ISSN 1992-6502 (Print) Вестник УГАТУ ISSN 2225-2789 (Online)
2016. Т. 20, № 2 (72). С. 117–122 http://journal.ugatu.ac.ru

УДК 004.056

СИСТЕМА ОБНАРУЖЕНИЯ ВРЕДОНОСНЫХ ПРОГРАММ В ОПЕРАЦИОННОЙ СИСТЕМЕ ANDROID

С. В. ЖЕРНАКОВ[1], Г. Н. ГАВРИЛОВ[2]

[1] zhsviit@mail.ru, [2] grigorijgavrilov@mail.ru

[1] ФГБОУ ВО «Уфимский государственный авиационный технический университет» (УГАТУ)
[2] ПАО «Уфимское моторостроительное производственное объединение» (УМПО)

Поступила в редакцию 27.05.2016

Аннотация. В рамках данной работы была поставлена задача повышения эффективности обнаружения вредоносных программ в операционной системе для мобильных устройств (на примере Android). Для достижения поставленной цели выполнен анализ защищенности операционной системы Android и формализация образцов вредоносных программ с целью выявления признаков, присущих их поведению. На основе полученной информации разработана экспериментальная выборка в состав которой входят векторы описывающие поведенческий характер двух типов программ: безопасные – ok и вредоносные – virus. В результате исследовательских экспериментов выбран метод классификации, который выполняет классификацию предложенной выборки с наиболее высокой точностью. Поставленная задача повышения эффективности обнаружения вредоносных программ решена с применением разработанной методики на основе машины опорных векторов и аппарата нечеткой логики. Данная методика реализована в виде исследовательского прототипа системы обнаружения вредоносных программ.

Ключевые слова: система обнаружения вредоносных программ, Android, машина опорных векторов, нечеткая логика, классификация, поведенческий характер.

В настоящее время операционная система (ОС) Android занимает 75–80% рынка всех мобильных устройств. Она имеет в своем составе множество возможностей, которые предоставляются за счет минимального количества ограничений по отношению к перечню доступных функций, что отрицательно влияет на ее защищенность в целом. ОС Android имеет открытый исходный код, большую свободу действий в отношении доступных функциональных возможностей. Встроенная модель безопасности ОС Android содержит надежные механизмы защиты, но тем не менее она обладает рядом недостатков по отношению к вредоносным программам [1]. Популярность и развитием современной сети Интернет позволяет с высокой скоростью распространить программы данного типа на множество мобильных устройств. Таким образом, угроза со стороны вредоносных программ является актуальной темой для мобильных устройств, работающих под управлением ОС Android. Изложенные выше факты послужили поводом для исследований в области обнаружения вредоносных программ на основе их поведенческого характера.

Анализ отечественных и зарубежных публикаций [2–6] по данной тематике показывает, что работы в данной области активно ведутся. Однако в результатах этих работ отсутствуют практические рекомендации, а также качественные и количественные характеристики разработанных программных проектов для систем комплексной защиты средств мобильной связи типа Android.

Цель данной работы – повышение эффективности обнаружения вредоносных программ в ОС для мобильных устройств (на примере Android) путем разработки методики обнаружения вредоносных программ на основе анализа их поведенческого характера.

ФОРМАЛИЗАЦИЯ И АНАЛИЗ ОБРАЗЦОВ ВРЕДОНОСНЫХ ПРОГРАММ

Для того чтобы описать поведенческий характер вредоносных программ, выполнена фор-

мализация образцов вредоносных программ, которые в настоящее время можно найти в сети Интернет. Формализация заключается в анализе исходного кода (classes.dex), конфигурационных файлов (androidmanifest.xml) и системных вызовов программы ОС Android [7]. Согласно полученным данным разработана экспериментальная выборка описывающая поведение как вредоносных так безопасных программ. Фрагмент выборки представлен в табл. 1. Она представляет собой экспериментальную выборку, заданную в бинарной форме (0 – отсутствует, 1 – присутствует) и включает в себя 100 векторов поведения программ. Столбцы с 1 по 10 содержат в себе информацию о выявленных в процессе формализации признаках программ. С 10 по 162 список всех разрешений. Столбец 163 и 164 – значения, полученные путем анализа используемых системных процессов в ОС для мобильных устройств.

