# Smartphone Malware Detection Model Based on Artificial Immune System

WU Bin[1], LU Tianliang[2], ZHENG Kangfeng[1], ZHANG Dongmei[1], LIN Xing[1]

[1] Information Security Laboratory, National Disaster Recovery Technology Engineering Laboratory, Beijing University of Posts and Telecommunications, Beijing 100876, P.R. China
[2] People's Public Security University of China, Beijing 100038, P.R. China

**Abstract:** In order to solve the problem that the traditional signature-based detection technology cannot effectively detect unknown malware, we propose in this study a smartphone malware detection model (SP-MDM) based on artificial immune system, in which static malware analysis and dynamic malware analysis techniques are combined, and antigens are generated by encoding the characteristics extracted from the malware. Based on negative selection algorithm, the mature detectors are generated. By introducing clonal selection algorithm, the detectors with higher affinity are selected to undergo a proliferation and somatic hyper-mutation process, so that more excellent detector offspring can be generated. Experimental result shows that the detection model has a higher detection rate for unknown smartphone malware, and better detection performance can be achieved by increasing the clone generation.

**Key words:** artificial immune system; smartphone malware; detection; negative selection; clonal selection

## I. INTRODUCTION

As the major threat to phone security, malware does harm to the users in various ways, including system damage, malicious deduction, information stealing, remote control, etc. According to the monitoring statistics of CNCERT and the annual reports of some security enterprises, in 2012 there were 162,981 malware samples on the mobile Internet[1].

In order to detect the unknown mobile malware effectively, we use in this study both static malware analysis and dynamic malware analysis, and propose a smartphone malware detection model (SP-MDM) based on artificial immune system according to the mechanism of biologic immune system that can protect us from infection by organisms. In this model, the static signatures and dynamic signatures of malware are extracted, and based on real-valued vector encoding the antigens are generated. The immature detector grows into a mature one if it goes through self-tolerance. Detector offspring with higher affinity are created after the optimization of mature detectors using clonal selection algorithm.

## II. RELATED WORK

### 2.1 Malware detection technologies

Malware detection technologies are mainly divided into two kinds: static detection technology and dynamic detection technology.

Through static detection technology, the possible malicious behaviors are inferred by analyzing the binary or source code without running the program.

Through dynamic detection technology, the malware is installed and run in a controllable and recoverable environment, and its behaviors are monitored.

In 2012, Zhang[2] put forward a malware detection model for smartphones using static detection and analytic hierarchy process. Based on behavior signatures for android platform, Burguera[3] proposed a malware detection system named Crowdroid. In 2012, Shabtai[4] presented

Andromaly - a framework for detecting malware in android mobile devices.

### 2.2 Artificial immune system

Artificial immune systems (AIS)[5] are adaptive systems inspired by the biological immunology and observed immune functions, principles and models, and applied to problem solving. AIS have been applied in many engineering fields including data mining[6], fault detection and diagnosis[7], optimization[8], intrusion detection[9] and malware detection[10].

In 1974, Jerne[11] proposed an immune network theory. His research established the foundation for AIS inspired by the mechanism of T-cell maturation and self tolerance in the immune system, Forrest[12] proposed the negative selection algorithm (NSA) in 1994. In 2000, de Castro & Von Zuben proposed the clonal selection algorithm[13], which was named CLONALG and formally described in 2002[14].

In information security field the artificial immune system is of a high research value[15]. The basic theory of the artificial immune system is introduced to solve the problem of smartphone malware detection in this paper.

