View metadata, citation and similar papers at core.ac.uk
brought to you by CORE
provided by Aston Publications Explorer

*Computers, Materials & Continua* CMC, vol.64, no.2, pp.1051-1061, 2020

# An Immunization Scheme for Ransomware

**Jingping Song[1], Qingyu Meng[1], Chenke Luo[2], Nitin Naik[3] and Jian Xu[1, *]**

**Abstract:** In recent years, as the popularity of anonymous currencies such as Bitcoin has made the tracking of ransomware attackers more difficult, the amount of ransomware attacks against personal computers and enterprise production servers is increasing rapidly. The ransomware has a wide range of influence and spreads all over the world. It is affecting many industries including internet, education, medical care, traditional industry, etc. This paper uses the idea of virus immunity to design an immunization solution for ransomware viruses to solve the problems of traditional ransomware defense methods (such as anti-virus software, firewalls, etc.), which cannot meet the requirements of rapid detection and immediate prevention of new outbreaks attacks. Our scheme includes two parts: server and client. The server provides an immune configuration file and configuration file management functions, including a configuration file module, a cryptography algorithm module, and a display module. The client obtains the immunization configuration file from server in real time, and performs the corresponding operations according to the configuration file to make the computer have an immune function for a specific ransomware, including an update module, a configuration file module, a cryptography algorithm module, a control module, and a log module. This scheme controls mutexes, services, files and registries respectively, to destroy the triggering conditions of the virus and finally achieve the purpose of immunizing a computer from a specific ransomware.

**Keywords:** Malware, ransomware, malware immunization.

## 1 Introduction

With the popularity of anonymous virtual currencies such as Bitcoin [Biryukov and Tikhomirov (2019); Sun, Bin, Jiang et al. (2019); Wang, Li, Kuang et al. (2019)], the ransomware is increasing rampantly [Sai, Buckley and Le Gear (2019); Al-Hawawreh, den Hartog and Sitnikova (2019); Homayoun, Dehghantanha, Ahmadzadeh et al. (2018); Li, Xu, Xian et al. (2019)]. The way that the extortionist collects the ransom is easily tracked before the emergence of anonymous virtual currencies, causing the extortionist to

[1] Software College, Northeastern University, Shenyang, 110169, China.
[2] School of Cyber Science and Engineering, Wuhan University, Wuhan, 430072, China.
[3] School of Computing, University of Portsmouth, Portsmouth, PO1 2DT, UK.
[*] Corresponding Author: Jian Xu. Email: xuj@mail.neu.edu.cn.
Received: 12 March 2020; Accepted: 14 April 2020.

be exposed and eventually arrested. The extortionist uses anonymous virtual currencies such as Bitcoin to collect the ransom, which makes it difficult to trace back to the source of the extortion. Because the value of the encrypted files is higher than the ransom, the victim has to recover the documents by paying the ransom. This has led to the result that unscrupulous hackers are more frequently attacking business enterprises and individuals by ransomwares.

At present, ransomware is a major threat to computer security [Zhu, Yang, Xiong et al. (2018); Xu, Wang, Wu et al. (2020)]. Unscrupulous hackers use ransomwares to attack other people's computers, extort them and steal important information, which may cause a huge loss to victims. For example, the popular WannaCry Bitcoin ransomware [Akbanov, Vassilakis and Logothetis (2019)] caused great damages to business enterprises, administration institutes, and education industry; The NotPetya ransomware [Farral (2017)] caused more than $90 million in loss to Nuance, which is the world's largest speech recognition company.

As a type of malware, ransomware is highly destructive, and it is difficult to recover the corrupted documents unless the ransom is paid to the extortionist. Once an enterprise's server is infected with a ransomware, the important files required for providing its services may be encrypted, the business based on the server will be paralyzed, and the enterprise may suffer from a great loss. After the personal computer is infected with ransomware, important private files such as photos, articles, archives, and papers may also be encrypted. Although the value of loss is not necessarily as great as the one of a business, those encrypted files will still inevitably cause some inconvenience. Some ransomware, such as NotPetya, often fail to recover the corrupted documents even if the victims have paid the ransom, resulting in irreversible losses. It shows that for both individuals and businesses, the damage of ransomware is very huge. There are common methods for preventing ransomwares including anti-virus softwares, firewalls, etc., but these methods are not able to meet the need of rapid detection and immediate prevention of rapid outbreaks of ransomwares. Therefore, it is urgent to find an effective method to keep computers safe from ransomware.