Таблица 1

Фрагмент экспериментальной выборки

	1	2	3	4	5	6	7	…	162	163	164	165
1	0	0	0	0	0	0	0	…	1	40	70	ok
2	0	0	0	0	0	0	0	…	0	0	11	ok
3	1	0	0	0	0	0	0	…	0	95	10	virus
4	1	1	1	1	1	1	1	…	1	41	0	virus
…	…	…	…	…	…	…	…	…	…	…	…	…
95	1	1	0	0	0	0	1	…	0	42	12	virus
96	1	0	0	0	0	1	1	…	0	41	11	virus
97	0	0	0	0	0	1	1	…	0	41	10	ok
98	0	0	1	0	0	0	0	…	1	39	9	ok
99	0	0	0	0	0	0	0	…	1	38	8	ok
100	0	0	1	0	0	0	0	…	0	36	7	ok

Задача обнаружения вредоносных программ сводится к задаче классификации предложенной экспериментальной выборки. Для выбора наиболее подходящего метода классификации проведены эксперименты с применением классических (иерархическая кластеризация, К-средних, факторный и дискриминантный методы классификации) и нейросетевых (радиально-базисная функция, линейная нейронная сеть, персептрон, сеть Ворда, модульная нейронная сеть, сеть прямого распространения, сеть прямого распространения с временным окном с шагов равным 12, сеть Элмана, рекуррентная нейронная сеть, сеть Кохонена) методов классификации, а также машины опорных векторов.

Постановка задачи: Пусть X – множество программ, Y – множество, состоящее из двух классов: virus, ok. В качестве метрики выбрано Евклидово расстояние между объектами:

$$\rho(x, x') = \left(\sum_{i=1}^{n} (x - x')^2\right)^{1/2}.$$

Задана конечная экспериментальная выборка объектов:

$$X^m = \{x_1, ..., x_m\} \subset X.$$

Требуется разбить выборку на непересекающиеся подмножества, именуемые кластерами так, чтобы каждый кластер состоял из объектов, близких по метрике ρ, а объекты разных кластеров существенно отличались по этой метрике. При этом каждому объекту $x_i \in X^m$ приписывается номер кластера y_i. Для нейросетевых методов была подобрана архитектура, подходящая для решаемой задачи, были выбраны функции активации и произведено обучение нейронных сетей. Для машины опорных векторов выбрана в качестве ядра радиально базисная функция:

$$\exp(-\gamma \mid x_i - x_j \mid^2).$$

Нейронные сети и машина опорных векторов обучены на первых 68 и тестировалась на остальных 32 значениях векторов программ экспериментальной выборки, приведенной в табл. 1.

Решение задачи: Требуется определить функцию $a: X \to Y$, которая любому объекту $x \in X$ ставит в соответствие номер кластера $y \in Y$. Множество Y в некоторых случаях известно заранее, однако чаще ставится задача определить оптимальное число кластеров, с точки зрения того или иного критерия качества кластеризации.

В качестве критерия точности работы классификаторов будем использовать следующую формулу:

$$OK = \frac{ЧО \times 100}{ЧН},$$

где ОК – общий процент, как вредоносных, так и безопасных программ ошибки классификации; ЧО – число ошибок классификации; ЧН – суммарное число наблюдений.

ИССЛЕДОВАТЕЛЬСКИЙ ЭКСПЕРИМЕНТ КЛАССИФИКАЦИИ РАЗРАБОТАННОЙ ВЫБОРКИ

Для выбора походящего метода классификации был проделан исследовательский эксперимент, в результате которого выбран оптимальный метод классификации. По его результатам установлено, что точность работы классических методов классификации (иерархической кластеризации и К – средних) невысокая: ошибки I рода – 26,08%, ошибки II рода – 22,7%. Классификация с применением нейросетевых методов показала лучшие результаты по сравнению с классическими методами преимущественно к большому уровню помех, а именно – когда признаки

безопасной программы присутствуют в различном соотношении во вредоносной программе [8]. Общие результаты проделанного эксперимента представлены в табл. 2.