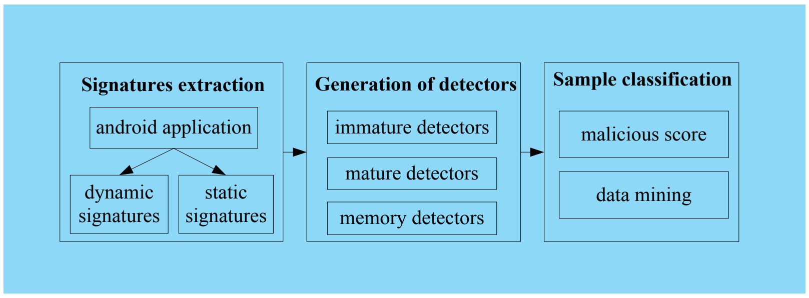

**Fig.1** *Detection model based on AIS*

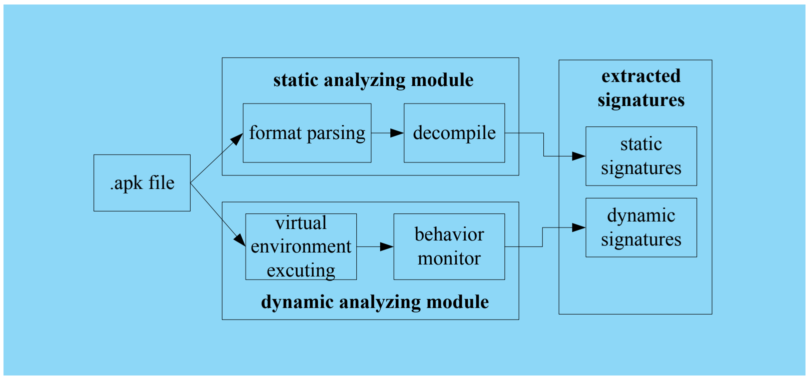

**Fig.2** *Extraction of signatures*

## III. Model Based on AIS

The malware detection model based on AIS includes three phases: signatures extraction, generation of detectors and sample classification, as shown in Fig1.

The main work of research and innovation in this paper are focused on the following points. (1) To reveal the comprehensive features of malware, both static signatures and dynamic signatures are extracted; (2) To generate high quality detectors more effectively, the Cauchy mutation is introduced during the clone and mutation of detectors and the gene pool is added to reserve the good features.

### 3.1 Signatures extraction and encoding

3.1.1 Signatures extraction

Both static and dynamic signatures are extracted, as shown in Fig 2.

Our analysis is focused on the detection techniques of android malware.

(1) Extraction of static signatures

The file format of android application is APK, actually a ZIP format that can be decompressed by Unzip.

Based on the file format parsing and decompilation, six types of static signatures are extracted, including: activities, services, broadcast receivers, permissions, hardware features and URLs.

(2) Extraction of dynamic signatures

Some researchers and developers have proposed and realized a few tools to analyze the behavior signatures of android applications, including: DroidBox [16], TaintDroid [17].

Based on virtual machine and behavior monitoring techniques, totally seven types of dynamic signatures are extracted, including: started services, file operation, network operation, message operation, phone operation, crypto operation and data leaks.

### 3.1.2 Signatures encoding

The real-valued vector encoding scheme is used to encode signatures extracted from samples.

The encoded real-valued vector contains 7 dimensions. The 1st dimension represents the types of signatures. The 2nd dimension represents the sub-types of signatures. If there is no sub-type, the 2nd dimension is assigned the value of 1. The 3rd to the 7th dimensions are utilized to store the encoded date of signatures. The detailed encoding scheme is shown in Table I.

Normalize the vector $F=(f_1,f_2,\cdots,f_7)$ . $f_i^{\max}$ denotes the maximum value of the *i*-th dimension of *F*, and $f_i^{\min}$ denotes the minimum value of the *i*-th dimension of *F*.$f_i$ is the value before normalization, and $f_i^{'}$ is the value after normalization.

$$f_i^{'}=\begin{cases}(f_i-f_i^{\min})/(f_i^{\max}-f_i) & f_i^{\max}>f_i^{\min}\\ 0.5 & f_i^{\max}=f_i^{\min}\end{cases} \qquad (1)$$

### 3.2 Generation of detectors

3.2.1 Negative selection

The negative selection algorithm is improved in our paper. Besides random generation, three new methods are introduced, including: 1) Encoding the signatures of malware files. 2) Clone and mutation of high affinity detectors. 3) Recombinant of gene segments of memory detectors.

The real-valued negative selection algorithm with variable-sized detector (V-Detector [18]) is explored.

