al. proposed a method that only relies on system API call information for detecting Android ransomware using machine learning—R-PackDroid. Detection is performed in three steps: preprocessing, feature extraction, and classification. The main idea of this method is to classify the ransomware by statically analyzing the Dalvik bytecode of the Android SDK API package. Because R-PackDroid relies entirely on static analysis, it is subject to some inherent limitations. For example, it is impossible to analyze ransomware that loads signatures or encryption classes at runtime, and it can only detect new ones using known training samples.

Alsoghyer etc. [29] also proposed an Android ransomware detection system named API-RDS based on the API. The system also only pays attention to the call information of the API software package and uses it as an indicator for detecting ransomware. Experiments show that when using random forest classification, its accuracy is as high as 97%. Moreover, API-RDS can not only distinguish ransomware from benign software, but also detect unknown ransomware with a high accuracy of 96.5%. Besides, author also compared it with R-PackDroid [27], and the results show that API-RDS performs better. The author's other important contribution is to develop a reliable
ransomware data set.

Considering that advanced bytecode and XML permission files are easily affected by obfuscation techniques, Lachtar [30] et al. evaluated the effectiveness of using native instructions to detect ransomware and proved that using native instructions can detect ransomware with high accuracy based on different machine learning models. The author builds a dictionary containing all opcodes extracted from 15,126 Android apps and evaluates the impact of choosing different opcodes as features on the model. Finally, experiments show that the detection accuracy of 99.87% can be achieved by the random forest algorithm.

As a new research area in the field of machine learning, deep learning has achieved very significant results in speech and image recognition due to its unique advantages. Bibi [31] and others propose a malware detection model based on deep learning for ransomware detection on the Android platform. The model first uses 8 different feature selection algorithms for feature selection and compares the results of various feature filtering techniques. It uses simple voting methods to select 19 important features, and then uses the LSTM algorithm for experimental evaluation. Accuracy and 5% false positive rate.

With the increasing number of lock-type ransomware, it also poses a huge threat to users' property and privacy. Liu et al. [32] systematically analyzed lock-type ransomware in Chinese social networks and proposed a lightweight and automated lock-type ransomware detection method. The author first disassembles lock-type ransomware to extract 6 types of features of display text and background operation, and finally uses 4 machine learning algorithms to detection. Experimental results show that the method has very good detection effect, and the accuracy can reach 99.98%. However, there are only 301 ransomwares in the sample set, so the accuracy of the results needs to be proved in detail.

3.4. Based on user intent
Considering that normal applications will remind users when performing sensitive operations, and ransomware must run secretly in the background or display deceptive windows to hide themselves when performing encryption operations, Chen et al. [33] proposed a new real-time detection system—RansomProber from the perspective of user intent. The system analyzes the user interface widgets of the relevant Activity and the coordinates of the user's finger movements and RansomProber can infer whether the file encryption operation comes from the user, thereby detecting encrypted ransomware. The experimental results show that RansomProber can detect encrypted ransomware with high accuracy—99%, and the false positive rate is less than 1%. In addition, file loss and memory consumption at runtime are also very small. However, because the system determines whether it is ransomware by detecting whether the UI indicators are met at the same time, if the ransomware repackages those applications that can be encrypted or compressed, it will break through this system.

Aiming at the shortcomings of RansomProber, Ko et al. [34] improved it and proposed a new method that can detect unknown ransomware in real time combining with the method based on the white list to detect unknown ransomware proposed in literature [35]. This method can theoretically improve the accuracy of detecting unknown ransomware, but it is not experimentally verified so the actual effect is unknown.

4. Comparison of ransomware protection technologies
In order to compare the existing Android ransomware protection technologies, this article proposes the following evaluation indicators to evaluate the level of system performance in order to overcome potential attacks and security threats.

4.1. Evaluation indicators

4.1.1. Data set. The source and size of the data set are a very important part of the detection scheme, and it often has a very large impact on the detection results. The more reliable the source of the data set, the larger the size, the more reliable the corresponding detection scheme is.
4.1.2. **Accuracy and false positive rate.** In the process of detecting malware, the percentage of detected malicious samples in all samples the accuracy rate, and the percentage of classified benign samples detected as malware samples is the false positive rate. The accuracy rate reflects the system's ability to detect malicious samples, so the higher the better. A system with a high false positive rate will bring a very poor user experience, so the lower the false positive rate, the better.

4.1.3. **Detection speed.** Since the encryption results caused by ransomware are difficult to recover, the slower the detection speed, the more files the ransomware encrypts, and the greater the user loss.