Таблица 2

Общая таблица результатов классификации

% ошибочных	% правильных	Ошибочно	Правильно	Всего	Классы	Тип нейронной сети
11	88,9	2	16	18	ok	РБФ
20	80	3	12	15	virus	
11,1	88,8	2	16	18	ok	Линейная
26,6	73,3	4	11	15	virus	
66,6	33,3	12	6	18	ok	Персептрон
0	100	0	15	15	virus	
100	0	48	0	48	ok	Ворда
0	100	0	52	52	virus	
61,1	38,9	11	7	18	ok	Модульная
0	100	0	15	15	virus	
33,3	66,7	6	12	18	ok	Прямого распространения
7,1	92,9	1	14	15	virus	
66,6	33,4	12	6	18	ok	Прямого распространения (12)
20	80	3	12	15	virus	
50	50	9	9	18	ok	Элмана
7,1	92,9	1	14	15	virus	
22,2	77,8	4	14	18	ok	Рекуррентная
66,6	33,4	10	5	15	virus	
26,6	73,3	4	11	15	ok	Кохонена
61,1	38,8	11	7	18	virus	
72,3	27,7	5	13	18	ok	Машина опорных векторов
100	0	0	15	15	virus	

Машина опорных векторов показала лучшие результаты по сравнению с нейронными сетями и классическими методами. Ошибочно выполнена классификация 5 типов программ, процент правильно классифицированных составил 72,3%. Количество ошибок I рода – 0%, ошибки II рода – 27,7%. В классе virus классификация была выполнена безошибочно и таким образом, что процент правильных составил 100%.

Следовательно, точность классификации предложенной выборки машиной опорных векторов выше по сравнению с другими рассматриваемыми методами, но она допускает ошибки при наличии большого количества помех.

МЕТОДИКА ОБНАРУЖЕНИЯ ВРЕДОНОСНЫХ ПРОГРАММ

На основе проделанных экспериментов предложен метод обнаружения вредоносных программ на основе машины опорных векторов. Установлено, что данный метод на фоне большого уровня помех, совершает ошибки классификации, поэтому был задействован аппарат нечеткой логики [9, 10], позволяющий выполнить дополнительную классификацию, то есть коррекцию точности работы машины опорных векторов при наличии высокого уровня помех. Метод обнаружения вредоносных программ объединяет в себе связку двух методов представленный на рисунке 1 [11].

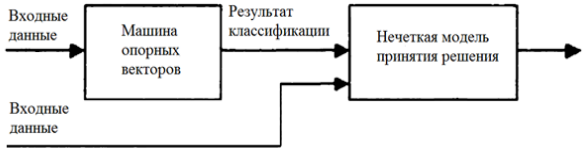

Рис. 1. Схема метода обнаружения вредоносных программ

В качестве входных данных выступает разработанная экспериментальная выборка на которой производится обучение машины опорных векторов. В состав обучающей выборки вошли 67 составленных векторов программ. Тестовая выборка включала в себя 33 вектора программ с внесением изменений с целью усложнить задачу обнаружения вредоносных программ. Далее формализуются дополнительные признаки и задаются в виде функций принадлежности для аппарата нечеткой логики. В состав функций принадлежности также входит результат классификации машиной опорных векторов. Затем формируется база правил для корректного функционирования аппарата нечеткой логики. На основании всех признаков получаем результат, выраженный в процентах и в виде критерии (безопасная, подозрительная и опасная).

Алгоритм функционирования метода обнаружения вредоносных программ состоит из следующих шагов:

1. Извлечение и формирование вектора признаков программы путем анализа файлов: androidmanifest.xml, clases.dex и системных вызовов;

2. На вход SVM-классификатора подается вектор программы;

Имеется экспериментальные данные вида: $\{(x_1, y_1), \ldots, (x_m, y_m)\}$, где каждому объекту x_i

поставлено в соответствие число y_i, принимающее знание 1 или –1 в зависимости от того, какому классу (virus или ok) принадлежит объект x_i. Каждому классу поставлен в соответствие вектор числовых значений характеристик $x_i = (x_i^1, x_i^2, ..., x_i^q)$ заданный в бинарной форме. Где x_i^j – числовое значение j-ой характеристики для i-ого объекта $(i = \overline{1,m}, j = \overline{1,q})$. Данная экспериментальная выборка использована для разработки классификатора машины опорных векторов для классификации новых объектов. Задача классификации с учетом теоремы Куна – Таккера эквивалентна двойственной задаче поиска седловой точки функции Лагранжа, которая сводиться к задаче квадратичного программирования, содержащей только двойственные переменные:

$$\begin{cases} -L(\lambda) = -\sum_{i=1}^{m}\lambda_i + \frac{1}{2}\cdot\sum_{j=1}^{m}\sum_{i=1}^{m}\lambda_i\cdot\lambda_j\cdot y_i\cdot y_j\times \\ \times\exp(-<x_i - x_j, x_i - x_j>/(2\cdot\sigma^2) - b); \\ \sum_{i=1}^{m}\lambda_i\cdot y_i = 0; \\ 0 \le \lambda_i \le C, i = \overline{1,m}. \end{cases}$$

