Уважаемый пользователь!
Обращаем ваше внимание, что система Антиплагиат отвечает на вопрос, является ли тот или иной фрагмент текста заимствованным или нет. Ответ на вопрос, является ли заимствованный фрагмент именно плагиатом, а не законной цитатой, система оставляет на ваше усмотрение. Также важно отметить, что система находит источник заимствования, но не определяет, является ли он первоисточником.

Информация о документе:

Имя исходного файла: дисерт-ALEX-PC_19_PR (1).docx
Имя компании: ТУСУР
Тип документа: Прочее
Имя документа: дисерт-ALEX-PC_19_PR (1).docx
Дата проверки: 04.04.2017 16:27
Модули поиска: Интернет (Антиплагиат), Диссертации и авторефераты РГБ, Модуль поиска ЭБС "Лань", Модуль поиска ЭБС БиблиоРоссика, Университетская библиотека онлайн, Коллекция юридических документов, Цитирования

Текстовые статистики:
Индекс читаемости: сложный
Неизвестные слова: в пределах нормы
Макс. длина слова: в пределах нормы
Большие слова: в пределах нормы

Источник	Ссылка на источник	Коллекция/ модуль поиска	Доля в отчёте	Доля в тексте
[1] Диссертация Биричевс...	http://www.spiiras.nw.ru/dissovet/wp-content/uploads/2015/12...	Интернет (Антиплагиат)	0	85,43%
[2] Диссертация Биричевс...	http://www.spiiras.nw.ru/dissovet/wp-content/uploads/2015/12...	Интернет (Антиплагиат)	0	17,8%
[3] Диссертация Биричевс...	http://www.spiiras.nw.ru/dissovet/wp-content/uploads/2015/12...	Интернет (Антиплагиат)	0	17,74%
[4] Диссертация Биричевс...	http://www.spiiras.nw.ru/dissovet/wp-content/uploads/2015/12...	Интернет (Антиплагиат)	0	17,71%
[5] Диссертация Биричевс...	http://www.spiiras.nw.ru/dissovet/wp-content/uploads/2015/12...	Интернет (Антиплагиат)	0	16,56%
[6] Диссертация Биричевс...	http://www.spiiras.nw.ru/dissovet/wp-content/uploads/2015/12...	Интернет (Антиплагиат)	0	12,3%
[7] Диссертация Биричевс...	http://www.spiiras.nw.ru/dissovet/wp-content/uploads/2015/12...	Интернет (Антиплагиат)	0	9,64%
[8] PDF	http://proceedings.spiiras.nw.ru/ojs/index.php/sp/issue/down...	Интернет (Антиплагиат)	4,34%	4,34%
[9] Диссертация	http://www.spiiras.nw.ru/dissovet/wp-content/uploads/2016/10...	Интернет (Антиплагиат)	0,7%	1,19%
[10] Доронин, Станислав Е...	http://dlib.rsl.ru/rsl01005000000/rsl01005378000/rsl01005378...	Диссертации и авторефераты РГБ	0,37%	1,06%
[11] Молдовян, Дмитрий Ни...	http://dlib.rsl.ru/rsl01005000000/rsl01005504000/rsl01005504...	Диссертации и авторефераты РГБ	0,18%	0,98%
[12] 50578	http://e.lanbook.com/books/element.php?pl1_id=50578	Модуль поиска ЭБС "Лань"	0,29%	0,74%
[13] Защита компьютерной ...	http://www.bibliorossica.com/book.html?&currBookId=5631	Модуль поиска ЭБС БиблиоРоссика	0%	0,73%
[14] 1122	http://e.lanbook.com/books/element.php?pl1_id=1122	Модуль поиска ЭБС "Лань"	0%	0,73%
[15] ko`chirish	http://library.tuit.uz/knigiPDF/79.pdf	Интернет (Антиплагиат)	0,01%	0,73%
[16] Защита компьютерной ...	http://biblioclub.ru/index.php?page=book_red&id=86475	Университетская библиотека онлайн	0%	0,72%
[17] Скачать ГОСТ ЭД1	http://meganorm.ru/list/all.htm	Интернет (Антиплагиат)	0,22%	0,71%
[18] 260320	http://biblioclub.ru/index.php?page=book_red&id=260320	Университетская библиотека онлайн	0%	0,69%
[19] Куприянов, Иван Алек...	http://dlib.rsl.ru/rsl01006000000/rsl01006610000/rsl01006610...	Диссертации и авторефераты РГБ	0,06%	0,66%
[20] http://izv-tn.tti.sf...	http://izv-tn.tti.sfedu.ru/wp-content/uploads/PDF/2015_5(166...	Интернет (Антиплагиат)	0,18%	0,64%
[21] Download	http://i-us.ru/Files/Pdfs/2014_5.pdf	Интернет (Антиплагиат)	0,46%	0,63%
[22] 231889	http://biblioclub.ru/index.php?page=book_red&id=231889	Университетская библиотека онлайн	0,03%	0,6%
[23] Крылов, Григорий Оле...	http://dlib.rsl.ru/rsl01004000000/rsl01004039000/rsl01004039...	Диссертации и авторефераты РГБ	0,04%	0,6%
[24] Гужва, Дмитрий Юрьев...	http://dlib.rsl.ru/rsl01004000000/rsl01004294000/rsl01004294...	Диссертации и авторефераты РГБ	0,01%	0,59%

[25] 3032	http://e.lanbook.com/books/element.php?pl1_id=3032	Модуль поиска ЭБС "Лань"	0%	0,58%
[26] Степашкин, Михаил Ви...	http://dlib.rsl.ru/rsl01003000000/rsl01003350000/rsl01003350...	Диссертации и авторефераты РГБ	0,01%	0,58%
[27] Отрицаемое шифровани...	http://ru.wikipedia.org/wiki/%d0%9e%d1%82%d1%80%d0%b8%d1%86%...	Интернет (Антиплагиат)	0,43%	0,49%
[28] Скачать/Цирлов - Осн...	http://www.e-reading.org.ua/download.php?book=134422	Интернет (Антиплагиат)	0,11%	0,49%
[29] Вехов, Виталий Борис...	http://dlib.rsl.ru/rsl01004000000/rsl01004401000/rsl01004401...	Диссертации и авторефераты РГБ	0,02%	0,49%
[30] Хартиков, Денис Миха...	http://dlib.rsl.ru/rsl01002000000/rsl01002739000/rsl01002739...	Диссертации и авторефераты РГБ	0%	0,48%
[31] Лившиц, Илья Иосифов...	http://dlib.rsl.ru/rsl01005000000/rsl01005488000/rsl01005488...	Диссертации и авторефераты РГБ	0,02%	0,48%
[32] Организация и технол...	http://biblioclub.ru/index.php?page=book_red&id=74298	Университетская библиотека онлайн	0%	0,43%
[33] 66085	http://e.lanbook.com/books/element.php?pl1_id=66085	Модуль поиска ЭБС "Лань"	0%	0,43%
[34] Твердый, Лев Вадимов...	http://dlib.rsl.ru/rsl01004000000/rsl01004177000/rsl01004177...	Диссертации и авторефераты РГБ	0,01%	0,43%
[35] 59240	http://e.lanbook.com/books/element.php?pl1_id=59240	Модуль поиска ЭБС "Лань"	0,01%	0,42%
[36] Краткий энциклопедич...	http://biblioclub.ru/index.php?page=book_red&id=58393	Университетская библиотека онлайн	0,05%	0,41%
[37] Информационная безоп...	http://www.bibliorossica.com/book.html?&currBookId=19051	Модуль поиска ЭБС БиблиоРоссика	0%	0,41%
[38] Рамзи Яхья Мохаммед ...	http://dlib.rsl.ru/rsl01006000000/rsl01006567000/rsl01006567...	Диссертации и авторефераты РГБ	0,1%	0,41%
[39] Источник 39	http://window.edu.ru/resource/099/65099/files/23.pdf	Интернет (Антиплагиат)	0,29%	0,34%
[40] 10517	http://e.lanbook.com/books/element.php?pl1_id=10517	Модуль поиска ЭБС "Лань"	0,06%	0,31%
[41] Бондарь, Виктория Ви...	http://dlib.rsl.ru/rsl01000000000/rsl01000345000/rsl01000345...	Диссертации и авторефераты РГБ	0%	0,3%
[42] 233689	http://biblioclub.ru/index.php?page=book_red&id=233689	Университетская библиотека онлайн	0%	0,3%
[43] Карпов, Андрей Влади...	http://dlib.rsl.ru/rsl01002000000/rsl01002881000/rsl01002881...	Диссертации и авторефераты РГБ	0%	0,29%
[44] 71733	http://e.lanbook.com/books/element.php?pl1_id=71733	Модуль поиска ЭБС "Лань"	0,1%	0,28%
[45] Основы электронной к...	http://www.bibliorossica.com/book.html?&currBookId=10466	Модуль поиска ЭБС БиблиоРоссика	0%	0,28%
[46] Прохоров, Павел Влад...	http://dlib.rsl.ru/rsl01002000000/rsl01002325000/rsl01002325...	Диссертации и авторефераты РГБ	0%	0,28%
[47] 253804	http://biblioclub.ru/index.php?page=book_red&id=253804	Университетская библиотека онлайн	0%	0,27%
[48] Шапченко, Кирилл Але...	http://dlib.rsl.ru/rsl01004000000/rsl01004737000/rsl01004737...	Диссертации и авторефераты РГБ	0,08%	0,26%
[49] Созыкин, Андрей Влад...	http://dlib.rsl.ru/rsl01004000000/rsl01004144000/rsl01004144...	Диссертации и авторефераты РГБ	0%	0,26%
[50] О РЕСПУБЛИКАНСКОЙ ЦЕ...	http://online.lexpro.ru/document/21314742	Коллекция юридических документов	0,25%	0,25%
[51] 5114	http://e.lanbook.com/books/element.php?pl1_id=5114	Модуль поиска ЭБС "Лань"	0,06%	0,25%
[52] О ТЕХНИЧЕСКИХ ТРЕБОВ...	http://online.lexpro.ru/document/20534966	Коллекция юридических документов	0,25%	0,25%
[53] 40849	http://e.lanbook.com/books/element.php?pl1_id=40849	Модуль поиска ЭБС "Лань"	0%	0,25%
[54] ОБ УТВЕРЖДЕНИИ ТРЕБО...	http://online.lexpro.ru/document/149635	Коллекция юридических документов	0,25%	0,25%
[55] Основы сетевой безоп...	http://www.bibliorossica.com/book.html?&currBookId=12070	Модуль поиска ЭБС БиблиоРоссика	0%	0,24%
[56] БАНКОВСКИЕ ОПЕРАЦИИ ...	http://www.bibliorossica.com/book.html?&currBookId=6114	Модуль поиска ЭБС БиблиоРоссика	0%	0,23%
[57] Атаманов, Александр ...	http://dlib.rsl.ru/rsl01005000000/rsl01005459000/rsl01005459...	Диссертации и авторефераты РГБ	0%	0,23%
[58] 252979	http://biblioclub.ru/index.php?page=book_red&id=252979	Университетская библиотека онлайн	0%	0,23%