Many researchers have carried out research work on malware detection and got a lot of research results. Kolbitsch et al. [Kolbitsch, Comparetti, Kruegel et al. (2009)] proposed an efficient malware detection model, which is more efficient than traditional malware detection methods, but it still has disadvantages that slow down the efficiency of the operating system. Santos et al. [Santos, Brezo, Nieves et al. (2010)] proposed an Opcode-Sequence-Based malware detection method. Firdausi et al. proposed a malware detection method based on behavior detection combined with machine learning [Firdausi, Lim, Erwin et al. (2010)]. These two methods detect and defend malware from static file detection and dynamic behavior detection respectively. However, these two methods are still belonging to traditional malware detection methods, and they still contain many shortcomings of traditional methods.

Computer virus immunity draws on biological immunity technologies, which is a way to refrain from detection false positives and false negatives detection, and to reduce the side effects of computers [Masood, Majeed, Samar et al. (2018); Jia, Yang and Guo (2017)]. At present, computer immunity technologies are mainly used to write and compile an

independent program to immunize a certain virus outbreak to spend the virus epidemic period. Every time the virus is overwritten, the program needs to be rewritten, compiled, and distributed. It takes time and resource consumption, and the release may not be timely. As a result, users may not get the latest vaccine. As mentioned above, now there is no immunization scheme that can quickly and effectively immunize a particular virus without making the computer overload.

This paper proposed an immunization scheme for ransomware to solve the problems mentioned above, by using virus immunity. The scheme enables the operating system running by users to be immune to the ransomware of a specific virus family. When an operating system is attacked, computers will not be infected by the virus family, thereby protecting the user's operating system from the attack of a ransomware.

## 2 Theory of immunization

The purpose of computer virus immunity is to make computers immune to virus before they are infected. Therefore, a computer running virus files or successfully be attacked by worm will not be infected. The implication is that computers can be protected from specific computer viruses by using such a relatively reliable and secure manner. Ransomware immunity method proposed in this paper can make computers not be infected with ransomware in a secure way. And it is using the same way as computer virus immunity. The theory of the immunization scheme proposed in this paper is to interfere with the triggering of the virus by controlling the mutex, service, registry and file in the operating system.

The theory of immunization by mutex control is shown in Fig. 1. The function of the mutex control module is creating a mutex. Once virus detects the mutex exists and determines that it has an instance of itself. It will satisfy the mutex condition rather than satisfy the trigger condition. Therefore, the virus cannot continue to execute subsequent process, achieving the goal of immunization.

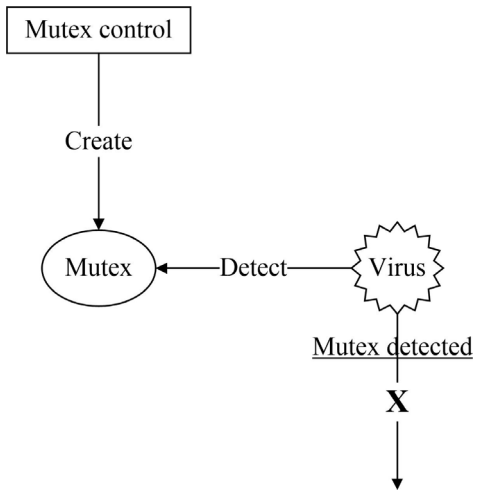


**Figure 1:** Process of mutex control immunization

Fig. 2 shows the process of service control immunization. The service control module creates a service which can be used by virus. If the virus detects the existence of infection through the detection service during the execution process, the virus will determine that the computer has been infected by itself, then it will start the harmless system to occupy the service and stop the subsequent infection operation, thereby achieving the purpose of immunization.

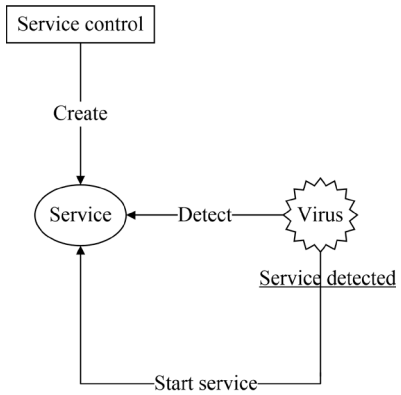


**Figure 2:** Process of service control immunization

The theory of immunization by file control is shown in Fig. 3, and it has four methods to immunize the virus.