4.1.4. **Performance Consumption.** Unlike PC, the memory and CPU of mobile phones are very small, but people have high requirements for the response speed of mobile phones. Therefore, if the performance consumption of the detection system is too high and it affects the user experience effect.

4.1.5. **Anti-aliasing.** In order to avoid more and more complete detection systems, ransomware has adopted a series of obfuscation methods such as deformation and packing to avoid detection by the detection system. If the anti-aliasing ability of the detection system is poor, as time goes by, there will be more and more ransomware variants, and the system will eventually be eliminated.

The above indicators are used to evaluate the performance of existing solutions. In order to get a better evaluation, this article further divides some of these indicators into three levels: high, middle, and low to measure the extent to which the existing scheme meets each demand. The comparison results are shown in Table 1 below:

**Table 1.** Comparison of the schemes.

| Category          | Time  | Data set size | Precise | FPR  | Detection speed | Performance consumption | Anti-aliasing |
|-------------------|-------|---------------|---------|------|-----------------|--------------------------|--------------|
| R-PackDroid [11]  | 2017  | 2045R+5560M   | 95.3%   | 0.6% | 8s              | low                      | low          |
| Kanwal et al. [12]| 2017  | +40888B       | -       | -    | -               | low                      | low          |
| RansomGuard [10]  | 2018  | 2721R+5396B   | 95.3%   | 0.6% | 8s              | low                      | low          |
| Song et al. [13]  | 2018  | -             | -       | -    | -               | -                        | -            |
| Sdguard [14]      | 2017  | -             | -       | -    | -               | -                        | -            |
| HelDroid [15]     | 2015  | M+81044B,75R+153982M | 84.7% | 0.07% | 4s              | low                      | low          |
| GreatEatlon [16]  | 2017  | +1239B        | 98.0%   | 0%   | 4s              | low                      | middle       |
| Yang et al. [17]  | 2015  | -             | -       | -    | -               | -                        | -            |
| DNA-Droid [18]    | 2017  | 1928R+2500B   | 98.1%   | 0.5% | 5m              | high                     | middle       |
| Ferrante et al. [19]| 2018 | 672R+2386B    | 98%     | 0%   | -               | high                     | high         |
| RanDroid [20]     | 2018  | 950R+500B     | 91%     | -    | -               | high                     | high         |
| Mercaldo et al. [23]| 2016 | +600B         | 99.6%   | 0%   | -               | -                        | high         |
| Talos [24]        | 2018  | +1500B        | 95.5%   | 0.5% | 1s-306s        | -                        | high         |
| Kaimi et al. [26] | 2017  | 300R+30B      | 97.3%   | -    | -               | -                        | low          |
| R-PackDroid [27]  | 2018  | R+30839B      | 97%     | 1%   | 4s              | low                      | middle       |
| API-RDS [29]      | 2019  | 2559R+500B    | 97%     | 2%   | -               | -                        | low          |
| Lachter et al. [30]| 2019 | B             | 99.8%   | -    | -               | -                        | high         |
| Bibi et al. [31]  | 2019  | +104857B      | 97.1%   | 1.8% | -               | -                        | low          |
| Liu et al. [32]   | 2019  | 301R+15751B   | 99.9%   | 0%   | 10.91s         | low                      | middle       |
4.2. Comparative results analysis

According to the developed indicators and the comparison results in the chart, it is found that the method based on static analysis has very low performance consumption and very fast detection speed. It is a very lightweight method. However, it has poor anti-aliasing and deformation capabilities. The accuracy rate is relatively low, and the threat identification ability against unknown and new varieties of ransomware is also relatively poor. In the method based on dynamic analysis, some scheme authors only perform theoretical proof without experimental verification, while some scheme authors only use one ransomware to verify, so their effects need to be studied. The method based on hybrid analysis combines the advantages of static analysis and dynamic analysis. The accuracy of it is generally high and the anti-aliasing ability is strong, But the performance consumption is also relatively large.

The technical accuracy and false positive rate based on the formal method are very high, and because the logic rules are formulated at the byte code level, the anti-aliasing ability is strong, and the performance consumption is low, but because the logic rules need to be manually formulated, the manual operation is increased.

The method based on machine learning is a hot topic in recent years. It also has the advantages of high accuracy, fast detection speed and low performance consumption. However, its anti-aliasing ability is stronger than the static detection method, but it still needs to be improved.

Considering the concealment of ransomware encryption and the feature of pinning the custom interface when locking the phone, a method based on user intent was proposed. This method not only has high accuracy and low false alarm rate, but also has fast detection speed and low performance consumption. It is a lightweight solution that is very suitable for checking new applications in daily life, but it also has limitations. Since the detection of the UI is one of the important indicators for detecting ransomware, when the ransomware is encrypted, the user clicking the irrelevant button may cause the indicator fail.