[59] Андреев, Олег Олегов...	http://dlib.rsl.ru/rsl01004000000/rsl01004662000/rsl01004662...	Диссертации и авторефераты РГБ	0%	0,23%
[60] 227774	http://biblioclub.ru/index.php?page=book_red&id=227774	Университетская библиотека онлайн	0,19%	0,23%
[61] 10927	http://e.lanbook.com/books/element.php?pl1_id=10927	Модуль поиска ЭБС "Лань"	0%	0,22%
[62] ОБЕСПЕЧЕНИЕ ИНФОРМАЦ...	http://online.lexpro.ru/document/228348	Коллекция юридических документов	0,15%	0,22%
[63] ОБ УТВЕРЖДЕНИИ КОНЦЕ...	http://online.lexpro.ru/document/23238953	Коллекция юридических документов	0,21%	0,21%
[64] Банковские платежные...	http://www.bibliorossica.com/book.html?&currBookId=20764	Модуль поиска ЭБС БиблиоРоссика	0%	0,21%
[65] Безопасность информа...	http://www.bibliorossica.com/book.html?&currBookId=20592	Модуль поиска ЭБС БиблиоРоссика	0,01%	0,21%
[66] 65912	http://e.lanbook.com/books/element.php?pl1_id=65912	Модуль поиска ЭБС "Лань"	0%	0,2%
[67] О ПРОВЕДЕНИИ РАБОТ П...	http://online.lexpro.ru/document/23585470	Коллекция юридических документов	0,03%	0,19%
[68] ОБ УТВЕРЖДЕНИИ КОНЦЕ...	http://online.lexpro.ru/document/25140519	Коллекция юридических документов	0,11%	0,19%
[69] 59434	http://e.lanbook.com/books/element.php?pl1_id=59434	Модуль поиска ЭБС "Лань"	0%	0,19%
[70] Ястребов, Илья Серге...	http://dlib.rsl.ru/rsl01004000000/rsl01004896000/rsl01004896...	Диссертации и авторефераты РГБ	0,01%	0,19%
[71] Банковские микропроц...	http://www.bibliorossica.com/book.html?&currBookId=14696	Модуль поиска ЭБС БиблиоРоссика	0,03%	0,17%
[72] 209462	http://biblioclub.ru/index.php?page=book_red&id=209462	Университетская библиотека онлайн	0%	0,17%
[73] Информационная безоп...	http://www.bibliorossica.com/book.html?&currBookId=19052	Модуль поиска ЭБС БиблиоРоссика	0%	0,17%
[74] 59241	http://e.lanbook.com/books/element.php?pl1_id=59241	Модуль поиска ЭБС "Лань"	0%	0,17%
[75] ОБ УТВЕРЖДЕНИИ ЛОКАЛ...	http://online.lexpro.ru/document/25199353	Коллекция юридических документов	0%	0,16%
[76] ОБ УТВЕРЖДЕНИИ ПОЛОЖ...	http://online.lexpro.ru/document/24697286	Коллекция юридических документов	0%	0,16%
[77] Источник 77	http://library.tuit.uz/knigiPDF/11.pdf	Интернет (Антиплагиат)	0,16%	0,16%
[78] 4971	http://e.lanbook.com/books/element.php?pl1_id=4971	Модуль поиска ЭБС "Лань"	0%	0,15%
[79] 64921	http://e.lanbook.com/books/element.php?pl1_id=64921	Модуль поиска ЭБС "Лань"	0%	0,15%
[80] ОБ ИСПОЛЬЗОВАНИИ ЭЛЕ...	http://online.lexpro.ru/document/23191312	Коллекция юридических документов	0%	0,13%
[81] Современная компьюте...	http://www.bibliorossica.com/book.html?&currBookId=7563	Модуль поиска ЭБС БиблиоРоссика	0%	0,12%
[82] О КОНЦЕПЦИИ ФОРМИРОВ...	http://online.lexpro.ru/document/593308	Коллекция юридических документов	0,01%	0,12%
[83] Источник 83		Цитирования	0,12%	0,12%
[84] Современные операцио...	http://www.bibliorossica.com/book.html?&currBookId=12193	Модуль поиска ЭБС БиблиоРоссика	0,06%	0,12%
[85] 49646	http://e.lanbook.com/books/element.php?pl1_id=49646	Модуль поиска ЭБС "Лань"	0%	0,11%
[86] 208656	http://biblioclub.ru/index.php?page=book_red&id=208656	Университетская библиотека онлайн	0%	0,1%
[87] 275128	http://biblioclub.ru/index.php?page=book_red&id=275128	Университетская библиотека онлайн	0,03%	0,1%
[88] 63241	http://e.lanbook.com/books/element.php?pl1_id=63241	Модуль поиска ЭБС "Лань"	0%	0,1%
[89] 253557	http://biblioclub.ru/index.php?page=book_red&id=253557	Университетская библиотека онлайн	0,1%	0,1%
[90] Источник 90	http://window.edu.ru/resource/669/56669/files/k_Martemyanov....	Интернет (Антиплагиат)	0%	0,1%
[91] 210025	http://biblioclub.ru/index.php?page=book_red&id=210025	Университетская библиотека онлайн	0,09%	0,09%
[92] Введение в операцион...	http://www.bibliorossica.com/book.html?&currBookId=6135	Модуль поиска ЭБС БиблиоРоссика	0%	0,09%
[93] 90922	http://biblioclub.ru/index.php?page=book_red&id=90922	Университетская	0%	0,09%

			библиотека онлайн		
☑	**[94]** Платежные карты. Биз...	http://www.bibliorossica.com/book.html?&currBookId=14718	Модуль поиска ЭБС БиблиоРоссика	0,04%	0,09%
☑	**[95]** 69958	http://e.lanbook.com/books/element.php?pl1_id=69958	Модуль поиска ЭБС "Лань"	0%	0,09%
☑	**[96]** Основы информационно...	http://www.bibliorossica.com/book.html?&currBookId=6459	Модуль поиска ЭБС БиблиоРоссика	0%	0,08%
☑	**[97]** ОБ УТВЕРЖДЕНИИ ПОЛИТ...	http://online.lexpro.ru/document/21128437	Коллекция юридических документов	0%	0,08%
☑	**[98]** Научно-технический в...	http://www.bibliorossica.com/book.html?&currBookId=17606	Модуль поиска ЭБС БиблиоРоссика	0%	0,07%
☑	**[99]** 9552	http://e.lanbook.com/books/element.php?pl1_id=9552	Модуль поиска ЭБС "Лань"	0%	0,06%
☑	**[100]** 252894	http://biblioclub.ru/index.php?page=book_red&id=252894	Университетская библиотека онлайн	0%	0,05%
☑	**[101]** Зегжда, Дмитрий Петр...	http://dlib.rsl.ru/rsl01002000000/rsl01002614000/rsl01002614...	Диссертации и авторефераты РГБ	0%	0,04%
☑	**[102]** скачать	http://bib.convdocs.org/v2815/?download=1	Интернет (Антиплагиат)	0%	0,04%
☑	**[103]** 63099	http://e.lanbook.com/books/element.php?pl1_id=63099	Модуль поиска ЭБС "Лань"	0%	0,03%
☑	**[104]** 241054	http://biblioclub.ru/index.php?page=book_red&id=241054	Университетская библиотека онлайн	0%	0,03%
☑	**[105]** Вестник Санкт-Петерб...	http://www.bibliorossica.com/book.html?&currBookId=16046	Модуль поиска ЭБС БиблиоРоссика	0%	0,02%

Оригинальные блоки: 89,65%

Заимствованные блоки: 8,97%

Заимствование из "белых" источников: 1,38%

Итоговая оценка оригинальности: **91,03%**

Федеральное государственное бюджетное учреждение науки
Санкт-Петербургский институт информатики и автоматизации
Российской академии наук
(СПИИРАН)

[8]На правах рукописи

[26]

Биричевский Алексей Романович

Методы защиты информации на основе псевдовероятностного преобразования для мобильных устройств телекоммуникационных систем

Специальность 05.13.19 – Методы и системы защиты информации, информационная безопасность

Диссертация на соискание [9]ученой [10]степени
Кандидата технических наук
Научный руководитель
д.т.н, [9]профессор
Молдовян Н.А.
Санкт-Петербург – 2017

[31]Оглавление

[38]

Актуальность темы исследования. Большинство практических задач обеспечения информационной безопасности информационно-телекоммуникационных систем решается на программно-аппаратном уровне с использованием разнотипных операционных систем (ОС), хотя встроенные механизмы их функционирования являются весьма сходными. Встраивание в ОС механизмов аутентификации и защиты информации сокращает сроки и затраты на разработку прикладного программного обеспечения для мобильных устройств различного типа и назначения.
Средства алгоритмической защиты информации широко применяются на практике, однако их уязвимой частью является согласование параметров защитного преобразования взаимодействующими в ходе информационного обмена пользователями. Для перехвата этих параметров злоумышленник может применять принуждающие атаки, т.е. средства подкупа и специальные средства воздействия на пользователя. Для защиты пользователей от принуждающих атак предложено применение

псевдовероятностного защитного преобразования. Интеграция псевдовероятностного защитного преобразования в [8]

подсистему защиты информации универсальной ОС обеспечит пользователю системы более высокий уровень защиты от принуждающих атак.
Данная работа посвящена решению актуальных научных и практических проблем: расширения функциональности средств защиты информации, обеспечения переносимости программных средств

защиты информации на различные типы мобильных устройств (на [8]

различные типы технических платформ) и встраивания механизмов защиты от атак с принуждением.
Степень разработанности темы. Исследования методов обеспечения информационной безопасности операционных систем освящены в работах Зыль С. и Махилёва В., Оладько А.Ю., Столлингс В [1], Шаньгина В.Ф. [2], [3], Безбогова А.А. [4], Котенко И.В, Молдовяна А.А. [5], Саенко И.Б., Лорина Г. [6], Дейтеля Х.М. и др.. Вопросы защиты информации от несанкционированного доступа освящены в работах Девянина П.Н., Семкин С.Н. и др. [7] - [11]. Исследования в области разработки псевдовероятностных защитных преобразований приведены в работах Молдовяна Н.А. [12], Щербаковой В.А. [13], Березина А.Н [14] и др. [15] - [17].
Цель и задачи исследования. Цель данной работы состоит в сокращении сроков и уменьшении затрат по разработке защищенных мобильных информационных технологий за счет расширения функциональности и обеспечения переносимости программных средств

защиты информации на различные типы мобильных устройств (на [8]

различные типы технических платформ) и встраивание механизмов защиты от атак с принуждением.
Для решения поставленной цели были сформулированы и решены следующие исследовательские задачи:
выполнение анализа функциональных возможностей и особенностей реализации существующих мобильных операционных систем и на его основе разработать модель угроз информационной безопасности объекта исследования, архитектуру и программный код универсальной защищенной операционной системы для мобильных систем;
разработка метода аутентификации пользователей стойкого к принуждающим атакам;
разработка метода защитного преобразования передаваемой по открытым каналам информации, стойкого к атакам с принуждением пользователя раскрыть ключ защитного преобразования;
разработка метода защиты программного обеспечения от дизассемблирования;
разработка метода защиты, хранимой информации стойкого к атакам с принуждением пользователя раскрыть ключ защитного преобразования.
Научная новизна диссертационного исследования заключается в следующем:
Разработан метод аутентификации пользователей, отличающийся

использованием одноразовых паролей, генерируемых с помощью алгебраического алгоритма псевдовероятностного защитного преобразования.
[8]

Разработан метод защитного преобразования передаваемой по открытым каналам информации, отличающийся выполнением требования вычислительной неразличимости по шифртексту от вероятностного защитного преобразования.
Разработан метод защиты программного обеспечения от дизассемблирования, отличающийся введением ложных веток кода с помощью псевдовероятностного защитного преобразования машинного кода.
Разработан новый метод хранения ключей шифрования, отличающийся выполнением псевдовероятностного защитного преобразования ключей.
Теоретическая и практическая значимость работы. Теоретическая значимость работы состоит в разработке архитектуры универсальной защищенной мобильной ОС и новых алгоритмах защитных преобразований информации, обеспечивающих вычислительную неразличимость по шифртексту от вероятностного защитного преобразования. Практическая значимость состоит в том, что

применение универсальной операционной системы в мобильных устройствах телекоммуникационных и информационных систем, в том числе в системах защиты информации, позволит унифицировать подходы к обеспечению безопасности при разработке таких систем. Данный подход упростит разработку и производство мобильных устройств. [8]

Область применения разработанной ОС включает разработку защищенных аутентифицирующих устройств (токенов, идентификаторов), систем охраны, устройств защиты программного обеспечения, персональных устройств хранения данных (защищенных файловых хранилищ), аппаратных средств для выполнения защитных преобразований данных.
Методология и методы исследования. В работе использован

аппарат и методы математической статистики, теории вероятности, алгебры, теории чисел, [11]

криптографии и вычислительные эксперименты. Объектом исследования являются мобильные операционные системы; предметом –способы, алгоритмы и протоколы обеспечения информационной безопасности в операционных системах.
Положения, выносимые на защиту:
Метод аутентификации пользователей по одноразовым паролям, обеспечивающий защиту от принуждающего несанкционированного доступа.
Метод защитного преобразования передаваемой по открытым каналам информации, обеспечивающий защиту от атак с принуждением к раскрытию ключа защитного преобразования.
Метод защиты программного обеспечения от активного и пассивного дизассемблирования, существенно повышающий вычислительную трудоемкость дизассемблирования машинного кода.
Метод хранения ключей шифрования обеспечивающий возможность сокрытия наличия резервных серий ключей.
Степень достоверности и апробация результатов. Обоснованность научных положений, выводов и практических рекомендаций, полученных в диссертационной работе,

обеспечивается анализом состояния исследований в данной области на сегодняшний день, [9]

формальными доказательствами, вычислительным экспериментом и апробацией результатов на всероссийской научно-практической конференции с международным участием «Комплексная защита объектов информатизации и измерительные технологии» (Санкт-Петербург, 16-18 июня 2014), юбилейной

XIII Санкт-Петербургской международной конференции «Региональная информатика ([9]РИ-2012)» ([19]Санкт-Петербург, 24-26 октября 2012), [9]VI межрегиональной [20]научно-практической конференции «Информационная безопасность и [11]защита персональных данных» ([20] Брянск, 2014), VIII Санкт-Петербургской межрегиональной конференции «Информационная безопасность регионов России (ИБРР-2013)» (Санкт-Петербург, 23-25 октября 2013 г), IX Санкт-Петербургской межрегиональной конференции «Информационная безопасность регионов России (ИБРР-2015)» (Санкт-Петербург, 28-30 октября 2015 г).
[9]

Результаты диссертационной работы внедрены в учебный процесс кафедры информационной безопасности Института точных наук и информационных технологий Сыктывкарского государственного университета на старших курсах обучения студентов по специальности «090900 – Информационная безопасность».
Основные результаты диссертации изложены в 12

публикациях, в том числе, в 3 статьях опубликованы в ведущих рецензируемых журналах, входящих в перечень ВАК, в 3 [19]

докладах на международной конференции и 6 докладах на российских конференциях.
Структура и объем работы.

Диссертационная работа изложена на 152 страницах, включает 4 главы, 36 рисунок, 7 таблиц и список литературы из 112 наименований.