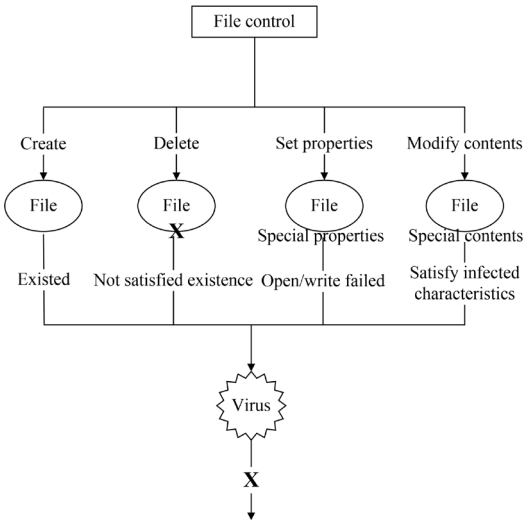


**Figure 3:** Process of file control immunization

1) The first method is to create a file. Some viruses will determine a computer has been infected by itself if it detects a certain file already exists. At this moment, the computer will not continue the infection process, and may create a file process. Some viruses may also delete themselves in order to hide malicious files. In this way, the computer achieves the purpose of immunization.

2) The second method is to delete a file. Some viruses are triggered by the presence of one or more files on a computer. And if these conditions are not met, the virus will not be triggered. Moreover, some viruses rely on certain files to infect a computer. Thus, if these files are deleted, the necessary conditions for triggering cannot be met, and the computer cannot be infected consequently. So we can achieve the goal of immunization.

3) The third method is to set some special properties in the file. Some viruses copy themselves to a directory when they infect a computer, or decrypt a malicious program to a directory, or download a malicious program from a C&C server to a directory, etc. If we set some special properties in these files, such as read-only, system, etc., the write process will fail, result in the infection will fail. Take an another example, some viruses must read and write a file when they run. If the permission fails to open, the virus process will crash or the subsequent infection process will not be possible ongoing. If it can interfere with the reading and writing of the virus, virus will crash and thereby achieve the purpose of immunization.

4) The fourth method is to modify a specified file's contents to specified contents. Some viruses will generate some marked files on the disk, such as encrypted files which be marked infection. If we analyze the characteristics of these files' contents and change the contents of these files to the specified contents on the user's host, the virus will determine that the infection has been completed, thus giving up further infection and achieving the purpose of immunization.

The theory of immunization by registry control is shown in Fig. 4.

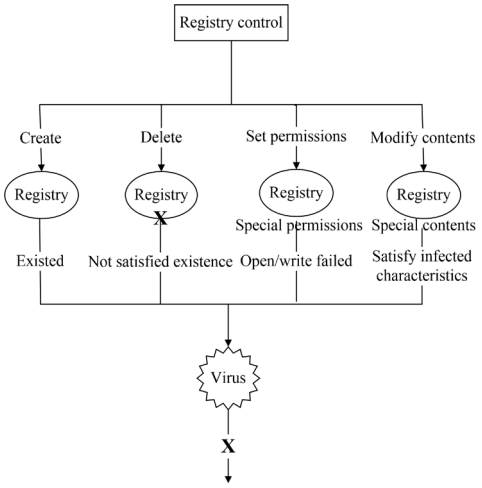


**Figure 4:** Process of registry immunization

Similar to the file control immunization process, registry control also contains four types: create, delete, set permissions, and modify contents. The theory of creating, deleting, and modifying contents is consistent with the file control, except that the operation object is changed from a file to a registry. And therefore it not be described repeatedly.

The only difference is to set permissions. Registry has no special properties, so we consider using the permission settings for immunization when designing registry control. If the permission of the registry is set to no permission to read or write, the virus will fail

to read or write the registry, causing the virus to be unable to infect the registry and achieving the purpose of immunization.

## 3 Immunization process

The immunization scheme proposed in this paper includes two processes: 1) the system administrator analyzes the virus, writes the immunization and deploys the update; 2) the client performs the immunization operation after the user initiates the immunization request.

Fig. 5 depicts the immunization process. Once obtaining a virus sample, it provides to system administrator for analysis. The system administrator analyzes the information that can be used for immunization from the sample, and then writes it into a configuration file and deploys it to the server.