где λ_i – двойственная переменная; x_i – объект из экспериментальной выборки; y_i – число (–1 или 1) характеризующее классовую принадлежность объекта x_i из экспериментальной выборки; $k(x_i, x_j) = \exp(-\prec x_i - x_j, x_i - x_j \succ /(2\cdot\sigma^2))$ – радиальная базисная функция ядра; C – параметр регуляризации (C>0); m – количество объектов в экспериментальной выборке; $i = \overline{1,m}$.

В результате обучения машины опорных векторов определяются опорные вектора, являющиеся векторами характеристик тех объектов x_i из экспериментальной выборки, для которых значения соответствующих им двойственных переменных λ_i отличны от нуля ($\lambda_i \neq 0$). Опорные вектора находятся ближе всего к гиперплоскости, разделяющей классы, и несут всю информацию о разделении классов. Так как задача квадратичного программирования решена, то классификация произвольного объекта λ будет выполнена по следующему правилу:

$$\alpha(z) = sign\sum_{i=1}^{m}\lambda_i\cdot y_i\cdot\exp(-<x_i - x_j, x_i - x_j>/$$

$$/(2\cdot\sigma^2) - b),$$

$$\text{где } b = <w, x_i> - y_i; w = \sum_{i=1}^{m}\lambda_i\cdot y_i\cdot x_i.$$

При этом суммирование в правиле выполняется только по опорным векторам.

3. Результат классификации и дополнительные признаки, заданные в виде функций, подаются блоку «система поддержки принятия решений»;

Система поддержки принятия решений основана на аппарате нечеткой логики, работающей по алгоритму Мамдани.

4. На основании правил производится анализ результатов;

5. Выводится результат в процентах и в виде категории.

На основе предложенного метода был разработан исследовательский прототип системы обнаружения вредоносных программ.

ИССЛЕДОВАТЕЛЬСКИЙ ПРОТОТИП СИСТЕМЫ ОБНАРУЖЕНИЯ ВРЕДОНОСНЫХ ПРОГРАММ

Разработанный исследовательский прототип модели системы обнаружения вредоносных программ представлен на рис. 2, он содержит следующие блоки:

1) программа представляет собой анализируемую программу, состоящую из архивного файла APK;

2) анализатор выполняет извлечение признаков, а также формирование вектора признаков, который будет подаваться на входы классификатора машины опорных векторов;

3) SVM-классификатор – обученная и протестированная на сформированной обучающей и тестовой выборке машина опорных векторов, выполняющая классификацию вектора признаков на два класса: virus и ok;

4) панель управления является интуитивно понятным пользовательским интерфейсом для работы с системой и содержит перечень всех необходимых инструментов для ввода признаков, генерации статистики и отчетов;

5) обучающая выборка содержит наборы векторов признаков программ, которые состоят из 100 векторов поведения программ: ok и virus, а также необходимы для обучения и тестирования машины опорных векторов;

6) система поддержки принятия решений на основе результата машины опорных векторов, дополнительно заданных признаков, характерных для того или иного типа программ, осуществляет дополнение к результату машины опорных векторов и отображает результат в процентах, а также определяет критерий опасности (безопасная, подозрительная, опасная).

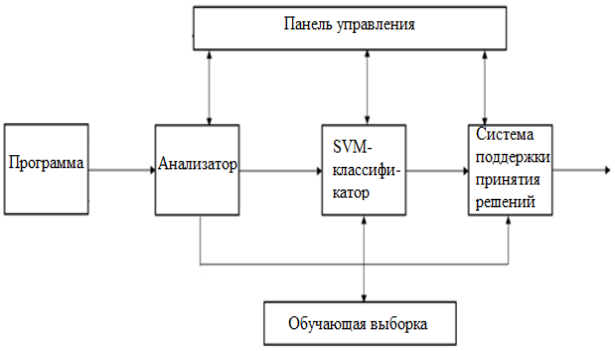

Рис. 2. Архитектура исследовательского прототипа модели системы обнаружения вредоносных программ

Исследовательский прототип модели системы обнаружения вредоносных программ способен выполнять обнаружение с учетом различных условий и помех, а также повысить эффективность обнаружения вредоносных программ.