В [11]первой главе [34]

была рассмотрена структура объекта исследования – мобильной операционной системы.
Далее были рассмотрены особенности реализации мобильных операционных систем на примере операционных систем для смарт-карт. В каждой из представленных операционных систем были выделены особенности реализации. На основе полученных данных об архитектуре существующих мобильных систем была построена модель угроз операционной системы для мобильных систем.
Вторая глава посвящена разработке методов аутентификации пользователей, контроля и управления доступом и методов активной защиты. Была рассмотрена подсистема разграничения доступа, которая является наиболее важной в составе операционной системы.
С целью повышения эффективности функционирования подсистемы аутентификации были разработаны: алгоритм аутентификации пользователей системы, протокол аутентификации на одноразовых паролях, которые обеспечивают защиту от атак с принуждением. Данные алгоритмы предназначены для защиты пользователей от принуждающей.
Третья глава посвящена разработке методов защиты хранимой и передаваемой информации, стойких к атакам с принуждением пользователя раскрыть ключ защитного преобразования. В работе представлены разработанные алгоритмы преобразований данных, обеспечивающих защиту от атак с принуждением. Также описаны эффективные методы применения данных алгоритмов для защиты данных в операционной системе.
В четвертой главе описывается архитектура разработанной защищенной операционной системы для мобильных систем. Описываются подсистемы разработанной операционной системы, такие как методы защиты программного обеспечения от анализа, архитектуру виртуальной программной среды, резервирования, безопасного обновления и т.д. Далее был произведен анализ эффективности применения выделенных средств защиты в мобильной ОС. При оценке принималась во внимание специфика применения мобильной ОС.
Основные обозначения и сокращения
JCOP
Java Card OpenPlatform
PAM
Pluggable Authentication Modules
НДВ
недекларированными возможностями
НСД
несанкционированный доступ
ОС
операционная система
ОШ
отрицаемое шифрование
СЗИ
средство защиты информации
СКЗИ
средство криптографической защиты информации
УЭК
универсальная электронная карта
ЧПУ
числовое программное управление
Глава 1. Обзор существующих мобильных операционных систем
.
1.1 Понятие операционной системы
Операционная система — комплекс программ, который управляет ресурсами компьютерной системы, осуществляет организацию вычислительных процессов в широком смысле и обеспечивает взаимодействие

между пользователями, программистами, прикладными программами, системными приложениями и аппаратным обеспечением компьютера. [18]
[84]

Процессор имеет большое количество аппаратных ресурсов (регистры, оперативная память, порты ввода вывода). Аппаратная организация ресурсов зависит от используемой архитектуры ядра процессора. Поэтому прикладному программному обеспечению придется адаптироваться под конкретную аппаратную архитектуру. В работах [19] - [21] описаны основные требования к функциональности операционных систем. Одна из основных задач операционной системы это управление ресурсами. Операционная система предлагает прикладному программному обеспечению простой и унифицированный механизм доступа к ресурсам. В данном случае операционная система представляет собой прослойку между аппаратным обеспечением и прикладной программой (рисунок 1).
Ресурсы системы
Операционная система
Приложение №1
Приложение №2
Рисунок 1. Взаимодействие приложений с аппаратными ресурсами

Так как операционная система расположена между аппаратными ресурсами и прикладной программой, система не только может организовать унифицированный доступ к ресурсам, но и проконтролировать правомочность такого доступа.
Операционная система представляет собой достаточно сложный продукт, состоящий из множества различных функциональных подсистем. «
Наиболее общим подходом к структуризации операционной системы является разделение всех ее модулей на две группы ...» [22].
Первую группу модулей системы условно называют ядром системы. Ядро —

центральная часть операционной системы, обеспечивающая приложениям координированный доступ к ресурсам компьютера, таким как процессорное время, [44]

оперативная память, внешнее оборудование. Ядро включает в себя наиболее критичные для работы операционной системы (низкоуровневые) подсистемы (подсистемы управления памятью, обработчики системных прерываний, стеки протоколов и другие).
Вторая группа модулей выполняет вспомогательные (дополнительные или высокоуровневые) функции операционной системы. Примером вспомогательного модуля может служить модуль архивирования данных. Прямого влияния на работу операционной системы модуль архивирования не оказывает.

Для надежного управления ходом выполнения приложений операционная система должна иметь по отношению к приложениям определенные привилегии. Иначе некорректно работающее приложение может вмешаться в работу ОС и, например, разрушить часть ее кодов. Все усилия разработчиков операционной системы окажутся напрасными, если их решения воплощены в незащищенные от приложений модули системы, какими бы элегантными и эффективными эти решения ни были. [18]
[39]

Ядро системы может выполняться в привилегированном (резидентном) режиме. Вспомогательные модули ОС, в зависимости от подразделения, могут иметь различные группы приоритетов. Например, можно разделить на функциональные группы:
утилиты программы, необходимые для сопровождения операционной системы, например, как программы файловый менеджер, архиватор;
системные программы текстовый редактор, консольный интерпретатор, компилятор, отладчики;
прикладные приложения пользовательский интерфейс, калькулятор, офисные приложения;
библиотеки процедур набор, импортируемых, функций операционной системы, который используется при написании прикладных приложений (например, библиотека процедур доступа к файловой системы).
Некоторые аппаратные платформы имеют возможность реализации приоритезации процессов на аппаратном уровне. К таким платформам относятся, например, микроконтроллеры на базе платформ ARM (ARM7TDMI, CORTEX-M3 и др.) В ядре CORTEX-M3 поддерживает приоритеты процессов на аппаратном уровне:
privileged mode привилегированный режим (используется для выполнения функций ядра);
user mode пользовательский режим (используется в прикладных приложениях).
В зависимости от выбранного режима работы ядра CORTEX-M3 аппаратные (низкоуровневые) команды имеют различные права на доступ ресурсам контроллера. Например, в пользовательском режиме операция чтения из регистра состояния процессора приведет к возникновению исключения типа HardFault.
Дополнительно для выделения дополнительных групп приоритетов в ядре CORTEX-M3 возможно использовать 15 приоритетов прерываний (например, для реализации функций файловой системы). Кроме того, данное ядро имеет в своем составе специализированный модуль управления прерываниями (с учетом приоритета), который называется NVIC (Nested Vectored Interrupt Controller) контроллер приоритетных век��орных прерываний.
Структурное разделение модулей операционной системы повышает расширяемость операционной системы. В зависимости от особенностей применения операционной системы в ядро включают различный набор подсистем. В работах [23] - [26] описаны примеры применения различных архитектур ядра операционной системы, в зависимости от назначения операционной системы. Выделяют несколько основных архитектур ядра операционных систем:
монолитное ядро;
модульное ядро;
микроядро:
экзоядро;
др.
Монолитное ядро классическая архитектура ядер операционных систем.

Все части монолитного ядра работают в одном адресном пространстве.
[44]

Модульное ядро – модификация архитектуры монолитных ядер.
В данной архитектуре некоторые функции выделяются в отдельные модули (например, драйвера). Изменение аппаратной архитектуры не влияет на работу всей ОС (необходимо лишь заменить необходимый модуль).
Микроядро включает минимальный набор функций для работы с оборудованием. Второстепенные функции ОС выполняются специализированными служебными приложениями (так называемыми сервисами).

Экзоядро ядро операционной системы, предоставляющее лишь функции для взаимодействия между процессами и [39]

управления ресурсами. Данное ядро нередко применяется в гипервизорах виртуальных машин. В данном случае нет необходимости реализовывать большое количество функций (так как они также реализованы в дочерних операционных системах).
1.1.1 Механизм прерываний
Также крайне важным в рамках операционных систем является вопрос обработки системных прерываний. Согласно [18] «

прерывания представляют собой механизм, позволяющий координировать параллельное функционирование отдельных устройств вычислительной системы и реагировать на особые состояния, возникающие при работе процессора». В [77]

зависимости от аппаратной платформы реализация прерываний может быть различным. Общим остается только принцип работы. Прерывание - это событие, которое немедленно переключает процессор на выполнение специализированного участка кода - обработчика прерывания. Аппаратный модуль процессора, выполняющий операции по обработке прерываний, называют контроллером прерываний. Прерывания могут быть различного рода. Основные распространенные типы прерываний:
прерывания системного таймера (обычно применяется для выполнения периодичных операций);
прерывания интерфейсов ввода-вывода;
прерывания сброса (возвращает процессор в начальное состояние);
прерывание "критическая ошибка" (происходит при выполнении процессором "невыполнимой" команды - например ссылка на несуществующее пространство в памяти).
В контроллерах также имеет смысл использовать принцип приоритетов. В работах [27], [28] рассмотрена проблема реализации приоритезации прерываний.
Процесс выполнения прерывания в наиболее простом контроллере прерываний, имеет одинаковый приоритет всех прерываний и два режима работы ядра процессора (пользовательский и привилегированный режим). Процесс прерывания начинается с возникновения прерывания

(рис. 2). Далее контроллер прерываний сохраняет контекст (окружение) пользовательского приложения, переключает процессор в привилегированный режим и передает управление соответственному обработчику прерывания. После выполнения обработчика прерывания происходит переход процессора обратно в пользовательский (не привилегированный) режим. Далее контроллер прерываний восстанавливает контекст (окружение) пользовательского приложения и приложение продолжает «прерванное» выполнение программы.

Пользовательский режим

Привилегированный

режим

Пользовательский режим

Рисунок 2. Процесс выполнения процесса прерывания

В современных процессорах применяется дополнительная приоритезация прерываний. Приоритеты прерываний позволяют эффективно обрабатывать события параллельно возникающих прерываний. Прерывание может, как прерывать обработчики прерываний с более низким приоритетом, так и выполнятся сразу после завершения прерывания с более высоким приоритетом.

1.1.2 Понятие процесса

Следующим ключевым понятием в операционных системах является процесс. Согласно [22] «процессом является выполняемая программа, вместе с текущими значениями счетчика команд, регистров и переменных». Третья задача операционной системы это распределение ресурсов процессора между процессами (реализация многопоточного режима выполнения прикладных программ). К таким ресурсам относятся, как оперативная память и регистры процессора, так и самый значимы ресурс - процессорное время. На рис. 3 изображено распределение процессорного времени между несколькими прикладными процессами. Распределение процессорного времени может быть основано на симметричном распределении или использовать различные принципы приоритетов.

Время

Процессы

1

2

3

4

Рисунок 3. Распределение процессорного времени между процессами

Жизненный цикл процесса в системе представляет собой множество состояний, в которых процесс может находиться. «

Самую простую модель можно построить, исходя из того, что в любой процесс времени процесс либо выполняется, либо не выполняется» [1].

На рисунке 4а изображена модель процесса с двумя состояниями. Однако данная модель является крайне ограниченной. Модель с двумя состояниями не описывает в частности такие состояния как:

запуск (создание) процесса;

блокирование процесса (происходит, например, когда процесс ожидает окончания операций ввода-вывода);

завершение процесса.

Большей информативностью обладает модель процесса с пятью состояниями (рис. 4б). Данная модель процесса применяется в большинстве современных операционных систем (например, в ядре операционной системы Linux [29])

Пауза

Диспетчеризация

Не выполняется

Выполняется

а) Модель процесса с двумя состояниями

Диспетчеризация

Тайм-аут

Ожидание события

Событие

Завершение

Запуск

Новый

процесс

Готов к выполнению

Выполняется

Процесс завершен

Процесс блокирован

б) Модель процесса с пятью состояниями

Рисунок 4. Модели процесса

Модель с пятью состояниями позволяет описать весь цикл жизни процесса в операционной системе:

создание (запуск) процесса (переход из состояния "Новый процесс" в состояние "Готов к выполнению");

выделение процессорного времени процессу при диспетчеризации (переход из состояния "Готов к выполнению" в состояние "Выполняется");

переход в ждущий режим (переход из состояния "Выполняется" в состояние "Готов к выполнению");

блокировка процесса (переход из состояния "Выполняется" в состояние "Процесс блокирован") - происходит вследствие "вынужденной" приостановки программы процесса для ожидания выполнения внешнего события;

активация процесса (переход из состояния "Процесс блокирован" в состояние "Готов к выполнению") - происходит при наступлении ожидаемого события;

завершение процесса (переход из состояния "Выполняется" в состояние "Процесс завершен").

Каждая прикладная программа представляет собой набор низкоуровневых команд, которые располагаются в основной памяти процессора.

Для старта программы в качестве процесса ядру операционной системы необходимо иметь следующие данные:

указатель начала программы в основной памяти;

указатель на стек процесса (область в оперативной памяти);

размер стека процесса;

дополнительные аргументы (приоритет процесса, родитель и т.д.).

Данный набор сведений является минимально необходимым для выполнения ядром операционной системы операции переключения контекста (окружения) процесса. Переключение "окружения" процесса необходимо для организации многопоточного режима работы операционной системы. Процесс переключения контекста процесса изображен на рисунке 5.

Переключение контекста начинается с перехода процессора в привилегированный режим (системное прерывание). Ядро системы производит перенос (копирование) активного контекста (счетчик команд, регистры процессора) в область оперативной памяти (стек) соответственного активного процесса (процесс № 3). Далее выбирается процесс, который будет загружен далее (процесс № 1). После этого ядро системы копирует контекст процесса №1 из области оперативной данных (стек) в регистры процессора. На заключительном этапе происходит возврат процессора из привилегированного режима в обычный. Процессор "продолжает" выполнение процесса №3.