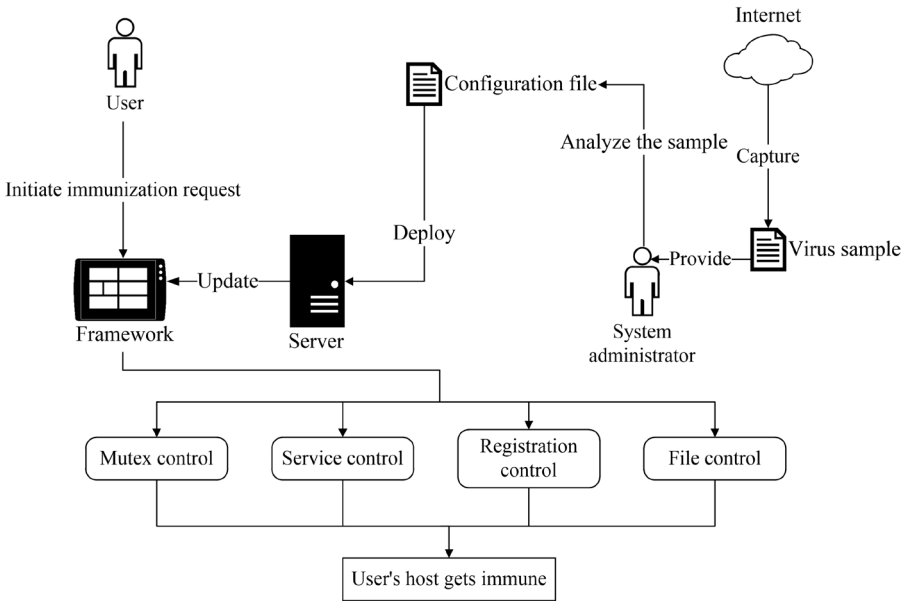


**Figure 5:** Immunization process

After a user initiates the immunization request to the client, the client downloads the latest immune configuration file from the server. And then it performs mutex control, service control, registry control, and file control according to the configuration file. And finally the user's host is immune.

As we can see from the immunization process, the normal work of the system depends on the correct analysis of the virus samples by the system administrator and the correct configuration of the immune configuration file.

When the content of the immune configuration file is rich enough, the user's computer can be immune to a variety of viruses at the same time. When a certain ransomware virus breaks out, as long as the system administrator analyzes the virus, writes the configuration file, and deploys the update in time, the user can obtain the immunity to the virus in time.

## 4 System design

### *4.1 Architecture*

In this paper, the ransomware-oriented immunization scheme is based on the C/S architecture, which is divided into servers and clients. Users use the client to get updates from the server and enable immunization on the client's host. The server provides clients with the latest immune configuration and version query service. The network architecture of the system is shown in Fig. 6.

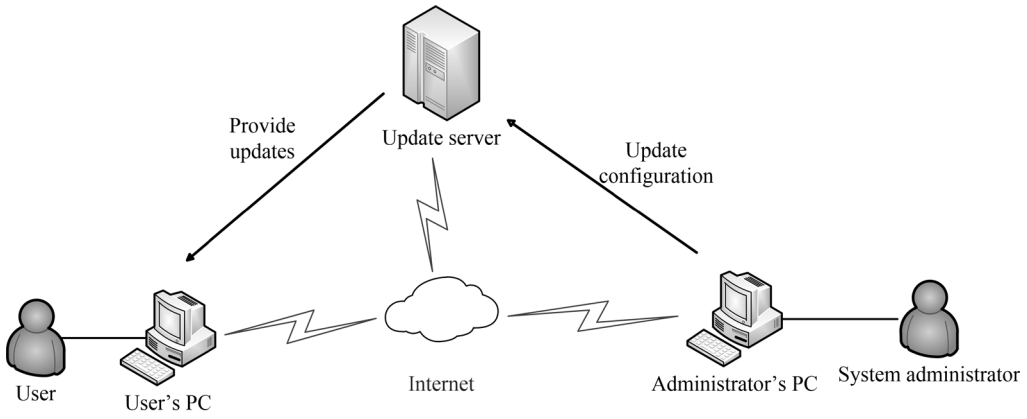


**Figure 6:** Network architecture of the system

The client runs on the user's PC, communicates with the update server through the Internet, and obtains the update from server. After the system administrator configures immune configuration file, the latest immune configuration is deployed to the server through Internet, and server provides the latest configuration file to users.

The client uses a three-layer architecture and is divided into a data layer, a logical layer, and a display layer.