Он реализован в виде программы на языке C++ и представляет собой программу имитирующую поведение двух типов программ, условий и помех, собирает статистику и ведут журнал событий. Главное окно программы представлено на рис. 3.

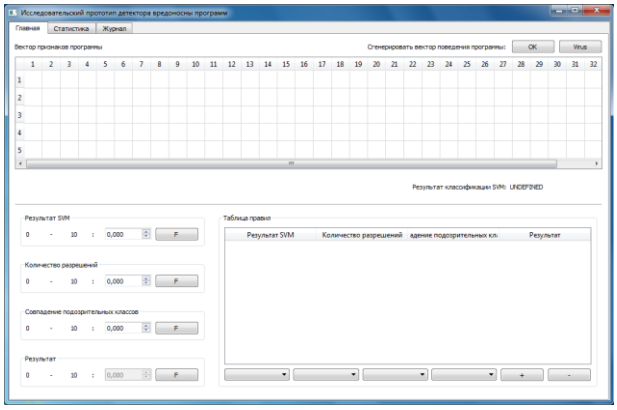

Рис. 3. Главное окно программы

С целью проверки эффективности работы разработанных методов был проведен эксперимент с классификацией 100 типов программ и сравнение полученных результатов с существующими в настоящее время антивирусными программами.

ЗАКЛЮЧЕНИЕ

Можно сделать вывод, что антивирусные программы обладают плохой эффективность обнаружения новых вредоносных программ [12], результат лучшего составил 60%, средний результат по всем рассматриваемым образцам составил 23,4%. Разработанный исследовательский прототип модели системы обнаружения вредоносных программ показал лучший результат в 80% и сравнительно небольшое количество ложных срабатываний равное в количестве 8 программ из 100 рассматриваемых образцов, что составило 19,35%, но при этом не учитывался результат работы аппарата нечеткой логики, который выполнил уточнение и коррекцию результата путем дополнительной классификации. Машина опорных векторов ошиблась, выдав вредоносную программу за безопасную, но нечеткая логика показала результат: подозрительная 45% и опасная 65%, таким образом, дополнив работу машины опорных векторов и улучшив показатели эффективности обнаружения. Следовательно, предложенный метод обнаружения вредоносных программ позволяет выполнять обнаружение путем анализа поведенческого характера программ, а также увеличить эффективность обнаружения, что положительно сказалось на защите информации в целом. Данный метод позволяет своевременно обнаружить и нейтрализовать угрозу со стороны как новых так уже имеющихся типов вредоносных программ, а также может выступать в дополнении к уже имеющимся классическим методам обнаружения.

СПИСОК ЛИТЕРАТУРЫ

1. **Жернаков С. В., Гаврилов Г. Н.** Обзор современного состояния защиты информации в мобильных системах // Вестник БГТУ им. В. Г. Шухова. 2016, № 2. С. 171–176. [S. V. Zhernakov, G. N. Gavrilov, “Overview of the current state of information security in mobile systems” (in Russian), in *Vestnik BGTU* V. G. Shukhov, Vol. 2, pp. 171-176, 2016.]

2. **Валеев С. С., Дьяконов М. Ю.** Нейросетевая система анализа аномального поведения вычислительных процессов в микроядерной операционной системе// Вестник УГАТУ. 2010, Т. 14, № 5(40). С. 190–204.

3. **Сравнения антивирусов**, DLP и других средств защиты [Электронный ресурс]. URL: http://www.anti-malware.ru/compare (дата обращения: 18.02.2016). [Comparisons antivirus, DLP and other remedies [Online], (in Russian). Avaliable: http://www.anti-malware.ru/compare]

4. **Fan Yuhui, Xu Ning.** The Analysis of Android Malware Behaviors // International Journal of Security and Its Applications. 2015. Vol. 9, No. 3 Available at: http://www.sersc.org/journals/IJSIA/vol9_no3_2015/25.pdf (Accessed 08 March 2015).

5. **Ken Dunham, Shane Hartman, Manu Quintans, Jose Andre Morales, Tim Strazzere.** Android Malware and Analysis. NY, CRCPress, 2015. 91 p.