Detectors are defined as follows.

$$d=<(x_1,x_2,\cdots,x_n),r_d>$$

Where *n* is the dimension of detectors, and $r_d$ is the radius of detectors. The detector is considered as a hyper-sphere with radius $r_d$ and centered with the point $(x_1,x_2,\cdots,x_n)$. The self-element is defined as $s=(y_1,y_2,\cdots,y_n)$. The distance between *d* and *s* is calculated by Euclidean distance.

$$Ed(d,s)=(\sum_{i=1}^{n}(x_i-y_i)^2)^{1/2} \qquad (2)$$

*D*=($d_1$,$d_2$... , $d_{Ns}$) denotes the detector set with the total number $N_d$. $S=\{s_1,s_2,\cdots,s_{Ns}\}$ denotes the self set with the total number $N_s$. The radius of self-elements is identical, with the value $r_s$. The mature process of detector is shown as follows.

**Table I** *Encoding of signatures*

| *1st*<br>*(Types)* | *2nd*<br>*Sub-types* | *3rd-7th*<br>*Data* |
|---|---|---|
| *activities(1)* | *no sub-types(1)* | *activity name* |
| *services(2)* | *no sub-types (1)* | *service name* |
| *broadcast (3)* | *no sub-types (1)* | *broadcast name* |
| *permissions(4)* | *no sub-types (1)* | *permission name* |
| *Hardware (5)* | *no sub-types (1)* | *hardware name* |
| *URLs(6)* | *no sub-types (1)* | *URL string* |
| *started services(7)* | *no sub-types (1)* | *service name* |
| *file operation(8)* | *read file(1),*<br>*write file(2)* | *file path* |
| *network*<br>*operation(9)* | *send data(1),*<br>*receive data(2)* | *IP(domain), port* |
| *message*<br>*operation(10)* | *send message(1),*<br>*receive message(2)* | *phone number* |
| *phone*<br>*operation(11)* | *make calls(1),*<br>*receive calls(2)* | *phone number* |
| *crypto*<br>*operation(12)* | *encryption(1), decryption(2)*<br>*message(1),* | *key, data* |
| *data leaks(13)* | *network(2),*<br>*file(3)* | *data, destination* |

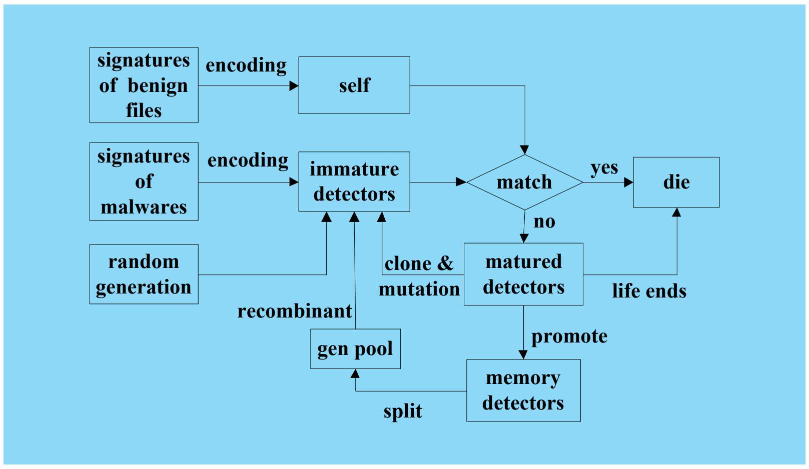

**Fig.3** *Extraction of signatures*

```
 for each element  in self set S {
   calculate the distance Ed between immature
 detector x and self element  $S_i$;
  if  $Ed - r_s \le r_d$
  $r_d = Ed - r_s$ ;  }
  if rd > rs
    detector  $d =< x, r_d >$  is mature,
   $D = D \cup \{d\}$ ;
```

### 3.2.2 Clone and mutation

The lymphocytes that recognize the antigens are selected to proliferate, which improves their affinity.

The CLONALG is used to optimize the affinity of mature detectors. The non-self space coverage of a detector and overlapped space with other detectors are used as affinity measurement. The affinity of detector is shown as follows.

$$affinity(d_i) = Vol(d_i) - \delta * Olp(d_i, D) \qquad (3)$$

$Vol(d_i)$ denotes the detector's non-self space coverage, and $\delta$ denotes the overlapping punishment factor that can be adjusted on demands. $Olp(d_i, D)$ denotes the overlapped space between detector $d_i$ and other detectors in set $D$.