Центральный процессор

Оперативная память

Стек процесса №1

- счетчик команд

- значения регистров
Стек процесса №3
- счетчик команд
- значения регистров
Счетчик команд процессора
Регистры
процессора
Стек процесса №2
- счетчик команд
- значения регистров
Рисунок 5. Процесс переключения контекста

С точки зрения процесса выполнение его программы идет в последовательном (однопоточном) режиме. Каждый процесс имеет свое собственное окружение (состояние регистров, счетчик команд, стек для хранения переменных, и тд.). Данный факт является крайне полезным при обеспечении безопасности функционирования процессов. Так как регистры, счетчик команд и стек имеет фиксированное расположение в памяти (например, выход за границы стека в архитектуре процессоров ARM приведет к исключению "критическая ошибка"), остается только проследить за прямым доступом процесса к памяти. Данная проблема может быть исключена на аппаратном уровне, используя контроллер памяти (при его наличии) или запретом прямого доступа к памяти.

1.1.3 Управление памятью

Вторым по значимости ресурсом в операционной системе является быстродействующая оперативная память процессора. В мобильных устройствах количество оперативной памяти весьма ограничено. Так как абсолютно каждое прикладное приложение в системе, в каком-либо виде использует некоторое количество оперативной памяти, то эффективность использования ограниченного количества оперативной памяти напрямую влияет на функциональность системы (ограничивается количество одновременно запущенных процессов).

Повышение эффективности использования оперативной памяти может осуществляться различными методами. Один из таких методов — это анализ и оптимизация использования переменных на этапе проектирования приложения. Методы контроля эффективности использования оперативной памяти:

контроль неиспользуемых переменных (переменные, которые были задекларированы, но не были использованы в теле программы);

перенос постоянных переменных из области оперативной памяти в постоянную память (применение директивы static);

контроль полного использования массива данных (в случае, когда программист декларирует массив данных с заведомо большим количеством элементов);

контроль возврата выделенной памяти (после использования памяти выделенной методом malloc() следует освободить память методом free()).

Контроллер памяти - программный или программно-аппаратный модуль, входящий в состав ядра операционной системы, который обеспечивает контроль за использование оперативной памяти процессора. Контроллер памяти также может применяться для контроля эффективности использования оперативной памяти. Контроллер памяти применяется и для обеспечения безопасности данных процессов.

Кроме контроля эффективности использования оперативной памяти модуль контроля памяти должен обеспечивать следующие методы:

выделение области памяти;

освобождение ранее выделенной области памяти;

операции доступа к памяти (чтение, запись).

Использование контроллера памяти может помочь полностью исключить прямой доступ к оперативной памяти процессами.

Важным вопросом использования памяти является принцип организации адресации ячеек памяти. Выделяют следующие типы адресации.

Абсолютная адресация представляет собой наиболее простой способ адресации. Каждая ячейка в памяти имеет свой собственный уникальный адрес. Данный вид адресации может быть использован как низкоуровневыми приложениями (ядром, системными службами), так и пользовательскими приложениями. Минусом такого вида адресации является ограничение размера общей памяти от размера адреса (например, при размере адреса в 4 бита может быть всего 16 блоков памяти)

Сегментная адресация чаще всего используется для организации адресации к памяти внутри определенного приложения. Адрес высчитывается относительно начала сегмента (например, начала блока данных приложения).

Дополнительно выделяют следующие типы адресации:

относительная указывается смещение относительно определенного значения;

косвенная указывается адрес ячейки, в которой содержится адрес необходимой ячейки (ссылка на ячейку);

индексная порядковый номер элемента в массиве данных;

регистровая представляет собой уникальный идентификатор регистра процессора (регистра общего назначения, регистра аппаратного обеспечения).

1.1.4 Файловая система

При работе процесса иногда необходимо иметь возможность долговременного хранения данных. Для быстрого, структурированного доступа к памяти в операционной системе должна быть реализована файловая система. Согласно [30] «

под файлом обычно понимают именованный набор данных, организованных в виде совокупности записей одинаковой структуры». Для [77]

структуризации хранения файлов и организации доступа к файлам и необходима система управления файлами (файловая система).

Файловая система представляет собой набор методов и объектов (таблиц размещения файлов, таблиц свойств файлов, и т.д.) предназначенных для организации систематизированного доступа к постоянной памяти. Минимально файловая система операционной системы должна предоставлять ряд следующих операций:

стандартные операции с файлами (создание файла, удаление файла, переименование файла);

стандартные операции доступа к файлам (чтение из файла, запись в файл, чтение и запись атрибутов файла);

операции управления аппаратными устройствами хранения данных (FLASH-память, NAND-память, NOR-память и другие).

Классическая файловая система размещается в памяти в трех основных блоках (областях памяти):

загрузочный блок - область памяти, где содержатся служебные данные, которые необходимы для первичной загрузки системы;

блок хранения заголовков объектов файловой системы (иногда данную область дублируют для достижения большей отказоустойчивости);

блок хранения данных.

Для организации простого и логичного доступа к файлам в операционной системе применяют:

«простое» именование файлов - файлу присваивается дополнительное имя (обычно строковое значение) удобное для представления человеком;

применение специального типа файлов - «каталог».

«Каталог» представляет собой специальный тип файла, в котором хранятся указатели на другие файлы (аналогично «папке» в бумажном делопроизводстве). Понятие каталога позволяет применять в файловой системе полноценную иерархическую (древовидную) организацию файлов (рис. 6). Древовидная структура всегда содержит в себе хотя бы один каталог. Данный каталог называется корневым каталогом дерева файлов. Корневой каталог как правило имеет фиксированное расположение в памяти. Каждый каталог может включать в себя как файлы, так и другие (дочерние) каталоги.

Корневой каталог
Каталог №1
Каталог №2
Каталог №3

Каталог №4
Рисунок 6. Древовидная структура каталогов
Так как каждый файл поименован (имеет уникальный идентификатор), то до каждого объекта в файловой системе может быть найден уникальный (просто поочередно двигаясь по «потомкам» от корневого каталога).
Простое именование файлов в виде текстовой строки (обычно ограниченной длины) позволяет еще больше упростить построение пути до объекта файловой системы. Например, путь до каталога №4 будет выглядеть следующем образом:
Корневой каталог / Каталог №1 / Каталог №4
Древовидная система каталогов совместно с простым именованием фалов представляет достаточно эффективную структуру. В дереве легко реализуются быстрые алгоритмы сортировки и поиска. Различные ветви дерева каталогов могут быть на различных физических носителях.
1.2 Существующие операционные системы
Наиболее подходящими и близкими к теме работы являются операционные системы, предназначенные для использования в смарт-картах. В данной главе будут рассмотрены наиболее популярные операционные системы для смат-карт:
CardOS (C) Siemens AG [31];
операционная система смарт-карты проекта УЭК (универсальной электронной карты) [32];
Java Card OpenPlatform (JCOP) [33];
MULTOS [34];
ОС Магистра [35].
К сожалению, основная часть производителей вышеуказанных операционных систем крайне скудно описывают особенности архитектуры своих продуктов. Это связанно с высокой конкуренцией среди средств защиты. Все дальнейшие данные предоставляются на основании информации представленной в инструктивных материалах, которые доступны на официальных сайтах производителей.
Ввиду предназначения, большинство операционных систем для смарт-карт реализуют функции международного стандарта ISO/IEC 7816. Стандарт состоит из 14 основных частей:
описывает физические параметры карт (Physical characteristics);
описывает расположение и назначение контактов (Cards with contacts — Dimensions and location of the contacts);
описывает

электрические параметры интерфейса и некоторые принципы установления связи для карт с [94]

асинхронным интерфейсом (Cards with contacts — Electrical interface and transmission protocols);
описывает протокол обмена и механизм действия команд (Organization, security and commands for interchange);
специфицирует процедуру регистрации в регулирующих органах идентификатора приложения (Registration of application providers).
специфицирует номера тэгов и формат записи для популярных типов данных: имен, дат, фотографий, биометрики и т. п. (Interindustry data elements for interchange);
специфицирует специализированный язык запросов;
специфицирует формат команд для доступа к криптографическим процедурам и менеджменту криптоключей. (Commands for security operations);
специфицирует формат команд для доступа к файловой системе карты. (Commands for card management);
специфицирует назначение контактов и принципы установления связи для карт с синхронным интерфейсом. (Electronic signals and answer to reset for synchronous cards);
специфицирует методы биометрической аутентификации;
специфицирует назначение контактов и принципы установления связи для карт с интерфейсом USB (

Cards with contacts — USB electrical interface and operating procedures);
[71]

специфицирует команды управления приложениями;
специфицирует криптографические функции (Cryptographic information application).
Реализация стандарта ISO/IEC 7816 позволит использовать операционную систему в самых популярных платежных системах. Имеется и локализация данного стандарта - ГОСТ Р ИСО/МЭК 7816. [36] - [40]
1.2.1 CardOS Siemens AG
Операционная система CardOS Siemens AG [31] разработана одноименной компанией Siemens и в настоящее время широко применяется (к примеру, в средствах аутентификации компании Aladdin). Текущая версия операционной системы - 4.2. Данная операционная система поддерживает большое количество специализированных контроллеров.
Отличительной особенностью операционной системы является ее тесная взаимосвязь с аппаратной платформой (в части реализации механизмов обеспечения безопасности) Командный интерфейс CardOS полностью поддерживает стандарт ISO 7816-4. Файловая система операционной системы защищается аппаратно-реализованными механизмами:
аппаратно-защищенные области памяти;
динамическое управление памятью (оптимизация EEPROM);
защита памяти от дефектов и ошибок.
В операционной системе реализовано большое количество криптографических алгоритмов RSA, SHA-1, Triple-DES (CBC), DES (ECB, CBC), MAC, Retail-MAC (большинство реализовано в контроллере на аппаратном уровне). Аппаратная реализация механизмов обеспечения безопасности, которая так широко применяется в CardOS, имеет ряд положительных особенностей:
относительно высокая скорость выполнения вычислительно сложных операций (например, криптографических операций);
защита от отладки приложений.
1.2.2 Операционная система смарт-карты проекта УЭК
УЭК (универсальная электронная карта) представляет собой универсальный российский электронный носитель, который используется для идентификации пользователя. Согласно [41]

универсальная электронная карта представляет собой материальный носитель, содержащий зафиксированную на нем в визуальной (графической) и электронной (машиносчитываемой) формах информацию о пользователе картой и обеспечивающий доступ к информации о пользователе картой, используемой для удостоверения прав пользователя картой на получение государственных и муниципальных услуг, а также иных услуг, в том числе для совершения в случаях, предусмотренных законодательством Российской Федерации, юридически значимых действий в электронной форме.

Специально для данного проекта была разработана как аппаратная платформа - чип в формате смарт-карты с поддержкой российских криптографических алгоритмов.
Технические характеристики аппаратной платформы:
защищенный 8-разрядный микроконтроллер MIK51 (частота до 30МГц);
Объем EEPROM - 72K;
Криптографические сопроцессоры блочных алгоритмов (ГОСТ 28147-89, DES, 3DES, AES);
Криптографический сопроцессор модулярной арифметики для вычисления и проверки ЭЦП по алгоритмам ГОСТ Р34.10-2001, ECDSA, RSA;
Поддержка протокола Mifare (протокол беспроводных меток).
Универсальная электронная карта также имеет и специально разработанную для данного проекта операционную систему для мобильных систем. Ядро операционной системы УЭК поддерживает различные аппаратные платформы, такие как MIK51 или 32-битные системы.