1) The data layer provides the relevant interfaces for the logical layer. The data layer consists of a log module, a cryptographic algorithm module, and a configuration file module. The log module provides interfaces for writing, reading, and clearing logs. The cryptographic algorithm module provides encryption and decryption interfaces for the data. And the configuration file module provides parsing and verification interfaces for the configuration file.

2) The logical layer provides the relevant interfaces for the display layer and is responsible for the execution of the actual operation. The logical layer consists of an update module, an interface control module, a registry control module, a mutex control module, a file control module, and a service control module. The update module obtains the update from the server and saves it. The interface control module provides a refresh interface to the display layer. The registry control module is responsible for the control logic execution of the registry. The mutex control module creates and deletes the control structure. The file control module is responsible for the control logic execution of the file. And the service control module adds and deletes system services.

3) The display layer provides interactive interfaces to users. The display layer consists of three parts: list display and selection, information display, and immune switch. The list

display and selection display the list of available immunizations to the user, and provides users with the interface to select the immunizations. The immune switch provides the user with an immune switch interface through which the user controls the start and stop of immunization.

The architecture of the server is designed as two layers, the data layer and the interaction layer.

1) The data layer provides interfaces to cryptographic algorithm and configuration files. The cryptographic algorithm module provides data encryption and decryption interfaces, and the configuration file module provides read, save, and verify interfaces for the configuration file.

2) The interaction layer is divided into two parts, one is to provide relevant information display to the user's PC, the other is to provide the administrator with the interface for displaying and changing the list and the interface for displaying and changing the configuration.

### *4.2 Function module*

1) Client

The client includes six modules, which are update module, configuration file module, cryptographic algorithm module, control module, log module, and network module.

The update module is divided into two parts, check for updates and get updates. Check for updates part gets the latest update version from the server and compares it to the local version. The get updates part is used to get the latest encrypted configuration file from the server and save it locally.

The configuration file module is divided into immune configuration reading part, interface configuration reading part, enable list saving part, and configuration signature verification part. The immune configuration reading part is used to read the configuration of the ransomware immunization from the local. The interface configuration reading is used to read the configuration for the interface control. The enable list saving is used to store the unclosed immune list locally, so that the previous immunization is still properly enabled after the operating system is restarted. The signature verification configuration ensures that the integrity of the configuration and the configuration is not been tampered with in the transmission and storage process, so as to ensure that the system will not be used to attack the user's computer.

The cryptographic algorithm module is divided into an encryption and decryption module and a hash module. The encryption and decryption module supplies encryption and decryption of AES CTR algorithm and RSA algorithm. The length of the AES Key is 128 bits, and the length of the RSA Key is 4096 bits. The hash module offers an implementation of the SHA 256 algorithm.

The control module is divided into five parts: mutex control, service control, registry control, file control, and interface control. Mutex control part is mainly used to create and delete mutexes. Service control part is mainly used to create and delete services. Registry control part is mainly used to create, modify and delete registry and values, and change permissions. File control part is mainly used to create, change, delete files and change

attributes of the files, and change permissions. The interface control part is mainly used to update the interface contents such as the available immune list and the ransomware introduction according to the interface configuration.

The log module is divided into four parts: write log, clear log, display log and export log. Write log part writes the operations and results of the control part to the local disk. Clear log part is used to periodically clear the expired content. The display log part mainly displays the log formatted. The export log part mainly exports logs to other files.

2) Server

The server is mainly divided into three modules: a configuration file module, a cryptographic algorithm module, and a display module.

The configuration file module is divided into configuration generation part and configuration reading part. The configuration generation part is divided into mutex configuration, service configuration, registry configuration, file configuration, and interface configuration, which are used to generate a configuration of a mutex, a service, a registry, a file, and a client interface respectively. The configuration reading part is used to read configuration files stored in a specific format on the server.

The cryptographic algorithm module offers the same functionality as the client's. The algorithm called is implemented in OpenSSL, but the calling interface and the compile platform is slightly different.

The display module is divided into a version display part and a configuration display part. The version display part is used to display the latest version and provides a version query interface. The configuration display part is used to display the latest configuration file and digest corresponding to the configuration file.

## 5 Conclusion

Based on computer immune technologies, we proposed an immunization scheme for ransomware. The scheme immunizes the ransomware from four parts: mutex, service, file, and registry. The mutex part is mainly used to preempt the mutexes used by the virus. The service part is mainly used to register the service in advance that the virus will register. The file part is mainly used to preempt the files used by the virus, so that the virus cannot open the required files. And the registry part is mainly used to create the registry key used by the virus in advance, so that the virus cannot open the required registry key. This program can immunize a variety of current ransomware and has certain practical application value.