Detector $d_i$ is a n-dimensioned hyper-sphere with the radius $r_d$. The volume of covered space is calculated as follows.

$$Vol(d_i) = \frac{\pi^{n/2}}{\Gamma(n/2+1)} \cdot r_d^{\ n} \qquad (4)$$

If n is even, let $m = n/2$, then

$$\Gamma(n/2+1) = \Gamma(m+1) = m! \qquad (5)$$

If $n$ is odd, let $m = (n+1)/2$, then

$$\Gamma(n/2+1) = \Gamma(m+1/2) = \frac{1\times 3\times\cdots\times(2m-1)}{2m}\sqrt{\pi} \qquad (6)$$

Use the method proposed by Zhao Xinchao et al. [19] to compute the approximation of the overlapped space of detector .

$$Olp(d_i, D) = \sum_{j\neq i} Olp(d_i, d_j) \qquad (7)$$

$$Olp(d_i, d_j) = \begin{cases} 0 & \| c_d^i - c_d^j \| \geq r_d^i + r_d^j \\ (\exp(\frac{r_d^i + r_d^j - \| c_d^i - c_d^j \|}{r_d^i + r_d^j}) - 1)^n & \| c_d^i - c_d^j \| < r_d^i + r_d^j \end{cases} \qquad (8)$$

The radius of detector $d_i$ and $d_j$ is $r_d^i$ and $r_d^j$. The center of detector $d_i$ and $d_j$ is $c_d^i$ and $c_d^j$, and $\| c_d^i - c_d^j \|$ denotes the distance between $d_i$ and $d_j$.

The detectors are optimized by cloning, pursuing more coverage in antigen space and less overlap with each other. The affinity of detector is calculated by Formula (3). Select the P best individuals based on affinity measure and clone them. The clone size is an increasing function of the detector's affinity. The clone size *N(d$_i$)* of detector $d_i$ is shown as follows, where $a$ is the clonal coefficient.

$$N(d_i) = \alpha * affinity(d_i) \qquad (9)$$

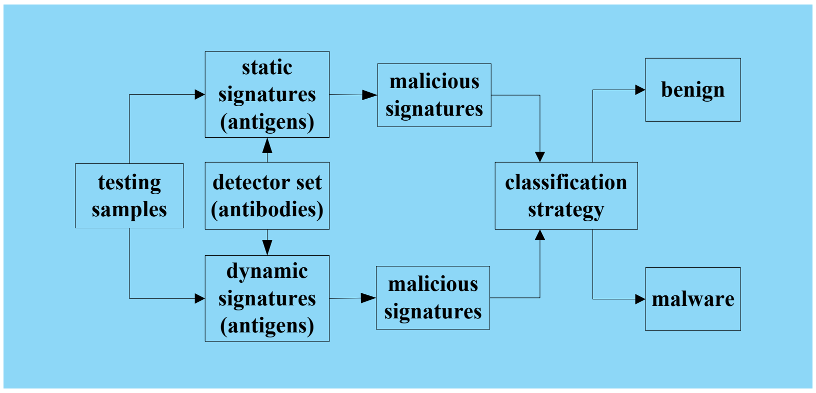

**Fig.4** *Sample Classification*

In order to find detectors with higher affinity, the detector offspring will be evolved to expand the searching space. To increase the global searching ability, the Cauchy mutation is introduced.

The one-dimensional Cauchy density function is defined by:

$$f(x; x_0, \gamma) = \frac{1}{\pi}\frac{\gamma}{\gamma^2 + (x - x_0)^2} \qquad (10)$$

The distribution function is:

$$F(x; x_0, \gamma) = \frac{1}{\pi}\arctan(\frac{x - x_0}{\gamma}) + \frac{1}{2} \qquad (11)$$

Where $x_0$ is the local parameter, specifying the location of the peak of the distribution, and γ is the scale parameter. The special case when $x_0$=0 and γ=1 is called the standard Cauchy distribution.