6. **Sanz B., Santos I., Nieves J., Laorden C., Alonso-Gonzalez I., G. Bringas P.** MADS: Malicious android applications detection through string analysis. Network and System Security, Springer Berlin Heidelberg, 2011, Vol. 5, No. Available at: http://www.researchgate.net/256194745MADS (Accessed 08 March 2015).

7. **Arp D., Spreitzenbarth M., Hubner M., Gascon H., Rieck K.** DREBIN: Effective and Explainable Detection of Android Malware in Your Pocket. NDSS Symposium 2014, Switzerland, 2014, Vol. 4, No. 1 Available at: https://user.informatik.unigoettingen.de/~krieck/docs/2014-ndss.pdf (Accessed 08 March 2015).

8. **Жернаков С. В., Гаврилов Г. Н.** Детектирование вредоносного программного обеспечения с применением классических и нейросетевых методов классификации // Вестник ВГУИТ. 2015. № 4. С. 85–92. [S. V. Zhernakov, G. N. Gavrilov, "Detection of malicious software using classical and neural network classification methods" (in Russian), in *Vestnik VGUIP*, Vol. 4, pp. 85-92, 2015.]

9. **Андрейчиков А. В., Андрейчикова О. Н.** Интеллектуальные информационные системы: Учебник. М.: Финансы и статистика, 2004. 424 с. [Andreychikov A. V., Andreichikova O. N. Intelligent information systems: Textbook, (in Russian). Moscow: Finance and Statistics, 2004.]

10. **Головко В. А.** Нейронные сети: обучение, организация и применение учеб. пособие для вузов Кн. 4. М.: ИИРЖР, 2001. 178 с. [Golovko V. A. Neural networks: education, organization and application. Bk. 4: Proc. manual for schools. Moscow: IIRZHR, 2001.]

11. **Васильев В. И.** Интеллектуальные системы защиты информации: учеб. пособие. М.: 2013. 82 с. [Vasilyev V. I. Intelligent protection of information systems: Textbook. Moscow: ENGINEERING, 2013.]

12. **Зак Ю. А.** Принятие решений в условиях нечетких и размытых данных: Fuzzy-технологии. М.: ЛИБРОКОМ, 2013. 179 с. [Zach Yu Decision-making in a fuzzy and blurry data: Fuzzy-technology. Moscow: LIBROKOM, 2013.]

ОБ АВТОРАХ

ЖЕРНАКОВ Сергей Владимирович, д.т.н., проф., заведующий каф. электроники и биомедицинских технол.

ГАВРИЛОВ Григорий Николаевич, асп. каф. электроники и биомедицинских технол. Ведущий спец. по защите информ. ПАО «УМПО».

METADATA

Title: System malware detection in the type of operating system ANDROID.

Authors: S. V. Zhernakov[1], G. N. Gavrilov[2]

Affiliation:

[1] Ufa State Aviation Technical University (UGATU), Russia.

[2] Ufa Engine Industrial Association (UMPO), Russia.

Email: [2] grigorijgavrilov@mail.ru.

Language: Russian.

Source: Vestnik UGATU (scientific journal of Ufa State Aviation Technical University), vol. 20, no. 2 (72), pp. 117-122, 2016. ISSN 2225-2789 (Online), ISSN 1992-6502 (Print).

Abstract: As part of this work has been tasked to improve the efficiency of detection of malicious software in the operating system for mobile devices such as Android. To achieve this goal the analysis of security of the Android operating system and the formalization of malware samples in order to identify features inherent in their behavior. Based on the information developed an experimental sample consisting of the vectors describing the behavioral nature of the two types of programs: safe - ok and malware - virus. As a result of research experiments chosen classification method which performs classification proposed sample with the highest accuracy. The problem is increasing the efficiency of detection of malicious software is solved with the use of the developed method based on support vector machines and apparatus of fuzzy logic. This method is implemented in the form of studies of the prototype malware detection system.

Key words: malware detection system, Android, support vector machine, fuzzy logic, classification, behavioral in nature.

About authors:

ZHERNAKOV, Sergey Vladimirovich, Doctor of Technical Sciences, Professor, Head of the Department of electronics and biomedical technologies Ufa State Aviation Technical University.

GAVRILOV, Grigoriy Nikolaevich, graduate student of the Department of Electronics and Biomedical technologies. Leading expert on data protection "UMPO".