Архитектура ОС изображена на рисунке 7.
В качестве среды выполнения прикладных программ используется известная платформа java (адаптированная для использования в системах с ограниченным количеством системных ресурсов).
Рисунок 7. Архитектура операционной системы УЭК
Программы на данной платформе представляют собой специальные выполняемые сценарии (апплеты). Данное решение имеет как положительные, так и отрицательные стороны. К положительным можно отнести:
популярный, известный язык программирования,
простой механизм апплетов,
возможность портирования (адаптации) апплетов с других операционных систем.
Для обеспечения положительных особенностей платформы java придется пожертвовать производительность. Механизм апплетов очень удобен для программиста, но для выполнения сценария необходимо выполнять преобразование в машинный код (компиляцию) либо каждый раз до выполнения приложения, либо в режиме реального времени. И первый и второй вариант приведет к неизбежному увеличению времени выполнения приложений. Конечно для не очень сложных приложений для смарт-карт, где многие функции реализованы аппаратно, скорость выполнения не является самым важным параметром операционной системы. Однако не самая эффективна в плане использования ресурсов платформа java отнимает вычислительные мощности, которые могут быть использованы для реализации большего количества функций конечного продукта при одинаковой аппаратной платформе.
2.2.3 Java Card OpenPlatform (JCOP)
Одна из наиболее популярных операционных систем которая применяется в смарт-картах. Данная операционная система для смарт-карт не зря является очень популярной. В ее состав входят большое количество полезных подсистем. Очень большое внимание разработчики JCOP уделили вопросу отказоустойчивости. Так как смарт-карта существует в очень неблагоприятных условиях (неожиданные отключения питания, статическое электричество), необходимо чтобы операционная система имела механизмы резервирования всех жизненно важных данных. В JCOP представляет собой виртуальную среду выполнения (виртуальную машину) java. О положительных сторонах JAVA уже упоминалось в предыдущем разделе. Прикладные программы также представляют собой сценарии на языке java (апплеты). Применение виртуальной среды разработки позволяет производить резервирование данных. По информации из руководства администратора операционной системы [42]: «большинство информации которая используется в операционной системе зарезервирована в энергонезависимой памяти». Данный факт позволяет при неожиданном отключении (к примеру, в случае, когда пользователь намерено отключил смарт-карту от платежного терминала) избежать значительного повреждения операционной системы и потери пользовательских данных.
2.2.4 MULTOS AG
Особенностью файловой системы является занимательная реализация файловой системы. А точнее наличие в файловой системе файловых объектов различных типов, которые применяются в зависимости от поставленных целей. На рисунке 8 изображена структура различных типов файлов операционной системы MULTOS AG (оригинальное изображение [34]).
Рисунок 8. Типы файловых объектов операционной системы MULTOS AG
В операционной системе представлены следующие типы файлов.
«Сырой файл» представляет собой файловый объект в виде непрерывного блока данных определенного размера. Размер файла может быть увеличен. Доступ к сохраненным данным осуществляется с применение функций бинарного (побайтного) доступа к файловой системе.
«Файл фиксированного размера» - это файл, поделенный на определенное количество блоков данных одинакового размера. Размер блока данных обычно обуславливается стандартным блоком записи на устройство хранения (например, карты памяти с файловой системой FAT32 поддерживают блоки размером 512 байт).
«Файл линейно переменного размеры» состоит из блоков различного размера. Данный вид файлов может быть полезен при записи переменных различного размера (например, переменных различных типов - float, int, byte и т.д.). Минусом данного типа файлов является нерациональность хранения данных (файл содержит достаточно большое количество служебной информации).
«Циклический файл» является одним из наиболее интересным. Для организации журналов фиксированного размера целесообразно использовать файлы данного типа. Данный тип фалов представляет собой блок данных фиксированного размера. Запись в данный файл ведется с периодической перезаписью более старых блоков.
Применение каждого из вышеуказанных типов файловых объектов в конкретных случаях может быть достаточно эффективным. Крайне перспективным может считаться «Циклический файл». Так как без применения различного типа журналов нельзя представить не одной операционной системы, возможность организовать эффективный журнал просто создав файл на диске является крайне полезной.
Для доступа к файлам смарт-карты (чтения, обновления) используется специализированная системная служба (Data Unit в терминологии операционной системы MULTOS AG). Данное решение является очень удачным. Организация доступа к файлам, расположенным на различных носителях, является ресурсоемким процессом. Более подробно о особенностях организации доступа файловой системе в мобильных операционных системах будет рассказано далее.
Для защиты хранимых в смарт-карте данных (кода программ, флэш-памяти, энергонезависимой памяти) в MULTOS AG предусмотрены специализированные меры защиты данных. На рисунке 9 (оригинальное изображение [34]) изображена логическая прикладного программного продукта в контексте операционной системы MULTOS AG.
Рисунок 9. Логическая схема прикладного приложения в операционной системе MULTOS AG
По заверению производителя ОС каждое прикладное приложение имеет свое собственное исполняемое пространство. К пространству приложения относится:
код программы расположен в статическом защищенном пространстве (пространство кода не может быть прочтено или записано);
данные приложения располагаются в защищенном пространстве (данные могут быть прочитаны и записаны, но не выполнены);
временные оперативные данные (данные сессии) располагаются в оперативной памяти.
Пространство выполнения каждого приложения защищено от других процессов защитным «экраном», который производит фильтрацию запросов на доступ к пространству процесса. Анализ поступающих запросов необходим в связи с тем, что у операционной системы нет возможности полной изоляции прикладного процесса от внешнего мира (процесс в любом случае необходимо вести обмен портами ввода-вывода и общими ресурсами). На рисунке 10 (оригинальное изображение [34]) изображена архитектура операционной системы MULTOS AG.
Рисунок 10. Архитектура операционной системе MULTOS AG
Операционная система имеет несколько функциональных уровней (в порядке возрастания уровня):
физическая платформа (функции на этом уровне представляют собой низкоуровневые команды процессора);
операционная система (ОС обеспечивает универсальные коммуникации, управление памятью, виртуальную машину);
модуль абстракции приложения модуль реализует интерфейс доступа к ресурсам операционной системы (API функции);
модуль языка MULTOS (реализует специализированный язык для написания аплетов, поддерживает язык C и Java);
Сертификат загрузки приложения (необходим для загрузки подписанного приложения);
Специализированный защитный «экран» (в терминологии производителя firewall);
Прикладные приложения.
Для защиты операционной системы от выполнения потенциально опасного программного кода в MULTOS AG применяются специализированный защитный «экран», виртуальная среда выполнения прикладных программ и система проверки подписи производителя программного продукта до запуска приложения.
2.2.5 ОС Магистра AG
Операционная система Магистра AG разработана ООО "СмартПарк". Текущая версия операционной системы - 1.3. Согласно информации из руководства программиста [35] операционная система Магистра AG имеет следующие отличительные особенности:
поддержка Российской криптографии;
поддержка контактного интерфейса ISO 7816 (протокол T0);

поддержка файловой системы на основе группы стандартов ISO 7816;
поддержка механизма расширений.
ОС реализована на микроконтроллере ST23YL18 производства компании STMicroelectronics. Приложение смарт-карты разрабатывается как совокупность структурных элементов (файлов, директорий) для хранения разнообразных данных, доступ к которым (запись, чтение, использование) определяется самим разработчиком [35].
Как уже упоминалось ранее, в виду того, что операционные системы для мобильных систем функционируют в неблагоприятной среде, существует большая вероятность возникновения нештатных ситуаций (например, вследствие неожиданного отключения электропитания). Резервирование должно являться неотъемлемой частью мобильной системы. Однако не существует ни одного стопроцентно гарантированного механизма резервирования. В случае возникновения ошибок в механизмах резервирования необходимо определить факт появления повреждений операционной системы. Для этих целей применяются механизмы самоконтроля и контроля целостности. Кроме того, контроль целостности критичных компонентов операционной системы позволяет обнаружить результаты злоумышленных действий нарушителя.
Отличительной особенностью данной операционной системы является наличие подсистемы самоконтроля. В ОС карты предусмотрены ряд автоматических (не отключаемых) и дополнительных (запускаемых вручную) средств контроля исправности карты и целостности данных. К автоматическим средствам контроля относятся:
самотестирование карты при старте;
аппаратный механизм коррекции однократных ошибок в EEPROM и детектирования многократных ошибок;
программно-вычисляемая контрольная сумма заголовков файлов, содержимого служебных файлов и пользовательских данных;
механизм буферизации записи и поддержки транзакций. [35]
При невозможности восстановить целостность аппаратно-программной среды исполнения, карта операционная в зависимости от настроек может производить переход в специализированный режим ожидания или производить принудительный сброс.
1.3 Описание нарушителя
Под безопасностью информации (согласно [43]) понимают

состояние защищенности информации [данных], при котором обеспечены [36]ее [[85]их] конфиденциальность, доступность и целостность.
[36]

Целью любого нарушителя безопасности является воздействие на состояние защищаемой информации (нарушение конфиденциальности, доступности и целостности информации). Перед выработкой необходимых мер защиты крайне необходимо иметь сведения о возможностях и умениях потенциального нарушителя.
Для построения как можно более стойкой системы защиты необходимо принять, что вероятный нарушитель информационной безопасности обладает следующими качествами:
нарушитель является специалистом во всех требуемых областях знаний (информационных технологиях, методах защиты информации, и т.д.);
нарушитель имеет знания о применяемой системе защиты информации;
нарушитель имеет достаточно высокие материальные возможности;
нарушитель имеет доступ к новейшим достижениям современной техники.
Наиболее простой классификацией нарушителей информационной безопасности является деление злоумышленников по положению относительно защищаемой информации:
внутренний нарушитель,
внешний нарушитель.
Не секрет, что парой действия внутреннего (легитимного или доверенного) пользователя могут нанести защищаемой информации больший урон нежели внешний нарушитель. Ввиду данного факта следует понимать, что большинство подсистем системы защиты потенциально могут стать подвергнуты атаке со стороны нарушителя.
В качестве нарушителя может выступать не только отдельный человек, но организованная группа лиц. В данную группу могут входить внутренние нарушители (доверенные пользователи)
1.4 Угрозы безопасности операционной системы
Определение уязвимых мест операционной системы является важной задачей процесса оценки ее уровня защищенности. Возможны различные подходы к определению критериев защищенности операционной системы ([44] - [46]). В данной работе будут определены актуальные угрозы для мобильной операционной системы и обозначены методы защиты нейтрализующие данные угрозы.
Угрозы безопасности в общем случае можно классифицировать по самым разным признакам. Согласно [47] выделяют следующие основные направления.
По природе возникновения выделяют искусственные и естественные угроз.

По степени преднамеренности выделяют случайные и преднамеренные угрозы.

По степени воздействия выделяют пассивные и активные угрозы. К [28]

пассивным угрозам можно отнести анализ журналов операционной системы. К активным можно о��нести реализацию вирусной атаки.
Можно привести и другие примеры классификации угроз [48], но наиболее популярной является классификация по методу нарушения состояния безопасности информации (см. определение в предыдущем разделе):

угрозы нарушения конфиденциальности информации;
угрозы нарушения целостности информации;
угрозы нарушения [60] доступности информации.

По принципу воздействия на операционную систему выделяют [49]:

использование известных (легальных) каналов получения информации (например, [12]

несанкционированное чтение файла пользователем вследствие неверной настройки прав доступа);
использование скрытых каналов получения информации (например, недокументированные возможности программного обеспечения);
создание дополнительных собственных каналов получения информации (например, посредствам внедрения программных закладок).
По типы используемой злоумышленником уязвимости защиты выделяют:
неадекватная настройка политики безопасности ОС;
ошибки и недокументированные возможности программного обеспечения;
внедренная программная закладка.
Далее в работе будут рассмотрены угрозы безопасности, которые характерны для исследуемого объекта (операционной системы).
1.4.1. Неправомерный доступ к ресурсам
Операционная система может оперировать большим количеством различных ресурсов (процессорное время, оперативная память, объекты файловой системы). Любой ресурс имеющий ценность для нарушителя может стать его целью. Правилами разграничения доступа в операционной системе может быть обеспечено разграничение доступа пользователей к ресурсам системы. В классической операционной системе предусмотрено ограничение доступа пользователей к системным файлам, которые необходимы для работы операционной системы. Для организации контроля доступа к защищаемым объектам прикладных программных продуктов в операционных системах реализуют различные механизмы разграничения доступа. Нарушение установленных ограничений с целью получить доступ к защищаемым ресурсам

является одной из наиболее актуальных угрозой безопасности операционной системы.

К атакам, которые позволяют осуществить неправомерный доступ к ресурсам операционной системы, можно отнести:

атаки на подсистему аутентификации (подбор пароля, подбор по словарям, перехват паролей специальными программными средствами кейлогерами);

атака превышения полномочий (путем использования ошибок программного обеспечения, применения вредоносных программ и др.).

1.4.2. Анализ (отладка) подсистем операционной системы

Анализ работы операционной системы может дать злоумышленнику данные для дальнейших атак на операционную систему. Операционная система имеет в своем составе большое количество потенциально подверженных анализу объектов. К таким объектам могут относиться:

конфигурационные данные подсистем ОС;

журналы работы ОС и прикладного программного обеспечения;

программный код программ и функциональных частей ОС;

оперативно и постоянно хранимые данные.

К атакам анализа ОС можно отнести атаки следующего типа:

сканирование файловой системы (оперативной памяти);

сборка мусора (анализ данных, подлежащих уничтожению);

анализ программного обеспечения.

Отладка приложения представляет собой процесс анализа программного продукта специализированными средствами с целью устранения ошибок. [6]

Невозможно представить процесс разработки программного продукта (любой сложности), в котором отсутствует этап отладки приложения. Исправление ошибок и доработка программных продуктов является частью жизненного цикла любого приложения. Существует большое количество программных и программно-аппаратных средств, предназначенных для отладки приложений. Но и как многие другие полезные инструменты, средства отладки имеют двойное назначение. Злоумышленник может использовать средства отладки (анализа) приложений для различных целей:

для получения эксклюзивных алгоритмов, реализованных в исследуемом приложении;

для анализа системы защиты приложения с целью поиска уязвимых мест;

для внедрения в алгоритм приложения вредоносного кода.

В соответствии с алгоритмом проведения анализа программного продукта выделяют следующие средства:

дизассемблеры - реализуют статические методы отладки программных продуктов;

отладчики - реализуют методы активного анализа приложений (отладка в процессе выполнения программы);

эмуляторы - комплексные средства отладки (анализа), которые реализуют виртуальное окружение приложения.

Средства анализа приложений также могут строится и по комбинированной схеме.

Статический метод анализа приложений основан на изучении низкоуровневых (машинных) команд приложения. Статический анализатор кода исследует собранное (скомпилированное) приложение, обнаруживает типичные блоки кода (циклы, условия, и т.д.), находит логические связи между блоками кода. На выходе злоумышленник получает простой для понимания код программы или блок схему ее работы.