New detector $d_i^{'}$ is generated from detector $d_i$ under the Cauchy mutation operator. The mutation process is as follows.

$$\begin{cases} d_i^{'}(j) = d_i(j) + \eta_i(j)\delta_j & j = 1, 2, \cdots, n \\ \eta_i^{'}(j) = \eta_i(j)\exp(\tau_a N(0,1) + \tau_b N_j(0,1)) & j = 1, 2, \cdots, n \end{cases} \qquad (12)$$

Where $\delta_j$ is a standard Cauchy random variable, and $\eta_i$ is a strategy parameter. $d_i(j)$, $d_i^{'}(j)$, $\eta_i(j)$, $\eta_i^{'}(j)$ denote the $j$-th component of vectors $d_i$, $d'_i$, $\eta_i$ , $\eta'_i$. $N(0,1)$ denotes a normally distributed one-dimensional random number with mean 0 and standard deviation 1.$N_j(0,1)$ indicates that the random number is generated anew for each value of $j$. The factors $\tau_a$ and $\tau_b$ is commonly set as follows.

$$\tau_a = (\sqrt{2\sqrt{n}})^{-1}, \quad \tau_b = (\sqrt{2n})^{-1}$$

According to Formula (3), the affinity of mutated offspring is calculated. The individuals with high affinity go through self-toleration process, and add to the mature detector set. The individuals with low affinity will be removed.

### 3.2.3 Memory detectors

If a mature detector can recognize enough antigens within its life span *T*, i.e. achieves activation threshold *A*, it will become a memory detector, and else it will die and be removed. In order to keep the excellent characteristics of memory detector set, a set capacity *Nm* is defined.

To preserve the good genes in memory detectors, gene pool is introduced. Based on sliding window, the memory detector to be replaced will be split into some gene segments. As one source of immature detectors, the split segments in gene pool are recombined.

### 3.3 Sample classification

Extract the signatures of smartphone testing samples, and encode these signatures to generate antigens. If the distance between an antigen and a detector (antibody) is less than detection radius $r_d$, then this antigen is considered as non-self malicious signature. Counting total numbers of the malicious signature that belongs to each signature type. After normalization, the signature describing vector $F=\{f_1,f_2,\cdots,f_{13}\}$ is generated. Elements of each dimension are defined over the range [0, 1].

Define danger weight $\omega_i$ for each signature type of vector $F$. The following two strategies for classifying malware are adopted.

Strategy I: Composite weighted score method. The malicious score of sample S is calculated using the following formula.

$$Score(\mathrm{S})=\sum_{i=1}^{13}\omega_i f_i \qquad (13)$$

Define the threshold $C_{Th}$ of malicious score. If $Score(\mathrm{S})\geq \mathrm{C}_{Th}$, the sample $S$ is classified as malware.

Strategy II: Method based on data mining algorithms. The weighted K-means algorithm is used in our model.

Define number of clusters $k$. The distance between cluster center $C=\{c_1,c_2,\cdots,c_{13}\}$ and vector $F=\{f_1,f_2,\cdots,f_{13}\}$ is calculated as follows.

$$d(C,F)=\sqrt{\sum_{i=1}^{13}\omega_i(c_i-f_i)^2} \qquad (13)$$

Set k=2, the testing samples are partitioned into two clusters. The cluster with higher mean malicious score is identified as malware set, and the other cluster is identified as benign set.

## IV.EXPERIMENTS AND RESULTS

We collect 20 malwares and 20 benign files as testing samples set. Malwares are mainly downloaded from security research organization [20], etc.

(1) Calculating the malicious score

Extract signatures of test samples, build vector F by matching detectors, and calculate the malicious score for each testing sample. Set *Nd*=1500, *Nc*=50, and the danger weight is

$$W=\{\omega_1,\omega_2,\cdots,\omega_{13}\}=\{0.05,0.05,0.05,0.1,0.05,0.05,0.05,\\ 0.1,0.1,0.1,0.1,0.05,0.15\}$$

The malicious score of testing samples is shown as Table II.