**Funding Statement:** This work is supported in part by the National Natural Science Foundation of China under grant No. 61872069, in part by the Fundamental Research Funds for the Central Universities (N2017012)

**Conflicts of Interest:** The authors declare that they have no conflicts of interest to report regarding the present study.

**References**

**Akbanov, M.; Vassilakis, V. G.; Logothetis, M. D.** (2018): Ransomware detection and mitigation using software-defined networking: the case of WannaCry. *Computers & Electrical Engineering*, vol. 76, pp. 111-121.

**Al-Hawawreh, M.; den Hartog, F.; Sitnikova, E.** (2019): Targeted ransomware: a new cyber threat to edge system of brownfield industrial internet of things. *IEEE Internet of Things Journal*, vol. 6, no. 4, pp. 7137-7151.

**Biryukov, A.; Tikhomirov, S.** (2019): Transaction clustering using network traffic analysis for bitcoin and derived block chains. *Proceedings of IEEE Conference on Computer Communications Workshops*, pp. 204-209.

**Farral, T.** (2017): Nation-state attacks: practical defences against advanced adversaries. *Network Security*, vol. 2017, no. 9, pp. 5-7.

**Firdausi, I.; Lim, C.; Erwin, A.; Nugroho, A. S.** (2010): Analysis of machine learning techniques used in behavior-based malware detection. *Proceedings of Second International Conference on Advances in Computing, Control and Telecommunication Technologies*, pp. 201-203.

**Homayoun, S.; Dehghantanha, A.; Ahmadzadeh, M.; Hashemi, S.; Khayami, R. et al.** (2019): DRTHIS: deep ransomware threat hunting and intelligence system at the fog layer. *Future Generation Computer Systems*, vol. 90, pp. 94-104.

**Jia, Z. J.; Yang Y. Y.; Guo, N.** (2017): Research on computer virus source modeling with immune characteristics. *Proceedings of 29th Chinese Control and Decision Conference*, pp. 4616-4619.

**Kolbitsch, C.; Comparetti, P. M.; Kruegel, C.; Kirda, E.; Zhou, X. Y. et al.** (2009): Effective and efficient malware detection at the end host. *The 18th USENIX Security Symposium*, pp. 351-366.

**Li, Y.; Xu, G. Q.; Xian, H. Q.; Rao L. L.; Shi, J. G.** (2019): Novel android malware detection method based on multi-dimensional hybrid features extraction and analysis. *Intelligent Automation and Soft Computing*, vol. 25, no. 3, pp. 637-647.

**Masood, Z.; Majeed, K.; Samar, R.; Raja, M. A. Z.** (2018): Design of epidemic computer virus model with effect of quarantine in the presence of immunity. *Fundamenta Informaticae*, vol. 161, no. 3, pp. 249-273.

**Sai, A. R.; Buckley, J.; Le Gear, A.** (2019): Assessing the security implication of bitcoin exchange rates. *Computers & Security*, vol. 86, pp. 206-222.

**Santos, I.; Brezo, F.; Nieves, J.; Penya, Y. K.; Sanz, B. et al.** (2010): Idea: opcode-sequence-based malware detection. *Engineering Secure Software and Systems. Proceedings Second International Symposium*, pp. 35-43.

**Sun, G. X.; Bin, S.; Jiang, M.; Cao, N.; Zheng, Z. Y. et al.** (2019): Research on public opinion propagation model in social network based on Blockchain. *Computers, Materials & Continua*, vol. 60, no. 3, pp. 1015-1027.

**Wang, X. M.; Li, J.; Kuang, X. H.; Tan, Y. A.; Li, J.** (2019): The security of machine learning in an adversarial setting: a survey. *Journal of Parallel and Distributed Computing*, vol. 130, pp. 12-23.

**Xu, J.; Wang, A. D.; Wu, J.; Wang, C.; Wang, R. J. et al.** (2020): SPCSS: social network-based privacy-preserving criminal suspects sensing. *IEEE Transactions on Computational Social Systems*, vol. 7, no. 1, pp. 261-274.

**Zhu, T. Q.; Yang, M. M.; Xiong, P.; Xiang, Y.; Zhou, W. L.** (2018): An iteration-based differentially private social network data release. *Computer Systems Science and Engineering*, vol. 33, no. 2, pp. 61-69.