Современный программист редко использует для написания сложных и функциональных приложений машинный язык программирования. Для таких задач используются специализированные автоматизированные среды разработки программных продуктов. Среды разработки предоставляют разработчикам все больше возможностей таких, как мастера построения типизированных проектов и многофункциональные программные библиотеки (например, библиотека Microsoft .Net framework). Программные библиотеки содержат наиболее типичные блоки кода. Современные программные библиотеки крайне обширны и охватывают различные области применения.

Широкое использование программных библиотек приносит пользу не только программистам, но и злоумышленникам. Современные языки программирования представляют собой "иерархические" структуры алгоритмов, такие как функции, макросы или библиотеки. По этой причине в программной библиотеке большое количество "постоянных" блоков кода (блоков с предопределенном кодом). Ввиду данного факта, зная тип библиотеки, на которой написан программный продукт, злоумышленник может с небольшими усилиями восстановить исходный код исследуемой программы.

Активные методы анализа приложений обычно используются для анализа особо сложных блоков кода. Активные методы более информативны чем статический. Принцип действия отладчиков представляет собой пошаговое выполнение исследуемого приложения. Во время выполнения отладчик исследует и внешнее окружение программы (регистры процессора, обращения к файловой системе, значения переменных в оперативной памяти).

Еще одной крайне функциональной особенностью активного метода отладки является возможность изменения внешнего окружения исследуемого программного продукта. Например, отладчик может поменять значение некоторой переменной в оперативной памяти при в ходе в определенную функцию и посмотреть на реакцию программы. Данный факт дает злоумышленнику очень большие возможности и может позволить обойти даже самые сложные системы защиты. По такому принципу взламываются многие коммерческие программы. По этой причине производители многое производители программных продуктов вносят в систему защиты своих продуктов элементы неопределенности (например, активацию программы через сеть интернет). Даже если злоумышленнику известен алгоритм аутентификации, он может не иметь сведений о преобразованиях на стороне сервера.

Эмуляторы позволяют злоумышленнику создать вокруг исследуемого программного продута собственное "виртуальное" окружение. Тем самым нарушитель может "заставить" приложения выполнять некоторые действия. Стратегий применения эмуляторов огромное количество. Существуют как самые простые эмуляторы (одной определенной подсистемы), так и эмуляторы, полностью воспроизводящие среду функционирования программы.

Все вышерассмотренные методы анализа программ могут быть использованы в связке друг с другом. Угроза анализа подсистем операционной системы является одной из наиболее важных.

1.4.3. Методы атак на криптографическую подсистему

Анализ надежности уязвимостей криптографических алгоритмов является не тривиальной задачей. В [50] выделяют четыре основных формальных метода анализа криптографических алгоритмов:

моделирование и проверка работы протокола;

создание экспертных систем для отладки различных нестандартных режимов работы алгоритмов;

моделирование требований к семейству криптопротоколо��;

разработка формальных моделей, основанных на алгебраических свойствах криптографических систем.

Вышеуказанные алгоритмы позволяют проанализировать работу криптографических алгоритмов.

Целью атаки на криптографическую подсистему операционной системы могут являться:

секретные данные;

нахождение секретного ключа.

В первом случае злоумышленник получит доступ к конкретному секретному сообщению. Получив ключевую информацию, злоумышленник получит возможность перехватывать все зашифрованные сообщения. Следующие подсистемы могут быть интересны для злоумышленника:

защищенные объекты виртуальной файловой системы;

криптографические контейнеры пользователей, которые могут содержать ключевую информацию и другие важные данные;

данные передаваемые через подсистему передачи данных.

Рисунок 11. Пассивный перехват зашифрованных данных

В [51] предлагается классифицировать атаки на алгоритмы шифрования в зависимости от набора информации, которая имеется у злоумышленника. К первой категории относятся атаки, при которых криптоаналитик имеет возможность пассивного прослушивания канала связи (рисунок 11). В данном случае злоумышленник имеет доступ шифртексту. Данный вид атаки называется атакой с известным

шифртекстом. Во второй категории атак целью криптоаналитика является ключ шифрования. В данном случае криптоаналитик обладает шифратором с введенным ключом и может проводить криптографические преобразования с помощью шифратора (рисунок 12).

Рисунок 12. Активное воздействие на шифратор

В зависимости от[данных, которые криптоаналитик использует для анализа, выделяют следующие виды атак:

атака с известным открытым ключом;

атака с выбранным открытым текстом;

адаптивная атака с выбором открытого текста;

атака с выбором шифртекста;

адаптивная атака с выбором шифртекста.

Так как мобильная операционная система выполняется на завершенных аппаратных платформах, актуальность имеют атаки на криптосистему основанные на ошибках вычислительной платформы. Так в [52] отмечается, что алгоритм симметричного шифрования ГОСТ 28147-89 является устаревшим алгоритмом шифрования из-за его подверженности атакам на основе аппаратных ошибок.

1.4.4. Нарушение работоспособности операционной системы

Операционная система представляет собой сложный программный продукт. Неработоспособность операционной системы может быть вызвана целым рядом различных причин. Начиная от программных ошибок, допущенных на этапе разработки операционной системы и заканчивая преднамеренными действиями злоумышленника (например, удалением или модификацией системных файлов ОС).

К характерным угрозам, нацеленным на нарушение работоспособности ОС, относятся:

программные закладки и вирусы;

программные ошибки (например, ошибки чрезмерного использования ресурсов операционной системы);

аппаратные ошибки (исключительные особенности работы конкретной аппаратной платформы).

1.4.5. Недекларированные

возможности программного обеспечения

[28] Согласно [53]

под недекларированными возможностями понимается функциональные возможности ПО, не описанные или не соответствующие описанным в документации, при использовании которых возможно нарушение конфиденциальности, доступности или целостности обрабатываемой информации.

В

операционной системе проблема НДВ в прикладном (клиентском) программном обеспечении имеет очень большое значение. Так как данное программное обеспечение разрабатывается сторонними производителями, имеется большая вероятность появления непредвиденных ошибок при выполнении. Также имеется возможность, что НДВ может быть организовано производителем преднамеренно с целью завладения пользовательскими данными.

1.5 Постановка задачи

В данной главе была рассмотрена структура объекта исследования. Операционная система представляет собой программный продукт, предназначенный для эффективного управления вычислительными ресурсами. К задачам операционной системы также относится организация псевдо параллельного выполнения нескольких прикладных программных продуктов (многозадачности). Ключевым понятием в понимании принципа организации многозадачности является прерывания. Прерывая работу прикладной задачи, операционная система производит сохранение окружения (регистров и оперативной памяти) и дальнейшую передачу управления другой прикладной программе. Задачами управления переключениями процессов, контроля использования оперативной памяти и организации протоколов доступа к аппаратным ресурсам занимается специализированный модуль операционной системы ядро операционной системы. От архитектуры ядра (центральной части операционной системы) зависит сфера применения ОС.

Большое количество элементов, которые взаимодействуют друг с другом, в процессе работы операционной системы (части ядра, системные службы, драйвера аппаратных устройств и прикладные процессы), делают операционную систему идеальным объектом для исследования методов защиты информации. Основная задача данной главы выделить наиболее актуальные для мобильной операционной системы угрозы информационной безопасности.

Были рассмотрены особенности реализации мобильных операционных систем на примере операционных систем для смарт-карт. Каждая из представленных операционных систем имеет в себе особенности реализации. Основной функцией операционной системы для смарт-карт является обеспечение функционирования протоколов работы с картой, которые описаны в стандарте ISO/IEC 7816. Данные протоколы взаимодействуют с криптографическими контейнерами, которые могут хранится как во внутренней (защищенной) памяти микроконтроллера, так и на внешнем хранилище данных. Для реализации защищенного хранилища криптографических контейнеров производители смарт-карт применяют два основных подхода:

реализация смарт-карты на основе специализированного микроконтроллера,

использование универсального микроконтроллера совместно с дополнительными мерами защиты.

Оба вышеуказанных подхода имеют свои положительные и отрицательные качества. В первом случае производитель применяет специализированные микроконтроллеры, которые имеют в своей архитектуре защищенное хранилище данных (пример смарт-карты проекта УЭК). Чаще всего данные хранилища представляет собой специализированный блок, который включает в себя собственно хранилище (энергонезависимая память или флэш-память) и контролер управления памятью. В задачи контроллера памяти входит управление хранением данных и контроль доступа к данным. Встроенный блок защищенного хранения данных имею сравнительно высокую производительность. Отрицательной использования специализированных контроллеров является привязка операционной системы к одной конкретной аппаратной платформе, что делается функции устройства зависимыми от аппаратной платформы.

Второй подход использует, например, электронные идентификаторы компании «Актив» Рутокены. [54] В аппаратной архитектуре данных устройств применяются универсальные микроконтроллеры архитектуры ARM. Для реализации защищенного хранилища на базе универсального контроллера необходимо применять дополнительные методы защиты данных. Это обусловлено тем, что универсальные контроллеры нацелены на большой круг устройств и имеют в своем составе средства диагностики и отладки. Не редко универсальные микроконтроллеры имею в своем составе и блоки защиты данных (например, MPU блок защиты данных в контроллерах фирмы STM Electronics). Но, к сожалению, такие средства защиты обычно имеют ограниченный функционал. Обычно они имеют только механизмы дискреционного разграничения доступа в ограниченном исполнении. Для данных в универсальных контроллерах применяют защищенные виртуальные файловые системы. Данные в памяти хранятся в зашифрованном виде.

Механизмы доступа к памяти также имеют важное значение. Так, например, MULTOS AG. имеет в своем составе подсистему контроля использования ресурсов системы прикладными процессами. Многое производители концертирует ресурсы системы в файловой системе. Файловая система является наиболее удобной структурированной иерархической структурой, которая может включать в себя объекты различного типа (файлы каталоги, ссылки на функции и т.д.). Реализация файловой системы, представленной в операционной системе MULTOS AG также имеет особенный интерес. В структуре файловой системы включены специализированный файловые объекты такие, как «цикличный файл».

В ходе работы над моделью угроз мобильной операционной системы были выделены актуальные угрозы мобильной операционной системе. Угрозы сгруппированы в классы. В соответствии с данными классами угроз будет производиться дальнейшая разработка структуры системы защиты мобильной операционной системы.

Операционные системы успешно применяются в средствах защиты информации. Многие классы средств защиты на аппаратном уровне имеют весьма схожее строение. Персональные идентификаторы (USB-токены, например, Rutoken компании Актив [54]) и ключи защиты программных продуктов (например, HASP фирмы Aladdin [55]) имеют в своей аппаратной платформе универсальный микроконтроллер с

интерфейсом USB. Чуть более сложную структуру имеют продукты ОКБ САПР. В основе СЗИ от НСД Аккорд-АМДЗ [56] котроллер общего назначения и матрица программируемой логики.

Применение универсальной операционной системы в системах защиты информации, позволит унифицировать подходы к обеспечению безопасности при разработке таких систем. Это значительно снизит затраты при разработке средств защиты информации и соответственно сделает такие продукты более [8]

конкурентно способным.

Цель данной работы состоит в сокращении сроков и уменьшении затрат по разработке мобильных информационных технологий. Для достижения этой цели решается общая научно-техническая задача расширения функциональности средств защиты информации, обеспечения переносимости программных средств

защиты информации на различные типы мобильных устройств (на [8]

различные типы технических платформ) и встраивание механизмов защиты от атак с принуждением. Для решения общей задачи необходимо выполнить следующие задачи:

выполнение анализа функциональных возможностей и особенностей реализации существующих мобильных операционных систем и на его основе разработать модель угроз информационной безопасности объекта исследования, архитектуру и программный код универсальной защищенной операционной системы для мобильных систем;

разработка метода аутентификации пользователей стойкого к принуждающим атакам;

разработка метода защитного преобразования передаваемой по открытым каналам информации, стойкий к атакам с принуждением пользователя раскрыть ключ защитного преобразования;

разработка метода защиты программного обеспечения от дизассемблирования;

разработка метода защиты хранимой информации стойкий к атакам с принуждением пользователя раскрыть ключ защитного преобразования.

Глава 2. Разработка методов аутентификации и разграничения доступа

Данная глава посвящена разработке методов аутентификации. В главе будут представлены разработанные алгоритмы усиленной аутентификации пользователей, описаны механизмы разграничения доступа в разработанной операционной системе. Также будут описаны принципы построения систем разграничения доступа.

2.1. Архитектура систем разграничения доступа

Подсистема разграничения доступа к ресурсам является одной из основных частей любой многопользовательской системы. Даже при самом пренебрежительном отношении к безопасности данных в системе, разработчик операционной системы ограничивает доступ пользователей к ресурсам ядра операционной системы. Неприкосновенность ресурсов ядра играет немалую роль в стабильности работы операционной системы. Из вышесказанного следует, что даже при отсутствии в системе каких-либо средств обеспечения безопасности данных, операционная система все равно имеет в своем составе хотя бы минимальную подсистему разграничения доступа пользователей к данным.

Рисунок 13.