Choose threshold $C_{Th}$=0.35, then the detection rate *DR=16/20=80%*, the false positive rate *FPR=2/20 =10%.*

(2) Detection performance affected by $N_c$ and $N_d$

The effect of $N_c$ and $N_d$ on detection rate *DR* is shown in Fig 5. *DR* rises with the increase of $N_d$, and the rising speed gradually slows down, finally leveled off. In general, for an identical $N_d$, *DR* rises with the increase of $N_c$.

The effect of $N_c$ and $N_d$ on false positive rate FPR is shown in Fig 6. *FPR* declines with the increase of $N_d$. In general, for an identical $N_d$, *FPR* declines with the increase of $N_c$.

(3) Result comparison

Get the mean detection rate of SP-MDM for each smartphone malware by repeating the experiment 20 times, and then compare with the detection rate of the following two classic detection methods: VirusTotal[26] which is a free online virus scanner with multiple antivirus engines, and the detection method based on the original negative selection algorithm (NSA) without clone and mutation.

The result is shown in Fig 7. For most of the

**Table II** *Malicious score of testing samples*

| Malware Samples | Malicious Score | Benign Samples | Score |
|---|---|---|---|
| FakeInst | 0.55 | MX Player | 0.14 |
| FakeNotify | 0.41 | Tiny Flashlight | 0.19 |
| Pirates | 0.22 | Sina Weibo | 0.32 |
| YZHCSMS | 0.31 | Instagram | 0.25 |
| Kmin | 0.68 | Google Calendar | 0.18 |
| Chuli | 0.39 | Google Maps | 0.32 |
| DroidKungFu | 0.54 | Baidu Input | 0.28 |
| Extension | 0.46 | Chese Free | 0.27 |
| FakeApp | 0.65 | Fruit Ninja | 0.19 |
| FakeTimer | 0.44 | WeChat | 0.36 |
| GingerMaster | 0.59 | UC Browser | 0.21 |
| InfoStealer | 0.60 | Adobe Reader | 0.11 |
| Koomer | 0.49 | Amazon Mobile | 0.35 |
| MailStealer | 0.58 | BBC News | 0.24 |
| Tascudap | 0.33 | Androidesk Wallpaper | 0.18 |
| Remote.Androrat | 0.53 | Note Anytime | 0.21 |
| Guggespy | 0.47 | iReader | 0.19 |
| FakeJobOffer | 0.30 | HiMarket | 0.30 |
| Privacy.Prospero | 0.49 | Carrot Fantasy | 0.15 |
| Spybubble | 0.57 | KaKao Talk | 0.32 |