5. Future and outlook

According to the analysis and comparison of the current Android ransomware protection technology in the previous section, it is found that the existing protection schemes still have shortcomings. The author proposes the following suggestions against these shortcomings:

First, it is known from the table that detection methods based on user intent and machine learning have higher performance in terms of accuracy, false alarm rate, and detection speed, so we can continue to start from these two aspects to improve their anti-aliasing capabilities in order to detect more types of ransomware and adapt to attacks of ransomware variants. Second, most of the existing android ransomware detection solutions are from the perspective of after ransomware infecting into mobile phones. The goal is to detect the ransomware with the highest accuracy and minimize the loss of users. But there are few solutions from the perspective of before ransomware infecting into mobile. Therefore, we can consider detecting from the spread of ransomware software, such as detection of mailboxes and download paths to reduce the probability of ransomware infecting the mobile phone and avoid the loss caused by ransomware. Third, HTTP traffic detection is one of the most common ways to detect malware on the Android platform. Ransomware is a special kind of malware, and many ransomwares have a process of interacting with C & C server traffic. Therefore, you can try to detect the ransomware by detecting the traffic between the ransomware and the C & C server. Forth, most of the research on ransomware protection is from the perspective of detection and many ransomware detection and protection systems have been proposed. But relatively little work has been done on ransomware traceability. This is due to the anonymity of Bitcoin, it is very difficult to track down ransomware attackers. However, if the attacker can be found by using the network decoy environment or other methods to assist tracing technology, then this will greatly prevent cybercrimes by using...
ransomware. Last but not least, because the earliest and widest range of ransomware attacks is usually the Windows platform, ransomware detection technology for the Windows platform is relatively mature, such as based on decoy files, file-based, model-based and so on. We can try to apply the technology in Windows platform to Android platform and find a more complete ransomware protection solution for Android platform.

6. Conclusion
Ransomware is one of the most prominent malware threats and is closely watched for its irreversible consequences. With the popularity of smart devices, the target of ransomware attack has spread from the PC to the Android mobile platform, causing very serious loss of property and privacy to users. In this paper, the detection and protection technology of ransomware for Android platform is investigated, and various protection technologies are comprehensively analyzed and compared. Finally, future research directions are discussed based on the analysis results.