Структура системы защиты от угроз нарушения конфиденциальности. [28]

Для эффективной работы любой системы защиты информации необходимо четко определить субъекты и объекты

доступа. [24]

Объект доступа — единица информационного ресурса

АС, [24]

доступ к которой регламентируется правилами разграничения доступа. [7].

Субъект доступа — [65]лицо или процесс, действие которого регламентируется правилами разграничения доступа [7]. [23]

Объектами доступа операционной системы являются: файловые объекты, процессы, объекты памяти, периферия. Все объекты разграничения доступа в операционной системе имеют свой уникальный идентификатор (для объекта оперативной памяти - уникальный адрес, для периферии - уникальный дескриптор, и т.д.). Чтобы сформировать четкие правила разграничения доступа к объектам, необходимо однозначно определить субъекты доступа. На рисунке 13 изображена структура системы защиты от угроз нарушения конфиденциальности. Процесс определения субъекта доступа начинается с идентификации и проверки подлинности. В ходе процесса идентификации будет определен субъект доступа. В операционной системе будет сознан уникальный идентификатор пользователя.

2.1.1. Понятие пользователя системы

Известных моделей разграничения доступа к ресурсам существует несколько: мандатная, ролевая и т.д..

В данном разделе намеренно взгляд остановлен на понятии "пользователь системы". Это связано с тем, что в большинстве популярных операционных систем в вопросах разграничения доступа к ресурсам системы применятся "упрощенная" схема. Пользователь системы - абстрактный, агрегированный субъект доступа в системе. По данной схеме субъектом доступа к системе является пользователь системы. Остальные субъекты (например, процессы) являются производными от родительского субъекта пользователь (процесс запускается от имени пользователя). Несомненно, в рамках определенной технологии могут быть использованы и другие субъекты доступа, но в рамках операционной системы все равно будет предусмотрен некоторый механизм наследования.

Абстрактный объект "пользователь" ассоциируется с "пользователем - человеком". По данной причине вышеуказанная модель наследования прав доступа легка для понимания человеком. В операционной системе могут быть предусмотрены различные группы пользователей (системные, удаленные пользователи и т.д.).

Неотъемлемой частью любой такой модели являются процессы идентификации и аутентификации пользователей в системе.

Идентификация - присвоение субъектам и объектам доступа идентификатора и (или) сравнение предъявляемого идентификатора с перечнем присвоенных идентификаторов. [57]

Аутентификация - проверка принадлежности субъекту доступа предъявленного им идентификатора; подтверждение подлинности. [57]

Так как под пользователем операционной системы в большинстве случаев понимают человека. Принято выделять три основных метода аутентификации пользователя в операционной системе [58]:

по владению некой информации (знание секрета);

по владению уникальным предметом (смарт карты, карты памяти и т.д.);

аутентификация по тому, что является неотъемлемой частью пользователя (биометрическая аутентификация).

Аутентификация по знанию секрета (пароля). Пароли применяется людьми уже очень много времени. По данной причине "парольный" метод аутентификации является наиболее "привычным", простым и понятным для человека. По виду применяемых паролей методы парольной аутентификации делятся на:

методы с применением многоразовых паролей;

методы с применением одноразовых паролей.

Аутентификацию по многоразовому паролю легко реализовать как в программных, так и в аппаратных устройствах. В операционной системе пароли пользователей могут храниться: в отрытом виде (в виде значения пароля), в виде значения односторонней функции (хеш-функции), в зашифрованном виде.

К сожалению, применение многоразовых паролей имеет и ряд очевидных недостатков:
зависимость безопасности системы от сложности выбранного пароля;
зависимость безопасности системы от сохранения пароля в секрете (пользователь не редко записывает пароли на бумажные носители);
возможность подбора пароля (в том числе с применением словарей паролей) злоумышленником;
возможность перехвата (выведывания) п��роля злоумышленником.
Экзотические способы аутентификации (графический пароль и др.) описаны в работах [59], [60].
Одноразовые пароли. Аутентификация в удаленных системах является более сложной задачей [61]. У злоумышленника появляется возможность перехвата передаваемых аутентификационных данных. Применение одноразовых паролей решает ряд проблем парольных систем аутентификации. Суть одноразовых паролей заключается в том, что каждый пароль действителен только при выполнении одной отдельной операции доступа к ресурсу операционной системы. Перехват одноразового пароля не даст злоумышленнику особых возможностей (особенно при применении многофакторной аутентификации).
Для генерации одноразового пароля в протоколах аутентификации на одноразовых паролях используется обычно два параметра:
секретное значение (например, аппаратный идентификатор генератора);
случайное число (полученное от сервера) или значение текущего времени.
Широко известным примером протокола аутентификации на одноразовых паролях является протокол S/KEY (стандарт интернета RFC1760). Данный протокол основан на использовании алгоритмов хеш-функций MD4, MD5.
Одноразовые пароли решают множество проблем классических (многоразовых) парольных систем, но, к сожалению, не все. Протоколы аутентификации становятся еще более уязвимыми при значительном удалении пользователя от операционной системы. Для обеспечения безопасного удаленного доступа, операционная система должна обеспечить конфиденциальность передаваемого по сети пароля. Существует достаточное количество протоколов "удаленной" аутентификации. Данные протоколы аутентификации подразделяют на:
протоколы простой аутентификации, в которых пароль тем или иным образом передается по сети проверяющей стороне;
протоколы строгой аутентификации, в которых пользователь не передает пароль по сети, а тем или иным образом доказывает знание секретного пароля.
"Удаленная" аутентификация в операционной системе подразумевает некий "диалог" между клиентом и операционной системой. Классической является клиент-серверная организация протокола аутентификации. Чаще всего протокол "удаленной" аутентификации является частью протокола удаленного взаимодействия с операционной системы.
Простые методы "удаленной" аутентификации основываются на передаче каким-либо методом пароля (или хеш-значения функции от пароля) от клиента к проверяющей стороне (серверу). Для сокрытия секретного пароля может применяться шифрование (симметричное или ассиметричное) или хеш-функции.
Отличительной особенностью протоколов строгой аутентификации является то, что пользователь доказывает свою аутентичность перед сервером (выполняя ряд действий) без фактической передачи пароля (в любом виде) через канал связи.
В качестве примеров протоколов строгой аутентификации описаны в работах [62], [63]. Популярным является протокол, описанный в стандарте ISO/IEC 9798-2. Данный стандарт предусматривает три варианта

аутентификации:
односторонняя аутентификация с использованием меток времени;
односторонняя аутентификация с использованием случайных чисел;
[12]

взаимная (двусторонняя) аутентификация.
Рассмотрим простой пример односторонней аутентификации с использованием случайных чисел (сторона А аутентифицируется перед стороной В):
A ←B: rb;
A B:Ekrb,B.
Обозначения:
A, B - идентификаторы сторон;
rb - случайное число стороны B;
Ek - функция шифрования на общем секретном ключе к.
Сторона В передает стороне А случайное значение (значение не является секретным). Сторона А шифрует данное значение совместно с идентификатором стороны В на общем симметричном ключе к. Далее сторона А передает зашифрованное значение стороне В. Сторона В расшифровывает полученное значение и проверяет свои идентификатор и случайное число. Если значения совпали сторона В считает сторону А аутентифицированной. Заметим, что передачи секретного значения (пароля) в процессе аутентификации не было.
Аутентификация с использованием уникального предмета. В данном случае пользователь доказывает свою подлинность операционной системой предъявляя уникальный предмет. Разработано большое количество предметов-аутентификаторов (RFID-метки, идентификаторы TouchMemory, USB токены). Безопасность метода аутентификации очень сильно зависит от конкретной реализации уникального предмета. Например, производитель персональных идентификаторов iButton (TouchMemory) компания Dallas Semiconductor гарантирует, что каждый произведенный ею идентификатор имеет уникальный идентификационный номер. Применение данных идентификаторов в схеме аутентификации по уникальному предмету могло бы быть достаточно безопасным. Однако, так как iButton использует для связи известный протокол 1-Wire, китайские производители реализовали аналогичные по функционалу идентификаторы, но с перезаписываемым идентификатором. Аутентификация по уникальному предмету чаще всего используется совместно с другими методами аутентификации (например, аутентификация уникальный предмет и ПИН-код).
Так как для организации биометрической системы аутентификации необходимы специализированные считыватели, данный вид аутентификации не будет рассматриваться в работе.
Идентификации и аутентификации объектов доступа в операционной системе применяется в различных сферах (аутентификация в системе, аутентификация в различных сетевых протоколах таких как FTP). Для этого подсистема аутентификации в операционной системе должна иметь унифицированный (универсальный) характер. Пример унифицированной подсистемы аутентификации можно рассмотреть на основе операционной системы Linux. В данной ОС предусмотрена система подключаемых модулей аутентификации (с англ. PAM – pluggable authentication module)
PAM представляет собой универсальную библиотеку — PAM API, которая используется для подключения модулей аутентификации. Разработчики модуля аутентификации могут, используя библиотеку PAM API, производить разработку своих собственных алгоритмов аутентификации. Подключаемый модуль аутентификации представляет собой скомпилированную библиотеку, описывает функции аутентификации. В операционной системе обычно по умолчанию реализованы модули, в которых описывают протоколы TACACS, RADIUS и др. Решение об аутентификации принимается в подключаемом модуле. Модули подключаются и регистрируются в операционной системе путем записи пути к библиотеке в конфигурационном файле (файл /ets/pam.conf).
Клиентские приложения также взаимодействуют с библиотекой PAM API. При необходимости аутентификации прикладное приложение обращается к библиотеке PAM API. Библиотека читает настройки с конфигурационного файла. Далее производится процедура аутентификации, которая описана в подключаемом модуле.
В работе [64] проведен анализ системы аутентификации на подключаемых модулях. Система PAM имеет удобный, унифицированный и прозрачный механизм аутентификации, который позволяет легко интегрировать новые модули аутентификации.
Применение многофакторной унификации позволяет многократно повысить эффективность системы аутентификации ([65], [66], [67]). Многофакторная аутентификация в концепции PAM также реализуема.
Анализ эффективности применения в мобильной ОС.

Метод защиты: Аутентификация на многоразовых паролях.
Уровень эффективности: средний.
Достоинства: простота применения, понятность конечному пользователю.
Недостатки: защищенность системы аутентификации значительно зависит от сложности применяемого пароля, вероятна атака подбора паролей, многоразовые пароли мало применимы для удаленных систем (отсутствует защита при передаче пароля по сети).
Метод защиты: Усиленная аутентификация
Уровень эффективности: высокий.
Достоинства: высокая эффективность метода (в связи с применение криптографических алгоритмов), адаптирована для применения для удаленных пользователей, отсутствует фактическая передача секрета по сети.
Недостатки: сложность реализации (как клиентская, так и серверная сторона должна быть оснащена средствами вычисления).
Метод защиты: Протоколы одноразовых паролей
Уровень эффективности: средний.
Достоинства: простота реализации алгоритма.
Недостатки: отсутствует защита от атак типа «человек по середине».
2.1.2. Модели разграничения доступа
Если пользователями операционной системы становится более чем один субъект, возникает необходимость в прозрачном, универсальном механизме разграничения доступа к ресурсам системы.
Модель систем дискреционного разграничения доступа. Данная модель характеризуется разграничением доступа между поименованными субъектами и объектами. Каждый субъект, который имеет определенное право, может делегировать его другому субъекту. Доступ субъекта к объекту определяется конечным множеством возможных операций. Для каждой тройки объект-субъект-операция в системе строго определено правило. Данная система может быть реализована в виде матрицы доступа. Столбцы и строки данной матрицы описывают соответственно множество субъектов и объектов. В ячейках данной матрицы указывается тип разрешенной операции. По меньшей мере имеется два подхода к

построению [70] дискреционного управления доступом:

каждый объект системы имеет привязанного к нему субъекта, называемого владельцем. Именно [60] владелец [70] устанавливает права доступа к объекту;

система имеет одного выделенного субъекта администратора, который имеет право устанавливать права владения для всех остальных субъектов системы.

[60]

Мандатное управление доступом. Для реализации этого принципа каждому субъекту и объекту должны сопоставляться специальные метки (мандаты). Каждый объект имеет уровень конфиденциальности, а каждый субъект имеет соответственно уровень доступа. В мандатной системе разграничения доступа должно быть реализовано два основных правила:
«нет записи вниз» субъекту запрещено писать в объект, если уровень доступа субъекта выше уровня конфиденциальности объекта;
«нет чтения вверх» субъекту запрещено читать объект, если уровень доступа субъекта ниже уровня конфиденциальности объекта.
Ролевое разграничение. Основной идеей управления доступом на основе ролей является идея о связывании разрешений доступа с ролями, назначаемыми каждому пользователю. Данная модель разграничения доступа очень понятна человеку. В других сферах деятельности человека часто встречаются роли (работник-начальник, студент-преподователь и т.д.). Каждая роль имеет свой строго определенный набор прав доступа к объектам. Как правило, данный подход применяется в системах защиты СУБД, а отдельные элементы реализуются в сетевых операционных системах.
2.2. Контроль и управление доступом
В разработанной операционной системы контроль и управление доступом в системе представлен двумя подсистемами: подсистемой аутентификации и системы контроля доступа к файловой системе. Применение концепции «все есть файл» позволило значительно унифицировать подходы разграничения доступа к файловым объектам, оперативной памяти и аппаратным устройствам.
2.2.1.