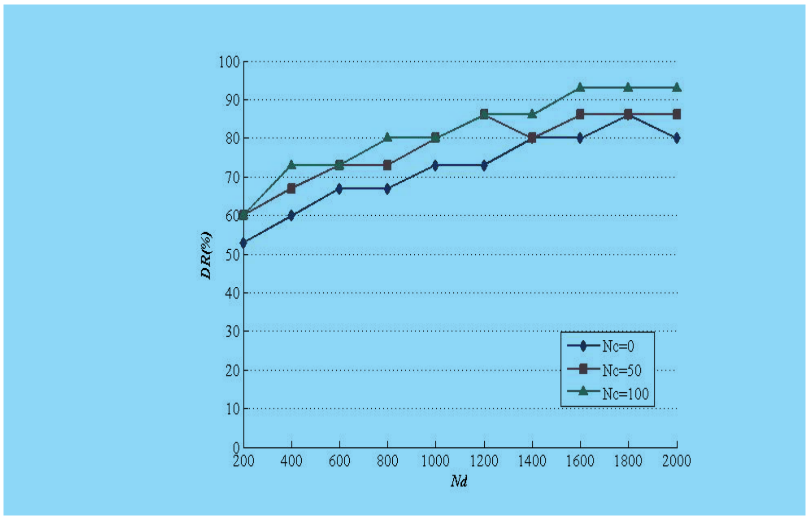

**Fig.5** *Simulation of detection rate*

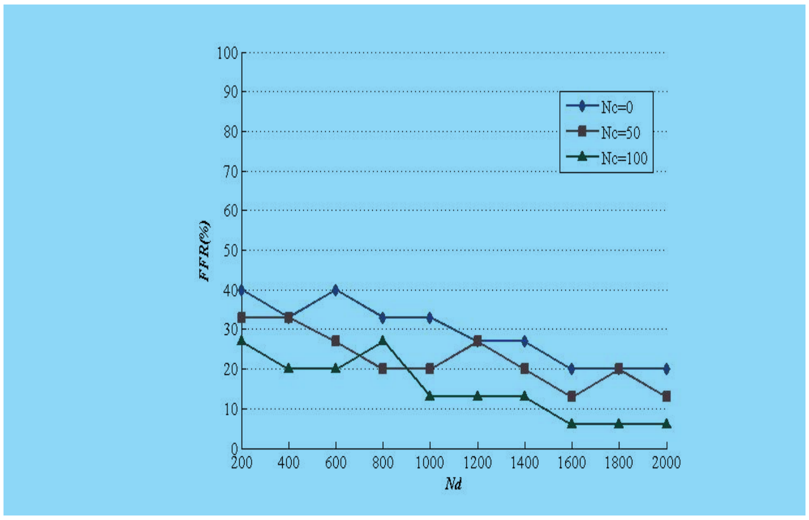

**Fig.6** *Simulation of false positive rate*

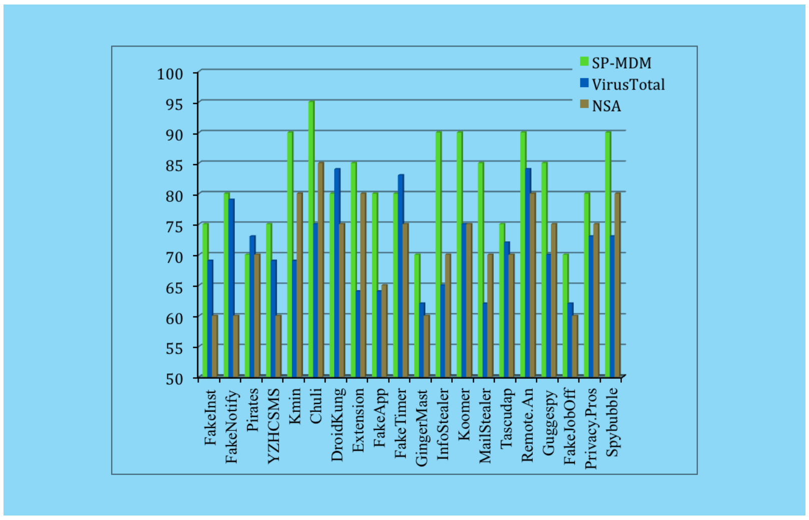

**Fig.7** *Comparison of detection rate*

malware samples, the detection rate of SP-MDM is higher than that of VirusTotal. For some malware samples, due to their low malicious score, the detection rate of SP-MDM is lower than that of VirusTotal. On the whole, the detection rate of SP-MDM is higher than that of NSA, which demonstrates the effect of clone and mutation of SP-MDM.

## V. Conclusion

Based on the artificial immune system, we propose a smartphone malware detection model (SP-MDM). Compared with other models, SP-MDM has the following advantages: 1) both static signatures and dynamic signatures of malware are taken into account, ensuring that the model can effectively detect unknown malware; 2) use variable-sized detectors to improve the detection performance; 3) the clone and mutation mechanism is introduced. Through multiple generation optimization, better detection performance can be achieved.

## Acknowledgments

This work was supported in part by National Natural Science Foundation of China under Grants No.61101108;National S&T Major Program under Grants No.2011ZX03002 -005-01.

## References

[1] CNCERT/CC, “Overview of Internet network security situation of China in 2012”, 2013-03-20.

[2] Zhang Miao, Yang Youxiu, Cheng Gong, et al, “Malware detection in smartphones using static detection and evaluation model based on analytic hierarchy process”, China Communications, 2012, 9(12): 144-152.

[3] Burguera I, Zurutuza U, Tehrani S N, “Crowdroid: behavior-based malware detection system for Android”, Proceedings of the 1st ACM workshop on Security and privacy in smartphones and mobile devices, New York, USA: ACM, 2011: 15-26.