References
[1] Kirda A, Arshad S, Mulliner C, et al. UNVEIL: A large-scale, automated approach to detecting ransomware (keynote)[C]// Proc of the 24th Int’l Conf on Software Analysis, Evolution and Reengineering (SANER), 2017: 1-1.
[2] Kharraz A, Robertson W, Balzarotti D, et al. Cutting the gordian knot: A look under the hood of ransomware attacks[C]//Proc of the 12th Int’l Conf on Detection of Intrusions and Malware, and Vulnerability Adressment, 2015:3-24.
[3] 2018 Internet Security Threat Report [EB/OL]. [2019-06-02]. https://www.symantec.com/content/dam/symantec/docs/reports/istr-23-2018-en.pdf
[4] 2019 Internet Security Threat Report [EB/OL]. [2019-06-02]. https://www.symantec.com/content/dam/symantec/docs/reports/istr-24-2019-en.pdf
[5] Monika, Zavarsky P, Lindskog D. Experimental analysis of ransomware on windows and android platforms: evolution and characterization[J]. Procedia Computer Science, 2016, 94:465-472.
[6] Bander A S A, Mohd A M, Syed Z M S. Ransomware threat success factors, taxonomy, and countermeasures: A survey and research directions[J], Computers & Security, 2018, 74:144-166.
[7] Desai U. A survey on Android ransomware and its detection methods[J]. International Research Journal of Engineering and Technology. 2019,06(03):3081-3087
[8] Mobile phone lock screen extortion first seen in China [EB/OL]. [2019-06-05]. http://blogs.360.cn/post/analysis_of_ransomware.html (in Chinese)
[9] Android Marshmallow will not go soft on mobile ransomware [EB/OL]. [2019-06-06]. https://www.symantec.com/connect/blogs/android-marshmallow-will-not-go-soft-mobile-ransomware.
[10] Wang Chi-heng, Chen Jing, Chen Xiang-yun, et al. An android ransomware detection method based on evidence chain generation[J]. Chinese Journal of Computers, 2018,41(10):2344-2358. (in Chinese)
[11] Maiorca D, Mercaldo F, Giacinto G, et al. R-PackDroid: API package-based characterization and detection of mobile ransomware[C]//Proc of the 32nd ACM SIGAPP Symposium on Applied Computing, 2017:1718-1723.
[12] Kanwal M, Thakur S, Lashkari R. An app based on static analysis for android ransomware[C]// Proc of the 8th ICCCNT, 2017:1-6.
[13] Song S, Kim B, Lee S. The effective ransomware prevention technique using process monitoring on android platform[J]. Mobile Information Systems, 2016:1-9.
[14] Hong Shuang-xi, Liu Chuan-chang, Ren Bing-fei, et al. Sguard:An android application implementing privacy protection and ransomware detection[C]// Proc of the 15th Annual International Conference on Mobile Systems, Applications, and Services, 2017:149-149.
[15] Andronio N, Zanero S, Maggi F. HelDroid: Dissecting and detecting mobile ransomware[C]// Proc of the 18th International Symposium on Recent Advances in Intrusion Detection, 2015:382-404.
[16] Zheng Cheng-yu, Dellarocca N, Andronio N, et al. GreatEatlon: Fast, static detection of mobile ransomware[C]// Proc of the 12th Security and Privacy in Communication Networks. 2016, 198:617-636.
[17] Yang Tian-da, Yang Yu, Qian Kai, et al. Automated detection and analysis for android ransomware[C]// Proc of the 17th Int’l Conf on Embedded Software and Systems, 2015:1338-1343.
[18] Gharib A, Ghorbani A. DNA-Droid: A real-time android ransomware detection framework[C]// Proc of the 11th Int’l Conf on Network and System Security, 2017:242-258.
[19] Ferrante A, Malek M, Martinelli F, et al. Extinguishing ransomware - a hybrid approach to android ransomware detection[C]// Proc of the 10th International Symposium on Foundations and Practice of Security, 2017:382.
[20] Abdulrahman A, Alshehri A, Alshahrani H, et al. RanDroid: Structural similarity approach for detecting ransomware applications in android platform[C]// Proc of the 13th Int’l Conf on Electro/Information Technology, 2018:892-897.
[21] He Fei, Zhang Li-jun. The preface of software formal verification topics[J]. Journal of Software, 2019,30(07):1901-1902. (in Chinese)
[22] Formal methods[EB/OL]. [2019-10-11]. https://en.wikipedia.org/wiki/Formal_methods.
[23] Mercaldo F, Nardone V, Santone A, et al. Ransomware steals your phone. Formal methods rescue it[C]// Proc of the 36th Int’l Conf on Formal Techniques for Distributed Objects, Components, and Systems, 2016:212-221.
[24] Cimitile A, Mercaldo F, Nardone V, et al. Talos: no more ransomware victims with formal methods[J]. International Journal of Information Security, 2018,17(06):719-738.
[25] Machine learning[EB/OL]. [2019-10-11]. https://en.wikipedia.org/wiki/Machine_learning.
[26] Karimi A, Moattar M H. Android ransomware detection using reduced opcode sequence and image similarity[C]//Proc of the 7th of Int’l Conf on Computer and Knowledge Engineering (ICCKE), 2017:229-234.
[27] R-PackDroid: practical on-device detection of android ransomware[EB/OL]. [2019-6-20]. https://arxiv.org/pdf/1805.09563v1.pdf
[28] Scalas M, Maiorca D, Mercaldo F, et al. On the effectiveness of system API-related information for Android ransomware detection
[29] Alsoghyer S, Almomani I. Ransomware detection system for Android applications, 2019
[30] N. Lachtar, D. Ibdah and A. Bacha, "The Case for Native Instructions in the Detection of Mobile Ransomware,” in IEEE Letters of the Computer Society, vol. 2, no. 2, pp. 16-19, 1 June 2019.
[31] Bibi I, Akhunzada A, Malik J, et al. An effective android ransomware detection through multi-factor feature filtration and recurrent neural network[C].
[32] Su Dan, Liu Ji-qiang, Wang Xiao-yang, et al. Detecting android locker-ransomware on Chinese social networks[J]. IEEE Access, 2019,7:20381-20393.
[33] Chen Jing, Wang Chi-heng, Zhao Zi-ming, et al. Uncovering the face of android ransomware: characterization and real-time detection[J]. IEEE Transactions on Information Forensics and Security, 2018,13(5):1286-1300.
[34] Ko J, Jo J, Kim D, et al. Real time android ransomware detection by analyzed android applications[C]// Proc of the 3th Int’l Conf on Electronics, Information and Communications (ICEIC), 2019:1-5.
[35] Kim D, Choi G, Lee J. White list-based ransomware real-time detection and prevention for user device protection[C]// Proc of the 32th Int’l Conf on Consumer Electronics, 2018.