Подсистема аутентификации

Подсистема аутентификации в разработанной операционной системе играет одну из важных ролей. В подсистеме аутентификации ОС [8]

реализован целый ряд функций:
аутентификация локальных пользователей;
аутентификация

удаленных пользователей.

Подсистема аутентификации представляет собой набор библиотек, которые используются для проведения процедур аутентификации. Прототипом структуры подсистемы аутентификации послужила применяемая в операционных системах семейства Linux архитектура PAM (Pluggable Authentication Modules - подключаемые модули аутентификации).

[8]

Ядро ОС
Библиотека аутентификации
Контейнер паролей
Лог файл
Рисунок 14. Подсистема аутентификации пользователей
Подсистема аутентификации локальных пользователей (рисунок 14) обеспечивает классическую аутентификацию пользователей с использование многоразовых паролей. Имеется возможность настройки параметров парольной политики:
длины пароля;
алфавита пароля;
условий блокировки пользователей.
Любые операций аутентификации и изменения аутентификационных данных пользователей пишутся в системный журнал.

Для защиты пользователей от принуждающей атаки в операционной системе был разработан алгоритм аутентификации пользователей с защитой от принуждающей атаки.

2.2.2 Алгоритм аутентификации пользователей

[8]

Далее предложен алгоритм защиты пользователей от принуждающей атаки в процессе аутентификации в ОС. Для защиты пользователя от принуждающей атаки в ОС генерируется дополнительный набор аутентификационных данных.

При этом для защиты пользователей было выбрано достаточно простое и эффективное решение.

[8]

Рисунок 15. Блок-схема процесса аутентификации локальных пользователей
Для этого выбирается имя два имени пользователя с похожим названием (к примеру, user1 и userl).

Один пользователь имеет необходимые права в системе. Второй пользователь имеет минимальные права. В случае ввода резервного пароля (при принуждающей атаке) будет выполнен вход ограниченным пользователем. Алгоритм аутентификации пользователей с защитой от принуждающей атаки [8]

изображен на блок-схеме (рисунок 15).
Процесс аутентификации заключается в следующем:
запросить у пользователя логин и пароль;
вычислить ключ шифрования как хеш-значение логина и пароля;
извлечь из хранилища шифртекст, в соответствии с логином пользователя;
расшифровать сообщение ключом, полученным на шаге 2;
если сообщение состоит из логина" (возможно отличного от введенного) и хеш-значения данного полученного логина" (для проверки результата), то вход в систему под пользователем логин", иначе переход к шагу 1.

При вводе резервного пароля будет расшифровано значение резервного логина и [8]

пользователь зайдет под пользователем с усеченными правами.
Анализ эффективности применения в мобильной ОС
Метод защиты: метод парольной аутентификации с защитой от принуждающей атаки.
Отличительные особенности: защита от принуждающей атаки, возможность применения в любых пользовательских ОС (ориентированность на архитектуру PAM).
2.2.3. Способ аутентификации на одноразовых паролях
В разработанной операционной системе предусмотрена возможность удаленного доступа. В дополнение к стандартным многоразовым паролям в ОС предусмотрена аутентификация с использованием одноразовых паролей.
Наиболее популярный протокол аутентификации на одноразовых паролях S/KEY описан в стандарте интернета RFC1760. Данный протокол основан на хеш-функции MD4. В разработанной операционной системе применен протокол одноразовых паролей, дополнительно усиленный отрицаемым шифрованием, предложенным в [68], что позволяет противодействовать вынуждающей атаке. Алгоритм работает по клиент-серверной схеме. Рассмотрим данный алгоритм более подробно.
Условные обозначения:
N - количество генерируемых паролей;
файл криптографических значений - файл, который содержит значения результатов шифрования, используется при проверке одноразового пароля (хранится на сервере);
R - случайное значение;
C - шифртекст;
ID пользователя - уникальный идентификационный номер пользователя;
h( ) - хеш функция ГОСТ Р 34.11-94 [69];
P - парольная фраза;
S - случайное значение, вектор инициализации.
На первом шаге пользователь выбирает парольную фразу P. Вектор инициализации C - случайное число - позволяет использовать пользователем одну и ту же парольную фразу для нескольких серверов. Далее производится операции "исключающий ИЛИ" значений P и S. Вычисление временных паролей производятся по следующим формулам:
KN=hP xor S, для N=1,
KN=hKN-1, для N>1 .
Так генерируются 2 списка из N паролей, при этом для каждого списка вектор инициализации также должен быть уникальный.
Для генерации файла криптографических значений для N одноразовых паролей выбираются 2N случайных значений. Дополнительно выбирается случайное значение K0 первого криптографического значения C.
MN=hID|R1N, MN=hID|R2N;
K=R1(N-1);
CN=MNK11Kp2p2-1mod p1+ MNK21Kp1p1-1mod p2mod p1p2.
Для получения последнего криптографического значения C используется следующие формулы:
M1=hID|R12, M1=hID|R22;
K=K0;
C1=M1K1NKp2p2-1mod p1+ M1K2NKp1p1-1mod p2mod p1p2.
На стороне клиента хранятся:
список из 2N паролей;
парольная фраза и 2 вектора инициализации (если необходима генерация паролей).
Использование одноразовых паролей начинается с последнего пароля (K1N).
На стороне сервера хранятся:
файл криптографических значений (содержит N криптографических значений);
номер текущего пароля пользователя;
значения простых чисел p1, p2;
временный общий ключ K.
Для дешифрования проверочного сообщения необходимо вычислить следующие значения.
M=(CK-1)K1N-1 mod p1
M=(CK-1)K2N-1 mod p2
В первоначальном состоянии временный ключ K=K0, а номер текущего пароля - 1. Для проверки одноразового пароля K1N сервер производит дешифрование криптографического значения C1 по следующей формуле.
M1=(C1K-1)K1N-1 mod p1
Стоит заметить, что так как сервер не имеет сведений какой серии пользователь предоставил пароль (K1N или K2N) серверу необходимо попробовать дешифровать значение используя и простое число p2.
M1≠(C1K-1)K1N-1 mod p2
В данном случае сервер не сможет дешифровать проверочное сообщение. Для проверки правильности проверочного сообщения сервер сравнивает первые 256 бит проверочного сообщения с вычисленным значением hID. В случае удачной аутентификации в качестве временного общего ключа принимается расшифрованное значение R12 (таким образом, производится завязка последовательности паролей).
В случае если пользователь передаст серверу пароль K2N (при осуществлении вынуждающей отаки), проверка подлинности пройдет успешно. Однако в качестве временного общего ключа будет принято неверное значение (а точнее значение R22) и следующий пароль (K1(N-1) или K2(N-1)) будет признан неправильным.
Анализ эффективности применения в мобильной ОС
Метод защиты: Способ применения отрицаемого шифрования в системах аутентификации на одноразовых паролях.
Отличительные особенности: защита пользователя от принуждающей атаки, возможность реализации в виде генераторов паролей и

парольных карточек (заранее сгенерированных).

2.2.4 Сервис контроля доступа к файловой системе

В работах [70] - [73] представлены системы контролируемого разграничения доступа к файлам. Иерархическая структура файловой системы является идеальным объектом применения моделей разграничения доступа.

Для обеспечения контроля доступа к ресурсам в операционной системе предусмотрено несколько различных механизмов. Одним из таких механизмов является специализированный сервис доступа к файловой системе. Данная системная служба предоставляет пользователям возможности доступа к файловой системе. Конечно, используя функции динамической библиотеки, прикладное программное обеспечение также может получить высокоскоростной доступ к файлам. Однако для осуществления процесса записи в файл процессу необходимо зарезервировать достаточно большой объем оперативной памяти (около 2 килобайт). Для процессов, которые не могут позволить себе такую расточительность (в условиях мобильной системы это большинство процессов) и существует служба доступа к файловой системе. Данная служба имеет весьма похожее строения с сервисом криптопровайдера, который был рассмотрен ранее. На рисунке 16 изображена структурная схема службы доступа к файловой системе.

Рисунок 16. Структурная схема службы доступа к файловой системе

Служба доступа к файловой системе состоит из двух основных блоков:

ядра службы, которое выполняет запросы прикладных процессов (в ядро входит очередь запросов на выполнение операции);

таблицы доступа.

Ядро службы обеспечивает очередное выполнение заданий процессов на операции доступа к файловой системе. Очередность выполнения операций может быть основана на приоритетах процессов в операционной системе. В качестве критерия очередности, в зависимости от настроек операционной системы, могут быть использованы и другие регулярные переменные.

Таблица доступа представляет собой матрицу, в которой отражается взаимосвязь номера процесса операционной, объекта доступа (ссылка на файл), метода доступа (запись, чтение, и т.д.) и время последнего доступа. Данная таблица заполняется ядром сервиса по мере обращения прикладных процессов новым файловым объектам (функция fopen).

Сервис контроля доступа к ФС обеспечивает:

оптимизацию занятой оперативной памяти (в процессе анализа "брошенных файлов");

анализ последствий аварийного завершения работы операционной системы (в зависимости от настройки ОС таблица доступа может писаться в энергонезависимую память и при аварии таблица останется невредимой);

обеспечения безопасности (аудит процессов доступа к файлам, работа сенсоров системы обнаружения вторжения).

К дополнительным функциям сервиса доступа к файловой системе относится:

обеспечение доступа к криптографическим функциям для выполнения операций над защищенными файлами;

унификация доступа к файловой системе.

Анализ эффективности применения в мобильной ОС

Метод защиты: сервис контроля доступа к файловой системе.

Отличительные особенности: эффективное распределение ресурсов, прозрачные правила разграничения доступа (дискреционное разграничение), выполняет роль СЗИ от НСД мобильной ОС (так как в системе применяется принцип "все есть файл"), унификация методов доступа к ФС.

2.3 Выводы ко второй главе

В данной главе была рассмотрена подсистема разграничения доступа, которая является наиболее важной в составе операционной системы. Было рассмотрено такое понятие как пользователь системы (как особый вид субъекта доступа). Далее была описана универсальная модель разграничения доступа, которая включает в себя процессы идентификации, аутентификации и др.

Была проанализирована эффективность алгоритмов аутентификации, применяемых в операционных системах. С целью повышения эффективности функционирования подсистемы были разработаны: Алгоритм аутентификации пользователей системы, способ аутентификации на одноразовых паролях, которые обеспечивают защиту от атак с принуждением.

В данной главе решены следующие задачи исследования:

разработка метода аутентификации пользователей стойкого к принуждающим атакам.

Глава 3. Разработка методов защиты хранимой и передаваемой информации

Данная глава посвящена разработке методов защиты хранимой и передаваемой информации, стойких к атакам с принуждением пользователя раскрыть ключ защитного преобразования. В задачи данной главы также входит выделение и анализ эффективности методов защитного преобразования информации, применяемых в операционных системах. Будут описаны разработанные методы защитного преобразования информации.

3.1. Методы защитного преобразования информации в ОС

В современных операционных система криптографические методы защиты информации используются повсеместно. Существует большое количество реализованных алгоритмов [74]. Для простого и унифицированного доступа к криптографическим функциям в операционной системе реализуется специализированный модуль. В данном модуле (модуль иногда называют криптопровайдером) реализуются различные криптографические функции. Остальные подсистемы операционной системы могут использовать криптопровайдер по своему усмотрению. В рамках криптопровайдера классически реализуются следующие криптографические алгоритмы.

Алгоритмы симметричного шифрования. Алгоритмы шифрования данной категории отличаются высокой производительность. Это свойство делает симметричное шифрование крайне эффективным при обработке большого количества данных (например, при прозрачном шифровании на магнитном диске).

Так как симметричное шифрование имеет неоспоримый недостаток — один

ключ как для шифрования, так и для расшифрования. [12]

На первый взгляд очевидным решением данной проблемы является применение двуключевых (ассиметричных) алгоритмов шифрования. Однако ассиметричное шифрование имеют сравнительно низкую производительность и при этом они имеют достаточно высокую вычислительную сложность. Именно по этой причине ассиметричное шифрование крайне редко реализуется в критопровайдерах.

Для компенсации недостатков симметричного шифрования в криптопровайдерах чаще всего используют алгоритмы обмена ключевой информацией. Такие алгоритмы также иногда называют алгоритмами создания общего ключа обмена. К таким алгоритмам относится протокол Диффи-Хеллмана (протокол описан в [75]). Данный протокол позволяет создать одинаковый ключ обмена у двух абонентов. Общий ключ обмена вычисляется из открытого ключа удаленного абонента и собственного закрытого ключа. После генерации общего ключа шифрования появляется возможность для обмена скоростные симметричные шифры. Вышеуказанная схема (применение протокола генерации общего ключа совместно с симметричным шифрованием) чаще всего применяется в современных средствах криптографической защиты информации. Примером удачного комбинирования ассиметричных и симметричных протоколов может служить протокол "безопасного переходника" SSL (secure sockets layer) [76]. Данный протокол применяется в web браузерах, и комбинирует в себе комбинацию протоколов RSA и DES или RSA и тройной DES.

Хранение ключевой информации. Безопасное хранение ключей в операционной системе является важной задач подсистемы защиты ОС. В процессе эксплуатации криптографических протоколов иногда приходится производить