[4] Shabtai A, Kanonov U, Elovici Y, et al, “Andromaly: a behavioral malware detection framework for android devices”, Journal of Intelligent Information Systems, 2012, 38(1): 161-190.

[5] de Castro L N, Timmis J, “Artificial immune

systems: a new computational intelligence approach", Springer, Berlin, 2002.

[6] Freitas A A, Timmis J, "Revisiting the foundations of artificial immune systems for data mining", IEEE Transactions on Evolutionary Computation, 2007, 30(5): 540-551.

[7] Silva G C, Palhares R M, Caminhas W M, "Immune inspired fault detection and diagnosis: a fuzzy-based approach of the negative selection algorithm and participatory clustering", Expert Systems with Applications, 2012, 39(16): 12474-12486.

[8] Yap David F W, Koh S P, Tiong S K, "A hybrid artificial immune systems for multimodal function optimization and its application in engineering problem", Artificial Intelligence Review, 2012, 38(4): 291-301.

[9] D'haeseleer P, Gonzalez F, "An immunity-based technique to characterize intrusion in computer networks", IEEE Transactions on Evolutionary Computation, 2002, 6(3): 1081-1088.

[10] Afaneh S, Zitar R A, Al-Hamami A, "Virus detection using clonal selection algorithm with Genetic Algorithm (VDC algorithm)", Applied Soft Computing, 2013, 13(1): 239-246.

[11] Jerne N K, "Towards a network theory of the immune system", Annals of Immunology, 1974, 125C: 373-389.

[12] Forrest S, Perelson A S, Allen L, et al, "Self-nonself discrimination in a computer", Proceedings of IEEE Symposium on Research in Security and Privacy, Los Alamitos, CA: IEEE Press, 1994: 202-212.

[13] de Castro L N, Von Zuben F J, "The clonal selection algorithm with engineering applications", Proceedings of the Genetic and Evolutionary Computation Conference, Las Vegas, Nevada, 2000: 36-42.

[14] de Castro L N, Von Zuben F J, "Learning and optimization using the clonal selection principle", IEEE Transactions on Evolutionary Computation, 2002, 6(3): 239-251.

[15] Harmer P K, Williams P D, "An artificial immune system architecture for computer security applications", IEEE Transactions on Evolutionary Computation, 2002, 6(3): 252-280.

[16] DroidBox,http://code.google.com/p/droidbox.

[17] TaintDroid, http://appanalysis.org.

[18] Zhou Ji, Dasgupta D, "Real-valued negative selection algorithm with variable-sized detectors", Proceedings of Genetic and Evolutionary Computation Conference, Seattle, WA, 2004: 287-298.

[19] Zhao Xinchao, Liu Guoli, Liu Huqiu, et al, "A new clonal selection immune algorithm with perturbation guiding search and non-uniform hypermutation", International Journal of Computational Intelligence Systems, 2010,3(1): 1-17.

[20] Mobile malware samples, http://contagiodump.blogspot.com/2011/03/take-sample-leave-sample-mobile-malware.html.

## Biographies

***WU Bin,*** received his Ph.D. degree in signal and information processing from Beijing University of Posts and Telecommunications. He is currently a lecture in the National Disaster Recovery Technology Engineering Laboratory, Beijing University of Posts and Telecommunications. His research interests include network security, intrusion detection, apt analysis, and mobile networks security.

***LU Tianliang,*** received his Ph.D. degrees in information security from Beijing University of Posts and Telecommunications. He is currently a lecture in Chinese People's Public Security University. His research interests include network security, intrusion detection, apt analysis, and mobile networks security.

***ZHENG Kangfeng,*** received his Ph.D. degree in signal and information processing from Beijing University of Posts and Telecommunications in 2003. He is currently a professor in the National Disaster Recovery Technology Engineering Laboratory, Beijing University of Posts and Telecommunications. His major research interests include network security, big data analysis, and APT detection.

***ZHANG Dongmei,*** received her Ph.D. degrees in computer science from Beijing University of Posts and Telecommunications. She is an associate professor in school of Compute Science, Beijing University of Posts and Telecommunications. Her major research interests include sensor network security, the internet of things security, and industrial control network security.